Notes for POLS 602: Quantitative Political Analysis[*]

Casey Crisman-Cox

Fall 2021

[*] These notes are for personal use only and not for distribution. They are rife with errors, typos, and are not referenced. This is not my own work, but simply notes for me to lecture from. These notes are adapted **heavily** from notes and works by Michael Peress, Dirk Kroese, William Greene and others. Sometimes the adaptations are excessive and extensive. As such I do not claim any original work other than the adaptation. Do not cite or disseminate these notes.

Contents

0 MATH CAMP: Probability and Inference

0.1 Probability basics

Probability is a mathematical tool we use to discuss known uncertainty. For example, in Fall 2020 we wanted to be able to talk intelligently about the probability that say in Joe Biden wins the presidential election: is it likely, unlikely, how likely? Before an event is observed, probability of any given outcome can be either known or unknown, but after an event is observed it becomes a known event (e.g., Biden won the 2020 election, there is no uncertainty over the outcome at this point).

We talk about probability when a yet unobserved event has a random component or chance element. We call a situation where we known all the possible outcomes of a situation (but not which one will occur) a random process. Let S be the set of all possible outcomes from a particular *random process*. We call this particular set the *sample space* as it contains all the outcomes. Each event $A \subset S$ is one or more particular outcomes.

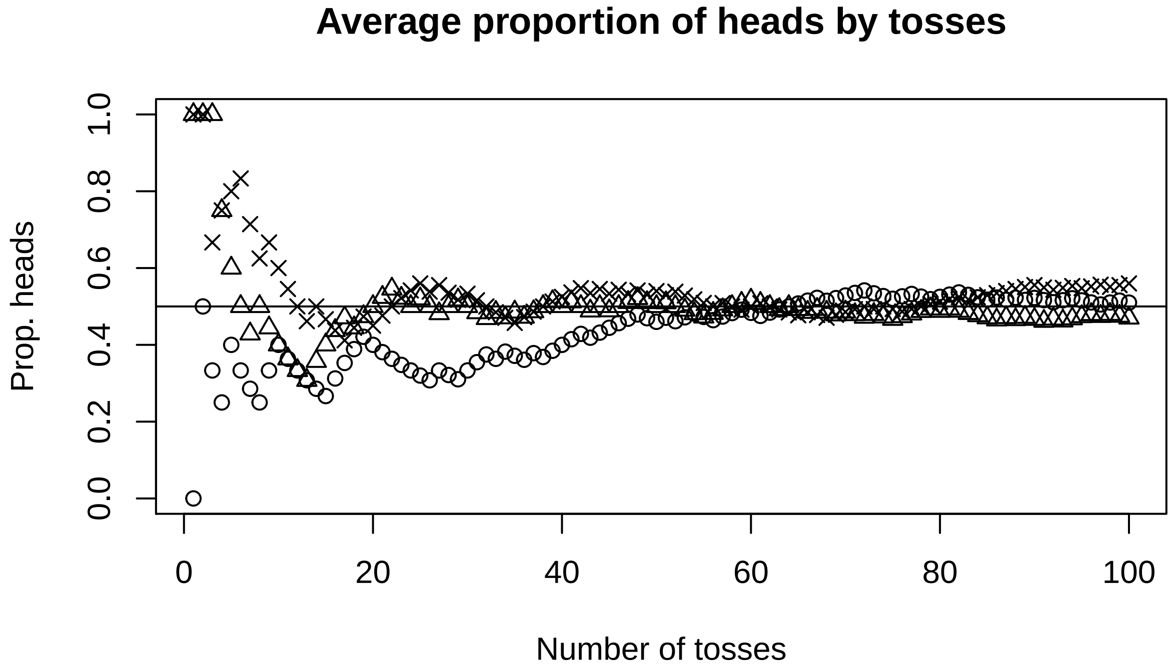
That's all well and good but what **is** probability? What does it mean for an event to have probability 1/3? For the most part, it's easiest to think about probability in terms of long-run frequency. This can be a little bit tricky as often times we want to know the probability of an event that doesn't repeat (2020 election, weather for tomorrow, etc). However, think about flipping a fair coin, if you do it a bunch of times, the proportion of heads will be about 50% (we'll return to this later when we move into the law of large numbers). But let's give it a try:

```r
set.seed(1)
x <- sample(0:1, 100, replace=TRUE)
table(x)
```

```
## x
##  0  1
## 49 51
```

```r
y <- cumsum(x)/(1:100)
plot(y=y, x=1:100, main='Average proportion of heads by tosses',
     xlab='Number of tosses', ylab='Prop. heads', ylim=c(0,1))
abline(h=.5)

# Let's repeat the process a few more times
x2 <- sample(0:1, 100, replace=TRUE)
```

```
y2 <- cumsum(x2)/(1:100)
points(y=y2, x=1:100, main='Average proportion of heads by tosses',
       xlab='Number of tosses', ylab='Prop. heads', pch=2)
x3 <- sample(0:1, 100, replace=TRUE)
y3 <- cumsum(x3)/(1:100)
points(y=y3, x=1:100, main='Average proportion of heads by tosses',
       xlab='Number of tosses', ylab='Prop. heads', pch=4)
```



The point here is that any particular event or series of events is different, but in the long run the observed average equals the probability of a head.

So let's back up to the sample space S. For a single coin toss $S = \{H, T\}$. For two tosses it becomes $S = \{HH, HT, TH, TT\}$. What about a more complicated example: The lifetime of a GMC bus engine in days? Now the sample space takes on all the positive integers $S = \mathbb{N} \cup 0 = \{0, 1, 2, 3, ...\}$. Finally, these can be increasingly complicated as in the income of 10 randomly selected individuals $S = \mathbb{R}^{10}_{+}$ (ten dimensions of positive real numbers). Unlike single tosses of a fair coin, the events within these sample spaces are not all equally likely.

As mentioned before, an event is a particular subset from the sample space. For example

1. If S is all possible rolls of a pair of dice, one event might be rolling a 10 or higher. Let A denote that event, in which case we have $A = \{(4, 6), (5, 5), (5, 6), (6, 4), (6, 5), (6, 6)\}$.
2. The event that our GMC bus engine lives less than 1000 days is $A = [0, 1000)$.

Let's introduce some additional set notation as we think about an example. Suppose we toss a fair coin 3 times and record all three tosses. The sample space then becomes

$$S = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$$

or $\{H, T\}^3$. R has a function called `expand.grid` that's helpful for these types of expansions:

```r
S <- expand.grid(c("H", "T"), c("H","T"), c("H","T"))
print(S)
```

```
##   Var1 Var2 Var3
## 1    H    H    H
## 2    T    H    H
## 3    H    T    H
## 4    T    T    H
## 5    H    H    T
## 6    T    H    T
## 7    H    T    T
## 8    T    T    T
```

Let A be the event that the third toss is a head, this is defined as

$$A = \{HHH, HTH, THH, TTH\}.$$

Let B be the event that the second toss is a head then we have

$$B = \{HHH, HHT, THH, THT\}.$$

We can now consider some set operations

1. The *union* of A and B is denoted $A \cup B$ and is the event that A or B (or both) occurs
2. The *intersection* of A and B is denoted $A \cap B$ and is the event that both A and B occur
3. The *complement* of A is denoted A' or A^c or $\neg A$ and is the event that A does not occur

Additionally, if A is a subset of B then knowing A has occurred implies that B has occurred. Let's go back to the coins.

1. What is the union of A and B? The event that **either** the second or third toss is heads:

$$A \cup B = \{HHH, HHT, HTH, THH, THT, TTH\}$$

2. What is the intersection of A and B? The event that **both** the second or third toss is heads:
$$A \cap B = \{HHH, THH\}$$
3. What is the complement of A? The event that the third toss is not heads (is tails)
$$A' = \{HHT, HTT, THT, TTT\}.$$
4. Is A a subset of B? Is B a subset of A? No. $\{HTH\} \in A, \notin B$ and $\{HHT\} \notin A, \in B$
5. Is $A \cap B$ a subset of A or B? Yes, both.
6. Is $A \cup B$ a subset of A or B? No, but A and B are both subsets of $A \cup B$.

Additionally, we can consider two more results that follow from De Morgan: Let $A_1, A_2, \ldots, A_N$ be a collection of sets then

1. $\left(\bigcup_{i=1}^N A_i\right)' = \bigcap_{i=1}^N A_i'$
2. $\left(\bigcap_{i=1}^N A_i\right)' = \bigcup_{i=1}^N A_i'$

So far so good, we have events that make up a sample space. But we need a little more structure to be able to talk meaningfully about probability as a description of uncertainty. There are some rules to how probability can be used to describe the likelihood or probability of observing an event E. These rules are called the axioms of probability. Let's define probability as a function Pr that satisfies the following

1. $0 \leq \Pr(E) \leq 1$ (all probabilities are bounded between 0 and 1; sets the scale of probability)
2. $\Pr(S) = 1$ Something possible happens
3. For mutually exclusive (disjoint) events $\Pr\left(\bigcup_{i=1}^N E_i\right) = \sum_{i=1}^N \Pr(E_i)$

Note that mutually exclusive means that if E_i occurs than E_j cannot also occur for $i \neq j$. Think of axiom 3 like you would a measure of area in geometry. You can break up a triangle or square into disjoint pieces and sum up the individual areas to get the total. Same in probability (but disjoint is the important aspect).

These axioms do a little work for us. Specifically they tell us some interesting properties of probability

Theorem 1 *Let A and B be events. Then,*

1. $\Pr(\emptyset) = 0$

2. If $A \subset B$, then $\Pr(A) \leq \Pr(B)$

3. $\Pr(A) \leq 1$

4. $\Pr(A') = 1 - \Pr(A)$

5. $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$

We'll prove these in turn

Proof. 1. By Axiom 2 we know that $\Pr(S) = 1$. We also know that any union with the empty set returns the original set as in $S = S \cup \emptyset \cup \emptyset \cup \ldots \cup \emptyset$. We also know that the sample space and the empty set are disjoint. By Axiom 3 it follows that

$$\begin{aligned}\Pr(S) &= \Pr(S) + \Pr(\emptyset) + \Pr(\emptyset) + \ldots + \Pr(\emptyset)\\ 1 &= 1 + \Pr(\emptyset) + \Pr(\emptyset) + \ldots + \Pr(\emptyset)\end{aligned}$$

Which is only true if $\Pr(\emptyset) = 0$.

2. If $A \subset B$ then we can define $B = A \cup (B \cap A')$ (i.e., B is the combination of A with everything in B but not in A). Note that these events are disjoint. As such by Axiom 3 we have

$$\Pr(B) = \Pr(A) + \Pr(B \cap A').$$

By Axiom 1 we know that $\Pr(B \cap A') \geq 0$ and so $\Pr(B) \geq \Pr(A)$.

3. A is a subset of S. By Property 2 $\Pr(A) \leq \Pr(S) = 1$

4. $S = A \cup A'$, which are disjoint by definition. By Axiom 3

$$\begin{aligned}\Pr(S) &= \Pr(A) + \Pr(A')\\ 1 &= \Pr(A) + \Pr(A')\\ \Pr(A) &= 1 - \Pr(A')\end{aligned}$$

5. Once again we rewrite $A \cup B$ as $A \cup (B \cap A')$, which is the union of disjoint events. As such we can write $\Pr(A \cup B) = \Pr(A) + \Pr(B \cap A')$. Note that $B = (A \cap B) \cup (A' \cap B)$ (also disjoint). As such by Axiom 3 we get

$$\begin{aligned}\Pr(B) &= \Pr(A \cap B) + \Pr(A' \cap B)\\ \Pr(A' \cap B) &= \Pr(B) - \Pr(A \cap B).\end{aligned}$$

Substitution then gives us

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$$

□

This gives us some interesting and important first steps into probability. Let's try a few example problems:

Consider a standard deck of 52 cards and answer the following:

1. What's the probability drawing a black king? $(2/52 = 0.04)$
2. What's the probability of drawing either a black card or a king? $(26/52 + 4/52 - 2/52 = 28/52 \approx 0.54)$

Now consider a survey of the American public. Let's say that 52% identify as Democrats. What's the probability that a randomly selected respondent is a Republican?

You thought it was 0.48, didn't you? Are the events exclusive? yes. Are they exhaustive? no. This means that we don't actually have enough information to answer the question. I pulled a trick on you. Let's try one more for you:

0.1.1 Conditional probability

Another bit of definition fun for you is the idea of (in)dependence. Two events are **independent** if knowing the outcome of one does not change the probability of observing the other. In these cases (and only in these cases) we have

$$\Pr(A \cap B) = \Pr(A)\Pr(B).$$

When events are not independent we say they're dependent, but how much does the probability of A change if we know B? This is a question is one of **conditional probability**.

In this situation we know that B has occurred so we can restrict our sample space from S to B *and* we now that A is not relevant event but $A \cap B$ is. So we construct the probability of A given B as:

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

For example, suppose we toss a pair of six-sided dice and someone tells you the total was a ten. What is the probability that exactly one of the dice is a six? Here, B is the event of rolling a ten, how many ways are there to do that?

$$B = \{(4,6),(5,5),(6,4)\}.$$

And A is the even that one six is rolled, how many ways are there to do that?

$$\{(1,6),(2,6),(3,6),\ldots,(6,1),(6,2),\ldots(6,5)\}.$$

So the intersection is $A \cap B = \{(4,6),(6,4)\}$ and for accounting S is 36 equally likely events so we have

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \frac{2/36}{3/26} = \frac{2}{3}.$$

We can simulate it for our own gratification:

```r
set.seed(1)
rolls <- replicate(5000, sample(1:6,size=2, replace=TRUE))
dim(rolls) #columns are rolls; rows are dice
```

```
## [1]    2 5000
```

```r
tens <- colSums(rolls) ==10 #Identify rolls that summed to 10
one6 <- colSums(rolls==6)==1
AcapB <- rolls[,tens&one6]
B <- rolls[,tens]
c(ncol(AcapB)/5000, 2/36)
```

```
## [1] 0.05620000 0.05555556
```

```r
c(ncol(B)/5000, 3/36)
```

```
## [1] 0.08620000 0.08333333
```

```r
ncol(AcapB)/ncol(B)
```

```
## [1] 0.6519722
```

Let's try another famous one: The Monte Hall problem. Once upon a time, there was a game show where the contestant had to pick from one of three doors. Behind one of the doors is a good prize, while behind the other two is nothing. The sequence of events is

1. The contestant picks a door A, B, or C
2. Monte Hall (the host) opens a different door that always reveals nothing (i.e., if the contestant chooses poorly Monte Hall always opens the other bad door, while if the contestant chooses correctly, then Monte Hall chooses to reveal from the two bad door with equal probability).
3. The contestant now has to choose between sticking with their original pick or switching to the other unopened door, what should they do?

Without loss of generality we can assume that the contestant picks A first. This allows us to put some easy notation on the problem. Starting from the contestant choosing A there are four possible outcomes:

1. A_B: the prize is behind A and Monte opens B (1/6)
2. A_C: the prize is behind A and Monte opens C (1/6)
3. B_C: the prize is behind B and Monte opens C (1/3)
4. C_B: the prize is behind C and Monte opens B (1/3)

Again without loss of generality, suppose that Monte opens B . Once Monte opens B, we can now condition on the subscript B. events 1 and 4 are now given (2 and 3 if you want to do C instead).. So the relevant probabilities are sticking with A $\Pr(A_B|A_B \cup C_B)$ versus $1 - \Pr(A_B|A_B \cup C_B)$. So let's start with this: What is the unconditional probability of $\Pr(A_B)$? Well there's 1/3 change of any given door having the prize at the start and then Monte independently chooses between B and C so it becomes $1/3 \times 1/2 = 1/6$. The same logic applies to A_C. For B_C we have a 1/3 chance that B is correct and Monte *has* to open C with certainty so $\Pr(B_C) = 1/3$ (same for C_B). Now we can do some substitution

$$\begin{aligned}
\Pr(A_B|A_B \cup C_B) &= \frac{\Pr(A_B \cap (A_B \cup C_B))}{\Pr(A_B \cup C_B)} \\
&= \frac{\Pr(A_B)}{\Pr(A_B) + \Pr(C_B)} \\
&= \frac{1/6}{1/6 + 1/3} = \frac{1}{3}.
\end{aligned}$$

Staying put wins 1 in 3 times. So switching wins 2 in 3 times: always switch. The intuition behind this problem is that Monte reveals some information because he has to open a bogus door. If you stand pat your chance stays 1/3 (the *a priori* chance of getting it right), but now there's only two options. We can simulate if you don't believe

```
monte.hall <- function(switch=TRUE){
  true.door <- sample(1:3, size=1)
  pick <- sample(1:3, size=1)
  reveal <- ifelse(pick==true.door,
                   sample((1:3)[-true.door], size=1),
                   (1:3)[-c(true.door,pick)])
  if(switch){
    pick <- (1:3)[-c(pick, reveal)]
  }
  win <- (true.door==pick)
  return(win)
}

with.switch <- replicate(10000, monte.hall())
without.switch <- replicate(10000, monte.hall(switch=FALSE))
c(mean(with.switch), mean(without.switch))
```

```
## [1] 0.6733 0.3387
```

Pretty cool, right?

The next thing you should know about is the **product rule** of probability. Specifically, we can rearrange the conditional formula such that

$$\Pr(A \cap B) = \Pr(A) \Pr(B|A).$$

Note that under independence $\Pr(B|A) = \Pr(B)$ so this is generalizes our independence product rule from above. We can actually generalized this further to get

Theorem 2 (Product rule) *Let* $A_1, A_2, \dots A_N$ *be a sequence of events with* $\Pr(A_1 \cap, \dots, \cap A_{N-1}) > 0$. *Then*

$$\Pr(A_1 \cap, \dots, \cap A_N) = \Pr(A_1) \Pr(A_2|A_1) \Pr(A_3|A_1, A2) \dots \Pr(A_N|A_1, A_2, \dots, A_{N-1}).$$

A sketch of this proof for $N = 3$ works as follows. We start by applying the conditional probability formula to both $\Pr(A_2|A_1)$ and $\Pr(A_3|A_1 \cap A_2)$ and rearranging some terms to get two equations

$$\begin{aligned}
\Pr(A_2 \cap A_1) &= \Pr(A_1)\Pr(A_2|A_1) \\
\Pr(A_2 \cap A_1) &= \Pr(A_3 \cap A_1 \cap A_2)/\Pr(A_3|A_1 \cap A_2)
\end{aligned}$$

Combining them using like terms gives us

$$\begin{aligned}
\Pr(A_3 \cap A_1 \cap A_2)/\Pr(A_3|A_1 \cap A_2) &= \Pr(A_1)\Pr(A_2|A_1) \\
\Pr(A_3 \cap A_1 \cap A_2) &= \Pr(A_1)\Pr(A_2|A_1)\Pr(A_3|A_1 \cap A_2).
\end{aligned}$$

Repeating this process for larger N will show the result.

Let's try another problem. Suppose we draw three balls from a bowl with 5 white and 5 black balls without replacement. What is the probability of drawing three black balls? There are three events to consider here, let B_i be the event that the ith draw is black for $i = 1, 2, 3$. We are interested in $\Pr(B_1 \cap B_2 \cap B_3)$. Applying the product rule we see that

$$\begin{aligned}
\Pr(B_1 \cap B_2 \cap B_3) &= \Pr(B_1)\Pr(B_2|B_1)\Pr(B_3|B_1 \cap B_2) \\
&= \frac{1}{2} \times \frac{4}{9} \times \frac{3}{8} \approx 0.0833
\end{aligned}$$

Conditional probabilities are hella cool, but often times we want to able to transition from $\Pr(A|B)$ to $\Pr(B|A)$. Note that these are generally not a complement as the conditioning outcome changes (and as such the how denominator). So how do we move from one to the other? We will start by writing both of these out and looking for common terms:

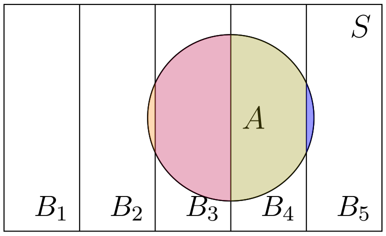
$$\begin{aligned}
\Pr(A \cap B) &= \Pr(A|B)\Pr(B) \\
\Pr(A \cap B) &= \Pr(B|A)\Pr(A).
\end{aligned}$$

You might see where this is going, but let's finish it up:

$$\begin{aligned}
\Pr(B|A)\Pr(A) &= \Pr(A|B)\Pr(B) \\
\Pr(B|A) &= \frac{\Pr(A|B)\Pr(B)}{\Pr(A)}.
\end{aligned}$$

This little switch-a-roo is called "Bayes' Rule." It will come up *a lot* in your graduate training.

We can explore this a little further, by noting that many times we may not directly know unconditional probability of A in these situations. So s continue our exploration of dependent events by considering a sequence of events $B_1, \ldots, B_N$ that partition the sample space as shown:



A **partition** is a set of disjoint events whose union is S. Note that we can find the total probability of A by taking the sum of

$$\Pr(A) = \sum_{i=1}^{N} \Pr(A \cap B_i)$$

or by applying conditional probability to the above to get

$$\Pr(A) = \sum_{i=1}^{N} \Pr(A|B_i)\Pr(B_i).$$

This sum is known as the **law of total probability**. We can use this result then to rewrite an expanded version of Bayes' rule

$$\Pr(B_j|A) = \frac{\Pr(A|B_j)\Pr(B_j)}{\sum_{i=1}^{N}\Pr(A|B_i)\Pr(B_i)}.$$

For example, consider the following Cancer statistics:

1. A random woman in her 40s has about a 1.5% chance of having breast cancer
2. Mammograms will correctly identify cancer in about 87% of women who have cancer
3. Mammograms will produce a false positive about 10% of the time for this age group

What is the probability that a women in her 40s who tested positive on a mammogram actually has breast cancer?

Let A be the event that the woman has cancer and B be the event that she tested positive. What do we want to know? $\Pr(A|B)$. What are we need?

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$$

1. $\Pr(A) = 0.015$
2. $\Pr(B|A) = 0.87$
3. $\Pr(B|A') = 0.10$
4. $\Pr(A') = 0.985$

Notice that $\Pr(B)$ is not in the given information, what do we do? Apply the law of total probability

$$\begin{aligned}\Pr(B) &= \Pr(B|A)\Pr(A) + \Pr(B|A')\Pr(A')\\ &= 0.87(0.015) + 0.10(0.985)\\ &\approx 0.112.\end{aligned}$$

Now we can roll

$$\begin{aligned}\Pr(A|B) &= \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|A')\Pr(A')}\\ &= \frac{0.87(0.015)}{0.112}\\ &= \frac{0.013}{0.112}\\ &\approx 0.12\end{aligned}$$

So the risk of that this woman has cancer has risen 8-fold but the balance of probability is still against cancer. How many tests positive tests in a row do we need before we're confident? Reapply Bayes rule, but what's the new baseline risk? 0.12 Let A_i be the number of positive tests in a row we can now rewrite

$$\Pr(A_i|B) = \frac{\Pr(B|A_i)\Pr(A_i)}{\Pr(B|A_i)\Pr(A_i) + \Pr(B|A_i')\Pr(A_i')}$$

Where $\Pr(A_i)$ increases every time we get a new positive. Let's program this

```r
base <- 0.015
true.pos <- 0.87
false.pos <- 0.10
bayes <- function(base, true.pos, false.pos){
  new.prob <- (true.pos * base)/
    (true.pos*base + false.pos*(1-base))
  return(new.prob)
}
```

```
update <- rep(0,5)
for(i in 1:5){
  base <- bayes(base, true.pos, false.pos)
  update[i] <- base
}
information <- update/c(0.015, update[1:4])
par(mfrow=c(1,2))
plot(y=update,x=1:5, xlab="Number of positive tests", ylab="Updated cancer risk")
plot(y=information,x=1:5, xlab="Number of positive tests",
     ylab="Relative increase in cancer risk")
```

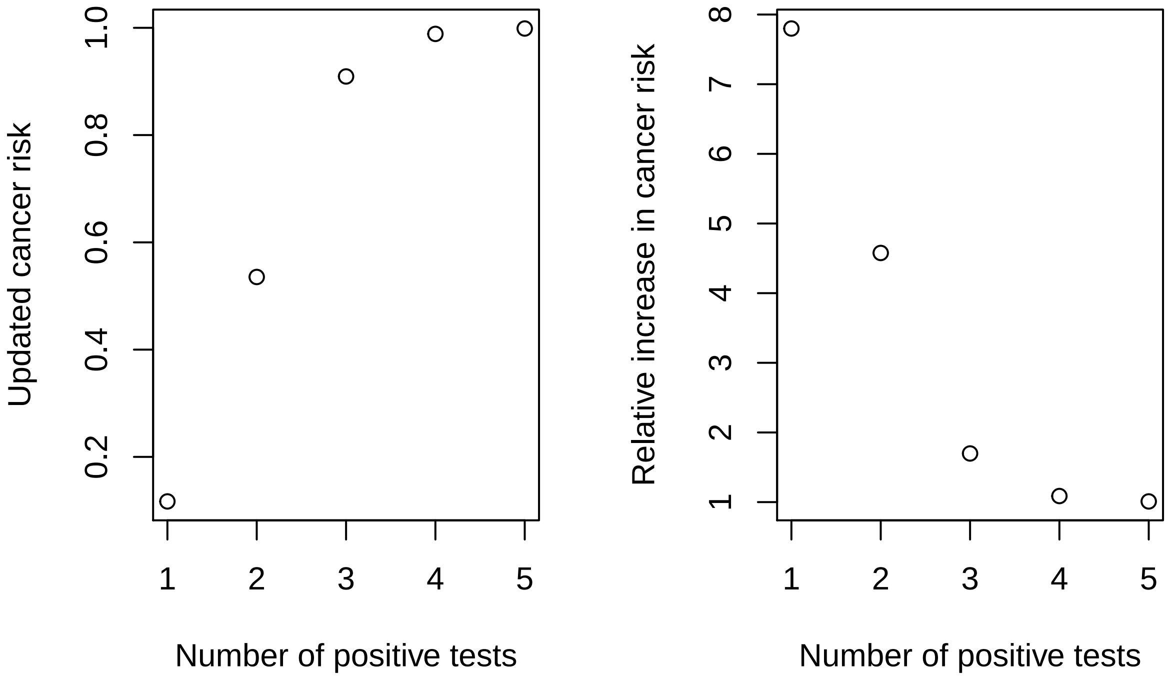


Figure 1: Bayesian updating

After 2 tests the balance of evidence favors cancer, but it's pretty even (54-46). By 3 positives, we're pretty confident (91%). This would obviously get more complicated if we considered a sequence of positive and negative results, but let's not worry about that right now.

0.1.2 Independent events

We'll pause our discussion of dependent events for a moment to return to independent events. Obviously, independence makes things easier as

$$\Pr(A \cap B) = \Pr(A)\Pr(B|A) = \Pr(A)\Pr(B).$$

This product rule for independence holds for any number of independent events and is both a necessary and sufficient condition for independence.
Often times we will assert independence and then have to justify it (with varying degrees of success). A lot of applied data analysis is either justifying independence or figuring out to include dependencies (and frequently both at the same time).

Let's consider a specific example here. Consider an experiment where we toss a coin N times. The sample space S becomes all possible ways that sequence of N tosses can go:

$$S = \{(0,0,\ldots,0),\ldots,(1,1,\ldots,1)\},$$

where 0 denotes Tails and 1 denotes Heads. Let A_i be the event that the ith toss is a heads. The tosses are now independent and identically distributed (i.i.d.). In this specific case $\Pr(A_i) = p$ for all i. We can say a little more about the probability function that defines this experiment. Let's start by saying we want to know what's the probability of have k heads and the $N-k$ tails? By independence that becomes

$$\Pr(1,1,\ldots,1,0,0,\ldots,0) = p^k(1-p)^{N-k}$$

Let B_k be the event that there are a total of k heads in the N tosses. This becomes a union of exclusive events, right? To see this consider three tosses and think about if $k=2$, every row with two heads is exclusive from the others.

```
expand.grid(1:0, 1:0, 1:0)
```

```
##   Var1 Var2 Var3
## 1    1    1    1
## 2    0    1    1
## 3    1    0    1
## 4    0    0    1
## 5    1    1    0
## 6    0    1    0
## 7    1    0    0
## 8    0    0    0
```

So the probability of $\{1, 1, 0\}$ is $p^2(1-p)$, but there are three ways to get two heads so for this particular case we have

$$\Pr(B_2) = 3 \times p^2(1-p)^1.$$

Or more generally

$$\Pr(B_k) = \binom{N}{k} p^k (1-p)^{N-k}.$$

Where the $\binom{N}{k}$ operator asks how many ways where there to get k success in N trials? this is called the binomial or the choose function:

$$\binom{N}{k} = \frac{N!}{(N-k)!k!}$$

The probability we derived

$$\Pr(B_k) = \binom{N}{k} p^k (1-p)^{N-k}, k = 0, 1, 2, \ldots, N$$

is called binomial distribution: our first named distribution

0.2 Random variables and distributions

0.2.1 Random variables basics

We turn our attention to random variables, which are a major building block of everything else we do hereon. A random variable is a function $f : S \to \mathbb{R}$. Intuitively it maps from outcomes in the sample space into numerical outcomes. For example, imagine that we once again toss a pair of dice and record the sum. The sample space remains the 36 paired outcomes denoted (with a slight abuse of notation)

$$S = \{(i, j)\}_{i,j=1}^{6}$$

while the random variable X that denotes the sum. Here X is a function $i + j$ that maps from S into the integers from 2 to 12. As you know there will be several different ways to get most outcomes. For instances there are 5 different ways to get an 8 and so $\Pr(X = 8) = \frac{5}{36}$. You can verify for yourself that there is an easy way to write out the distribution of X such that

$$\Pr(X = x) = \frac{6 - |7 - x|}{36}, x = 2, 3, \ldots, 12$$

While random variables are, and always will be, functions, it is often useful to think about them in terms of outcomes of an observed experiment. The intuition here is that we have an experiment in mind and the random variable will record the outcome when we run in the future. In the present we are simply thinking about all the things that might happen when we run the experiment and how likely those different outcomes are. We tend to use a capital letter to denote a random variable X, Y, Z and lower case letters for specific outcomes x, y, z.

Recall from the above that we had an experiment where we flipped a coin N times and recorded the number of heads. This is an example of a random variable. Let X be the total number of heads from N flips. Then we know from what we did last time that

$$\Pr(X = x) = \binom{N}{k} p^x (1 - p)^{N-x}, x = 0, 1, \ldots, N.$$

As mentioned before this is the binomial distribution. Which is to say that X is a binomial random variable. The binomial is the first of several **probability distributions** we will encounter. To satisfy yourself that X is a function note that it takes the sample space (raw coin tosses) and sums up the number of heads. Almost anything you want to study that is of interest can be thought of as a random variable (even when the sample space is perhaps too large to produce).

1. The amount of spending on a campaign
2. Whether of not there is a civil war in a given country-year
3. The percentage of the Black vote will a candidate receive

So long as there is uncertainty over what the outcome will be before observation you can imagine it as a random variable. There are two main kinds of random variables:

1. Discrete random variables take on a set of specific, countable values (i.e., integers, categories, years)
2. Continuous random variables are measured in intervals (i.e., inches, years, dollars) There can be ambiguity here in cases where its unclear if something discrete is better/easier thought of continuously.

0.2.2 Distributions

There are several ways to think about probability distributions for a random variable X. Specifically we can think about specific outcomes $\Pr(X = x)$ or intervals $\Pr(a \leq X \leq b)$. One common way to proceed is in terms of the **cumulative distribution function** (CDF) defined as a function $F : \mathbb{R} \to [0, 1]$ such that $F(X) = \Pr(X \leq x)$. A few interesting facts about F that follow from the axioms of probability are

1. F is weakly increasing $F(x) \leq F(x')$ for all $x \leq x'$.
2. F does not have to be continuous, but it will be right-hand continuous $\lim_{h \to 0^+} F(x + h) = F(x)$
3. $\lim_{x \to \infty} F(X) = 1$ and
4. $\lim_{x \to -\infty} F(X) = 0$
5. $0 \leq F(x) \leq (1)$
   We will not prove these for time, but you can look them up if you're interested.

The CDF allows us to quantify the probability of observing any interval result. Specifically,

$$\Pr(a < X \leq b) = F(b) - F(a).$$

To see this note the following

$$\begin{aligned}
\Pr(X \leq b) &= \Pr(X \leq a \cup (a < X \leq b)) \\
&= F(a) + \Pr(a < X \leq b) \\
\Pr(a < X \leq b) &= F(b) - F(a)
\end{aligned}$$

The events on the right-hand side of the first line are disjoint which allows us to use the sum rule. After that we just rearrange terms. How we include $\Pr(X = a)$ will depend on whether the random variable is continuous or discrete.

0.2.3 Discrete distributions

A random variable X has a discrete distribution if X is a discrete random variable. In other words, if there is a countable set of outcomes (finite or not) that X can take on then X is discrete and follows a discrete distribution. The **probability mass function** (PMF) f

describes the probability of each event such that

$$\begin{aligned}
f(x) &= \Pr(X = x)\\
F(X) &= \Pr(X \leq x)\\
\Pr(X = A) &= \sum_{i:x_i \in A} f(x_i)\\
1 &= \sum_i f(x_i), \forall x_i \in \text{supp}(X).
\end{aligned}$$

Where $\text{supp}(X)$ is the **support** of X (all values that X can take on with positive probability).

Suppose you roll two dice and let X be the maximum of the two rolls. What's the distribution of X?

```r
library(xtable)
rolls <- expand.grid(1:6, 1:6)
xout <- rep(0,6)
for(i in 1:6){
  xout[i] <- nrow(rolls[rolls[,1]==i & rolls[,2]<=i |
                        rolls[,2]==i & rolls[,1]<=i,])/nrow(rolls)
}
xout <- matrix(xout, nrow=1)
colnames(xout) <- 1:6
rownames(xout) <- c("$f(x)$")
print(xtable(xout, caption="PMF of maximum roll"),
      sanitize.text.function = function(x){x}, comment = FALSE)
```

	1	2	3	4	5	6
$f(x)$	0.03	0.08	0.14	0.19	0.25	0.31

Table 1: PMF of maximum roll

```r
sum(xout)
```

```
[1] 1
```

So the probability that the maximum is 3 or less? $\sum_{x=1}^{3} f(x) = 0.25$.

0.2.4 Continuous distributions

A random variable X has a continuous distribution if X is a continuous random variable such that there is a positive function f such that for all a and b

$$\Pr(a < X \leq b) = F(b) - F(a) = \int_a^b f(u)du$$

and

$$\int_{-\infty}^{\infty} f(u)du = 1.$$

Now f is called a **probability density function** (pdf), and the cdf $F(x) = \Pr(X \leq x) = \int_{-\infty}^{x} f(u)du$. Working that in reverse we can also note that $f(x) = D_x F(x)$.

The caveat of interest here is that f is not a probability it is a probability density. For a continuous variable X, $\Pr(X = x) = 0$ for all x. You'll never draw/observe any exact number from a continuous distribution.

Suppose that X represents a randomly drawn number from the interval $[0, 2]$ where every number is equally possible. What is the pdf and cdf of X?

If every value is "equally possible" then the pdf will be a constant so we need to know what constant c satisfies

$$\begin{aligned}
1 &= \int_0^2 cdu \\
&= cu|_0^2 \\
&= 2c - 0 \\
c &= 1/2
\end{aligned}$$

Which means that the pdf is

$$f(x) = \begin{cases} 1/2 & 0 \leq x \leq 2 \\ 0 & \text{otherwise.} \end{cases}$$

Applying the integral from 0 to x gives us the CDF

$$\begin{aligned}
F(x) &= \int_0^x 1/2du \\
&= u/2|_0^x \\
&= x/2
\end{aligned}$$

or more precisely

$$F(x) = \begin{cases} 0 & x < 0 \\ x/2 & 0 \leq x \leq 2 \\ 1 & x > 2 \end{cases}$$

This particular distribution is called a **uniform** distribution. Generally it is defined by parameters a and b such that

$$f(x) = \begin{cases} 1/(b-a) & a \leq x \leq b \\ 0 & \text{otherwise.} \end{cases}$$

and

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{cases}$$

0.2.5 Expected value

The functions F and f contain all the information housed within a random variable. However, we are often interested in particular descriptions of random variable. Most commonly we want to know the expected value and variance. We will consider these in turn.

The expected value or expectation or a random variable is a weighted average of all the values that X can take, specifically we say that for a discrete X

$$\mathrm{E}[X] = \sum_{x \in \operatorname{supp}(X)} x \Pr(X = x)$$

Note that the expected value does not have to be in the support of X. Consider rolling a fair die once. Each roll is equally likely and we get $\sum_{x=1}^{6} \frac{x}{6}$.

```
sum(1:6/6)
```

```
## [1] 3.5
```

Which is outside the support of X.

One easy way to think about expected values is in terms of expected payoff. Suppose we play a game of dice where I pay you X dollars per round where X is the sum of two randomly tossed dice. What entry fee y would make this game even on average?

Here we need to set

$$
\begin{aligned}
d &= \mathrm{E}[X] \\
&= 2\Pr(X=2) + 3\Pr(X=3) + \ldots + 12\Pr(X=12)
\end{aligned}
$$

```r
rolls <- expand.grid(1:6, 1:6)
pr <- table(rowSums(rolls))/36 #calculate the probabilities
Ex <- sum(2:12*pr)
print(Ex)
```

```
## [1] 7
```

For a $7 entry fee the game is "fair." What does this mean? Remember this is about long-run thinking. If we play this game an infinite number of times the payoff will be 0 for a seven dollar fee.

```r
set.seed(1)
X <- replicate(1000,sum(sample(1:6, size=2, replace=TRUE)))
y <- X - 7
par(mfrow=c(1,2))
plot(y=y, x=1:1000, xlab="Number of plays", ylab="Profit")
hist(y, xlab="Payout", freq=FALSE, main="")
```

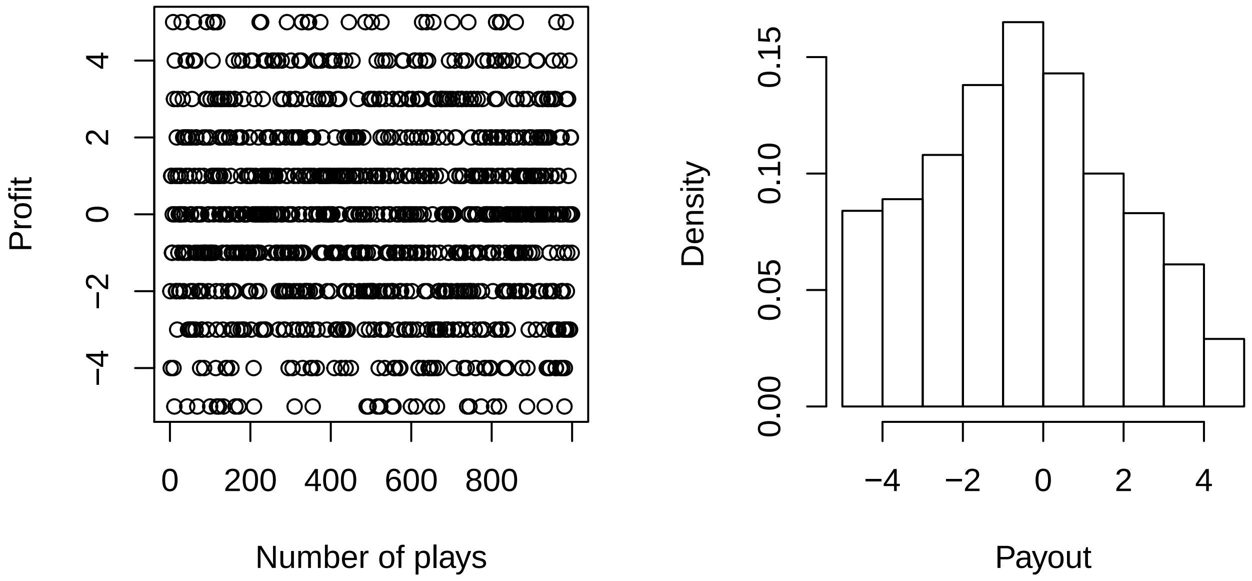


Figure 2: Distribution of outcomes from the game

As you can see, at any given moment you may be up or down, but the payoff is centered around that expected value and X is centered around it's expected value. "Centered" however is a slight misnomer here, as an expected value is really about the "center of mass" of the random variable. Suppose we redid the above but I paid out the maximum rather than the sum. Redefine X to be the maximum and now we have to balance

$$\begin{aligned} d &= \mathrm{E}[X] \\ &= 1\,\Pr(X=1) + 2\,\Pr(X=2) + \ldots + 6\,\Pr(X=6) \end{aligned}$$

```r
Ex <- sum(xout*1:6) #xout was defined above for maximum
print(Ex)
```

```
## [1] 4.472222
```

```r
set.seed(1)
X <- replicate(1000,max(sample(1:6, size=2, replace=TRUE)))
y <- X - Ex
par(mfrow=c(1,2))
plot(y=y, x=1:1000, xlab="Number of plays", ylab="Profit")
hist(y, xlab="Payout", freq=FALSE, main="")
```

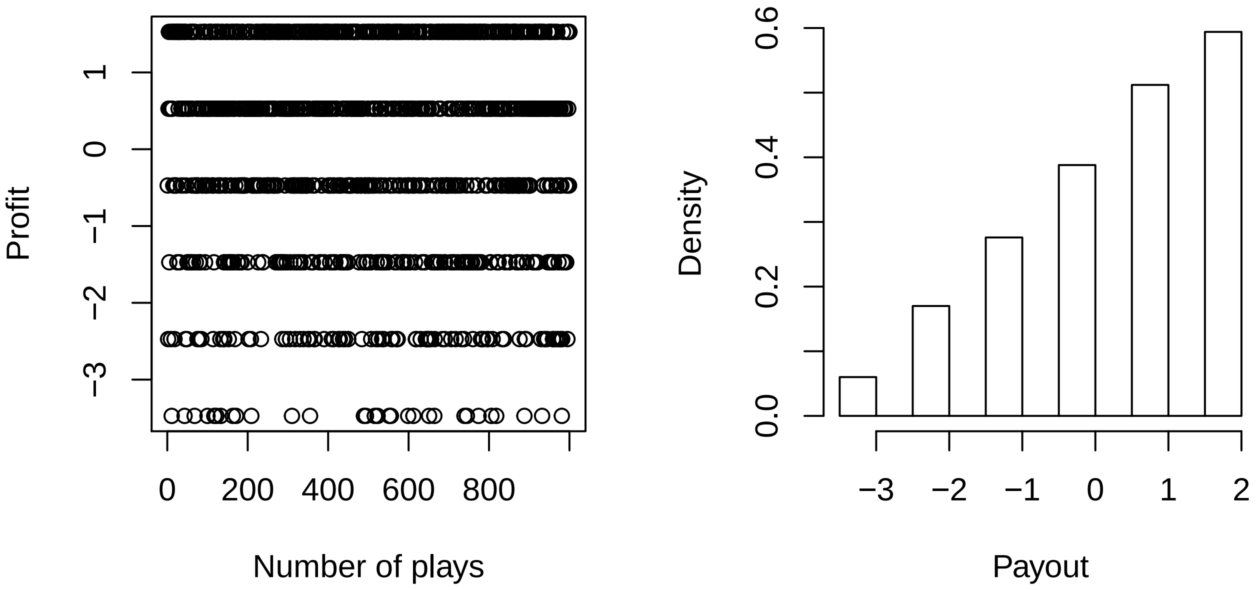


Figure 3: Distribution of outcomes from the new game

Note here that the expected value of y is 0 (more on this in a second), but it's not peaked or centered there. This is because the distribution of the maximum is skewed, not symmetric. When a distribution is symmetric its expected value will be close to the middle (if it exists).

Otherwise, it'll be "pulled" towards longer tail ("left-skewed" here) and lower than the peak value.

For continuous random variables the expectation is similarly defined. Such that for continuous X we have

$$\mathrm{E}[X] = \int_x x f(x) dx.$$

One theorem we'll use **a lot** is the following

Theorem 3 **(*The Law of the Unconscious Statistician)*** *Let X with pmf/pdf f be a random variable whose expected value exists. For any real valued function g*

$$\mathrm{E}[g(X)] = \sum_x g(x) f(x)$$

for discrete X and

$$\mathrm{E}[g(X)] = \int_x g(x) f(x) dx$$

for continuous X.

Proof. We will prove the discrete case and leave the continuous case on my assertion. Let $Y = g(X)$ with pmf f_Y. What do we know about f_Y?

$$\begin{aligned} f_Y(y) &= \Pr(Y = y) \\ &= \Pr(g(X) = y) \\ &= \sum_{x:g(x)=y} \Pr(X = x) \\ &= \sum_{x:g(x)=y} f(x). \end{aligned}$$

Now we can look at the expected value of Y

$$\begin{aligned} \mathrm{E}[Y] &= \sum_y y f_Y(y) \\ &= \sum_y y \sum_{x:g(x)=y} f(x). \\ &= \sum_y \sum_{x:g(x)=y} y f(x). \\ &= \sum_x g(x) f(x). \end{aligned}$$

□

Let's give it a try: Let X be the outcome of a rolling a single die. What is the expectation of X^2?

$$\mathrm{E}[X^2] = \sum_{x=1}^{6} x^2 f(x) = \frac{1}{6}\sum_{x=1}^{6} x^2$$

```r
sum((1:6)^2 * (1/6))
```

```
## [1] 15.16667
```

Two highly relevant facts fall out of this theorem are that expectations are linear:

1. $\mathrm{E}[aX + b] = a\,\mathrm{E}[X] + b$
2. $\mathrm{E}[g(X) + h(X)] = \mathrm{E}[g(X)] + \mathrm{E}[h(X)]$

These both follow directly from the fact that sums and integrals can be split along summation signs.

One other thing to note is the median of a random variable is the value m such that $\Pr(X \leq m) = \Pr(X \geq m) = 0.5$ The value of m may be either a single value or a pair that co-define the median (median of a single die roll: $\{3, 4\}$; median of the sum of two dice $\{7\}$). Sometimes people average a pair median together; let the circumstances dictate for you.

0.2.6 Variance

Another interesting descriptor of a random variable is the **variance**, which measures the spread of a random variable around its expected value. Specifically,

$$\mathrm{Var}(X) = \mathrm{E}[X^2] - \mathrm{E}[X]^2 = \mathrm{E}[(x - \mathrm{E}[X])^2]$$

The main properties/restatements we'll consider right now is that

$$\begin{aligned}
\mathrm{Var}(c) &= 0 \\
\mathrm{Var}(aX + b) &= a^2\,\mathrm{Var}(X) \\
\mathrm{Var}(g(X)) &= \mathrm{E}[(g(x) - \mathrm{E}[g(X)])^2] \\
&= \mathrm{E}[g(X)^2] - \mathrm{E}[g(X)]^2]
\end{aligned}$$

where c, a, and b are constants.

The variance reports the squared distance that each outcome is from the expected value. Squared distance is nice because the direction of the distance doesn't matter, but squared units can be difficult to understand/interpret/use. We frequently look at the standard deviation instead which is defined as

$$\mathrm{sd}(X) = \sqrt{\mathrm{Var}(X)}.$$

Standard deviations are nice because they are on the same units as X.

0.2.7 Moments

Expected value and variance are examples of **moments**. A moment is a descriptive aspect of a random variance and can be used to fully describe a distribution. Specifically the **moment generating function** of a random variable X is defined as

$$M(s) = \mathrm{E}[e^{sX}],$$

where $e = 2.718\ldots$ is a mathematical constant with the following properties:

$$\begin{aligned}
e^x &= \exp(x) \text{ just another to write it} \\
e^0 &= 1 \\
e^1 &= e \\
e^{-x} &= 1/e^x \\
\exp(x)^y &= e^{xy} \\
e^{x+y} &= e^x e^y \\
e^{x-y} &= e^x/e^y \\
\exp(x) &> 0, \forall x \in \mathbb{R} \\
D_x e^x &= e^x > 0 \text{ Strictly increasing} \\
D_x^2 e^x &= e^x > 0 \text{ Convex function} \\
\lim_{N\to\infty}\left(1 + \frac{x}{N}\right)^N &= e^x
\end{aligned}$$

```
curve(exp(x), from=-4, to=4)
```

Back to the main event: Moment generating functions have a uniqueness property, wherein we can say that X and Y have identical moment generating functions if and only if they follow the same distribution.

Individual moments come from evaluating derivatives of the moment generating functions

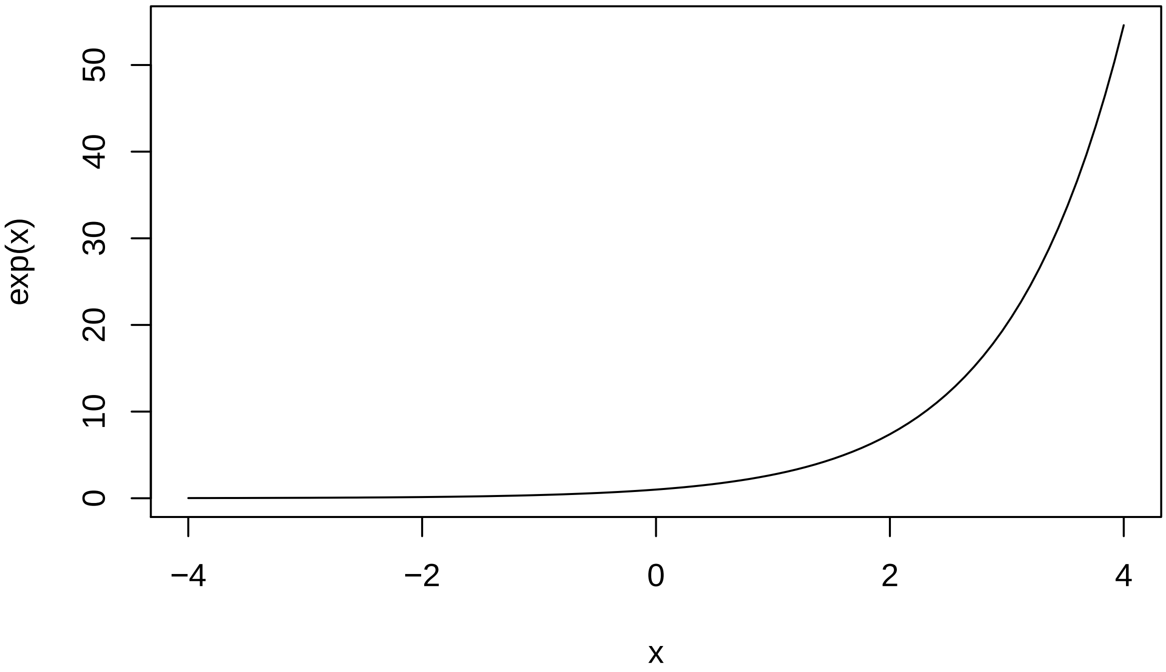


Figure 4: The function e^x

(mgfs) at 0 (if these derivatives exist). Specifically the rth (raw) moment is defined as,

$$E[X^r] = D_s^r M(0).$$

Such that the expected value and variance are given by

$$\begin{aligned} \mathrm{E}[X] &= D_s M(0) \\ \mathrm{Var}(X) &= D_s^2 M(0) - (D_s M(0))^2 \end{aligned}$$

One useful result we can cover is the following

Theorem 4 *Let X be a random variable with mgf $M_X(s)$, and let $Y = aX + b$ where a and b are constant. Then the mgf of Y is given by*

$$M_Y(s) = e^{bs} M_X(as).$$

Proof. By definition

$$M_Y(s) = \mathrm{E}[e^{sY}] = \mathrm{E}[e^{s(aX+b)}] = \mathrm{E}[e^{saX} e^{Sb}] = e^{sb}\,\mathrm{E}[e^{saX}] = e^{sb} M_X(as),$$

which is what we had to show. □

This result allows us to talk about central moments. The central moments are defined by first centering the distribution. The kth **central moment** of X is defined as

$$E[(X-\mu)^k] = D_s^k e^{-\mu k} M(0).$$

Note that the first central moment will always be 0 ($\mathrm{E}[X]-\mu$), but the second central moment will be the variance.

Two additional qualities of distributions include the skewness

$$\text{Skew} = \frac{E[(X-\mu)^3]}{\sigma^3}$$

and kurtosis

$$\text{Kurtosis} = \frac{E[(X-\mu)^4]}{\sigma^4}.$$

The former defines how asymmetric the distribution is with 0 being symmetric, positive values being right skewed (long right tail), and negative values being left skewed (long left tailed). The latter defines how fat the tails are, or how weird outliers are. When tails are fat outliers are less unusual (**leptokurtic**; kurtosis > 3; e.g., t). When tails are thin outliers are very unusual (**platykurtic**; kurtosis < 3; e.g., Uniform). A distribution is **mesokurtic** (normal level of outliers) when kurtosis about 3 (e.g., Normal).

Another result that will be helpful to us later is the following

Theorem 5 *Let $X_1, \ldots, X_N$ be a sequence of independent random variable with mgf $M_i(s)$, and let $Y = \sum_{i=1}^N X_i$. Then the mgf of Y is given by*

$$M_Y(s) = \prod_{i=1}^N M_i(s).$$

Proof. By definition

$$M_Y(s) = \mathrm{E}[e^{sY}] = \mathrm{E}[e^{s\sum_{i=1}^N X_i}] = \mathrm{E}[e^{s(X_1+X_2+\ldots+X_N)}] = \mathrm{E}\left[\prod_{i=1}^N e^{sX_i}\right].$$

Since we specified that X_i are independent for $i \neq j$, then

$$\mathrm{E}\left[\prod_{i=1}^N e^{sX_i}\right] = \prod_{i=1}^N \mathrm{E}\left[e^{sX_i}\right] = \prod_{i=1}^N M_i(s),$$

which is what we had to show. □

0.2.8 Normal distribution

We've introduced two important distributions so far: binomial and uniform. There are many, many more to consider, but we'll focus on one that's extra important to us going forward: the Normal distribution (sometimes called the Gaussian). A random variable X is normally distributed with expected value μ and variance σ^2 if it has a pdf

$$f(X;\sigma^2,\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, x \in \mathbb{R}$$

We write this as $X \sim N(\mu, \sigma^2)$. If $\mu = 0$ and $\sigma^2 = 1$ we have what's called the "standard normal" or

$$\phi(X) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, x \in \mathbb{R}$$

which is so cool that it gets to be ϕ instead of f.

You've seen normal distributions all over the place, they're the most famous "bell curves"

```r
curve(dnorm(x, mean=0, sd=0.5), from=-5,to=6, ylab=expression(f(x)))
curve(dnorm(x, mean=0, sd=1), from=-5,to=6, add = TRUE, col="blue")
curve(dnorm(x, mean=2, sd=2), from=-5,to=6, add = TRUE, col="red")
legend(x="topright",
       legend =c("N(0, 0.25)", "N(0,1)", "N(2,4)"),
       col = c("black", "blue", "red"), lty=1)
```

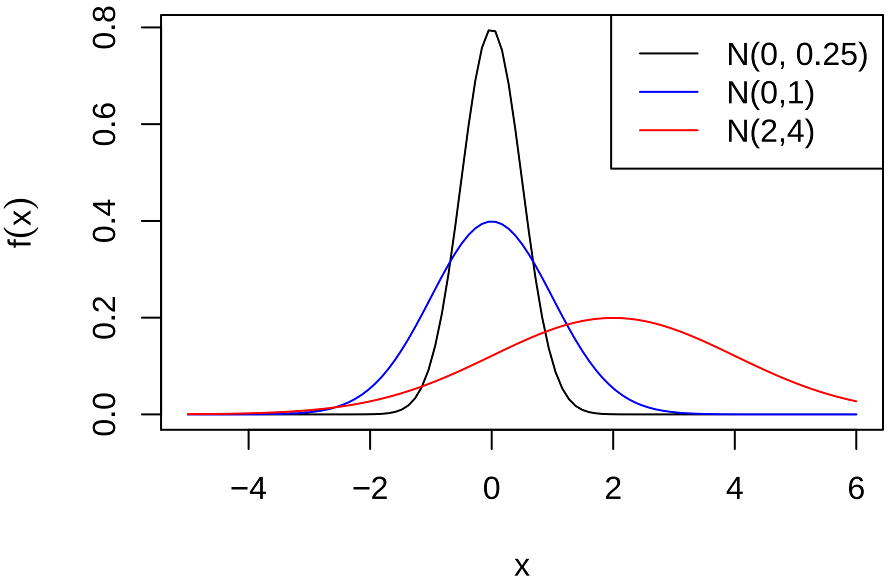


Figure 5: Different normals

As you can see, the expected value parameter μ centers the distribution and σ^2 (the variance) changes the spread. However, they're all bell and all symmetric.

Fun facts about the normal distribution:

1. If $X \sim N(\mu, \sigma^2)$, then $Z = \frac{X-\mu}{\sigma} \sim N(0,1)$. This is called a Z transformation. To see this consider

$$\begin{aligned}
\Pr(Z \leq z) &= \Pr\left(\frac{X-\mu}{\sigma} \leq z\right) \\
&= \Pr\left(X \leq z\sigma + \mu\right) \\
&= \int_{-\infty}^{z\sigma+\mu} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx \\
&= \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz \\
&= \Phi(z)
\end{aligned}$$

   Where the transition from line 3-4 follows from integration by substitution where the substitution function is $g(x) = \frac{x-\mu}{\sigma}$.

2. Immediately following from the above is the identity that if $X \sim N(\mu, \sigma^2)$ then X can be written as $X = \mu + \sigma Z$, where $Z \sim N(0,1)$. Any normal random variable is an **affine transformation** of a standard normal random variable (constant plus weighted random variable).

3. The expected value, median, and mode of X is μ

4. The moment generating function of a normal random variables is

$$M_X(s) = e^{s\mu + s^2\sigma^2/2}$$

5. For large N and mid-ranged p the binomial distribution is approximately normal with mean Np and variance $Np(1-p)$

```r
set.seed(1)
B <- rbinom(1000, size=100, prob=.7)
hist(B, freq=FALSE)
curve(dnorm(x, mean=100*.7, sd=sqrt(100*.7*.3)), col="red", add=T)
```

   a. In general the sum of N iid random variables will be normally distributed for large N (more on this Thursday)

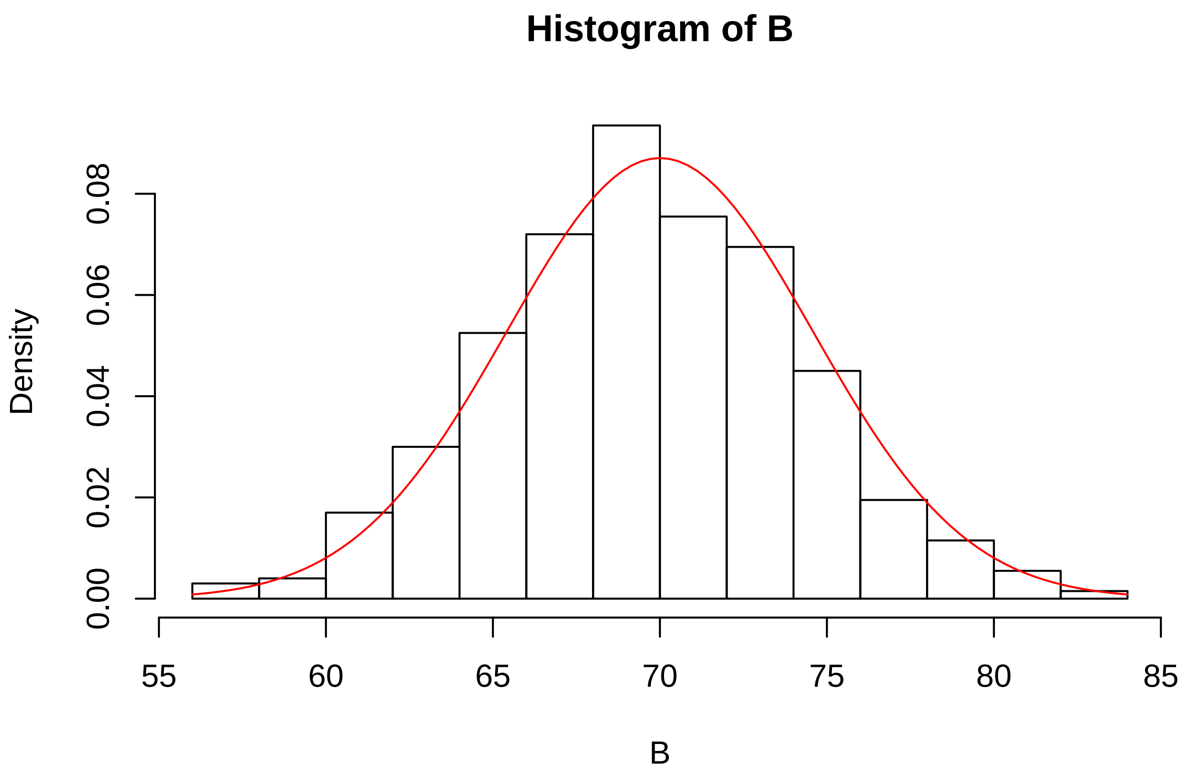


Figure 6: Normal approximation to the binomial

We can consider some useful functions for the normal

```
pnorm(0) #CDF of the standard normal
```

```
## [1] 0.5
```

```
pnorm(1.96, lower.tail=FALSE) # 1- CDF
```

```
## [1] 0.0249979
```

```
qnorm(0.025) #inverse CDF
```

```
## [1] -1.959964
```

```
dnorm(0) #pdf
```

```
## [1] 0.3989423
```

```
rnorm(4) #generate random normals
```

```
## [1] 0.3146242 0.8015725 0.4860589 1.7255611
```

```
#see help files for more
```

0.2.9 Functions of random variables

Let $X_1, \ldots, X_N$ be the outcomes of an experiment. We typically don't want to present or explain every individual data point, instead we are interested in using functions of the random variables to build an argument. Quantities like the average, minimum, maximum, variance, etc. As functions of random variables these quantities are themselves a random variable.

The general approach to describing the distribution of a random variable $Y = g(X)$ is to rewrite the CDF of Y in terms of the CDF of X. For example let $Y = a + bX$ where $b \neq 0$. To find the distribution of Y we first want to rewrite $F_Y(y)$ in terms of X and then rearrange (assume $b > 0$ for now)

$$F_Y(y) = \Pr(Y \leq y) = \Pr(a + bX \leq y) = \Pr(X \leq (y-a)/b) = F_X((y-a)/b).$$

To find the PDF

$$D_y F_X((y-a)/b) = \frac{1}{b} f_X((y-a)/b).$$

Note that when $b < 0$, $F_Y(y) = 1 - F_X((y-a)/b)$ with pdf $f_Y(y) = -\frac{1}{b} f_X((y-a)/b)$. Combining these we get

$$f_y(y) = \frac{1}{|b|} f_X((y-a)/b).$$

Despite being a very specific example, the above actually leads us into a more general formula for transformations of random variables. Specifically, let $Y = g(X)$ where g is invertable and differentiable and the PDF of Y is given by

$$f_Y(y) = \sum_{x:g(x)=y} f_X(g^{-1}(y))/|D_x g(x)|.$$

Let's try it out with (1) from above, where $X \sim N(0,1)$ and we want to find $Y = X^2$.

$$\begin{aligned}
g(x) &= x^2 \\
D_x g(x) &= 2x \\
g^{-1}(y) &= \pm\sqrt{y} \\
f_Y(y) &= \phi(-\sqrt{y})/|2x| + \phi(\sqrt{y})/|2x| \\
&= \left(\frac{1}{\sqrt{2\pi}} e^{-(-\sqrt{y})^2/2} + \frac{1}{\sqrt{2\pi}} e^{-(\sqrt{y})^2/2}\right) / (2\sqrt{y}) \\
&= \left(\frac{2}{\sqrt{2\pi}} e^{-y/2}\right) / \, (2\sqrt{y}) \\
&= \left(\frac{1}{\sqrt{2y\pi}} e^{-y/2}\right) \\
&= \left(\frac{y^{-1/2} e^{-y/2}}{2^{1/2}\Gamma(1/2)}\right),
\end{aligned}$$

which is the χ^2_1 pdf (fun fact $\Gamma(1/2) = \sqrt{\pi}$). The Γ function is one of those weird math things that occasionally pops up. Don't lose sleep over it.

We can use the same technology to find all sorts of relationships between distributions. If we had more time and a lot of patience we could show

1. If $X_i \sim N(\mu_i, \sigma_i^2)$ for $i = 1, \dots, N$, then $Y = \sum_{i=1}^N X_i \sim N\left(\sum_{i=1}^N \mu_i, \sum_{i=1}^N \sigma_i^2\right)$ (if X_i and X_j are independent for all $i \neq j$; if dependent, then apply the variance of a sum formula from above).
2. If $X_i \sim N(0,1)$ for $i = 1, \dots, N$, then $Y = \sum_{i=1}^N X_i^2 \sim \chi^2_N$ (if X_i and X_j are independent for all $i \neq j$)
3. If $X \sim N(0,1)$ and $Y \sim \chi^2_k$, then $t = X/\sqrt{Y/k} \sim t_k$ (helpful, but damn messy and beyond our scope) (if X and Y are independent)
4. If $X \sim t_k$, then $Y = X^2 \sim F(1,k)$
5. If $X \sim \chi^2_k$ and $Y \sim \chi^2_j$, then $F = \frac{X/k}{Y/j} \sim F(k,j)$ (if X and Y are independent)

These are helpful relationships we should remember. Set the t stuff aside for now, we'll get to that later in math camp, and we'll return to the F in 602. For the χ^2_k let's consider a few simulations

```r
X <- rnorm(1000)
Xsq <- X^2
par(mfrow=c(2,2))
hist(X, main="", xlab=expression(X%~%N(0,1)), freq=FALSE, breaks="Scott")
```

```r
curve(dnorm(x), add=T, col="red")
hist(X^2, main="", xlab=expression(X^2%~%{chi^2}[1]), freq=FALSE, breaks="Scott")
curve(dchisq(x,df=1), add=T, col="red")

# More
Xn <- replicate(10, rnorm(1000))
Xn2 <- rowSums(Xn^2)
hist(Xn2, main="",freq=FALSE, breaks="Scott",
     xlab=expression(sum({X[i]}^2, i==1, 10)%~%{chi^2}[10]))
curve(dchisq(x,df=10), add=T, col="red")

Xn <- replicate(25, rnorm(1000))
Xn2 <- rowSums(Xn^2)
hist(Xn2, main="", freq=FALSE, breaks="Scott",
     xlab=expression(sum({X[i]}^2, i==1, 25)%~%{chi^2}[25]))
curve(dchisq(x,df=25), add=T, col="red")
```

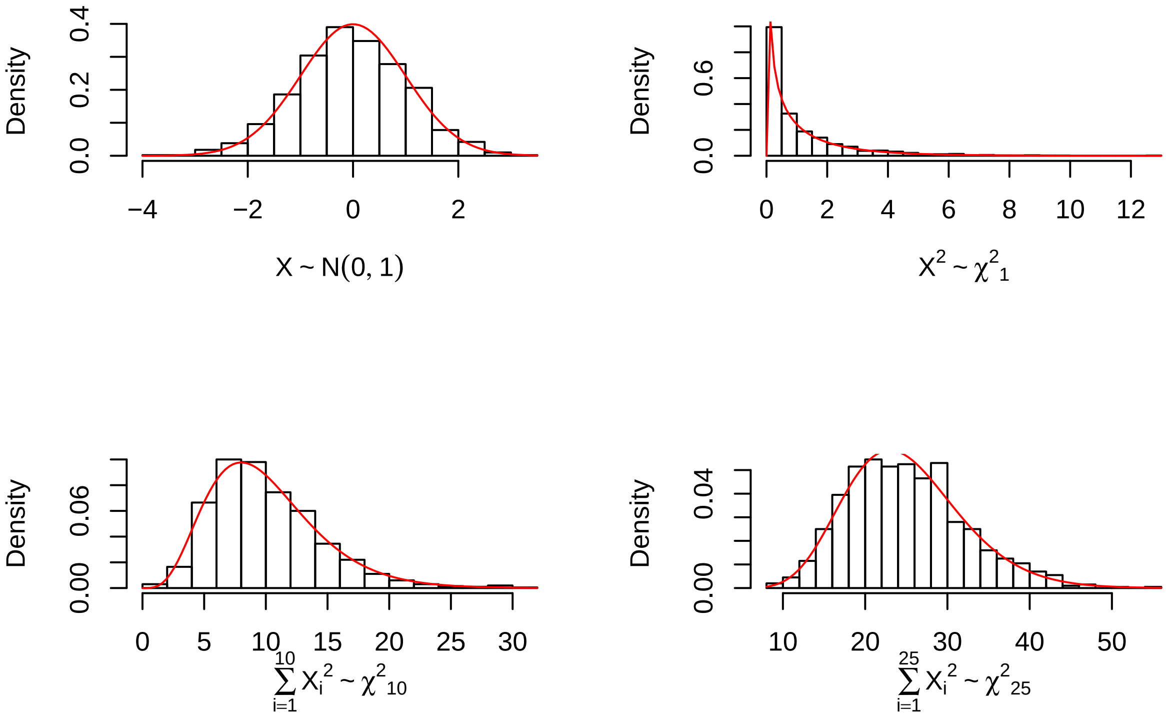


Figure 7: Relationship between the standard Normal and χ^2 distribution

Let’s try a couple practice problems: 1. Suppose we have independent random variables $X \sim N(0,1)$ and $Y \sim N(1,2)$, find the distributions of $X+Y$ and $X-Y$. Based on result

(1) above we know that $X + Y \sim N(1, 3)$ and $X - Y \sim N(-1, 3)$ 2. Suppose that X and Y (above) are dependent with $Cov(X, Y) = -1/4$, Find the distributions of $X + Y$ and $X - Y$. Based on result (1) *and* above we know that $X + Y \sim N(1, 3 + 2(-1/4)) = N(1, 2.5)$ and $X - Y \sim N(-1, 3 - 2(-1/4)) = N(1, 3.5)$

0.2.10 Approximations

One final note before we move on. There will be times in your life where you will be interested in finding $\mathrm{E}[g(X)]$ and $\mathrm{Var}(g(X))$, but direct calculation (as in the above) will be impractical. Let $\mu = \mathrm{E}[X]$, and we can use a **Taylor series approximation** to approximate $g(X)$.

A full Taylor series can approximate a function $g(x)$ that has k derivatives around a given point a as:

$$g(x) = g(a) + D_x g(a)(x - a) + \frac{(x-a)^2}{2!} D_x^2 g(a) + \ldots + \frac{(x-a)^k}{k!} D_x^k g(a) + o(|x - a|^k).$$

Here o is "little o" notation and refers to a function o that converges to 0 as x goes to a (you may never see that again, it's just a remainder term). This approximation gets better for values of x that are close to a. For our purposes we will often do an expansion around the expected value $\mathrm{E}[X] = \mu$ and really we usually only need $k = 1$ or 2 most of the time. The expected value makes for a good choice of a because it's the point that minimizes the average distance from X to a over all $x \in \mathrm{supp}(X)$.

Here's what we want to do with this:

$$\begin{aligned}
g(X) &= g(\mu) + D_X g(\mu)(X - \mu) + \text{Remainder} \\
&\approx g(\mu) + D_X g(\mu)(X - \mu) \\
\mathrm{E}[g(X)] &\approx g(\mu) + D_X g(\mu)\,\mathrm{E}[X - \mu] \\
&\approx g(\mu) \\
&\approx g(\mathrm{E}[X])
\end{aligned}$$

This is an approximation and it can be more or less wrong. If we know something about the shape of the function we can say a little about how wrong it is using **Jensen's inequality.** Jensen tells us

1. If g is convex ($D_x^2 g(a) > 0$) then $E[g(X)] \leq g(E[X])$. Some convex functions include exp and x^2

2. If g is concave ($D_x^2 g(a) < 0$) then $E[g(X)] \geq g(E[X])$. Some concave functions include log and $\sqrt{x}$

Say $g(x) = e^x$, then the first-order approximation will be? (Ans: weakly larger than the truth). If this bothers us, we can extend this into a second order expansion to get a better approximation and mitigates the Jensen problem:

$$\begin{aligned}
\mathrm{E}[g(X)] &\approx g(\mu) + D_X g(\mu)\,\mathrm{E}[X-\mu] + \frac{1}{2}D_X^2 g(\mu)\,\mathrm{E}[(X-\mu)^2] \\
&\approx g(\mu) + \frac{1}{2}D_X^2 g(\mu)\,\mathrm{Var}(X).
\end{aligned}$$

Applying just the first-order expansion with the variance we get

$$\mathrm{Var}(g(X)) \approx D_X g(\mu)^2\,\mathrm{Var}(X),$$

which is the basis of the **Delta Method** (we'll see this again again and state it formally tomorrow).

Practice: let X be a normal random variable with mean 1 and variance 1, find the first and second order approximation to $E[\exp(X)]$.

The first order is pretty straightforward: $E[g(X)] \approx g(\mathrm{E}[X])$ or $E[\exp(X)] = \exp(E[X]) = e^1 = e$. The second order is a little more involved

$$\begin{aligned}
\mathrm{E}[g(X)] &\approx g(\mathrm{E}[X]) + \frac{1}{2}D_x^2 g(\mathrm{E}[X])\,\mathrm{Var}(X) \\
&= e + \frac{1}{2}\exp(\mathrm{E}[X])\,\mathrm{Var}(X) \\
&= e + \frac{1}{2}e \\
&= \frac{3e}{2}
\end{aligned}$$

0.3 Joint and conditional distributions

Up until now we've only considered random variables as solo acts. However, it would be a very boring world if everything was independent of everything else. This will lead us into the idea of distributions that describe two or more random variables.

Let $X_1, X_2, \ldots, X_N$ be random variables that describe some experiment. When we had a single X we were interested in the CDF that represented $\Pr(X \leq x)$, now however we are interested in a **joint CDF** that we define as

$$F(x_1, x_2, \dots, x_N) = \Pr(X_1 \leq x_1 \cap X_2 \leq x_2 \cap \dots \cap X_N \leq x_N).$$

We're going to (mostly) focus on just situations of two random variables, but everything carries up.

0.3.1 Discrete joint distributions

Let X_1 and X_2 be discrete random variables. The function f is a **joint pmf** of X_1, X_2 if

$$f(x_1, x_2) = \Pr(X_1 = x_1, X_2 = x_2).$$

Note that for a well specified f we can find all possible probabilities for combinations of X_1, X_2. Specifically, we can also find the pmf of any *just* X_1 by summing over all values of X_2

$$f(x_1) = \Pr(X_1 = x_1) = \sum_{x \in \text{supp}(X_2)} \Pr(X_1 = x_1, X_2 = x)$$

The intuition can be seen in a two by two table. Imagine drawing a card from an ordinary deck and we care if its an odd, even, or face card and whether it's black or red.

X/Y	Black	Red	total
Face	6/52	6/52	12/52
Odd	10/52	10/52	20/52
Even	10/52	10/52	20/52
total	1/2	1/2	1

The individual cells of the table show the joint probabilities of these events. For example, $\Pr(X = \text{Face}, Y = \text{Red}) = 6/52$ (Jack, Queen, King of Hearts/Diamonds). The distribution of just X (the marginal) is the final column, while the marginal of Y is the final row. Note, that this generally isn't reversible, we can't get from the marginal to the joint.

The one cases where we *can* move from joints and marginals without trouble is when the random variables are independent. Random variables X_1, X_2 are independent only if

$$\Pr(X_1 = x_1, X_2 = x_2) = \Pr(X_1 = x_1) \Pr(X_2 = x_2)$$

for all possible x_1, x_2.

Is the card example above independent? Yes, notice that each joint can be written as the product of the marginals. We could make it not independent by, say, remove all of the red face cards then

X/Y	Black	Red	total
Face	6/46	0	6/46
Not face	20/46	20/46	40/46
total	26/46	20/46	1

Now we can see that joints are no longer equal to the product of the marginals, which implies that events are not independent. A simple extension of the above result tells us that a sequence $X_1, \ldots, X_N$ is independent if and only if

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{N} f_i(x_i)$$

0.3.2 Continuous joint distributions

As before the results for continuous distributions will look similar to discrete ones. Now we are interested in the joint pdf of continuous random variables $X_1, \ldots, X_N$

$$\Pr(a_1 \leq X_1 \leq b_1, \ldots, a_N \leq X_N \leq b_N) = \int_{a_1}^{b_1} \ldots \int_{a_N}^{b_N} f(x_1, \ldots, x_n) dx_1 \ldots dx_N.$$

That... looks like a nightmare, but don't worry it looks a lot worse than it'll ever be in practice. A two dimensional case might be easier

$$\Pr(a_1 \leq X_1 \leq b_1, a_2 \leq X_2 \leq b_2) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x_1, x_2) dx_1 dx_2.$$

As before we can back out the marginal by integrating over all values of one variable such that

$$f(x_1) = \int_{-\infty}^{\infty} f(x_1, x_2) dx_2.$$

Also like before we have an independence result where $X_1, \ldots, X_N$ are independent continuous random variables if and only if

$$f(x_1, \ldots, x_N) = \prod_{i=1}^{N} f_i(x_i)$$

for all $\{x_1, \ldots, x_N\} \in \mathbb{R}^N$.

0.3.3 Expectations and variance

The means, variances, and higher moments of any specific variable in a joint distribution are defined with respect to their marginal distribution as in:

$$\begin{aligned}
\mathrm{E}[X_1] &= \int_{-\infty}^{\infty} x_1 f(x_1) dx_1 \\
&= \int_{-\infty}^{\infty} x_1 \int_{-\infty}^{\infty} f(x_1, x_2) dx_2 dx_1 \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 f(x_1, x_2) dx_2 dx_1 \\
\mathrm{Var}(X_1) &= \int_{-\infty}^{\infty} (x_1 - \mathrm{E}[X_1])^2 f(x_1) dx_1 \\
&= \int_{-\infty}^{\infty} (x_1 - \mathrm{E}[X_1])^2 \int_{-\infty}^{\infty} f(x_1, x_2) dx_2 dx_1 \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_1 - \mathrm{E}[X_1])^2 f(x_1, x_2) dx_2 dx_1
\end{aligned}$$

Some properties that will be useful. As we saw before the expectation of a sum of random variables will be the sum of the expectations. Specifically let $Y = a + b_1 X_1 + b_2 X_2 + \ldots + b_N X_N$, and $a, b_1, \ldots, b_N$ are constants. Then

$$\mathrm{E}[Y] = a + b_1 \,\mathrm{E}[X_1] + \ldots + b_N \,\mathrm{E}[X_N].$$

This follows because constants can move out of sums/integrals and sums/integrals can be split along addition/subtraction.

For example suppose that $X_1, \ldots, X_N$ are identically distributed Bernoulli with parameter p. The Bernoulli distribution describes a random variable that only takes on values 1 or 0 with probability p and $1 - p$, respectively. It is a special case of the binomial distribution where $N = 1$. The expected value of a Bernoulli is

$$\mathrm{E}[X_i] = 1p + 0(1 - p) = p$$

and so the expected value of $B = \sum_{i=1}^{N} X_i$ is

$$\mathrm{E}[B] = \mathrm{E}\left[\sum_{i=1}^{N} X_i\right] = \sum_{i=1}^{N} \mathrm{E}\left[X_i\right] = \sum_{i=1}^{N} p = Np.$$

Another useful property: Suppose that $X_1, \ldots, X_N$ are independent, then

$$\mathrm{E}[X_1 X_2 \ldots X_N] = \mathrm{E}[X_1]\,\mathrm{E}[X_2] \ldots \mathrm{E}[X_N].$$

We will consider a two dimensional proof: let $f(x_1, x_2)$ be the joint pdf of X_1, X_2 with marginals $f(x_1)$ and $f(x_2)$, then

$$\begin{aligned}
\mathrm{E}[X_1 X_2] &= \int \int x_1 x_2 f(x_1, x_2) dx_1 dx_2 \\
&= \int \int x_1 x_2 f(x_1) f(x_2) dx_1 dx_2 \text{ by independence} \\
&= \int x_1 f(x_1) dx_1 \int x_2 f(x_2) dx_2 \\
&= \mathrm{E}[X_1]\,\mathrm{E}[X_2].
\end{aligned}$$

Generalizing this to discrete or to more than two variables is straightforward.

0.3.4 Covariance

Something we haven't seen yet is the idea of **covariance**. Covariance describes how **linearly** dependent two variables are on each other. The covariance of random variable X_1 and X_2 is given by

$$\mathrm{Cov}(X_1, X_2) = \mathrm{E}[(X_1 - \mathrm{E}[X_1])(X_2 - \mathrm{E}[X_2])].$$

Notice that if $X_1 = X_2$ this simplifies in the variance. Covariance is interpretable at the sign level. Positive covariance means that large (small) values of X_1 are associated with large (small) values of X_2. Negative covariance means that small (large) values of X_1 are associated with large (small) values of X_2.

Some important things to remember about variances and covariances for random variables X, Y, Z, W and constants a, b, c, d:

1. $\mathrm{Var}(X) = \mathrm{E}[X^2] - \mathrm{E}[X]^2$
2. $\mathrm{Var}(aX + b) = a^2\,\mathrm{Var}(X)$
3. $\mathrm{Cov}(X, Y) = \mathrm{E}[XY] - \mathrm{E}[X]\,\mathrm{E}[Y]$
4. $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$
5. $\mathrm{Cov}(X, a) = 0$
6. $\mathrm{Cov}(aX + bY, Z) = a\,\mathrm{Cov}(X, Z) + b\,\mathrm{Cov}(Y, Z)$
7. $\mathrm{Cov}(aX + bY, cW + dZ) = ac\,\mathrm{Cov}(X, W) + ad\,\mathrm{Cov}(X, Z) + bc\,\mathrm{Cov}(Y, W) + bd\,\mathrm{Cov}(Y, Z)$
8. $\mathrm{Cov}(X, X) = \mathrm{Var}(X)$

9. $\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X,Y)$
10. $\mathrm{Var}(X-Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) - 2\,\mathrm{Cov}(X,Y)$
11. $\mathrm{Var}\left(\sum_{i=1}^N b_i X_i\right) = b_i^2 \sum_{i=1}^N X_i$, for independent X_i's

These are straight forward to prove to yourself using the definitions of covariance and independence.

Covariance is cool, but we can often get a little more information from the correlation of X_1, X_2:

$$\rho(X_1, X_2) = \frac{\mathrm{Cov}(X_1, X_2)}{sd(X_1)sd(X_2)}.$$

The correlation will always be in the interval -1 to 1. Values closer to ±1 describe a more linear relationship.

0.3.5 Conditional distributions

Suppose that X and Y are random variables with pmf/pdf $f(x,y)$ where the marginal $f(x) > 0$. Then the conditional pdf/pmf of Y given that $X = x$ is given as the conditional pmf/pdf

$$f(y|X = x) = \frac{f(x,y)}{f(x)}.$$

This function tell us that the pdf/pmf for a fixed value of X. There's a Bayes' rule for conditional distributions too,

$$f(y|X = x) = \frac{f(x|Y = y)f(y)}{f(x)}.$$

This extension allows us to define things like the conditional expectation of Y given that we've observed $X = x$:

$$\mathrm{E}[Y|X = x] = \int y f(y|X = x)dy.$$

Or the variance,

$$\mathrm{Var}[Y|X = x] = \mathrm{E}\left[Y^2|X = x\right] - \mathrm{E}[Y|X = x]^2 = \mathrm{E}\left[(Y - E[Y|X = x])^2|X = x\right].$$

Overall, conditional distributions behave the same as other distributions. The fixed values can be treated as constants and we deal with the variable(s) left over.

An important result going forward is the law of iterated expectations:

$$E[Y] = \mathrm{E}_X[\mathrm{E}[Y|X=x]],$$

which is to say that we can find the *unconditional* expected value of Y by averaging over all the conditional expectations. To see this in the discrete case

$$\begin{aligned}
\mathrm{E}_X[\mathrm{E}[Y|X=x]] &= \sum_x \mathrm{E}[Y|X=x]f(x) \\
&= \sum_x \left(\sum_y yf(y|X=x)\right) f(x) \\
&= \sum_x \sum_y y\left(f(x|Y=y)\right)f(y) \quad \text{Bayes' rule} \\
&= \sum_y yf(y)\left(\sum_x f(x|Y=y)\right) \\
&= \sum_y yf(y)1 \\
&= E[Y]
\end{aligned}$$

Conditional distributions appear everywhere so keep your eyes peeled going forward.

0.4 Estimation and hypothesis testing

Let's suppose that we had some guess as to what distribution $f(x;\theta)$ is. For now let's suppose it's a normal distribution where $\theta = (\mu, \sigma^2)$ is a set of true, but unknown, population parameters. Most of our goals in data analysis is to find good ways to guess θ from x. A parametric approach is when we assume we know what f is. Two ways we might think about estimating θ are to find **point estimates** of σ and μ. A point estimate, denoted $\hat{\theta}$ is a single best guess of the parameter(s). Point estimates typically come with a **standard error** which is the standard deviation of the sampling distribution of the estimate(s) $\hat{\theta}$. Alternatively we might build an **interval estimate** which is a range of values that will contain the true value θ with a pre-determined probability.

0.4.1 The sample mean and general properties of estimators

An **estimator** is a function that takes the data and produces an estimate. To estimate the true but unknown μ we might come up with a few different approaches:

1. $\hat{\mu} = \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
2. $\hat{\mu}_1 = x_1$

Intuitively we suspect that (1) is a better estimator, but how would we know? Some basic properties we can consider are the **finite sample** properties. These properties hold regardless of sample size. Later we will consider **asymptotic properties** (large sample properties).

Within a finite sample we tend to consider an estimator's **bias** and **efficiency** An estimator of θ is unbiased if the mean of its sampling distribution is θ, which is to say

$$\mathrm{E}[\hat{\theta}] = \theta$$

or

$$\mathrm{E}[\hat{\theta} - \theta] = 0.$$

Unbiasedness is nice, but more than one estimator can be unbiased. Estimators (1) and (2) are both unbiased, however, so this doesn't get us far in this case. As such, we turn to efficiency. Estimator $\hat{\theta}_1$ is more efficient that $\hat{\theta}_2$ if $\mathrm{Var}(\hat{\theta}_1) < \mathrm{Var}(\hat{\theta}_2)$. Efficiency gives us some bite on deciding between (1) and (2). Specifically, we showed above that variance of (1) is σ^2/N, while the same logic tells us that the variance of (2) is σ^2. So long as $N > 1$ (1) is more efficient.

What does efficiency actually mean? It means that on average estimates produced by a specific estimator will be closer to that estimators expected value than. For unbiased estimators it means that, while both (1) and (2) return the truth in expectation, (1) will produce estimates that are closer to it. Let's check it out

```r
set.seed(1)
x <- replicate(1000, rnorm(5, mean=3, sd=1))
mu1 <- colMeans(x)
mu2 <- x[1,]
tblue <- rgb(red = 0, green = 0, blue = 1, alpha = 0.3)
tred<- rgb(red = 1, green = 0, blue = 0, alpha = 0.3)
hist(mu2, freq=F, col=tred, ylim=c(0,1), main="", xlab="estimates")
hist(mu1, freq=F, add=T, col=tblue)
legend("topright", legend=c(expression(hat(mu)[1]), expression(hat(mu)[2])),
       col = c(tblue, tred), lty=1,lwd=2)
```

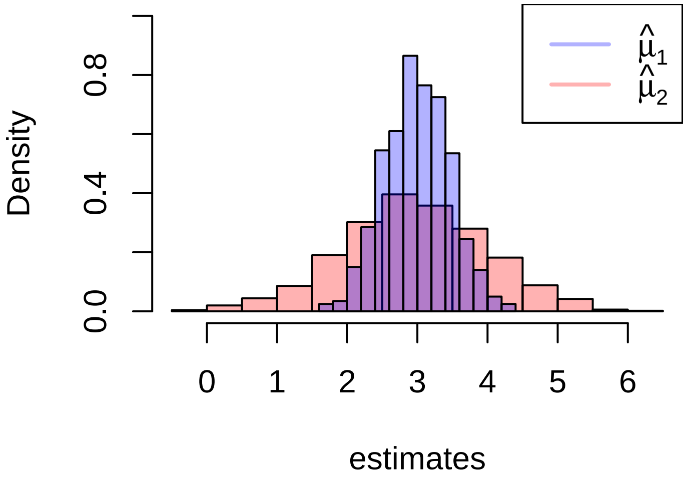


Figure 8: Sampling distributions for $\hat{\mu}_1$ and $\hat{\mu}_2$

```
c(mean(mu1), mean(mu2))
```

```
## [1] 2.996812 2.961216
```

```
c(var(mu1), var(mu2))
```

```
## [1] 0.2145218 1.0622903
```

Both are centered at the truth, but even with $N = 5$ we see a notably tighter sampling distribution.

There are two main schools of though on how to compare estimators. One is to find the minimum variance unbiased estimator. The other is to find that the estimator with the lowest **mean squared error (MSE)**. The MSE of an estimator is defined as

$$MSE(\hat{\theta}) = \text{Var}(\hat{\theta}) + \text{Bias}(\hat{\theta})^2,$$

although sometimes we use the root-mean-squared error (RMSE) which is the square root of the above. The former is more typical in classic treatments informs a lot of practices today, but the latter is more trendy at the moment. One example that may be of note is using the sample variance to estimate σ^2. Traditionally we write the sample variance as

$$s_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2.$$

Note that $N - 1$ is used over the more intuitive N in order to make s_x^2 an unbiased estimate of σ^2. Let's check it out

The expected value and variance of s_x^2 are given as:

$$\mathrm{E}[s_x^2] = \sigma^2$$
$$\mathrm{Var}[s_x^2] = \frac{2\sigma^4}{N-1}$$

Now let $\hat{\sigma}^2$ be the "intuitive" estimator given by $\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$. The expected value and variance of this estimator are given as:

$$\mathrm{E}[\hat{\sigma}^2] = \frac{N-1}{N}\sigma^2$$
$$\mathrm{Var}[\hat{\sigma}^2] = \left(\frac{N-1}{N}\right)^2 \frac{2\sigma^4}{N-1}$$

So on average the estimates will be attenuated, but they'll have a lower variance:

```r
library(matrixStats)
s2 <- colVars(x)
sigma2 <- colSums(t((t(x)-colMeans(x))^2))/5
hist(s2, freq=F, col=tred, ylim=c(0,.8), main="", xlab="estimates")
hist(sigma2, freq=F, add=T, col=tblue)
legend("topright", legend=c(expression(s[x]^2), expression(hat(sigma)^2)),
       col = c(tred, tblue), lty=1,lwd=2)
```

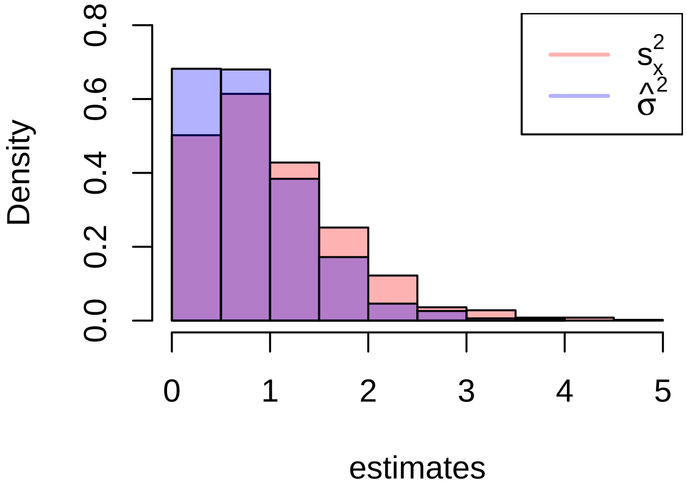


Figure 9: Sampling distributions for s^2 and $\hat{\sigma}^2$

```r
c(mean(s2), mean(sigma2))
```

```
## [1] 1.0494768 0.8395814
```

```r
c(var(s2), var(sigma2))
```

```
## [1] 0.5238803 0.3352834
```

```r
c((mean(s2)-1)^2 + var(s2),(mean(sigma2)-1)^2 + var(sigma2))
```

```
## [1] 0.5263282 0.3610175
```

Here you can see that s_x^2 has an average value of about 1 (and more of a peak there), but has a higher variance (see the longer tail). In contrast $\hat{\sigma}^2$ has an average less than 1 (about $\frac{N-1}{N} = 0.8$, as expected), but less variance and a lower overall MSE. As sample size increases the differences diminish to the point where it doesn't really matter.

0.4.2 Sampling distribution and hypothesis testing

If we continue to assume that the population is normally distributed we can say a few more things about the sample mean and variance in finite samples. First, we can say that $(\bar{x} - \mu) \sim N(0, \sigma^2/N)$. To see this notice that $\bar{x}$ is the sum of iid normal random variables and as we said before the sum of normal random variables is normal. We've characterized the distribution of $\bar{x} - \mu$ completely, and if we know σ this becomes usable to answer interesting questions.

Second, we can show that the scaled random variable $\frac{N-1}{\sigma^2} s_x^2 \sim \chi^2_{N-1}$. Roughly,

$$\begin{aligned} s_x^2 &= \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2 \\ &= \frac{\sigma^2}{N-1} \sum_{i=1}^{N} \left(\frac{x_i - \bar{x}}{\sigma} \right)^2 \end{aligned}$$

The inside portion is roughly the sum of squared standard normals, which as we noted above is χ^2 (NOTE: THIS IS NOT A REAL PROOF. It is in fact **wrong** because $\bar{x} \neq \mu$. The real proof relies on Cochran's theorem, but this gives you a basic idea for where this comes from.)

Now we go back to the distribution of $\bar{x}-\mu$ to see if we can morph it into something usable.

$$\begin{aligned}
\bar{x}-\mu &\sim N(0,\sigma^2/N) \\
\sqrt{N}(\bar{x}-\mu)/\sigma &\sim N(0,1) \\
\frac{N-1}{\sigma^2}s^2 &\sim \chi^2_{N-1} \\
\frac{\sqrt{N}(\bar{x}-\mu)/\sigma}{\sqrt{\frac{N-1}{\sigma^2}s_x^2/N-1}} &\sim ?? && \text{See transformation notes} \\
\frac{\sqrt{N}(\bar{x}-\mu)}{s_x} &\sim t_{N-1} && \text{simplified}
\end{aligned}$$

We can explore this visually, too.

```r
set.seed(1)
library(matrixStats)
N <- 10
mu <- 45
sigma <- 5
x <- replicate(1000,rnorm(N, mu, sigma))
s <- colSds(x)
xbar <- colMeans(x)
par(mfrow=c(3,2))
hist(xbar, freq=F, breaks="Scott", xlab=expression(bar(x)),
     main=expression(paste("Sampling distribution of ", bar(x))))
curve(dnorm(x, mean=mu, sd=sigma/sqrt(N)), add=T, col="red")
legend(x="topright", legend=c(expression(N(mu, sigma^2/N))),
       col=c("red"), lty=1,lwd=2)
hist(xbar-mu,  freq=F, breaks="Scott",xlab=expression(bar(x)-mu),
     main=expression(paste("Sampling distribution of ", bar(x)-mu)))
curve(dnorm(x, mean=0, sd=sigma/sqrt(N)), add=T, col="red")
legend(x="topright", legend=c(expression(N(0, sigma^2/N))),
       col=c("red"), lty=1,lwd=2)

hist((N-1)/sigma^2 * s^2, freq=F, breaks="Scott",
     xlab=expression(frac(N-1,sigma^2)*s^2),
     main=expression(paste("Sampling distribution of ", frac(N-1,sigma^2)*s^2)))
curve(dchisq(x, df=N-1), add=T, col="darkgreen")
```

```
legend(x="topright", legend=c(expression({chi^2}[N-1])),
       col=c("darkgreen"), lty=1,lwd=2)

hist((xbar-mu)/(sigma/sqrt(N)), freq=F, breaks="Scott",
     main=expression(paste("Sampling distribution of ",
                           frac(bar(x)-mu, sigma/sqrt(N)))),
     xlab=expression(sqrt(N)(bar(x)-mu)/sigma), ylim=c(0,.4))
curve(dnorm(x, mean=0, sd=1), add=T, col="red")
curve(dt(x, df=N-1), add=T, col="blue")
legend(x="topright", legend=c(expression(N(0, 1)), expression(t[N-1])),
       col=c("red", "blue"), lty=1,lwd=2)
hist((xbar-mu)/(s/sqrt(N)),  freq=F, breaks="Scott",
     main=expression(paste("Sampling distribution of ",
                           frac(bar(x)-mu, s/sqrt(N)))),
     xlab=expression(sqrt(N)(bar(x)-mu)/s), ylim=c(0,.4))
curve(dnorm(x, mean=0, sd=1), add=T, col="red")
curve(dt(x, df=N-1), add=T, col="blue")
legend(x="topright", legend=c(expression(N(0, 1)), expression(t[N-1])),
       col=c("red", "blue"), lty=1,lwd=2)

hist((xbar-mu)/(s/sqrt(N)),  freq=F, breaks="Scott",  ylim=c(0,.02), xlim=c(2,4),
     main=expression(paste("Sampling distribution of ",
                           frac(bar(x)-mu, s/sqrt(N)))),
     xlab=expression(sqrt(N)(bar(x)-mu)/s))
curve(dnorm(x, mean=0, sd=1), from=1.7, add=T, col="red")
curve(dt(x, df=N-1), from=1.7, add=T, col="blue")
legend(x="topright", legend=c(expression(N(0, 1)), expression(t[N-1])),
       col=c("red", "blue"), lty=1,lwd=2)
```

Now we can explore the world of hypothesis testing. Hypothesis testing is a framework for answering yes/no questions that take the form is a a plausible value for the true but unknown population parameter θ? Tests take the form a **null hypothesis** and an **alternative hypothesis** such that

$$H_0 : \theta = a$$

$$H_A : \theta \neq a$$

where the alternative can be $\neq$ (two tailed test) or $<$ / $>$ (one tailed). The **Neyman-**

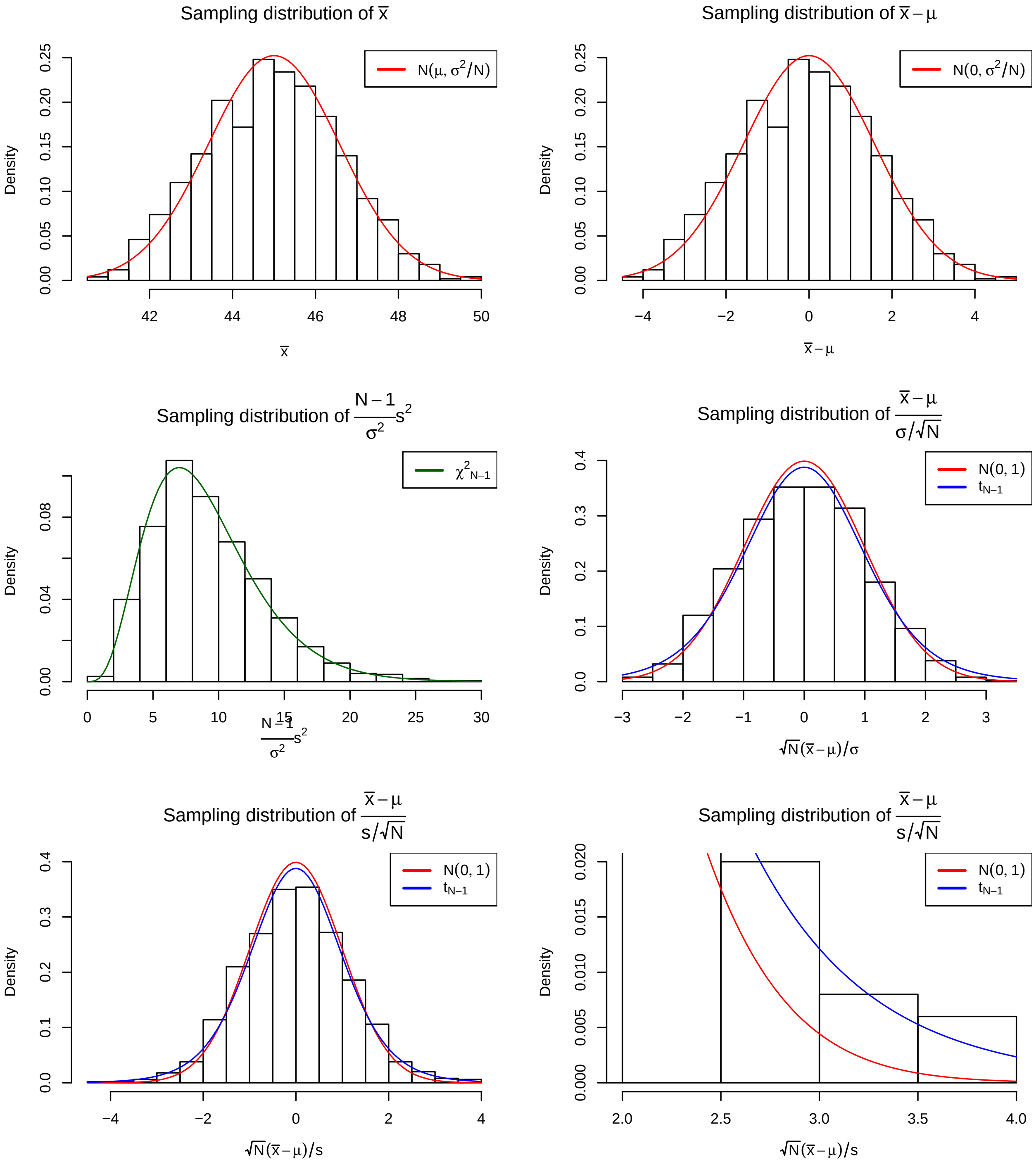


Figure 10: From the Normal to the t

Pearson approach to hypothesis testing requires the analyst to form a test statistic and compare the value of that statistic to a partition of statistic's the sampling distribution (a **rejection region** or an **acceptance region**). If the statistic falls in the rejection region, the null hypothesis is rejected. Otherwise, the null is not rejected (despite the name, don't conclude that the null is accepted–it's just not rejected). This obviously leads to four possible outcomes

Null is...	True	False
Not rejected	Good	Type II error
Rejected	Type I Error	Good

For example, we might produce an estimate that is very far from the true parameter value. This could lead us to make a Type 1 error. We obviously never know where we land on the table, but we can quantify. The **size** of the test refers to the probability of type 1 error and it is reflects by where we set the rejection region. A small rejection region (only reject when we have *very strong* evidence against the null) will reduce the probability of a type 1 error. It will also increase the probability of type II error. The size is often denoted by $\alpha \in (0, 1)$ where $\alpha \in \{0.05, 0.10\}$ are conventional choices. The **power** of a test is how likely are we to reject a false null hypothesis. This is $1 - \Pr(\text{Type II error}) = 1 - \beta$. Once we fix a significance level α we want to have a test that minimizes β (maximizes power). We will set aside more thorough discussions of hypothesis testing, but you may come back to ideas like power in courses on behavior or the causal inference class. Power is most frequently discussed in experimental work where you have more control over sample sizes.

What we will say is that when you build a test statistic you do the following:

1. Set α
2. Assume the null hypothesis is true
3. Build a test statistic based on a true null and the sampling distribution of your estimate
4. Find the probability of observing a test statistic **as or more extreme** than what you observe given a true null. This is called the p value. Note the p value of a one sided test is half that of a two sided test. (we'll draw some pictures)
5. If $p < \alpha$ your test statistic is in the rejection region. Otherwise it is not.

Effectively what this says is: Assume a true null hypothesis, is there enough evidence to make that assumption highly unlikely? If so reject the null.

Moving on. Let's recap: What do we have?

1. A null hypothesis that $\mu = \mu_0$
2. A statistic $\bar{x}$

3. A sampling distribution $\bar{x} \sim N(\mu, \sigma^2/N)$
4. A test statistic $z = \frac{\sqrt{N}(\bar{x}-\mu_0)}{\sigma}$.
5. A statistic s^2 to use in place of σ^2
6. A sampling distribution of (scaled) s_x^2: $\frac{N-1}{\sigma^2}s^2 \sim \chi^2_{N-1}$
7. A new test statistic based on s_x^2: $t = \frac{\sqrt{N}(\bar{x}-\mu_0)}{s}$.
8. A distribution for the test statistic $t \sim t_{N-1}$.

Now we're cooking. Let's try an example by looking at some traffic data

	type	date	6th	13th	diff	location
1	traffic	1990, July	139246	138548	698	loc 1
2	traffic	1990, July	134012	132908	1104	loc 2
3	traffic	1991, September	137055	136018	1037	loc 1
4	traffic	1991, September	133732	131843	1889	loc 2
5	traffic	1991, December	123552	121641	1911	loc 1
6	traffic	1991, December	121139	118723	2416	loc 2
7	traffic	1992, March	128293	125532	2761	loc 1
8	traffic	1992, March	124631	120249	4382	loc 2
9	traffic	1992, November	124609	122770	1839	loc 1
10	traffic	1992, November	117584	117263	321	loc 2

Our research question is whether the amount of traffic on Fridays the 13th is different than on another ordinary Friday. We do this by comparing traffic at 10 specific intersections on Fridays the 13 and Fridays the 6th.

Our hypotheses become

$$H_0 : \mu_{\text{diff}} = 0$$
$$H_A : \mu_{\text{diff}} \neq 0$$

We find $\bar{x}$ and s. Then we build our test statistic.

```r
diffs <- c(698, 1104, 1037, 1889, 1911, 2416, 2761,4382,1839,321)
N <- length(diffs)
x.bar <- mean(diffs)
s <- sd(diffs)
t <- (sqrt(N)*x.bar)/s
pt(t, df=N-1, lower.tail = FALSE)*2 #explain this
```

```
## [1] 0.0008061844
```

Here we find a p value less than 0.05 and conclude that we have enough evidence to reject the null hypothesis. The evidence suggests that the number of accidents on Friday the 13th is different from other Fridays. The sample mean of 1835.8 suggests that the number of accidents is greater on the 13th.

0.4.3 Interval estimation

Interval estimation is different from point estimation in the sense that it provides an entire range of plausible values of θ. These ranges are known as confidence intervals. However, we'll see in a second that there is a deep connection between confidence intervals and hypothesis testing with point estimates. An interval estimate is often formed using a pivotal point (point estimate) $\hat{\theta}$ plus/minus uncertainty

To build a parametric interval estimate you need three things:

1. A point estimate
2. The sampling distribution
3. The standard error of the point estimate
4. A level of "confidence" $1 - \alpha$.

For the sample mean we have an estimate $\bar{x}$ for a pivotal point and we know that the transformation $\sqrt{N}(\bar{x} - \mu)/s$ is distributed t_{N-1}. To form confidence interval then we need to construct an interval based on the idea that

$$\Pr(a < g(\hat{\theta}; \theta) < b) = 1 - \alpha.$$

Where g is the transformation that allows us to have a known distribution. For the sample mean this gives us

$$\begin{aligned} \Pr(a < \sqrt{N}(\bar{x} - \mu)/s < b) &= 1 - \alpha \\ \Pr(-\bar{x} + a\frac{s}{\sqrt{N}} < -\mu < -\bar{x} + b\frac{s}{\sqrt{N}}) &= 1 - \alpha \\ \Pr(\bar{x} - b\frac{s}{\sqrt{N}} < \mu < \bar{x} - a\frac{s}{\sqrt{N}}) &= 1 - \alpha \end{aligned}$$

Choosing the right values of a and b give you $1 - \alpha$ confidence. Note that confidence here means something very specific. It refers to your confidence *in the procedure*. For 95% confidence that means that we are confident that 95% of 95% confidence intervals will contain the true θ. Any given confidence interval either has θ it is **not** so there is not a 95% chance that your estimated interval contains θ *ex post* (it does mean this *a priori*).

Now in order to find a and b we need to fix $1-\alpha$ and use the parametric distribution in question. In this case let's set $1-\alpha = 0.95$ to get a 95% confidence interval. This means that we need to cover 95% of the sampling distribution with the interval $[a, b]$. To put this another we want to partition the sampling distribution into three parts $(-\infty, a)$, $[a, b]$, and (b, ∞) such that they cover 2.5%, 95%, 2.5% of the density, respectively. With symmetric distributions like the t and normal, this means that $-a = b$ and we can find a for a normal using

```
qnorm(.025)
```

```
## [1] -1.959964
```

About -1.96 for a and 1.96 for b. So for a normal 95% confidence interval you get

$$\bar{x} \pm 1.96 \frac{s}{\sqrt{N}}.$$

For a t you need to know the degrees of freedom. Let's try one with at 90% CI.

```
set.seed(1)
x <- rnorm(10, mean=pi, sd=2)
x.bar <- mean(x)
se <- sd(x)/sqrt(10)
t.crit <- qt(.05, df=9)
c(x.bar+t.crit*se, x.bar-t.crit*se)
```

```
## [1] 2.501016 4.310980
```

```
good <- replicate(10000, {
  x <- rnorm(10, mean=pi, sd=3)
  x.bar <- mean(x)
  se <- sd(x)/sqrt(10)
  t.crit <- qt(.05, df=24)
  (pi > (x.bar+t.crit*se)) & (pi< (x.bar-t.crit*se))
})
mean(good)
```

```
## [1] 0.8817
```

About 10% of the intervals do not contain the true value.

There is an intimate connection between hypothesis testing and interval estimation. Let's return to the traffic data:

We can get to the same place as before using interval estimation

```
diffs <- c(698, 1104, 1037, 1889, 1911, 2416, 2761,4382,1839,321)
N <- length(diffs)
x.bar <- mean(diffs)
se <- sd(diffs)/sqrt(N)
t.crit <- qt(0.025, df=N-1, lower=FALSE)
c(x.bar - t.crit*se, x.bar+t.crit*se)
```

```
## [1]  994.5304 2677.0696
```

Now we conclude that the range of plausible differences between Fridays the 13th and Fridays the 6th is between 995 and 2677 accidents. Notice that this range is all positive so we can say that 0 (no differences) is not a reasonable guess of μ. That these approaches agree is not an accident. A $1-\alpha$ confidence interval is equivalent to a test where the probability of a type 1 error is α.

0.4.4 The method of moments

The older framework for finding parameter estimates from parametric models. Method of moments works by setting empirical moments equal to the theoretical moments of the parametric distribution. As such we need to return to our old friend the moment generating function. The number of theoretical moments we need depends on the number of parameters we want to estimate. For example consider some data $X_1, \ldots, X_N$ that we believe to have come from a uniform distribution. Further suppose that we know that the lower bound of this distribution is 0 but we don't know the upper limit. So we have that $x_1, \ldots, x_N$ are iid $U(0, \theta)$ where we want to estimate θ.

We have 1 parameter so we need one moment. In this case we take the empirical moment $\bar{x}$ and relate it to the theoretical moment given by $E[X]$. We don't really need the mgf for this one as $\mathrm{E}[X] = \frac{1}{2}(\theta + 0)$. As such we set these equal:

$$\begin{aligned}\bar{x} &= \frac{1}{2}(\theta)\\ 2\bar{x} &= \hat{\theta}_{\mathrm{MoM}}.\end{aligned}$$

So the moment estimator is twice the sample mean. Not an unreasonable approach, but

```r
set.seed(4)
X <- runif(5, min=0, max=100)
c(2*mean(X), max(X))
```

```
## [1] 79.17740 81.35742
```

It is possible to get an “impossible” estimate. Here we find that our best guess of θ would make at least one of our observations impossible to observe. This is a rare, but possible occurrence with some moment estimators. Moment estimators may be biased (this one isn’t).

Let’s try another example: Estimate the parameters μ and σ^2 from a normal distribution. Now we have two moments to solve for we will start with μ

$$\begin{aligned} \bar{x} &= \mu \\ \bar{x} &= \hat{\mu}_{\text{MoM}}. \end{aligned}$$

awesome.

Now onto σ^2. For this we need the second empirical moment $\frac{1}{N}\sum_{i=1}^{N} x_i^2$ and the second theoretical moment given by $\mathrm{E}[X^2]$. Using the normal mgf, we find that:

$$\mathrm{E}[X^2] = \mu^2 + \sigma^2$$

which is our second moment. Now we solve

$$\begin{aligned} \mu^2 + \sigma^2 &= \frac{1}{N}\sum_{i=1}^{N} x_i^2 \\ \sigma^2 &= \frac{1}{N}\sum_{i=1}^{N} x_i^2 + \mu^2 \\ \hat{\sigma}^2_{\text{MoM}} &= \frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})^2 \end{aligned}$$

which (as mentioned above) is a biased estimate (but better MSE than the sample variance estimate).

0.4.5 The method of maximum likelihood

The method of maximum likelihood is probably the workhorse method for parametric models. You’ll take a whole course on it next semester. We’ll cover the most basic aspects of it here and some properties later. Maximum likelihood asks the question: what parameter values for parametric model f are most likely given the observed data. Specifically we want to know the likelihood of parameter guess θ given observed X. For a sequence of independent data $x_1, \dots, x_N$ this becomes

$$\mathcal{L}(\theta|x) = \prod_{i=1}^{N} f(x|\theta).$$

Where f is the pdf or pmf of the parametric model in question. We want to find the value of θ that maximizes the function. More often it's easier to deal with the log of this function. Before doing that here are some important properties of logs

$$\begin{aligned}
\log(xy) &= \log(x) + \log(y) \\
\log \prod_{i=1}^{N} x_i &= \sum_{i=1}^{N} \log(x_i) \\
\log(e^x) &= x \\
e^{\log(x)} &= x \\
\log(y^x) &= x\log(y) \\
\log(x/y) &= \log(x) - \log(y) \\
\log(x) &= \text{undefined}, x < 0 \\
\log(0) &= -\infty \\
\log(x) &< 0,\ \ x \in (0,1) \\
\log(1) &= 0 \\
\log(x) &> 0,\ \ x > 1 \\
D_x \log(x) &= 1/x > 0 \text{ Strictly increasing for } x > 0 \\
D_x^2 \log(x) &= -1/x^2 < 0 \text{ Concave function}
\end{aligned}$$

```r
curve(log(x), from=0, to=10, n=100001)
```

Returning to the problem we have

$$L(\theta|x) = \sum_{i=1}^{N} \log\left(f(X|\theta)\right).$$

Let's work on the uniform problem from above, here we have

$$f(x|\theta) = \begin{cases} \frac{1}{\theta} & x \in [0, \theta] \\ 0 & \text{otherwise.} \end{cases}$$

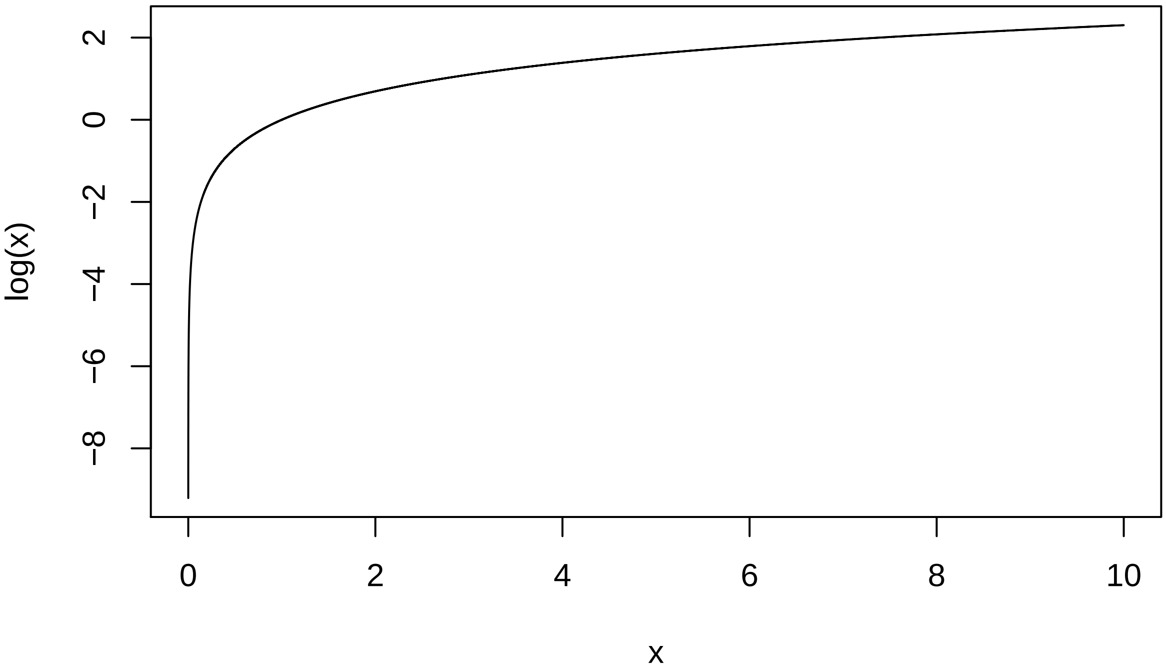


Figure 11: The log function

So our log-likelihood becomes

$$L(\theta|x) = \sum_{i=1}^{N} \begin{cases} -\log(\theta) & x_i \in [0, \theta] \\ -\infty & \text{otherwise.} \end{cases}$$

or

$$L(\theta|X) = \begin{cases} -N\log(\theta) & x \in [0, \theta] \; \forall \; i \\ -\infty & \text{otherwise.} \end{cases}$$

We can ignore the N because it's a constant. So what value of θ maximizes this function? Note that L is strictly decreasing in θ (i.e., $D_\theta L(\theta|X) = -N/\theta$ where it is defined). So we want the smallest value of θ, but only to a point. Only to what point? We don't θ to be larger than the sample values! Why? If we pick $\theta < \max(X)$ then $L(\theta|X) = -\infty$ (this is called a **zero likelihood problem**). So smallest value of θ that avoids this problem is $\max(X)$. This is the maximum likelihood estimator.

Another way to think about this is to recast it as a simple constrained optimization problem

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}(\theta)$$
$$s.t.\ \theta \geq \max(X)$$

this is only satisfied at $\hat{\theta} = \max(X)$ Also notice that this estimate is certainly biased, why? Note that $\max(X) < \theta$ with near certainty (although this bias will decease as N increases). Finally note that if we had defined the uniform pdf to not include the end points

$$f(x|\theta) = \begin{cases} \frac{1}{\theta} & x \in (0, \theta) \\ 0 & \text{otherwise.} \end{cases}$$

then the MLE doesn't exist. At $\hat{\theta} = \max(X)$ we now hit a zero likelihood problem. The ML estimate can always improve so long as it gets arbitrarily close to $\max(X)$. This example demonstrates that ML estimates won't always exist for any given problem.

Now let's consider the normal distribution. Recall that the normal pdf is given by

$$f(x|\sigma^2, \mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

The log likelihood is then

$$L(\sigma^2, \mu|x) = \sum_{i=1}^{N} \left(-\frac{1}{2} \log\left(\sigma^2\right) - \log(\sqrt{2\pi}) - \frac{1}{2\sigma^2} (x_i - \mu)^2 \right)$$

We can remove constant terms because they don't matter for maximization

$$L(\sigma^2, \mu|x) = -\frac{N}{2} \log\left(\sigma^2\right) - \sum_{i=1}^{N} \left(\frac{1}{2\sigma^2} (x_i - \mu)^2 \right)$$

To find the maxima and minima we take derivatives and set them equal to 0. As such

$$\begin{aligned} D_\mu L(\sigma^2, \mu|x) &= \sum_{i=1}^{N} \frac{1}{\sigma^2}(x_i - \mu) \\ 0 &= \frac{1}{\sigma^2} \sum_{i=1}^{N} (x_i - \mu) \\ 0 &= \sum_{i=1}^{N} (x_i - \mu) \\ 0 &= N\mu + \sum_{i=1}^{N} x_i \\ N\mu &= \sum_{i=1}^{N} x_i \\ \hat{\mu}_{MLE} &= \frac{1}{N} \sum_{i=1}^{N} x_i \end{aligned}$$

and

$$\begin{aligned}
D_{\sigma^2} L(\sigma^2, \mu|x) = 0 = -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{N}(x_i-\mu)^2 \\
\frac{N}{2\sigma^2} = \frac{1}{2\sigma^4}\sum_{i=1}^{N}(x_i-\mu)^2 \\
\frac{2\sigma^4}{2\sigma^2} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2 \\
\hat{\sigma}^2_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2
\end{aligned}$$

The MLE estimate of μ is unbiased while the estimate of σ^2 is biased.

A super cool thing about MLE is **Invariance**: If $\hat{\theta}$ is an maximum likelihood estimate of θ then $g(\hat{\theta})$ is an MLE of $g(\theta)$. This is insanely useful. We'll cover the other important properties of MLE when we get to asymptotics.

In these two examples the estimates have closed form solutions. It is nice when this happens, but rare. Most of the time you'll need to use numerical optimization to solve the maximization problem. This can be quite involved if the problem gets sufficiently complex. A real life example: suppose we didn't want to do calculus to find estimates of σ^2 and μ above. We could do the following

```r
X <- rnorm(1000, mean=-2, sd=2) #mu=-2, sigma^2=4
log.lik <- function(theta, X){
  # Log likelihood function for mean and variance of the normal
  # inputs:
  #     theta: the parameter vector. theta[1] = mu, theta[2] =log(sigma^2)
  #     X: the data
  # outputs: the negative of the log-likelihood
  ## Notes:
  ## 1. For numerical optimization you need the computer to be able
  ##    to guess any value. estimating theta[2]= log(sigma^2) allows this.
  ##    Now log(sigma^2) can be any real number, while exp(theta[2])=sigma^2
  ##    will always be positive as required
  ## 2. optim is a minimizing program so we minimize the negative
  ##    log-likelihood instead of maximizing the log-likelihood
  mu <- theta[1]
  sigma2 <- exp(theta[2])
  N <- length(X)
  LL <- -(N/2) *log(sigma2) - 1/(2*sigma2)*sum( (X-mu)^2)
  return(-LL)
}
gradient <- function(theta, X){
  # gradient of the log likelihood function for mean and variance of
  #     the normal
  # inputs:
  #     theta: the parameter vector. theta[1] = mu, theta[2] =log(sigma^2)
  #     X: the data
  # outputs: the negative of first derivative of the log-likelihood wrt
  #     to theta[1] and theta[2]
  ## Notes:
  ## 1. For numerical optimization you need the computer to be able
  ##    to guess any value. estimating theta[2]= log(sigma^2) allows this.
  ##    Now log(sigma^2) can be any real number, while exp(theta[2])=sigma^2
  ##    will always be positive as required
  ## 2. optim is a minimizing program so we minimize the negative gradient
  mu <- theta[1]
  sigma2 <- exp(theta[2])
  N <- length(X)
  Dmu <- -1/(sigma2)*sum(mu-X)
  Dln.sigma2 <- -(N/2) + 1/(2*sigma2)*sum( (X-mu)^2 )
  return(-c(Dmu, Dln.sigma2))
}
ML.ests <- optim(c(0,0),  #starting guesses
                 fn = log.lik, gr=gradient, #functions
                 X=X,  #fixed inputs
                 method="BFGS") #other stuff
ML.ests$par #MLE for mu and for Log(sigma^2)
```

```
## [1] -2.041597  1.440262
```

```r
exp(ML.ests$par[2]) #this is still the MLE for sigma^2  INVARIANCE!
```

```
## [1] 4.221803
```

Note that in the above, the gradient function does not look exactly like the first derivative we discussed above. This is because the parameters are now $\theta = (\mu, \log(\sigma^2))$. As such the gradient we supply to `optim` needs to be with respect to $\theta = (\mu, \log(\sigma^2))$. Gradients are optional, but very helpful to most numeric optimization programs, I always recommend them unless it's too difficult to derive. In this example, we can reparameterize the log-likelihood in terms of θ:

$$\begin{aligned}
L(\theta|x) &= -\frac{N}{2}\log\left(\exp(\theta_{[2]})\right) - \sum_{i=1}^{N}\left(\frac{1}{2\exp(\theta_{[2]})}\left(x_i - \theta_{[1]}\right)^2\right) \\
&= -\frac{N\theta_{[2]}}{2} - \sum_{i=1}^{N}\frac{1}{2e^{\theta_{[2]}}}\left(x_i - \theta_{[1]}\right)^2 \\
D_{\theta_{[1]}}L(\theta|x) &= -\sum_{i=1}^{N}\frac{1}{2e^{\theta_{[2]}}}2\left(\theta_{[1]} - x_i\right) \\
&= -\sum_{i=1}^{N}\frac{1}{\sigma^2}\left(\mu - x_i\right) \\
D_{\theta_{[2]}}L(\theta|x) &= -\frac{N}{2} - \sum_{i=1}^{N}\frac{-1}{2e^{\theta_{[2]}}}\left(x_i - \theta_{[1]}\right)^2 \\
&= -\frac{N}{2} + \sum_{i=1}^{N}\frac{(x_i - \mu)^2}{2\sigma^2}
\end{aligned}$$

0.4.6 Large sample properties of estimators

That was a fun detour through estimation and testing. The last thing we'll do in Math Camp are some important results that help us understand the asymptotic properties of estimators. We'll start with a focus on sums Let $Y = X_1 + \ldots + X_N$ be a sum of iid random variables where $\mathrm{E}[X_i] = \mu$ and $\mathrm{Var}(X_i) = \sigma^2$. From the above we know that $\mathrm{E}[Y] = N\mu$ and $\mathrm{Var}(Y) = N\sigma^2$.

Finally we will need to be familiar with the **Markov Inequality** to get through some of this. The Markov inequality states that for a non-negative random variable X and a positive constant a, $\Pr(X > a) \leq \frac{\mathrm{E}[X]}{a}$. Intuitively, this follows from the following rearrangement of terms

$$\begin{aligned}
\mathrm{E}[X] &= \Pr(X < a)\,\mathrm{E}[X|X < a] + \Pr(X \geq a)\,\mathrm{E}[X|X \geq a] \\
&\geq a\Pr(X \geq a) \\
\Pr(X \geq a) &\leq \frac{\mathrm{E}[X]}{a}
\end{aligned}$$

These inequalities follow because $\mathrm{E}[X|X \geq a] \geq a$ and $\Pr(X < a)\,\mathrm{E}[X|X < a] \geq 0$

We can now state a few important results.

Theorem 6 (Weak Law of Large Numbers) *If $X_1, \ldots, X_N$ are iid with expectation μ and variance σ^2 (both finite) then for all $\varepsilon > 0$*

$$\lim_{N \to \infty} \Pr\left(\left|\frac{Y}{N} - \mu\right| \geq \varepsilon\right) = 0$$

Note that in words this says that the random variable Y/N **converges in probability** to μ. Convergence in probability is sometimes denoted $Y/N \xrightarrow{p} \mu$ or with plim. What it means is that for some positive ε and some value δ there is a sample size N such that for an $N' > N$ $\Pr(|Y/N - \mu| > \varepsilon) < \delta$.

Proof. Let $Z = (Y/N - \mu)^2$ and consider a $\varepsilon^2 > 0$. By the Markov inequality we have

$$\Pr(Z \geq \varepsilon^2) \leq \frac{\mathrm{E}[(Y/N - \mu)^2]}{\varepsilon^2}.$$

We rewrite the left-hand side by taking the positive square root of both sides, and the right-hand side is the variance of $Y/N = \sigma^2/N$ (this is now Chebyshev's inequality)

$$\Pr(|Y/N - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{N\varepsilon^2}.$$

Now for any $\varepsilon > 0$, the right-hand side tends to 0 as $N \to \infty$. As such the left-hand will tend to 0 as well, which is what we needed to show. □

The LLN justifies our "long run" approach to understanding probability. It tells that as we observe more iid realizations of a random variable the mean of those realizations will converge to the true expected value. Note that the LLN does not actually requires the finite variance assumption. However, it makes the proof easier, which is why I included it.

Our next result is a central tenant of statistical analysis.

Theorem 7 (Central Limit Theorem) *Let $X_1, \ldots, X_N$ be iid random variables with expected value μ and variance σ^2 (both finite), then for all $x \in \mathbb{R}$*

$$\lim_{N \to \infty} \left(\frac{\sqrt{N}(Y/N - \mu)}{\sigma}\right) \sim N(0, 1).$$

The proof here involves using moment generating functions (mgfs) to show that the mgfs of the right hand side converge to the normal mgfs.

To start, let $X_i^* = (X_i - \mu)/\sigma$. The X_i^* are iid with mean 0 and variance 1. And let

$$Z = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} X_i^* = \frac{\sqrt{N}(Y/N - \mu)}{\sigma}.$$

The mgf of X_i^* is

$$M(s) = \mathrm{E}[e^{sX_i^*}]$$

A full Taylor series expansion of this mgf around 0

$$\begin{aligned} M(s) &= M(0) + sD_sM(0) + \frac{1}{2}s^2D_s^2M(0) + \frac{1}{3!}s^3D_s^3M(0) + \dots \\ &= 1 + s\,\mathrm{E}[X_i^*] + \frac{1}{2}s^2\,\mathrm{E}[X_i^{*2}] + \frac{1}{3!}s^3D_s^3M(0) + \dots \\ &= 1 + \frac{1}{2}s^2 + \frac{1}{3!}s^3D_s^3M(0) + \dots \end{aligned}$$

Because the X_i^* are iid we can describe that the mgf of Z, specifically we can apply theorems 4 and 5 to get:

$$\begin{aligned} M_Z(s) &= \prod_{i=1}^{N} M\left(\frac{s}{\sqrt{N}}\right) \\ &= M\left(\frac{s}{\sqrt{N}}\right)^N \\ &= \left(1 + \frac{1}{2N}s^2 + \frac{1}{3!N^{3/2}}s^3D_s^3M(0) + \dots\right)^N \end{aligned}$$

What happens to this in the limit? One fun fact that you may want to know is

Fact 1 *Suppose we have a sequence* a_N *such that* $\lim_{N\to\infty} a_N = b$ *then* $\lim(1 + a/N)^N = e^b$.

Let

$$a_N = \frac{1}{2}s^2 + \frac{1}{3!N^{1/2}}s^3D_s^3M(0) + \dots,$$

what is the limit here as $N \to \infty$?

$$\lim_{N\to\infty} a_N = \lim_{N\to\infty} \left(\frac{1}{2}s^2 + \frac{1}{3!N^{1/2}}s^3D_s^3M(0) + \dots\right) = \frac{s^2}{2} = b.$$

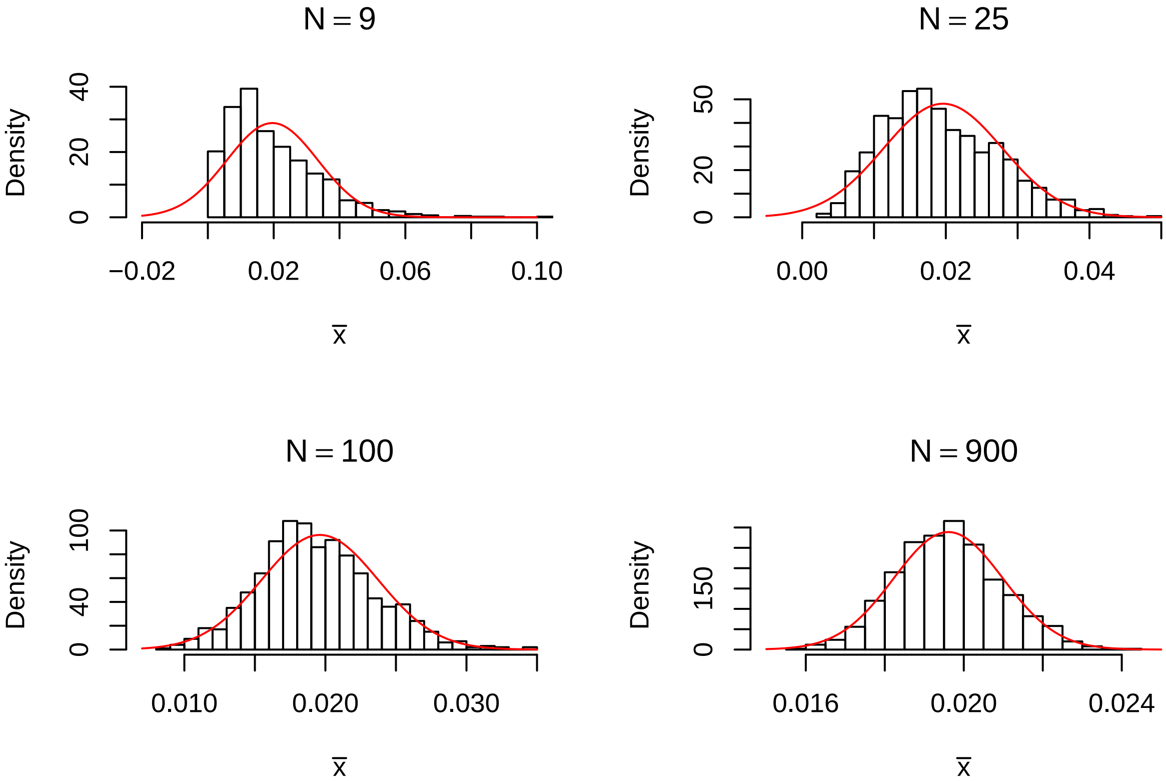
So now we can apply the fun fact to get

$$\lim_{N\to\infty} M_Z(s) = e^{s^2/2}.$$

This is the standard normal mgf ($\mu = 0, \sigma^2 = 1$). This is not a complete proof, but it gives you an idea as to what a full proof would entail.

More importantly it tells us that *regardless of the distribution of X_i* the sample mean will be normally distributed for large N. This means that we can use z tests for any hypothesis about a true population mean for any underlying population so long as we have enough data. For example, let's look at a Beta(.2,10) the expected value is about 0.02.

```r
#CLT demo
set.seed(1)
samples9 <- colMeans(replicate(1000, rbeta(9,shape1 = .2,shape2=10)))
samples25 <- colMeans(replicate(1000, rbeta(25,shape1 = .2,shape2=10)))
samples100 <- colMeans(replicate(1000, rbeta(100,shape1 = .2,shape2=10)))
samples900 <- colMeans(replicate(1000, rbeta(900,shape1 = .2,shape2=10)))
true.mu <- .2/(.2+10)
true.sigma <- sqrt(2/(10.2^2 * 11.2))
par(mfrow=c(2,2))
hist(samples9, freq=F, breaks="Scott", xlim=c(-.02,.1),
     main=expression(N==9), xlab=expression(bar(x)))
curve(dnorm(x, mean=true.mu, sd=true.sigma/3), add=T, col="red")
hist(samples25, freq=F, breaks="Scott", xlim=c(-.005,.05), main=expression(N==25),
     xlab=expression(bar(x)))
curve(dnorm(x, mean=true.mu, sd=true.sigma/5), add=T, col="red")
hist(samples100, freq=F, breaks="Scott", xlim=c(0.007,.035),
     main=expression(N==100), xlab=expression(bar(x)))
curve(dnorm(x, mean=true.mu, sd=true.sigma/10), add = T, col="red")
hist(samples900, freq=F, breaks="Scott", main=expression(N==900),
     xlim=c(0.015,.025), xlab=expression(bar(x)))
curve(dnorm(x, mean=true.mu, sd=true.sigma/30), add=T, col="red")
```



Increasingly normal looking.

Now that we're in asymptotic land, we can introduce one new property of an estimator. An estimator is **consistent** if it converges in probability to the true but unknown parameter. An easy way to think about this is that the estimator is asymptotically unbiased and has a variance that is vanishingly small as N increases. Note that biased estimators can be consistent and that unbiased estimators can be inconsistent. Formally, an estimator is consistent if and only if

$$\hat{\theta}_n \xrightarrow{p} \theta.$$

Or if its bias and variance both go to zero. In problem set 0, you will demonstrate consistency for the MLE of the uniform problem we considered before. This will be generally true of MLE estimators under very general conditions.

Theorem 8 (Consistency of Maximum likelihood estimators) *Under some regularity conditions on $f(y;\theta)$ the MLE $\hat{\theta}$ is consistent.*

We won't go into too much detail here in the interest of time, but we can sketch a heuristic proof. Let's assume that we have a unique MLE $\hat{\theta}$ and the log likelihood function

$$L(\theta|y) = \frac{1}{N}\sum_{i=1}^{N}\log(f(y_i|\theta)).$$

Now consider the true but unknown θ (for ease let's call it θ^*). The law of large numbers tells us that

$$\frac{1}{N}\sum_{i=1}^{N}\log(f(y_i|\theta)) \xrightarrow{p} \mathrm{E}_{\theta^*}[\log(f(Y|\theta))].$$

We know that $\hat{\theta}$ is the maximum of the left hand side, we want to show that the maximum of the right hand side is θ^*.

Let's consider the right hand side in more detail. Specifically, let's consider the gap between the expectation of the log likelihood at the true value versus any other choice of θ:

$$\mathrm{E}_{\theta^*}[\log(f(Y|\theta))]-\mathrm{E}_{\theta^*}[\log(f(Y|\theta^*))] = \mathrm{E}_{\theta^*}[\log(f(Y|\theta))-\log(f(Y|\theta^*))] = \mathrm{E}_{\theta^*}\left[\log\left(\frac{f(Y|\theta)}{f(Y|\theta^*)}\right)\right].$$

We're going to now apply the second part of Jensen's inequality which says that concave (negative 2nd derivatives) functions g, $\mathrm{E}[g(x)] \leq g(\mathrm{E}[x])$. Note that logs are concave so

$$\begin{aligned}
\mathrm{E}_{\theta^*}\left[\log\left(\frac{f(Y|\theta)}{f(Y|\theta^*)}\right)\right] &\leq \log\left(\mathrm{E}_{\theta^*}\left[\frac{f(Y|\theta)}{f(Y|\theta^*)}\right]\right)\\
&\leq \log\left(\int_{-\infty}^{\infty}\frac{f(y|\theta)}{f(y|\theta^*)}f(y|\theta^*)dy\right)\\
&\leq \log\left(\int_{-\infty}^{\infty}f(y|\theta)dy\right)\\
&\leq \log(1) = 0.
\end{aligned}$$

All this is to say that the gap between $\mathrm{E}_{\theta^*}[\log(f(Y|\theta))]$ and $\mathrm{E}_{\theta^*}[\log(f(Y|\theta^*))]$ is weakly negative for any θ, so $\mathrm{E}_{\theta^*}[\log(f(Y|\theta))]$ obtains a maximum at θ^*. As such $\hat{\theta}$ converges to θ^* under the law of large numbers.

We now know the large sample expected value of maximum likelihood estimates, but what about the variance? Define the **Fisher Information** contained in a sample $X_1, \ldots, X_N$ as a measurement of the amount of information contained in X about the unobserved parameter θ. Note that if the log-likelihood $L(\theta|X)$ is sharply peaked with respect to θ than it is easy to find the true value. In this case X provides a lot of information about θ. If $L(\theta|X)$ is very flat with respect to different values of θ there is little-to-no information to be obtained from X. We can formalize this thinking by defining the Fisher Information as

$$I(\theta) = \mathrm{E}_{\theta}[D_{\theta}L(\theta|X)^2],$$

which is the variance of the first order condition.

Under some regularity conditions on f and L that you'll learn about next semester it can be shown that

$$I(\theta) = -\,\mathrm{E}_\theta[D^2_\theta L(\theta|X)].$$

When the Fisher Information is large, it means that the likelihood is sharp at ML estimate and the variance in our estimate is small (we're more confident in $\hat{\theta}$). In the limit the variance of the ML estimate convergences to

$$\mathrm{Var}(\hat{\theta}_n) \xrightarrow{p} \frac{1}{I(\theta)},$$

this boundary is called the **Cramér-Rao lower bound** for an unbiased estimator and represents the lowest possible variance that can be obtained for an unbiased estimator. Putting this all together, we can say that as N increases, MLE will become best unbiased estimator (no other unbiased estimator will have lower variance); with enough data you can't do better than MLE if you have the right parametric assumption. Additionally, ML estimates (like the sample mean) will be asymptotically normal under these conditions (a result of the central limit theorem and some other theorems we haven't introduced yet).

To recap, under common conditions the maximum likelihood estimate of θ is consistent and asymptotically normal such that

$$\begin{aligned}
\sqrt{N}(\hat{\theta} - \theta) &\xrightarrow{d} N\left(0, \left(\frac{I(\theta)}{N}\right)^{-1}\right) \\
\hat{\theta} &\overset{asy}{\sim} N\left(\theta, I(\theta)^{-1}\right) \\
\hat{\theta} &\xrightarrow{p} \theta.
\end{aligned}$$

This means that we can test hypotheses about θ using z tests if N is large.

A brief aside now that we have the variance of an MLE estimate, we can use the **delta method** to find the variance of a function or transformation of one or more estimates. Recall that we estimated $\theta_{[2]} = \log(\sigma^2)$ in place of σ^2 so that `optim` could take guesses from the entire real line for a parameter that must be positive. We can use the invariance property to find the MLE, but that doesn't help us with the variance. This is where the delta method comes in.

Theorem 9 (The Delta Method) *Let θ_n be a sequence of random variables such that $\sqrt{N}(\theta_n - \theta) \xrightarrow{d} N(0, \sigma^2)$, and $g(\theta)$ is a function such that $D_\theta g(\theta)$ is nonzero and finite, then*

$$\sqrt{N}(g(\theta_n) - g(\theta)) \xrightarrow{d} N(0, \sigma^2(D_\theta g(\theta))^2)$$

The main requirement for using the delta method is that we have a normally distributed estimate that we want to transform. Note that the delta method is just an first order Taylor approximation of $g(\hat{\theta})$ around it's expected value $\mathrm{E}[\hat{\theta}]$. Because this estimator is consistent the approximation will get better as N increases.

$$\begin{aligned}
g(\hat{\theta}) &\approx g(\mathrm{E}[\hat{\theta}]) + D_{\hat{\theta}} g(\mathrm{E}[\hat{\theta}])(\hat{\theta} - \mathrm{E}[\hat{\theta}]) \\
\mathrm{E}(g(\hat{\theta})) &\approx g(\mathrm{E}[\hat{\theta}]) \\
\mathrm{Var}(g(\hat{\theta})) &\approx \left(D_{\hat{\theta}} g(\hat{\theta})\right)^2 \mathrm{Var}(\hat{\theta}) \\
g(\hat{\theta}) &\overset{asy}{\sim} N(g(\theta), D_{\theta} g(\theta)^2 \mathrm{Var}(\theta))
\end{aligned}$$

In our specific example this becomes:

$$\begin{aligned}
\mathrm{Var}(\hat{\sigma}^2) &\approx \left(D_{\hat{\theta}_{[2]}} \exp(\hat{\theta}_{[2]})\right)^2 \mathrm{Var}(\hat{\theta}_{[2]}) \\
&\approx \exp(\hat{\theta}_{[2]})^2 \mathrm{Var}(\hat{\theta}_{[2]}),
\end{aligned}$$

and we can apply that to our problem to the parameter and standard error we actually care about.

```r
#Redo the estimation but ask for a Hessian estimate too
ML.ests <- optim(c(0,0),  #starting guesses
                 fn = log.lik, gr=gradient, #functions
                 X=X,  #fixed inputs
                 hessian=TRUE, #ask it to estimate a hessian
                 method="BFGS") #other stuff
Hessian <- ML.ests$hessian
c(ML.ests$par[1],  sqrt(1/(Hessian)[1,1])) #MLE for mu
```

```
## [1] -2.0415967  0.0649754
```

```r
se.sigma2 <- sqrt(exp(ML.ests$par[2])^2 * 1/(Hessian)[2,2])
c(exp(ML.ests$par[2]),  se.sigma2) #MLE for sigma2
```

```
## [1] 4.2218028 0.1888054
```

1 Correlation and simple regression

We will begin our (real) time together with a discussion of what regression is and what it's good for. To start: **simple regression** is used describes the relationship between a single independent variable X and a dependent variable y. Regression has three main uses that we'll encounter. It can be used to:

1. **Predict** values of y for new values of X that maybe we haven't observed yet (e.g., forecasting election outcomes using unemployment rates or predict when a civil war will begin)
2. **Describe** how do changes in y match changes in X (on average voters of color tend to vote at lower rates than white votes)
3. **Identify the treatment (causal) effect** that a change in x_i will have on values of y_i (i.e., How many more votes can a candidate get by spending another dollar, on average?)

We will often have one or more of these goals in mind when conducting political science research, and in some cases it makes sense for them to overlap. Other times, focusing on one (say making the best forecast) may make another goal more difficult (like finding a treatment effect). For example, a study that describes or identifies the relationship between October unemployment and the incumbent vote share in November may help us make good predictions about the election outcome. Alternatively, a Republican presidential candidate may spend money in states where the Democratic candidate is also spending a lot of money and as such we may predict that Republicans do worse in states where they spent money on ads even if the true casual relationship is different.

When we discussed covariance and correlation of random variables during Math Camp we said that they are measures of the linear relationship between X and y. As such, they will form the basis of our discussion here. Recall that the true, but unknown, relationship between X and y is unobserved to us. This population covariance is defined as

$$\sigma_{xy} = \mathrm{E}[(X - \mu_x)(y - \mu_y)],$$

which we will estimate using the sample covariance

$$s_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}).$$

This extends our unbiased variance estimator s^2 to the covariance case. Note that s_{xy} is also an unbiased estimator of the population covariance, it is sign-interpreted, and values of 0 suggest that there is no *linear* relationship between X and y in the population. The drawback here is that covariance has weird units so we often look at correlation instead.

The population correlation is given as $\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$ while its sample counterpart is given as $r_{xy} = \frac{s_{xy}}{s_x s_y}$. Some examples are shown with this figure. Correlation is bounded to be between -1 and 1. Values further from 0 indicate stronger *linear* relations. Values at 0 indicate *linear* independence. The magnitude of the variables does not affect the correlation as it is a unitless measure. However, correlation still does not tell us everything we could want to know about the relationship between X and y.

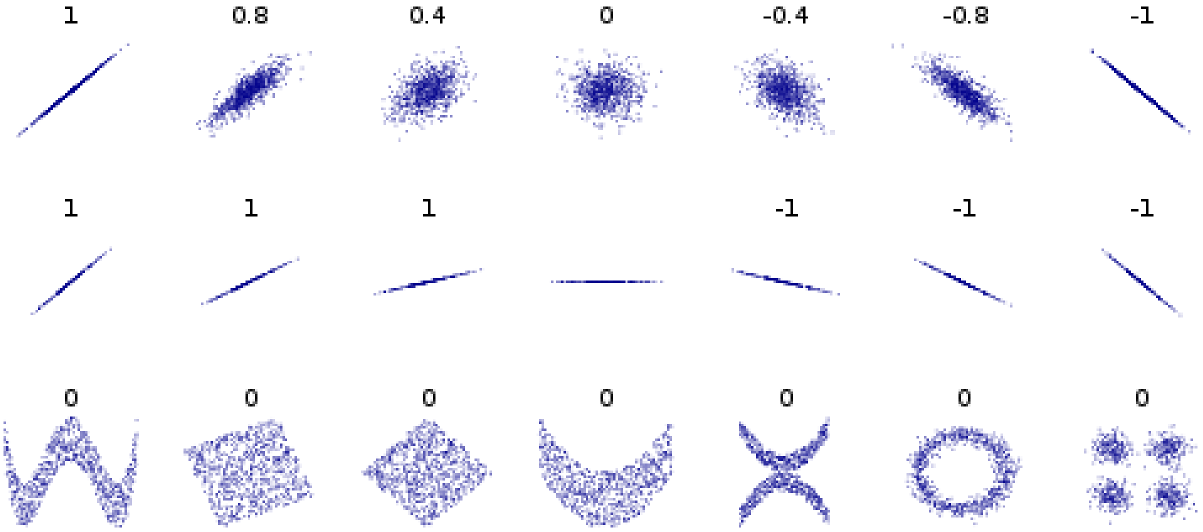


Figure 12: Various correlations

1.1 Simple regression

Consider predicting an incumbents vote share from based on the unemployment rate. For example, suppose we have data on the unemployment rate nine months before the election. One way we could form predictions is by using a model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i. \quad \text{Assumption A1}$$

Let's take a moment to work through everything going on here, we have

1. Outcome variable y_i: incumbent party vote share
2. Independent variable x_i: unemployment rate 9 months before an election

3. β_0: a parameter that centers our model (intercept)
4. β_1: a parameter that relates X and y (slope)
5. ε_i: a parameter that allows for distance between data points and the line formed by the rest of the model (error term)

The error term is essential here as we *know* that our data will never form an exact line implied by an error-less model. There is no perfect linear relationship between X and y for any this or any other interesting question. Even presidents who face very similar economies will have different outcomes through debates, gaffs, personality, foreign crises, shark attacks, weather, sun spots, etc. In this case each ε_i represents a single draw from a population distribution of all the other factors not included in our linear model.

Assumption A1 tells us that we relate X to y using a linear specification with all other factors "pushed" into ε. This assumption puts the linear in linear models, but we'll need a few more assumptions to get anywhere interesting. Specifically,

Assumption A2 x_i are fixed (not random variables).

Assumption A3 $s_x^2 > 0$ The variance in X is greater than 0.

Assumption A4 $\mathrm{E}[\varepsilon_i] = 0$.

Assumption A5 $\mathrm{Var}[\varepsilon_i] = \mathrm{E}[\varepsilon_i^2] = \sigma_\varepsilon^2$ (called homoskedasticity).

Assumption A6 ε_i and ε_j are independent for $i \neq j$.

Assumption A7 ε_i are normally distributed.

These 7 assumptions form the classic regression model. The population regression model is defined by three parameters $(\beta_0, \beta_1, \sigma_\varepsilon^2)$. Our goal (in order to say predict vote share) is to find estimates of these parameters ($\hat{\beta}_0$, $\hat{\beta}_1$ and $\hat{\sigma}_\varepsilon^2$). To produce these estimates we will follow a simple "procedure." For a given "guess" of the parameters we can produce predictions of y (call them $\hat{y}$) and compare these predictions to the true y values. Prediction error then becomes the difference between $y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + x_i\hat{\beta}_1)$. We want to minimize the prediction error and since we don't care if we over or underpredict (just that we get as close as possible) we will use the squared prediction error to get rid of signs. Further if we treat all observations equally we get the following optimization problem:

$$(\hat{\beta}_0, \hat{\beta}_1) = \operatorname*{argmin}_{\beta_0,\beta_1} \sum_{i=1}^{N} (y_i - \beta_0 - x_i\beta_1)^2.$$

This is to say we want to draw a line through our data that gets it as close as possible to data as a whole, or that we want to minimize the total gap between reality (data) and model (the regression line).

Using calculus we can solve this problem by taking the derivatives with respect to β_0 and β_1. Starting with β_0

$$\begin{aligned}
D_{\beta_0}\sum_{i=1}^{N}(y_i-\beta_0-x_i\beta_1)^2 &= -2\sum_{i=1}^{N}(y_i-\beta_0-x_i\beta_1)\\
0 &= \sum_{i=1}^{N}y_i-\sum_{i=1}^{N}\beta_0-\sum_{i=1}^{N}x_i\beta_1\\
&= \frac{1}{N}\sum_{i=1}^{N}y_i-\frac{1}{N}\sum_{i=1}^{N}\beta_0-\frac{1}{N}\sum_{i=1}^{N}x_i\beta_1\\
&= \bar{y}-\beta_0-\bar{x}\beta_1\\
\hat{\beta}_0 &= \bar{y}-\bar{x}\beta_1.
\end{aligned}$$

That's a start, but we need to find $\hat{\beta}_1$ in order to sub it into the above.

$$\begin{aligned}
D_{\beta_1}\sum_{i=1}^{N}(y_i-\beta_0-x_i\beta_1)^2 &= -2\sum_{i=1}^{N}x_i(y_i-\beta_0-x_i\beta_1)\\
0 &= \sum_{i=1}^{N}x_iy_i-\sum_{i=1}^{N}x_i\beta_0-\sum_{i=1}^{N}x_ix_i\beta_1\\
&= \frac{1}{N}\sum_{i=1}^{N}x_iy_i-\frac{1}{N}\sum_{i=1}^{N}x_i\beta_0-\frac{1}{N}\sum_{i=1}^{N}x_ix_i\beta_1\\
&= \frac{1}{N}\sum_{i=1}^{N}x_iy_i-\bar{x}\beta_0-\frac{1}{N}\sum_{i=1}^{N}x_i^2\beta_1\\
\frac{1}{N}\sum_{i=1}^{N}x_iy_i &= \bar{x}\beta_0+\frac{1}{N}\sum_{i=1}^{N}x_i^2\beta_1
\end{aligned}$$

Insert the estimate of β_0

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}x_iy_i &= \bar{x}\bar{y}-\bar{x}^2\beta_1+\frac{1}{N}\sum_{i=1}^{N}x_i^2\beta_1\\
\beta_1 &= \frac{\frac{1}{N}\sum_{i=1}^{N}x_iy_i-\bar{x}\bar{y}}{\frac{1}{N}\sum_{i=1}^{N}x_i^2-\bar{x}^2}\\
&= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2}
\end{aligned}$$

After all that we are left with

$$\hat{\beta}_1 = \frac{s_{xy}}{s_x^2}$$
$$\hat{\beta}_0 = \bar{y} - \bar{x}\hat{\beta}_1,$$

so long as $s_x^2 > 0$ (Assumption A3). Note that if $s_x^2 = 0$, then *any* $(\hat{\beta}_0, \hat{\beta}_1)$ that solves $\hat{\beta}_0 = \bar{y} - \bar{x}\hat{\beta}_1$ will satisfy the first order conditions (i.e., there will be an infinite number of estimates). However, so long as the relatively low-key condition that $s_x^2 > 0$ holds we will have unique estimates. These estimates are called the **Ordinary Least Squares** estimates of β_0, β_1. We will verify that the OLS estimates satisfy the second order conditions later.

An intuitive way to rewrite $\hat{\beta}_1$ is to exploit the relationship between correlation and covariance to get

$$\hat{\beta}_1 = r_{xy}\frac{s_y}{s_x}.$$

This approach makes the units of $\hat{\beta}_1$ even more clear and it also makes it clear that the regression coefficient is just a scaled measure of *linear* correlation. We can interpret $\hat{\beta}_1$ as the average increase in y for every 1 unit increase in X (i.e., $\hat{\beta}$ is measured in y's per X). In our unemployment example, this becomes, "For every 1 percentage point increase in the unemployment rate nine months before an election, the incumbent party vote share decreases by $\hat{\beta}_1$ percentage points, on average."

We still have one parameter left to pin down, σ_ε^2. One place to start is by defining an estimate of ε_i using Assumption A1.

$$\hat{\varepsilon}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i.$$

Recall that $\sigma_\varepsilon^2 = \text{Var}(\varepsilon_i)$ by Assumption A5, which suggests the estimator:

$$\hat{\sigma_\varepsilon^2} = \frac{1}{N-2}\sum_{i=1}^{N}\hat{\varepsilon}_i^2 = \frac{1}{N-2}\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2.$$

Much like using $N-1$ in sample variance we use $N-2$ to get an unbiased estimate.

1.2 Properties of the OLS estimator

We have our OLS estimates, are they any good? Remember from Math Camp that we have some properties we can look at: Bias, Efficiency, Consistency, and Mean Squared Error. We'll consider all of these eventually, but lets start with bias. Recall the definition of bias for an estimator $\hat{\theta}$ is

$$\text{bias}(\hat{\theta}) = \text{E}[\hat{\theta}] - \theta,$$

and an unbiased estimator has 0 bias.

Let's start with the slope $\hat{\beta}_1$ and recall that by (Assumption A2) that x_i is a constant (i.e., not a realization from a random variable). As such,

$$\begin{aligned}
\text{E}[\hat{\beta}_1] &= \text{E}\left[\frac{\frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})^2}\right] \\
&= \frac{\frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})\,\text{E}\left[(y_i - \bar{y})\right]}{\frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})^2} \\
\text{E}\left[(y_i - \bar{y})\right] &= \text{E}\left[y_i - \frac{1}{N}\sum_{i=1}^N y_i\right] \\
&= \text{E}\left[\beta_0 + x_i\beta_1 + \varepsilon_i - \frac{1}{N}\sum_{i=1}^N (\beta_0 + x_i\beta_1 + \varepsilon_i)\right] \\
&= \beta_0 + x_i\beta_1 + \text{E}\left[\varepsilon_i\right] - \frac{1}{N}\sum_{i=1}^N (\beta_0 + x_i\beta_1 + \text{E}\left[\varepsilon_i\right]) \\
&= \beta_0 + x_i\beta_1 - \frac{1}{N}\sum_{i=1}^N (\beta_0 + x_i\beta_1) \\
&= \beta_0 + x_i\beta_1 - \beta_0 - \beta_1\bar{x} \\
&= \beta_1(x_i - \bar{x}) \\
\text{E}[\hat{\beta}_1] &= \frac{\frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})\beta_1(x_i - \bar{x})}{\frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})^2} \\
&= \beta_1.
\end{aligned}$$

The OLS estimate of β_1 is unbiased.

For $\hat{\beta}_0$:

$$\begin{aligned}
\text{E}[\hat{\beta}_0] &= \text{E}[\bar{y}] - \text{E}[\hat{\beta}_1]\bar{x} \\
&= \frac{1}{N}\sum_{i=1}^N (\beta_0 + x_i\beta_1 + \text{E}[\varepsilon_i]) - \beta_1\bar{x} \\
&= \beta_0 + \beta_1\frac{1}{N}\sum_{i=1}^N x_i - \beta_1\bar{x} \\
&= \beta_0.
\end{aligned}$$

The OLS estimate of the constant is also unbiased. We'll consider the other properties in more detail when we consider multiple regression. For now let's just note the variances (under assumptions A1-A6) are

$$
\begin{aligned}
\mathrm{Var}(\hat{\beta}_1) &= \mathrm{Var}\left[\frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2}\right] \\
&= \left(\frac{\sigma_\varepsilon^2}{\sum_{i=1}^{N}(x_i-\bar{x})^2}\right) \\
\mathrm{Var}\,\hat{\beta}_0 &= \mathrm{Var}\left(\bar{y}-\bar{x}\hat{\beta}_1\right) \\
&= \sigma_\varepsilon^2\left(\frac{1}{N}+\frac{\bar{x}^2}{\sum_{i=1}^{N}(x_i-\bar{x})^2}\right)
\end{aligned}
$$

Often times, we want to be able to test hypotheses about what the true value of $\hat{\beta}_1$ is in the population regression. Under Assumptions A1-A7, these tests be formed using t statistics such that:

$$
\begin{aligned}
\frac{\hat{\beta}_1-b_1}{se(\hat{\beta}_1)} &\sim t_{N-2} \\
\frac{\hat{\beta}_0-b_0}{se(\hat{\beta}_0)} &\sim t_{N-2}
\end{aligned}
$$

Note that these look nearly identical to the t tests we considered above that take the form of

$$\frac{\text{Estimate} - \text{Null Value}}{SE(\text{Estimate})}.$$

The individual values change, but not the form. Given what we know about t distributions we can also note that as N increases these distributions will converge into standard normal distributions. Note that the degress of freedom is $N-2$ because we use 2 parameters (β_0 and β_1) to estimate σ_ε^2.

1.3 Application

Let's take a quick break to see an example of simple regression in comparative politics.

```r
library(readstata13)
voting.data <- read.dta13("Rcode/datasets/econ.dta")
head(voting.data)
```

```
##     country ccode elecyr votelead gr_an unem_an gr_m1_an unem_m1_an timeidx
## 1 Australia     1   1946     49.7    NA      NA       NA         NA       1
## 2 Australia     1   1949     46.0    NA      NA       NA         NA       2
## 3 Australia      1    1951      47.6     NA       NA       NA         NA       3
## 4 Australia      1    1954      38.6     NA       NA       NA         NA       4
## 5 Australia      1    1955      39.7     NA       NA       NA         NA       5
## 6 Australia      1    1958      37.2     NA       NA       NA         NA       6
```

Here we have the vote share for the incumbent prime minister's party in OECD countries. Possible independent variables are growth and unemployment. Let's look at a scatterplot of vote share and growth

```
plot(votelead~gr_an, data=voting.data, xlab="Growth", ylab="Vote share")
```

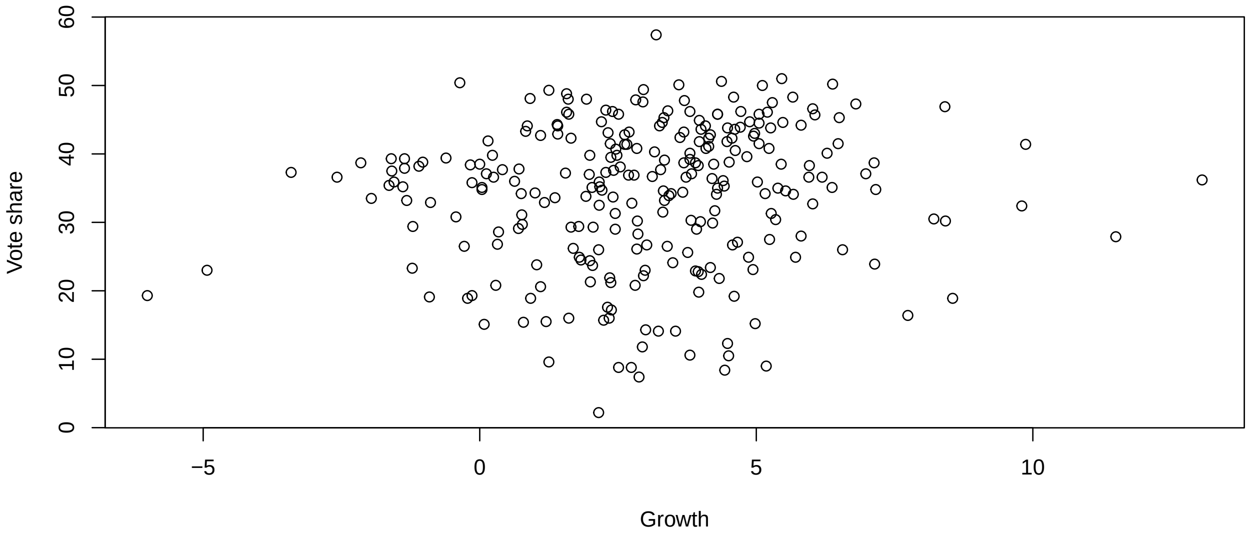


Figure 13: Scatterplot of economic growth and vote sharea across OECD Countries

A visual inspection doesn't allow for a very clear understanding here. We can use some statistics to summarize the data and tell a more coherent story. First, we can consider just the basic correlation between the variables. The correlation is

```
with(voting.data, cor(votelead,gr_an, use="complete"))
```

```
## [1] 0.1041626
```

so we know that if there is a relationship, it appears to be positive. How about a regression?

```
voting.model <- lm(votelead~gr_an, data=voting.data)
summary(voting.model)
```

```
##
## Call:
## lm(formula = votelead ~ gr_an, data = voting.data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.555  -7.393   2.164   7.671  23.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  32.8277     0.9975  32.910   <2e-16 ***
## gr_an         0.4313     0.2530   1.705   0.0894 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.44 on 265 degrees of freedom
##   (119 observations deleted due to missingness)
## Multiple R-squared:  0.01085,    Adjusted R-squared:  0.007117
## F-statistic: 2.907 on 1 and 265 DF,  p-value: 0.08938
```

We can note from the scatter plot that both variables appear to be measured in percentage points. So we can interpret our regression estimates with these units. Specifically, here we see that **a 1 percentage point increase in growth is associated with a 0.43 percentage point increase in incumbent party vote share**. The intercept β_0 is the $\text{E}[y_i|X = 0]$, which is to say that the vote share for a prime minister's party in a country with no growth is 32.83 percentage points, on average. If growth was at 5 percentage points some year, we would expect that the incumbent party receives

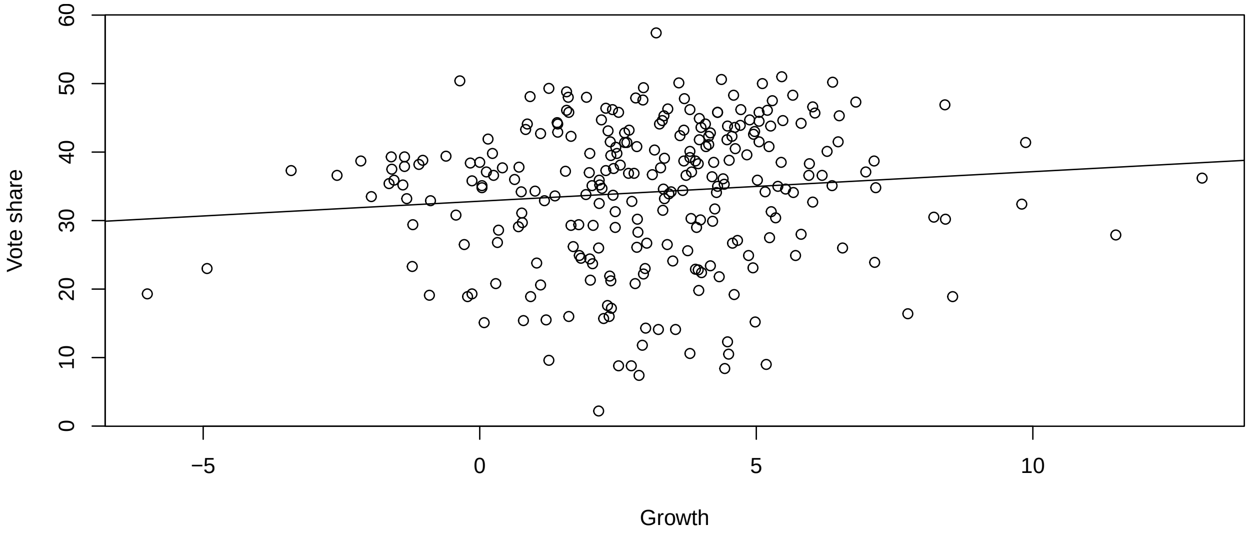
```
32.83+0.43*5
```

```
## [1] 34.98
```

percent of the vote.

We can visualize our regression model

```
plot(votelead~gr_an, data=voting.data, xlab="Growth", ylab="Vote share")
abline(reg=voting.model)
```



Again, the relationship is fairly flat, but positive.

Note that growth in measured during the entire election year. This measurement is slightly unhelpful from a prediction standpoint, as we can't make a prediction for a new election. The data also contain a value of growth from the previous year. Let's try the regression with lagged growth as our predictor variable.

```
voting.model2 <- lm(votelead~gr_m1_an, data=voting.data)
summary(voting.model2)
```

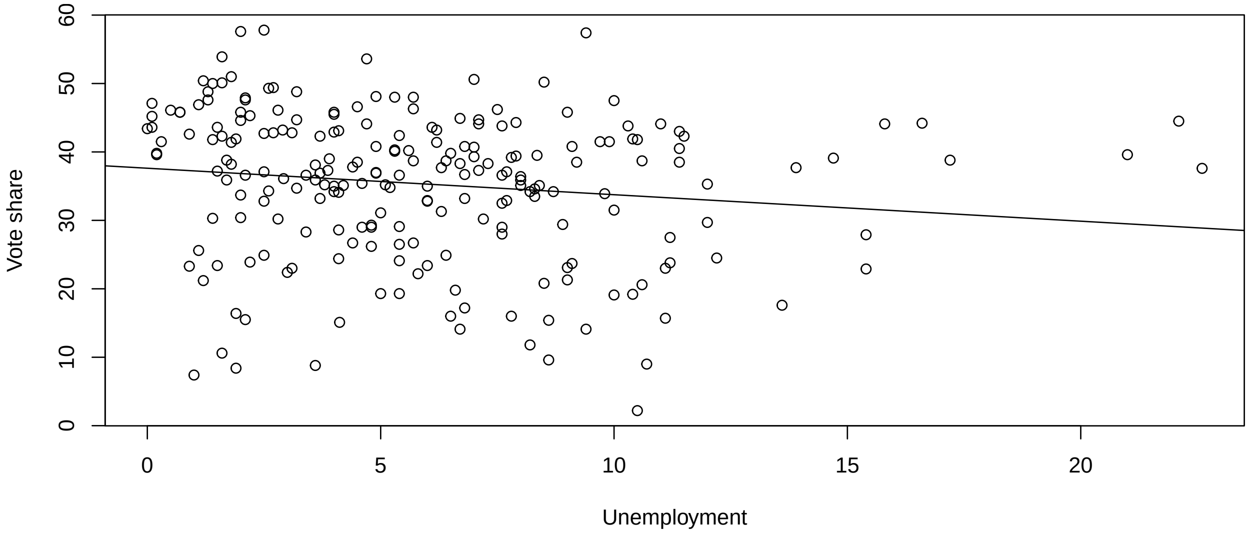
```
##
## Call:
## lm(formula = votelead ~ gr_m1_an, data = voting.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.663  -6.541   1.648   8.029  24.345
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  32.4919     1.0978   29.60   <2e-16 ***
## gr_m1_an      0.4813     0.2750    1.75   0.0813 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.34 on 246 degrees of freedom
##   (138 observations deleted due to missingness)
## Multiple R-squared:  0.0123, Adjusted R-squared:  0.008284
## F-statistic: 3.063 on 1 and 246 DF,  p-value: 0.08133
```

Interestingly, the effect sizes are about the same. Now we see that, on average, a one percentage point increase in growth the year before an election is associated with a 0.48 percentage point increase in vote share. Finally, we can conduct a similar analysis with unemployment

```
plot(votelead~unem_m1_an, data=voting.data, xlab="Unemployment", ylab="Vote share")
voting.model3 <- lm(votelead~unem_m1_an, data=voting.data)
summary(voting.model3)
```

```
##
## Call:
## lm(formula = votelead ~ unem_m1_an, data = voting.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.356  -6.516   1.875   7.873  23.418
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.6196     1.2612  29.829   <2e-16 ***
## unem_m1_an   -0.3870     0.1756  -2.204   0.0285 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.65 on 222 degrees of freedom
##   (162 observations deleted due to missingness)
## Multiple R-squared:  0.02142,    Adjusted R-squared:  0.01701
## F-statistic: 4.858 on 1 and 222 DF,  p-value: 0.02854
```

```
abline(reg=voting.model3)
```



Here we see that the correlation (and thus the regression line) are negative: higher levels of unemployment are associated with an average decrease in the incumbent party vote share. Specifically, for every 1 percentage point increase in last year's unemployment rate there is, on average, a 0.39 percentage point **decrease** in the incumbent party vote share.

In these examples there a few aspects of the simple regression model we might find limiting:

1. At this point we only know how to consider one independent variable per regression. A better model might have both growth *and* unemployment
2. While not a major concern here, we might eventually want to consider variables with non-linear effects on y
3. We made some very strong and questionable assumptions regarding fixed regressors, normal errors, homoskedasticity, and independence of errors
4. If the regressors are themselves random variables we have to worry about how that effects our results on bias.

We can solve all of these with the linear model, but it's going to require some **math!** But some bad news, even just adding a second independent variable makes all of the above algebra *very* hard. Solving the FOC to find the OLS estimates, while possible is a mess with ordinary algebra. To make our life a lot easier we're going to take a detour into **Matrix Algebra**

2 Linear Algebra

Linear algebra (also called matrix algebra) provides us with a very useful set of tools for understanding linear models. We will build up from the most basic parts here.

2.1 Vectors

A **vector** of length N is a collection of N numbers containing elements $v = (v_1, v_2, \dots, v_N)$. One way to think about vector is a point in an N dimensional space where each element denotes a position along a particular axis. As such the zero vector is denoted $0 = (0, 0, \dots, 0)$. This may be confusing, but context should usually make it clear whether 0 is a scalar or vector.

We can perform basic operations on vectors with vectors or vectors with scalars. Let a be a scalar and v a vector then we can perform the following operations

1. Addition $v + a = (v_1 + a, \dots, v_N + a) = (v_i + a)_{i=1}^N$
2. Subtraction $v - a = (v_1 - a, \dots, v_N - a) = (v_i - a)_{i=1}^N$
3. Multiplication $v \times a = (v_1 \times a, \dots, v_N \times a) = (v_i \times a)_{i=1}^N$
4. Division $v/a = (v_1/a, \dots, v_N/a) = (v_i/a)_{i=1}^N$

Operations on two vectors require that they be the same length

1. Addition $v + w = (v_1 + w_1, \dots, v_N + w_N) = (v_i + w_i)_{i=1}^N$
2. Subtraction $v - a = (v_1 - w_N, \dots, v_N - w_N) = (v_i - w_i)_{i=1}^N$

Element-wise multiplication and division are not the default way we think about vectors and need to be specified as special operations when the arise. Let v, w, and x be length N vectors and let a and b be scalars. Some additional properties of vectors and scalars are

Zero is zero $v + 0 = 0 + v = v$

Associative property $(v + w) + x = v + (w + x)$

Commutative property $v + w = w + v$

Distributive law 1 $a(v + w) = av + aw$

Distributive law 2 $(a + b)v = av + bv$

Distributive law 3 $(ab)v = a(bv)$

All the additions can also be subtractions. For example, let $v = (5, 8, 5)$ and $w = (8, 1, 4)$. We can subtract $v - w = (-3, 7, 1)$.

2.2 Matrices

A matrix is a collection $N \times M$ elements ordered into rows and columns such that

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{bmatrix}.$$

As before we use 0 to refer to a special case where all elements are 0

$$0 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

A **square** matrix is any matrix where $N = M$ (same number of rows and columns). A special square matrix is a type of "one" matrix called the **identity matrix**. The identity matrix, I, is a square matrix where all the diagonal elements are 1 and the off-diagonals are 0.

$$I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

Sometimes we will use the notation I_N to denote an identity matrix of size $N \times N$, while othertimes we will just imply the size from the context.

The **transpose** of a matrix is an operation that "flips" a matrix along its diagonal. For example the transpose of the $N \times M$ matrix A from above is the $M \times N$ matrix denoted A' where $[A']_{ji} = [A]_{ij}$ for $i = 1, \dots, N$ and $j = 1, \dots, M$. For example let $A = \begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix}$, then $A' = \begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix}' = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$.

A matrix is **symmetric** when $A = A'$. Put another way $[A]_{ij} = [A']_{ji}$ for all i, j. The zero matrix and the identity matrix are symmetric, while the matrix in the last example is not.

```
A <-  matrix(c(1,3,3,5), nrow=2)
A
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    3    5
```

```
isSymmetric(A)
```

```
## [1] TRUE
```

```
A == t(A)
```

```
##      [,1] [,2]
## [1,] TRUE TRUE
## [2,] TRUE TRUE
```

```
B <- matrix(c(1,3,-3,5), nrow=2)
B
```

```
##      [,1] [,2]
## [1,]    1   -3
## [2,]    3    5
```

```
isSymmetric(B)
```

```
## [1] FALSE
```

```
B == t(B)
```

```
##       [,1]  [,2]
## [1,]  TRUE FALSE
## [2,] FALSE  TRUE
```

```
I = diag(4)
I
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```
isSymmetric(I)
```

```
## [1] TRUE
```

2.2.1 Matrix addition

The addition function is only defined for two cases: Matrix with a scalar or Two matrices of the same size. For the former a matrix plus or multiplied by a scalar performs that operation on every element of the matrix, such that for scalar a and matrix A we have $[aA]_{ij} = a[A]_{ij}$

Suppose that A and B are both $N \times M$ matrices, then their sum is a matrix $A + B$ with elements $[A + B]_{ij} = [A]_{ij} + [B]_{ij}$ for $i = 1, \ldots, N$ and $j = 1, \ldots, M$. The difference is similarly defined with $A - B$ having elements $[A - B]_{ij} = [A]_{ij} - [B]_{ij}$.

Some properties of matrix addition/subtraction for identically sized matrices A, B, and C and scalars a and b

Zero is zero $A + 0 = A - 0 = A$

Commutative property $A + B = B + A$

Associative property $(A + B) + C = A + (B + C)$

Distributive property 1 $a(A + B) = aA + aB$

Distributive property 2 $(a + b)A = aA + bA$

Transpose property $(A + B)' = A' + B'$

For example

```
A = matrix(c(1,4,9,-2), nrow=2)
B = matrix(c(2,0,-10,1), nrow=2)
A
```

```
##      [,1] [,2]
## [1,]    1    9
## [2,]    4   -2
```

```
B
```

```
##      [,1] [,2]
## [1,]    2  -10
## [2,]    0    1
```

```
A+B
```

```
##      [,1] [,2]
## [1,]    3   -1
## [2,]    4   -1
```

2.2.2 Matrix multiplication

As we implied above, multiplication means something slightly different than what you may be used to in the world of matrices and vectors. Let A be an $N \times K$ matrix and B a $K \times M$ matrix. The product of A and B is denoted AB with elements

$$[AB]_{ij} = \sum_{k=1}^{K} [A]_{ik}[B]_{kj}, \quad i = 1, \ldots N; , j = 1, \ldots, M.$$

Note that AB is an $N \times M$ matrix and that matrix multiplication is only defined when the number of columns in A matches the number of rows in B. Further from the definition we see that the ijth element of AB is the dot product (see below) of the ith row of A with the jth column of B. In other words the "inside" dimensions have to align, while the product's dimensions match the "outside" dimensions.

We can consider some properties of matrix multiplication. Let A, B, and C be matrices with appropriate dimensions and a be a scalar.

Zero is zero $A0 = 0$

One is one $AI = A$

Associative property $(AB)C = A(BC)$

Distributive property 1 $(A + B)C = AC + BC$

Distributive property 2 $A(aB) = aAB$

Transpose property $(AB)' = B'A'$ and generally $\left(\prod_{i=1}^{N} A_i\right)' = \prod_{i=1}^{N} A'_{N-i+1}$

Important: matrix multipication is associative, but *not* communtative! That is property 3 is true, but AB does not have to (and generally won't) equal BA. As such we can move parentheses around without too much trouble, but be careful with the ordering of matrices.

For example,

$$
\begin{aligned}
A &= \begin{bmatrix} 1 & -4 \\ 2 & 0 \end{bmatrix} \\
B &= \begin{bmatrix} -1 \\ 0.25 \end{bmatrix} \\
AB &= \begin{bmatrix} 1 \times -1 + -4 \times 0.25 \\ 2 \times -1 + 0 * 0.25 \end{bmatrix} \\
&= \begin{bmatrix} -2 \\ -2 \end{bmatrix}
\end{aligned}
$$

```r
A <- matrix(c(1,2,-4,0), nrow=2)
B <- matrix(c(-1, 0.25), nrow=2)
A %*% B
```

```
##      [,1]
## [1,]   -2
## [2,]   -2
```

Unless otherwise stated we will assume that all vectors are column vectors ($N \times 1$ matrix). So the vector v is

$$
v = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix}
$$

and its transpose is a row vector. We can multiply a row vector by a column vector of the same size to find

$$
x'y = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \sum_{i=1}^{N} x_i y_i.
$$

This operation is sometimes called the inner product or dot product of two vectors and is frequently denoted as $x \cdot y$.

The dot product is an interesting operator that is frequently useful. One important property here is that $x'y = y'x$, which can be helpful for finding new properties and simplifications. The dot product can also be used to define distance where $\sqrt{(x-y)'(x-y} = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$

is the commonly used Euclidean distance between x and y (as in the Pythagorian theorem when $N = 2$).

Note, we now have enough new tools to represent the multiple regression problem in matrix notation. Let $\beta = (\beta_0, \beta_1, \ldots, \beta_m)$ be the length $k = m + 1$ vector of regression coefficients, and let $x_i = (1, x_{i1}, x_{i2}, \ldots, x_{im})$ be a length k vector representing m independent variables. We can write the regression objective function as

$$\sum_{i=1}^{N} (y_i - \beta' x_i)^2 = \sum_{i=1}^{N} \left(y_i - \beta_0 - \beta_1 x_{i1} - \ldots - \beta_m x_{im}\right)^2 .$$

Note that we have a leading 1 in x_i to represent the constant term. We can move this from vector into matrix notation with relative ease now.
Define

$$X = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1m} \\ 1 & x_{21} & \ldots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & \ldots & x_{Nm} \end{bmatrix},$$

and now the OLS objective function is

$$(y - X\beta)'(y - X\beta).$$

This is a start, but we need some more technology before we can do anything with this representation.

2.3 Linear equations

Consider a system of N linear equations:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1M}x_M &= b_1 \\ &\vdots \\ a_{N2}x_1 + a_{N2}x_2 + \ldots + a_{NM}x_M &= b_N. \end{aligned}$$

Note that given the above, we can rewrite this system in matrix notation

$$Ax = b,$$

where A is an $N \times M$ matrix, x is length M vector, and b is a length N vector. In a system of linear equations we tend to think of A and b as known and x has a value we need to solve for.

For example, consider a system of two equations and two unknowns

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2. \end{aligned}$$

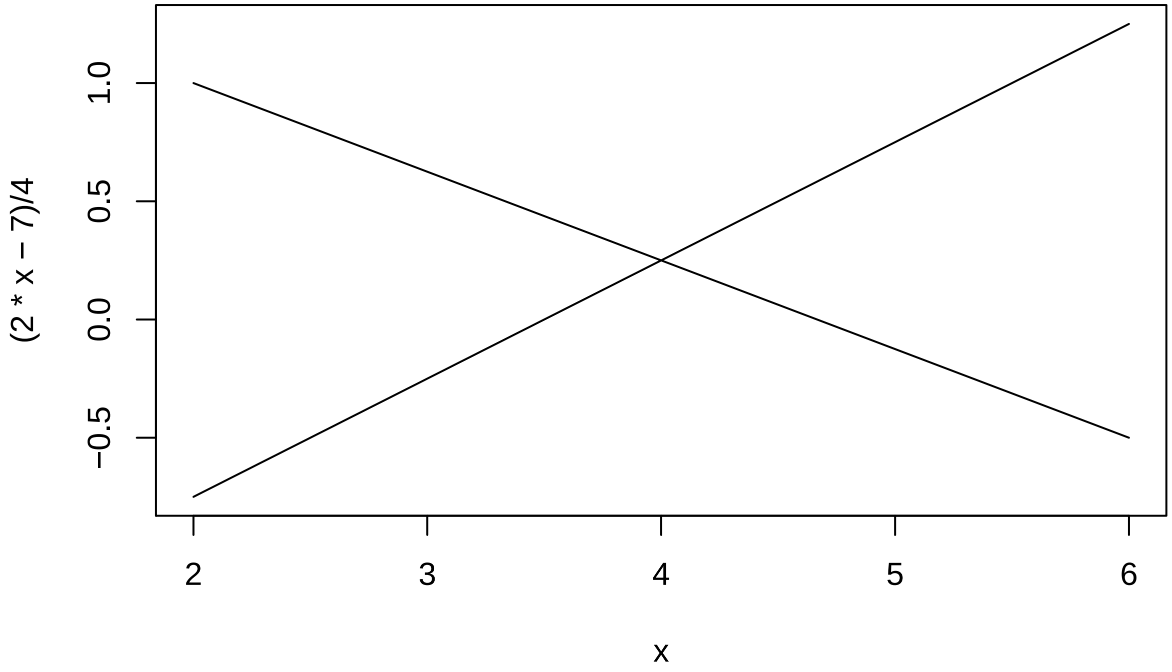
For fixed values of A and b we can draw this system as a pair of lines. If the lines intersect at exactly one point then we have a unique solution for x. If the lines are parallel, there are no solutions. Finally, if the lines are identical, there are an infinite number of solutions.

As a more concrete example consider:

$$\begin{aligned} 2x_1 - 4x_2 &= 7 \\ 3x_1 + 8x_2 &= 14. \end{aligned}$$

We can solve these by first noting that $x_2 = \frac{2x_1 - 7}{4}$ and plugging this into equation 2 we find that $x_1 = 4$. Plugging this into equation 1 we then find $x_2 = 1/4$. Here there is a unique solution.

```r
curve((2*x-7)/4, from=2, to=6)
curve((14-3*x)/8, from=2, to=6, add=TRUE)
```

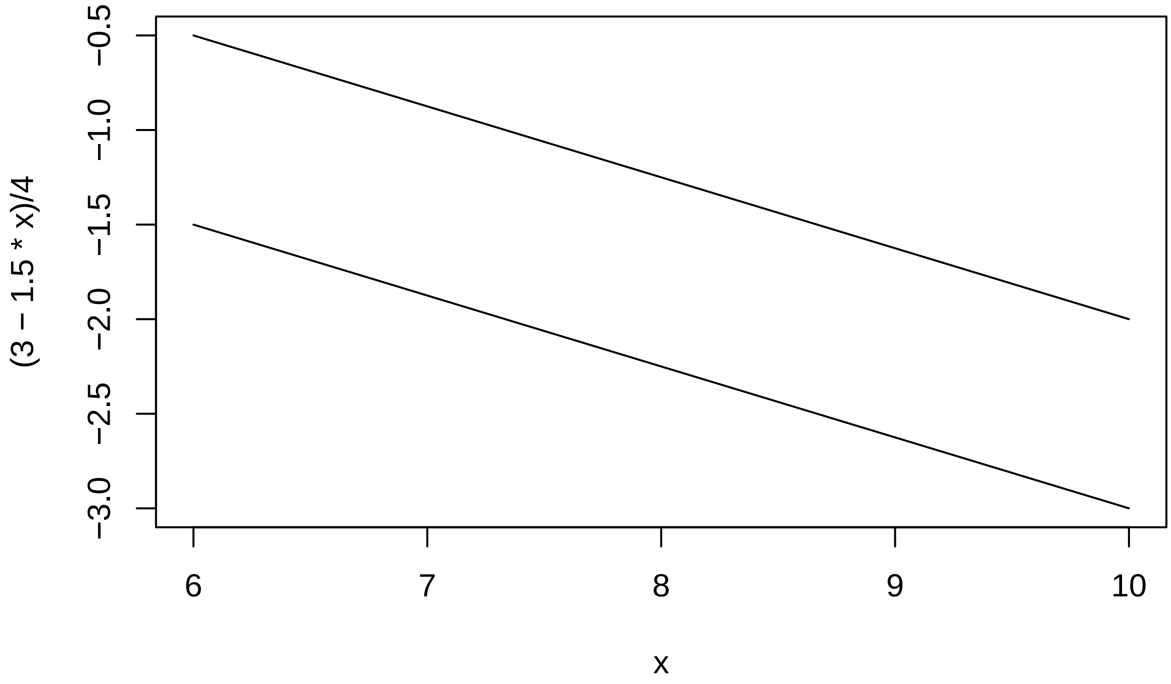


Now consider:

$$
\begin{aligned}
1.5x_1 + 4x_2 &= 3 \\
3x_1 + 8x_2 &= 14
\end{aligned}
$$

As before we start with $x_2 = \frac{3-1.5x_1}{4}$. Plugging this into equation 2, however, produces the nonsense result $6 = 14$. There is no solution to these equations, as the lines are parallel.

```
curve((3- 1.5*x)/4, from=6, to=10, ylim=c(-3,-1/2))
curve((14-3*x)/8, from=6, to=10, add=TRUE)
```

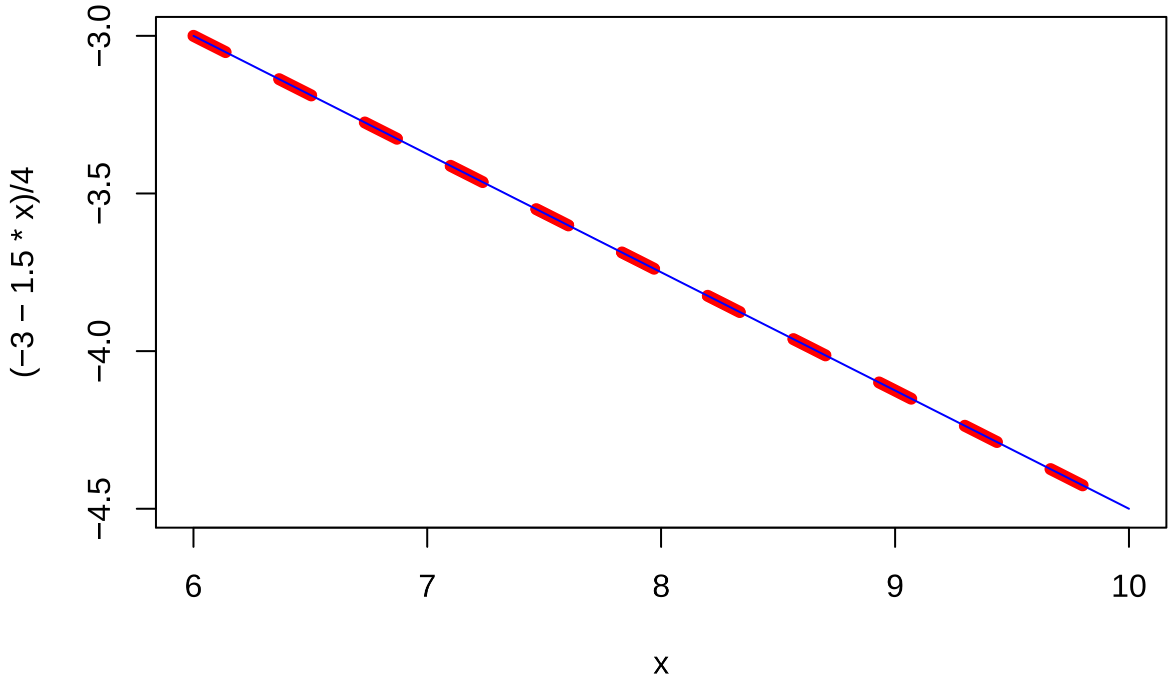


Now consider:

$$
\begin{aligned}
1.5x_1 + 4x_2 &= -3 \\
3x_1 + 8x_2 &= -6
\end{aligned}
$$

As before we start with $x_2 = \frac{-3-1.5x_1}{4}$. Plugging this into equation 2, however, produces the "too much sens" result $-6 = -6$. There are an infinite number of solution to these equations: they're the same! The points (-2,0),(-4,3/4), (0,-3/4) are all valid solutions.

```
curve((-3- 1.5*x)/4, from=6, to=10, lwd=6, lty="dashed", col="red")
curve((-6-3*x)/8, from=6, to=10, add=TRUE, col="blue")
```



But let's go back to a system with a unique solution:

$$\begin{aligned} 2x_1 - 4x_2 &= 7 \\ 3x_1 + 8x_2 &= 14. \end{aligned}$$

We can solved this algebraically, but note that we can also write it in matrix form:

$$\begin{bmatrix} 2 & -4 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 14 \end{bmatrix}.$$

Can we find solutions using matrix algebra?

2.4 Matrix inversion

Yes! Define the inverse of a square matrix A as the matrix A^{-1} such that $AA^{-1} = A^{-1}A = I$. This is as close as we get to matrix division. A few facts about inversion:

1. Only square matrices can be inverted
2. If an inverse exists, it is unique
3. If an inverse doesn't exist, then the matrix is said be **singular** or non-invertable
4. The inverse of a 1×1 matrix is $A^{-1} = [a_{11}]^{-1} = [1/a_{11}]$
5. The inverse of a size N diagonal matrix A is diagonal with non-zero elements $(1/a_{ii})_{i=1}^{N}$

6. The inverse of a nonsingular 2×2 matrix A is

$$A^{-1} = \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}.$$

7. If A is invertable than A' is invertable and $(A')^{-1} = (A^{-1})'$

Further, suppose that A and B are both invertable matrices, then

10. AB is invertable
11. $(AB)^{-1} = B^{-1}A^{-1}$

Now let's return to our system of linear equations: $Ax = b$. If A is non-singular, then $x = A^{-1}b$ provides the unique solution to these equations. If the number of equation N is less than the number of unknowns M then there will be no solutions, while if there are more unknowns than equations $M > N$ there will be an infinite number of solutions. The exception to this is if the equations are redundant. In which case A will have a **rank** less than N. The rank of a matrix is a measure of how many linear independent rows it has. Note that in the above example we produced a singular matrix by making one row just twice the other (clearly not linear independent). When the rank of A is equal to the number of rows in A we say that A is full rank.

To measure linear independence consider a row a_i in A. If a_i is a linear combination of the other rows a_{-i} then we can write $a_i = \sum_{j \neq i}^{N} c_j a_j$, where c's are constants. The matrix A is full rank if we can't find a set of constants for any row to make it a linear function of the other rows. Consider the matrix $A = \begin{bmatrix} 1.5 & 4 \\ 3 & 8 \end{bmatrix}$. Note that the second row is just double the first, or to put it another way $a_2 = 2a_1$, so A has rank 1. If we try and invert this matrix we find that the denominator is $1.5 \times 8 - 4 \times 3 = 0$, this matrix is singular.

Going back to a system with a unique solution:

$$2x_1 - 4x_2 = 7$$
$$3x_1 + 8x_2 = 14.$$

The matrix form of this problem will have full rank, and we can solve for X

$$\begin{aligned}
A &= \begin{bmatrix} 1.5 & 4 \\ 3 & 8 \end{bmatrix} \\
b &= \begin{bmatrix} 7 \\ 14 \end{bmatrix} \\
x &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
Ax &= b \\
A^{-1}Ax &= A^{-1}b \\
x &= A^{-1}b
\end{aligned}$$

```r
A <- matrix(c(2,3,-4,8), nrow=2)
A
```

```
##      [,1] [,2]
## [1,]    2   -4
## [2,]    3    8
```

```r
qr(A)$rank #check the rank
```

```
## [1] 2
```

```r
b <- c(7,14)
solve(A) %*% b
```

```
##      [,1]
## [1,] 4.00
## [2,] 0.25
```

```r
## singular case
A <- matrix(c(1.5,3,4,8), nrow=2)
A
```

```
##      [,1] [,2]
## [1,]  1.5    4
## [2,]  3.0    8
```

```r
qr(A)$rank #check the rank
```

```
## [1] 1
```

```r
try(solve(A) %*% b)
```

```
## Error in solve.default(A) :
##   Lapack routine dgesv: system is exactly singular: U[2,2] = 0
```

2.5 Determinants

The determinant of a matrix A is denoted $\det(A)$ or sometimes $|A|$. For a 1×1 matrix $A = [a]$ the determinant is $\det(A) = a$. For a 2×2 matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

the determinant $\det(A) = a_{11}a_{22} - a_{12}a_{21}$, look familiar? The determinant of a matrix is 0 if and only if the matrix is singular.

Other properties include.

1. $\det(AB) = \det(A)\det(B)$
2. $\det(I) = 1$
3. $\det(A^{-1}) = \det(A)^{-1}$

2.6 Eigenvalues

A complex, non-zero vector v is an **eigenvector** of a square matrix A if there exists a (possibly) complex constant λ such that $Av = \lambda v$, where λ is called an **eigenvalue**. For an $N \times N$ square matrix there will be N eigenvectors (each of length N) and N eigenvalues. These values may not be unique. When A is symmetric the eigenvalues will be real numbers, otherwise they may be complex.

Lets rewrite the identity $Av = \lambda v$ to be $(A - \lambda I)v = 0$. Recall that v is non-zero, this implies that the matrix $A - \lambda I$ must be singular (non-invertable). To see this suppose that $A - \lambda I$ is invertable and we will derive a contradiction. Start by front multiplying by the inverse and get

$$\begin{aligned}(A-\lambda I)^{-1}(A-\lambda I)v &= (A-\lambda I)^{-1}0\\ Iv &= 0\\ v &= 0,\end{aligned}$$

which contradicts v being non-zero. Because we have found a contradiction, it must be that $A - \lambda I$ is singular and as such $\det(A - \lambda I) = 0$. We can exploit this relationship to find all the eigenvalues by solving $\det(A - \lambda I) = 0$ for λ. In the 2×2 case this gives us

$$\begin{aligned}\det(A-\lambda I) = (a_{11}-\lambda)(a_{22}-\lambda) - a_{12}a_{21} &= 0\\ = \lambda^2 - \lambda(a_{11}+a_{22}) + a_{11}a_{22} - a_{12}a_{21} &= 0,\end{aligned}$$

which is a quadratic equation wrt to λ meaning there will be two solutions, one or more of which may be complex.

For example let $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$. Using the above, we will see that eigenvalues solve $\lambda^2 - 4\lambda + 3 = 0$. The quadratic formula tells us that $\lambda = 1$ or $\lambda = 3$. These are the eigenvalues of A. We can find the vector associated with the first eigenvalue $\lambda(1) = 1$ by solving the equations

$$\underbrace{\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} v_{1[1]} \\ v_{1[2]} \end{bmatrix}}_{\text{First eigenvector } v_1} = \underbrace{1}_{\text{First eigenvalue } \lambda_1} \underbrace{\begin{bmatrix} v_{1[1]} \\ v_{1[2]} \end{bmatrix}}_{\text{First eigenvector } v_1}.$$

Solving these we end up with two identical equations,

$$\begin{aligned}v_{1[1]} &= -v_{1[2]}\\ v_{1[1]} &= -v_{1[2]}\end{aligned}$$

As such we can say that the first eigenvector is

$$v_1 = c_1 \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

where c_1 is a constant. Eigenvectors are only identified up to a constant multiplier as you might suspect by the original identity $Av = \lambda v$. Multiplying/dividing by a constant does not change this result. Repeating this exercise of $\lambda_2 = 3$ we get

$$v_2 = c_2 \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

```
A = matrix(c(2,1,1,2), 2)
eigen(A)
```

```
## eigen() decomposition
## $values
## [1] 3 1
##
## $vectors
##           [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

```
# R's eigenvector here set the constants to 1/sqrt(2) more on this below
eigen(A)$vectors*sqrt(2)
```

```
##      [,1] [,2]
## [1,]    1   -1
## [2,]    1    1
```

Note that we can use eigenvalues to calculate determinants: $\det(A) = \prod_{i=1}^{N} \lambda_i$ where λ_i is the ith eigenvalue of A. A singular matrix will always have at least 1 zero eigenvalue and the rank of matrix equal to the number of non-zero eigenvalues. Finally, if A is invertable with ith eigenvalue λ_i then A^{-1} an ith eigenvalue equal to $1/\lambda_i$

```
det(A)
```

```
## [1] 3
```

```
prod(eigen(A)$values)
```

```
## [1] 3
```

```
eigen(solve(A))$values
```

```
## [1] 1.0000000 0.3333333
```

As another example consider the singular matrix $A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$. This matrix is singular because the second row is a multiple of the first. We can find the eigenvalues solve $\lambda^2 - 3\lambda = 0$, which has solutions as 0 and 3.

Another useful property of eigenvalues is that they can be used to decompose some matrices.

A square matrix A is called **diagonalizable** if it has an **eigenvalue decomposition** $A = V\Lambda V^{-1}$, where V is amatrix where the ith column is the ith eigenvector of A and $\Lambda = \lambda I$ is a diagonal matrix of associated eigenvalues. The matrix A will be diagonalizable if it does not have any repeated eigenvalues. All real symmetric matrices are diagonalizable.
Indeed when A is symmetric we can choose constants c such that such that we have $V^{-1} = V'$.

Let's go back to $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$. Here $V = \begin{bmatrix} c_1 & c_2 \\ -c_1 & c_2 \end{bmatrix}$ and $\Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$. We can quickly find $V^{-1} = \frac{1}{2c_1c_2} \begin{bmatrix} c_2 & -c_2 \\ c_1 & c_1 \end{bmatrix}$. Now we verify that

$$V\Lambda V^{-1} = \frac{1}{2c_1c_2} \begin{bmatrix} c_1 & c_2 \\ -c_1 & c_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} c_2 & -c_2 \\ c_1 & c_1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Note that choosing $c_1 = c_2 = 1/\sqrt{2}$ gives us $V' = V^{-1}$.

A symmetric matrix M is said to be **idempotent** if $MM = M$. Idempotent matrices tend to have interesting properties.
Theorem 10 *If M is symmetric and idempotent then its eigenvalues are all 0 or 1.*

Proof. Because M is symmetric we can use eigenvalue decomposition to rewrite $M = C\Lambda C'$ where $CC' = C'C = I$. As such we have

$$C\Lambda C' = M = MM = C\Lambda C'C\Lambda C' = C\Lambda\Lambda C'.$$

This implies that $\Lambda = \Lambda\Lambda$. Since Λ is diagonal, this only works for $\lambda_i \in \{0, 1\}$. □

More importantly for us later though is the following:
Theorem 11 *Let M be a symmetric and idempotent matrix of size N and let x be a vector of iid standard normal random variables. Then $x'Mx \sim \chi^2_p$ where $p = rank(M)$*

Proof. As before, we note that M is an $N \times N$ symmetric and has a eigenvalue decomposition to write $M = C\Lambda C'$ where $CC' = C'C = I$. This gives us $x'Mx = (C'x)'\Lambda C'x$. We know that $C'x$ is normally distributed because it is a linear combination normal random variables. Further we know that $\text{E}[C'x]$ is 0 and it has variance $C'IC = I$. Thus we have that $y \sim N(0, I)$ and $x'Mx$ simplifies into $\sum_{i=1}^N \lambda_i y_i^2$. Since M is idempotent $\lambda_i \in \{0, 1\}$ for all i and, as mentioned above, the number of non zero eigenvalues is p. This means that λ is composed of p 1s and $N - p$ zeros. Since y is the sum of squared standard normals it will be distributed χ^2_p. □

Finally, define the **trace** of a square matrix as the sum of its diagonal elements $\text{tr}(A) = \sum_{i=1}^{N} [A]_{ii}$. One fun fact is that $\text{tr}(A) = \sum_i \lambda_i$ or that the trace is the sum of the eigenvalues. Some additional properties of the trace include:

1. $\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B)$
2. $\text{tr}(cA) = c\text{tr}(A)$
3. $\text{tr}(AB) = \text{tr}(BA)$

2.7 Quadratic representation

A symmetric matrix A is positive (semi) definite if for all $x \neq 0$, $x'Ax > 0$ $(x'Ax \geq 0)$. In regression we will often be dealing with positive (semi) definite matrices. We can generalize the idea of a "square root" that applies to positive scalars to matrices using the **Cholesky decomposition**.[1] Specifically, we say that a semi-positive definite matrix A has a Cholesky Decomposition formed by a lower-triangular matrix L such that $A = LL'$. Note that if A admits this decomposition and we let $y = L'x$ then we can verify that $x'Ax = y'y \geq 0$, as $y'y$ is an inner product. Cholesky decompositions are often written as $\text{chol}(A)$ or $A^{1/2}$. If A is positive definite, this decomposition will be unique (no guarantees for positive semi-definite).

Further, a symmetric positive (semi) definite matrix will have all (weakly) positive eigenvalues and determinant. Additionally, if A is a symmetric, positive definite, and invertable matrix, then its inverse is symmetric and positive definite as well. To see this consider $A = LL'$, which means that $A^{-1} = (L')^{-1}L^{-1}$ and consider any vector $x \neq 0$ and $y = L^{-1}x$. This gives us $x'(L')^{-1}L^{-1}x = (L^{-1}x)'L^{-1}x = y'y$, which is what we had to show.

```
A
```

```
##      [,1] [,2]
## [1,]    2    1
## [2,]    1    2
```

```
chol(A) #note R returns it as an Upper triangle U=L'
```

```
##          [,1]      [,2]
## [1,] 1.414214 0.7071068
## [2,] 0.000000 1.2247449
```

[1] This is not the only way or even the most common way to think about matrix square roots, but it will be good for us.

```
t(chol(A)) %*% chol(A)
```

```
##      [,1] [,2]
## [1,]    2    1
## [2,]    1    2
```

```
chol(solve(A)) #not an error
```

```
##           [,1]       [,2]
## [1,] 0.8164966 -0.4082483
## [2,] 0.0000000  0.7071068
```

2.8 Vector calculus and optimization of multiple variables

We now consider some rules for calculus on vectors. Throughout this section we will focus on a scalar z, a vector of K variables x, and a vector of L variables y.

Let $z = f(x)$ be a function that inputs x and returns a single value X. You can imagine the regression objective function that inputs guesses of β and returns the sum of squared errors. The derivative $D_x f(x)$ is called the gradient (sometimes denoted $\nabla f(x)$). The gradient is the vector of first derivatives of f with respect to each element of x, such that

$$D_x f(x) = \begin{bmatrix} D_{x_1} f(x) \\ \vdots \\ D_{x_K} f(x) \end{bmatrix}.$$

The Jacobian is a special case of first derivatives for when we have a mapping that takes a vector as an input and returns a vector as an output. For example let $y = g(x)$ take the K length input X and return the L length output y. The Jacaboian becomes

$$J(x) = D_x g(x) = \begin{bmatrix} D_{x1} g(x)_1 & D_{x2} g(x)_1 & \dots & D_{x_K} g(x)_1 \\ D_{x1} g(x)_2 & D_{x2} g(x)_2 & \dots & D_{x_K} g(x)_2 \\ \vdots & \vdots & \vdots & \vdots \\ D_{x1} g(x)_L & D_{x2} g(x)_L & \dots & D_{x_K} g(x)_L \end{bmatrix}$$

Where the notation $D_{x_k} g(X)_\ell$ says to take the partial derivative of the ℓth output from $g(X)$ with respect to the kth input X_k. In essence, the Jacobian is a matrix where each row is the gradient for the ℓth output of the function (or each column is the derivative with respect to a specific input).

All the basic calculus rules like the product rule and the "chain" rule still hold (i.e., $D(f(g(x))) = Df(g(x))Dg(x)$ regardless of if x is a single variable or a vector and regardless of if f or g return single values or vectors). Some additional properties (supposing that everything here exists)

1. $D_x f(x') = D_x f(x)'$
2. $D_x(Ax + b) = A$
3. $J(g(X))^{-1} = J(g^{-1}(y))$ for $y = g(X)$ (Inverse function theorem)
4. $D_x(x'x) = 2x'$
5. $D_x(x'Ax) = x'(A + A')$

The last two are easy to show using the product rule:

$$D_x f(x)'g(x) = f(x)'D_x g(x) + g(x)'D_x f(x).$$

2.8.1 Optimization

Now that we have some calculus we want to use it for optimization. Let $f(x)$ be a function that returns a single value from a vector x that is twice continuously differentiable. We write the gradient, as we did above

$$D_x f(x) = \begin{bmatrix} D_{x_1} f(x) \\ \vdots \\ D_{x_K} f(x) \end{bmatrix},$$

and the second derivative is called the Hessian (we saw this a little bit at the end of math camp).

$$H(x) = D^2_{xx'} f(x) = \begin{bmatrix} D^2_{x_1 x_1} f(x) & D^2_{x_2 x_1} f(x) & \dots & D^2_{x_K x_1} f(x) \\ D^2_{x_1 x_2} f(x) & D^2_{x_2 x_2} f(x) & \dots & D^2_{x_K x_2} f(x) \\ \vdots & \vdots & \ddots & \vdots \\ D^2_{x_1 x_K} f(x) & D^2_{x_2 x_K} f(x) & \dots & D^2_{x_K x_K} f(x) \end{bmatrix}.$$

The multivariable extension of the necessary first order condition (FOC) for $x*$ to be a minimum or maximum is that the gradient $D_x f(x^*) = 0$. If x^* meets the FOC then we consider the second order sufficient condition. For multiple variables we say that x^* is a local minimum (maximum) if $H(x^*)$ is symmetric positive (negative) definite. These locals become global if f is globally convex (concave). We can check $H(x^*)$ using a few of our tools. Notably, we can check the eigenvalues: $H(x*)$ is symmetric positive (negative) definite if its eigenvalues are all strictly positive (negative).

2.9 Probability theory with vectors

We briefly alluded to probability theory with multiple variables when we described joint distributions and relations among variables. Notably, we described the covariance between variables, but now we can kick it up just a little bit. Let $X = (X_1, \ldots, X_K)$ be a vector of random variables we can define the expectation and variance of X as

$$\mathrm{E}[X] = (\mathrm{E}[X_1], \ldots, \mathrm{E}[X_K]),$$

and

$$\mathrm{Var}(X) = \mathrm{E}[(X-\mu)(X-\mu)'] = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \ldots & \mathrm{Cov}(X_1, X_K) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \ldots & \mathrm{Cov}(X_2, X_K) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_K, X_1) & \mathrm{Cov}(X_K, X_2) & \ldots & \mathrm{Var}(X_K) \end{bmatrix}$$

The **variance matrix** or **variance-covariance matrix** is a symmetric, positive semi definite matrix. To see this note that for $y \neq 0$ we can show

$$\begin{aligned} y' \mathrm{Var}(X) y &= y' \mathrm{E}[(X - \mathrm{E}[X])(X - \mathrm{E}[X])'] y \\ &= \mathrm{E}[y'(X - \mathrm{E}[X])(X - \mathrm{E}[X])' y \end{aligned}$$

Let $Z = (X - \mathrm{E}[X])' y$

$$= \mathrm{E}[Z'Z]$$

Where $Z'Z$ will be positive semi definite by the same logic as LL' above. It will be positive definite if X has full rank.

Additional things to note with vector of random variables X, a matrix of constants A, and vector of constants b:

1. $\mathrm{E}[AX + b] = A\,\mathrm{E}[X] + B$
2. $\mathrm{Var}(AX + b) = A\,\mathrm{Var}(X) A'$

These are the generalizations from the univariable case.

Finally, for a multi-variable transformation $Y = g(X)$ we have

$$f_Y(y) = f_X(g^{-1}(y))/|\det(J(g(X)))|,$$

which looks similiar to what we saw before, but with the addition of the determinant, and the use of the Jacobian for the first derivative. Let's try an example.

Consider a multivariate version extension of the standard normal distribution as the joint distribution of K independent standard normal random variables:

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x_1^2}{2}} \frac{1}{\sqrt{2\pi}} e^{\frac{-x_2^2}{2}} \dots \frac{1}{\sqrt{2\pi}} e^{\frac{-x_K^2}{2}} = \frac{1}{\sqrt{(2\pi)^K}} e^{\frac{x'x}{2}}.$$

Here we can see that $\mathrm{E}[X] = (0, 0, \dots, 0)$ and $\mathrm{Var}(X) = I_K$. Imagine a matrix $\Omega^{1/2}$ such that $\Omega^{1/2}\left(\Omega^{1/2}\right)' = \Omega$, where Ω is positive definite and symmetric. Now let's consider the transformation $Y = \Omega^{1/2}X + \mu$. The expectation of Y is straight forward

$$\mathrm{E}[Y] = \Omega^{1/2}\,\mathrm{E}[X] + \mu = \mu.$$

The variance is also easy

$$\mathrm{Var}[Y] = \Omega^{1/2}\,\mathrm{Var}(X)(\Omega^{1/2})' = \Omega.$$

Now we want to find the pdf of Y. We can build the Jacobian easily enough

$$D_{x_k} g(X)_\ell = [\Omega^{1/2}]_{\ell,k}$$

making the full Jacobian

$$J = \Omega^{1/2}$$

Also we can find $X = g^{-1}(Y) = (\Omega^{1/2})^{-1}(Y - \mu)$. Now we can apply the change formula

$$\begin{aligned}
f_Y(y) &= f_X(x)/\det(\Omega^{1/2}) \\
&= \frac{1}{\sqrt{(2\pi)^K}} e^{\frac{-x'x}{2}} \left(\det(\Omega^{1/2})\right)^{-1} \\
&= \frac{1}{\sqrt{(2\pi)^K}} e^{\frac{-1}{2}((\Omega^{1/2})^{-1}(y-\mu))'(\Omega^{1/2})^{-1}(y-\mu)} \left(\det(\Omega^{1/2})\right)^{-1} \\
&= \frac{1}{\sqrt{(2\pi)^K}} e^{\frac{-1}{2}((y-\mu)'((\Omega^{1/2})^{-1})'(\Omega^{1/2})^{-1}(y-\mu)} \left(\det(\Omega^{1/2})\right)^{-1} \\
&= \frac{1}{\sqrt{(2\pi)^K}} e^{\frac{-1}{2}(y-\mu)'((\Omega^{1/2})(\Omega^{1/2})')^{-1}(y-\mu)} \left(\det(\Omega^{1/2})\right)^{-1} \\
&= \frac{1}{\sqrt{(2\pi)^K}} e^{\frac{-1}{2}(y-\mu)'\Omega^{-1}(y-\mu)} \left(\det(\Omega^{1/2})\right)^{-1}
\end{aligned}$$

From property 1 of determinants (above) we note that

$$\begin{aligned}
\det(\Omega) &= \det(\Omega^{1/2}\Omega^{1/2}) = \det(\Omega^{1/2})^2 \\
\sqrt{\det(\Omega)} &= \det(\Omega^{1/2})
\end{aligned}$$

Plugging that in

$$\begin{aligned}
&= \frac{1}{\sqrt{(2\pi)^K}} e^{\frac{-1}{2}(y-\mu)'\Omega^{-1}(y-\mu)} \left(\sqrt{\det(\Omega)}\right)^{-1} \\
&= \frac{1}{\sqrt{(2\pi)^K \det(\Omega)}} e^{\frac{-1}{2}(y-\mu)'\Omega^{-1}(y-\mu)}
\end{aligned}$$

This is the full multivariate normal distribution.

Multivariate distributions are not always the easiest to deal with and when we want to test hypotheses about the parameters of multiple normal random variables, it is often easier to combine them. Let $X \sim N(\mu, \Omega)$ with a symmetric and positive definite Ω. By the Cholesky decomposition we can write a transformed variables $\left(\Omega^{1/2}\right)^{-1}(X - \mu)$ This transformed variable has an expected value of 0 and variance

$$\begin{aligned}
\text{Var}\left(\left(\Omega^{1/2}\right)^{-1}(X-\mu)\right) &= \left(\Omega^{1/2}\right)^{-1}\Omega\left(\left(\Omega^{1/2}\right)^{-1}\right)' \\
&= \left(\Omega^{1/2}\right)^{-1}\Omega^{1/2}\left(\Omega^{1/2}\right)'\left(\left(\Omega^{1/2}\right)^{-1}\right)' \\
&= I
\end{aligned}$$

This tells us that $\left(\Omega^{1/2}\right)^{-1}(X-\mu) \sim N(0,I)$ and thus $(X-\mu)'\Omega^{-1}(X-\mu) \sim \chi^2_K$. This will help us out going forward.

3 Multiple regression in finite samples

With all of these tools in hand we are now ready to consider and analyze the multiple regression model. In multiple regression we are looking to predict/explain/describe values of an outcome variable y_i using $K-1$ independent variables. Specifically we will write our regression equation as

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots \beta_m x_{im} + \varepsilon_i.$$

Here, $\beta = (\beta_0, \ldots, \beta_m)$ are the $K = m+1$ true population values of interest and $x_i = (1, x_{i1}, \ldots x_{im})$ is a vector of length $K = m+1$ characteristics. We can rewrite the model in vector form

$$\underset{1\times 1}{y_i} = \underset{K\times 1}{\beta}{}' \underset{K\times 1}{x_i} + \underset{1\times 1}{\varepsilon_i},$$

where β and x_i are both $K \times 1$ column vectors. Or we can use matrix form

$$\underset{N\times 1}{y} = \underset{N\times K}{X} \underset{K\times 1}{\beta} + \underset{N\times 1}{\varepsilon}.$$

The OLS estimator is unchanged:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta' x_i)^2$$

or

$$= \underset{\beta}{\operatorname{argmin}} (y - X\beta)'(y - X\beta)$$

We can take the FOC of the vector representation

$$\begin{aligned} D_\beta \sum_{i=1}^{N} (y_i - \beta' x_i)^2 &= \sum_{i=1}^{N} -2(y_i - \beta' x_i) x_i \\ 0 &= \sum_{i=1}^{N} -2(y_i - \beta' x_i) x_i \end{aligned}$$

We can rewrite this to be

$$\begin{aligned}
&= \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta' x_i)x_i \\
&= \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta' x_i)x_i \\
&= \frac{1}{N}\sum_{i=1}^{N} y_i x_i - \frac{1}{N}\sum_{i=1}^{N}\beta' x_i x_i \\
\frac{1}{N}\sum_{i=1}^{N} x_i y_i &= \frac{1}{N}\sum_{i=1}^{N} x_i \beta' x_i \\
\frac{1}{N}\sum_{i=1}^{N} x_i y_i &= \frac{1}{N}\sum_{i=1}^{N} x_i x_i' \beta \\
\hat{\beta} &= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i y_i.
\end{aligned}$$

Or in matrix notation

$$\hat{\beta} = (X'X)^{-1}X'y.$$

We want to make sure that this value is a minimum of the objective function so we check the second order condition.

$$D^2_{\beta\beta'}\sum_{i=1}^{N}(y_i - \beta' x_i)^2 = \sum_{i=1}^{N} 2x_i x_i'.$$

We want to show that this is positive definite. First, note that for any nonzero vector z of length K we know that $z' x_i x_i' z = (z' x_i)^2 \geq 0$. As such it follows that $z'\left(2\sum_{i=1}^{N} x_i x_i'\right) z \geq 0$, so the Hessian is semi-positive definite and that its eigenvalues are weakly positive. Let's suppose that $\sum_{i=1}^{N} x_i x_i'$ has full rank (i.e., $X'X$ is invertable), then we know that none of the eigenvalues are zero (and are thus strictly positive), and this tells us that the Hessian is positive definite. Thus the OLS estimates are indeed a local minimum of the objective function. Since the objective function is quadratic, there is only one minimum and the OLS estimates are the global minimum.

3.1 Finite sample properties of the OLS estimator with (possibly) stochastic regressors

We are now going to assess the OLS estimator under a new set of assumptions.
Overall, these will look very similar to what we've seen before. The main difference is that

we will be allowing X to realizations from a random variable.

Assumption B1 *The data generating process (the population model) is linear in the parameters:*

$$y_i = \beta' x_i + \varepsilon_i.$$

Assumption B1 also allow us to rewrite $\hat{\beta}$ as

$$\begin{aligned}
\hat{\beta} &= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i y_i \\
&= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i(\beta' x_i + \varepsilon_i) \\
&= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i(x_i'\beta + \varepsilon_i) \\
&= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i x_i'\beta + x_i\varepsilon_i) \\
&= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \left(\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)\beta + \left(\frac{1}{N}\sum_{i=1}^{N} x_i\varepsilon_i\right)\right) \\
&= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)\beta + \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} x_i\varepsilon_i\right) \\
&= \beta + \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i\varepsilon_i.
\end{aligned}$$

or

$$\hat{\beta} = \beta + (X'X)^{-1}X'\varepsilon.$$

Let's add a little more structure to see what this does for us.

Assumption B2 *The error term is mean 0, conditional on x_i:* $\mathrm{E}[\varepsilon_i|x_i] = 0$

Here we note that this zero-conditional mean assumption requires that there be strict independence (no linear or non-linear relationship) between x_i and ε_i. This assumption also implies that $\mathrm{E}[\varepsilon_i] = 0$ (unconditional mean zero) and that $\mathrm{E}[x_i\varepsilon_i] = 0$ (if this value exists). This assumption is untestable, but it is the linchpin of all regression analysis. If it holds we say that x_i is exogenous, otherwise we have an endogeneity problem (much more on this to come).

Assumption B3 *The pairs (x_i, ε_i) are independent and identically distributed.*

These three assumptions are going to buy us a lot. Now we can describe the expected value of the OLS estimates Let's check it out

$$\begin{aligned}
\mathrm{E}[\hat{\beta}] &= \mathrm{E}_x[\mathrm{E}[\hat{\beta}|X]] \\
&= \mathrm{E}_x\left[\mathrm{E}\left[\beta + \left(\frac{1}{N}\sum_{i=1}^{n} x_i x_i'\right)^{-1}\left(\frac{1}{N}\sum_{i=1}^{n} x_i \varepsilon_i\right)|X\right]\right] \\
&= \mathrm{E}_x\left[\beta + \left(\frac{1}{N}\sum_{i=1}^{n} x_i x_i'\right)^{-1}\left(\frac{1}{N}\sum_{i=1}^{n} x_i\, \mathrm{E}\left[\varepsilon_i|X\right]\right)\right] \\
&= \mathrm{E}_x\left[\beta + \left(\frac{1}{N}\sum_{i=1}^{n} x_i x_i'\right)^{-1}\left(\frac{1}{N}\sum_{i=1}^{n} x_i\, \mathrm{E}\left[\varepsilon_i|x_i\right]\right)\right] \\
&= \mathrm{E}_x\left[\beta\right] \\
&= \beta.
\end{aligned}$$

By applying the law of iterated expectations we find that $\mathrm{E}[\hat{\beta}] = \beta$.

Property B1 *Our first result for multiple regression is that the under Assumptions B1-B3 the OLS estimates of β are unbiased if $(X'X)^{-1}$ exists.*

That last bit requires that $(X'X)^{-1}$ to exists. Before we go much further, we should put some structure on the problem to ensure that it will.

Assumption B4 $\mathrm{E}[x_i x_i'] < \infty$ *and* $\mathrm{E}[x_i \varepsilon_i] < \infty$

Assumptions B2 and B4 combine to tell us more about the relationship of $x_i\varepsilon_i$, specifically, we can now say that $\mathrm{E}[\varepsilon_i x_i] = \mathrm{Cov}(x_i, \varepsilon_i) = 0$.

Assumption B5 *The* $\mathrm{E}[x_i x_i']$ *is symmetric and positive definite.*

These assumption puts some conditions on the distributions that generate the data x_i and the error term ε_i, that allow us to say that $(X'X)^{-1}$ will likely exist. In our finite sample, we should note that $\mathrm{rank}(X'X) = \mathrm{rank}(XX') = \mathrm{rank}(X) = \mathrm{rank}(X')$. For $\sum_{i=1}^{N} x_i x_i'$ to have full rank is the same as saying that $X'X$ has rank K or that X has rank K. The easiest way to think about this is to say that no variable in X can be a linear combination of one or more other variables (including the column of 1s). While we are looking at finite sample properties it will be useful to add

Assumption B5.A *The matrix* $\sum_{i=1}^{N} x_i x_i'$ *has full rank.*

Assumptions B4 and B5 imply the OLS estimate there is a unique and finite solution to the minimization problem unless we're really unlucky in how the X's are realized. Assumption B5.A tells us that we weren't unlucky on that front.

Property B2 *Under Assumptions B1-B5.A, the OLS estimator exists for any N.*

The next two assumptions go together:
Assumption B6 *The variance of $x_i\varepsilon_i$ is finite* $\mathrm{Var}(x_i\varepsilon_i) < \infty$

Assumption B7 *The error terms are homoskedastic* $\mathrm{E}[\varepsilon_i^2|x_i] = \sigma_\varepsilon^2$

Let $\hat{\varepsilon}_i$ be the estimated residual $\hat{\varepsilon}_i = y_i - \hat{\beta}'x_i$ and let

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{N-K}\sum_{i=1}^N \hat{\varepsilon}_i^2$$

be our estimate of σ^2. One thing to note here is that:

$$\begin{aligned}\sum_{i=1}^N \hat{\varepsilon}_i^2 &= \hat{\varepsilon}'\hat{\varepsilon} \\ &= (y - X\hat{\beta})'(y - X\hat{\beta}) \\ &= (X\beta + \varepsilon - X\hat{\beta})'(X\beta + \varepsilon - X\hat{\beta}) \\ &= (\varepsilon - X(\hat{\beta} - \beta))'(\varepsilon - X(\hat{\beta} - \beta)) \\ &= (\varepsilon - X(X'X)^{-1}X'\varepsilon)'(\varepsilon - X(X'X)^{-1}X'\varepsilon) \\ &= (M\varepsilon)'(M\varepsilon) \\ &= \varepsilon'M'M\varepsilon,\end{aligned}$$

where $M = I - X(X'X)^{-1}X'$. It turns out that M is an interesting and recurring matrix; M is called the **residual maker** as $My = \hat{\varepsilon}$ or the **annihilator matrix** (because it annihilates all the X out of y). A few facts about it: M is symmetric

$$M' = I - (X(X'X)^{-1}X')' = I - X(X'X)^{-1}X' = M,$$

and

$$\begin{aligned}MM &= (I - X(X'X)^{-1}X')(I - X(X'X)^{-1}X') \\ &= I - 2X(X'X)^{-1}X' + X(X'X)^{-1}X'X(X'X)^{-1}X' \\ &= I - 2X(X'X)^{-1}X' + XI(X'X)^{-1}X' \\ &= I - X(X'X)^{-1}X' \\ &= M\end{aligned}$$

M is idempotent.

Because M is idempotent we have $\hat{\varepsilon}'\hat{\varepsilon} = \varepsilon' M M \varepsilon = \varepsilon' M \varepsilon$. This allows us to have some fun

$$\begin{aligned}
\mathrm{E}[\hat{\varepsilon}'\hat{\varepsilon}|X] &= \mathrm{E}[\underset{1\times 1}{\varepsilon' M \varepsilon}|X] \\
&= \mathrm{E}[\mathrm{tr}(\varepsilon' M \varepsilon)|X] && \text{Let } A = \varepsilon',\ B = M\varepsilon. \text{ and recall } \mathrm{tr}(AB) = \mathrm{tr}(BA) \\
&= \mathrm{E}[\mathrm{tr}(M\varepsilon\varepsilon')|X] \\
&= \mathrm{tr}(\mathrm{E}[M\varepsilon\varepsilon'|X]) \\
&= \mathrm{tr}(M\,\mathrm{E}[\varepsilon\varepsilon'|X]) \\
&= \mathrm{tr}(M\sigma^2 I) \\
&= \sigma^2 \mathrm{tr}(M),
\end{aligned}$$

where

$$\begin{aligned}
\mathrm{tr}(M) &= \mathrm{tr}(I_N) - \mathrm{tr}(X(X'X)^{-1}X') \\
&= \mathrm{tr}(I_N) - \mathrm{tr}(X'X(X'X)^{-1}) \\
&= \mathrm{tr}(I_N) - \mathrm{tr}(I_k) \\
&= N - K,
\end{aligned}$$

as such

$$\begin{aligned}
\mathrm{E}[\hat{\varepsilon}'\hat{\varepsilon}] &= \mathrm{E}_x[\mathrm{E}[\hat{\varepsilon}'\hat{\varepsilon}|X]] \\
&= \mathrm{E}_x[\sigma^2(N-K)] = \sigma^2(N-K)
\end{aligned}$$

Now, since we set

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{N-K}\sum_{i=1}^{N} \hat{\varepsilon}_i^2$$

we now have

$$\mathrm{E}[\hat{\sigma}_\varepsilon^2] = \frac{1}{N-K}\,\mathrm{E}[\hat{\varepsilon}'\hat{\varepsilon}] = \sigma_\varepsilon^2.$$

We now have another property of OLS.

Property B3 *Under Assumptions B1-B7, $\hat{\sigma}_\varepsilon^2$ is a unbiased estimate of σ_ε^2.*

Further, recall that

$$\hat{\beta} = \beta + (X'X)^{-1}X'\varepsilon.$$

We can now find the variance given X

$$\begin{aligned}
\mathrm{Var}(\hat{\beta}|X) &= \mathrm{Var}((X'X)^{-1}X'\varepsilon|X) \\
&= (X'X)^{1}X'\,\mathrm{Var}(\varepsilon|X)X(X'X)^{-1} \\
&= (X'X)^{1}X'\,\mathrm{E}\left[(\varepsilon - \mathrm{E}[\varepsilon|X])(\varepsilon - \mathrm{E}[\varepsilon|X]|X)'\right]X(X'X)^{-1} \\
&= (X'X)^{1}X'\,\mathrm{E}[\varepsilon\varepsilon'|X]X(X'X)^{-1} \\
&= (X'X)^{1}X'[\sigma_\varepsilon^2 I_N]X(X'X)^{-1} \\
&= \sigma_\varepsilon^2(X'X)^{-1}X'X(X'X)^{-1} \\
&= \sigma_\varepsilon^2(X'X)^{-1}.
\end{aligned}$$

Our next property of OLS describes the variance of the OLS estimates
Property B4 *Under Assumptions B1-B7* $\mathrm{Var}(\hat{\beta}|X) = \sigma_\varepsilon^2(X'X)^{-1}$.

One final assumption completes our baseline assumptions
Assumption B8 *ε_i is normally distributed for all i.*

Normal errors buy as a few things going forward. First though it gets us another property:
Property B5 *Under Assumptions B1-B8, $\hat{\beta}$ is normally distributed, conditional on X (due to linearity),*

$$\hat{\beta}|X \sim N(\beta, \sigma_\varepsilon^2(X'X)^{-1}).$$

From here we can use the assumptions to consider the distributions of our estimates when σ_ε^2 is unknown and has to be estimated, let's start with $\hat{\sigma}_\varepsilon^2$. Start with

$$\begin{aligned}
\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2} &= \frac{(N-K)\frac{1}{N-K}\sum_{i=1}^{N}\hat{\varepsilon}_i^2}{\sigma_\varepsilon^2} \\
&= \frac{\hat{\varepsilon}'\hat{\varepsilon}}{\sigma_\varepsilon^2} \\
&= \frac{\varepsilon' M\varepsilon}{\sigma_\varepsilon^2} \\
&= \frac{\varepsilon}{\sigma_\varepsilon}' M \frac{\varepsilon}{\sigma_\varepsilon},
\end{aligned}$$

where the eagle-eyed observer will note

$$\frac{\varepsilon}{\sigma_\varepsilon} \sim N(0,1).$$

Applying Theorem 11 from above we see that this transformation of $\hat{\sigma}_\varepsilon^2$ will follow a $\chi^2_{\mathrm{rank}(M)}$

distribution. What is the rank of M? Three facts will help us here, recall:

1. Rank of M is equal to the number of non-zero eigenvalue $\text{rank}(M) = \sum_{i=1}^{N} \mathbb{I}(\lambda_i \neq 0)$;
2. The $\text{tr}(M) = \sum_{i=1}^{N} \lambda_i$ where λ_i is the ith eigenvalue of M;
3. Because M is idemopotent, $\lambda_i \in \{0, 1\}$ for all i.

Putting these together we can note that for an idemopotent matrix $\text{rank}(M) = \text{tr}(M)$. As such we can say that our next property of OLS is:

Property B6 *Under Assumptions B1-B8,*

$$\left.\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2}\right| X \sim \chi^2_{N-K},$$

but this conditional distribution does not depend on X and as such the marginal distribution is

$$\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2} \sim \chi^2_{N-K}.$$

This property means that we can test null hypotheses that $\sigma_\varepsilon^2 = s$ by constructing the test statistic $\frac{(N-K)\hat{\sigma}_\varepsilon^2}{s}$.

We need to prove one additional result before proceeding.

Theorem 12 *Let $z \sim N(0, I)$ be a vector of standard normal random variables. Then Az and Bz are independent if $AB' = 0$.*

Proof. Note that $(Az, Bz) = (A, B)z$ is a linear function of normal random variables and thus Az and Bz are jointly normal. As such, in order to show independence we need their covariance to be 0.

$$\text{Cov}(Az, Bz) = \text{E}[Az(Bz)'] - \text{E}[Az]\,\text{E}[Bz]' = A\,\text{E}[zz']B' = AIB' = AB' = 0.$$

□

Now we are close to describing the distribution of $\hat{\beta}$ when we don't know σ_ε^2. Start with the fact that

$$\frac{\hat{\beta} - \beta}{\sigma_\varepsilon} = (X'X)^{-1}X'\frac{\varepsilon}{\sigma_\varepsilon},$$

and

$$\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2} = (M\frac{\varepsilon}{\sigma_\varepsilon})'M\frac{\varepsilon}{\sigma_\varepsilon}.$$

Now, let $A = (X'X)^{-1}X'$, let $B = M$, and note that $z = \frac{\varepsilon}{\sigma_\varepsilon} \sim N(0, I)$, then

$$AB' = (X'X)^{-1}X'(I - X(X'X)^{-1}X') = (X'X)^{-1}X' - (X'X)^{-1}X'X(X'X)^{-1}X' = 0.$$

Using Theorem 12 we can find the next property of OLS.
Property B7 *Under Assumptions B1-B8* $\frac{\hat{\beta}-\beta}{\sigma_\varepsilon}$ *and* $\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2}$ *are statistically independent.*

This property moves us closer to forming the distribution of $\hat{\beta}$ with unknown σ_ε^2. Recall from math camp that we said that if $U \sim N(0, 1)$ and $V \sim \chi_v^2$ and U and V are independent, then $U/\sqrt{V/v} \sim t_v$. Now note that $\frac{\hat{\beta}_k - \beta_k}{\sigma_\varepsilon\sqrt{[(X'X)^{-1}]_{kk}}} \sim N(0, 1)$, for $k = 1, \ldots, K$. Additionally, note that this is unconditional X. Why? Because the standard normal doesn't depend on X; it just doesn't enter! This is wicked good for us, because if we have conditional distributions that don't involve the conditioned value, then the conditional and the marginal are the same! We already showed that $\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2} \sim \chi_{N-K}^2$. So

$$\frac{\frac{\hat{\beta}_k - \beta_k}{\sigma_\varepsilon\sqrt{[(X'X)^{-1}]_{kk}}}}{\sqrt{\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2} \Big/ N-K}} = \frac{\hat{\beta}_k - \beta_k}{\hat{\sigma}_\varepsilon\sqrt{[(X'X)^{-1}]_{kk}}} \sim t_{N-K}.$$

We now have enough for hypothesis testing in finite samples:
Property B8 *Under Assumptions B1-B8,* $\frac{\hat{\beta}_k - \beta_k}{\hat{\sigma}_\varepsilon\sqrt{[(X'X)^{-1}]_{kk}}} \sim t_{N-K}$.

This result opens the door to hypothesis testing with respect to the true but unknown β parameters. Individual tests of the from

$$H_0 : \beta_k = b,$$

can now be considered against one or two sided alternatives. Under the null we have

$$\frac{\hat{\beta}_k - b}{\hat{\sigma}_\varepsilon\sqrt{[(X'X)^{-1}]_{kk}}} \sim t_{N-K}.$$

And importantly, these various tests are not conditional distributions. Commonly these tests are two sided tests of the hypothesis that $\beta_k = 0$ for each $k = 1, \ldots, K$. These are the tests that get reported by default by `lm` or any other basic regression function. While these tests are important and are the crux of most research questions, they are only the beginning of an entire world of hypothesis testing.

3.1.1 Linear hypothesis tests

Fancier hypothesis tests move us beyond asking whether or not $\beta_k = b$. Instead our research question may lead us to ask whether $\beta_k + \beta_\ell = b$ or if $\beta_k = \beta_\ell$ or if $\beta_k = 0$ & $\beta_\ell = 1$. Hypotheses like these can be formed by common research questions and we'll encounter some of these in practice going forward.

What's notable about hypotheses tests like these as that they can all be written as a set of linear equations

$$A\beta = b.$$

As an example let $K = 3$ and consider the hypothesis that $\beta_1 = \beta_2$. Here $A = [0, 1, -1]$ and $b = 0$, then

$$\begin{aligned} A\beta &= b \\ 0 \times \beta_0 + 1 \times \beta_1 - 1 \times \beta_2 &= 0 \\ \beta_1 - \beta_2 &= 0 \end{aligned}$$

Under a different null where $\beta_0 = 0$ and $\beta_1 = 1$, we have $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Let J be the number of rows in A and b, this number matches the number of restrictions (equal signs) in the null hypothesis.

To find a test statistic for these tests we will exploit the following fact from math camp

Fact 2 *Let $U \sim \chi^2_u$ and $V \sim \chi^2_v$ and U and V be statistically independent. Then $\frac{U/u}{V/v} \sim F(u, v)$.*

The F distribution here is found as the ratio of two scaled χ^2 random variables. It is a common distribution in linear models, it has two parameters u and v that both reflect degrees of freedom.

For our hypothesis tests we have a null

$$H_0 : A\beta = b$$

and alternative

$$H_A : A\beta \neq b.$$

Under the null hypothesis we write

$$A\hat{\beta} - b = A\hat{\beta} - A\beta = A(\hat{\beta} - \beta).$$

Since we know that $\hat{\beta}|X \sim N(\beta, \sigma_\varepsilon^2 (X'X)^{-1})$, we can say that the

$$A(\hat{\beta} - \beta)|X \sim N(0, \sigma_\varepsilon^2 A(X'X)^{-1}A'),$$

or

$$\left[\sigma_\varepsilon^2 A(X'X)^{-1}A'\right]^{-1/2} A(\hat{\beta} - \beta) \sim N(0, I_J),$$

where the notation $-1/2$ here means the inverse of the Cholesky decomposition. To make this easier to deal with we we will "square" it like we described earlier

$$(\hat{\beta} - \beta)'A' \left[\sigma_\varepsilon^2 A(X'X)^{-1}A'\right]^{-1} A(\hat{\beta} - \beta) \sim \chi_J^2.$$

This would be great... except for that σ_ε^2 in there. Note that since this expression is a function of $\hat{\beta}$ and $\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2}$ is a function of $\hat{\sigma}_\varepsilon^2$ then these two quantities are independent (see Property B7, above). We can apply Fact 2 to show that

$$\begin{aligned} F &= \frac{(\hat{\beta} - \beta)'A' \left[\sigma_\varepsilon^2 A(X'X)^{-1}A'\right]^{-1} A(\hat{\beta} - \beta) \Big/ J}{\frac{(N-K)\hat{\sigma}_\varepsilon^2}{\sigma_\varepsilon^2} \Big/ N-K} \\ &= \frac{(\hat{\beta} - \beta)'A' \left[A(X'X)^{-1}A'\right]^{-1} A(\hat{\beta} - \beta)}{J\hat{\sigma}_\varepsilon^2} \\ &= \frac{(A\hat{\beta} - A\beta)' \left[A(X'X)^{-1}A'\right]^{-1} (A\hat{\beta} - A\beta)}{J\hat{\sigma}_\varepsilon^2} \\ &\sim F_{J,N-K}. \end{aligned}$$

This leads to:

Property B9 *Under Assumptions B1-B8, null hypotheses of the form $A\beta = b$ can be tested with the F statistic*

$$\frac{(A\hat{\beta} - b)' \left[A(X'X)^{-1}A'\right]^{-1} (A\hat{\beta} - b)}{J\hat{\sigma}_\varepsilon^2} \sim F_{J,N-K},$$

which is purely a function of data, estimates, and hypothesized values.

3.1.2 Application

Before proceeding, let's consider how to interpret the parameters $\beta = (\beta_0, \beta_1, \beta_2)$. Things changa little bit from simple regression land, but not a lot. Let $x_{-1,i}$ be the x_i excluding the column of 1s. For β_0 we are still looking at

$$\mathrm{E}[y|x_{-1,i} = 0] = \mathrm{E}[\beta' x_i + \varepsilon | x_{-1,i} = 0] = \beta_0.$$

So $\hat{\beta}_0$ is the estimated expected value of y_i when all the independent variables are set to 0. How interesting that is depends on whether $X_{-1} = 0$ is an interesting (or plausible) profile. For the remaining βs we will be interested in the **marginal effect** given by the derivative of y_i with respect to each element x_i.

$$D_{x_{ij}} y_i = D_{x_{ij}} \left(\beta' x_i + \varepsilon_i\right) = \beta_j.$$

Put another way:

> On average, a 1 [UNIT OF X_j] increase in X_j is associated with a $\hat{\beta}_j$ [UNIT OF y] increase in y, holding all the other independent variables constant.

What does this last part mean? It means that we consider the change in y as a function of X_j, **after** removing any changes in y due to the other variables. This follows from a result known as the **Frisch-Waugh-Lovell** theorem. Put simply, in a model where we split the regressors into two vectors x_i and z_i and

$$y_i = \beta_1' x_i + \beta_2' z_i + \varepsilon_i,$$

you can estimate β_2 either by multiple regression or by fitting the regression of

$$M_X y = M_X Z \beta_2 + M_X \varepsilon.$$

Here $M_X = I - X(X'X)^{-1}X'$ is the annihilator matrix that removes all the variance in y explained by X.

Let's suppose our research questions are: Does candidate spending effect vote share and do challenger and incumbent spending have the same effect in campaigns? We could imagine a model

$$y_i = \beta_0 + n_i \beta_1 + c_i \beta_2 + \varepsilon_i$$

where y_i is the incumbent's share of the vote, n_i in incumbent spending, and c_i is challenger spending. Here $x_i = (1, n_i, c_i)$, and our hypothesis of interest are

$$
\begin{aligned}
H1_0 &: \beta_1 = 0 \\
H1_A &: \beta_1 \neq 0 \\
H2_0 &: \beta_2 = 0 \\
H2_A &: \beta_2 \neq 0 \\
H3_0 &: \beta_1 = -\beta_2 \\
H3_A &: \beta_2 \neq -\beta_2
\end{aligned}
$$

We have some data on senate races, let's see what happens

```
library(readstata13)
library(car) #Companion to Applied Regression

senate.data <- read.dta13("Rcode/datasets/senate_expanded.dta")
head(senate.data)
```

```
##   year st_abr    st_name st_code st_south   st_pop st_uemp inc_icpsr inc_pos
## 1 1980     AK     Alaska      81        0   401851     9.4     12105  -0.285
## 2 1980     AL    Alabama      41        1  3893888     8.4     14711  -0.226
## 3 1980     AR   Arkansas      42        1  2286435     7.5     14300  -0.313
## 4 1980     AZ    Arizona      61        0  2718215     6.7      3658   0.608
## 5 1980     CA California      71        0 23667902     6.9     12103  -0.409
## 6 1980     CO   Colorado      62        0  2889964     5.9     14305  -0.409
##   inc_spend ch_qual ch_spend ch_wealthy inc_2p_share st_id inc_tenure inc_rep
## 1        NA      NA       NA         NA           NA    NA         12       0
## 2        NA      NA       NA         NA           NA  26.2          2       0
## 3    220861       0   119196          0    0.5911147  11.5          6       0
## 4    949992       0  2085242          1    0.5054943  23.4         12       1
## 5   2823607       0  1152272          0    0.6033472   7.4         12       0
## 6   1142304       3  1085205          0    0.5082657  15.8          6       0
```

```
#Let's use spending per capita
senate.data$inc_spend_capita <- senate.data$inc_spend /senate.data$st_pop
senate.data$ch_spend_capita <- senate.data$ch_spend / senate.data$st_pop

model.sen <- lm(inc_2p_share~inc_spend_capita+ch_spend_capita, data=senate.data)
```

```r
summary(model.sen)
```

```
##
## Call:
## lm(formula = inc_2p_share ~ inc_spend_capita + ch_spend_capita,
##     data = senate.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21801 -0.05961 -0.00514  0.04432  0.37040
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.631851   0.007553  83.660  < 2e-16 ***
## inc_spend_capita  0.022629   0.006261   3.615 0.000354 ***
## ch_spend_capita  -0.084801   0.008681  -9.769  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08614 on 292 degrees of freedom
##   (77 observations deleted due to missingness)
## Multiple R-squared:  0.2709, Adjusted R-squared:  0.266
## F-statistic: 54.26 on 2 and 292 DF,  p-value: < 2.2e-16
```

```r
#time out interpret the coefficients
#time out what hypothesis tests are included here?

#Let's verify these
dat <- na.omit(subset(senate.data,
                      select=c("inc_2p_share",
                               "inc_spend_capita",
                               "ch_spend_capita")))
X <- with(dat, cbind(1,inc_spend_capita, ch_spend_capita))
y <- dat$inc_2p_share
b.hat <- solve(t(X) %*% X) %*% t(X) %*% y
cbind(b.hat, model.sen$coefficients)
```

```
##                         [,1]        [,2]
##                   0.63185127  0.63185127
## inc_spend_capita  0.02262911  0.02262911
## ch_spend_capita  -0.08480088 -0.08480088
```

```
# By FWL
N <- nrow(X)
K <- ncol(X)
Mch <- diag(N) - X[,3] %*% solve(t(X[,3]) %*% X[,3]) %*% t(X[,3])
ystar <- Mch %*% y #annihilate ch spending out of y
Xstar <- Mch %*% X[,1:2]#annihilate ch spending out of inc spending and 1
solve(t(Xstar) %*% Xstar) %*% t(Xstar) %*% ystar
```

```
##                        [,1]
##                  0.63185127
## inc_spend_capita 0.02262911
```

```
Minc<- diag(N) - X[,2] %*% solve(t(X[,2]) %*% X[,2]) %*% t(X[,2])
ystar <- Minc %*% y #annihilate inc spending out of y
Xstar <- Minc %*% X[,c(1,3)]#annihilate inc spending out of ch spending and 1
solve(t(Xstar) %*% Xstar) %*% t(Xstar) %*% ystar
```

```
##                        [,1]
##                  0.63185127
## ch_spend_capita -0.08480088
```

```
#Estimate the variance
sigma2.hat <- sum( (y-X%*%b.hat)^2) /(N-K)
c(sigma2.hat, summary(model.sen)$sigma^2)
```

```
## [1] 0.007419449 0.007419449
```

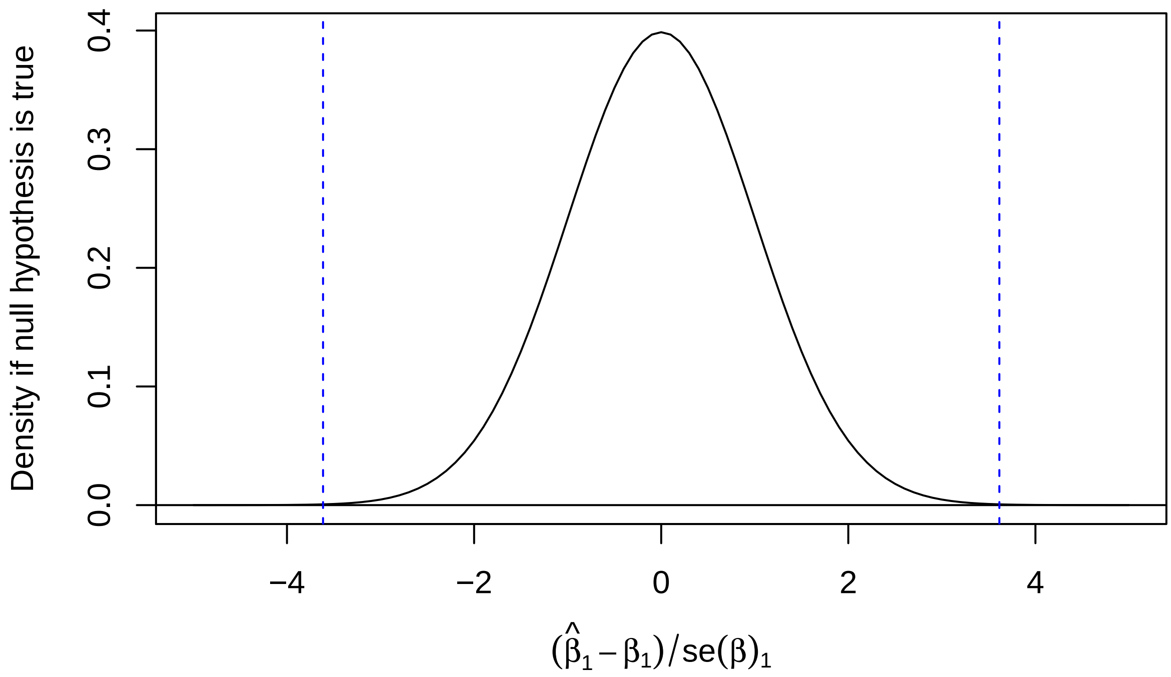
```
V <- sigma2.hat * solve(t(X) %*% X)
V
```

```
##                                 inc_spend_capita ch_spend_capita
##                   5.704168e-05    -2.581596e-05   -2.113666e-06
## inc_spend_capita -2.581596e-05     3.919555e-05   -3.538724e-05
## ch_spend_capita  -2.113666e-06    -3.538724e-05    7.535747e-05
```

```r
Var.bhat  <- diag(V)
se.bhat  <- sqrt(Var.bhat)
cbind(se.bhat, sqrt(diag(vcov(model.sen))))
```

```
##                         se.bhat            
##                     0.007552594 0.007552594
## inc_spend_capita 0.006260635 0.006260635
## ch_spend_capita  0.008680868 0.008680868
```

```r
#hypotheses 1:
t1 <- (b.hat[2]-0)/se.bhat[2]
curve(dt(x, df=(N-K)), from=-5, to=5,
      xlab=expression((hat(beta)[1]-beta[1])/se(beta)[1]),
      ylab="Density if null hypothesis is true")
abline(h=0)
abline(v=t1, col="blue", lty="dashed")
abline(v=-t1, col="blue", lty="dashed")
```



```r
pt(abs(t1),df=(N-K), lower.tail=FALSE) * 2
```

```
## inc_spend_capita
##     0.0003542927
```

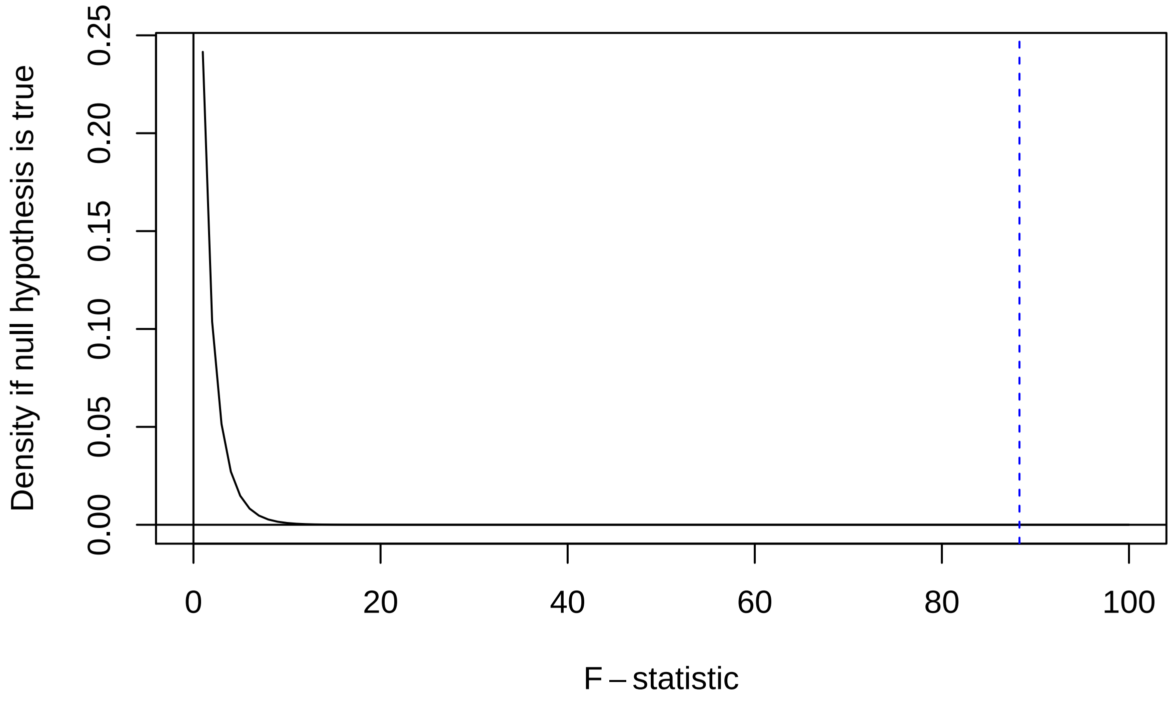
```r
#hypothesis 2: (see summary output)

#hypothesis 3:
linearHypothesis(model.sen,
                 c("inc_spend_capita=-ch_spend_capita"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## inc_spend_capita  + ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ inc_spend_capita + ch_spend_capita
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    293 2.8216
## 2    292 2.1665  1   0.65508 88.293 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
A <- matrix(c(0,1,1), nrow=1)
b <- 0
bread <- A %*% model.sen$coef -b #We want to know is this close enough to 0?
meat <- solve(A %*% solve(t(X) %*% X) %*% t(A) )
F.stat <- (t(bread) %*% meat %*% bread)/(1 * sigma2.hat)
F.stat
```

```
##          [,1]
## [1,] 88.29276
```

```r
curve(df(x, df1=nrow(A), df2=(N-K)), from=-0.000, to=100,
      xlab=expression(F-"statistic"),
      ylab="Density if null hypothesis is true")
abline(h=0); abline(v=0)
abline(v=F.stat, col="blue", lty="dashed")
```



```
pf(F.stat,
   df1=nrow(A), #number of restrictions
   df2=N-K,
   lower.tail = F)
```

```
##              [,1]
## [1,] 1.697763e-18
```

```
#What have we learned
```

3.2 Prediction

Now that we have OLS estimates, we might be interested in predicting values of the outcome for a new or counterfactual observation, let's call it x^*. Predictions will contain two parts:

1. The predicted value y^*
2. The uncertainty $\text{Var}(y^*|X = x^*)$

Note that we can draw a distinction between predictions (what value of y^* do we predict for a new and previously unobserved x^*) and expectations (what is the expected value of y^* when X takes on the value x^*). The second of these is a question about $\text{E}[y|X = x^*]$. The estimate is identical, but the variance is less because the $\text{Var}(\text{E}[y|X = x^*])$ does not need to include contain the population variance σ^2_ε. The main distinction is that a prediction is about what is a plausible *new* value of y for some newly observed x^*, while an expectation is what does the model say about y when $X = x^*$? It should be obvious that the only technical difference here is that a prediction will have larger confidence intervals, but these are in fact two different substantive questions. Choose between them for substantive reasons.

So, how do we make these predictions? Start with the population model

$$y^* = \beta' x^* + \varepsilon^*,$$

replacing the population values with estimates we get

$$\hat{y}^* = \hat{\beta}' x^* + \varepsilon^* = \hat{\beta}' x^*.$$

We don't know ε^* so we replaced it with it's average value (zero) to form the prediction. However, we will want to account for that in the uncertainty of our prediction.

Let's consider the finite properties:

$$\begin{aligned}
\mathrm{E}[\hat{y}^* - y^* | x^*] &= \mathrm{E}[\hat{\beta}' x^* - \beta' x^* - \varepsilon^* | x^*] = 0 = \mathrm{E}[\hat{y}^* - y^*] \\
\mathrm{Var}[\hat{y}^* | x^*] &= \mathrm{Var}(\hat{\beta}' x^* + \varepsilon^* | x^*) \\
&= \mathrm{Var}(\hat{\beta}' x^* | x^*) + \mathrm{Var}(\varepsilon^* | x^*) - 2\,\mathrm{Cov}(\hat{\beta}' x^*, \varepsilon^* | x^*) \\
&= x^{*\prime}\,\mathrm{Var}(\hat{\beta} | x^*) x^* + \mathrm{Var}(\varepsilon^* | x^*) - 2 x^*\,\mathrm{Cov}(\hat{\beta}, \varepsilon^* | x^*) \\
&= x^{*\prime}\,\mathrm{Var}(\hat{\beta} | x^*) x^* + \mathrm{Var}(\varepsilon^* | x^*).
\end{aligned}$$

Note that we include ε_i^* in the variance because we want our prediction to reflect the uncertainty of generating a "new" observation.

If we make use of independence, homoskedasticity, and normality, we get a nice statement for the variance

$$\mathrm{Var}[\hat{y}^* | x^*] = \sigma_\varepsilon^2 (1 + x^{*\prime} (X'X)^{-1}),$$

and the hypothesis test:

$$\frac{\hat{\beta}' x^* - y^*}{\sigma_\varepsilon \sqrt{1 + x^{*\prime} (X'X)^{-1} x^*}} \sim N(0, 1).$$

Of course we don't know σ_ε. So we end up have to use $\hat{\sigma}_\varepsilon$ and we find ourselves back with the t distribution as usual.

Property B10 *Under assumptions B1-B8, we can get hypotheses about predicted values with* *the test statistic*

$$\frac{\hat{\beta}'x^* - y^*}{\hat{\sigma}_\varepsilon\sqrt{1 + x^{*\prime}(X'X)^{-1}x^*}} \sim t_{N-K},$$

and form confidence intervals for predictions based on distribution

$$CI_\alpha(\hat{y}^*) = \hat{\beta}'x^* \pm t_{N-K,\alpha} \times \hat{\sigma}_\varepsilon\sqrt{1 + x^{*\prime}(X'X)^{-1}x^*}.$$

In comparison, if we were only interested in the expected value of y at some point x^* (not a prediction) then we get

$$\begin{aligned}
\mathrm{E}[\hat{y}^*|X = x^*] &= \hat{\beta}'x^* \\
\mathrm{Var}\left(\mathrm{E}[\hat{y}^*|X = x^*]\right) &= x^{*\prime}\,\mathrm{Var}(\hat{\beta}|x^*)x^* \\
CI_\alpha(\mathrm{E}[\hat{y}^*|x^*]) &= \hat{\beta}'x^* \pm t_{N-K,\alpha} \times \sqrt{x^{*\prime}\hat{\mathrm{Var}}(\hat{\beta}|X)x^*}
\end{aligned}$$

One last note on predictions before moving on. We will consider the one variable case to explore some of the mechanics of prediction. Let $X = [1, X_1]$ and let $x^* = (1, b)'$. We can build the variance of $\hat{\beta}$ based on

$$\begin{aligned}
(X'X)^{-1} &= \begin{bmatrix} N & \sum_{i=1}^N x_{i1} \\ \sum_{i=1}^N x_{i1} & \sum_{i=1}^N x_{i1}^2 \end{bmatrix}^{-1} \\
&= \frac{1}{N}\begin{bmatrix} 1 & \overline{x_1} \\ \overline{x_1} & \overline{x_1^2} \end{bmatrix}^{-1} \\
&= \frac{1}{N(\overline{x_1^2} - \overline{x_1}^2)}\begin{bmatrix} \overline{x_1^2} & -\overline{x_1} \\ -\overline{x_1} & 1 \end{bmatrix}
\end{aligned}$$

Now, consider

$$\begin{aligned}
x^{*\prime}(X'X)^{-1}x^* &= (1,b)\left(\frac{1}{N(\overline{x_1^2}-\overline{x_1}^2)}\begin{bmatrix}\overline{x_1^2} & -\overline{x_1}\\ -\overline{x_1} & 1\end{bmatrix}\right)(1,b)'\\
&= \frac{\overline{x_1^2}-2b\overline{x_1}+b^2}{N(\overline{x_1^2}-\overline{x_1}^2)}\\
&= \frac{\overline{x_1^2}-\overline{x_1}^2+\overline{x_1}^2-2b\overline{x_1}+b^2}{N(\overline{x_1^2}-\overline{x_1}^2)}\\
&= \frac{\overline{x_1^2}-\overline{x_1}^2}{N(\overline{x_1^2}-\overline{x_1}^2)}+\frac{\overline{x_1}^2-2b\overline{x_1}+b^2}{N(\overline{x_1^2}-\overline{x_1}^2)}\\
&= \frac{1}{N}+\frac{(b-\overline{x_1})^2}{N(\overline{x_1^2}-\overline{x_1}^2)}
\end{aligned}$$

All of that is to say that we get a prediction interval of

$$\hat{\beta}_0+b\hat{\beta}_1 \pm t_{N-K,\alpha}\hat{\sigma}_\varepsilon\sqrt{1+\frac{1}{N}+\frac{(b-\overline{x_1})^2}{N(\overline{x_1^2}-\overline{x_1}^2)}}.$$

When is this going to be wider/narrower? Narrowest at $b=\overline{x_1}$ and wider the further you get. Predictions are most precise when looking at an average case. As you move further away from the mean these intervals will get larger.

3.2.1 Application

Let's try another quick application with our presidential vote data. Here's a relevant example, let's try to predict presidential vote share using

- incumbent president's approval rating
- growth (Projection of annual growth using the first three quarters of the election year, Ray Fair's data)
- inflation (Absolute increase in the GDP deflator over first 15 quarters in office, Ray Fair's data)

```r
library(readstata13)
presdata <- read.dta13("Rcode/datasets/presvote.dta")
presdata$IncShare[33:34] <- c(0.4631,0.5196 )
model2 <- lm(IncShare~App+Growth+Inflation, data=presdata)
summary(model2)
```

```
## 
## Call:
## lm(formula = IncShare ~ App + Growth + Inflation, data = presdata)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.047855 -0.018897 -0.002957  0.022635  0.059368 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.3615128  0.0412504   8.764 2.75e-07 ***
## App         0.0024016  0.0006348   3.783  0.00180 ** 
## Growth      0.0122148  0.0034032   3.589  0.00268 ** 
## Inflation   0.0009765  0.0038339   0.255  0.80241    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.03389 on 15 degrees of freedom
##   (15 observations deleted due to missingness)
## Multiple R-squared:  0.6429, Adjusted R-squared:  0.5715 
## F-statistic: 9.001 on 3 and 15 DF,  p-value: 0.001187
```

We can interpret these results and hypotheses tests. Go ahead and wow me. What we're most interested in here is forming a prediction for the vote share in 2016 and 2020. Caution on 2020: Growth is way below anything seen in the data. This means we're forecasting in uncharted territory, which is dangerous and prone to error.

```
## Make a prediction for 2016, 2020
new.data <- data.frame(App=c(54,46),
                       Growth=c(0.97,-5.07), #Ray Fair 10/29/2020 data
                       Inflation=c(1.42,1.8)) #Ray Fair 10/29/2020 data
rownames(new.data) <- c("2016", "2020")
fitted <- predict(model2, newdata = new.data, interval ="prediction")
## NOTE
## interval="confidence" gives you the interval for \hat{E[y|X=x.star]}
## interval="prediction" gives you the interval for \hat{y}|X=x.star
fitted
```

```
##            fit       lwr       upr
## 2016 0.5044332 0.4280662 0.5808001
## 2020 0.4118142 0.3161864 0.5074420
```

```
#true 2016
(65853514)/(65853514+62984828)
```

```
## [1] 0.5111329
```

```
#true 2020
(74216154)/(74216154+81268924)
```

```
## [1] 0.4773201
```

```
## calculate "by hand"
X <- na.omit(with(presdata, cbind(1, App, Growth, Inflation)))
x.star <- as.matrix(cbind(1,new.data))
sigma.hat2 <- summary(model2)$sigma^2
y.hat <- x.star %*% model2$coef
var.yhat <- sigma.hat2*(diag(2) + x.star %*% solve(t(X) %*% X) %*% t(x.star))
se.yhat <- sqrt(diag(var.yhat))
t.crit <- qt(.975, df=nrow(X)-ncol(X))
predictions.byhand <- cbind(y.hat,
                            y.hat - t.crit*se.yhat,
                            y.hat + t.crit*se.yhat)
colnames(predictions.byhand) <- c("Fitted", "CI.Low", "CI.high")
rownames(predictions.byhand) <- c("2016", "2020")
predictions.byhand
```

```
##         Fitted    CI.Low   CI.high
## 2016 0.5044332 0.4280662 0.5808001
## 2020 0.4118142 0.3161864 0.5074420
```

Which is what we wanted to show.

3.3 Best estimation

One criterion that we can use to justify OLS, or indeed select any estimator, is the idea choosing the unbiased estimator with the least variance. How does OLS do on this scale? Let's also restrict ourselves to searching over linear estimators (we like linear estimators for their ease and interpretability). Consider an arbitrary unbiased linear estimator of $\hat{\beta}_A = Cy + d$. For $\hat{\beta}_A$ to be unbiased we need the expectation to be β, so

$$\begin{aligned}
\mathrm{E}[\hat{\beta}_A|X] &= \mathrm{E}[Cy + d|X] \\
&= \mathrm{E}[CX\beta + \varepsilon|X] + d \\
&= CX\beta + C\,\mathrm{E}[\varepsilon|X] + d \\
&= CX\beta + d = \beta,
\end{aligned}$$

which only works for $CX = I$ and $d = 0$. Notice that OLS satisfies this condition.

Let's check the variance of $\hat{\beta}_A$.

$$\begin{aligned}
\mathrm{Var}(Cy|X) &= C\,\mathrm{Var}(y|X)C' \\
&= C\,\mathrm{Var}(X\beta + \varepsilon|X)C' \\
&= C\,\mathrm{Var}(\varepsilon|X)C' \\
&= \sigma_\varepsilon^2 CC'
\end{aligned}$$

Define $D = C - (X'X)^{-1}X'$, then we have $DX = CX - (X'X)^{-1}X'X = I - I = 0$ and $C = D + (X'X)^{-1}X'$. Putting a specific form of does not make the proof any less general as C as this can still be any linear unbiased estimator so long as $DX = 0$. Writing it this way *does* help us understand the variance though.

$$\begin{aligned}
\sigma_\varepsilon^2 CC' &= \sigma_\varepsilon^2(D + (X'X)^{-1}X')(D + (X'X)^{-1}X')' \\
&= \sigma_\varepsilon^2(DD' + DX(X'X)^{-1} + (X'X)^{-1}(DX)' + (X'X)^{-1}X'X(X'X)^{-1} \\
&= \sigma_\varepsilon^2 DD' + \sigma_\varepsilon^2(X'X)^{-1}.
\end{aligned}$$

Note that the second term is the variance of $\hat{\beta}$ and we can show that $\sigma^2 DD'$ is a positive semidefinite matrix. As such we find that

$$z'\,\mathrm{Var}(Cy|X)z = z'\left(\mathrm{Var}(\hat{\beta}|X) + \sigma^2 DD'\right)z = z'\,\mathrm{Var}(\hat{\beta}|X)z + z'\sigma^2 DD'z \geq z'\,\mathrm{Var}(\hat{\beta}|X)z.$$

So as a whole the variance-covariance will be weakly larger. As such we say that OLS has the smallest variance amongst unbiased linear estimators, a condition called BLUE: the best linear unbiased estimator. This is the next property of OLS

Property B11 Gauss-Markov theorem *OLS is the minimum variance estimator among all linear unbiased estimators under Assumptions B1-B7 (note we didn't need Assumption B8 for this).*

4 Large sample properties of OLS

The above provides us with good insight into the OLS estimator and shows us many desirable properties in finite samples However, homoskedastiscity and normal errors are still very restrictive. But, without them we lose some of our finite sample properties. As such, we probably want to know what we can and can't do with out them. To figure this out we'll need to consider the large sample (asymptotic) properties of OLS. We will reintroduce our assumptions as we go.

4.1 Asymptotic properties of the OLS estimator

Note that the OLS estimator remains

$$\hat{\beta} = (X'X)^{-1}X'y$$

We will now reintroduce our assumptions.
Assumption B1 *The population DGP is linear* $y_i = \beta' x_i + \varepsilon_i$

Assumption B2 $\mathrm{E}[\varepsilon_i | x_i] = 0$

Assumption B3 *The pairs of* (x_i, ε_i) *are independent and identically distributed.*

Recall that these three assumptions are all we need for OLS to be unbiased (if the OLS estimator exists).

Before moving further, we will need to remind ourselves about a result from math camp: the (weak) Law of Large Numbers.
Theorem 13 (Weak Law of Large Numbers) *Let* $X_1, \ldots, X_N$ *be an iid sequence of random variables, where each* X_i *has a finite absolute first moment* $\mathrm{E}[X_i] = \mu < \infty$*. Then* $\frac{1}{N}\sum_{i=1}^{N} X_i \xrightarrow{p} \mathrm{E}[X_i]$.

In order to apply this result we need two more assumptions
Assumption B4 $\mathrm{E}\left[x_i x_i'\right] < \infty$ *and* $\mathrm{E}\left[x_i \varepsilon_i\right] < \infty$*.*

Assumption B5 $\mathrm{E}[x_i x_i']$ *is symmetric and positive definite*

Finally, we will state three more useful results:
Theorem 14 *Let* g *be a continuous function and let* X *and* Y *be random variables*

- *If* X *and* Y *are independent, then* $g(X)$ *and* $g(Y)$ *are independent random variables*

- *If X and Y are identically distributed, then $g(X)$ and $g(Y)$ are identically distributed*

Theorem 14 is a deceptively powerful result. We can now say that since we know x_i and ε_i are each iid, then $g(x_i)$ and $g(\varepsilon_i)$ will retain these properties for continuous g.

The next two theorems are clutch for analyzing asymptotic properties.

Theorem 15 (Continuous mapping theorem) *Consider a sequence of random variables $X_n = X_1, \dots X_N$ and a continuous function g*

- *If $X_n \xrightarrow{d} X$, then $g(X_n) \xrightarrow{d} g(X)$*
- *If $X_n \xrightarrow{p} X$, then $g(X_n) \xrightarrow{p} g(X)$*

Theorem 16 (Slutsky's Theorem) *Let X_i and Y_i be sequences of random variables.*

1. *If $X_n \xrightarrow{d} X$ and $Y_n \xrightarrow{p} y$ (where y is a constant), then*
   - $X_n Y_n \xrightarrow{d} Xy$
   - $Y_n^{-1} X_n \xrightarrow{d} (y^{-1})X$, *if y^{-1} exists*
2. *If $X_n \xrightarrow{p} x$ and $Y_n \xrightarrow{p} y$ (where x and y are constants), then*
   - $X_n Y_n \xrightarrow{p} xy$
   - $Y_n^{-1} X_n \xrightarrow{p} (y^{-1})X$, *if y^{-1} exists*

This second form is the result of the simple fact that convergence in distribution to a constant implies convergence in probability.

With these tools in hand we can characterize the asymptotic distribution of the OLS estimator. We'll do this in parts:

1. Let's start with the expression $\frac{1}{N}\sum_{i=1}^{N} x_i x_i'$, what happens here as N grows? By Assumption B4 we know that $\mathrm{E}\left[x_i x_i'\right] < \infty$ and that by Theorem 14 $(x_i x_i')_{i=1}^{N}$ are N iid matrices (i.e., if x_i and x_j are iid, then x_i^2 and x_j^2 are iid). As such, the conditions of Theorem 13 are satisfied and we have that

$$\frac{1}{N}\sum_{i=1}^{N} x_i x_i' \xrightarrow{p} \mathrm{E}[x_i x_i'].$$

Further the determinant is a continuous function. By Assumption B5 $\det(\mathrm{E}[x_i x_i']) > 0$, and

$$\det\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right) \xrightarrow{p} \det(\mathrm{E}[x_i x_i']) > 0$$

by the continuous mapping theorem. This means that for sufficiently large N, $\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1}$, and thus $\hat{\beta}$ will exist with near certainty. Likewise, the matrix inverse is a continuous function for matrices with non-zero determinants, so

$$\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \xrightarrow{p} \mathrm{E}[x_i x_i']^{-1},$$

by the CMT.

2. Next consider $\frac{1}{N}\sum_{i=1}^{N} x_i \varepsilon_i$. It follows from Assumption B3 that $(x_i\varepsilon_i)_{i=1}^{N}$ are N iid vectors. By Assumptions B4 and B2, x_i and ε_i are linearly independent of each other with
$$\mathrm{E}\left[x_i\varepsilon_i\right] = \mathrm{E}_x[\mathrm{E}[x_i\varepsilon_i|X]] = \mathrm{E}_x[x_i\,\mathrm{E}[\varepsilon_i|X]] = \mathrm{E}_x[x_i\,\mathrm{E}[\varepsilon_i|x_i]] = 0.$$

   The conditions of Theorem 13 are satisfied with

$$\frac{1}{N}\sum_{i=1}^{N} x_i\varepsilon_i \xrightarrow{p} \mathrm{E}[x_i\varepsilon_i] = 0.$$

3. We can now apply Slutsky's theorem to say:

$$\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i\varepsilon_i \xrightarrow{p} \mathrm{E}[x_i x_i']0 = 0$$

   And since the sum operator is continuous we can again apply the continuous mapping theorem to get
$$\begin{aligned}\hat{\beta} &= \beta + \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i\varepsilon_i \\ &\xrightarrow{p} \beta + 0 \text{ By CMT and Slutsky} \\ &\xrightarrow{p} \beta.\end{aligned}$$

This takes us to our next property
Property B12 *Under Assumptions B1-B5, OLS exists with near certainty for large enough N and $\hat{\beta}^N \xrightarrow{p} \beta$.*

Note that Property B12 is gives us both existence (for large enough N) and consistency of the OLS estimator. Consistency is great, but we'd also like to know something about the distribution of $\hat{\beta}$. Without an assumption on ε, however, we cannot characterize the finite sample distribution. Instead, we will rely on our old friend the Central Limit Theorem, which we restate, in vector form

Theorem 17 *(Central Limit Theorem) Let $X_1, \ldots, X_N$ be iid random varibles with expected value μ and variance Ω (both finite), then for all $x \in \mathbb{R}$*

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} X_i - \mu\right) \xrightarrow{d} N(0, \Omega).$$

To satisfy the conditions of Theorem 17 we need the following assumption
Assumption B6 $\operatorname{Var}(x_i\varepsilon_i)$ *is finite.*

Now, we're cooking. Let's rewrite $\hat{\beta}$ to look like the CLT.

$$\sqrt{N}\left(\hat{\beta}^N - \beta\right) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1}\left(\frac{\sqrt{N}}{N}\sum_{i=1}^{N} x_i\varepsilon_i\right).$$

We know that $(x_i\varepsilon_i)_{i=1}^{\infty}$ are a sequence of iid random variables with expectation 0 and finite variance by Assumptions B2, B3, B4, and B6. This means that we have satisfied the conditions for Theorem 17. This gives us:

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} x_i\varepsilon_i\right) \xrightarrow{d} N(0, \operatorname{Var}(x_i\varepsilon_i)),$$

Likewise, by the arguments we used to show Property B12 we have

$$\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \xrightarrow{p} \mathrm{E}[x_i x_i']^{-1}.$$

Combining terms and applying Slutsky's theorem we get

$$\sqrt{N}\left(\hat{\beta}^N - \beta\right) \xrightarrow{d} \mathrm{E}[x_i x_i']^{-1} Z$$

where $Z \sim N(0, \operatorname{Var}(x_i\varepsilon_i))$. As such we find ourselves the next property
Property B13 *Under Assumptions B1-B6, the OLS estimates are asymptotically normal such that*

$$\sqrt{N}\left(\hat{\beta}^N - \beta\right) \xrightarrow{d} N\left(0, \mathrm{E}[x_i x_i']^{-1}\operatorname{Var}(x_i\varepsilon_i)\,\mathrm{E}[x_i x_i']^{-1}\right) = N(0, \Omega).$$

In order to use Property B13 we need to have expressions for $\mathrm{E}[x_i x_i']$ and $\operatorname{Var}(x_i\varepsilon_i)$ in terms of data. Note that

$$\operatorname{Var}(x_i\varepsilon_i) = \mathrm{E}[x_i x_i'\varepsilon_i^2] - \mathrm{E}[x_i]\,\mathrm{E}[\varepsilon_i] = \mathrm{E}[x_i x_i'\varepsilon_i^2]$$

We can backout consistent estimates based on assumptions we've made:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N} x_i x_i' &\xrightarrow{p} \mathrm{E}[x_i x_i'] \\ \frac{1}{N}\sum_{i=1}^{N} x_i x_i' \hat{\varepsilon}_i^2 &= \frac{1}{N}\sum_{i=1}^{N} x_i x_i'(y_i - \hat{\beta}' x_i)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} x_i x_i'(\beta' x_i + \varepsilon_i - \hat{\beta}' x_i)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} x_i x_i'(\varepsilon_i - (\hat{\beta} - \beta)' x_i)^2\end{aligned}$$

Given that $(\hat{\beta} - \beta) \xrightarrow{p} 0$, we can apply Slutsky's theorem and the CMT to we get

$$\frac{1}{N}\sum_{i=1}^{N} x_i x_i' \hat{\varepsilon}_i^2 \xrightarrow{p} \mathrm{Var}(x_i \varepsilon_i).$$

Note that we've made some additional assumptions on the existence moments of x_i, but no real harm done, and it's not overly helpful to get into the weeds on this. Now however, we can now write out a consistent variance estimator for $\sqrt{N}(\hat{\beta} - \beta)$ in terms of the data. Specifically, we have

$$\hat{\Omega} = \left[\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right]^{-1} \left[\frac{1}{N}\sum_{i=1}^{N} x_i x_i' \hat{\varepsilon}_i^2\right] \left[\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right]^{-1} \xrightarrow{p} \mathrm{E}[x_i x_i']^{-1}\,\mathrm{Var}(x_i \varepsilon_i)\,\mathrm{E}[x_i x_i']^{-1}.$$

Which can be computed as

$$\begin{aligned}\widehat{\mathrm{avar}(\hat{\beta})} &= \frac{1}{N}\hat{\Omega} \\ &= (X'X)^{-1}\left(\sum_{i=1}^{N} x_i x_i' \hat{\varepsilon}_i^2\right)(X'X)^{-1} \\ &= (X'X)^{-1}\,(X_{\hat{\varepsilon}}' X_{\hat{\varepsilon}})\,(X'X)^{-1},\end{aligned}$$

where $X_{\hat{\varepsilon}}$ is the matrix X with each column is element-by-element multiplied by $\hat{\varepsilon}$. This estimate of $\frac{1}{N}\Omega$ is called the "sandwich" estimator or the Huber-White covariance matrix, or the "robust" variance matrix. You'll often see reference to "robust" standard errors, this is where they come from. This particular estimate is only asymptotically valid (don't forget that). One final tweak. We often add in a degrees of freedom correction to the robust version, such that

$$\widehat{\mathrm{avar}(\hat{\beta})} = \frac{N}{N-K}(X'X)^{-1}\,(X_{\hat{\varepsilon}}' X_{\hat{\varepsilon}})\,(X'X)^{-1} = \frac{1}{N-K}\hat{\Omega}.$$

For larger N this doesn't make a big difference. This format will open the door for us to make large-N inferences about β without putting strong assumptions on ε like homoskedasticity or normality. Notably let $rse(\hat{\beta}_k)$ be the "robust" standard error from taking the diagonal of the asymptotic covariance matrix. Then for large N we can say that

$$\frac{\hat{\beta}_k - b}{rse(\hat{\beta}_k)} \overset{a}{\sim} N(0,1).$$

Why does this matter? It allows us to test hypotheses if we believe that homoskedasticity fails to hold. One common reason for homoskedasticity failing is if there's more "noise" at some values of x_i. For example, if we want use income to predict expenditure on meals we might notice that at low incomes the variance in meal choice is more limited and there is less variance. As income rises, there are more options and thus more variance (Whataburger versus The Republic).

However, let's suppose we are in a homoskedastic world
Assumption B7 $\mathrm{E}[\varepsilon_i^2|x_i] = \sigma_\varepsilon^2$

Now we can make some simplifications:

$$\begin{aligned}\mathrm{Var}(x_i\varepsilon_i) &= \mathrm{E}[x_i x_i' \varepsilon_i^2] - \mathrm{E}[(x_i\varepsilon_i)]^2 \\ &= \mathrm{E}[x_i x_i' \,\mathrm{E}[\varepsilon_i^2|x_i]] \\ &= \mathrm{E}[x_i x_i' \sigma_\varepsilon^2] \\ &= \sigma_\varepsilon^2 \,\mathrm{E}[x_i x_i']\end{aligned}$$

Substituting this into the above we find

$$\Omega_0 = \mathrm{E}[x_i x_i']^{-1}\,\mathrm{Var}(x_i\varepsilon_i)\,\mathrm{E}[x_i x_i']^{-1} = \sigma_\varepsilon^2\,\mathrm{E}[x_i x_i']^{-1},$$

which looks a lot like our finite sample version. As such we have a new result
Property B14 *Under Assumptions B1-B7, the OLS estimates are asymptotically normal such that*

$$\sqrt{N}\left(\hat{\beta}^N - \beta\right) \xrightarrow{d} N\left(0, \sigma_\varepsilon^2\,\mathrm{E}[x_i x_i']^{-1}\right).$$

Time out for a second. We now have two forms of the asymptotic covariance matrix of $\hat{\beta}$. The homoskedastic version is estimated as

$$\widehat{\mathrm{avar}}(\hat{\beta}) = \hat{\sigma}_\varepsilon^2 (X'X)^{-1} = \frac{1}{N}\hat{\Omega}_0,$$

which is aysmptotically valid under assumptions B1-B7. Under the same assumptions, it is used for $\widehat{\text{Var}}(\hat{\beta}|X)$, so we can use this formula in either large or small samples without concern. The heteroskedastic version is estimated as

$$\widehat{\text{avar}}(\hat{\beta}) = \frac{N}{N-K}(X'X)^{-1}\,(X_{\hat{\varepsilon}}'X_{\hat{\varepsilon}})\,(X'X)^{-1} = \frac{1}{N-K}\hat{\Omega},$$

which is only valid in large samples. We'll return to these differences later on. For now, just file them away. Note that if homoskedasticity is a good assumption, than these will be asymptotically equivalent.

Let's go back to the characterization under Property B13. Namely that

$$\sqrt{N}\left(\hat{\beta}^N - \beta\right) \xrightarrow{d} N\left(0, \text{E}[x_i x_i']^{-1}\,\text{Var}(x_i\varepsilon_i)\,\text{E}[x_i x_i']^{-1}\right) = N(0,\Omega).$$

Now suppose we want to test a set of J linear hypotheses (like before) such that

$$\begin{aligned} H_0 &: A\beta = b \\ H_A &: A\beta \neq b \end{aligned}$$

What would our test statistic be? Like before we start by assuming a true null hypothesis such that

$$\begin{aligned} A\hat{\beta} - b &= A\hat{\beta} - A\beta \\ &= A(\hat{\beta} - \beta) \\ \sqrt{N}(A\hat{\beta} - b) &= A\sqrt{N}(\hat{\beta} - \beta) \\ \sqrt{N}(A\hat{\beta} - b) &\xrightarrow{d} N(0, A\Omega A') \\ [A\Omega A']^{-1/2}\sqrt{N}(A\hat{\beta} - b) &\xrightarrow{d} N(0, I_J) \\ N(A\hat{\beta} - b)'[A\Omega A']^{-1}(A\hat{\beta} - b) &\xrightarrow{d} \chi^2_J \\ N(A\hat{\beta} - b)'[A\hat{\Omega} A']^{-1}(A\hat{\beta} - b) &\xrightarrow{d} \chi^2_J \\ (A\hat{\beta} - b)'\left[A\frac{1}{N}\hat{\Omega}A'\right]^{-1}(A\hat{\beta} - b) &\xrightarrow{d} \chi^2_J \\ (A\hat{\beta} - b)'\left[A\widehat{\text{avar}}(\hat{\beta})A'\right]^{-1}(A\hat{\beta} - b) &\xrightarrow{d} \chi^2_J \end{aligned}$$

Note that can replace Ω with $\hat{\Omega}$ or $\hat{\Omega}_0$ as either can be consistent for Ω, depending whether we want to use Assumption B7. Unsurprisingly this looks like our finite sample F test from before.

Anyway going back to the linear hypothesis test we have our next property:

Property B15 *Under Assumptions B1-B6, the null hypothesis $H_0 : A\beta = b$ implies a tests statistic*

$$W = (A\hat{\beta} - b)' \left[A\widehat{\text{avar}}(\hat{\beta})A'\right]^{-1} (A\hat{\beta} - b) \xrightarrow{d} \chi^2_J.$$

Not all interesting tests are linear however. Sometimes we might have a length J nonlinear hypothesis of the form $C(\beta)$. In order to make progress here, we will need to know the distribution of $C(\hat{\beta})$. This is a question about a distribution of a nonlinear function of a normally distributed random variable.

We've seen nonlinear transformations of random variables before. The first thing that you might think to do is to take a Taylor series approximation of $C(\hat{\beta})$ around the true β.

$$C(\hat{\beta}) = C(\beta) + D_\beta C(\beta)(\hat{\beta} - \beta) + o(|\hat{\beta} - \beta|)$$

Recall that $o(|\hat{\beta} - \beta|)$ means that $o(|\hat{\beta} - \beta|) \to 0$ "faster" than $|\hat{\beta} - \beta|$ goes to 0. Note that $\hat{\beta} - \beta \xrightarrow{p} 0$ (at a rate of $\sqrt{N}$), and so we're entire right hand side converges to $C(\beta)$ by the continuous mapping theorem and Slutsky's theorem:

$$C(\hat{\beta}) \xrightarrow{p} C(\beta).$$

Recall that OLS is asymptotically normal, $\sqrt{N}(\hat{\beta} - \beta) \xrightarrow{d} N(0, \Omega)$. If we rearrange some terms in the Taylor series from above we get

$$\sqrt{N}(C(\hat{\beta}) - C(\beta)) = \underbrace{D_\beta C(\beta)}_{\text{constant}} \underbrace{\sqrt{N}(\hat{\beta} - \beta)}_{\text{Converges to } N(0,\Omega)} + \underbrace{\sqrt{N} o(|\hat{\beta} - \beta|)}_{\text{converges to 0}}.$$

We can apply Slutsky's theorem and the CMT to get:

Property B16 *Under Assumptions B1-B6, the distribution of $C(\hat{\beta}) = 0$, where C is continuously differentiable, is asymptotically normal with:*

$$\sqrt{N}(C(\hat{\beta}) - C(\beta)) \xrightarrow{d} N\left(0, D_\beta C(\beta)\Omega D_\beta C(\beta)'\right).$$

Notice that this is the delta method expression for the variance of $C(\hat{\beta})$. In the scalar case we squared the derivative of C and multiplied by the variance. This expression is the multivariate extension of that. This result gives us a firm basis for using the delta method to produce standard errors for transformations of OLS estimates.

Applying the logic from the linear hypothesis test we can get to a usable test statistic for the hypothesis that $C(\beta) = 0$, specifically we can standardize and square to get

$$\underbrace{C(\hat{\beta})'}_{\text{Obs}-\text{Null}} \underbrace{\left[D_\beta C(\beta) \frac{1}{N} \Omega D_\beta C(\beta)'\right]^{-1}}_{\text{Delta method variance}} \underbrace{C(\hat{\beta})}_{\text{Obs}-\text{Null}} \xrightarrow{d} \chi^2_J,$$

where again we exploit the convergence of $C(\hat{\beta}) \to C(\beta)$ to rewrite this as a new property
Property B17 *Under Assumptions B1-B6, the null hypothesis $H_0 : C(\beta) = 0$ implies a tests statistic*

$$W = C(\hat{\beta})' \left[D_\beta C(\hat{\beta}) \widehat{\text{avar}}(\hat{\beta}) D_\beta C(\hat{\beta})'\right]^{-1} C(\hat{\beta}) \xrightarrow{d} \chi^2_J.$$

This test is called a **Wald test**, and it's straightforward to see that it encompasses the linear hypothesis case. With Assumption B7 Ω is replaced by Ω_0.

If we make a final assumption
Assumption B8 *ε are normally distributed*

then we can get back all the finite sample tests from before, but we also get one new asymptotic result (and also a new finite result). In fact, the final property asymptotic property we will show depends heavily on Assumption B8. Under B8 We can start by rewriting the data generating process to be

$$y_i \mid x_i \overset{iid}{\sim} N(\beta' x_i, \sigma^2_\varepsilon).$$

This representation gives us the following likelihood model for y_i

$$\begin{aligned}
\mathcal{L}(\beta, \sigma^2_\varepsilon | y, X) &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2_\varepsilon}} \exp\left(-\frac{1}{2\sigma^2_\varepsilon}(y_i - \beta' x_i)^2\right) \\
L(\beta, \sigma^2_\varepsilon | y, X) &= \sum_{i=1}^{N} -\frac{1}{2}\left(\log(2\pi) + \log\left(\sigma^2_\varepsilon\right)\right) + \left(-\frac{1}{2\sigma^2_\varepsilon}\left(y_i - \beta' x_i\right)^2\right) \\
&= -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2_\varepsilon) + \sum_{i=1}^{N}\left(-\frac{1}{2\sigma^2_\varepsilon}(y_i - \beta' x_i)^2\right).
\end{aligned}$$

With FOC

$$\begin{aligned}
D_\beta L(\beta, \sigma_\varepsilon^2|y, X) &= \frac{1}{\sigma_\varepsilon^2}\sum_{i=1}^{N}(y_i - \beta' x_i)x_i \\
0 &= \frac{1}{N}\sum_{i=1}^{N}(y_i x_i - x_i x_i'\beta) \\
0 &= \frac{1}{N}\sum_{i=1}^{N} x_i y_i - \frac{1}{N}\sum_{i=1}^{N} x_i x_i'\beta \\
\hat{\beta}_{MLE} &= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} x_i y_i\right) = (X'X)^{-1}X'y.
\end{aligned}$$

This is our next property of OLS
Property B18 *Under Assumptions B1-B8, $\hat{\beta}$ is the MLE of β.*

So $\hat{\beta}_{MLE} = \hat{\beta}$. What about $\hat{\sigma}_\varepsilon^2$?

$$\begin{aligned}
D_{\sigma_\varepsilon^2} L(\beta, \sigma_\varepsilon^2|y, X) &= -\frac{N}{2\sigma_\varepsilon^2} + \sum_{i=1}^{N}\frac{1}{2\sigma_\varepsilon^4}(y_i - \beta' x_i)^2 \\
0 &= -N + \frac{1}{\sigma_\varepsilon^2}\sum_{i=1}^{N}(y_i - \beta' x_i)^2 \\
\hat{\sigma}_{\varepsilon,MLE}^2 &= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{\beta}' x_i)^2 = \frac{\hat{\varepsilon}'\hat{\varepsilon}}{N},
\end{aligned}$$

which is not the unbiased estimate $\hat{\sigma}_\varepsilon^2$. Asymptotically, the fact that $\hat{\beta}$ is an MLE tells us atleast one new fact about it. Recall that if $\hat{\theta}$ is an MLE, and certain regularity conditions hold, then

$$\sqrt{N}(\hat{\theta} - \theta) \xrightarrow{d} N\left(0, \left(\frac{I(\theta)}{N}\right)^{-1}\right),$$

where $I(\theta) = -\operatorname{E}[H_\theta \log(L(\theta; y))]$ is the estimated Fisher Information.
Recall that the Cramér-Rao lower bound tells us that the lowest possible variance any unbiased estimator can achieve is equal to the inverse Fisher Information, and that MLE asymptotically obtain the Cramér-Rao lower bound.

Let's see what this lower bound is for unbiased estimators. Let $\theta = (\beta, \sigma^2)$ and recall that the log-likelihood under our current assumptions is

$$L(\beta, \sigma_\varepsilon^2|y, X) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma_\varepsilon^2) + \sum_{i=1}^{N}\left(-\frac{1}{2\sigma_\varepsilon^2}(y_i - \beta' x_i)^2\right)$$

with gradient

$$D_\theta L(\beta, \sigma_\varepsilon^2 | y, X) = \begin{bmatrix} \frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^N (y_i - \beta' x_i) x_i \\ -\frac{N}{2\sigma_\varepsilon^2} + \sum_{i=1}^N \frac{1}{2\sigma_\varepsilon^4} (y_i - \beta' x_i)^2 \end{bmatrix}$$

and the hessian

$$H_\theta L(\beta, \sigma_\varepsilon^2 | y, X) = \begin{bmatrix} -\frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^N x_i x_i' & -\frac{1}{\sigma_\varepsilon^4} \sum_{i=1}^N (y_i - \beta' x_i) x_i \\ -\frac{1}{\sigma_\varepsilon^4} \sum_{i=1}^N (y_i - \beta' x_i) x_i & \sum_{i=1}^N \frac{1}{2\sigma_\varepsilon^4} - \frac{1}{\sigma_\varepsilon^6} (y_i - \beta' x_i)^2 \end{bmatrix}.$$

The Fisher Information

$$\begin{aligned} \frac{I(\theta)}{N} &= -\operatorname{E}\left[\frac{1}{N} H_\theta L(\beta, \sigma_\varepsilon^2 | y, X)\right] \\ &= \begin{bmatrix} \frac{1}{\sigma_\varepsilon^2} \operatorname{E}[x_i x_i'] & \frac{1}{\sigma_\varepsilon^4} \operatorname{E}[x_i \varepsilon_i] \\ \frac{1}{\sigma_\varepsilon^4} \operatorname{E}[x_i \varepsilon_i] & -\frac{1}{2\sigma_\varepsilon^4} + \frac{1}{\sigma_\varepsilon^6} \operatorname{E}[(y_i - \beta' x_i)^2] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_\varepsilon^2 \operatorname{E}[x_i x_i'] & 0 \\ 0 & \frac{1}{2\sigma_\varepsilon^4} \end{bmatrix} \\ \left(\frac{I(\theta)}{N}\right)^{-1} &= \begin{bmatrix} \sigma_\varepsilon^2 \operatorname{E}[x_i x_i']^{-1} & 0 \\ 0 & 2\sigma_\varepsilon^4 \end{bmatrix}. \end{aligned}$$

Note that the upper left looks like the OLS covariance matrix with homoskedastic errors. This implies with with normality and homoskedasticity, no asymptotically unbiased estimator can have smaller variance than OLS.

Property B19 *Under Assumptions B1-B8, OLS is asymptotically efficient. No asymptotically unbiased estimator is better than OLS (linear or not).*

To clarify

$$\sqrt{N}(\hat{\theta} - \theta) \xrightarrow{d} N\left(0, \left(\frac{I(\theta)}{N}\right)^{-1}\right)$$

$$\hat{\theta} \overset{asy}{\sim} N\left(\theta, I(\theta)^{-1}\right)$$

or

$$\hat{\beta}_{MLE} \overset{asy}{\sim} N\left(\beta, \frac{\sigma^2}{N} \operatorname{E}[x_i x_i']^{-1}\right)$$

and

$$\hat{\sigma}^2_{MLE} \overset{asy}{\sim} N\left(\sigma^2, \frac{2\sigma^4}{N}\right)$$

Recall that asymptotic variance of the MLE is the Cramér-Rao lower bound, which is the lowest possible variance of an unbiased estimator. So if we have Assumptions B1-B8 then

OLS is asymptotically efficient hit the Cramér-Rao lower bound. It can be shown that OLS is the unique minimum variance unbiased estimator for any sample size under B1-B8, but we haven't introduced enough tools for that, so take my word for it.

Property B20 *OLS is the minimum variance estimator for β among all unbiased estimators (linear or not) under Assumptions B1-B8.*

Another justification for the OLS estimator can be found if we want to find the optimal predictor of y. If $\mathrm{E}[y_i|x_i]$ is linear and we have our Gauss-Markov assumptions, it should be clear that OLS will be the optimal predictor. But what if $\mathrm{E}[y_i|x_i]$ isn't actually linear? What if we lose Assumption B1 and in it's place we just have $y_i = \mathrm{E}[y_i|x_i] + \varepsilon_i$? Then the MSE in predicting nonlinear y using a potential incorrect linear functional form is given by

$$MMSE_y(\hat{\beta}) = \operatorname*{argmin}_{\beta} E[(y_i - \beta' x_i)^2].$$

With a little bit of algebra (replace y with $\mathrm{E}[y_i|x_i] + \varepsilon_i$, rearrange and simplify), we can rewrite this to be

$$MMSE_y(\hat{\beta}) = \operatorname*{argmin}_{\beta} \mathrm{E}[(y_i - \mathrm{E}[y_i|x_i])^2] + \mathrm{E}[(\mathrm{E}[y_i|x_i] - \beta' x_i)^2]$$

The first term doesn't contain β, so to minimize the MSE we focus on the last term and solve the FOC

$$\begin{aligned} D_\beta \, \mathrm{E}[(\mathrm{E}[y_i|x_i] - \beta' x_i)^2] &= 0 \\ -2\,\mathrm{E}[(\mathrm{E}[y_i|x_i] - \beta' x_i)x_i] &= 0 \\ \mathrm{E}[x_i \, \mathrm{E}[y_i|x_i]] &= \mathrm{E}[x_i x_i']\beta. \end{aligned}$$

Let's look more at the left hand side

$$\begin{aligned} \mathrm{E}[x_i \, \mathrm{E}[y_i|x_i]] &= \mathrm{E}[x_i(y_i - \varepsilon_i)] \\ &= \mathrm{E}[x_i y_i] - \mathrm{E}[x_i \varepsilon_i] \\ &= \mathrm{E}[x_i y_i] \end{aligned}$$

Now we see that the FOC for finding the minimum MSE is that

$$\hat{\beta} = \mathrm{E}[x_i x_i']^{-1} \, \mathrm{E}[x_i y_i].$$

In sample, we would estimate these with the sample means

$$\hat{\beta} = \left(\frac{1}{N} \sum_{i=1}^{N} x_i x_i' \right)^{-1} \frac{1}{N} \sum_{i=1}^{N} x_i y_i.$$

Applying the usual LLN arguments provides consistency of the sample means for their expected values. The end result is that

Property B21 *Under Assumptions B2-B6, OLS is the minimum MSE predictor of y_i among all linear estimators.*

The full set of OLS justifications that we have derived is:

Assumptions	Properties
B1-B3	Unbiased for β (B1)
B2-B5	BLE for y (B21)
B1-B5	Consistent for β (B12)
B1-B5	$\hat{\beta}$ almost surely exists as N increases (B12)
B1-B5.A	$\hat{\beta}$ exists for any $N \geq K$ (B2)
B1-B6	$\hat{\beta} \overset{asy}{\sim} N$ (B13)
B1-B6	Asymptotic hypothesis tests for linear and nonlinear hypotheses about β (B13, B15, B17)
B1-B6	Delta Method provides asymptotically correct variance for nonlinear transformations of $\hat{\beta}$ (B16)
B1-B6	Asymptotic confidence intervals for $\mathrm{E}[y \mid X = x^*]$
B1-B7	BLUE for β (B11)
B1-B8	MLE/BUE for β (B18-B20)
B1-B8	Exact hypothesis tests for linear hypotheses about β (B8-B9)
B1-B8	Exact confidence intervals for $\mathrm{E}[y \mid X = x^*]$ and predictions B10

Note: Anything listed as B1-B6 can be B1-B7 if you use Ω_0 instead of Ω.

4.2 Application

Let's step aside and consider a specific problem with real data. We're going to look at how counties evaluate the value of homes for property tax. Specifically, we want to know if property assessments by the county accurately reflect real home values. We will look at some data from the Shadyside neighborhood of Pittsburgh, Pennsylvania. Let A_i denote the county assessment of house i and let S_i be the sale price of the home. If assessments are accurate we would expect $A_i = S_i$, but we want to allow for some fluctuations and random chance differences between assessments and sales so we want to relate sales price to assessments plus noise

$$A_i = S_i + \varepsilon_i,$$

where ε_i denotes the random error that generates the gaps between "true" home value and assessments and $\mathrm{E}[\varepsilon|S_i] = 0$. Note that we are assuming the sales price reflects true value. We will consider the consequences of having sales price be a noisy reflect of true value later.

We can make this a more general more by rewriting it in linear regression format

$$A_i = \beta_0 + \beta_1 S_i + \varepsilon_i.$$

Now the expected sales price of house i given it's assessment is $\mathrm{E}[A_i|S_i] = \beta_0 + \beta_1 S_i$. If our original model is correct then we expect $\beta_0 = 0$ and $\beta_1 = 1$. We are interested in testing these hypotheses. We will make assumptions B1-B8, which include: Independence of observations, Homoskedasticity, Normality, and Full rank and finiteness of

$$\mathrm{E}\begin{bmatrix} 1 & S_i \\ S_i & S_i^2 \end{bmatrix}.$$

Note that this will be full rank so long as it is not the case that S_i is not realized as a constant c with probability 1. With this structure on the problem we can test our two hypotheses both individual and jointly such that

$$H_0 : \beta_k = b_k$$
$$H_0 : A\beta = b.$$

With the above assumptions this produces test statistics

$$\frac{\hat{\beta}_k - b_k}{\hat{\sigma}_\varepsilon\sqrt{[(X'X)^{-1}]_{kk}}} = \frac{\hat{\beta}_k - b_k}{s.e.(\hat{\beta}_k)} \sim t_{N-2}$$
$$\frac{(A\hat{\beta} - b)'[A\hat{\sigma}_\varepsilon^2(X'X)^{-1}A']^{-1}(A\hat{\beta} - b)}{2} \sim F_{2,N-2}.$$

Where A is an identity matrix of size 2 and $b = (0,1)'$. Note that if we wanted to drop assumption B8, we get similar tests, but we have to rely on the asymptotic distributions,

$$\frac{\hat{\beta}_k - b_k}{\hat{\sigma}_\varepsilon\sqrt{[(X'X)^{-1}]_{kk}}} = \frac{\hat{\beta}_k - b_k}{s.e.(\hat{\beta}_k)} \xrightarrow{d} N(0,1)$$

$$(A\hat{\beta} - b)'[A\hat{\sigma}^2_\varepsilon(X'X)^{-1}A']^{-1}(A\hat{\beta} - b) \xrightarrow{d} \chi^2_2.$$

We can also drop assumption B7 if we use the robust covariance matrix.

$$\frac{\hat{\beta}_k - b_k}{\sqrt{\left[\frac{1}{N-K}\hat{\Omega}\right]_{kk}}} = \frac{\hat{\beta}_k - b_k}{rse(\hat{\beta}_k)} \xrightarrow{d} N(0,1)$$

$$(A\hat{\beta} - b)'\left[A\left[\frac{1}{N-K}\hat{\Omega}\right]A'\right]^{-1}(A\hat{\beta} - b) \xrightarrow{d} \chi^2_2.$$

Let's try it out.

```
library(readstata13)
library(lmtest)
library(sandwich)
library(stargazer)
library(car)

shady <- read.dta13("Rcode/datasets/shady.dta")
shady$salepric <- shady$salepric/1000000 #hundreds of thousands of dollars
shady$asprice <- shady$asprice/1000000
X <- cbind(1, shady$salepric)
eigen((t(X) %*% X)/nrow(X), only.values = TRUE)
```

$values [1] 1.02003029 0.01040915

$vectors NULL

```
model1 <- lm(asprice~salepric,data=shady)

#exact type doesn't matter much, but HC1 is what's in the notes.
# HC3 (default) is probably better in finite samples according to recent sims
robust <- vcovHC(model1,type = "HC1")



stargazer(model1, model1, no.space = TRUE, label='tab:example1',
          header=FALSE, column.labels = c("Classic S.E.'s", "Robust S.E.'s"),
          covariate.labels = c("Sales price", "Intercept"),
          dep.var.labels = "Apraisal", title = "Shadyside Apraisals",
          keep.stat = "n",
          type='latex', #switch to latex for problem sets
          se = list(NULL, #NULL uses default classical errors (homoskedastic)
                    sqrt(diag(robust))))
```

Table 3: Shadyside Apraisals

	Dependent variable:	
	Apraisal	
	Classic S.E.'s	Robust S.E.'s
	(1)	(2)
Sales price	0.786***	0.786***
	(0.013)	(0.037)
Intercept	0.023***	0.023***
	(0.002)	(0.005)
Observations	1,097	1,097
Note:	*p<0.1; **p<0.05; ***p<0.01	

The results of the regression are reported in Table 3. We can test our hypothesis separately. First, let's consider $H_0 : \beta_0 = 0$. We test this by finding that the t ratio $\frac{\hat{\beta}_0 - 0}{se(\hat{\beta}_0)} = 10.322$. This is greater than the critical t value of ± 1.962 or the critical z value of ± 1.960. Because our test statistic is more extreme than our critical values, we reject the null hypothesis that β_0 is zero.

Second, consider $H_0 : \beta_1 = 1$. Now our t statistic is $\frac{\hat{\beta}_1 - 1}{se(\hat{\beta}_1)} =$ -16.726 . We can compare this to the same critical value and see that this is below the negative critical value. As such, we reject the hypothesis that $\beta_1 = 1$. Note we can also conduct these individual tests as a Wald test.

```
linearHypothesis(model1,"(Intercept)=0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## (Intercept) = 0
```

```
##
## Model 1: restricted model
## Model 2: asprice ~ salepric
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1096 2.2954
## 2   1095 2.0918  1   0.20355 106.55 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
round(sqrt(106.55),3)
```

```
## [1] 10.322
```

```r
linearHypothesis(model1,"salepric=1")
```

```
## Linear hypothesis test
##
## Hypothesis:
## salepric = 1
##
## Model 1: restricted model
## Model 2: asprice ~ salepric
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1096 2.6263
## 2   1095 2.0918  1   0.53445 279.76 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
round(sqrt(279.76),3)
```

```
## [1] 16.726
```

Notice that the F statistics here are the square of the individual t statistics. Finally, we can do the more appropriate joint hypothesis tests. We'll look at this 3 ways: with all assumptions, without B8, and without B7 or B8.

```r
linearHypothesis(model1, c("(Intercept)=0", "salepric=1"))
```

```
## Linear hypothesis test
```

```
##
## Hypothesis:
## (Intercept) = 0
## salepric = 1
##
## Model 1: restricted model
## Model 2: asprice ~ salepric
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1097 2.6815
## 2   1095 2.0918  2   0.58966 154.33 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model1, c("(Intercept)=0", "salepric=1"), test = "Chisq")
```

```
## Linear hypothesis test
##
## Hypothesis:
## (Intercept) = 0
## salepric = 1
##
## Model 1: restricted model
## Model 2: asprice ~ salepric
##
##   Res.Df    RSS Df Sum of Sq  Chisq Pr(>Chisq)
## 1   1097 2.6815
## 2   1095 2.0918  2   0.58966 308.66  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model1, c("(Intercept)=0", "salepric=1"),
                 test = "Chisq", vcov = robust)
```

```
## Linear hypothesis test
##
## Hypothesis:
## (Intercept) = 0
## salepric = 1
##
## Model 1: restricted model
## Model 2: asprice ~ salepric
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df Chisq Pr(>Chisq)
## 1   1097
## 2   1095  2 39.33  2.881e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice that they all agree. That's a good thing for our analysis. We will also make sure that these R commands do what we think they do, by doing everything "by hand"

```
X <- cbind(1, shady$salepric)
y <- shady$asprice
b.hat <- solve(t(X) %*% X) %*% t(X) %*% y
cbind(b.hat, model1$coefficients)
```

```
##                   [,1]       [,2]
## (Intercept) 0.02306402 0.02306402
## salepric    0.78579266 0.78579266
```

```
e.hat <- drop(y-X%*%b.hat)
N <- length(y)
sigma.hat2 <-  sum(e.hat^2)/(N-2)
V0 <- sigma.hat2* solve(t(X)%*%X)
meat <- crossprod(e.hat*X)
bread <- solve(crossprod(X))
V <- (bread %*% meat %*% bread) *(N/(N-2))
cbind(V, robust)
```

```
##                                          (Intercept)     salepric
## (Intercept)  2.089289e-05 -0.000164845  2.089289e-05 -0.000164845
## salepric    -1.648450e-04  0.001375536 -1.648450e-04  0.001375536
```

```
A <- diag(2)
b <- c(0,1)
(t(A%*%b.hat - b) %*% solve(A %*% V0 %*% t(A)) %*% (A%*%b.hat-b))/2
```

```
##          [,1]
## [1,] 154.3308
```

```
t(A%*%b.hat - b) %*% solve(A %*% V0 %*% t(A)) %*% (A%*%b.hat-b)
```

```
##          [,1]
## [1,] 308.6616
```

```
t(A%*%b.hat - b) %*% solve(A %*% V %*% t(A)) %*% (A%*%b.hat-b)
```

```
##          [,1]
## [1,] 39.33002
```

```
#Both the F and Chi2 are positive only so
## we want just 5% in the top tail
qf(.95, df1=2, df2=N-2)
```

```
## [1] 3.003943
```

```
qchisq(.95, df=2)
```

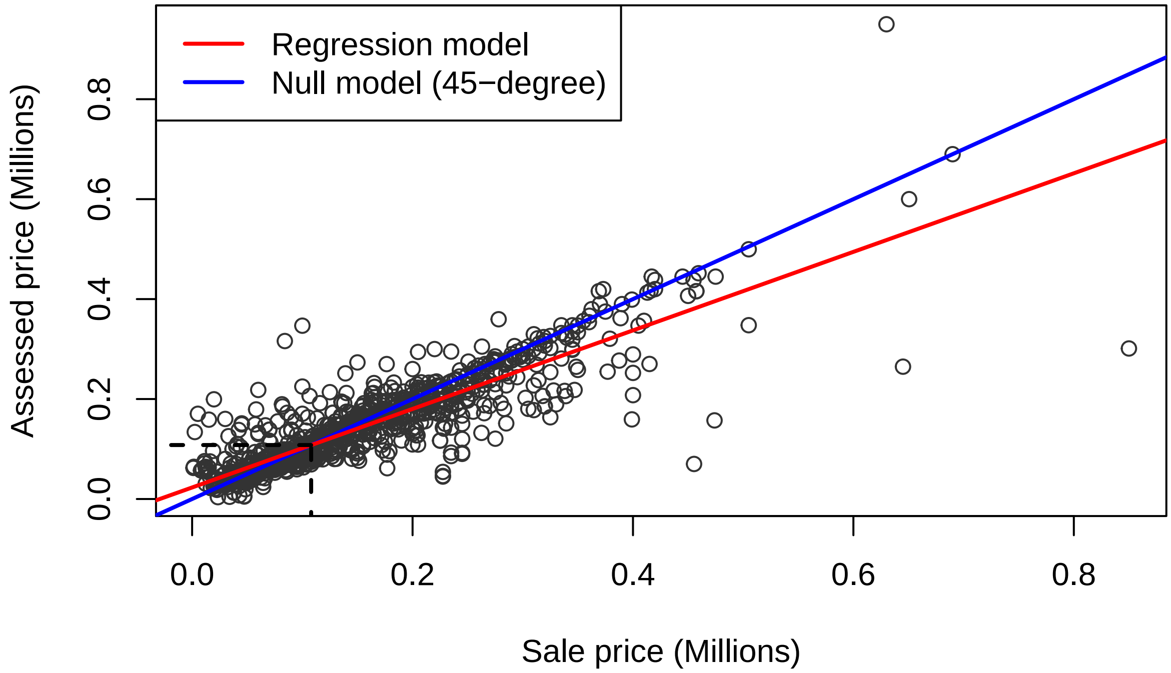
```
## [1] 5.991465
```

So far, we've seen strong evidence against the null model that $A_i = S_i$. We can explore this a little further. First of all note that, because the slope is less than one, appraisals are increasing less than a dollar every time actual value increases by a dollar. Specifically, we see that for every million dollar increase in "real" value there is only a $790,000 increase in appraisal, on average. Likewise, a there is a floor on appraisals. A home with a true value of 0 would be appraised at an average of about $23,000. This tells us that low-valued homes are over-assessed on average, but high value homes will be under-assessed. we can find the inflection point by setting $\mathrm{E}[A^*|S^*] = \beta_0 + \beta_1 S^* = S^*$ and solving for S^* to get

$$\hat{S}^* = 0.023/(1 - 0.786) = 0.108.$$

Homes with a true value of about $108,000 are accurately assessed on averaged. Homes with a value less than that are over appraised, while those above are under appraised, based on the fitted model. We can consider that visually too,

```r
plot(asprice~salepric,data=shady,
     col="grey20",
     xlab="Sale price (Millions)",
     ylab="Assessed price (Millions)")
abline(a=0,b=1, col="blue",lwd=2)
abline(model1, col="red",lwd=2);
segments(x0=c(0.108,0.108),x1 =c(0.108,-0.05),
         y0=c(-0.05,0.108), y1=c(0.108,0.108), lwd=2,lty = 2)
legend(x="topleft", legend=c("Regression model", "Null model (45-degree)"),
       col=c("red", "blue"), lty=1,lwd=2)
```



We can also apply the delta method to build a confidence interval around that inflection point. Note that S^* is a nonlinear function of parameters $S^* = C(\beta) = \frac{\beta_0}{1-\beta_1}$. To use the delta method we need the gradient w.r.t to β.

$$D_\beta C(\beta) = \left(\frac{1}{1-\beta_1}, \frac{\beta_0}{(1-\beta_1)^2} \right).$$

The standard error of $\hat{S}^*$ then becomes

$$s.e.(\hat{S}^*) = \sqrt{D_\beta C(\hat{\beta}) \hat{V} D_\beta C(\hat{\beta})'},$$

where $\hat{V}$ is either the robust or classic variance matrix. Let's stick with the robust for this example since we have a large N and it makes fewer assumptions. Let's compute our standard error first and then find the interval.

```
DC <- c(1/(1-model1$coef['salepric']),
        model1$coef['(Intercept)']/(1-model1$coef['salepric'])^2)
delta.se <- drop(sqrt(DC %*% robust %*% DC))
```

Our confidence interval under the delta method is normal so we use the z value of 1.96 to build a 95% confidence interval

```
s.star <- model1$coef['(Intercept)']/(1-model1$coef['salepric'])
lo <- s.star-1.96*delta.se
hi <- s.star+1.96*delta.se
c(lo, hi)
```

```
## (Intercept) (Intercept)
##   0.0970738   0.1182691
```

```
# Using the car package
# deltaMethod used to have a problem with
# (Intercept), so make sure you have the most
# recent version
deltaMethod(model1,
            g. = "(Intercept)/(1-salepric)",
            vcov. = robust)
```

```
##                           Estimate       SE    2.5 % 97.5 %
## Intercept/(1 - salepric) 0.107672 0.005407 0.097074 0.1183
```

How do we interpret this? We are 95% confident that the true home value where appraisals are accurate is between \$97,074 and \$118,269. Remember that 95% confidence means confidence in the procedure. 95% of 95% confidence intervals contain the true but unknown value of S^*. If we were to repeat this whole experiment many, many times, the proportion of intervals containing the true value would tend to 95%.

5 Specification concerns and extensions

5.1 Model fit measures

The linear model we fit to the data is a model. It's an approximation. Often times we want to know whether one approximation is better or worse than another. This is the job of model fit. Note that $\hat{y}_i = \hat{\beta}' x_i$, $\hat{\varepsilon}_i = y_i - \hat{y}_i$, and as such $y_i = \hat{y}_i + \hat{\varepsilon}_i$. This allows us to break the total sum of squares into two parts:

$$\begin{aligned}
y'y &= (\hat{y} + \hat{\varepsilon})'(\hat{y} + \hat{\varepsilon}) \\
&= \hat{y}'\hat{y} + \hat{\varepsilon}'\hat{\varepsilon} + 2\hat{y}'\hat{\varepsilon} \\
&= \hat{y}'\hat{y} + \hat{\varepsilon}'\hat{\varepsilon} + 2\hat{\beta}'X'\hat{\varepsilon} \\
&= \hat{y}'\hat{y} + \hat{\varepsilon}'\hat{\varepsilon}
\end{aligned}$$

This formation allows us to form what's called the "uncentered" R^2,

$$R^2_{\text{uncentered}} = 1 - \frac{\hat{\varepsilon}'\hat{\varepsilon}}{y'y}.$$

This is called the uncentered version because it measures the variation in y around 0, we can "center" the measure by measuring things around the mean of y.

$$y'y - N\bar{y}^2 = \hat{y}'\hat{y} - N\bar{y}^2 + \hat{\varepsilon}'\hat{\varepsilon}.$$

This first (two) term is now the "centered" explained sum of squares and as always the last term is the residual sum of squares. The centered R^2 is now

$$R^2 = 1 - \frac{\hat{\varepsilon}'\hat{\varepsilon}}{y'y - N\bar{y}^2}.$$

As we saw before R^2 is a measure of distance between points and lines. It is sometimes referred to as the "variance [in y] explained [by X]." However, it has some limitations as a measure of model fit. First and foremost R^2 always increases when you add a new variable. To see this considered the partition data matrix $X = [X_1, X_2]$. Define a restricted estimator that forces β_2 to be 0 as

$$\hat{\beta}_R = \underset{\beta:\beta_2=0}{\text{argmin}}(y - X\beta)'(y - X\beta).$$

Let's see where this takes us

$$\begin{aligned}
\hat{\varepsilon}_U'\hat{\varepsilon}_U &= \min_{\beta}\left(y - X\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}\right)'\left(y - X\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}\right)\\
&\leq \min_{\beta}\left(y - X\begin{bmatrix}\beta_1\\ 0\end{bmatrix}\right)'\left(y - X\begin{bmatrix}\beta_1\\ 0\end{bmatrix}\right)\\
&= \hat{\varepsilon}_R'\hat{\varepsilon}_R\\
\hat{\varepsilon}_U'\hat{\varepsilon}_U &\leq \hat{\varepsilon}_R'\hat{\varepsilon}_R.
\end{aligned}$$

So the restricted estimator will always have a weakly **larger** RSS. So what? Well that means the amount of "variance explained" will be smaller because more is left in the residual.

$$R_U^2 = 1 - \frac{\hat{\varepsilon}_U'\hat{\varepsilon}_U}{y'y - N\bar{y}^2} \geq 1 - \frac{\hat{\varepsilon}_R'\hat{\varepsilon}_R}{y'y - N\bar{y}^2} = R_R^2.$$

Adding *any* variable to model will improve "fit." This may be troubling if we were interested in using R^2 for adjudicating among models. However, it's not the end of the world. Maybe we can adjust R^2 to penalize the addition of "bad" regressors. One attempt at this is called the "adjusted R^2" where instead of comparing sums of squares we look at the variance estimators

$$R_{adj.}^2 = 1 - \frac{\hat{\sigma}_\varepsilon^2}{s_y} = 1 - \frac{\hat{\varepsilon}'\hat{\varepsilon}/(N-K)}{(y'y - N\bar{y}^2)/(N-1)} = 1 - (1 - R^2)\left(\frac{N-1}{N-K}\right).$$

What's going on here? Well this adjusted measure still depends on the SSR and SST, but it also includes K. Each new variable will improve $\hat{\varepsilon}'\hat{\varepsilon}$, but if the improvement isn't by very much then the "penalty" of including K may cause the adjusted R^2 to decrease.

There are other model fit measures that weigh the relative fit gains against added complexity of a new parameter. Most commonly are the Akaike (Ah kai e kay) Information Criterion

$$AIC = N\log\left(\frac{\hat{\varepsilon}'\hat{\varepsilon}}{N}\right) + 2K$$

and the Schwarz (Bayesian) Information Criterion (BIC)

$$BIC = N\log\left(\frac{\hat{\varepsilon}'\hat{\varepsilon}}{N}\right) + K\log(N).$$

What are differences here? Both the AIC and BIC depend on $\hat{\varepsilon}'\hat{\varepsilon}$, but they impose different complexity penalties. Note, that R uses a slightly different formula for AIC and BIC, but the gist is unchanged. Notably the BIC more heavily penalizes new parameters for values of $N > 7$. Unlike adjusted R^2, lower values of the AIC and BIC are preferred. All three of these methods for comparative model fit are "better" than R^2 in the sense that they do not always increase with new variables, however the penalties are completely arbitrary. As such, it is not a great basis for variable selection (more on this later).

5.2 Restricted model testing

Another way to think about model fit and comparison is in terms of "restricted models." Above we thought about the comparing models based on specific fit statistics, now we're going to frame it in terms of comparative model testing. Unlike the above measures, these will be conducted in the framework of the Wald test for testing linear hypotheses on β. Further, this framework will be considering model fit within construct of **nested** model testing. A **nested** model is one that can be written as a "restricted" form of another "unrestricted" model. Specifically define the restricted OLS estimator as

$$\hat{\beta}_R = \underset{\beta : A\beta = b}{\operatorname{argmin}} (y - X\beta)'(y - X\beta).$$

This is a **constrained** minimization problem. Have you seen these?
Constrained optimization problems are solved using a method called "Lagrange multipliers." Notably we write the constrained minimization problem as an unconstrained problem

$$\begin{aligned} &\min f(x) \\ &s.t.\ g(x) = c \end{aligned}$$

can be written as

$$L(x, \lambda) = f(x) - \lambda g(x).$$

Here, λ are called "Lagrange Multipliers" and there are as many multipliers as there are constraints to satisfy. Minimizing the Lagrangian function L with respect to x solves the constrained problem.

Going back to our restricted OLS estimator, the Lagrangian is given by

$$L(\beta, \lambda) = (y - X\beta)'(y - X\beta) - \lambda'(A\beta - b).$$

We can take the first order conditions to find:

$$\begin{aligned}
D_\beta L(\beta,\lambda) &= -2X'y + 2X'X\hat{\beta}_R - A'\hat{\lambda} = 0 \\
D_\lambda L(\beta,\lambda) &= A\hat{\beta}_R - b = 0
\end{aligned}$$

solving the first condition for $\hat{\beta}_R$

$$\hat{\beta}_R = (X'X)^{-1}(X'y + \frac{1}{2}A'\hat{\lambda}).$$

This means we need to solve for λ. The FOC wrt to λ is the original constraint so we plug the estimate $\hat{\beta}_R$ into it.

$$\begin{aligned}
b &= A\left((X'X)^{-1}(X'y + \frac{1}{2}A'\hat{\lambda})\right) \\
&= A(X'X)^{-1}X'y + A(X'X)^{-1}\frac{1}{2}A'\hat{\lambda} \\
b - A(X'X)^{-1}X'y &= \frac{1}{2}A(X'X)^{-1}A'\hat{\lambda} \\
\hat{\lambda} &= 2\left(A(X'X)^{-1}A'\right)^{-1}(b - A(X'X)^{-1}X'y).
\end{aligned}$$

NOW, we've got something. We can plug this multiplier $\hat{\lambda}$ into the expression for $\hat{\beta}_R$ to get a value that only includes data and constraints.

$$\begin{aligned}
\hat{\beta}_R &= (X'X)^{-1}(X'y + A'\left(A(X'X)^{-1}A'\right)^{-1}(b - A(X'X)^{-1}X'y)) \\
&= \hat{\beta}_U - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b).
\end{aligned}$$

This expression represents the differences in the coefficients between the restricted and unrestricted models. We can use those to find the differences in model fit:

$$\begin{aligned}
\hat{\varepsilon}_R = y - X\hat{\beta}_R &= y - X\hat{\beta}_U + X(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b) \\
&= \hat{\varepsilon}_U + X(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b).
\end{aligned}$$

We can measure the increase in "model fit" in the unrestricted versus the restricted ($A\beta = b$)

as

$$\begin{aligned}
\hat{\varepsilon}_R'\hat{\varepsilon}_R &= (\hat{\varepsilon}_U + X(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b))' \\
&\quad (\hat{\varepsilon}_U + X(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b)) \\
&= \hat{\varepsilon}_U'\hat{\varepsilon}_U + 2(A\hat{\beta}_U - b)'A'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1}X'\hat{\varepsilon}_U \\
&\quad + (A\hat{\beta}_U - b)'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1}X'X(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b) \\
&= \hat{\varepsilon}_U'\hat{\varepsilon}_U + 2(A\hat{\beta}_U - b)'A'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1}0 \\
&\quad + (A\hat{\beta}_U - b)'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b) \\
&= \hat{\varepsilon}_U'\hat{\varepsilon}_U + (A\hat{\beta}_U - b)'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b) \\
\hat{\varepsilon}_R'\hat{\varepsilon}_R - \hat{\varepsilon}_U'\hat{\varepsilon}_U &= (A\hat{\beta}_U - b)'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b).
\end{aligned}$$

We can write the “gap” between the restricted and the unrestricted models as a null or restricted model

$$\hat{\varepsilon}_R'\hat{\varepsilon}_R - \hat{\varepsilon}_U'\hat{\varepsilon}_U = (A\hat{\beta}_U - b)'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b).$$

What does this look like? (Pause for effect). The F test we look at earlier for linear hypothesis testing! We can rewrite it

$$F = \frac{(A\hat{\beta}_U - b)'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b)/J}{\hat{\sigma}_\varepsilon^2} = \frac{(\hat{\varepsilon}_R'\hat{\varepsilon}_R - \hat{\varepsilon}_U'\hat{\varepsilon}_U)/J}{\hat{\varepsilon}_U'\hat{\varepsilon}_U/(N-K)} \sim F_{J,N-K}.$$

This means that our F tests can be written in terms of whether the amount of “improvement” in the unrestricted model is strong enough evidence against the null hypothesis restrictions. We can press on this “relative improvement” interpretation a little bit more.

Recall,

$$\begin{aligned}
R_U^2 &= 1 - \frac{\hat{\varepsilon}_U'\hat{\varepsilon}_U}{y'y - N\bar{y}^2} \\
R_R^2 &= 1 - \frac{\hat{\varepsilon}_R'\hat{\varepsilon}_R}{y'y - N\bar{y}^2}
\end{aligned}$$

Rewrite these to get

$$\begin{aligned}
\hat{\varepsilon}_U'\hat{\varepsilon}_U &= (y'y - N\bar{y}^2)(1 - R_U^2) \\
\hat{\varepsilon}_R'\hat{\varepsilon}_R &= (y'y - N\bar{y}^2)(1 - R_R^2),
\end{aligned}$$

Plug these into our F statistic.

$$F = \frac{((y'y - N\bar{y}^2)(1 - R_R^2) - (y'y - N\bar{y}^2)(1 - R_U^2))\,/J}{(y'y - N\bar{y}^2)(1 - R_U^2)/(N-K)} = \frac{(R_U^2 - R_R^2)/J}{(1 - R_U^2)/(N-K)} \sim F_{J,N-K}.$$

This means that we can consider restricted versus unrestricted models based on the relative improvements in R^2. In some cases, this will allow us to consider whether two regression models reported in a paper are statistically different from each other based on only the reported R^2 and samples. One special case we frequently consider is called an "omnibus test" where the null is $H_0 : \beta_1 = \beta_2 = \ldots = \beta_m = 0$. Here we're saying that all the included (non-constant) covariates have no effect on y. This is a comparison test of whether our imposed model has any explanatory power (R reports this as the `F-statistic` on the bottom line of the `summary` output. Under the null, there are no X's (except for the constant) so the amount of "variance explained" by X is? (answer: 0)

$$\begin{aligned}
R_R^2 &= 1 - \frac{\hat{\varepsilon}'_R \hat{\varepsilon}_R}{y'y - N\bar{y}^2} \\
&= 1 - \frac{(y - \mathbf{1}'\hat{\beta}_R)'(y - \mathbf{1}'\hat{\beta}_R)}{y'y - N\bar{y}^2}
\end{aligned}$$

What is $\hat{\beta}_R$ when there's only a constant? In this case $\mathrm{E}[y|X_{-0} = 0] = \mathrm{E}[y] = \bar{y}$.

$$\begin{aligned}
&= 1 - \frac{(y - \mathbf{1}'\bar{y})'(y - \mathbf{1}'\bar{y})}{y'y - N\bar{y}^2} \\
&= 1 - \frac{y'y - 2\mathbf{1}'y\bar{y} + \mathbf{1}'\mathbf{1}\bar{y}^2}{y'y - N\bar{y}^2} \\
&= 1 - \frac{y'y - 2\frac{N}{N}\mathbf{1}'y\bar{y} + \mathbf{1}'\mathbf{1}\bar{y}^2}{y'y - N\bar{y}^2} \\
&= 1 - \frac{y'y - 2N\bar{y}^2 + N\bar{y}^2}{y'y - N\bar{y}^2} \\
&= 1 - 1 = 0.
\end{aligned}$$

When the regression is only a constant, R^2 is zero. This make sense, how can X explain any variation when X is constant. Our F statistic becomes

$$F = \frac{(R_U^2 - R_R^2)/J}{(1 - R_U^2)/(N-K)} = \frac{(R^2)/(K-1)}{(1 - R^2)/(N-K)} \sim F_{K-1,N-K}.$$

This result means that we can conduct this test using only some very commonly reported summary statistics (i.e., without access to the underlying data). How neat is that?

5.2.1 Finite sample properties of restricted models (Advanced)

Another way to think about model fit and comparison within this nested framework is in the finite sample properties of the restricted and unrestricted estimator. Let's start with the variance of the two under homoskedasticity

$$
\begin{aligned}
\mathrm{Var}(\hat{\beta}_U|X) &= \sigma^2_\varepsilon (X'X)^{-1} \\
\mathrm{Var}(\hat{\beta}_R|X) &= \mathrm{Var}(\hat{\beta}_U - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b)) \\
&= \mathrm{Var}\left(\hat{\beta}_U - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\hat{\beta}_U\right. \\
&\qquad \left. +(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}b\right) \\
&= \mathrm{Var}\left(\left(I - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\right)\hat{\beta}_U\right. \\
&\qquad \left. +(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}b\right) \\
&= \left(I - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\right)\mathrm{Var}\left(\hat{\beta}_U\right) \\
&\qquad \left(I - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\right)' \\
&= \left(\mathrm{Var}\left(\hat{\beta}_U\right) - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\,\mathrm{Var}\left(\hat{\beta}_U\right)\right) \\
&\qquad \left(I - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\right)' \\
&= \mathrm{Var}\left(\hat{\beta}_U\right) - 2(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\,\mathrm{Var}\left(\hat{\beta}_U\right) \\
&\qquad + (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A\,\mathrm{Var}\left(\hat{\beta}_U\right)A'\left(A(X'X)^{-1}A'\right)^{-1}A'(X'X)^{-1} \\
&= \mathrm{Var}\left(\hat{\beta}_U\right) - 2\sigma^2_\varepsilon(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1} \\
&\qquad + \sigma^2_\varepsilon(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A'(X'X)^{-1} \\
&= \mathrm{Var}\left(\hat{\beta}_U\right) - \sigma^2_\varepsilon(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1},
\end{aligned}
$$

or put in comparative terms

$$
\mathrm{Var}(\hat{\beta}_U|X) - \mathrm{Var}(\hat{\beta}_R|X) = \sigma^2_\varepsilon(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1}.
$$

The right-hand side is symmetric and positive semidefinite (everything is in quadratic form). As such we conclude that the restricted estimator has a (weakly) lower variance than the unrestricted one. Imposing restrictions can improve an estimator's variance, what about bias and mean squared error?

Let's look at bias and assume that, as usual, $y = X\beta + \varepsilon$ reflects the truth and that the unrestricted model captures it correctly. What damage is done by fitting the restricted model? Note that we already know that the unrestricted model is unbiased $\mathrm{E}[\hat{\beta}_U|X] = \beta$. For the restricted model

$$\begin{aligned}
\mathrm{bias}(\hat{\beta}_R) &= \mathrm{E}[\hat{\beta}_R] - \beta \\
&= \mathrm{E}[\hat{\beta}_U - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\hat{\beta}_U - b)] - \beta \\
&= -(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\beta_U - b)
\end{aligned}$$

and the MSE gap

$$\begin{aligned}
\mathrm{MSE}(\hat{\beta}_U) - \mathrm{MSE}(\hat{\beta}_R) = {}& \sigma_\varepsilon^2(X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1} \\
& - (X'X)^{-1}A'\left(A(X'X)^{-1}A'\right)^{-1}(A\beta_U - b)(A\beta_U - b)'\left(A(X'X)^{-1}A'\right)^{-1}A(X'X)^{-1}
\end{aligned}$$

Under some values of b (which we choose) we can make the bias quite small. This means that it's possible to find a restricted estimator that is biased but has an overall lower mean squared error that the OLS estimator. Note that this does not violate the Gauss-Markov theorem, because that only considers the set of unbiased estimators.

5.3 Omitted and irrelevant variables

So far we've considered measures of model fit and some tests of comparative model testing. But we haven't really talked about the consequences of variable selection. We've hinted above that the bias and variance differences between the restricted and unrestricted models will exist, but let's explore these a little more thoroughly. Suppose that the true model is

$$y = X_1\beta_1 + X_2\beta_2 + \varepsilon,$$

but we only fit the restricted model that includes X_1. The OLS estimates of the restricted model are

$$\hat{\beta}_R = (X_1'X_1)^{-1}X_1'y.$$

We can rewrite this

$$\begin{aligned}\hat{\beta}_R &= (X_1'X_1)^{-1}X_1'(X_1\beta_1 + X_2\beta_2 + \varepsilon)\\ &= (X_1'X_1)^{-1}X_1'X_1\beta_1 + (X_1'X_1)^{-1}X_1'X_2\beta_2 + (X_1'X_1)^{-1}X_1'\varepsilon\\ &= \beta_1 + (X_1'X_1)^{-1}X_1'X_2\beta_2 + (X_1'X_1)^{-1}X_1'\varepsilon,\end{aligned}$$

which has an expected value of

$$\mathrm{E}[\hat{\beta}_R|X] = \beta_1 + (X_1'X_1)^{-1}X_1'X_2\beta_2.$$

As you can see this is **not** always equal to β_1. In this case the OLS estimator is unbiased only if $(X_1'X_1)^{-1}X_1'X_2\beta_2 = 0$. A few ways that we get unbiasedness in this case are

1. $\beta_2 = 0$, which is to say that the variables in X_2 are irrelevant
2. $X_1'X_2 = 0$, which is related to the correlation of the variables in X_1 and X_2.

Note that this right here is one big reason why we say that $\hat{\beta}$ reflects the "association" between X and y rather than the causal relationship. If we had the perfectly specified model there would be no omitted variable bias and we could interpret things causally. A lot of research uses the control variables to get as close as possible to a causal effect by trying different robustness checks (control specifications). However, the method of control variables is a very unrealistic way to identify a causal effect. Because of omitted factors, we frequently find ourselves working with correlation analysis to make a causal story and seeing "how robust" it is. But it's best to not use language about "cause" unless you have a more solid identification argument. We'll return to this later with instrumental variables, which offer a big leap towards causal identification.

Do p values and hypothesis tests help with causal versus correlation effects? Nah. Small p values reflect strong evidence against the null hypothesis **if all the assumptions behind the test are true**. Implicit in Assumption B1 is that we have the correct specification. Also, let's be clear, holding $\hat{\beta}$ fixed, as N increases the standard errors decrease, the test statistics increase, and the p-values decrease. So we can have an estimate of $\hat{\beta}_1 = 0.0000001$ and with large enough N we can use that to reject a null that the truth is $\beta_1 = 0$. Here is where you need to be clear on the difference between statistical and substantive significance. I can't help you much with that as it requires you to know what a substatively interesting finding is for your speific research question. For the most part small p values tell us that the true correlation is unlikely to be exactly 0 (or whatever we're testing), and then it's up to us to determine if the marginal effects are interesting.

Side bit aside, let's look at two variables and constant to better understand what omitted variable bias looks like. Let

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$$

represent the truth, but suppose we fit a model with only x_1. The OLS estimate is now given by the same simple OLS estimator we looked early on (covariance to variance)

$$\begin{aligned}
\hat{\beta}_{R1} &= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)(y_i-\bar{y})}{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)^2} \\
&= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)(\beta_1(x_{i1}-\bar{x}_1)+\beta_2(x_{i2}-\bar{x}_2)+(\varepsilon_i-\bar{\varepsilon}))}{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)^2} \\
&= \beta_1 + \beta_2\frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)(x_{i2}-\bar{x}_2)}{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)^2} + \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)(\varepsilon_i-\bar{\varepsilon})}{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)^2} \\
&= \beta_1 + \beta_2\frac{s_{x_1,x_2}}{s_{x_1}^2} + \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)(\varepsilon_i-\bar{\varepsilon})}{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)^2}.
\end{aligned}$$

The expected value of this quantity then becomes

$$\begin{aligned}
\mathrm{E}[\hat{\beta}_R|x_1,x_2] &= \beta_1 + \beta_2\frac{s_{x_1,x_2}}{s_{x_1}^2} + \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)E[(\varepsilon_i-\bar{\varepsilon})|x_1,x_2]}{\frac{1}{N-1}\sum_{i=1}^{N}(x_{i1}-\bar{x}_1)^2} \\
&= \beta_1 + \beta_2\frac{s_{x_1,x_2}}{s_{x_1}^2} \\
&= \beta_1 + \beta_2\frac{s_{x_2}}{s_{x_1}}r_{x_1,x_2}.
\end{aligned}$$

When is the bias 0? When $\beta_2 = 0$ (i.e., x_2 is irrelevant), x_1 and x_2 are uncorrelated, or $s_{x_2} = 0$ (the omitted variable is a constant). The direction of the bias is signable if you have an expectation about the omitted variable. If x_1 and x_2 both have a positive effect on y then omitting x_2 will bias $\hat{\beta}_{R1}$ downwards if the correlation between them is negative and vice-versa. When there is more than one relevant omitted variable, this effort at signing the bias becomes overwhelming.

Okay, so what's the harm in just including more and more variables? If the truth is

$$y = X_1\beta_1 + \varepsilon = X[\beta_1, 0],$$

what happens if we include these extra variables with true parameters 0 as part of an unrestricted model? The OLS estimates are

$$\hat{\beta} = (X'X)^{-1}X'y = (X'X)^{-1}X'(X[\beta,0]+\varepsilon) = [\beta,0] + (X'X)^{-1}X'\varepsilon.$$

The expected value checks out

$$\mathrm{E}[\hat{\beta}|X] = [\beta, 0],$$

which matches the truth. As such, we don't induce bias by including irrelevant variables. However, the variance of the restricted estimator is (derived above)

$$\mathrm{Var}(\hat{\beta}_U|X) - \mathrm{Var}(\hat{\beta}_R|X) = \sigma_\varepsilon^2 (X'X)^{-1} A' \left(A(X'X)^{-1} A' \right)^{-1} A(X'X)^{-1}.$$

This expression is positive semidefinite (all in quadratic on the RHS). Thus,

$$z' \mathrm{Var}(\hat{\beta}_U|X) z \geq z' \mathrm{Var}(\hat{\beta}_R|X) z, \text{ for any } z \neq 0.$$

Including variables weakly raises the variance in the fitted model. The consequences are that excluding relevant variables will result in biased (and inconsistent) OLS estimates, but the variance (and possibly the MSE) will be lower than OLS with the correct variables. Extra variance is the main consequence of "kitchen sink" modeling.

The third thing we should think about though is how do the estimated standard errors and variance matrices change when we add new variables? For this exploration let's focus on the following expression for the variance of $\hat{\beta}_j$ given X

$$\mathrm{Var}(\hat{\beta}_j|X) = \frac{\sigma_\varepsilon^2}{\left(1 - R_j^2\right) \sum_{i=1}^N (x_{ij} - \bar{x}_j)}.$$

and the estimated version

$$\widehat{\mathrm{Var}}(\hat{\beta}_j|X) = \frac{\hat{\sigma}_\varepsilon^2}{\left(1 - R_j^2\right) \sum_{i=1}^N (x_{ij} - \bar{x}_j)}.$$

Note that the difference here is that σ_ε^2 is unchanging across models but $\hat{\sigma}_\varepsilon^2$ is changing because the model won't condition on the unincluded X. This will bias our standard errors upward because $\hat{\sigma}_\varepsilon^2$ will be including the variance of the excluded variables in the data generating process

Situation	$\mathrm{Var}(\hat{\beta}_j|X)$	$\widehat{\mathrm{Var}}(\hat{\beta}_j|X)$
Add a relevant but unrelated variable	Increases	Either direction
Add a relevant but related variable	Increases	Either direction
Add an irrelevant variable	Increases	Increases

What have we learned from this stroll through model fit?

1. Adding more variables always improves R^2

2. You can use adjusted R^2, AIC, or BIC to penalize complexity and these work for comparative model fit (not necessarily nested either), but

   - No testing
   - Arbitrary penalties
   - No critical values to tell us better one model should be to say it is better

3. Restricted model testing works for comparing nested models and can be done using R^2. Can be thought of as a test based on how much more R^2 is needed to tell models apart.

4. Making increasingly general models by adding many variables will raise the overall variance (estimates can end far away from truth).

5. Leaving out relevant (effect both main treatment and outcome) leads to bias

6. Goal is to include enough controls to get a "good" estimate for your main parameter of interest

7. Current best research practice tends to be to have one or two parameters of interest and choose covariates such that you reduce OVB, but without throwing in the "kitchen sink" (or garbage can). Sometimes a fine line. Reading helps. Do whatever reviewers tell you.

5.3.1 Application

```r
library(readstata13)
library(stargazer)
library(car)
library(lmtest)

presdata <- read.dta13("Rcode/datasets/presvote.dta")
presdata$IncShare[33:34] <- c(0.4631,0.5196 ) #add 2008, 2012
presdata$IncShare <- presdata$IncShare*100



model1 <- lm(IncShare~App, data=presdata)
```

```r
model2 <- lm(IncShare~App+Growth, data=presdata)
model3 <- lm(IncShare~App+Growth+Inflation, data=presdata)
stargazer(model1,model2,model3,no.space = TRUE, label='tab:example3',
          header=FALSE, keep.stat = c("n", "rsq", "adj.rsq", "F"),
          type='latex',
          covariate.labels = c("Approval", "Growth", "Inflation", "Intercept"),
          dep.var.labels = "Vote share")
```

Table 5

		Dependent variable:	
		Vote share	
	(1)	(2)	(3)
Approval	0.226^{**}	0.235^{***}	0.240^{***}
	(0.078)	(0.059)	(0.063)
Growth		1.203^{***}	1.221^{***}
		(0.323)	(0.340)
Inflation			0.098
			(0.383)
Intercept	40.408^{***}	36.729^{***}	36.151^{***}
	(4.238)	(3.344)	(4.125)
Observations	19	19	19
R^2	0.330	0.641	0.643
Adjusted R^2	0.290	0.597	0.571
F Statistic	8.354^{**} (df = 1; 17)	14.305^{***} (df = 2; 16)	9.001^{***} (df = 3; 15)
Note:			*p<0.1; **p<0.05; ***p<0.01

Let's go back to our regression model for predicting incumbent party vote share in U.S. presidential elections, and we'll consider three models using approval, growth, and inflation. Notice that when we move from Model 2 to 3 (adding inflation), the adjusted R^2 decreases. What happens to our other measures of model fit?

```r
c(AIC(model2), AIC(model3))
```

```
## [1] 103.8899 105.8079
```

```r
c(BIC(model2), BIC(model3))
```

```
## [1] 107.6677 110.5301
```

The AIC and the BIC both increase (closer to 0). All three of these measures thus indicate that adding inflation does not improve model fit enough to justify using the extra parameter. We can test the hypothesis that the effects of approval, growth, and inflation are all 0.

$$\begin{aligned} H_0 :& \beta_{\text{approval}} = \beta_{\text{growth}} = \beta_{\text{inflation}} = 0 \\ H_A :& \text{ Not } H_0 \end{aligned}$$

Recall that this test is an F test based on the R^2 of model 3.

$$F = \frac{R^2/(K-1)}{(1-R^2)/(N-K)} = \frac{0.643/3}{(1-0.643)/(19-4)} = 9.001.$$

Notice that this matches the F statistic reported for model 3. We want to know if the probability of observing a test statistic as or more extreme than 9.001.

```
r2 <- summary(model3)$r.sq
F.stat <- (r2/(4-1)) / ((1-r2)/(19-4))
F.stat
```

```
## [1] 9.00095
```

```
pf(F.stat, df1=3,df2=19-4,lower.tail = FALSE)
```

```
## [1] 0.001187221
```

The probability of observing an F statistic as or more extreme than 9.001 if the null hypothesis is true, is $p = 0.001 < 0.05$; this is strong enough evidence to reject the null hypothesis. We can also find this using the linear hypothesis functions

```
linearHypothesis(model3, c("App=0", "Growth=0", "Inflation=0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## App = 0
## Growth = 0
## Inflation = 0
##
## Model 1: restricted model
## Model 2: IncShare ~ App + Growth + Inflation
##
```

```
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     18 482.40
## 2     15 172.27  3    310.13 9.0009 0.001187 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can also consider the joint hypothesis that the effects of both growth and inflation are equal to 0.

$$H_0 : \beta_{\text{growth}} = \beta_{\text{inflation}} = 0$$
$$H_A : \text{Not } H_0$$

This is still an F test and this is also a restricted model test where the restrictions are given by the null hypothesis. Note that this is a comparison test between models 1 and 3. The test statistic is again an F test

$$F = \frac{(R^2_U - R^2_R)/J}{(1 - R^2_U)/(N - K)} = \frac{(0.643 - 0.330)/2}{(1 - 0.643)/(19 - 4)} = 6.576.$$

We can use this to find a p value

```
r2.Res <- summary(model1)$r.sq
F.stat <- ((r2-r2.Res)/2)/((1-r2)/(19-4))
F.stat #rounding differences
```

```
## [1] 6.581363
```

```
pf(F.stat, df1=2, df2=19-4,lower.tail = F)
```

```
## [1] 0.008874099
```

```
## two identical versions of this test ##
linearHypothesis(model3, c("Growth=0", "Inflation=0")) #regular F test
```

```
## Linear hypothesis test
##
## Hypothesis:
## Growth = 0
## Inflation = 0
##
## Model 1: restricted model
## Model 2: IncShare ~ App + Growth + Inflation
```

```
##
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     17 323.45
## 2     15 172.27  2    151.17 6.5814 0.008874 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
waldtest(model3, model1) #F test using R^2
```

```
## Wald test
##
## Model 1: IncShare ~ App + Growth + Inflation
## Model 2: IncShare ~ App
##   Res.Df Df      F   Pr(>F)
## 1     15
## 2     17 -2 6.5814 0.008874 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

which is less than 0.05. We can reject the null hypothesis that both growth and inflation have no effect on incumbent vote share.

We can also consider a simulation to explore the three factors of interest: bias in $\hat{\beta}$, true variance in $\hat{\beta}$, and bias in the standard errors of $\hat{\beta}$.

```
### Omitted and irrelevant Variables ####
set.seed(1)
N <- 50
SIGMA <- matrix(c(1,.7,0,.8,
                  .7,1,0,.8,
                  0,0,1, -.45,
                  .8, .8,-.45,1), nrow=4)
X <- replicate(4, rnorm(N))
X <- t(t(chol(SIGMA) ) %*% t(X))
cor(X)
```

```
##             [,1]       [,2]        [,3]      [,4]
## [1,] 1.00000000 0.62596655  0.02628723  0.7487934
## [2,] 0.62596655 1.00000000  0.02002898  0.7848929
## [3,] 0.02628723 0.02002898  1.00000000 -0.4362338
## [4,] 0.74879342 0.78489292 -0.43623377  1.0000000
```

```
sigma2 <- 4
eps <- rnorm(N, sd=sqrt(sigma2))
beta <- c(-1, 1.5,-3,4, 0)
X <- cbind(1, X)
y <- X %*% beta + eps
cor(cbind(y,X[,-1]))
```

```
##             [,1]        [,2]        [,3]        [,4]       [,5]
## [1,]  1.00000000 -0.06022011 -0.35314090  0.79257259 -0.5867707
## [2,] -0.06022011  1.00000000  0.62596655  0.02628723  0.7487934
## [3,] -0.35314090  0.62596655  1.00000000  0.02002898  0.7848929
## [4,]  0.79257259  0.02628723  0.02002898  1.00000000 -0.4362338
## [5,] -0.58677072  0.74879342  0.78489292 -0.43623377  1.0000000
```

```
# What the model is estimating instead of sigma_epsilon
# sqrt( var(epsilon) + var(X2)*beta2^2 + var(X3)*beta3^2 + var(X4)*beta4^2
#         + 2*cov(X2, X3)*beta2*beta3+ 2*cov(X3, X4)*beta3*beta4)
# sqrt( var(epsilon) + var(X3)*beta3^2 + var(X4)*beta4^2 + 2*cov(X3, X4)*beta3*beta4)
# sqrt( var(epsilon) + var(X4)*beta4^2 )
# sqrt( variance of epsilon )
sigma.1 <- sqrt(29)
sigma.2 <-  sqrt(20)
sigma.3 <- 2
sigma.4 <- 2

model1 <- lm(y~X[,2])
model2 <- lm(y~X[,2:3])
model3 <- lm(y~X[,2:4])
model4 <- lm(y~X[,-1])
summary(model1) #biased beta and sigma^2
```

```
##
## Call:
## lm(formula = y ~ X[, 2])
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3917 -3.6476 -0.0597  2.9386 10.8527
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.9488     0.6995  -2.786  0.00762 **
## X[, 2]       -0.3526     0.8436  -0.418  0.67783
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.91 on 48 degrees of freedom
## Multiple R-squared:  0.003626,   Adjusted R-squared:  -0.01713
## F-statistic: 0.1747 on 1 and 48 DF,  p-value: 0.6778
```

```
summary(model2) #unbiased beta, smaller standard errors, biased sigma^2
```

```
##
## Call:
## lm(formula = y ~ X[, 2:3])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4020 -3.1885 -0.3253  2.0340 12.6776
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.7008     0.6514  -2.611  0.01207 *
## X[, 2:3]1     1.5485     0.9994   1.549  0.12800
## X[, 2:3]2    -2.8482     0.9373  -3.039  0.00387 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.536 on 47 degrees of freedom
## Multiple R-squared:  0.1672, Adjusted R-squared:  0.1318
## F-statistic: 4.719 on 2 and 47 DF,  p-value: 0.01356
```

```r
summary(model3) #unbiased beta, even smaller standard errors, unbiased sigma^2
```

```
## 
## Call:
## lm(formula = y ~ X[, 2:4])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -5.551 -1.466 -0.099  1.160  4.324 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -1.0291     0.3263  -3.153  0.00284 ** 
## X[, 2:4]1     1.4430     0.4935   2.924  0.00535 ** 
## X[, 2:4]2    -2.8739     0.4628  -6.210 1.40e-07 ***
## X[, 2:4]3     4.3099     0.3557  12.117 6.46e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.239 on 46 degrees of freedom
## Multiple R-squared:  0.8013, Adjusted R-squared:  0.7884 
## F-statistic: 61.84 on 3 and 46 DF,  p-value: 3.562e-16
```

```r
summary(model4) #unbiased beta, larger standard error, unbiased sigma^2
```

```
## 
## Call:
## lm(formula = y ~ X[, -1])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -5.304 -1.328 -0.107  1.419  3.830 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -0.9887     0.3319  -2.979  0.00465 ** 
## X[, -1]1      1.9504     0.8225   2.371  0.02207 *  
## X[, -1]2     -2.3006     0.8753  -2.629  0.01169 *
## X[, -1]3      3.8147     0.7335   5.201  4.7e-06 ***
## X[, -1]4     -1.2473     1.6136  -0.773  0.44357
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.249 on 45 degrees of freedom
## Multiple R-squared:  0.8039, Adjusted R-squared:  0.7865
## F-statistic: 46.13 on 4 and 45 DF,  p-value: 2.291e-15
```

```r
### The "true biased" variance###
# These matrices are what the estimator thinks the variance conditional on X
# will be given that you've misspecified the model.
# They are biased because the model is not conditioning on the
# X that are left in the error term, which biases the estimate of sigma^2
V1.hat <- sigma.1^2 * solve(t(X[,1:2]) %*% X[,1:2])
V2.hat <- sigma.2^2 * solve(t(X[,1:3]) %*% X[,1:3])
V3.hat <- sigma.3^2 * solve(t(X[,1:4]) %*% X[,1:4])
V4.hat <- sigma.4^2 * solve(t(X) %*% X)

#semi pd: add "irrelevant" (hat sigma unchanged, R^2_j increased)
round(eigen(V4.hat-rbind(cbind(V3.hat,0),0), only.values = TRUE)$value,7)
```

```
## [1] 3.160439 0.000000 0.000000 0.000000 0.000000
```

```r
#not clear; we've added a "relevant variable" (hat sigma decreased, R^2_j increased)
round(eigen(V3.hat-rbind(cbind(V2.hat,0),0), only.values = TRUE)$value,7)
```

```
## [1]  0.1014938 -0.2428657 -0.3553703 -1.1897005
```

```r
#not clear; we've added a "relevant variable" (hat sigma decreased, R^2_j increased)
round(eigen(V2.hat-rbind(cbind(V1.hat,0),0), only.values = TRUE)$value,7)
```

```
## [1]  1.1713685 -0.1574934 -0.2214032
```

```r
## True variance conditional on X ##
V1 <- sigma2 * solve(t(X[,1:2]) %*% X[,1:2])
V2 <- sigma2 * solve(t(X[,1:3]) %*% X[,1:3])
V3 <- sigma2 * solve(t(X[,1:4]) %*% X[,1:4])
V4 <- sigma2 * solve(t(X) %*% X)
```

```r
#These will all be positive semi-definite
round(eigen(V4-rbind(cbind(V3,0),0), only.values = TRUE)$value,7)
```

```
## [1] 3.160439 0.000000 0.000000 0.000000 0.000000
```

```r
round(eigen(V3-rbind(cbind(V2,0),0), only.values = TRUE)$value,7)
```

```
## [1] 0.1034267 0.0000000 0.0000000 0.0000000
```

```r
round(eigen(V2-rbind(cbind(V1,0),0), only.values = TRUE)$value,7)
```

```
## [1] 0.2481757 0.0000000 0.0000000
```

```r
MCout <- matrix(0, 1000,2)
for(i in 1:1000){
  eps <- rnorm(N, sd=sqrt(sigma2))
  y <- X %*% beta + eps
  MCout[i,] <- lm(y~X[,2])$coef
}
var(MCout)
```

```
##             [,1]        [,2]
## [1,]  0.07796720 -0.01481009
## [2,] -0.01481009  0.12155042
```

```r
V1
```

```
##             [,1]        [,2]
## [1,]  0.08119161 -0.01186295
## [2,] -0.01186295  0.11810009
```

```r
V1.hat
```

```
##             [,1]        [,2]
## [1,]  0.58863919 -0.08600639
## [2,] -0.08600639  0.85622562
```

```r
vcov(model1)
```

```
##             (Intercept)      X[, 2]
## (Intercept)  0.48926093 -0.07148618
## X[, 2]      -0.07148618  0.71167151
```

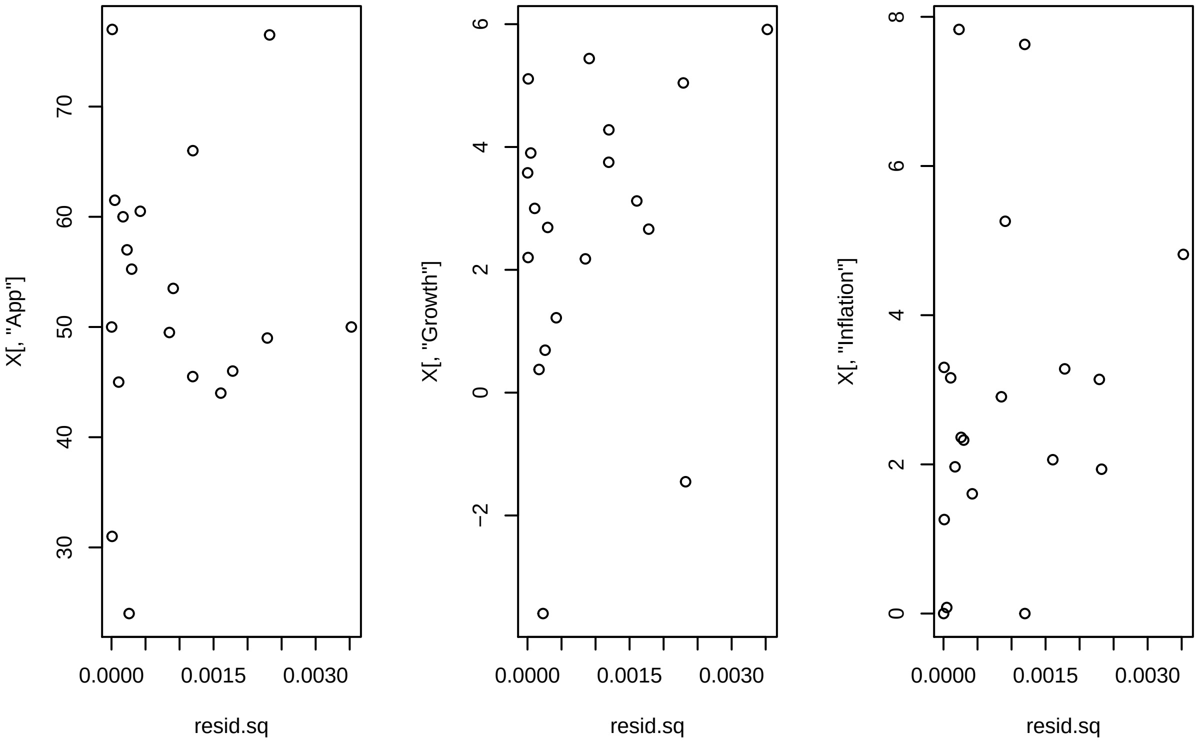
5.4 Heteroskedasticity and Normality assumptions

So far we've used the homoskedasticity and normality assumptions more often than not. However, we would like to be able to know how good of an assumption these are in any given situation.

5.4.1 Tests for heteroskedasticity

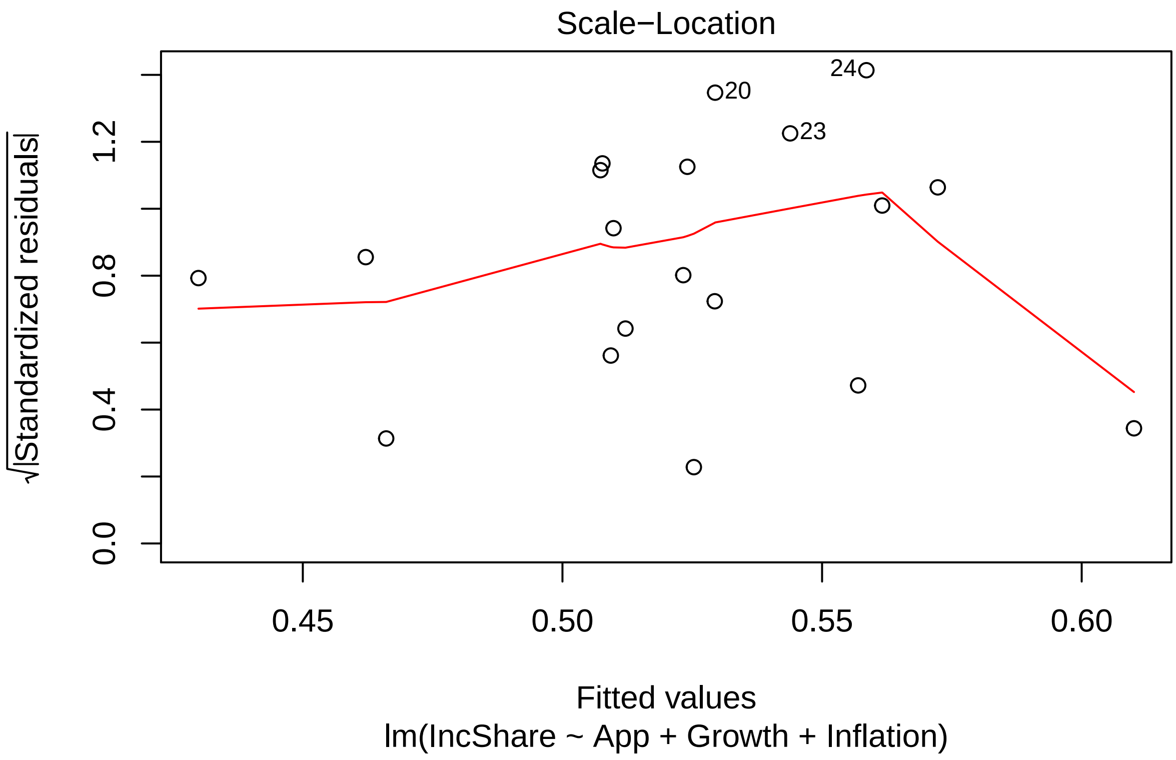
There are both heuristic and formal diagnostics for heteroskedasticity. One of the easier approaches involve looking at plots of squared residuals. Specifically, we can start by just looking at plots of each covariate against the squared residuals. Under homoskedasticity, these should be flat-ish lines.

```r
library(readstata13)
library(lmtest) #for bptest and coeftest
library(stargazer)
library(sandwich)
library(moments)

presdata <- read.dta13("Rcode/datasets/presvote.dta")
presdata$IncShare[33:34] <- c(0.4631,0.5196 )
model3 <- lm(IncShare~App+Growth+Inflation, data=presdata, x=T)
resid.sq <- model3$residuals^2
X <- model3$x #avoids any concerns about NAs

#Visually inspect for relationships between X and variance
par(mfrow=c(1,3))
plot(X[,"App"]~resid.sq)
plot(X[,"Growth"]~resid.sq)
plot(X[,"Inflation"]~resid.sq)
```



There is also a built in diagnostic plot for `lm` objects. The `plot` function applied to an `lm` object produces 6 diagnostic plot and presents 4 of them; the third relates to homoskedasticity. Under homoskedasticity this line should be flat.

```
par(mfrow=c(1,1))
plot(model3, which=3)
```



So we can start to see a little more clearly now that there may be some bouncing around here, but enough? Are these slopes distinguishable from zero? While visuals are helpful, a formal test can give us a more clear answer sometimes.

The Breusch-Pagan test provides a framework for testing The hypothesis test

$$H_0 : \mathrm{E}[\varepsilon_i^2|x_i] = \sigma^2, \forall i$$
$$H_A : \mathrm{E}[\varepsilon_i^2|x_i] = \sigma_\varepsilon^2 + \gamma' z_i.$$

Note that this involves testing a specific functional form assumption on z. The test is implemented using a regression of $\hat{\varepsilon}_i^2$ on z_i with a constant and then conducting an omnibus test. This test is either the F test we used last time or $NR^2 \overset{asy}{\sim} \chi_P^2$ where P is the length of z_i (excluding the constant). Why does an omnibus test tell us what we want to know?

The White test is a specific version of this test that uses x_i, x_{i1}^2, $\ldots$, x_{im}^2, $\{x_{ik}x_{i\ell}\}_{k=2,\ell>k}^m$ That is all two way multiplications within x_i including with itself.

```
###### Hypothesis testing #####
## H0: E[e^2_i|X] = sigma^2_e for all i
## HA: E[e^2_i|X] = sigma^2_e + \gamma'z_i for some i
```

```r
# This is a test of if all \gamma=0: An omnibus test on
#  e^2 ~ sigma^2_e + \gamma'z_i
# We don't have e, so we use hat(e)



#bptest is in the lmtest package
bptest(model3) # equivalent to varformula = ~App+Growth+Inflation
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model3
## BP = 3.1902, df = 3, p-value = 0.3632
```

```r
bptest(model3, #Using the White specification
       varformula = ~poly(App,Growth,Inflation,degree=2, raw=TRUE),
       data=presdata)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model3
## BP = 12.717, df = 9, p-value = 0.1758
```

```r
# This is equiv to
# ~App+Growth+Inflation+
#          I(App^2)+I(Growth^2) + I(Inflation^2)+
#          App:Growth + App:Inflation+
#          Growth:Inflation

#white test by hand
model3.white <-  lm(I(model3$residuals^2)~poly(App,Growth,Inflation,degree=2, raw=TRUE),
                    data=model3$model)
summary(model3.white)
```

```
##
```

```
## Call:
## lm(formula = I(model3$residuals^2) ~ poly(App, Growth, Inflation, 
##     degree = 2, raw = TRUE), data = model3$model)
## 
## Residuals:
##        Min         1Q     Median         3Q        Max 
## -1.086e-03 -3.835e-04  1.090e-06  4.032e-04  1.005e-03 
## 
## Coefficients:
##                                                              Estimate Std. Error
## (Intercept)                                                -6.527e-03  5.833e-03
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)1.0.0  1.967e-04  2.121e-04
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)2.0.0 -1.100e-06  1.807e-06
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.1.0  1.814e-04  1.292e-03
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)1.1.0 -1.499e-05  1.298e-05
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.2.0  1.357e-04  8.204e-05
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.0.1  1.900e-03  1.696e-03
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)1.0.1 -3.320e-05  3.033e-05
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.1.1  4.709e-05  6.674e-05
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.0.2 -6.119e-05  5.092e-05
##                                                            t value Pr(>|t|)
## (Intercept)                                                 -1.119    0.292
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)1.0.0   0.927    0.378
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)2.0.0  -0.609    0.558
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.1.0   0.140    0.891
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)1.1.0  -1.154    0.278
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.2.0   1.654    0.133
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.0.1   1.120    0.292
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)1.0.1  -1.095    0.302
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.1.1   0.706    0.498
## poly(App, Growth, Inflation, degree = 2, raw = TRUE)0.0.2  -1.202    0.260
## 
## Residual standard error: 0.0008139 on 9 degrees of freedom
## Multiple R-squared:  0.6693, Adjusted R-squared:  0.3386 
## F-statistic: 2.024 on 9 and 9 DF,  p-value: 0.1542
```

```
white <- 19*summary(model3.white)$r.sq
white
```

```
## [1] 12.71691
```

```
df <- length(model3.white$coef)-1
pchisq(white, df=df, lower=F)
```

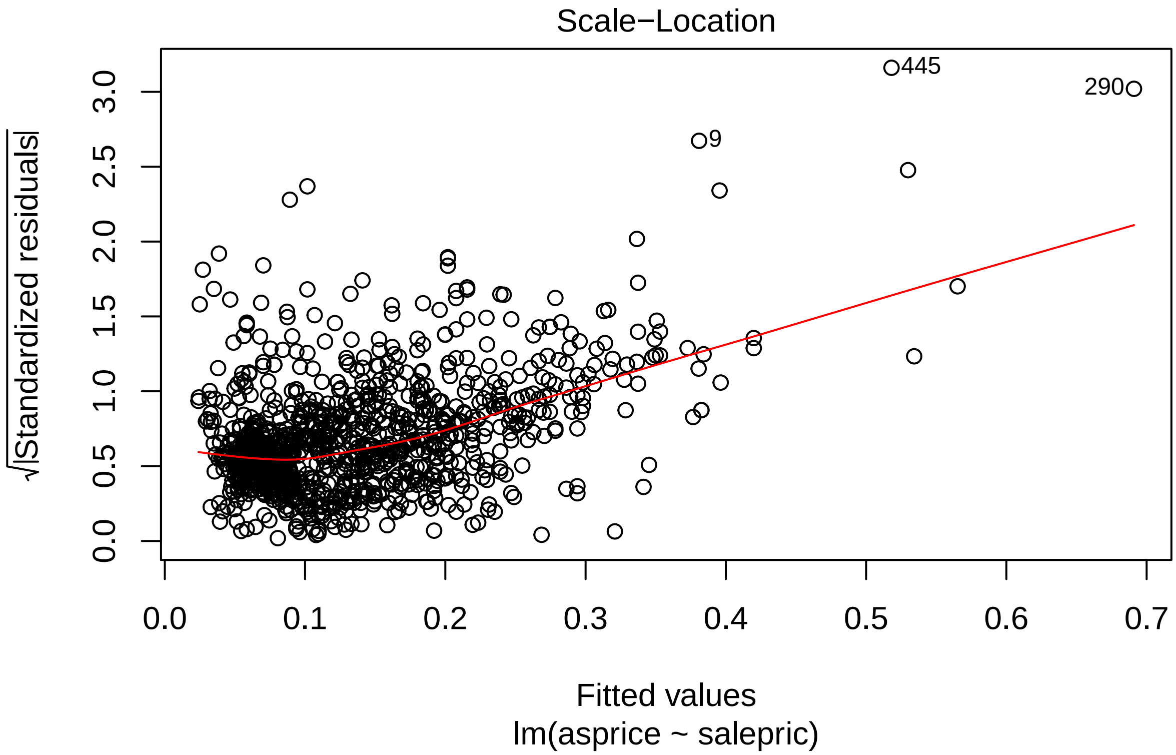
```
## [1] 0.1758398
```

We do not reject the null of homoskedasticity is good for us, because this is a small sample. We would be in trouble if we had found stronger evidence of heteroskedasticity because with only 19 observations it would be a reach to appeal to asymptotic results. We would have to change the model in some way to remove the heteroskedasticity.

```
#### Another Application
shady <- read.dta13("Rcode/datasets/shady.dta")
print(cbind(colnames(shady), attributes(shady)$var.labels))
```

```
##       [,1]       [,2]
##  [1,] "obs"      "observation number"
##  [2,] "aircon"   "air conditioning"
##  [3,] "asprice"  "assessed price of home"
##  [4,] "bedroom"  "number of bedrooms"
##  [5,] "cond"     "condition of home"
##  [6,] "fireplc"  "fireplace"
##  [7,] "fullbth"  "number of full bathrooms"
##  [8,] "grade"    "grade of home"
##  [9,] "halfbth"  "number of half bathrooms"
## [10,] "htype"    "dwelling type"
## [11,] "live_fsq" "heated square feet"
## [12,] "live_usq" "unheated square feet"
## [13,] "owncode"  "owner code"
## [14,] "salepric" "sale price of home"
## [15,] "saleyr"   "sale year"
## [16,] "totroom"  "total number of rooms"
## [17,] "yearblt"  "year home was built"
```

```
shady$salepric <- shady$salepric/1000000
shady$asprice <- shady$asprice/1000000
model1 <- lm(asprice~salepric,data=shady)

#visual
plot(model1, which=3)
```



```
#White test
bptest(model1, varformula = ~salepric +I(salepric^2),
       data=shady)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 438.51, df = 2, p-value < 2.2e-16
```

```
summary(lm(I(model1$residuals^2)~salepric +I(salepric^2), data=model1$model))
```

```
##
## Call:
```

```
## lm(formula = I(model1$residuals^2) ~ salepric + I(salepric^2),
##     data = model1$model)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.052276 -0.001624 -0.000291  0.000680  0.134556
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0048370  0.0004887   9.898   <2e-16 ***
## salepric      -0.0707760  0.0050474 -14.022   <2e-16 ***
## I(salepric^2)  0.2310972  0.0104118  22.196   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007012 on 1094 degrees of freedom
## Multiple R-squared:  0.3997, Adjusted R-squared:  0.3986
## F-statistic: 364.3 on 2 and 1094 DF,  p-value: < 2.2e-16
```

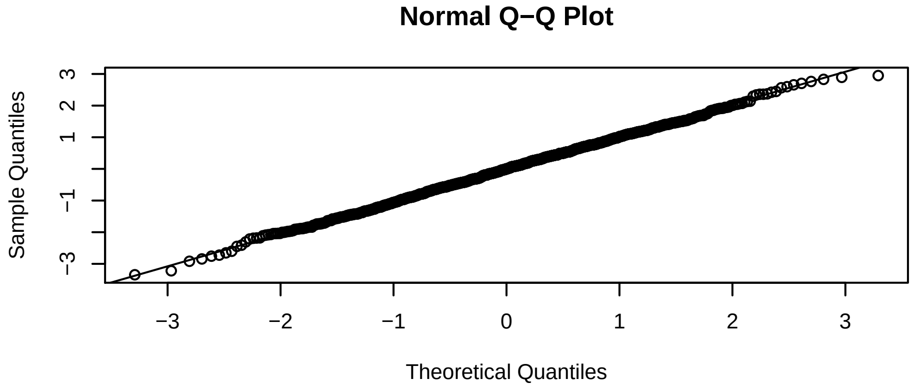
5.4.2 Tests for normality

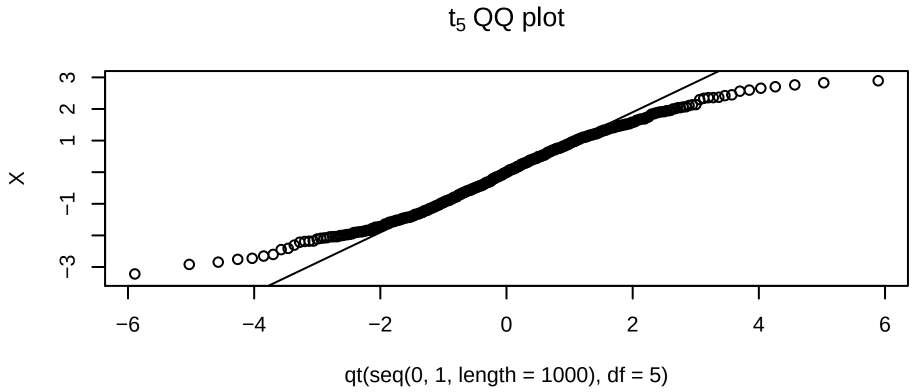
Before considering the efficiency gains, we want to know if worth mentioning here is that we also have tests for the normality of an error term. These tests are all based on the distribution of the residuals. Some informal diagnostics include looking at a QQ plot. Let's start with that.

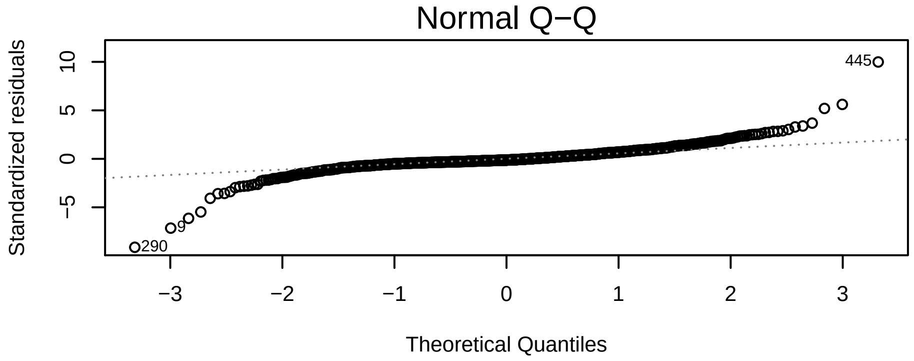
A QQ plot compares the empirical quantiles of a sample to the theoretical quantiles of a given distribution. For example, with the standard normal we know that 2.5% of observations should be below -1.96, half should be below 0, and 97.5% should be below 1.96.
The QQ plot sees how well a set of observations matches a set of expectations.

```
X <- rnorm(1000)
par(mfrow=c(3,1))
qqnorm(X)
qqline(X)
qqplot(qt(seq(0,1,length=1000),df=5),X, main=expression(t[5]~"QQ plot"))
qqline(X, distribution =function(p){qt(p, df = 5)})
```

```r
plot(model1, which=2)
```







The QQ plot of the residuals is iffy. The middle part looks fine, but are the end points bad enough to rule out normality? To be more certain we may want to consider an hypothesis test. Specifically the Jarque-Bera test is one such test we can consider. The JB test is based on the fact that under the null hypothesis the residuals will have a skew of 0 and kurtosis of 3 (which is true of any normal distribution). This is a test of if the 3rd and 4th empirical moments of the residuals match the normal's theoretical moments. The test statistic is based on the sample skewness:

$$S = \frac{\mathrm{E}[(\varepsilon - \mathrm{E}[\varepsilon])^3]}{\mathrm{E}[(\varepsilon - \mathrm{E}[\varepsilon])^2]^{3/2}} = \frac{\frac{1}{N}\sum_{i=1}^{N} \hat{\varepsilon}_i^3}{\left(\frac{1}{N}\sum_{i=1}^{N} \hat{\varepsilon}_i^2\right)^{3/2}}$$

and the sample kurtosis

$$K = \frac{\mathrm{E}[(\varepsilon - \mathrm{E}[\varepsilon])^4]}{\mathrm{E}[(\varepsilon - \mathrm{E}[\varepsilon])^2]^2} = \frac{\frac{1}{N}\sum_{i=1}^{N} \hat{\varepsilon}_i^4}{\left(\frac{1}{N}\sum_{i=1}^{N} \hat{\varepsilon}_i^2\right)^{2}}.$$

The JB test combines these into

$$JB = N\left(\frac{1}{6}S^2 + \frac{1}{24}(K-3)^2\right) \sim \chi_2^2.$$

The down side of the JB test is that it can only tell you if these two moment conditions fail. It can't distinguish between the normal and other distributions with skew 0 and kurtosis 3.

Alternative non-parametric tests include the Shapiro-Wilk and the Kolmogorov-Smirnov tests. Let's see how these do with the Shadyside data.

```
# JB test (implemented)
jarque.test(model1$residuals)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  model1$residuals
## JB = 19141, p-value < 2.2e-16
## alternative hypothesis: greater
```

```
# Does it do what we think it should?
N <- length(model1$residuals)
S <- mean(model1$residuals^3)/mean(model1$residuals^2)^(3/2)
K <- mean(model1$residuals^4)/mean(model1$residuals^2)^(2)
c(skewness(model1$residuals), S)
```

```
## [1] -0.04268419 -0.04268419
```

```
c(kurtosis(model1$residuals), K)
```

```
## [1] 23.46344 23.46344
```

```r
JB <- N*(S^2/6 + (K-3)^2/24)
JB
```

```
## [1] 19140.82
```

```r
pchisq(JB, lower=FALSE, df=2)
```

```
## [1] 0
```

```r
# Non parametric tests: you should probably stick with these, even
# if they're trickier to understand. They are more powerful tests
# Remember that power means the probability of *not* making a type II error
# Or the probability of not finding a false negative (we don't reject a false)

shapiro.test(model1$residuals) #just for normality
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model1$residuals
## W = 0.81088, p-value < 2.2e-16
```

```r
ks.test(scale(model1$residuals), "pnorm") #can compare to any distribution; here we use
```

```
## Warning in ks.test(scale(model1$residuals), "pnorm"): ties should not be present
## for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(model1$residuals)
## D = 0.14839, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```r
# a standard normal.
# NOTE: scale(x) standardizes x
# (x-mean(x))/sd(x)
# if the residuals are normal
# then scale(residuals) should be standard
# normal.
```

Here, we reject the null hypothesis of normality. This limits us to asymptotic test results and undercuts some of the efficiency results we have.

Going back to the presidential vote data, we can see

```r
shapiro.test(model3$residuals) #just for normality
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model3$residuals
## W = 0.96789, p-value = 0.7337
```

```r
ks.test(scale(model3$residuals), "pnorm") #can compare to any distribution;
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(model3$residuals)
## D = 0.099682, p-value = 0.9817
## alternative hypothesis: two-sided
```

In this case we do not reject the null of normality in any situation. Again this is good, because we have a very small number of observations. If we had rejected normality here, we'd have to make a very weak appeal to large sample properties to test hypotheses.

So what's next? Robust standard errors are great for large N because they don't require you to assume homoskedasticity ε. OLS with robust standard errors is still

1. Unbiased
2. Consistent
3. Asymptotically normal

But it's not BLUE! Lower variance unbiased and consistent estimators exist. Furthermore, the test statistics we derived for various hypothesis tests no longer have known finite distributions (we used homoskedasticity and normality for all our finite sample tests). We also lose asymptotic efficiency even if the errors are normal (OLS is no longer the MLE). We can gain more efficient estimates by modeling the error terms using whats called Generalized Least

Squares (GLS). In short it works by setting

$$\begin{aligned} y_i &\sim N(\beta' x_i, \sigma^2_{\varepsilon_i}) \\ \sigma^2_{\varepsilon_i} &= h(\gamma' z_i) \end{aligned}$$

where $h : \mathbb{R} \to \mathbb{R}_+$, say $e^{\gamma' z_i}$, and then estimating β, γ (usually by MLE). In theory you have the skills now to do this, but the efficiency gains are often minimal and are frequently questionable. If you have a good reason to use GLS, you can read on. But in the interest of teaching you things you might use, we will skip this for now.

5.4.3 Modeling with GLS

Robust standard errors are great for large N becauase they don't require you to assume either homoskedasticity or normality of ε. However, they can be inefficient in the sense that they don't use all the information on the table. This inefficiency can be particularly true if ε is normal but heteroskedastic. We can gain more efficient estimates by modeling the error terms.

Let's being with the assumption that the heteroskedasticity takes the form

$$\text{Var}(\varepsilon) = \begin{bmatrix} \sigma^2_{\varepsilon 1} & 0 & \dots & 0 \\ 0 & \sigma^2_{\varepsilon 2} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma^2_{\varepsilon N} \end{bmatrix} = \sigma^2_\varepsilon \begin{bmatrix} \omega_1 & 0 & \dots & 0 \\ 0 & \omega_2 & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & 0 & \omega_N \end{bmatrix} = \sigma^2_\varepsilon \Omega.$$

This format allows us to normalize Ω such that $\text{trace}(\Omega) = \sum_{i=1}^N \omega_i = N$. This normalization can always be found by setting $\sigma^2_\varepsilon = \text{trace}(\text{Var}(\varepsilon))/N$. In other words σ^2_ε as the common variance and the ωs reflect disturbances from this homoskedastic world. When $\omega = \mathbf{1}$, we have homoskedasticity.

We will now expand on our OLS framework by creating the **generalized least squares** estimator

$$\hat{\beta}_{GLS} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}y = \beta + (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\varepsilon.$$

Note that this looks a lot like the OLS

$$\hat{\beta}_{OLS} = (X'X)^{-1}X'y = \beta + (X'X)^{-1}X'\varepsilon.$$

Likewise the variances will look similar:

$$\begin{aligned}
\operatorname{Var}(\hat{\beta}_{GLS}|X) &= (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\operatorname{Var}(\varepsilon|X)\Omega^{-1}X(X'\Omega^{-1}X)^{-1} \\
&= (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\sigma_\varepsilon^2\Omega\Omega^{-1}X(X'\Omega^{-1}X)^{-1} \\
&= \sigma_\varepsilon^2(X'\Omega^{-1}X)^{-1}X'\Omega^{-1}X(X'\Omega^{-1}X)^{-1} \\
&= \sigma_\varepsilon^2(X'\Omega^{-1}X)^{-1} \\
\operatorname{Var}(\hat{\beta}_{OLS}|X) &= (X'X)^{-1}X'\operatorname{Var}(\varepsilon|X)X(X'X)^{-1} \\
&= \sigma_\varepsilon^2(X'X)^{-1}X'\Omega X(X'X)^{-1}
\end{aligned}$$

Which of these is better? Let's compare the variances.

$$\begin{aligned}
\operatorname{Var}(\hat{\beta}_{OLS}|X) - \operatorname{Var}(\hat{\beta}_{GLS}|X) &= \sigma_\varepsilon^2(X'X)^{-1}X'\Omega X(X'X)^{-1} - \sigma_\varepsilon^2(X'\Omega^{-1}X)^{-1} \\
\text{Let } A &= (X'X)^{-1}X' - (X'\Omega^{-1}X)^{-1}X'\Omega^{-1} \\
\text{Then } \sigma_\varepsilon^2 A\Omega A' &= \sigma_\varepsilon^2(X'X)^{-1}X'\Omega X(X'X)^{-1} - \sigma_\varepsilon^2(X'\Omega^{-1}X)^{-1}.
\end{aligned}$$

So the difference between the two variances is a positive constant times a quadratic, wrapped around a symmetric positive definite matrix. In other words, the differences between the variance of the GLS and the OLS is symmetric and positive definite. The GLS is thus more efficient than the OLS. However, it is not just the case that GLS is more efficient in this setup, it is the **most** efficient among unbiased linear estimates–it is BLUE.

Let $\tilde{\beta}$ be another linear estimator Cy. It is unbiased so

$$\begin{aligned}
\operatorname{E}[\tilde{\beta}] &= \operatorname{E}[\operatorname{E}[\tilde{\beta}|X]] \\
&= \operatorname{E}[\operatorname{E}[Cy|X]] \\
&= \operatorname{E}[\operatorname{E}[C(X\beta + \varepsilon)|X]] \\
&= \operatorname{E}[\operatorname{E}[CX\beta + C\varepsilon)|X]] \\
&= \operatorname{E}[\operatorname{E}[CX\beta|X] + C\operatorname{E}[\varepsilon)|X]] \\
&= CX\beta = \beta \\
CX &= I
\end{aligned}$$

and the variance is

$$\begin{aligned}
\operatorname{Var}(\tilde{\beta}|X) &= \operatorname{Var}(Cy|X) \\
&= \operatorname{Var}(CX\beta + C\varepsilon|X) \\
&= C\operatorname{Var}(\varepsilon|X)C' \\
&= \sigma_\varepsilon^2 C\Omega C'
\end{aligned}$$

Now let $D = C - (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}$. Note that

$$DX = CX - (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}X = I - I = 0,$$

and $C = D + (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}$. Now,

$$\begin{aligned}
\text{Var}(\tilde{\beta}|X) &= \sigma_\varepsilon^2 C\Omega C' \\
&= \sigma_\varepsilon^2 \left[D + (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\right] \Omega \left[D + (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}\right]' \\
&= \sigma_\varepsilon^2 D\Omega D' + \sigma_\varepsilon^2 (X'\Omega^{-1}X)^{-1} + \sigma_\varepsilon^2 DX(X'\Omega^{-1}X)^{-1} + \sigma_\varepsilon^2 (X'\Omega^{-1}X)^{-1}X'D' \\
&= \sigma_\varepsilon^2 D\Omega D' + \sigma_\varepsilon^2 (X'\Omega^{-1}X)^{-1} + 0 + 0 \\
&= \sigma_\varepsilon^2 D\Omega D' + \text{Var}(\hat{\beta}_{GLS}|X)
\end{aligned}$$

This means that

$$\text{Var}(\tilde{\beta}|X) - \text{Var}(\hat{\beta}_{GLS}|X) = \sigma_\varepsilon^2 D\Omega D.$$

Or, that the GLS is more efficient than any other LUE under this assumption.

Moving on, let's also consider the MLE under this setup of $\text{Var}(\varepsilon) = \sigma_\varepsilon^2\Omega$. Let's add a normality assumption back in, such that we assume that $\varepsilon_i \sim N(0, \sigma_{\varepsilon i}^2)$. Our likelihood becomes

$$\mathcal{L}(\beta|y, X) = \prod_{i=1}^{N} \frac{1}{\sigma_{\varepsilon i}\sqrt{2\pi}} e^{-\frac{1}{2\sigma_{\varepsilon i}^2}(y_i - \beta' x_i)^2}.$$

This is a mess, so let's take the log-likelihood

$$L(\beta|y, X) = \sum_{i=1}^{N} -\log(\sigma_{\varepsilon i}) - \frac{1}{2}\log(2\pi) - \frac{1}{2\sigma_{\varepsilon i}^2}(y_i - \beta' x_i)^2.$$

The FOC is then

$$\begin{aligned}
0 &= \sum_{i=1}^{N} \frac{1}{\sigma_{\varepsilon i}^2}(y_i - \hat{\beta}'_{MLE} x_i)x_i' \\
&= \frac{1}{\sigma_\varepsilon^2} X'\Omega^{-1}(y - X\hat{\beta}_{MLE}) \\
\hat{\beta}_{MLE} &= (X'\Omega^{-1}X)^{-1}(X'\Omega^{-1}y)
\end{aligned}$$

So with normal errors and heteroskedasticity, the MLE is the GLS estimator. Good news! This gives us two justifications for GLS if we suspect heteroskedasticity. It's BLUE and the asymptotically efficient. Bad news! This discussion has assumed we know Ω.
Most of the time this won't be the case. What do we do then?

5.4.4 Modeling Unknown heteroskedasticy

When we don't know Ω, but we want to obtain efficient estimates we have to model the heteroskedasticity. Specifically let $\mathrm{E}[\varepsilon_i^2|x_i] = h(z_i, \gamma)$. Some common options for this include

$$\begin{aligned} h(z_i,\gamma) &= \gamma' z_i && \text{Simple, but can be nonsense} \\ &= e^{\gamma' z_i} && \text{Always positive so that's nice} \end{aligned}$$

Assuming normal errors we can write a log-likelihood for this problem

$$L(\beta, \gamma|y, X) = \sum_{i=1}^{N} -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(h(z_i,\gamma)) - \frac{1}{2}h(z_i,\gamma)^{-1}(y_i - \beta' x_i)^2.$$

The FOC for β and γ give us

$$\begin{aligned} D_\beta L(\beta,\gamma|y,X) &= \sum_{i=1}^{N} h(z_i,\gamma)^{-1}(y_i - \beta' x_i)x_i' \\ D_\gamma L(\beta,\gamma|y,X) &= \sum_{i=1}^{N} -\frac{1}{2}\frac{D_\gamma h(z_i,\gamma)}{h(z_i,\gamma)} + \frac{1}{2}h(z_i,\gamma)^{-2}D_\gamma h(z_i,\gamma)(y_i - \beta' x_i)^2 \end{aligned}$$

These need to be solved simultaneously, but sadly they do not admit a closed for solution. We will need to solve these numerically using `optim`.

Alternatively, we could do this as a two-step. If we estimate γ first, then we can plug this $\hat{\gamma}$ into the FOC for β. This FOC then has a closed form solution in terms of the feasible GLS (FGLS) estimator:

$$\hat{\beta}_{FGLS} = \operatorname*{argmin}_{\beta} h(z_i;\hat{\gamma})^{-1}(y_i - \beta' x_i)^2 = \left[\frac{1}{N}\sum_{i=1}^{N}\frac{x_i x_i'}{h(z_i;\hat{\gamma})}\right]^{-1}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{x_i y_i}{h(z_i;\hat{\gamma})}\right]$$

or

$$\hat{\beta}_{FGLS} = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{y}$$

where $\tilde{X} = \begin{bmatrix} 1/\sqrt{h(z;\hat{\gamma})} & X_1/\sqrt{h(z;\hat{\gamma})} & \ldots & X_m/\sqrt{h(z;\hat{\gamma})} \end{bmatrix}$ and $\tilde{y} = y/\sqrt{h(z;\hat{\gamma})}$.

The consistency of this estimator follows from the properties of MLE. The asymptotic variance can be found by constructing the information matrix (Inverse of the negative Hessian evaluated at **truth**)

$$\begin{aligned}
D^2_{\beta\beta}L(\beta,\gamma|y,X) &= -\sum_{i=1}^{N} h(z_i;\gamma)x_i x_i' \\
D^2_{\beta\gamma}L(\beta,\gamma|y,X) &= -\sum_{i=1}^{N} h(z_i;\gamma)^{-2}D_\gamma h(z_i;\gamma)(y_i-\beta' x_i)x_i' \\
D^2_{\gamma\gamma}L(\beta,\gamma|y,X) &= H_\gamma && \text{Not interesting for us}
\end{aligned}$$

What is interesting is that

$$\begin{aligned}
\mathrm{E}[-D^2_{\beta\gamma}L(\beta,\gamma|y,X)] &= \mathrm{E}\left[\sum_{i=1}^{N} h(z_i;\gamma)^{-2}D_\gamma h(z_i;\gamma)(y_i-\beta' x_i)x_i'\right] \\
&= \mathrm{E}\left[\sum_{i=1}^{N} h(z_i;\gamma)^{-2}D_\gamma h(z_i;\gamma)\varepsilon x_i'\right] \\
&= 0
\end{aligned}$$

As such the information matrix is block diagonal

$$I(\beta,\gamma) = \begin{bmatrix} \mathrm{E}[h(z_i,\gamma)^{-1}x_i x_i']^{-1} & 0 \\ 0 & -\mathrm{E}[H_\gamma]^{-1} \end{bmatrix}.$$

This block diagonal structure means that we haven't lost any efficiency due to the fact that we've estimated γ, which is unusual for two-step approaches. Notice that part of the matrix that relates to β is the GLS variance with known variance such that

$$\sqrt{N}(\hat{\beta}_{MLE}-\beta) \xrightarrow{d} \sqrt{N}(\hat{\beta}_{GLS}-\beta_0),$$

or the MLE converges to the efficient GLS estimates. Wild.

The FGLS estimator can also be built without the nonlinear MLE model. Instead we can use three OLS steps to get here:

1. Find $\hat{\beta}_{OLS}$ and use this to form $\hat{\varepsilon}_i^2$ (consistent)
2. Regress $h^{-1}(\hat{\varepsilon}_i^2)$ on z_i to get $\hat{\gamma}$ (h^{-1} is log for the exponential model).
3. Find the FGLS estimates by weighting each observation by $h(\hat{\gamma}' z_i)$

Aside: there is subtle difference between the MLE and FGLS here, In MLE model we can set $h(\gamma' z_i) = \sigma_\varepsilon^2 \omega_i$, but in the GLS context this becomes $h(\gamma' z_i + u_i) = \sigma_\varepsilon^2 \omega_i$ in order to fit the auxiliary model. Further recall that σ_ε^2 and ω_i are not separately identified. The weights we get are estimates of ω_i that match whatever the final homoskedastic variance term is from the final FGLS step (step 3), $\sigma^2_{\tilde{\varepsilon}}$. Because it emerges in the final FGLS step, $\sigma^2_{\tilde{\varepsilon}}$ remains on the right-hand side of the variance model (step 2).

Where does this take us:

$$\begin{aligned}
h^{-1}(\sigma^2_{\tilde{\varepsilon}}\omega_i) &= \gamma' z_i + u_i \\
\log(\sigma^2_{\tilde{\varepsilon}}\omega_i) &= \gamma' z_i + u_i && \text{let } h(\cdot) = \exp(\cdot) \\
\log(\sigma^2_{\tilde{\varepsilon}}) + \log(\omega_i) &= \gamma' z_i + u_i && \text{Property of logs} \\
\log(\omega_i) &= \underbrace{\gamma_0 - \log(\sigma^2_{\tilde{\varepsilon}})}_{\text{constant in aux regression}} + \sum_{j=1}^{M} \gamma_j z_{ij} + u_i,
\end{aligned}$$

and

$$\begin{aligned}
\sqrt{\sigma^2_{\tilde{\varepsilon}}\omega_i} &= \gamma' z_i + u_i && \text{let } h(\cdot) = (\cdot)^2 \\
\sqrt{\omega_i} &= \left(\frac{\gamma}{\sigma_{\tilde{\varepsilon}}}\right)' z_i + \frac{u_i}{\sigma_{\tilde{\varepsilon}}}
\end{aligned}$$

As such the estimated variance of ε_i at point z_i is:

$$\begin{aligned}
\widehat{\mathrm{E}(\varepsilon_i^2|X = x_i)} &= h(\hat{\gamma}'_{\text{MLE}} z_i) && \text{any } h \text{ and MLE} \\
\widehat{\mathrm{E}(\varepsilon_i^2|X = x_i)} &= \hat{\sigma}^2_{\tilde{\varepsilon}} \exp(\hat{\gamma}'_{\text{FGLS}} z_i) && h = \exp \text{ and FGLS} \\
\widehat{\mathrm{E}(\varepsilon_i^2|X = x_i)} &= \hat{\sigma}^2_{\tilde{\varepsilon}} \left(\hat{\gamma}'_{\text{FGLS}} z_i\right)^2 && h(\cdot) = (\cdot)^2 \text{ and FGLS,}
\end{aligned}$$

where $\hat{\sigma}^2_{\tilde{\varepsilon}}$ is formed using the residuals of the final FGLS step (step 3), above.

However, the likelihood approach is often just as easy these days, and the multi-step FGLS is rarely used as result. Like the MLE, this version of the FGLS is asymptotically efficient with unknown variances even with the different steps.
Iterating this version of the FGLS will produce the MLE, but the properties are the same with or without iteration.

Both of the above are methods to model the heteroskedasticity that emerges in the data. However, they take some amount of work and require a functional form assumption on h. What's wrong with OLS? It's unbiased and consistent. We lose some efficiency, but is that a big deal? Recall that with heteroskedasticity the "classical" standard errors are incorrect, but robust standard errors are asymptotically correct. Make life easy for yourself, if you find heteroskedasticity and you have more than 50 observations use the OLS with robust standard errors. If you have to deal with these issues in smaller samples, it's best to collect more data.

5.4.5 Application

```
set.seed(1)
library(readstata13)
library(lmtest)
library(sandwich)
library(stargazer)
shady <- read.dta13("Rcode/datasets/shady.dta")
shady$salepric <- shady$salepric/1000000
shady$asprice <- shady$asprice/1000000
model1 <- lm(asprice~salepric,data=shady)
```

Above we saw evidence to suggest that heteroskedasticity was present. We can use robust standard errors or model it with GLS. Let's start by comparing the robust to the classic standard errors to see how much difference it makes

```
robust <- vcovHC(model1)
coeftest(model1)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.0230640  0.0022344  10.322 < 2.2e-16 ***
## salepric    0.7857927  0.0128068  61.358 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model1,vcov=robust)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.0230640  0.0046988  4.9085 1.057e-06 ***
## salepric    0.7857927  0.0381265 20.6101 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These are some pretty big differences here differences. We can try using the different two step approaches:

```
sigma2.i <- model1$residuals^2
model2.a <- lm(log(sigma2.i)~salepric, data=model1$model)
model2.b <- lm(asprice~salepric, data=model1$model,
               weights = 1/exp(model2.a$fitted.values))
coeftest(model2.b)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.0216608  0.0017367  12.472 < 2.2e-16 ***
## salepric    0.7957973  0.0154181  51.614 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Not much changed here. We could iterate or we could directly get the MLE

```
#### MLE for FLGS
#write the log-likelihood (negative)
LL <- function(theta, y, X,Z,h){
  ## theta: guess of the parameters order (beta, gamma)
  ## y: dependent variable
  ## X: regressors associated with beta
  ## z: regressors associated with gamma
  ## h: function for transforming Z*gamma into sigma^2.

  beta <- theta[1:ncol(X)]
  gamma <- theta[(ncol(X)+1):length(theta)]
  sigma2 <-h(Z=Z, gamma=gamma, grad=FALSE)

  log.lik <- -log(sigma2)/2 - ((y-X%*%beta)^2/(2*sigma2))
  log.lik <- sum(log.lik)
  return(-log.lik)
}
```

```r
grad <- function(theta, y, X,Z,h){
  ## theta: guess of the parameters order (beta, gamma)
  ## y: dependent variable
  ## X: regressors associated with beta
  ## z: regressors associated with gamma
  ## h: function for transforming Z*gamma into sigma^2.
  ## If grad=TRUE, h also returns the first derivative of h wrt to gamma


  beta <- theta[1:ncol(X)]
  gamma <- theta[(ncol(X)+1):length(theta)]
  hout <-  h(Z=Z, gamma=gamma, grad=TRUE)
  sigma2 <- hout[[1]]
  mu <- drop(X%*%beta)
  Dh <- hout[[2]]

  Dbeta <- ((y-mu)/sigma2)*X
  Dgamma <- (Dh*((y-mu)^2)/(2*sigma2^2) - Dh/(2*sigma2))
  return(-colSums(cbind(Dbeta, Dgamma)))
}

## Build the data
X <- as.matrix(cbind(1, model1$model$salepric))
Z <- X
y <- model1$model$asprice
h <- function(Z, gamma, grad=FALSE){
  output <- exp(drop(Z%*%gamma))
  if(grad==TRUE){
    output <- list(output,
                   Z*exp(drop(Z%*%gamma)))
  }
  return(output)
}

set.seed(1)
theta0 <- runif(4)
```

```r
## minimize the negative loglikelihood
MLE.exp <- optim(theta0, fn = LL,
                 gr=grad,
                 y=y, X=X,Z=Z,h=h,
                 method="BFGS", hessian=TRUE)
theta.hat.exp <- MLE.exp$par
theta.se.exp <- sqrt(diag(solve(MLE.exp$hessian)))

# individually test that true parameters are 0
t.exp <- theta.hat.exp/theta.se.exp
p.exp <- 2*pnorm(abs(t.exp), lower.tail = F)
print(cbind(theta.hat.exp, theta.se.exp, t.exp, p.exp))
```

```
##      theta.hat.exp theta.se.exp      t.exp        p.exp
## [1,]     0.0214343  0.001776764   12.06367 1.642943e-33
## [2,]     0.7981652  0.014218521   56.13560 0.000000e+00
## [3,]    -7.3629853  0.063503117 -115.94683 0.000000e+00
## [4,]     5.2350995  0.333943699   15.67659 2.186725e-55
```

```r
#try another functional form
h <- function(Z, gamma, grad=FALSE){
  output <- (drop(Z%*%gamma)^2)
  if(grad==TRUE){
    output <- list(output,
                   2*drop(Z%*%gamma)*Z)
  }
  return(output)
}
MLE.sq <- optim(theta0, fn = LL,
                y=y, X=X,Z=Z,h=h,
                gr=grad,
                method="BFGS", hessian=TRUE)
theta.hat.sq <- MLE.sq$par
theta.se.sq <- sqrt(diag(solve(MLE.sq$hessian)))
t.sq <- theta.hat.sq/theta.se.sq
p.sq <- 2*pnorm(abs(t.sq), lower.tail = F)
```

```
print(cbind(theta.hat.sq, theta.se.sq, t.sq, p.sq))
```

```
##      theta.hat.sq  theta.se.sq     t.sq          p.sq
## [1,]   0.02187353 0.0017755401 12.31937  7.125169e-35
## [2,]   0.79440084 0.0138985185 57.15723  0.000000e+00
## [3,]   0.02438908 0.0009473996 25.74319 3.842623e-146
## [4,]   0.09947869 0.0070059502 14.19917  9.270227e-46
```

The estimates of β are roughly unchanged.
All of the FGLS methods produce smaller standard errors than the Huber-White estimator, but we used the normal likelihood for the MLE version. The sandwich estimator doesn't need normal error terms. However, the substantive conclusions are unchanged.

```
names(theta.hat.exp) <- names(theta.hat.sq) <-
  names(theta.se.exp) <- names(theta.se.sq) <- names(model1$coef)
#for write ups
stargazer(model1, model1, model2.b,
          model1, model1, #What is this?
          coef=list(NULL, NULL, NULL, theta.hat.exp, theta.hat.sq),
          column.labels = c("OLS", "OLS-robust", "FGLS-LS",
                            "FGLS-MLE (exp)", "FGLS-MLE (sq)"),
          dep.var.labels = "Apraisal price",
          covariate.labels = "Sale price",
          keep.stat = "n", #Don't report stats when using "dummy" models
          header=FALSE,
          digits=2,
          se=list(NULL,sqrt(diag(robust)), NULL, theta.se.exp, theta.se.sq))
```

What have gained here? Not much. The robust standard errors provide the most conservative output, so it's probably wise to stick with that. Almost no one uses the FGLS for just heteroskedasticity, but the framework re-appears in other settings (time series, panel data).

5.4.6 Linear probability model

While we're talking about normality and heteroskedasticity let's consider the special case where $y_i \in \{0, 1\}$ (i.e., an event did or didn't happen). We will still impose Assumption B1 $y_i = \beta' x_i + \varepsilon_i$. This special case is called the linear probability model (LPM). The main thing

Table 6

	Dependent variable:				
	Apraisal price				
	OLS	OLS-robust	FGLS-LS	FGLS-MLE (exp)	FGLS-MLE (sq)
	(1)	(2)	(3)	(4)	(5)
Sale price	0.79^{***}	0.79^{***}	0.80^{***}	0.80^{***}	0.79^{***}
	(0.01)	(0.04)	(0.02)	(0.01)	(0.01)
Constant	0.02^{***}	0.02^{***}	0.02^{***}	0.02^{***}	0.02^{***}
	(0.002)	(0.005)	(0.002)	(0.002)	(0.002)
Observations	1,097	1,097	1,097	1,097	1,097
Note:					*p<0.1; **p<0.05; ***p<0.01

to note here is that in order for y to be 0 or 1 it must be that

$$\varepsilon_i = \begin{cases} -\beta' x_i & y_i = 0 \\ 1 - \beta' x_i & y_i = 1 \end{cases}.$$

Further note that ε_i only takes on two values. Let p_i be the probability that y_i is 1 given x_i, then the expected value of ε_i is given by

$$\begin{aligned} \mathrm{E}[\varepsilon_i|x_i] = 0 &= p_i(1 - \beta' x_i) + (1 - p_i)(-\beta' x_i) \\ 0 &= p_i - \beta' x_i p_i - \beta' x_i + p_i \beta' x_i \\ p_i &= \beta' x_i = \Pr(y_i = 1|x_i) \end{aligned}$$

and variance

$$\begin{aligned} \mathrm{E}[\varepsilon_i^2|x_i] &= p_i(1 - \beta' x_i)^2 + (1 - p_i)(-\beta' x_i)^2 \\ &= \beta' x_i(1 - \beta' x_i) \end{aligned}$$

The constraints on the errors are very specific requirements that are unlikely to be met in practice, but it does tell us that normality is gone and heteoskedasticity is built in by construction. As such, we are firmly in the world of relying on large-sample properties and robust standard errors are a must.

Note that based on the above we can say that y_i is a Bernoulli outcome with $\mathrm{E}[y_i|x_i] =$

$\beta' x_i = \Pr(y_i|x_i)$. The OLS estimator still gives us the basic results that

$$\mathrm{E}[\hat{\beta}|X] = \beta + (X'X)^{-1}X' \, \mathrm{E}[\varepsilon|X] = \beta.$$

As such, we still have unbiasedness (you can conduct a similar exercise for consistency). The marginal effects are similar: On average, a one [unit of x_k] increase in x_k is associated with a $\hat{\beta}_k$ increase in the **probability** that $y = 1$, holding all other independent variables constant.

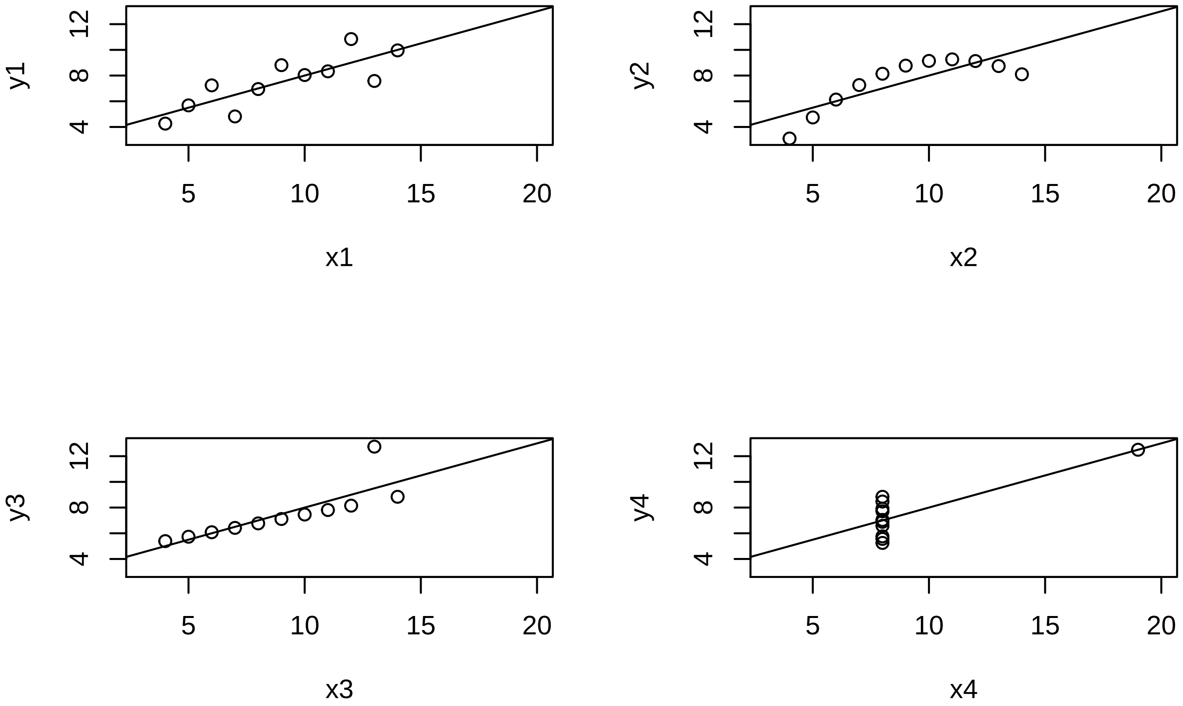
Note however, that when $\hat{\beta}' x_i$ is outside the unit interval (which can happen) we have an estimate for $p_i = \Pr(y_i|x_i)$ that is nonsense. Others have shown that when this happens it implies a misspecification in the LPM leading to biased and inconsistent estimates. These nonsense results (and the inconsistency they imply) leads to alternatives like logit and probit regression (more on this next semester).

However, even with nonsense probabilities, the bias in β doesn't appear to be too bad in practice, and so you should think carefully about whether you want to give up the linear model. And, the LPM does have some advantages over these nonlinear alternatives. In more complicated panel or instrumental variables settings the LPM can make your life a lot easier (both of which we'll cover if there's more time).

5.5 Outliers and multicolinearity

Two other issues we're going to consider are outliers and multicolinearity. Consider the following four data sets

```r
par(mfrow=c(2,2))
plot(y1~x1, data=anscombe, ylim=c(3,13), xlim=c(3,20))
abline(reg=lm(y1~x1,data=anscombe))
plot(y2~x2, data=anscombe, ylim=c(3,13), xlim=c(3,20))
abline(reg=lm(y2~x2,data=anscombe))
plot(y3~x3, data=anscombe, ylim=c(3,13), xlim=c(3,20))
abline(reg=lm(y3~x3,data=anscombe))
plot(y4~x4, data=anscombe, ylim=c(3,13), xlim=c(3,20))
abline(reg=lm(y4~x4,data=anscombe))
```



These four examples are called "Anscombe's Quartet". All $X's$ have the same mean and variance, all the y's have the same mean and variance, all the pairings have the same covariance, correlation, and regression parameters.

```
round(colMeans(anscombe),2)
```

```
##  x1  x2  x3  x4  y1  y2  y3  y4
## 9.0 9.0 9.0 9.0 7.5 7.5 7.5 7.5
```

```
round(apply(anscombe, 2, var),2)
```

```
##    x1    x2    x3    x4    y1    y2    y3    y4
## 11.00 11.00 11.00 11.00  4.13  4.13  4.12  4.12
```

```
with(anscombe,cor(x1,y2))
```

```
## [1] 0.8162365
```

```
with(anscombe,cor(x2,y2))
```

```
## [1] 0.8162365
```

```
with(anscombe,cor(x2,y2))
```

```
## [1] 0.8162365
```

```r
with(anscombe,cor(x2,y2))
```

```
## [1] 0.8162365
```

However, what should be clear here is that only the the first one really looks like an appropriate use of the linear model. The second one looks like it has a clear bend that should be accounted for, the bottom two have clearly influential points that "pull" the line towards themselves. The fourth one is straight up misleading. Graphing data like this provides clear insight into the problems of influence and is usually a good place to start with data analysis. Of course this gets less informative with more covariates.

Because these cases get less clear in real life we can consider some tools for diagnosing influential points and outliers. To do this we're going to want to rescale the residuals (the gap between point and line) to be on a *standardized measure*. What tool do we have for that? We can z transform them. To do this we need to calculate the variance of each residual (note that this is different than estimation of σ^2_ε), here we finding each residual's individual standard error rather than estimating the distribution they're drawn from. Recall that M is the residual maker matrix and is idempotent and symmetric.

$$\begin{aligned}\text{Var}(\hat{\varepsilon}|X) &= \text{Var}(My|X)\\ &= M\,\text{Var}(y)M\end{aligned}$$

Under homoskedasticity and independence we get

$$\begin{aligned}\text{Var}(\hat{\varepsilon}|X) &= \sigma^2_\varepsilon M\\ &= \sigma^2_\varepsilon(I - X(X'X)^{-1}X')\end{aligned}$$

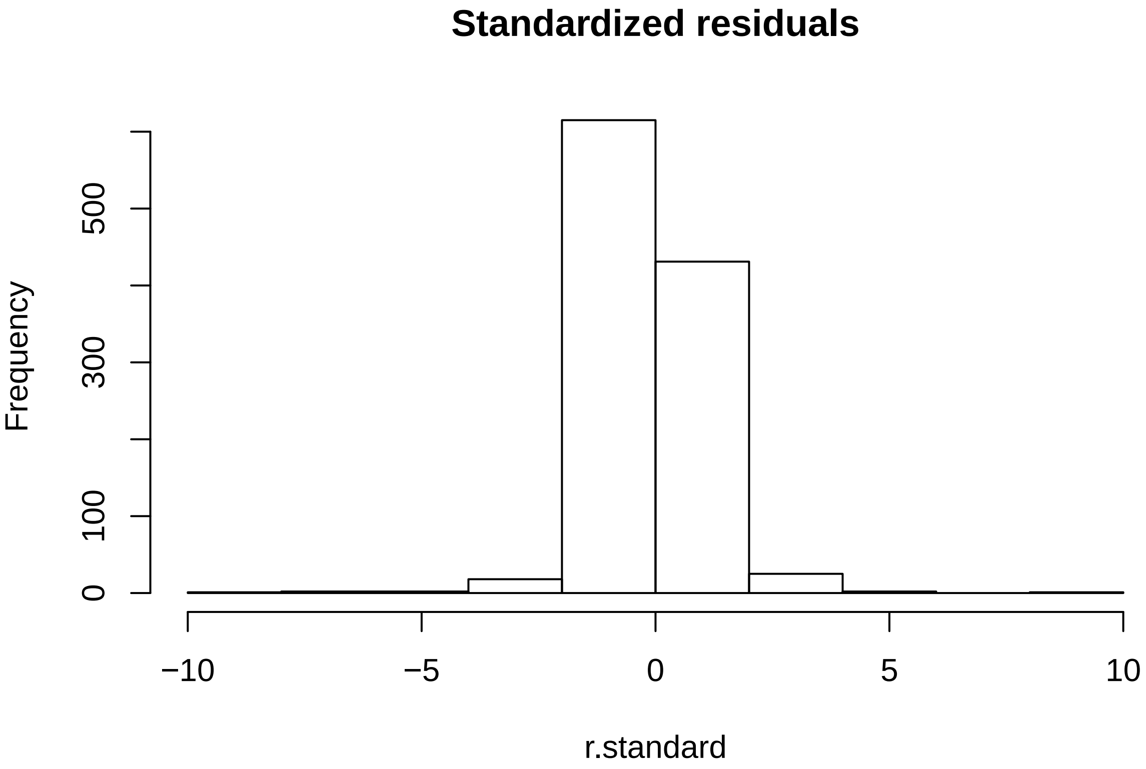
Let $H = X(X'X)^{-1}X'$ this is often called the **hat matrix** (or the **projection matrix**), because it puts a hat on y $Hy = \hat{y}$. Measures of influence are frequently based on the diagonal elements of H or M. Let h be the diagonal elements of H, then h_i is called the "leverage" of observation i (sometimes leverage is defined as $h_i/(1-h_i)$.

We can then standardize the residuals such that

$$\tilde{\varepsilon}_i = \frac{\hat{\varepsilon}_i}{\hat{\sigma}_\varepsilon\sqrt{(1-h_i)}}.$$

If we assume that the error term is normally distributed then $\tilde{\varepsilon}_i$ will be distributed t with $N-K$ degrees of freedom. Under homoskedasticity and normality, absolute values greater than 3 would be considered unlikely. Hard to say about other conditions, but look for long tails. For example

```
r.standard <- rstandard(model1)
hist(r.standard,
     main="Standardized residuals")
```



```
head(sort(abs(r.standard), decreasing = TRUE))
```

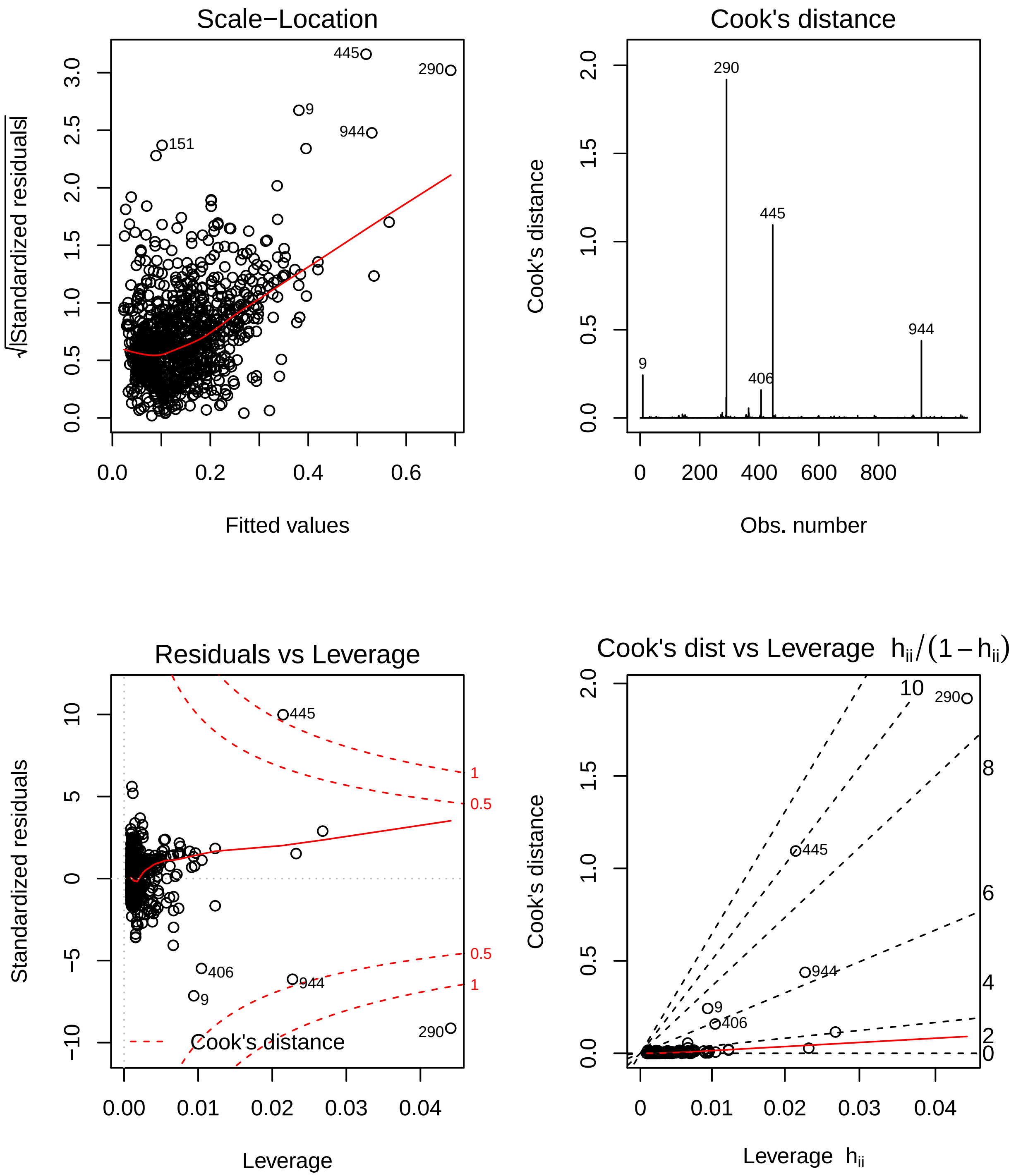
```
##      445      290        9      944      151      406 
## 9.989009 9.121431 7.145148 6.135457 5.611967 5.479210
```

Note that these cut points are sensitive to the assumptions of normality and homoskedasticity (both of which we rejected in this data). However, they do still flag some observations to look at. Another measure of influence is called "Cook's distance." Like the standardized residuals uses the diagonals from the hat matrix to capture the influence of a point. The Cook's distance is given as

$$D_i = \frac{\hat{\varepsilon}_i^2 h_i}{K\hat{\sigma}_\varepsilon^2(1-h_i)^2}.$$

This works out to be a scaled measure of how much the model predictions for all the remaining points would change if we dropped i. The basic "rule of thumb" here is that any observation with a distance greater than 1 should be investigated. however, this rule of thumb also relies on the distributional assumptions.

```r
par(mfrow=c(2,2))
plot(model1, which=3:6, id.n=5)
```



```r
# 3 same plot we used to look at heteroskedasticity
# 4-6 are outlier/leverage specific (different ways to look)
which(cooks.distance(model1) > 1)
```

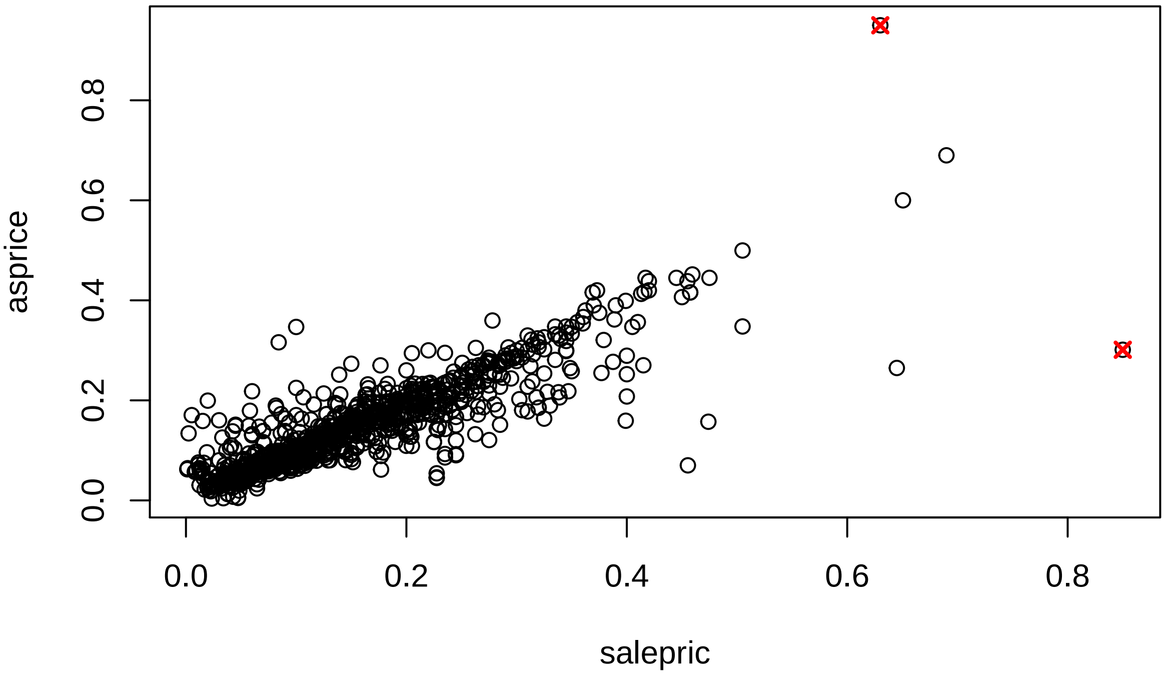
```
## 290 445
## 290 445
```

```
head(sort(hatvalues(model1), decreasing=T))
```

```
##        290        289        365        944        445        142
## 0.04409478 0.02680814 0.02322143 0.02273828 0.02145893 0.01230014
```

Basically, these are telling us to consider observations 290 and 445 to make sure they aren't too weird/influential.

```
par(mfrow=c(1,1))
plot(asprice~salepric, data=shady)
points(asprice~salepric, data=shady[c(290,445),], col="red", pch=4,lwd=2)
```



```
model1.not290 <- lm(asprice~salepric,data=shady[-c(290),])
model1.not445 <- lm(asprice~salepric,data=shady[-c(445),])
model1.neither <- lm(asprice~salepric,data=shady[-c(290,445),])
mod.list <- list(model1, model1.not290, model1.not445, model1.neither)
se.list <- lapply(mod.list,
                  function(x){sqrt(diag(vcovHC(x)))})
stargazer(mod.list,
          dep.var.labels = "Apraisal price",
          covariate.labels = "Sale price",
          # type="text",
          se=se.list,
          title="Checking for outliers",
          column.labels = c("Full sample", "Drop 290",
                            "Drop 445", "Drop Both"),
          label="tab:outliers",
          header = FALSE,
          keep.stat = c("n"))
```

Table 7: Checking for outliers

	Dependent variable:			
	Apraisal price			
	Full sample	Drop 290	Drop 445	Drop Both
	(1)	(2)	(3)	(4)
Sale price	0.786***	0.811***	0.767***	0.792***
	(0.038)	(0.030)	(0.033)	(0.024)
Constant	0.023***	0.020***	0.025***	0.022***
	(0.005)	(0.004)	(0.004)	(0.003)
Observations	1,097	1,096	1,096	1,095
Note:				*p<0.1; **p<0.05; ***p<0.01

The differences are pretty minor, so we'll proceed with the full sample.

When you hear people talk about colinerity they are concerned about the full rank of $X'X$. Perfect multicolinearity occurs when one variable is a perfect linear combination of one or more of the other independent variables (including the constant). For example, if you have a variable that is years of education, another that is years of military service, and a third that is candidate quality that is the sum of military service and education, then that would be perfectly colinear and $X'X$ would be non-invertable. Likewise if you included a variable for whether a candidate was male and your sample happened to only be a set of elections with only male candidates, then this variable perfectly correlates with with the constant and $X'X$ is not full rank (and thus non-invertable). If $X'X$ inverts then you don't have perfect multicolinearity.

In cases of less than perfect multicolinearity it is the case that some of the independent variables are highly correlated with each other (but not at ±1). These high correlations can make it difficult for the model to suss out which variable is having what effect on y. As such the estimates may become sensitive to small changes in the data, there may be ridiculously large standard errors, weird signs, etc. This is usually a problem of having too few observations, so the first thing to do is to try and collect more data. Some people will tell you to check things like the variable inflation factor (VIF) as a diagnostic and make changes if the VIF is greater than 10. The VIF is given as

$$\text{VIF}(X_j) = \frac{1}{1 - R_j^2} \text{ and } \text{Var}(\hat{\beta}_j|X) = \text{VIF}(X_j)\frac{\sigma_\varepsilon^2}{\sum_{i=1}^N (x_{ij} - \bar{x}_j)^2}.$$

Getting worried about the VIF is generally not great advice, however. Don't bother checking for colinearity unless someone tells you (reviewer, adviser, etc). If

1. $(X'X)^{-1}$ exists,
2. You have no theoretical reason to suspect colinearity,
3. None of the estimates or standard errors look like you tried to divide by zero,

Then you should be okay. Alternatively, if you have two or more variables that are so highly related that you think they're causing problems you can try combining them using something like an average, or a minimum, or some kind of factor analysis.

```r
library(car)
library(readstata13)
rm(list=ls())
## car package has a VIF function (but don't do this)
presdata <- read.dta13("Rcode/datasets/presvote.dta")

presdata$IncShare[33:34] <- c(0.4631,0.5196 )
presdata$IncShare <- presdata$IncShare*100
votemodel <- lm(IncShare~App+Growth+Inflation, data=presdata, x=T)
vif(votemodel)
```

```
##       App    Growth Inflation
##  1.093159  1.050508  1.134004
```

```r
which(cooks.distance(votemodel)>1) #nothing flagged
```

```
## named integer(0)
```

5.6 Functional form changes

Now we're going to assess the "linear" part of linear regression. Assumption B1 is that y is a linear function of Xs. However, we know many situations where a curve or discontinuity or other oddity may be more appropriate. Fear not however, as the linear part of our assumption refers to "linear in the parameters." This means that, as appropriate, we can tweak our variables in order to get nonlinear outputs within the linear framework by using creative transformations of the data.

Sometimes these changes are justified as a "correction" for non-normality or heteroskedasticity, but take care there. For one thing, these "problems" don't affect the bias or consistency, so don't induce new complications to "fix" them. We should do use transformations when we have theoretical reasons for suspecting a nonlinear relationship and/or we think it's the right way to answer our question.

5.6.1 Logs

One of the most common (and useful) transformations is the log transformation. There are four possible models to consider here

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i && \text{linear-linear model} \\ y_i &= \beta_0 + \beta_1 \log(x_{1i}) + \beta_2 \log(x_{2i}) + \varepsilon_i && \text{linear-log model} \\ \log(y_i) &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i && \text{log-linear model} \\ \log(y_i) &= \beta_0 + \beta_1 \log(x_{1i}) + \beta_2 \log(x_{2i}) + \varepsilon_i && \text{log-log model} \end{aligned}$$

All of these models are linear models in the sense of "linear in the parameters." However, the interpretations change as a result of these different functional forms.

Linear-linear Here we have our basic interpretation: "On average, a one [unit of X_1] increase in X_1 is associated with a $\hat{\beta}_1$ [unit of y] increase in y, holding X_2 constant."

Linear-log Here is where things start to get funky. Let's start by taking the derivative wrt to X_1 on both sides. We'll switch to partial notation for this

$$\begin{aligned} \frac{\partial y_i}{\partial x_{1i}} &= \frac{1}{x_{1i}} \beta_1 \\ \beta_1 &= \frac{\partial y_i}{\partial x_{1i}/x_{1i}} \end{aligned}$$

Notice that $\partial x_{1i}/x_{1i}$ is a change in X_1 over X_1, this is the formula for percentage change (for small changes). Recall that percent change is often framed as

$$\%\text{Increase} = \frac{a_1 - a_0}{a_0} \times 100.$$

Using a bit of linear extrapolation, we can say that the rate of change reflected in β_1 is in terms of how many more y's per a 1 percentage increase in X_1.

$$\beta_1/100 = \frac{\partial y_i}{100 \times \partial x_{1i}/x_{1i}}$$

Our becomes interpretation: "On average, a one percent increase in X_1 is associated with a $\hat{\beta}_1/100$ [unit of y] increase in y, holding X_2 constant."

Log-linear Starting from the same point we'll take the derivative

$$\begin{aligned}
\log(y_i) &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i \\
y_i &= \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i) \\
\frac{\partial y_i}{\partial x_{1i}} &= \beta_1 \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i) \\
&= \beta_1 y_i \\
\frac{\partial y_i/y_i}{\partial x_{1i}} &= \beta_1 \\
\frac{100 \times \partial y_i/y_i}{\partial x_{1i}} &= \beta_1 \times 100.
\end{aligned}$$

Now what we see is that the percentage change is going on with y. Using the same line extrapolation we reach the interpretation: "On average, a one [unit of X_1] increase in X_1 is associated with a [$\hat{\beta}_1 \times 100$] percent increase y, holding X_2 constant."

Log-log Starting from the same point we'll take the derivative

$$\begin{aligned}
\log(y_i) &= \beta_0 + \beta_1 \log(x_{1i}) + \beta_2 x_{2i} + \varepsilon_i \\
y_i &= \exp(\beta_0 + \beta_1 \log(x_{1i}) + \beta_2 x_{2i} + \varepsilon_i) \\
\frac{\partial y_i}{\partial x_{1i}} &= \frac{1}{x_{1i}} \beta_1 \exp(\beta_0 + \beta_1 \log(x_{1i}) + \beta_2 x_{2i} + \varepsilon_i) \\
&= \beta_1 \frac{y_i}{x_{1i}} \\
\frac{\partial y_i / y_i}{\partial x_{1i} / x_{1i}} &= \beta_1 \\
\frac{100 \times \partial y_i / y_i}{100 \times \partial x_{1i} / x_{1i}} &= \beta_1.
\end{aligned}$$

Now what we see is that the percentage change is going on with both sides. Using the same line extrapolation we reach the interpretation: "On average, a one percent increase in X_1 is associated with a $\hat{\beta}_1$ percent increase y, holding X_2 constant."

When it comes to choosing between these different forms, The most important guide is what you think makes sense between a linear effect (no increase or decrease in the effect of X on y) and diminishing or increasing returns. For example, the effectiveness of money on vote share: when the spending is at low levels, a \$1000 increases might have a big impact, but when spending is in the millions, another \$1000 might have a tiny impact. Plotting your data can help here (but this gets complicated with many X's) You can compare log-log to log-linear using R^2, adjusted R^2, AIC, and BIC. Likewise for linear-linear to linear-log. However, these measures require the same dependent variable, so you can't compare log-linear to linear-linear with them. Later, we'll decsribe a test that allows you to ask if y is normal versus log-normal. Generally speaking, my best advice is read papers similar to yours, see what they do, plan accordingly.

We'll look at some other non-linearity approaches and then try some examples.

5.6.2 Polynomials

Another way to introduce nonlinearities and curves are polynomials. Polynomials are incredibly flexible way to capture all sorts of bendy relationships. With enough polynomials any complicated continuous function can be approximated arbitrarily well. However, most of the time we're interested in 1 or 2 bends (what theories really lead us to more than that?). Imagine a regression that predicts domestic terrorist attacks as a function of how democratic a state is. At the low end we have repressive North Korea style states which prevent any kind of organization and thus have a lower risk of domestic terrorism. At the high end we have liberal, highly open societies. In these states there are many non-violent options for political expression and so terrorism is less attractive for engineering real change. In the middle however, are the anocracies. In these states, there is neither a effective political outlet nor an inherently strong repressive regime. Here we might expect the most terrorist attacks. The regression could then be:

$$A_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3' z_i + \varepsilon_i,$$

which is a quadratic function. This functional form puts a U or an upside-down U relationship between X and y.

The marginal effects are, as always, based on the first derivative

$$D_{x_i} A_i(x^*) = \beta_1 + 2\beta_2 x^*.$$

What's new here is that the marginal effect is *based* on the value of X. If β_2 is negative, then we have a marginal effect that is **decreasing** (moving towards $-\infty$ as X increases. When the marginal effect is 0, we are at the value of X associated with the maximum (or minimum if $\beta_2 > 0$) number of expected attacks,

$$\begin{aligned} D_{x_i} A_i &= \beta_1 + 2\beta_2 x_i \\ 0 &= \beta_1 + 2\beta_2 x_i \\ x_i &= \frac{-\beta_1}{2\beta_2}. \end{aligned}$$

We can add more polynomials if we think the relationship is complicated, but this becomes increasing hard to interpret (and increasingly suspect for outside readers). Standard errors for marginal effects can be calculated as

$$\begin{aligned} \operatorname{Var}(\widehat{D_{x_i} A_i}|X) &= \operatorname{Var}(\hat{\beta}_1 + 2\hat{\beta}_2 x_i|X) \\ &= \operatorname{Var}(\hat{\beta}_1|X) + 4x_i^2 \operatorname{Var}(\hat{\beta}_2|X) + 4x_i \operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_2|X) \\ &= \begin{bmatrix} 0 & 1 & 2x_i & 0 & \dots & 0 \end{bmatrix} \operatorname{Var}(\hat{\beta}|X) \begin{bmatrix} 0 & 1 & 2x_i & 0 & \dots & 0 \end{bmatrix}' \\ se(\widehat{D_{x_i} A_i}|X) &= \sqrt{\operatorname{Var}(\hat{\beta}_1|X) + 4x_i^2 \widehat{\operatorname{Var}}(\hat{\beta}_2|X) + 4x_i \operatorname{Cov}(\hat{\beta}_1, \hat{\beta}_2|X)} \end{aligned}$$

Sometimes the maximum/minimum will be of interest. This is nonlinear function of two estimates so we will use the delta method to form a standard error of the maximum/minimum

$$\text{avar}\left(\frac{-\hat{\beta}_1}{2\hat{\beta}_2}\right) = \left(D_\beta \frac{-\hat{\beta}_1}{2\hat{\beta}_2}\right)' \text{avar}(\hat{\beta}) \left(D_\beta \frac{-\hat{\beta}_1}{2\hat{\beta}_2}\right)$$

$$D_\beta \frac{-\hat{\beta}_1}{2\hat{\beta}_2} = \left(0, -\tfrac{1}{2\hat{\beta}_2}, \tfrac{\beta_1}{2\beta_2^2}\right)$$

$$se\left(\frac{-\hat{\beta}_1}{2\hat{\beta}_2}|X\right) = \sqrt{\left(0, -\tfrac{1}{2\hat{\beta}_2}, \tfrac{\beta_1}{2\beta_2^2}\right)' \widehat{\text{avar}}(\hat{\beta}) \left(0, -\tfrac{1}{2\hat{\beta}_2}, \tfrac{\beta_1}{2\beta_2^2}\right)}$$

Many times, it'll be of interest to just plot the marginal effects with a 95% confidence interval as a function of X.

5.6.3 Dummy variables

Our next attempt at introducing flexibility into the functional form is in the form of "dummy" variables. Dummy variables take on only two possible values 0 and 1. In most samples, gender will be coded as a dummy with 1 indicating female and 0 indicating male, although this might change as we collect better data. Regions of the country (or the world) can be coded as individual dummies: (Northeast, Midwest, South, West). When including a set of one or more dummies it is important to always omit one category. For example, unless your sample is very inclusive you will not be able to include both a Female and Male dummy variable. The need to exclude a category results from the colinearity results we described above. If the two variables Male and Female cover all individuals exclusively then we get $\text{Female}_i + \text{Male}_i = 1$, which is perfectly colinear with the constant term. With the regions we would include three of the four regions and leave the fourth out. To recap any set of dummies must have an excluded category.

This leave-one-out strategy effects the interpretation of the coefficient on the dummy. In the case of a single dummy reflecting two categories this is average effect (or difference between) of moving from a 0 (excluded category) into a 1 (included category). For gender $\hat{\beta}$ captures the average difference between Women and Men (holding all else constant). In the regions case suppose we exclude the Midwest, then we have

1. $\hat{\beta}_{NE}$ is the average difference between the NE and the MW
2. $\hat{\beta}_S$ is the average difference between the S and the MW
3. $\hat{\beta}_W$ is the average difference between the W and the MW

Note it doesn't matter which one we exclude. The magnitudes will change but the relative differences will be intact. Consider the question of how to include age in a regression that predicts turnout, we could do any of the following

1. Include age as a single, untransformed variable (linear, constant positive effect).
2. Include age as a single, logged variable (nonlinear, effect is always positive buy we may suspect it levels out at some point age)
3. Include age and age squared (nonlinear, effect of age is positive then possibly negative)
4. Include age dummies for groups 18-24, 25-44, 45-64, 65+ (nonlinear, allows each group to have their own average turnout rate)

5.6.4 Interactions

The last specification we'll look at in this section is an "interaction model." Here we are interested in whether the effect of one of our independent variables is depends on another independent variable. For example, the effect that migrant remittances have on a group's decision to engage in political violence. We might suspect that this relationship depends on what kind of country the groups is in. In more democratic states, more financial resources through remittances may lead to less violence as there are legitimate political institutions available that are more effective with money. In less democratic states, there may not be such legimitate outlets so violence increases with resources. We can test this hypothesis using an interaction model where we multiply one variable by the other, such that

$$A_i = \beta_0 + \beta_1 r_i + \beta_2 d_i + \beta_3 r_i d_i + \beta_4' x_i + \varepsilon_i.$$

Here, A_i is the number of violent incidents, r_i the amount of remittances being sent to country i, and d_i is a measure of democracy in country in i (maybe a dummy, maybe a scale).

The marginal effect of remittances then becomes

$$D_{r_i} A_i(d^*) = \beta_1 + \beta_3 d^*.$$

If $\beta_3 < 0$ then the effect of remittances on violence is decreasing as democracy levels increase. Likewise depending on the signs and magnitudes of these βs it might be that the marginal effect is positive under some values of democracy and negative under others. Note that β_1 here now reflects the effect of remittances when the democracy measure is 0. If democracy is a dummy, this is the marginal effect of remittances within non-democracies.

When the interaction variable is a dummy we often present the marginal effects when the dummy is “active” (democracy) and “inactive” (non-democracy) along with a confidence interval. When the the interaction variable takes on several or many values it is often a good idea to plot the marginal effect of your main variable of interest at these different values of the interaction. A confidence interval of the marginal effect, like in the polynomial case can be found using the variance

$$\begin{aligned}\operatorname{Var}\left(\widehat{D_{r_i}A_i}|d=d^*\right) &= \operatorname{Var}\left(\hat{\beta}_1+\hat{\beta}_3 d^*|X\right)\\ &= \operatorname{Var}\left(\hat{\beta}_1|X\right)+(d^*)^2\operatorname{Var}\left(\hat{\beta}_3|X\right)+2d^*\operatorname{Cov}(\hat{\beta}_1,\hat{\beta}_3|X)\\ &= \begin{bmatrix}0 & 1 & 0 & d^* & 0 & \dots & 0\end{bmatrix}\operatorname{Var}(\hat{\beta}|X)\begin{bmatrix}0 & 1 & 0 & d^* & 0 & \dots & 0\end{bmatrix}'\end{aligned}$$

As in the polynomials CI can be build using either the standard normal depending (large sample) or the t (small sample, homoskedsastic, and normal errors).

5.6.5 Vuong’s Non-nested model test

Comparative model testing a great tool for choosing among competing specifications. We’ve done some of these before in the context of nested models. In those cases we compared a restricted model to an unrestricted model using a Wald test. With non-nested model testing we want to kick that up a notch. Consider two linear models

$$\begin{aligned}y_i &= \beta' x_i+\varepsilon_i\\ h(y_i) &= \gamma' z_i+\xi_i.\end{aligned}$$

where h is some transformation of y (i.e., a log). The densities of y implied by each model are

$$f(y_i|\beta,x_i,z_i)$$

and

$$g(y_i|\gamma,x_i,z_i).$$

We want to know if f or g is a better model of y.

Technical take: The first model is nested within the second model if for all possible values of β we can find a γ such that $f(y_i|\beta,z_i,x_i)=g(y_i|\gamma,z_i,x_i)$ for all (x,z,y). When $h(y_i)=y_i$ then these models are nested if one set of regressors is a subset of the other. They are **non-nested and overlapping** if they have some variables in common, including the constant. In this case there are some β and γ such that $f(y_i|\beta,z_i,x_i)=g(y_i|\gamma,z_i,x_i)$ for some (x,z,y). Finally, models are **strictly non-nested** if they have no variables in common, including the constant.

In this case there are no β and γ such that $f(y_i|\beta, z_i, x_i) = g(y_i|\gamma, z_i, x_i)$ for any (x, z, y). Another way that two models can be strictly non-nested is in terms of h. In this case, models can have the same variables, but in one case y is transformed.

Simple take:

- If the covariates in model 1 are a subset of the covariates in model 2 and they have the exact same dependent variable then model 1 is nested in model 2 (use F or χ^2 tests from before)
- If the covariates in model 1 are not a subset of the covariates in model 2 (and vice-versa), but there are some common covariates, and they have the same dependent variable then the models are overlapping and non-nested
- If none of the covariates in model 1 are in model 2 (and vice-versa) and they have the same dependent variable then the models are strictly non-nested
- If model 1 has dependent variable y and model 2 has dependent variable $h(y)$ then they are strictly non-nested

Let $h(y_i) = \log(y_i)$ and let $x_i = z_i$. Now we've got a case of (strictly) non-nested models even with the same regressors. Notice

$$\begin{aligned}
y_i &= \beta' x_i + \varepsilon_i & \text{Model I: } y_i|x_i \sim N(\beta' x_i, \sigma^2_\varepsilon)\\
\log(y_i) &= \gamma' x_i + \xi_i & \text{Model II: } y_i|x_i \sim \text{Log-Normal}(\beta' x_i, \sigma^2_\xi)\\
y_i &= \exp(\gamma' x_i + \xi_i) & \\
&= e^{\gamma' x_i} e^{\xi_i} &
\end{aligned}$$

These two models are strictly non-nested despite containing the same covariates because they have completely different likelihoods. Model I has an additive error term, while Model II is multiplicative.

Vuong (1989) proposes a two-part test. First, he proposes a test for whether the models are distinguishable. Based on the results of that test, there are different tests for model comparison. The first test of whether the models overlapping or not is a bit cumbersome; we'll skip the derivation.

(Technical details; skip and hit the application) However, if we reject the null of indistinguishable models we can build Vuong's non-nested test based on the what's known as the likelihood ratio:

$$LR(\hat{\beta}, \hat{\gamma}) = \sum_{i=1}^{N} \log\left(\frac{f(y_i|x_i, \hat{\beta})}{g(y_i|x_i, \hat{\gamma})}\right).$$

Notice that if f is the better model then we would expect that $f(y_i|x_i, \hat{\beta})$ would be greater than $g(y_i|x_i, \hat{\gamma})$ on average (bigger likelihood implies better fit). The null hypothesis in question then becomes

$$\begin{aligned} H_0 &: \mathrm{E}[LR(\hat{\beta}, \hat{\gamma})] = 0 \\ H_f &: \mathrm{E}[LR(\hat{\beta}, \hat{\gamma})] > 0 \quad \text{Model I is better} \\ H_g &: \mathrm{E}[LR(\hat{\beta}, \hat{\gamma})] < 0 \quad \text{Model II is better} \end{aligned}$$

Note that this means we will be conducting an unusual two-sided hypothesis test to see which model 'wins.' Further note that both models can be terrible, but the Vuong test helps us see which one is "closer" to the truth. To form the test statistic we get a familiar-ish setup

$$V = \frac{LR(\hat{\beta}, \hat{\gamma}) - 0}{\sqrt{N}\hat{\omega}},$$

where $\hat{\omega}$ is the estimated variance of the LR such that

$$\hat{\omega}^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\log\left(\frac{f(y_i|x_i, \hat{\beta})}{g(y_i|x_i, \hat{\gamma})}\right)\right)^2 - \left(\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{f(y_i|x_i, \hat{\beta})}{g(y_i|x_i, \hat{\gamma})}\right)\right)^2.$$

The main result from Vuong (1989) is that for if $\omega > 0$ then $V \xrightarrow{d} N(0, 1)$ under the null. The first test is whether $\omega = 0$, if we fail that test, then there is not enough variance in the LR to move onto the second test. If H_f is true than $V \xrightarrow{p} \infty$, while under H_g ,$V \xrightarrow{p} -\infty$. If V is greater than a critical z^* we conclude that Model 1 is better, and if V is less than $-z^*$ we conclude Model II is better. So if we're testing at the 5% level, we set $z^* = 1.96$ and compare. Note that if V is between these cutpoints, we can't conclude that either is better.

Finally, note that if the models "fail" the initial test, then we conclude that we can't use data to tell them apart. There are workarounds if you make some different assumptions, but we're going to skip that for now.

5.7 Applications

We did a lot of work there. Let's put it to use with some careful data analysis.

```
library(readstata13)
library(lmtest)
library(sandwich)
```

```r
library(car)
library(nonnest2)
library(margins)
rm(list=ls())
shady <- read.dta13("Rcode/datasets/shady.dta")

shady$salepric <- shady$salepric/10000 # Tens of thousands
model1 <- lm(salepric~cond+grade+aircon+fireplc+bedroom+fullbth
             +halfbth+live_fsq,
             data=shady, x=TRUE)
model1.log <- update(model1, log(salepric)~.)

round(coeftest(model1, vcov=vcovHC), 4)
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.6559     1.2556 14.0615   <2e-16 ***
## cond         -2.5317     0.2723 -9.2986   <2e-16 ***
## grade        -2.6301     0.3261 -8.0652   <2e-16 ***
## aircon        1.6970     0.4605  3.6852   0.0002 ***
## fireplc       3.0089     0.6252  4.8126   <2e-16 ***
## bedroom       0.4865     0.2923  1.6644   0.0963 .
## fullbth       0.2146     0.4252  0.5049   0.6138
## halfbth       2.7852     0.5041  5.5248   <2e-16 ***
## live_fsq      0.0041     0.0004  9.5678   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
round(coeftest(model1.log, vcov=vcovHC),4) # beta.hat * 100 % increase
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.8197     0.1039 27.1455   <2e-16 ***
## cond         -0.1809     0.0216 -8.3725   <2e-16 ***
## grade        -0.2465     0.0269 -9.1756   <2e-16 ***
## aircon        0.0974     0.0366  2.6586    8e-03 **
## fireplc       0.1876     0.0419  4.4745   <2e-16 ***
## bedroom       0.0716     0.0200  3.5745    4e-04 ***
## fullbth       0.0557     0.0284  1.9623    5e-02 *
## halfbth       0.1881     0.0372  5.0512   <2e-16 ***
## live_fsq      0.0002     0.0000  6.4191   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#test for heteroskedasticity

bptest(model1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 83.552, df = 8, p-value = 9.398e-15
```

```r
bptest(model1.log)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1.log
## BP = 19.86, df = 8, p-value = 0.01088
```

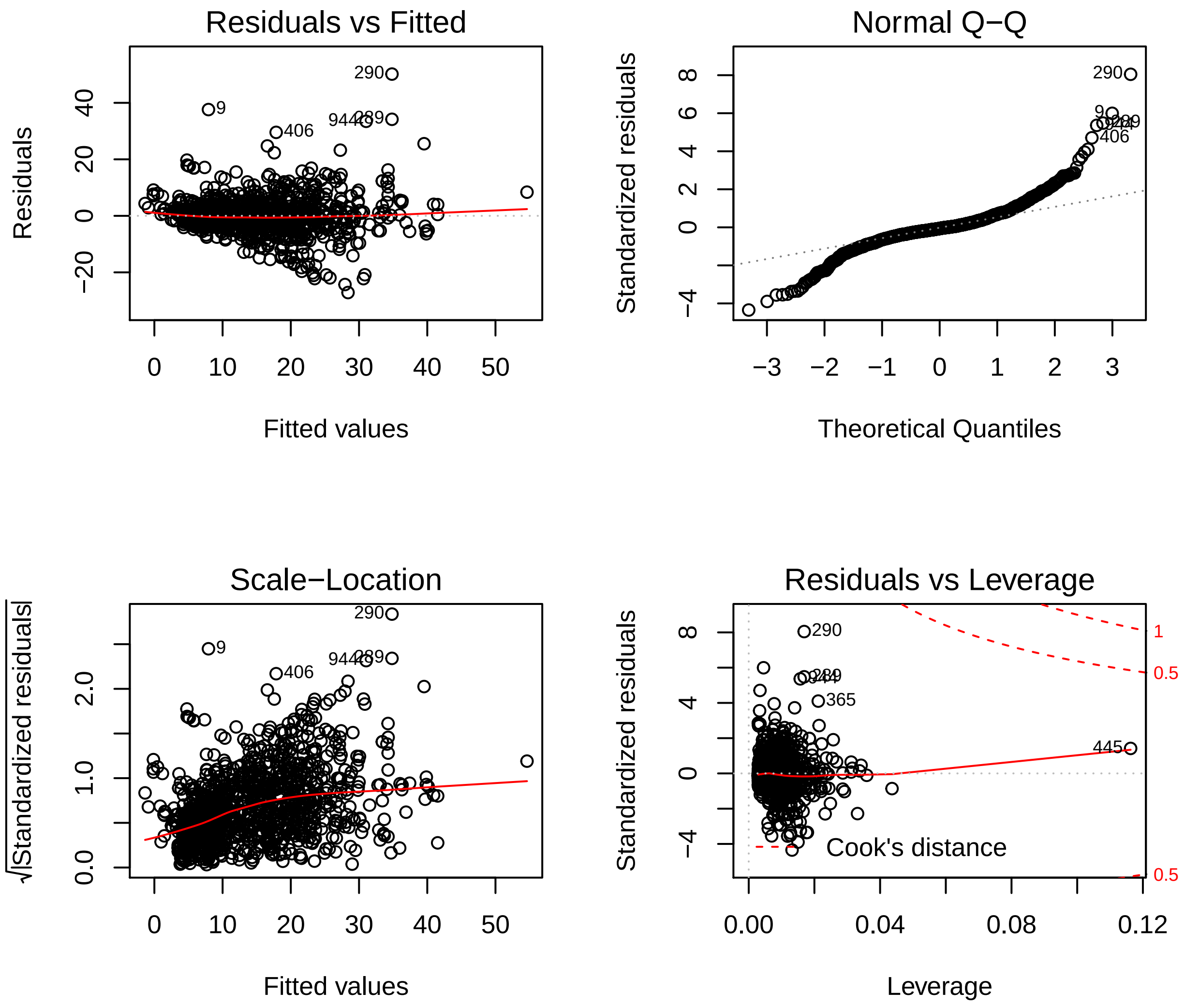
```r
#Shapiro test for normality
shapiro.test(model1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model1$residuals
## W = 0.88245, p-value < 2.2e-16
```

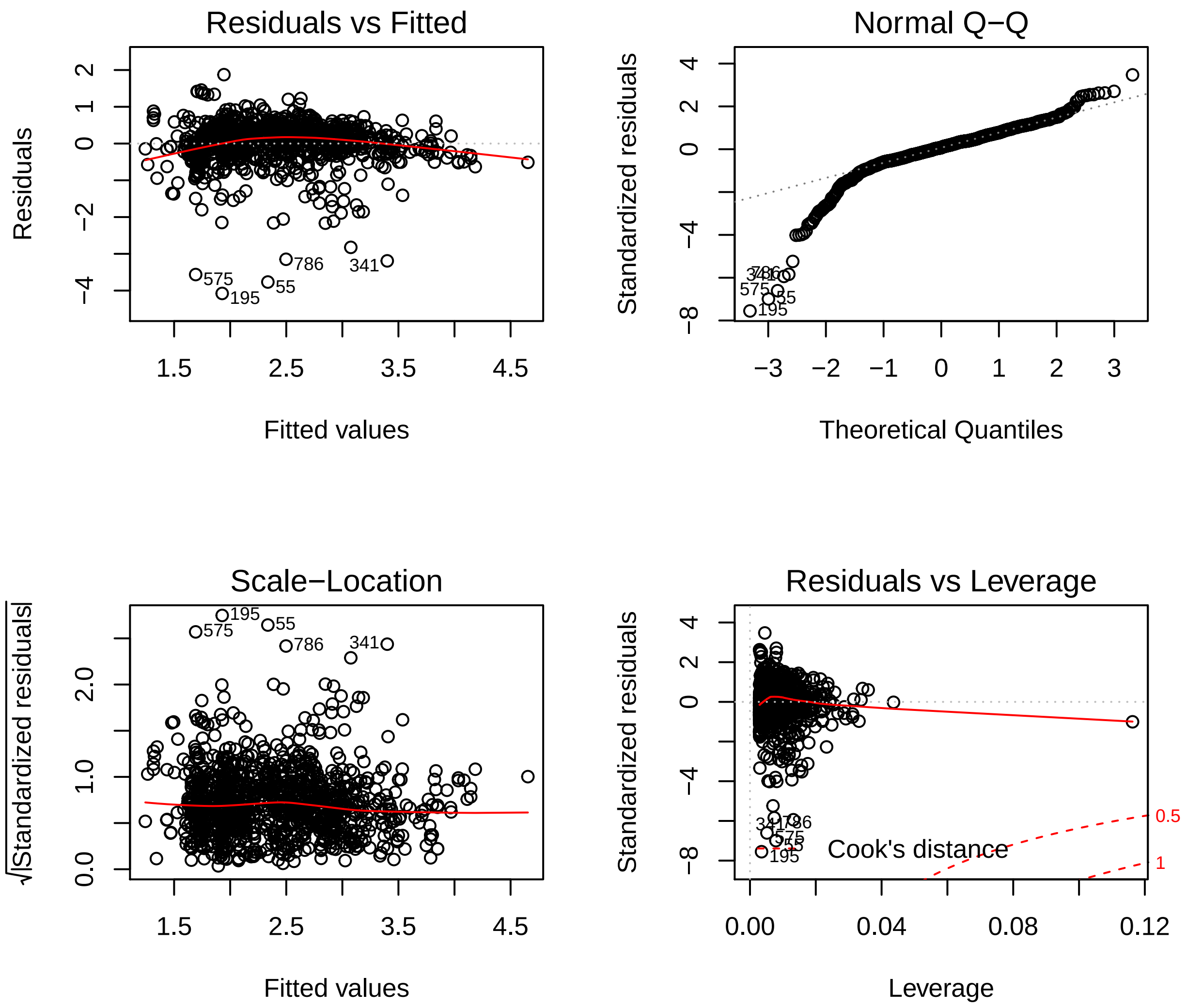
```
shapiro.test(model1.log$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model1.log$residuals
## W = 0.86064, p-value < 2.2e-16
```

```
#visual
par(mfrow=c(2,2))
plot(model1, id.n=5)
```



```
plot(model1.log, id.n=5)
```



```
## What about dummies for the discrete conditions and grade?
table(shady$grade)
```

```
##
##   1   2   3   4
## 129 441 500  27
```

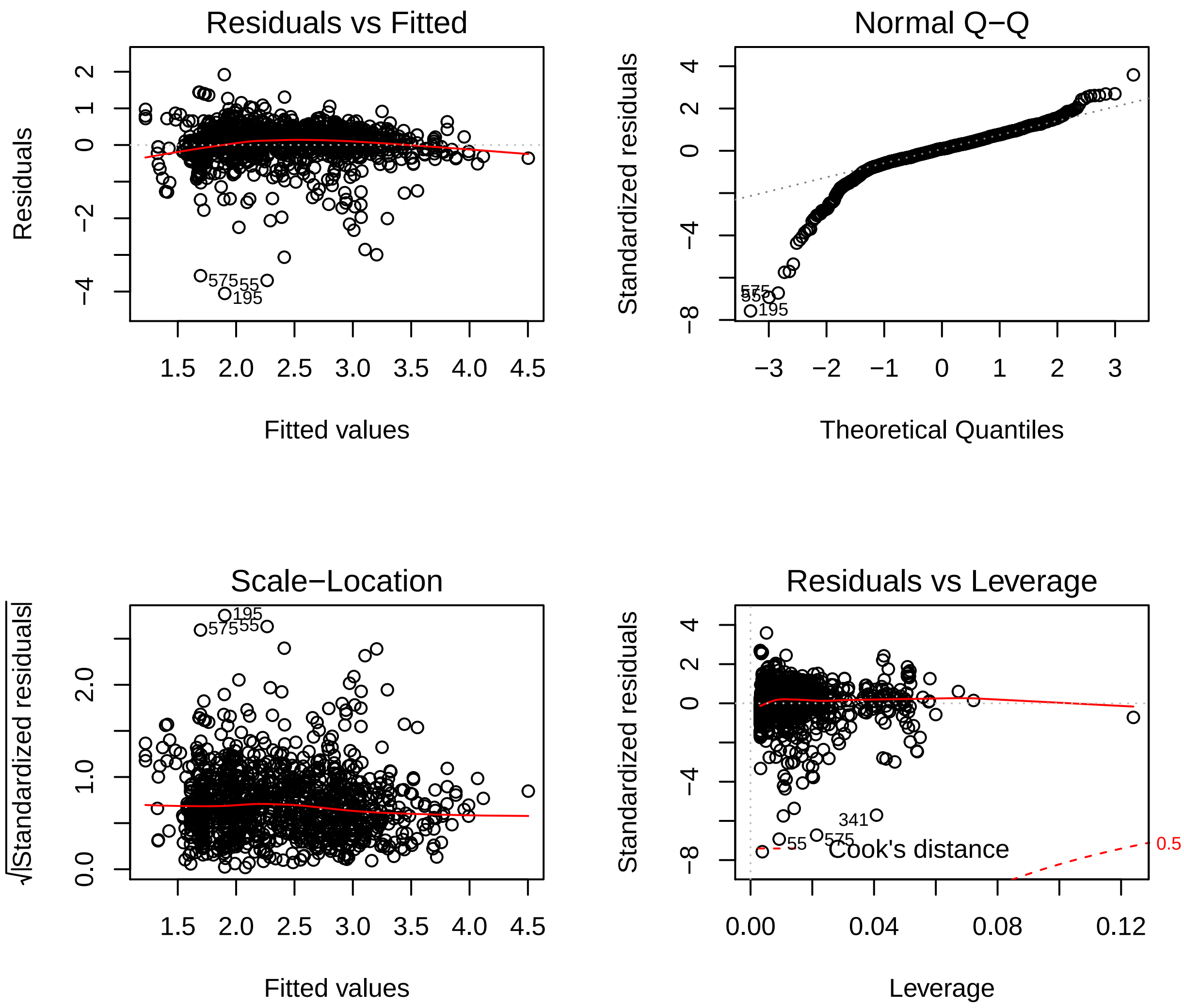
```
table(shady$cond)
```

```
##
##   1   2   3   4   5   6
##  29 143 205 626  63  31
```

```
model1.log.dummies <- update(model1.log,
                             . ~ . -cond+factor(cond)-grade+factor(grade))
round(coeftest(model1.log.dummies, vcov=vcovHC), 4)
```

```
##
## t test of coefficients:
##
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       2.1145     0.1486 14.2270   <2e-16 ***
## aircon            0.1194     0.0363  3.2874   0.0010 ***
## fireplc           0.1785     0.0419  4.2629   <2e-16 ***
## bedroom           0.0601     0.0206  2.9203   0.0036 ** 
## fullbth           0.0639     0.0290  2.2024   0.0278 *  
## halfbth           0.1668     0.0382  4.3662   <2e-16 ***
## live_fsq          0.0002     0.0000  6.4661   <2e-16 ***
## factor(cond)2     0.0429     0.1410  0.3046   0.7607    
## factor(cond)3    -0.0381     0.1412 -0.2699   0.7873    
## factor(cond)4    -0.3736     0.1370 -2.7276   0.0065 ** 
## factor(cond)5    -0.4840     0.1712 -2.8262   0.0048 ** 
## factor(cond)6    -0.4786     0.2004 -2.3889   0.0171 *  
## factor(grade)2   -0.1255     0.0489 -2.5636   0.0105 *  
## factor(grade)3   -0.4261     0.0560 -7.6028   <2e-16 ***
## factor(grade)4   -0.7790     0.1867 -4.1733   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(model1.log.dummies)
```



```r
## definitely some differing effects in conditions
## Some are positive and some are negative relative to the excluded.
## For grade the prices may be decreasing at a linear rate?

linearHypothesis(model1.log.dummies, c("factor(cond)2-factor(cond)3=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)2 - factor(cond)3 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
```

```
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1   1083
## 2   1082  1 2.1025 0.1473
```

```
linearHypothesis(model1.log.dummies, c("factor(cond)2-factor(cond)4=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)2 - factor(cond)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1   1083
## 2   1082  1 63.596 3.862e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.dummies, c("factor(cond)2-factor(cond)5=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)2 - factor(cond)5 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
```

```
##
##   Res.Df Df      F    Pr(>F)
## 1   1083
## 2   1082  1 21.834 3.347e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.dummies, c("factor(cond)2-factor(cond)6=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)2 - factor(cond)6 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1   1083
## 2   1082  1 11.527 0.0007108 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.dummies, c("factor(cond)3-factor(cond)4=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)3 - factor(cond)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
```

```
##
##   Res.Df Df      F    Pr(>F)
## 1   1083
## 2   1082  1 47.279 1.039e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model1.log.dummies, c("factor(cond)3-factor(cond)5=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)3 - factor(cond)5 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1   1083
## 2   1082  1 15.984 6.823e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model1.log.dummies, c("factor(cond)3-factor(cond)6=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)3 - factor(cond)6 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
```

```
## 
##   Res.Df Df      F   Pr(>F)   
## 1   1083                      
## 2   1082  1 8.2791 0.004089 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.dummies, c("factor(cond)4-factor(cond)5=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
## 
## Hypothesis:
## factor(cond)4 - factor(cond)5 = 0
## 
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth + 
##     live_fsq + factor(cond) + factor(grade)
## 
## Note: Coefficient covariance matrix supplied.
## 
##   Res.Df Df      F Pr(>F)
## 1   1083                 
## 2   1082  1 0.9695  0.325
```

```
linearHypothesis(model1.log.dummies, c("factor(cond)4-factor(cond)6=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
## 
## Hypothesis:
## factor(cond)4 - factor(cond)6 = 0
## 
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth + 
##     live_fsq + factor(cond) + factor(grade)
## 
## Note: Coefficient covariance matrix supplied.
## 
##   Res.Df Df      F Pr(>F)
## 1   1083
## 2   1082  1 0.4806 0.4883
```

```
linearHypothesis(model1.log.dummies, c("factor(cond)5-factor(cond)6=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(cond)5 - factor(cond)6 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1   1083
## 2   1082  1 0.0011 0.9736
```

```
## No differences between
# 1&2
# 1&3
# 2&3
# 4&5
# 4&6
# 5&6
# significant differences in price for the other groups

#differences across grades?
linearHypothesis(model1.log.dummies, c("factor(grade)2-factor(grade)3=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(grade)2 - factor(grade)3 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df    F    Pr(>F)
## 1   1083
## 2   1082  1 62.1 7.918e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.dummies, c("factor(grade)2-factor(grade)4=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(grade)2 - factor(grade)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1   1083
## 2   1082  1 13.802 0.0002136 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.dummies, c("factor(grade)3-factor(grade)4=0"), vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(grade)3 - factor(grade)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F  Pr(>F)
## 1   1083
## 2   1082  1 4.0684 0.04394 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# significant differences between all grades

# IS the jump about the same across grades? (nonlinear trend?)
# is the gap between 1 and 3 equal to 2 gaps from 1-2?
# is the gap between 1 and 4 equal to 3 gaps from 1-2?
linearHypothesis(model1.log.dummies, c("factor(grade)3-2*factor(grade)2=0"), vcov=vcovHC
```

```
## Linear hypothesis test
##
## Hypothesis:
## - 2 factor(grade)2  + factor(grade)3 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F   Pr(>F)
## 1   1083
## 2   1082  1 6.7298 0.009609 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.dummies, c("factor(grade)4-3*factor(grade)2=0"), vcov=vcovHC
```

```
## Linear hypothesis test
```

```
##
## Hypothesis:
## - 3 factor(grade)2  + factor(grade)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F  Pr(>F)
## 1   1083
## 2   1082  1 4.3193 0.03792 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

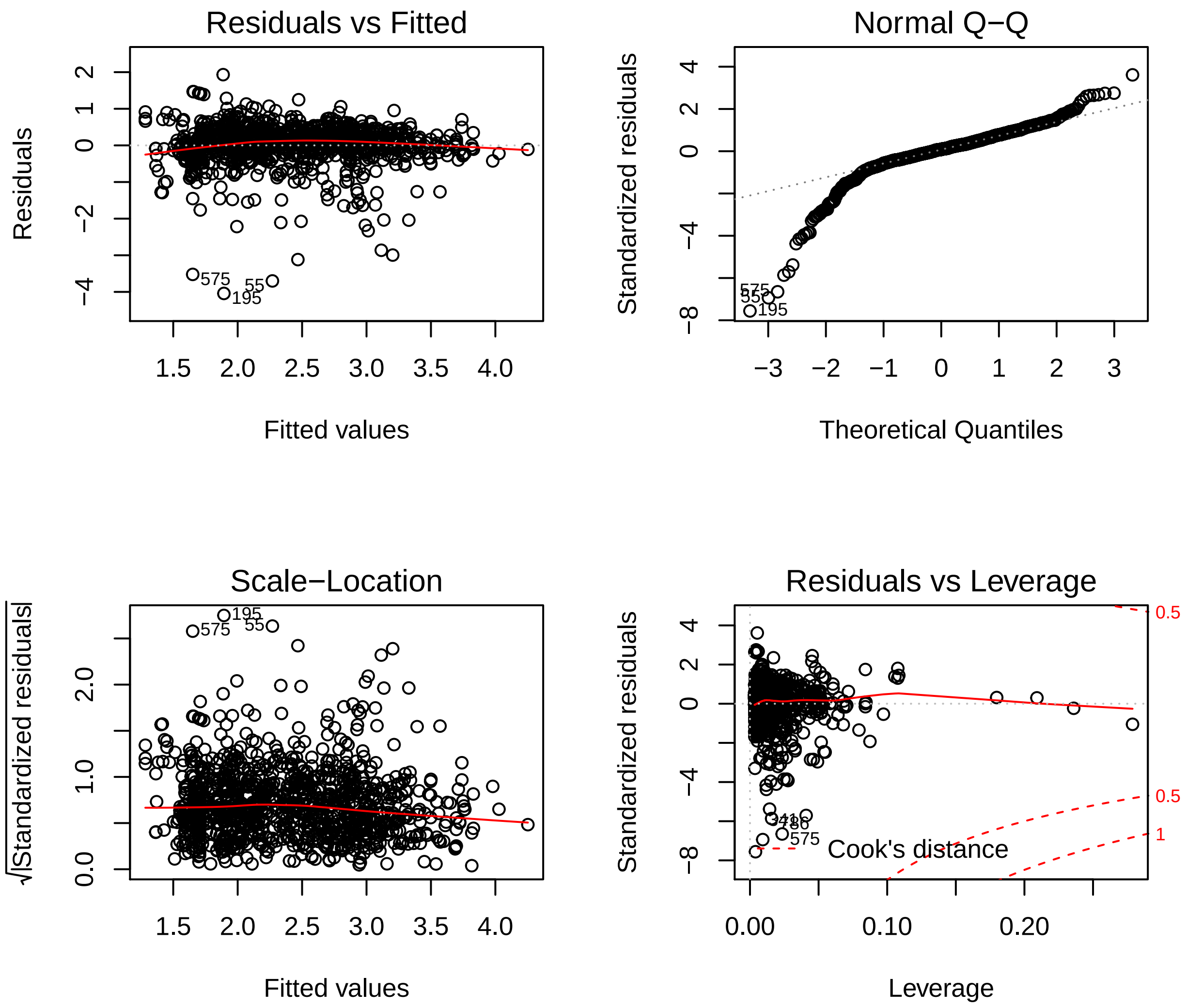
```
# OR

#Is the gap from 3-4 equal to the gap between 1-2?
linearHypothesis(model1.log.dummies, c("factor(grade)4-factor(grade)3-factor(grade)2=0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## - factor(grade)2 - factor(grade)3  + factor(grade)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1   1083
## 2   1082  1 1.6795 0.1953
```

```
#Is the gap from 3-4 equal to the gap between 2-3?
linearHypothesis(model1.log.dummies, c("factor(grade)4-factor(grade)3=factor(grade)3-fac
```

```
## Linear hypothesis test
##
## Hypothesis:
## factor(grade)2 - 2 factor(grade)3  + factor(grade)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1   1083
## 2   1082  1 0.0822 0.7744
```

```
# If nothing else, there is something btwn 2-3 that doesn't look like 1-2
# some evidence for nonlinearity in the first set, but not across all categories
# Still it seems that letting the dummies do their thing is valuable flexibility

###### INTERACTIONS #####
# What about a model that interacts the living space with with the grade?

model1.log.interactions <- update(model1.log.dummies,
                                  . ~ . + factor(grade):live_fsq)
plot(model1.log.interactions)
```



```r
round(coeftest(model1.log.interactions, vcov=vcovHC),4)
```

```
##
## t test of coefficients:
##
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.2363     0.1447 15.4567   <2e-16 ***
## aircon                      0.1251     0.0365  3.4263   0.0006 ***
## fireplc                     0.1838     0.0421  4.3682   <2e-16 ***
## bedroom                     0.0524     0.0208  2.5213   0.0118 *
## fullbth                     0.0585     0.0298  1.9596   0.0503 .
## halfbth                     0.1707     0.0382  4.4692   <2e-16 ***
## live_fsq                    0.0001     0.0000  5.9146   <2e-16 ***
## factor(cond)2               0.0447     0.1414  0.3161   0.7520
## factor(cond)3            -0.0435     0.1419 -0.3066   0.7592
## factor(cond)4            -0.3684     0.1374 -2.6804   0.0075 **
## factor(cond)5            -0.5139     0.1767 -2.9084   0.0037 **
## factor(cond)6            -0.5006     0.2079 -2.4077   0.0162 *
## factor(grade)2           -0.2544     0.0754 -3.3717   0.0008 ***
## factor(grade)3           -0.6136     0.0966 -6.3492   <2e-16 ***
## factor(grade)4           -0.7123     0.4724 -1.5079   0.1319
## live_fsq:factor(grade)2   0.0001     0.0000  1.6054   0.1087
## live_fsq:factor(grade)3   0.0001     0.0001  1.6538   0.0985 .
## live_fsq:factor(grade)4   0.0000     0.0002 -0.1671   0.8674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.interactions,
                 "live_fsq+live_fsq:factor(grade)2 =0", vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## live_fsq  + live_fsq:factor(grade)2 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade) + live_fsq:factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1   1080
## 2   1079  1 31.763 2.223e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.interactions,
                 "live_fsq+live_fsq:factor(grade)3 =0", vcov=vcovHC)
```

```
## Linear hypothesis test
##
```

```
## Hypothesis:
## live_fsq  + live_fsq:factor(grade)3 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade) + live_fsq:factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df     F    Pr(>F)
## 1   1080
## 2   1079  1 16.19 6.129e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1.log.interactions,
                 "live_fsq+live_fsq:factor(grade)4 =0", vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## live_fsq  + live_fsq:factor(grade)4 = 0
##
## Model 1: restricted model
## Model 2: log(salepric) ~ aircon + fireplc + bedroom + fullbth + halfbth +
##     live_fsq + factor(cond) + factor(grade) + live_fsq:factor(grade)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1   1080
## 2   1079  1 0.1868 0.6657
```

```
# # MARGINAL EFFECTS "BY HAND"
# names <- c("live_fsq",
#             "live_fsq:factor(grade)2",
#             "live_fsq:factor(grade)3",
#             "live_fsq:factor(grade)4")
```

```r
# D <- rbind(c(1, 0,0,0),
#            cbind(1, diag(3)))
# D
# b.needed <- model1.log.interactions$coef[names]
# ME <- D  %*% b.needed
# V <- vcovHC(model1.log.interactions)[names, names]
# VME <- D %*% V %*% t(D)
# CI.ME <- 1.96 * sqrt(diag(VME))
# plotME <- data.frame(ME=ME*100,
#                      lo=100*(ME-CI.ME),
#                      hi=100*(ME+CI.ME),
#                      grade=1:4)


Marginals <- margins(model1.log.interactions,
                     variables = "live_fsq",
                     at=data.frame(grade=c(1:4)),
                     vcov=vcovHC(model1.log.interactions))
summary(Marginals)
```

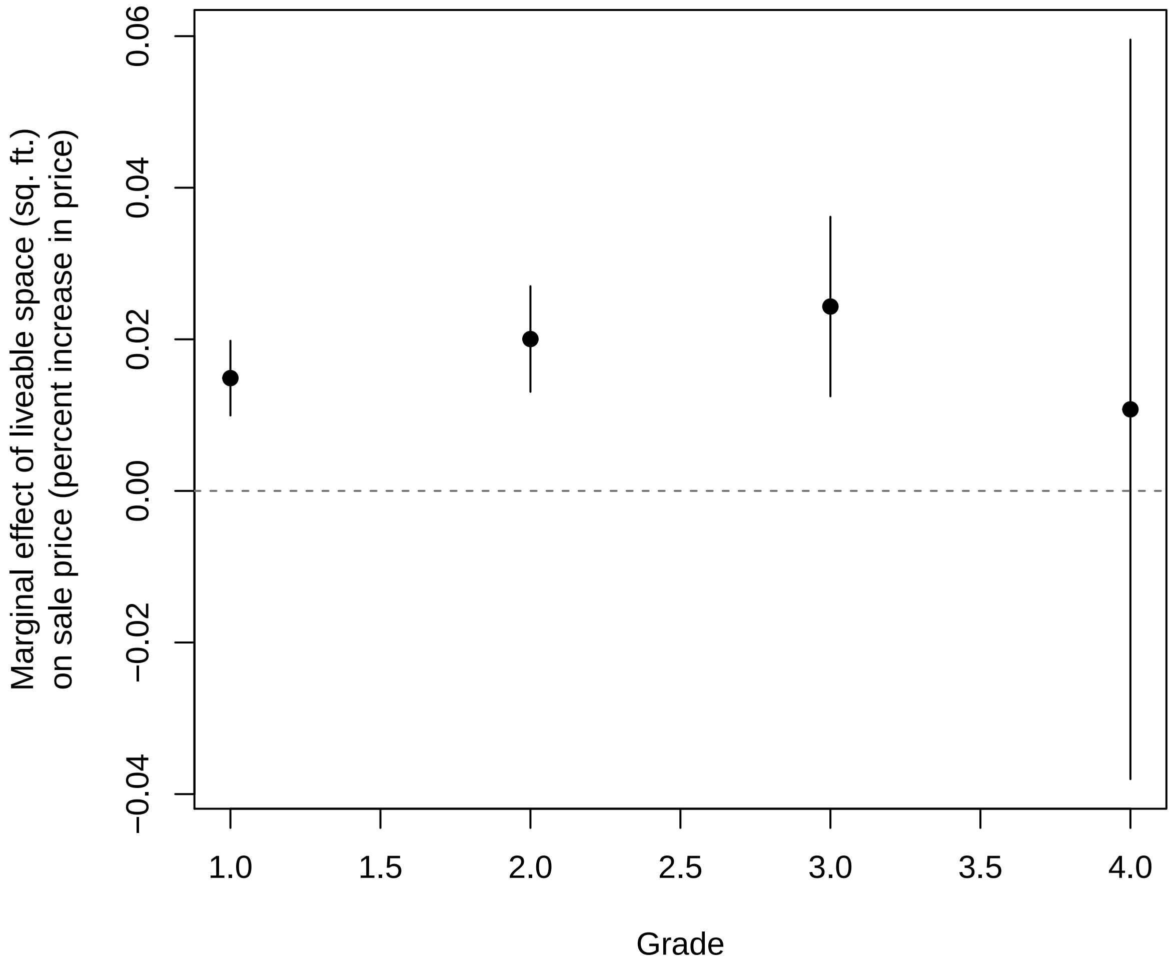
```
##    factor  grade    AME     SE      z      p   lower  upper
##  live_fsq 1.0000 0.0001 0.0000 5.9146 0.0000  0.0001 0.0002
##  live_fsq 2.0000 0.0002 0.0000 5.6358 0.0000  0.0001 0.0003
##  live_fsq 3.0000 0.0002 0.0001 4.0237 0.0001  0.0001 0.0004
##  live_fsq 4.0000 0.0001 0.0002 0.4322 0.6656 -0.0004 0.0006
```

```r
class(summary(Marginals))
```

```
## [1] "summary.margins" "data.frame"
```

```r
plotME <- summary(Marginals)
# NOTE WHAT MARGINS DOESN'T DO IS GIVE US PERCENTAGES
# WE NEED TO MULIPLY BY 100 OURSELVES
plotME$AME <- plotME$AME*100
plotME$SE <- plotME$SE*100
plotME$lower  <- plotME$lower*100
plotME$upper  <- plotME$upper*100
```

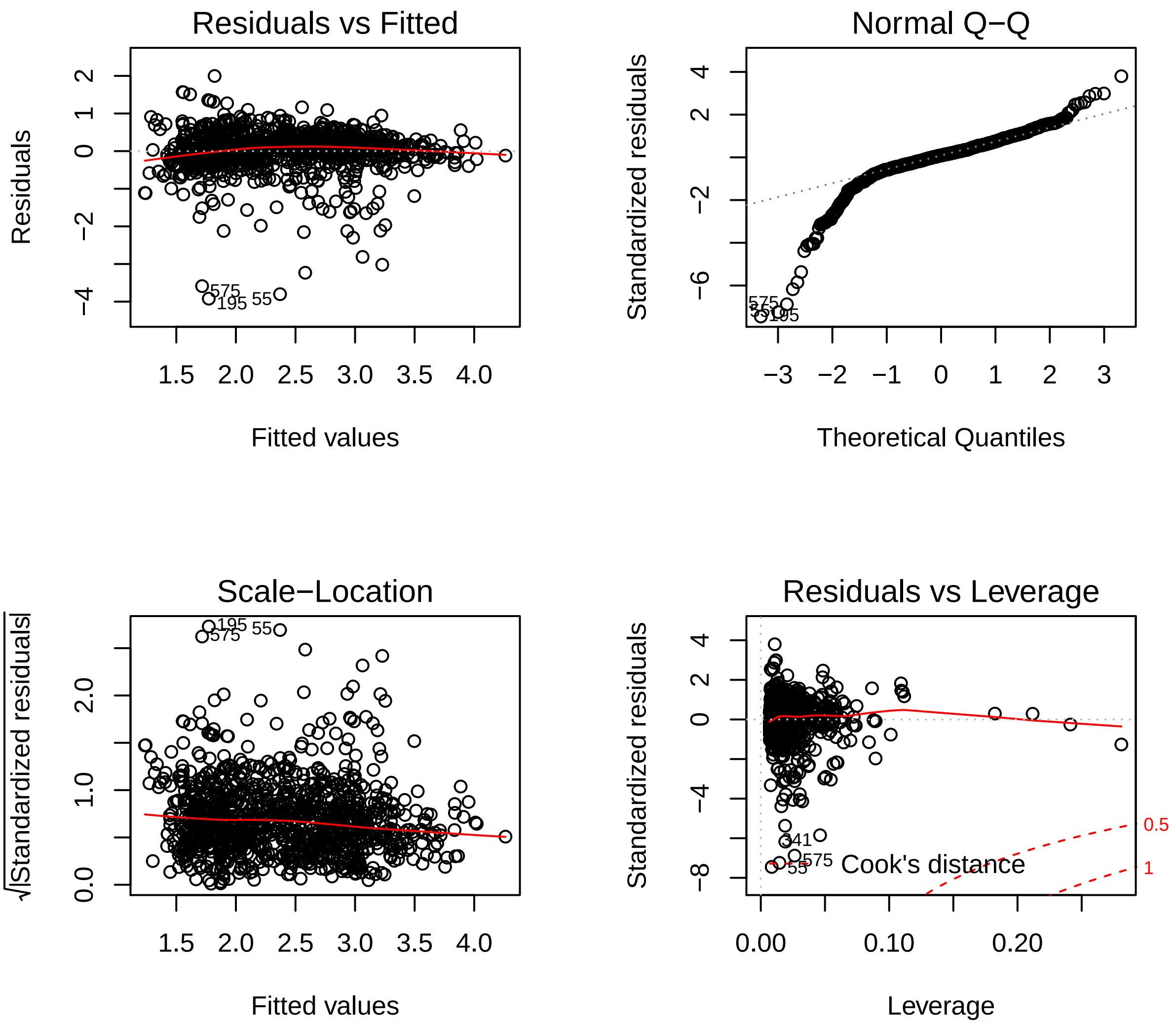
```
#For adjusting the margins the order
#bottom, left, top, right
#default values are c(5, 4, 4, 2) +.1
#see ?par for more details
par(mfrow=c(1,1), mar=c(5, 5, 4, 2)+.1)
plot(AME~grade, data=plotME,
     ylim=c(min(plotME$lower),max(plotME$upper)),
     pch=19,
     ylab="Marginal effect of liveable space (sq. ft.)\non sale price (percent increase
     xlab="Grade")
segments(y0=plotME$lower, y1=plotME$upper, x0=plotME$grade)
abline(h=0, col="grey45", lty="dashed")
```



```
## finally what about some time effects
model1.log.time <- update(model1.log.interactions,
                          . ~ . +yearblt+ factor(saleyr))
round(coeftest(model1.log.time, vcov=vcovHC),4)
```

```
##
## t test of coefficients:
##
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.9648     1.0870 -0.8876   0.3750
## aircon                    0.1033     0.0373  2.7725   0.0057 **
## fireplc                   0.1970     0.0433  4.5539   <2e-16 ***
## bedroom                   0.0616     0.0214  2.8859   0.0040 **
## fullbth                   0.0337     0.0299  1.1268   0.2601
## halfbth                   0.1448     0.0370  3.9105   0.0001 ***
## live_fsq                  0.0002     0.0000  7.8283   <2e-16 ***
## factor(cond)2             0.0062     0.1458  0.0427   0.9659
## factor(cond)3            -0.0746     0.1458 -0.5114   0.6092
## factor(cond)4            -0.3829     0.1405 -2.7257   0.0065 **
## factor(cond)5            -0.5479     0.1796 -3.0504   0.0023 **
## factor(cond)6            -0.5197     0.2104 -2.4705   0.0136 *
## factor(grade)2           -0.1872     0.0792 -2.3651   0.0182 *
## factor(grade)3           -0.6043     0.0965 -6.2612   <2e-16 ***
## factor(grade)4           -0.5421     0.4773 -1.1357   0.2563
## yearblt                   0.0016     0.0006  2.8248   0.0048 **
## factor(saleyr)1998        0.1086     0.0459  2.3664   0.0181 *
## factor(saleyr)1999        0.1844     0.0485  3.8010   0.0002 ***
## factor(saleyr)2000        0.2538     0.0502  5.0603   <2e-16 ***
## factor(saleyr)2001        0.1293     0.0640  2.0204   0.0436 *
## live_fsq:factor(grade)2   0.0000     0.0000  0.8220   0.4113
## live_fsq:factor(grade)3   0.0001     0.0001  1.5895   0.1122
## live_fsq:factor(grade)4  -0.0001     0.0003 -0.4559   0.6486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(model1.log.time)
```



```
# We can at least see what does better between the normal and the log
vuongtest(model1, model1.log)
```

```
## Warning in imhof(n * omega.hat.2, lamstar^2): Note that Qq + abserr is positive.

##
## Model 1
##  Class: lm
##  Call: lm(formula = salepric ~ cond + grade + aircon + fireplc + bedroom + ...
##
## Model 2
##  Class: lm
##  Call: lm(formula = log(salepric) ~ cond + grade + aircon + fireplc + ...
##
## Variance test
##   H0: Model 1 and Model 2 are indistinguishable
##   H1: Model 1 and Model 2 are distinguishable
##     w2 = 3.910,   p = <2e-16
##
## Non-nested likelihood ratio test
##   H0: Model fits are equal for the focal population
##   H1A: Model 1 fits better than Model 2
##     z = -41.095,   p = 1
##   H1B: Model 2 fits better than Model 1
##     z = -41.095,   p = < 2.2e-16
```

```
vuongtest(update(model1.log.interactions, salepric ~ .),
          model1.log.interactions)
```

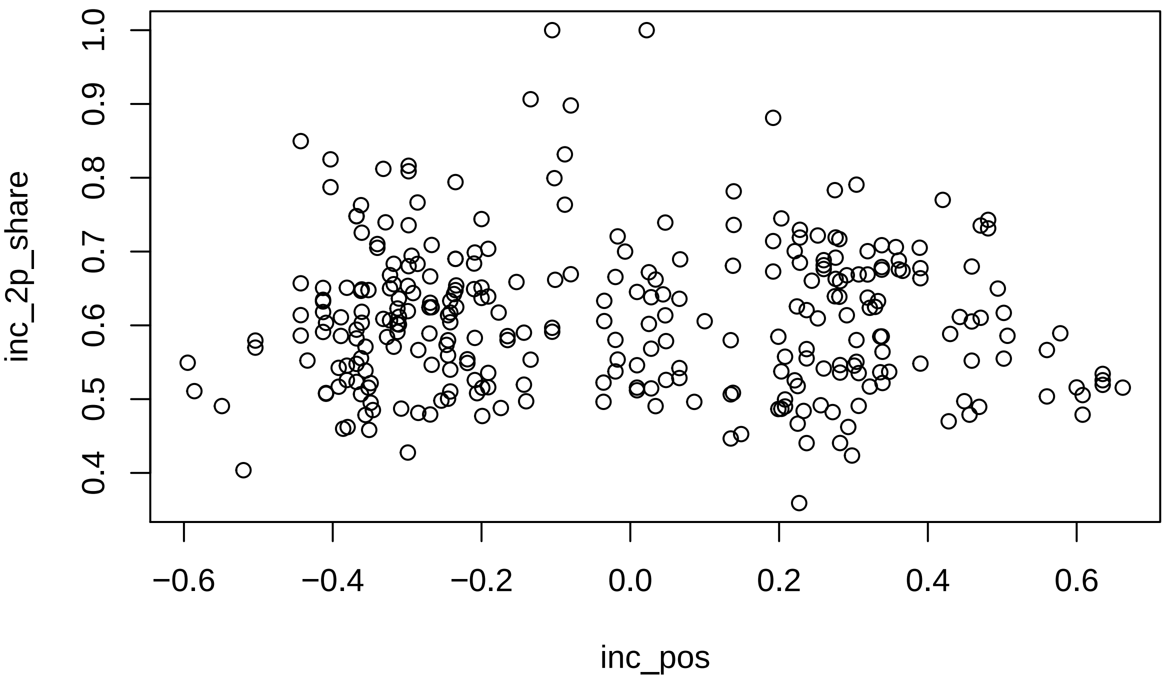
```
##
## Model 1
##  Class: lm
##  Call: lm(formula = salepric ~ aircon + fireplc + bedroom + fullbth + ...
##
## Model 2
##  Class: lm
##  Call: lm(formula = log(salepric) ~ aircon + fireplc + bedroom + fullbth + ...
##
## Variance test
##   H0: Model 1 and Model 2 are indistinguishable
##   H1: Model 1 and Model 2 are distinguishable
##     w2 = 3.697,   p = 1.92e-08
##
## Non-nested likelihood ratio test
##   H0: Model fits are equal for the focal population
##   H1A: Model 1 fits better than Model 2
##     z = -41.968,   p = 1
##   H1B: Model 2 fits better than Model 1
##     z = -41.968,   p = < 2.2e-16
```

That was fun let's try another one. Let's consider the relationship between a U.S. Senator's ideolgoy and vote share. There's a line of thought that suggests more moderate senators should acheive higher vote shares. We can check it out.

```r
library(readstata13)
library(lmtest)
library(sandwich)
library(car)
library(margins)
rm(list=ls())


senate.data <- read.dta13("Rcode/datasets/senate_expanded.dta")
# print(cbind(colnames(senate.data), attributes(senate.data)$var.labels))
head(senate.data)
```

```
##   year st_abr    st_name st_code st_south   st_pop st_uemp inc_icpsr inc_pos
## 1 1980     AK     Alaska      81        0   401851     9.4     12105  -0.285
## 2 1980     AL    Alabama      41        1  3893888     8.4     14711  -0.226
## 3 1980     AR   Arkansas      42        1  2286435     7.5     14300  -0.313
## 4 1980     AZ    Arizona      61        0  2718215     6.7      3658   0.608
## 5 1980     CA California      71        0 23667902     6.9     12103  -0.409
## 6 1980     CO   Colorado      62        0  2889964     5.9     14305  -0.409
##   inc_spend ch_qual ch_spend ch_wealthy inc_2p_share st_id inc_tenure inc_rep
## 1        NA      NA       NA         NA           NA    NA         12       0
## 2        NA      NA       NA         NA           NA  26.2          2       0
## 3    220861       0   119196          0    0.5911147  11.5          6       0
## 4    949992       0  2085242          1    0.5054943  23.4         12       1
## 5   2823607       0  1152272          0    0.6033472   7.4         12       0
## 6   1142304       3  1085205          0    0.5082657  15.8          6       0
```

```r
plot(inc_2p_share~inc_pos, data=senate.data) #positives are more conservative
```



```
#suspect a polynomial, but linear, squared, or cubic?

model0 <- lm(inc_2p_share~inc_pos, data=senate.data)
model1 <- lm(inc_2p_share~inc_pos+I(inc_pos^2), data=senate.data)
model1a <- lm(inc_2p_share~inc_pos+I(inc_pos^2)+I(inc_pos^3), data=senate.data)

which.max(c(summary(model0)$adj.r.squared,
            summary(model1)$adj.r.squared,
            summary(model1a)$adj.r.squared))
```

```
## [1] 2
```

```
which.min(c(AIC(model0), AIC(model1), AIC(model1a)))
```

```
## [1] 2
```

```
which.min(c(BIC(model0), BIC(model1), BIC(model1a)))
```

```
## [1] 2
```

```
waldtest(model0, model1, vcov=vcovHC)
```

```
## Wald test
##
## Model 1: inc_2p_share ~ inc_pos
## Model 2: inc_2p_share ~ inc_pos + I(inc_pos^2)
##   Res.Df Df      F   Pr(>F)   
## 1    294                      
## 2    293  1 8.0501 0.004868 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
waldtest(model0, model1a, vcov=vcovHC)
```

```
## Wald test
## 
## Model 1: inc_2p_share ~ inc_pos
## Model 2: inc_2p_share ~ inc_pos + I(inc_pos^2) + I(inc_pos^3)
##   Res.Df Df     F  Pr(>F)  
## 1    294                   
## 2    292  2 3.954 0.02021 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
waldtest(model1, model1a, vcov=vcovHC)
```

```
## Wald test
## 
## Model 1: inc_2p_share ~ inc_pos + I(inc_pos^2)
## Model 2: inc_2p_share ~ inc_pos + I(inc_pos^2) + I(inc_pos^3)
##   Res.Df Df      F Pr(>F)
## 1    293                 
## 2    292  1 0.3952 0.5301
```

```r
#having made an initial choice let's check it out
bptest(model1)
```

```
## 
##  studentized Breusch-Pagan test
## 
## data:  model1
## BP = 6.0178, df = 2, p-value = 0.04935
```

```r
shapiro.test(model1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model1$residuals
## W = 0.97703, p-value = 0.000109
```

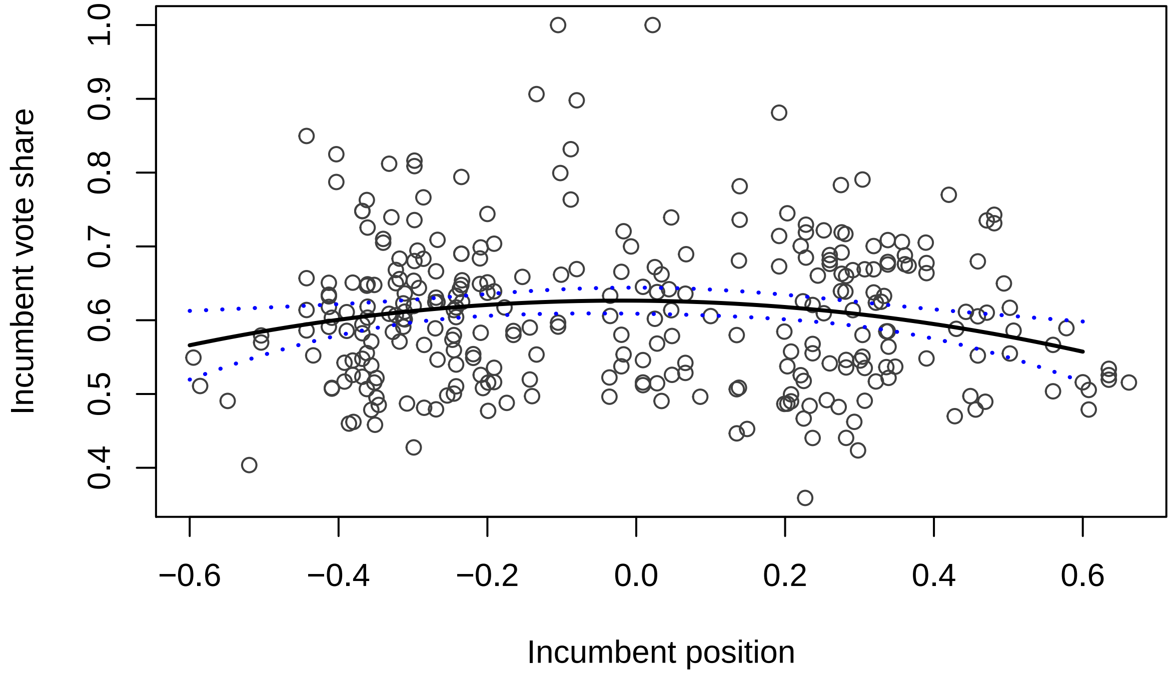
```r
coeftest(model1, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   0.6264730  0.0095611 65.5228 < 2.2e-16 ***
## inc_pos      -0.0072500  0.0174857 -0.4146  0.678719
## I(inc_pos^2) -0.1798957  0.0634045 -2.8373  0.004868 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model1, c("inc_pos=0", "I(inc_pos^2)=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## inc_pos = 0
## I(inc_pos^2) = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ inc_pos + I(inc_pos^2)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F   Pr(>F)
## 1    295
## 2    293  2 5.2744 0.005619 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
newdata <- data.frame(inc_pos=seq(-.6, .6, length=100))
yhat <- predict(model1, newdata = newdata,
                interval = "confidence")
#positives are more conservative
plot(inc_2p_share~inc_pos, data=senate.data,
     xlab="Incumbent position", ylab="Incumbent vote share",
     col="grey25")
lines(y=yhat[,"fit"],x=newdata$inc_pos, lwd=2)
lines(y=yhat[,"lwr"],x=newdata$inc_pos, col="blue", lty="dotted", lwd=2)
lines(y=yhat[,"upr"],x=newdata$inc_pos, col="blue", lty="dotted", lwd=2)
```



```
ME <- margins(model1,
              variables="inc_pos",
              at=list(inc_pos=seq(-0.6, 0.6, by=0.1)),
              vcov=vcovHC(model1))
```

```
## Warning in check_values(data, at): A 'at' value for 'inc_pos' is outside
## observed data range (-0.595000028610229,0.662000000476837)!
```

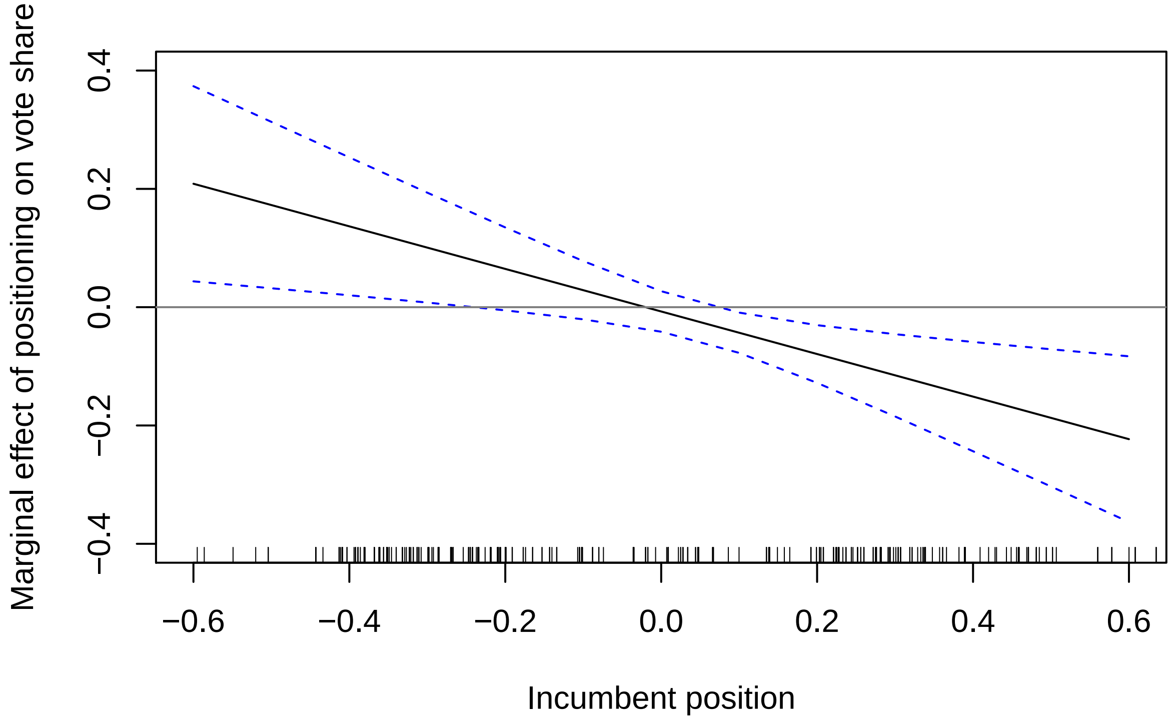
```
summary(ME)
```

```
##    factor inc_pos     AME     SE       z      p   lower   upper
##   inc_pos -0.6000  0.2086 0.0842  2.4790 0.0132  0.0437  0.3736
##   inc_pos -0.5000  0.1726 0.0718  2.4060 0.0161  0.0320  0.3133
##   inc_pos -0.4000  0.1367 0.0595  2.2980 0.0216  0.0201  0.2532
##   inc_pos -0.3000  0.1007 0.0474  2.1242 0.0337  0.0078  0.1936
##   inc_pos -0.2000  0.0647 0.0358  1.8099 0.0703 -0.0054  0.1348
##   inc_pos -0.1000  0.0287 0.0251  1.1435 0.2528 -0.0205  0.0780
##   inc_pos  0.0000 -0.0073 0.0175 -0.4146 0.6784 -0.0415  0.0270
##   inc_pos  0.1000 -0.0432 0.0174 -2.4878 0.0129 -0.0773 -0.0092
##   inc_pos  0.2000 -0.0792 0.0249 -3.1817 0.0015 -0.1280 -0.0304
##   inc_pos  0.3000 -0.1152 0.0355 -3.2461 0.0012 -0.1847 -0.0456
##   inc_pos  0.4000 -0.1512 0.0471 -3.2082 0.0013 -0.2435 -0.0588
##   inc_pos  0.5000 -0.1871 0.0592 -3.1621 0.0016 -0.3031 -0.0711
##   inc_pos  0.6000 -0.2231 0.0715 -3.1222 0.0018 -0.3632 -0.0831
```

```
plot.ME <- summary(ME)
plot(AME~inc_pos,data=plot.ME, type="l",
     ylab="Marginal effect of positioning on vote share", ylim=c(-0.4, 0.4),
     xlab="Incumbent position")
lines(plot.ME$lower, x=plot.ME$inc_pos, type="l", lty="dashed", col="blue")
lines(plot.ME$upper, x=plot.ME$inc_pos, type="l", lty="dashed", col="blue")
rug(jitter(senate.data$inc_pos)) #jitter spreads out clumped data
```

```
## Warning in rug(jitter(senate.data$inc_pos)): some values will be clipped
```

```
abline(h=0,col="grey50")
```



```
#What vote share is the maximum? Is is at inc_pos=0?
# By Hand
# max.hat <- -model1$coef["inc_pos"]/(2*model1$coef["I(inc_pos^2)"])
# V <- vcovHC(model1)
# Dmax <- c(0, #D_\beta0
#           -1/(2*model1$coef["I(inc_pos^2)"]),#D_\beta1
#           model1$coef["inc_pos"]/(2*model1$coef["I(inc_pos^2)"]^2)) #D_\beta2
# se.max.hat <- sqrt(Dmax %*% V %*% Dmax)
# c(max.hat - 1.96*se.max.hat, max.hat+1.96*se.max.hat)

# Using the delta method
# H0: -b1/(2*b2) =0
# HA: -b1/(2*b2) != 0



deltaMethod(model1,
            g="-inc_pos/(2*`I(inc_pos^2)`)", #note the back ticks ``
            vcov=vcovHC,
            rhs=0)
```

```
##                                    Estimate        SE     2.5 %    97.5 %
## -inc_pos/(2 * `I(inc_pos^2)`) -0.020151  0.051658 -0.121399  0.081098
##                                Hypothesis z value Pr(>|z|)
## -inc_pos/(2 * `I(inc_pos^2)`)   0.000000 -0.3901   0.6965
```

```
# Can't reject H0; 95% CI contains 0.




# Can't reject H0



### Let's factor in spending per capita
senate.data$ch_spend_pc <- senate.data$ch_spend/senate.data$st_pop
senate.data$inc_spend_pc <- senate.data$inc_spend/senate.data$st_pop

model2 <- lm(inc_2p_share~inc_pos+I(inc_pos^2)
             +inc_rep +inc_tenure +ch_qual+ st_uemp
             +ch_spend_pc+inc_spend_pc,
             data=senate.data)
coeftest(model2, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                  Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)   0.69217613  0.02572159 26.9103 < 2.2e-16 ***
## inc_pos       0.05351640  0.04214181  1.2699   0.20515
## I(inc_pos^2) -0.14902374  0.06145193 -2.4250   0.01593 *
## inc_rep      -0.04011813  0.02672581 -1.5011   0.13443
## inc_tenure    0.00179475  0.00072305  2.4822   0.01363 *
## ch_qual      -0.01818382  0.00312897 -5.8114 1.650e-08 ***
## st_uemp      -0.00360359  0.00219207 -1.6439   0.10129
## ch_spend_pc  -0.06342234  0.01082199 -5.8605 1.269e-08 ***
## inc_spend_pc  0.01362182  0.00672828  2.0246   0.04384 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# On average, a 1 dollar/person increase in challenger spending
# is associated with a 6.3 percentage point decrease in incumbent party
# vote share, holding everything else constant
# On average, a 1 dollar/person increase in incumbent spending
# is associated with a 1.4 percentage point increase in incumbent party
# vote share, holding everything else constant

# These seem a bit unbelievable, there are probably diminishing
# returns to money maybe those should be logged
senate.data$log_ch_spend_pc <- log(senate.data$ch_spend_pc+1)
senate.data$log_inc_spend_pc <- log(senate.data$inc_spend_pc+1)


model2a <- lm(inc_2p_share~inc_pos+I(inc_pos^2)
              +inc_rep +inc_tenure +ch_qual+ st_uemp
              +log_ch_spend_pc+log_inc_spend_pc,
              data=senate.data)
coeftest(model2a, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                     Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)       0.68995626  0.02625149 26.2826 < 2.2e-16 ***
## inc_pos           0.04613681  0.04069958  1.1336  0.257914
## I(inc_pos^2)     -0.11668730  0.05882707 -1.9836  0.048260 *
## inc_rep          -0.04069692  0.02558691 -1.5905  0.112819
## inc_tenure        0.00155544  0.00067793  2.2944  0.022493 *
## ch_qual          -0.01480001  0.00288155 -5.1361 5.196e-07 ***
## st_uemp          -0.00363742  0.00205431 -1.7706  0.077688 .
## log_ch_spend_pc  -0.16500975  0.01650632 -9.9968 < 2.2e-16 ***
## log_inc_spend_pc  0.05568772  0.01555663  3.5797  0.000404 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# NOTE: the dividing/multiplying by 100 cancel each other out
# in this case because the vote share is proportion (in between 0 and 1)

# On average, a 1% increase in challenger spending
# is associated with a 0.165 percentage point
# decrease in incumbent party
# vote share, holding everything else constant
#   (-0.16500975/100)                * 100
# Interpret logged X (1%) | From prob scale to pp

#####################################################
# On average, a 1% increase in incumbent spending
# is associated with a 0.06 percentage point increase in incumbent party
# vote share, holding everything else constant
#   (0.05568772/100)                * 100
# Interpret logged X (1%) | From prob scale to pp

## Does tenure affect the effectiveness of spending?
model3 <- lm(inc_2p_share~inc_pos+I(inc_pos^2)
             +inc_rep
               +inc_tenure*log_inc_spend_pc
              +inc_tenure*log_ch_spend_pc
             +ch_qual+ st_uemp,
             data=senate.data, x=TRUE)
coeftest(model3, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                 0.6897879  0.0297271 23.2040 < 2.2e-16 ***
## inc_pos                     0.0471371  0.0408860  1.1529   0.24993
## I(inc_pos^2)               -0.1170935  0.0589255 -1.9871   0.04787 *
## inc_rep                    -0.0410795  0.0256949 -1.5987   0.11099
## inc_tenure                  0.0014197  0.0014393  0.9864   0.32479
## log_inc_spend_pc            0.0323810  0.0289879  1.1171   0.26492
## log_ch_spend_pc             -0.1239383  0.0278366 -4.4523 1.222e-05 ***
## ch_qual                     -0.0150155  0.0029251 -5.1334 5.287e-07 ***
## st_uemp                     -0.0034145  0.0020573 -1.6597   0.09808 .
## inc_tenure:log_inc_spend_pc  0.0018438  0.0020283  0.9090   0.36410
## inc_tenure:log_ch_spend_pc  -0.0032265  0.0020713 -1.5577   0.12041
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(senate.data$inc_tenure) #what are some reasonable values?
```

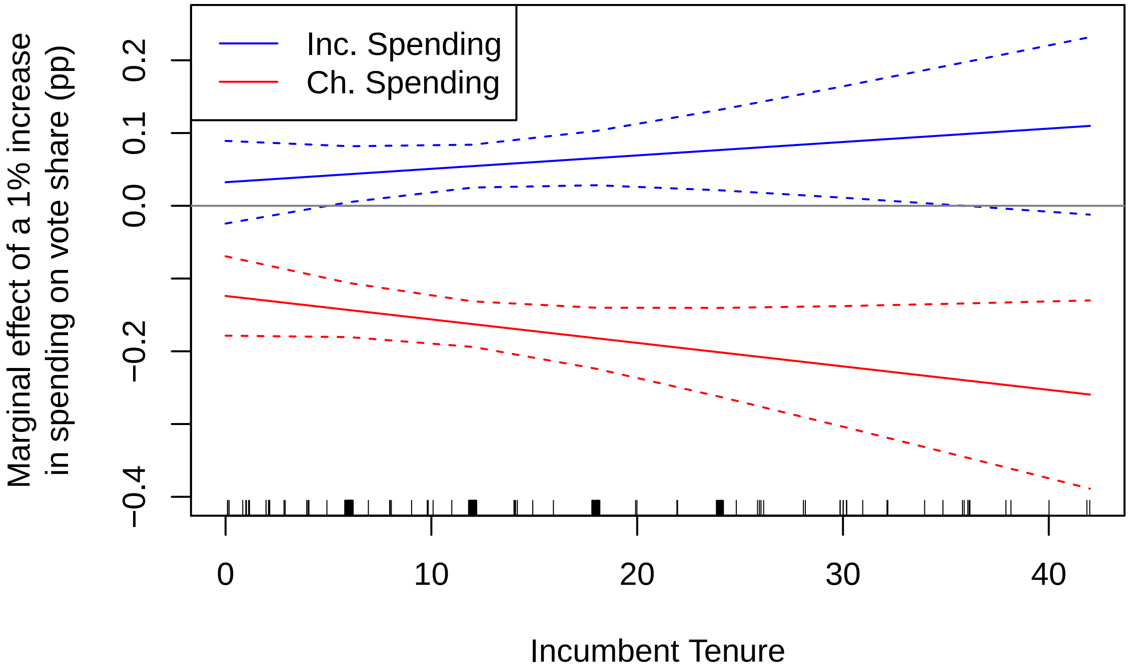
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    6.00   12.00   12.97   18.00   42.00       1
```

```r
# ### MARGINAL EFFECTS BY HAND  ###
# marginal.inc <- model3$coefficients['log_inc_spend_pc']+
# model3$coefficients['inc_tenure:log_inc_spend_pc']*seq(0, 42, by=6)
# V <- vcovHC(model3)
# ## Recall that var(X+bY) = var(X) + b^2 * var(Y) + 2b*cov(X,Y)
# var.marginal.inc <- diag(V)['log_ch_spend_pc'] +  #var beta_s
#   seq(0, 42, by=6)^2 * diag(V)['inc_tenure:log_ch_spend_pc'] + #t^2 var(beta_int)
#   2*V['log_ch_spend_pc','inc_tenure:log_ch_spend_pc']*seq(0, 42, by=6)
# se.marginal.inc <- sqrt(var.marginal)
# signif(cbind(marginal.inc, se.marginal.inc),2)
#
# marginal.ch <- model3$coefficients['log_ch_spend_pc']+
# model3$coefficients['inc_tenure:log_ch_spend_pc']*seq(0, 42, by=6)
# ## Recall that var(X+bY) = var(X) + b^2 * var(Y) + 2b*cov(X,Y)
# var.marginal.ch <- diag(V)['log_ch_spend_pc'] +  #var beta_s
#   seq(0, 42, by=6)^2 * diag(V)['inc_tenure:log_ch_spend_pc'] + #t^2 var(beta_int)
#   2*V['log_ch_spend_pc','inc_tenure:log_ch_spend_pc']*seq(0, 42, by=6)
# se.marginal.ch <- sqrt(var.marginal.ch)
# signif(cbind(marginal.ch, se.marginal.ch),2)

ME <- margins(model3,
              variables=c("log_inc_spend_pc", "log_ch_spend_pc"),
              at=list(inc_tenure=seq(0, 42, by=6)),
              vcov=vcovHC(model3))
summary(ME)
```

```
##            factor inc_tenure     AME     SE        z      p   lower   upper
##   log_ch_spend_pc     0.0000 -0.1239 0.0278  -4.4523 0.0000 -0.1785 -0.0694
##   log_ch_spend_pc     6.0000 -0.1433 0.0190  -7.5518 0.0000 -0.1805 -0.1061
##   log_ch_spend_pc    12.0000 -0.1627 0.0159 -10.2029 0.0000 -0.1939 -0.1314
##   log_ch_spend_pc    18.0000 -0.1820 0.0214  -8.5130 0.0000 -0.2239 -0.1401
##   log_ch_spend_pc    24.0000 -0.2014 0.0311  -6.4690 0.0000 -0.2624 -0.1404
##   log_ch_spend_pc    30.0000 -0.2207 0.0423  -5.2175 0.0000 -0.3037 -0.1378
##   log_ch_spend_pc    36.0000 -0.2401 0.0540  -4.4435 0.0000 -0.3460 -0.1342
##   log_ch_spend_pc    42.0000 -0.2595 0.0660  -3.9301 0.0001 -0.3888 -0.1301
##  log_inc_spend_pc     0.0000  0.0324 0.0290   1.1171 0.2640 -0.0244  0.0892
##  log_inc_spend_pc     6.0000  0.0434 0.0196   2.2144 0.0268  0.0050  0.0819
##  log_inc_spend_pc    12.0000  0.0545 0.0150   3.6282 0.0003  0.0251  0.0840
##  log_inc_spend_pc    18.0000  0.0656 0.0190   3.4430 0.0006  0.0282  0.1029
##  log_inc_spend_pc    24.0000  0.0766 0.0282   2.7163 0.0066  0.0213  0.1319
##  log_inc_spend_pc    30.0000  0.0877 0.0391   2.2454 0.0247  0.0111  0.1642
##  log_inc_spend_pc    36.0000  0.0988 0.0505   1.9554 0.0505 -0.0002  0.1977
##  log_inc_spend_pc    42.0000  0.1098 0.0622   1.7647 0.0776 -0.0121  0.2318
```

```r
plot.ME <- summary(ME)
par(mar=c(5, 6, 4, 2)+.1)
plot(AME~inc_tenure,data=plot.ME, subset=factor=="log_inc_spend_pc",
     type="l", col="blue",
     ylab="Marginal effect of a 1% increase\nin spending on vote share (pp)",
     ylim=c(-.4, .25),
     xlab="Incumbent Tenure")
lines(lower~inc_tenure,data=plot.ME, subset=factor=="log_inc_spend_pc",
      type="l", lty="dashed", col="blue")
lines(upper~inc_tenure,data=plot.ME, subset=factor=="log_inc_spend_pc",
      type="l", lty="dashed", col="blue")
lines(AME~inc_tenure,data=plot.ME, subset=factor=="log_ch_spend_pc",
      type="l", col="red")
lines(lower~inc_tenure,data=plot.ME, subset=factor=="log_ch_spend_pc",
      type="l", lty="dashed", col="red")
lines(upper~inc_tenure,data=plot.ME, subset=factor=="log_ch_spend_pc",
      type="l", lty="dashed", col="red")
```

```
rug(jitter(senate.data$inc_tenure)) #jitter spreads out clumped data
legend("topleft", legend=c("Inc. Spending", "Ch. Spending"),
       col=c("blue", "red"), lty=1)
abline(h=0,col="grey50")
```



```
# The effective of spending is increasing with tenure, but not by
# much on either side. We might be to argue that incumbent spending
# has minimal effect, while challenge spending is effect.
# But that doesn't feel right. Probably what we should ask ourselves is whether
# the interaction model is reasonable and correctly specified.
waldtest(model3, model2a, vcov=vcovHC)
```

```
## Wald test
##
## Model 1: inc_2p_share ~ inc_pos + I(inc_pos^2) + inc_rep + inc_tenure *
##     log_inc_spend_pc + inc_tenure * log_ch_spend_pc + ch_qual +
##     st_uemp
## Model 2: inc_2p_share ~ inc_pos + I(inc_pos^2) + inc_rep + inc_tenure +
##     ch_qual + st_uemp + log_ch_spend_pc + log_inc_spend_pc
##   Res.Df Df      F Pr(>F)
## 1    284
## 2    286 -2 1.2164 0.2978
```

```
which.min(c(BIC(model3), BIC(model2a)))
```

```
## [1] 2
```

```
# these basic diagnostics tell us that maybe the interaction isn't all the helpful
# for model fit.

#let's say we just wanted to predict a win/loss we could set it up as an LPM
lpm <- lm(I(inc_2p_share>.5)~log_inc_spend_pc+log_ch_spend_pc, data=senate.data)
coeftest(lpm, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error t value  Pr(>|t|)
## (Intercept)       0.901853   0.036575 24.6578 < 2.2e-16 ***
## log_inc_spend_pc  0.206031   0.061763  3.3358 0.0009605 ***
## log_ch_spend_pc  -0.477033   0.080118 -5.9542 7.491e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

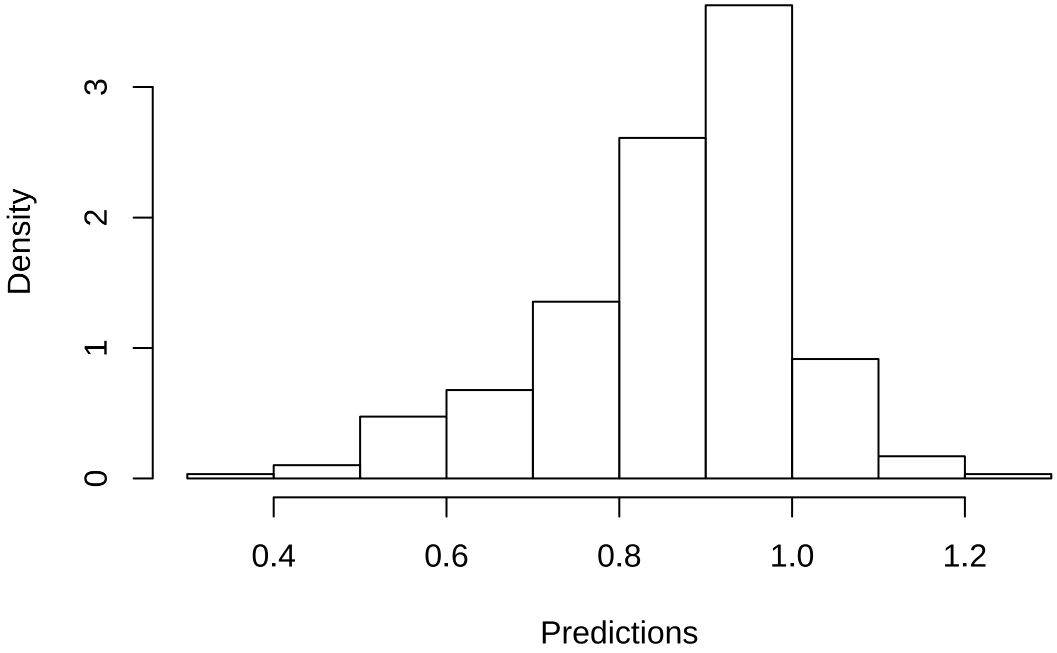
```
#what might some good values be?
summary(senate.data[,c("log_inc_spend_pc", "log_ch_spend_pc")])
```

```
##  log_inc_spend_pc log_ch_spend_pc
##  Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.3812   1st Qu.:0.07201
##  Median :0.5992   Median :0.27186
##  Mean   :0.6845   Mean   :0.37834
##  3rd Qu.:0.8955   3rd Qu.:0.58021
##  Max.   :2.0350   Max.   :1.79312
##  NA's   :71       NA's   :76
```

```
X.star <- rbind(c(1, .6,.3), #"median" race
                c(1, 2, 1.8), #big spending race
                c(1, 0, 3)) #insane
X.star %*% lpm$coefficients
```

```
##             [,1]
## [1,]  0.8823616
## [2,]  0.4552555
## [3,] -0.5292466
```

```
hist(lpm$fitted.values, freq=FALSE, main="", xlab="Predictions")
```



Finally how about a conflict example.

```
library(readstata13)
library(lmtest)
library(sandwich)
library(car)
library(margins)
library(nonnest2)
rm(list=ls())

cw.data <- read.dta13("Rcode/datasets/civwar.dta")
```

```
## Warning in read.dta13("Rcode/datasets/civwar.dta"):
##    Factor codes of type double or float detected in variables
##
##    region
##
##    No labels have been assigned.
##    Set option 'nonint.factors = TRUE' to assign labels anyway.
```

```
# print(cbind(colnames(cw.data), attributes(cw.data)$var.labels))

# Let's start basic
model0 <- lm(onset~lgdpenl1, data=cw.data)
dim(model0$model)
```

```
## [1] 6373    2
```

```
signif(coeftest(model0, vcov=vcovHC),2)
```

```
## 
## t test of coefficients:
## 
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   0.0930     0.0130     7.0  3.8e-12 ***
## lgdpenl1     -0.0099     0.0016    -6.1  9.0e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(cw.data$lgdpenl1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   3.871   6.835   7.598   7.651   8.408  11.108     237 
```

```
predict(model0, newdata=data.frame(lgdpenl1=c(3.871,11.108)))
```

```
##           1           2 
##  0.05449778 -0.01679099 
```

```
#unlikely bounds
summary(model0$fitted.values)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## -0.016796  0.009803  0.017781  0.017260  0.025299  0.054496 
```

```
## Let's kick it up a notch
#regions: N Africa and ME is reference
model1 <- lm(onset~lpopl1+polity2l+lgdpenl1+
               western+eeurop+lamerica+ssafrica+asia
             +colbrit+colfra+lmtnest+ncontig+oil
             +ethfrac+relfrac,
             data=cw.data)
signif(coeftest(model1, vcov=vcovHC), 2)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.07300    0.02800    2.60   0.0094 **
## lpopl1       0.00340    0.00140    2.40   0.0170 *
## polity2l     0.00069    0.00033    2.10   0.0390 *
## lgdpenl1    -0.01100    0.00290   -3.70   0.0002 ***
## western     -0.02600    0.00820   -3.20   0.0013 **
## eeurop      -0.01100    0.01100   -1.00   0.3000
## lamerica    -0.01400    0.00710   -1.90   0.0520 .
## ssafrica    -0.00800    0.00890   -0.91   0.3700
## asia        -0.00750    0.00900   -0.84   0.4000
## colbrit     -0.00760    0.00550   -1.40   0.1700
## colfra      -0.00910    0.00640   -1.40   0.1600
## lmtnest      0.00150    0.00130    1.20   0.2300
## ncontig      0.01600    0.00790    2.00   0.0470 *
## oil          0.01400    0.00730    1.90   0.0570 .
## ethfrac      0.00530    0.00940    0.57   0.5700
## relfrac      0.00600    0.00940    0.64   0.5200
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#' On average,
#' A 1% increase in (last years) GDP per captia decreases the probability
#' of civil war onset by about 0.011 percentage point
#'   (-0.01090571/100)                   * 100
#'  Interpret logged X (1%)  | From prob scale to pp
#'
#' A 100% increase in (last years) Population increases
#' the probability of a civil war onset by about 0.0034 percentage points.
#'   (0.00341965/100)                 * 100
#'  Interpret logged X (1%)  | From prob scale to pp
```

```
#'
#' Comparing a true autocracy (polity -10) to a full democracy (+10)
#' We see an average increase of 1.37 percentage points
#' in the risk of civil conflict, holding all else equal
#' 0.00068536               * 20             *100
#'  One unit increase | increase of 20 units | From prob scale to pp
#'
#' holding the other variables fixed

summary(model1$fitted.values)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.029148  0.006895  0.016271  0.017228  0.026047  0.082384
```

```
head(cw.data[names(which(model1$fitted.values<0) ), c("country", "year")])
```

```
##    country year
## 56  CANADA 1945
## 57  CANADA 1946
## 58  CANADA 1947
## 59  CANADA 1948
## 60  CANADA 1949
## 61  CANADA 1950
```

```
#' Canada remains super unlikely to experience civil war onset.

#' Let's consider the issues of comparative theory testing
#' by considering non-nested models

# 1. Political Institutions: Democracy, GDP per capita, colonial status
# 2. Geographic Theory: Contiguity, mountains
# 3. Heterogeniety theories: Ethnic frac

#' In all three we'll include the region dummies and population
#' We will test these theories using first an "encompassing model" (which we have)
#' then model fit stats and finally a Vuong test

#' So far we can see that the ethnic heterogeneity is the least appealing
```

```r
model.political <- lm(onset~lpopl1
                      +polity2l+lgdpenl1
                      +western+eeurop+lamerica+ssafrica+asia
                      +colbrit+colfra
                      # +lmtnest+ncontig + oil
                      # +ethfrac +relfrac
                      ,
                      data=model1$model)

model.geography <- lm(onset~lpopl1
                      # +polity2l+lgdpenl1+
                      +western+eeurop+lamerica+ssafrica+asia
                      # +colbrit+colfra
                      +lmtnest+ncontig+oil
                      # +ethfrac +relfrac
                      ,
                      data=model1$model)

model.het <- lm(onset~lpopl1
                # +polity2l+lgdpenl1+
                +western+eeurop+lamerica+ssafrica+asia
                # +colbrit+colfra
                # +lmtnest+ncontig +oil
                +ethfrac +relfrac
                ,
                data=model1$model)

#' Remember we can use AIC, BIC, and adj R^2 to compare nested or non-nested models
#' So long as they have the same y and sample
#'
#' Can't use nested model tests here
which.min(c(AIC(model.political), AIC(model.geography), AIC(model.het)))
```

```
## [1] 1
```

```r
which.min(c(BIC(model.political), BIC(model.geography), BIC(model.het)))
```

```
## [1] 2
```

```r
which.max(c(summary(model.political)$adj.r,
            summary(model.geography)$adj.r,
            summary(model.het)$adj.r))
```

```
## [1] 1
```

```r
#' What get's us the closest? That's for Vuong tests to decide
vuongtest(model.political, model.geography) # pass the first test, but can't tell
```

```
##
## Model 1
##  Class: lm
##  Call: lm(formula = onset ~ lpopl1 + polity2l + lgdpenl1 + western + ...
##
## Model 2
##  Class: lm
##  Call: lm(formula = onset ~ lpopl1 + western + eeurop + lamerica + ssafrica + ...
##
## Variance test
##   H0: Model 1 and Model 2 are indistinguishable
##   H1: Model 1 and Model 2 are distinguishable
##     w2 = 0.008,   p = 0.00171
##
## Non-nested likelihood ratio test
##   H0: Model fits are equal for the focal population
##   H1A: Model 1 fits better than Model 2
##     z = 0.380,   p = 0.352
##   H1B: Model 2 fits better than Model 1
##     z = 0.380,   p = 0.648
```

```r
vuongtest(model.political, model.het) # pass the first test, political wins
```

```
##
## Model 1
##  Class: lm
##  Call: lm(formula = onset ~ lpopl1 + polity2l + lgdpenl1 + western + ...
##
## Model 2
##  Class: lm
##  Call: lm(formula = onset ~ lpopl1 + western + eeurop + lamerica + ssafrica + ...
##
## Variance test
##   H0: Model 1 and Model 2 are indistinguishable
##   H1: Model 1 and Model 2 are distinguishable
##     w2 = 0.005,   p = 0.000416
##
## Non-nested likelihood ratio test
##   H0: Model fits are equal for the focal population
##   H1A: Model 1 fits better than Model 2
##     z = 1.786,   p = 0.0371
##   H1B: Model 2 fits better than Model 1
##     z = 1.786,   p = 0.9629
```

```
vuongtest(model.geography, model.het) # pass the first test, geography wins
```

```
##
## Model 1
##  Class: lm
##  Call: lm(formula = onset ~ lpopl1 + western + eeurop + lamerica + ssafrica + ...
##
## Model 2
##  Class: lm
##  Call: lm(formula = onset ~ lpopl1 + western + eeurop + lamerica + ssafrica + ...
##
## Variance test
##   H0: Model 1 and Model 2 are indistinguishable
##   H1: Model 1 and Model 2 are distinguishable
##     w2 = 0.003,   p = 0.0415
##
## Non-nested likelihood ratio test
##   H0: Model fits are equal for the focal population
##   H1A: Model 1 fits better than Model 2
##     z = 1.660,   p = 0.0485
##   H1B: Model 2 fits better than Model 1
##     z = 1.660,   p = 0.9515
```

```
#' So we can probably feel good rejecting the heterogeneity theory.
#' We can't really tell if political institutions
#' or the geographic conditions matter more.
#' If we had to select a single theory it might be
#' the political institutions based on the AIC and adjusted R^2.
```

6 Endogeniety and instrumental variables

So far we've focused on relaxing the assumptions of normality and homoskedasticity, while also skirting around linearity. Now we will consider the problem of endogeneity, or situations where $\mathrm{E}[\varepsilon_i|x_i] \neq 0$. There are three main ways in which this assumption is violated: omitted variables, measurement error in the independent variables, and when the treatment and outcome are simultaneously determined. First, we will demonstrate that all three of these conditions can lead to endogeneity. Then we will consider how to tackle the problem. Throughout we will show that $\mathrm{E}[\varepsilon_i x_i] \neq 0$ which in turn implies that $\mathrm{E}_x[x_i \mathrm{E}[\varepsilon_i|x_i]] \neq 0$. In turn this implies that for some x_i $\mathrm{E}[\varepsilon_i|x_i] \neq 0$. Endogeneity is the bane of causal inference with observational data, and so this is the launching point for most of what you'll be worried about for years to come.

6.1 Sources of endogeneity

6.1.1 Omited variables

We've briefly considered OVB before, but let's recap. Suppose we wanted to identify the effect that years of schooling has on wages, but further suppose that that wages are determined by both years of schooling and an (unobserved) variable called intelligence. Additionally, assume that years of schooling depends on intelligence. As such we have

$$w_i = \beta_0 + \beta_1 s_i + \underbrace{\beta_2 q_i + \varepsilon_i}_{\text{error}},$$

where w_i denotes wages, s_i denotes years of schooling, and q_i denotes intelligence. Note that the population regression will satisfy the main assumptions that $\mathrm{E}[\varepsilon_i|s_i, q_i] = 0$. We have to force the unconditional expected value of our new error term to be 0, so we add and subtract $\beta_2 \mathrm{E}[q_i]$ from the above to get

$$\begin{aligned} w_i &= \underbrace{\beta_0 + \beta_2 \mathrm{E}[q_i]}_{\text{constant}} + \beta_1 s_i + \underbrace{\beta_2(q_i - \mathrm{E}[q_i]) + \varepsilon_i}_{\text{error}} \\ &= \alpha_0 + \beta_1 s_i + u_i \end{aligned}$$

We need to know if $u_i = \beta_2(q_i - \mathrm{E}[q_i]) + \varepsilon_i$ is related to s_i, or to be precise if $\mathrm{E}[u_i s_i]$ is 0 or not. If they are unrelated we're good to go, otherwise Assumption B3 is violated.

$$\begin{aligned}
\mathrm{E}[u_i s_i] &= \mathrm{E}[(\beta_2(q_i - \mathrm{E}[q_i]) + \varepsilon_i) s_i] \\
&= \mathrm{E}[(\beta_2(q_i - \mathrm{E}[q_i]) + \varepsilon_i) s_i] \\
&= \mathrm{E}[\beta_2(q_i - \mathrm{E}[q_i]) s_i + \varepsilon_i s_i] \\
&= \mathrm{E}[\beta_2(q_i - \mathrm{E}[q_i]) s_i] + \mathrm{E}[\varepsilon_i s_i] \\
&= \beta_2 \, \mathrm{E}[(q_i - \mathrm{E}[q_i]) s_i] + 0 \\
&= \beta_2 \, \mathrm{E}[q_i s_i - \mathrm{E}[q_i] s_i] \\
&= \beta_2 \left(\mathrm{E}[q_i s_i] - \mathrm{E}[\mathrm{E}[q_i] s_i] \right) \\
&= \beta_2 \, \mathrm{Cov}(q_i, s_i)
\end{aligned}$$

In order for OLS to be unbiased and consistent we need this to be 0. This will happen if either $\beta_2 = 0$ (there is no direct relationship between intelligence and wages) or if $\mathrm{Cov}(q_i, s_i) = 0$ schooling does not relate to intelligence. Omitted variable bias is thus a form of endogenity. OLS is biased and inconsistent due to a failure of the exogeneity assumption. Recall that we derived the bias earlier for the two variable case earlier and found

$$\mathrm{E}[\hat{\beta}_1] = \beta_1 + \beta_2 \frac{s_{qs}}{s_q^2} \xrightarrow{p} \beta_1 + \beta_2 \frac{\mathrm{Cov}(q, s)}{\mathrm{Var}(s)}.$$

Note we can try to proxy for intelligence by using an imperfect measure $\tilde{q}_i$. However, so long as we there is still an aspect q_i^* omitted from the new measure, we can make the same argument for bias. This is why proxies are poor solutions to the problem (but sometimes a poor solution is all you got).

- Additionally, this discussion points to the first two possible solutions to endogeneity
  1. Experiments (lab/survey or natural/quasi). Randomization limits any concerns about OVB
  2. Use the right control variables (most common strategy; but obviously limited)

6.1.2 Measurement error

Another source of endogeneity is measurement error. We start with a model

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

where $\mathrm{E}[\varepsilon_i|x_i] = 0$. So far so good. But suppose we don't observe x_i, but instead we can only observe a noisy proxy

$$\tilde{x}_i = x_i + \nu_i,$$

where $\mathrm{E}[\nu_i] = 0$ and $\mathrm{E}[x_i \nu_i] = \mathrm{E}[\nu_i \varepsilon_i] = 0$. Plugging in the noisy version we see

$$\begin{aligned} y_i &= \beta_0 + \beta_1(\tilde{x}_i - \nu_i) + \varepsilon_i \\ &= \beta_0 + \beta_1 \tilde{x}_i + \varepsilon_i - \beta_1 \nu_i \\ &= \beta_0 + \beta_1 \tilde{x}_i + u_i \end{aligned}$$

Now note that in order for OLS to be unbiased we once again need $\mathrm{E}[\tilde{x}_i u_i]$ to be zero.

$$\begin{aligned} \mathrm{E}[\tilde{x}_i u_i] &= \mathrm{E}[(x_i + \nu_i)(\varepsilon_i - \beta_1 \nu_i)] \\ &= \mathrm{E}[x_i \varepsilon_i - x_i \beta_1 \nu_i + \nu_i \varepsilon_i - \beta_1 \nu_i^2] \\ &= \mathrm{E}[x_i \varepsilon_i] - \beta_1 \, \mathrm{E}[x_i \nu_i] + \mathrm{E}[\nu_i \varepsilon_i] - \beta_1 \, \mathrm{E}[\nu_i^2] \\ &= 0 - 0 + 0 - \beta_1 \, \mathrm{E}[\nu_i^2] \\ &= -\beta_1 \, \mathrm{E}[\nu_i^2] \\ &= -\beta_1 \, \mathrm{Var}[\nu_i] \end{aligned}$$

This will only equal 0 if $\beta_1 = 0$ (there's no direct effect of x_i on y_i) or if ν_i is constant for all i.

Can we say something about what that bias looks like? Let's try it. We're looking at a simple regression here (1 X and 1 y). so we can write

$$\begin{aligned}
\hat{\beta}_1 &= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x}_i - \bar{\tilde{x}}_i)(y_i - \bar{y})}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} \\
&= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i + \nu_i - \bar{\nu}_i)(\beta_0 + \beta_1 x_i + \varepsilon_i - \beta_0 - \beta_1\bar{x}_i - \bar{\varepsilon})}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} \\
&= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)(\beta_1 x_i - \beta_1\bar{x}_i + \varepsilon_i - \bar{\varepsilon})}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} + \frac{\frac{1}{N-1}\sum_{i=1}^{N}(\nu_i - \bar{\nu}_i)(\beta_1 x_i - \beta_1\bar{x}_i + \varepsilon_i - \bar{\varepsilon})}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} \\
&= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)(\beta_1 x_i - \beta_1\bar{x}_i)}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} + \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)(\varepsilon_i - \bar{\varepsilon})}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} \\
&\quad + \frac{\frac{1}{N-1}\sum_{i=1}^{N}(\nu_i - \bar{\nu}_i)(\beta_1 x_i + \beta_1\bar{x}_i + \varepsilon_i - \bar{\varepsilon})}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} \\
\mathrm{E}[\hat{\beta}_1|X] &= \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)(\beta_1 x_i - \beta_1\bar{x}_i)}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} + \frac{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x}_i)\,\mathrm{E}[(\varepsilon_i - \bar{\varepsilon}_i)]}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} \\
&\quad + \frac{\frac{1}{N-1}\sum_{i=1}^{N}\mathrm{E}[(\nu_i - \bar{\nu}_i)](\beta_1 x_i + \beta_1\bar{x}_i + \varepsilon_i - \bar{\varepsilon}_i)}{\frac{1}{N-1}\sum_{i=1}^{N}(\tilde{x} - \bar{\tilde{x}}_i)^2} \\
&= \frac{s_x^2}{s_{\tilde{x}}^2}\beta_1
\end{aligned}$$

This makes the total bias of $\hat{\beta}_1$

$$\mathrm{bias}(\hat{\beta}_1) = \frac{s_x^2}{s_{\tilde{x}}^2}\beta_1 - \beta_1 = \beta_1\frac{s_x^2 - s_{\tilde{x}}^2}{s_{\tilde{x}}^2}.$$

Note that this does not improve with sample size. We can show that $s_{\tilde{x}}^2 \xrightarrow{p} \sigma_x^2 + \sigma_\nu^2$ and as such

$$\hat{\beta}_1 \xrightarrow{p} \frac{\sigma_x^2}{\sigma_x^2 + \sigma_\nu^2}\beta_1.$$

Note that because $\sigma_{(\cdot)}^2 > 0$ that this bias will always be toward 0 (attenuation bias).

Before moving on, let's ask the obvious next question: what about measurement error in y_i? Suppose we have

$$y = X\beta + \varepsilon,$$

but the outcome we observed is measured with error, such that

$$\tilde{y}_i = y_i + \nu_i.$$

Let $\mathrm{E}[\nu|X] = 0$ and consider the OLS estimate

$$\begin{aligned}\hat{\beta} &= (X'X)^{-1}X'\tilde{y}\\ &= (X'X)^{-1}X'(y-\nu)\\ &= (X'X)^{-1}X'y - (X'X)^{-1}X'\nu\\ &= (X'X)^{-1}X'X\beta + (X'X)^{-1}X'\varepsilon - (X'X)^{-1}X'\nu\\ &= \beta + (X'X)^{-1}X'(\varepsilon - \nu)\end{aligned}$$

Note that $\varepsilon - \nu$ is a compound error term, but it still has expectation 0. So long as $\mathrm{E}[\varepsilon_i - \nu_i|x_i] = 0$, we're good to go. Random error in the dependent variable just adds noise to the model and increases the overall variance, but could be worse. OLS is still unbiased and consistent.

6.1.3 Simultaneity

Another way we might find endogeneity is when X and y cause each other. Consider the relationship between economic growth and civil war. We might imagine that growth does not increase the likelihood of civil war but maybe civil war decreases growth. In this case if we regressed civil war on growth we might find a negative and significant coefficient estimate, despite there not being a real causal relationship of growth on civil war (instead we accidentally pick up the negative effect of conflict on growth). This type of endogeneity is a more concerning perhaps than OVB, because it can occur even if you have all the right variables.

To tease this out, we might write a model that includes of both growth and conflict

$$\begin{aligned}w_i &= \beta_0 + \beta_1 g_i + \beta_2 x_i + \varepsilon_i\\ g_i &= \gamma_0 + \gamma_1 w_i + \gamma_2 x_i + \nu_i.\end{aligned}$$

Let's put some structure in the sense that $\mathrm{E}[\varepsilon_i|x_i] = \mathrm{E}[\nu_i|x_i] = 0$. We need $\mathrm{E}[\varepsilon_i g_i] = \mathrm{E}[\nu_i w_i] = 0$ in order for OLS to be consistent.

We can rewrite the above to be in "reduced form" which is to say that they are only in terms of the exogenous variables.

$$\begin{aligned}w_i &= \frac{\beta_0 + \beta_1\gamma_0}{1-\beta_1\gamma_1} + \frac{\beta_1\gamma_2 + \beta_2}{1-\beta_1\gamma_1}x_i + \frac{\varepsilon_i + \beta_1\nu_i}{1-\beta_1\gamma_1}\\ g_i &= \frac{\gamma_0 + \gamma_1\beta_0}{1-\beta_1\gamma_1} + \frac{\gamma_1\beta_2 + \gamma_2}{1-\beta_1\gamma_1}x_i + \frac{\nu_i + \gamma_1\varepsilon_i}{1-\beta_1\gamma_1}.\end{aligned}$$

How we can check the relevant expectations

$$\begin{aligned}
\mathrm{E}[\nu_i w_i] &= \mathrm{E}\left[\nu_i\left(\frac{\beta_0+\beta_1\gamma_0}{1-\beta_1\gamma_1}+\frac{\beta_1\gamma_2+\beta_2}{1-\beta_1\gamma_1}x_i+\frac{\varepsilon_i+\beta_1\nu_i}{1-\beta_1\gamma_1}\right)\right]\\
&= \mathrm{E}\left[\nu_i\right]\frac{\beta_0+\beta_1\gamma_0}{1-\beta_1\gamma_1}+\mathrm{E}\left[\nu_i x_i\right]\frac{\beta_1\gamma_2+\beta_2}{1-\beta_1\gamma_1}+\mathrm{E}\left[\frac{\varepsilon_i\nu_i+\beta_1\nu_i^2}{1-\beta_1\gamma_1}\right]\\
&= 0+0+\frac{\mathrm{E}\left[\varepsilon_i\nu_i\right]+\beta_1\,\mathrm{E}\left[\nu_i^2\right]}{1-\beta_1\gamma_1}\\
&= \frac{\mathrm{E}\left[\varepsilon_i\nu_i\right]}{1-\beta_1\gamma_1}+\beta_1\frac{\mathrm{Var}\left(\nu_i\right)}{1-\beta_1\gamma_1}\\
\mathrm{E}[\varepsilon_i g_i] &= \frac{\mathrm{E}\left[\varepsilon_i\nu_i\right]}{1-\beta_1\gamma_1}+\gamma_1\frac{\mathrm{Var}\left(\varepsilon_i\right)}{1-\beta_1\gamma_1}
\end{aligned}$$

Getting both of these to be 0 is a bit of a stretch. We could have both $\gamma_1 = 0$, $\beta_1 = 0$, **and** $\mathrm{E}[\varepsilon_i\nu_i] = 0$. If all three of those hold then there is no relationship among our equations and we're back to ordinary OLS. If any of them hold, then one of these will be inconsistent. Or we could let γ_1 and β_1 be free but have $\mathrm{E}[\varepsilon_i\nu_i]$ be whatever value it needed to be to satisfy these equations. This would be a **very weird** assumption.

Well maybe we could just fit the exogenous reduced form model

$$\begin{aligned}
w_i &= \frac{\beta_0+\beta_1\gamma_0}{1-\beta_1\gamma_1}+\frac{\beta_1\gamma_2+\beta_2}{1-\beta_1\gamma_1}x_i+\frac{\varepsilon_i+\beta_1\nu_i}{1-\beta_1\gamma_1}\\
g_i &= \frac{\gamma+\gamma_1\beta_0}{1-\beta_1\gamma_1}+\frac{\gamma_1\beta_2+\gamma_2}{1-\beta_1\gamma_1}x_i+\frac{\nu_i+\gamma_1\varepsilon_i}{1-\beta_1\gamma_1}.
\end{aligned}$$

This only contains exogenous x_i and it has β_1 and γ_1 in it. Can we back out β_1 from these estimates? No we have too many pieces. Notice that we have 6 parameters to ID, $\theta = (\beta_0, \beta_1, \gamma, \beta_2, \gamma_0, \gamma_1, \gamma_2)$, but only 4 pieces of information.

However, what if $\beta_2 = 0$? Well then we have

$$\begin{aligned}
w_i &= \frac{\beta_0+\beta_1\gamma_0}{1-\beta_1\gamma_1}+\frac{\beta_1\gamma_2}{1-\beta_1\gamma_1}x_i+\frac{\varepsilon_i+\beta_1\nu_i}{1-\beta_1\gamma_1}\\
g_i &= \frac{\gamma+\gamma_1\beta_0}{1-\beta_1\gamma_1}+\frac{\gamma_2}{1-\beta_1\gamma_1}x_i+\frac{\nu_i+\gamma_1\varepsilon_i}{1-\beta_1\gamma_1}.
\end{aligned}$$

Now we still have too many unknowns to back out everything, but…notice that

$$\frac{\beta_1\gamma_2}{1-\beta_1\gamma_1}\Big/\frac{\gamma_2}{1-\beta_1\gamma_1}=\beta_1.$$

We can identify β_1 from the ratio of the coefficients on x_i from the two reduced form regressions. HOW. COOL. IS. THAT? However, we would need another restriction in order to find γ. Note that this restriction works because when $\beta_2 = 0$ and $\gamma_2 \neq 0$ we have a variable x_i that effect growth (and thus has an indirect effect on civil war), but it has **no direct effect** on civil war.

Here x_i is an example of an **instrumental variable**. This will be a third strategy for combating endogeneity. If we wanted to identify γ_1 we would need another instrument: something that effects civil war onset, but has no direct effect on growth. It only influences conflict through its effect on growth.

There are other forms of endogeneity that you will certainly come across:

1. Non-random selection into the sample (People are not randomly assigned to run for office or enter the work force)
2. Feedback or other dynamic effects
3. Attrition (non random exiting from the sample)
4. Strategic interaction among the participants

Note that in most cases, endogeneity can be thought of as a problem of inferring causation from correlation. With the exception of measurement error, most of these setups are situations where regressing y_i on x_i leads us to incorrectly conclude that observed correlation indicates that x_i causes y_i (education cause wages, growth reduces the conflict risk, etc).

Two main methods exist for working with endogeneity concerns: structural solutions (model the relationships that induce endogeneity using additional equations and structure) and reduced form estimation based on instrumental variables (or another tool to remove the endogeneity concerns like regression discontinuity or difference-in-differences, you'll return to these if you take causal inference). For the most part reduced form approaches dominate modern empirical work and are inline with the scope of the course. We will focus on the central tool for that: the method of instrumental variables.

6.2 2SLS

The main estimation framework for dealing with endogeneity is two-stage least squares (2SLS). This approach is based on the idea that we will need 1 or more instruments Z for each endogeneous variable in X. We will build a new set of assumptions here

Assumption C1 *The DGP is linear:* $y_i = \beta' x_i + \varepsilon_i$

Assumption C2 *The pairs (x_i, ε_i) are independently and identicially distributed*

Assumption C3 *The instruments are exogeneous:* $\mathrm{E}[\varepsilon_i | z_i] = 0$

Assumption C4 *The instruments are relevant: rank*$(\mathrm{E}[x_i z_i'])$ $\geq$ *dim*(x_i)

Assumption C5 *The instruments are not redundant: rank*$(\mathrm{E}[z_i z_i'])$ $=$ *dim*(z_i)

Assumption C6 *Other technical assumptions on various moments that allow for the law of large numbers and the CLT*

Note that exogenous variables in X (including the constant term) are able to instrument for themselves. As such variables in X may also be in Z. Note that Assumption C4 also tells us that $\dim(z_i) \geq \dim(x_i)$. To find $\hat{\beta}_{2SLS}$ you can either do it all at once or you can do it as two-steps (thus the name):

1. Regress all X on Z to get $\hat{\Gamma} = (Z'Z)^{-1}Z'X$. Set $\hat{X} = Z\hat{\Gamma}$
2. Regress y on $\hat{X}$

Here $\hat{X}$ is the part of X that "comes from" Z. Because Z is exogenous, but related to X, we only include the exogenous part of X determined by Z. The 2SLS estimator then becomes

$$\begin{aligned}\hat{\beta}_{2SLS} &= (\hat{X}'\hat{X})^{-1}\hat{X}'y \\ &= (X'Z(Z'Z)^{-1}Z'X)^{-1}(X'Z(Z'Z)^{-1}Z'y) \\ &= (X'H_Z X)^{-1}(X'H_Z y)\end{aligned}$$

The second line is the "one-step" version, which is what you would actually use. The idempotent matrix $H_Z = Z(Z'Z)^{-1}Z'$. In the special case where $\dim(z_i) = \dim(x_i)$ (i.e., we have one endogenous variable in X and one instrument for it in Z), we can get an easier formula

$$\begin{aligned}\hat{\beta}_{2SLS} &= (X'Z(Z'Z)^{-1}Z'X)^{-1}(X'Z(Z'Z)^{-1}Z'y) \\ &= (Z'X)^{-1}(Z'Z)(X'Z)^{-1}(X'Z)(Z'Z)^{-1}Z'y \\ &= (Z'X)^{-1}(Z'Z)(Z'Z)^{-1}Z'y \\ &= (Z'X)^{-1}Z'y\end{aligned}$$

The 2SLS estimator is biased, but consistent. First, let's rewrite the estimator

$$\hat{\beta}_{2SLS} = \beta + \left(\frac{1}{N}X'Z\left(\frac{1}{N}Z'Z\right)^{-1}\frac{1}{N}Z'X\right)^{-1}\left(\frac{1}{N}X'Z\left(\frac{1}{N}Z'Z\right)^{-1}\frac{1}{N}Z'\varepsilon\right)$$

To see the consistency start with the follow LLN arguments for iid observations

$$\begin{aligned}
\frac{1}{N}X'Z &\xrightarrow{p} \mathrm{E}[x_i z_i'] \\
\frac{1}{N}Z'X &\xrightarrow{p} \mathrm{E}[z_i x_i'] \\
\left(\frac{1}{N}Z'Z\right)^{-1} &\xrightarrow{p} \mathrm{E}[z_i z_i']^{-1} && \text{by C5 and CMT} \\
\frac{1}{N}Z'\varepsilon &\xrightarrow{p} \mathrm{E}[z_i \varepsilon_i] = 0 && \text{by C3} \\
\det\left(\frac{1}{N}X'Z\left(\frac{1}{N}Z'Z\right)\frac{1}{N}Z'X\right) &\xrightarrow{p} \det\left(\mathrm{E}[x_i z_i']\,\mathrm{E}[z_i z_i']^{-1}\,\mathrm{E}[x_i' z_i]\right) > 0 && \text{by C5, C4, Slutsky, and CMT}
\end{aligned}$$

All together this gives us

$$\left(\frac{1}{N}X'Z\left(\frac{1}{N}Z'Z\right)^{-1}\frac{1}{N}Z'X\right)^{-1}\left(\frac{1}{N}X'Z\left(\frac{1}{N}Z'Z\right)^{-1}\frac{1}{N}Z'\varepsilon\right) \xrightarrow{p} 0$$

by Slutsky and the continuous mapping theorem, or

$$\hat{\beta}_{2SLS} \xrightarrow{p} \beta.$$

The intuition behind the bias comes from the fact than no instrument will be perfect in finite samples so you get some endogeneity bias "slipping back in." We will return to this in a moment.

Property C1 *Under Assumptions C1-C6, the 2SLS estimator exists for large enough N and* $\hat{\beta}^N_{2SLS} \xrightarrow{p} \beta$.

We can consider the asymptotic distribution of $\sqrt{N}(\hat{\beta}^N_{2SLS} - \beta)$:

$$\sqrt{N}(\hat{\beta}^N_{2SLS} - \beta) = \left(\frac{1}{N}X'Z\left(\frac{1}{N}Z'Z\right)^{-1}\frac{1}{N}Z'X\right)^{-1}\left(\frac{1}{N}X'Z\left(\frac{1}{N}Z'Z\right)^{-1}\frac{\sqrt{N}}{N}Z'\varepsilon_i\right),$$

where the CLT tells us

$$\frac{\sqrt{N}}{N}Z'\varepsilon_i \xrightarrow{d} N(0, \mathrm{Var}(z_i\varepsilon_i)) = N(0, \mathrm{E}[\varepsilon_i^2 z_i z_i']).$$

Using the same large sample arguments as before we get,

Property C2 *Under Assumptions C1-C6,*

$$\sqrt{N}(\hat{\beta}^N_{2SLS} - \beta) \xrightarrow{d} N(0, A^{-1}BA^{-1})$$

where

$$
\begin{aligned}
B &= \mathrm{E}[x_i z_i'] \, \mathrm{E}[z_i z_i']^{-1} \, \mathrm{E}[\varepsilon_i^2 z_i z_i'] E[z_i z_i']^{-1} \, \mathrm{E}[z_i x_i'] \\
A &= \mathrm{E}[x_i z_i'] \, \mathrm{E}[z_i z_i']^{-1} \, \mathrm{E}[z_i x_i']
\end{aligned}
$$

This gives us a full robust variance matrix of

$$\widehat{\text{avar}}(\hat{\beta}) = (X'H_Z X)^{-1} \left[X'Z(Z'Z)^{-1} \left(\sum_{i=1}^{N} \hat{\varepsilon}_i^2 z_i z_i' \right) (Z'Z)^{-1} Z'X \right] (X'H_Z X)^{-1}$$

If you want to make a homoskedasticity assumption (we haven't yet) this simplifies, but 2SLS is only asymptotically valid, so why not use the asymptotic covariance matrix? Note that when $\dim(z) = \dim(x)$ (the same number of true instruments as endogenous variables and letting each exogenous variable instrument itself), we can rearrange terms and let things cancel. As such the estimated variance becomes:

$$\sqrt{N}(\hat{\beta}_{2SLS}^{N} - \beta) \xrightarrow{d} N(0, \mathrm{E}[z_i x_i']^{-1} \, \mathrm{E}[\varepsilon_i^2 z_i z_i'] \, \mathrm{E}[x_i z_i']^{-1}).$$

or

$$\widehat{\text{avar}}(\hat{\beta}) = (Z'X)^{-1} \left(\sum_{i=1}^{N} \hat{\varepsilon}_i^2 z_i z_i' \right) (X'Z)^{-1}$$

Which looks a lot like the ordinary sandwich estimator.

6.2.1 Weak instruments and the 2SLS

Story time: Angrist and Krueger (1991) is a very famous paper for perhaps the wrong reasons. In the paper they want to estimate the effect that schooling has on wages. To avoid endogeniety concerns they instrument for education using the quarter of the year that you're born in. Here's the argument:

1. Children born earlier in the year will be enrolled in school "sooner." These kids will receive more education by the time they turn 16 and are allowed to dropout (relevance)
2. Quarter of birth is random so it should have no effect on wages outside of how much schooling someone received (validity)

So far this makes a degree of sense, but the effect of birth quarter on schooling must be very small (if any). This lack of a strong first-stage relationship makes it a **weak** instrument. Weak instruments can lead to real problems in terms of both bias and variance in the 2SLS estimates. Specifically, let's consider the following model

$$\begin{aligned} y_i &= \beta x_i + \varepsilon_i \\ x_i &= \gamma' z_i + \nu_i \end{aligned}$$

For ease, we'll only have a single endogenous regressor and let $\beta_0 = 0$. The 2SLS estimator is given by

$$\begin{aligned} \hat{\beta}_{2sls} &= \beta + (x'H_Z x)^{-1}(x'H_Z\varepsilon) \\ \hat{\beta}_{2sls} - \beta &= (x'H_Z x)^{-1}\left((Z\gamma+\nu)'H_Z\varepsilon\right) && \text{First stage relationship} \\ &= (x'H_Z x)^{-1}(\gamma' Z'\varepsilon) && \text{Distribute} \\ &\quad + (x'H_Z x)^{-1}(\nu' H_Z\varepsilon) && \text{Note: } Z'H_Z = Z' \end{aligned}$$

Now we have a problem. Unlike with OLS we can't just take conditional expectations to break up these terms. Specifically, X includes an error term so we would need to pass the expectation operator into the inverse (which we can't do, the inverse is a non-linear function). What we're going to do instead is take a first-order approximation. This approach is sometimes called "group-wise" asymptotics. The premise is that it works by assuming that the number of instruments in increasing with the sample size. This assumption maintains the "weakness" of the instruments as N increase.

$$\begin{aligned} \mathrm{E}[\hat{\beta}_{2sls} - \beta] &\approx (\mathrm{E}[x'H_Z x])^{-1}(\mathrm{E}[\gamma' Z'\varepsilon]) \\ &\quad + (\mathrm{E}[x'H_Z x])^{-1}(\mathrm{E}[\nu' H_Z\varepsilon]) \\ &\approx (\mathrm{E}[x'H_Z x])^{-1}(\gamma' 0) && \text{By C3} \\ &\quad + (\mathrm{E}[x'H_Z x])^{-1}(\mathrm{E}[\nu' H_Z\varepsilon]) \\ &\approx (\mathrm{E}[x'H_Z x])^{-1}(\mathrm{E}[\nu' H_Z\varepsilon]) \\ &\approx (\mathrm{E}[(Z\gamma+\nu)'H_Z(Z\gamma+\nu)])^{-1}(\mathrm{E}[\nu' H_Z\varepsilon]) \\ &\approx (\mathrm{E}[\gamma' Z'H_Z Z\gamma] + \mathrm{E}[\nu' H_Z\nu])^{-1}(\mathrm{E}[\nu' H_Z\varepsilon]) \\ &\approx (\mathrm{E}[\gamma' Z' Z\gamma] + \mathrm{E}[\nu' H_Z\nu])^{-1}(\mathrm{E}[\nu' H_Z\varepsilon]). \end{aligned}$$

To make exposition even easier we will assume homoskedasticity for ν and ϵ. Then we can use arguments similar to what we did with Property B3:

$$\mathrm{E}[\underbrace{\nu' H_Z \nu}_{1\times 1}] = \mathrm{E}\left[\mathrm{tr}\left(\nu' H_Z\nu\right)\right] = \sigma_\nu^2 \mathrm{tr}(H_Z) = \sigma_\nu^2 L.$$

where L is the number of columns in Z. Similar trickery shows us that

$$\mathrm{E}[\nu' H_Z \varepsilon] = \sigma_{\varepsilon\nu}\mathrm{tr}(H_Z) = \sigma_{\varepsilon\nu} L.$$

As such

$$\mathrm{bias}(\hat{\beta}_{2sls}) \approx (\mathrm{E}[\gamma' Z' Z \gamma] + \sigma_\nu^2 L)^{-1} \sigma_{\varepsilon\nu} L$$

Let's multiply the left side by $\sigma_\nu^2/\sigma_\nu^2$ to get

$$\mathrm{bias}(\hat{\beta}_{2sls}) \approx \left(\frac{\mathrm{E}[\gamma' Z' Z \gamma]/L}{\sigma_\nu^2} + 1 \right)^{-1} \frac{\sigma_{\varepsilon\nu}}{\sigma_\nu^2}$$

Interestingly, the term

$$\frac{\mathrm{E}[\gamma' Z' Z \gamma]/L}{\sigma_\nu^2}$$

might look familiar to you. If we let $A = I_L$, $b = 0$, then the F test for linear hypothesis that $\gamma = 0$ is:

$$F = \frac{\hat{\gamma}'(Z'Z)\hat{\gamma}/L}{\hat{\sigma}_\nu^2}.$$

GOLLY! Where does this take us?

$$\begin{aligned}
\mathrm{E}[\hat{\beta}_{2sls}] - \beta &\approx \frac{\sigma_{\varepsilon\nu}}{\sigma_\nu^2} \left(\frac{\mathrm{E}[\gamma' Z' Z \gamma]/L}{\sigma_\nu^2} + 1 \right)^{-1} \\
&\approx \frac{\sigma_{\varepsilon\nu}}{\sigma_\nu^2} (F + 1)^{-1} \\
&\approx \frac{\sigma_{\varepsilon\nu}}{\sigma_\nu^2} \frac{1}{F + 1} \\
&\xrightarrow{p} \frac{\sigma_{\varepsilon\nu}}{\sigma_\nu^2} \frac{1}{F + 1}.
\end{aligned}$$

What does this tell us? First, it shows us where the bias in 2SLS comes from: weak instruments. Second, under ordinary asymptotics $F \to \infty$ as N increases and we obtain the consistency of 2SLS. However, if $\gamma = 0$ then $F = 0$ and the 2SLS estimator converges to $\beta + \frac{\sigma_{\varepsilon\nu}}{\sigma_\nu^2}$, which is also what OLS converges to in this case:

$$\hat{\beta}_{\mathrm{OLS}} = \frac{\mathrm{Cov}(y_i, x_i)}{\mathrm{Var}(x_i)} = \frac{\mathrm{Cov}(\beta x_i + \varepsilon_i, x_i)}{\mathrm{Var}(x_i)} = \beta + \frac{\mathrm{Cov}(\varepsilon_i, x_i)}{\mathrm{Var}(x_i)} = \beta + \frac{s_{\varepsilon\nu}}{s_\nu^2}.$$

If γ is just small, then 2SLS will be biased towards OLS, but consistent, but it may take a lot of observations to overcome this bias. Finally, if we add irrelevant instruments in an effort to get something, we make these problems worse as L increases, leading F to decrease.

We can think of $1/(1+F)$ as a measure of how biased the 2SLS estimates are towards OLS. The larger F is the better we're doing. In practice, checking this F test is often done by conducting the linear hypothesis that coefficients on the instruments for the endogenous regressors are 0 in the first stage regression (with robust standard errors if appropriate). We didn't derive this for heteroskedastic data, but it carries over the just identified case. In most cases, $F \geq 10$ is the long-running rule of thumb for a "good" place to be (new work suggests that $F \geq 105$ may be more accurate...) Before we turn to a "real" application let's see how this solves the problems we considered above.

6.2.2 Omitted variables

Let's return to the model where we consider wages as a function of schooling and intelligence is an omitted variable

$$\begin{aligned} w_i &= \underbrace{\beta_0 + \beta_2 \operatorname{E}[q_i]}_{\text{constant}} + \beta_1 s_i + \underbrace{\beta_2(q_i - \operatorname{E}[q_i]) + \varepsilon_i}_{\text{error}} \\ &= \alpha_0 + \beta_1 s_i + u_i \end{aligned}$$

Suppose we have an instrument k_i for s_i. As such we have $x_i = (1, s_i)$ and $z_i = (1, k_i)$. Assume that $\operatorname{E}[u_i z_i] = 0$ and that $\operatorname{rank}(\operatorname{E}[x_i z_i']) = 2$. This means that k_i is a good instrument (valid and relevant). Let's explore what this means for the relationships between k_i, s_i, and q_i. We have:

$$\begin{aligned} 0 = \operatorname{E}[u_i z_i] &= \begin{bmatrix} \operatorname{E}[u_i] \\ \operatorname{E}[(\beta_2(q_i - \operatorname{E}[q_i]) + \varepsilon_i)k_i] \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \operatorname{E}[(\beta_2(q_i - \operatorname{E}[q_i]) + \varepsilon_i)k_i] \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \beta_2 \operatorname{Cov}(q_i, k_i) \end{bmatrix} \\ \operatorname{Cov}(q_i, k_i) = 0 \end{aligned}$$

and

$$\begin{aligned}
\mathrm{E}[x_i z_i] &= \mathrm{E}\left[\begin{bmatrix}1\\ s_i\end{bmatrix}\begin{bmatrix}1 & k_i\end{bmatrix}\right]\\
&= \begin{bmatrix}1 & \mathrm{E}[k_i]\\ \mathrm{E}[s_i] & \mathrm{E}[s_i k_i]\end{bmatrix}\\
\det(\mathrm{E}[x_i z_i]) &= \mathrm{E}[s_i k_i] - \mathrm{E}[s_i]\,\mathrm{E}[k_i] = \mathrm{Cov}(s_i, k_i)\\
\mathrm{rank}(\mathrm{E}[x_i z_i]) = 2 &\iff \mathrm{Cov}(s_i, k_i) \neq 0
\end{aligned}$$

Okay, what have we learned? We showed that when OVB is the concern an instrument k_i is a good choice if it is

1. **Uncorrelated** with the unobserved confounding variable(s) $\mathrm{Cov}(q_i, k_i) = 0$ (validity)
2. **Correlated** with endogenous variable $\mathrm{Cov}(s_i, k_i) \neq 0$ (relevant)

Together this says we need to find a k such that it affects schooling but is unassociated with intelligence (or any other omitted variables). As such it affect wages **only** through its affect on schooling. Cost of schooling might be a candidate if want to control for intelligence, but be careful to consider how it might relate to other unobserved confounders.

6.2.3 Measurement error

One cool factoid is that two bad measures of x can be used to instrument for each other. Suppose we have

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

where $\mathrm{E}[\varepsilon_i | x_i] = 0$. But suppose we don't observe x_i, but instead we can only have two noisy measure

$$\begin{aligned}
\tilde{x}_i &= x_i + \nu_i\\
x_i^* &= x_i + \eta_i.
\end{aligned}$$

Plugging the first in, we want to see if we can use the second as an instrument

$$\begin{aligned}
y_i &= \beta_0 + \beta_1(\tilde{x}_i - \nu_i) + \varepsilon_i\\
&= \beta_0 + \beta_1 \tilde{x}_i + \varepsilon_i - \beta_1 \nu_i\\
&= \beta_0 + \beta_1 \tilde{x}_i + u_i
\end{aligned}$$

where $\mathrm{E}[\nu_i] = \mathrm{E}[\eta_i] = \mathrm{E}[x_i\eta_i] = \mathrm{E}[x_i\nu_i] = \mathrm{E}[\nu_i\varepsilon_i] = \mathrm{E}[\eta_i\varepsilon_i] = \mathrm{E}[\nu_i\eta_i] = 0$. This buys us:

$$\begin{aligned}
\mathrm{E}[x_i^* u_i] &= \mathrm{E}[x_i^*(\varepsilon_i - \beta_1\nu_i)] \\
&= \mathrm{E}[(x_i + \eta_i)(\varepsilon_i - \beta_1\nu_i)] \\
&= \mathrm{E}[x_i\varepsilon_i] - \beta_1\,\mathrm{E}[x_i\nu_i] + \mathrm{E}[\eta_i\varepsilon_i] - \beta_1\,\mathrm{E}[\eta_i\nu_i] \\
&= 0
\end{aligned}$$

This shows that x_i^* is a valid instrument. Is it relevant?

$$\begin{aligned}
\mathrm{Cov}(x_i^*, \tilde{x}_i) &= \mathrm{Cov}(x_i + \eta_i, x_i + \nu_i) \\
&= \mathrm{Cov}(x_i, x_i) + \mathrm{Cov}(x_i, \eta_i) + \mathrm{Cov}(x_i, \nu_i) + \mathrm{Cov}(\nu_i, \eta_i) \\
&= \mathrm{Var}(x_i) \\
&> 0
\end{aligned}$$

So long as x_i is not drawn from a degenerate distribution, this will be a valid instrument. Of course, any other valid and relevant instrument (correlates with x_i but only affects y_i through its effect on x_i) will correct for measurement error, but this is a neat choice if available (Note: you must have a second measure with error, you can't use $\tilde{x}$ to instrument itself).

6.2.4 Simultaneity

We now return to the Civil War and Growth conundrum. We want to show how instruments help us identify the parameters of interest in the structural equations

$$\begin{aligned}
w_i &= \beta_0 + \beta_1 g_i + \beta_2 x_i + \beta_3 k_i + \varepsilon_i \\
g_i &= \gamma_0 + \gamma_1 w_i + \gamma_2 x_i + \gamma_3 r_i + \nu_i.
\end{aligned}$$

Assume that only w_i and g_i are endogenous, such that $\mathrm{E}[\varepsilon_i] = \mathrm{E}[\varepsilon_i x_i] = \mathrm{E}[\varepsilon_i k_i] = \mathrm{E}[\varepsilon_i r_i] = \mathrm{E}[\nu_i] = \mathrm{E}[\nu_i x_i] = \mathrm{E}[\nu_i k_i] = \mathrm{E}[\nu_i r_i] = 0$

The reduced form is now

$$\begin{aligned}
w_i &= \frac{\beta_0 + \beta_1\gamma_0}{1 - \beta_1\gamma_1} + \frac{\beta_1\gamma_2 + \beta_2}{1 - \beta_1\gamma_1}x_i + \frac{\beta_1\gamma_3}{1 - \beta_1\gamma_1}r_i + \frac{\beta_3}{1 - \beta_1\gamma_1}k_i + \frac{\varepsilon_i + \beta_1\nu_i}{1 - \beta_1\gamma_1} \\
g_i &= \frac{\gamma_0 + \gamma_1\beta_0}{1 - \beta_1\gamma_1} + \frac{\gamma_1\beta_2 + \gamma_2}{1 - \beta_1\gamma_1}x_i + \frac{\gamma_1\beta_3}{1 - \beta_1\gamma_1}k_i + \frac{\gamma_3}{1 - \beta_1\gamma_1}r_i + \frac{\nu_i + \gamma_1\varepsilon_i}{1 - \beta_1\gamma_1}.
\end{aligned}$$

Here we now have eight estimates and eight parameters. We can solve this system and back out all the parameters from the reduced form regressions. We can do this because we have two instruments:

1. r_i affects war only through growth
2. k_i affect growth only through war

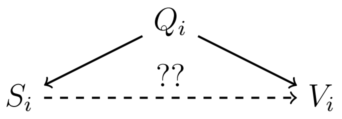
If we only have r_i then we can identify β_1 but not γ_1 or vice-versa. In practice, you could either regress both in reduced form and solve the estimates or use 2SLS on each equation to estimate the parameters of interest directly. There are other methods for estimating the system of equations as a pair (SUR or 3SLS), but these generally make bold efficiency claims at the cost of unrealistic modelling assumptions, so we we will skip these.

A candidate for r_i is rainfall (this is real paper). The argument is that in many developing countries agriculture is the primary driver of growth. Rain affects agriculture affects growth; the instrument is relevant. Miguel et al. 2014 argue that it can't possibly affect conflict on its own (and thus is valid). I'll leave that up to you to decide, but new work by Scott, Timm, and Florian shows that rainfall is generally problematic as an instrument because of concerns about spatial correlation. Also rainfall has been used as an instrument for **many** things since then, if we believe some of them then we lose all of them.

6.3 Applications

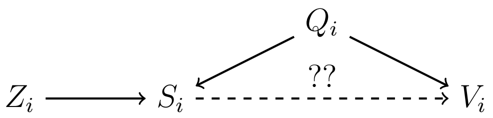
We will now consider an application from Gerber (1998). He wanted to know: What is the effect of campaign spending on vote share? Note this is a case where we have some strong unobserved confounding.

- Suppose that a candidate's unobservable quality (charisma, speaking skills, etc) affect vote share
- Suppose that high quality candidate raise more money
- Suppose that there is actually no relationship between spending and vote share



If we regress vote share on spending we might mistakenly conclude that spending is effective, when in reality we are just capturing the effect of the omitted variable. An instrumental variable approach might help us out here

We just need to find one that doesn't also affect Q.



```
library(readstata13)
library(lmtest)
library(sandwich)
library(car)
library(dplyr)
library(ivreg)

senate.data <- read.dta13("Rcode/datasets/senate_expanded.dta")

senate.data$inc_spend_capita <- senate.data$inc_spend /senate.data$st_pop
senate.data$ch_spend_capita <- senate.data$ch_spend / senate.data$st_pop
senate.data$log_inc_spend_capita <- log(1 + senate.data$inc_spend_capita)
senate.data$log_ch_spend_capita <- log(1 + senate.data$ch_spend_capita)
senate.data$ch_qual[senate.data$ch_qual ==5] <- 0 #fix a mistake


model.ols <- lm(inc_2p_share~log_inc_spend_capita + log_ch_spend_capita,
                data=senate.data)

#We want to know if spending effects vote share, we can
#consider the following tests

## H_0: log_inc_spend_capita = 0 Does incumbent spending have an effect?
## H_0: log_ch_spend_capita = 0  Does challenger spending have an effect?
coeftest(model.ols, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                       Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)           0.633282   0.010053  62.9932 < 2.2e-16 ***
## log_inc_spend_capita  0.076866   0.014430   5.3269 1.999e-07 ***
## log_ch_spend_capita  -0.205049   0.016841 -12.1760 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# On average, a 1% increase in challenger spending
# is associated with a 0.21 percentage point
# decrease in incumbent party
# vote share, holding everything else constant
#   (-0.205049/100)                * 100
# Interpret logged X (1%) | From 0-1 scale to pp
####################################################
# On average, a 1% increase in incumbent spending
# is associated with a 0.08 percentage point
# decrease in incumbent party
# vote share, holding everything else constant
#   (0.076866/100)                * 100
# Interpret logged X (1%) | From 0-1 scale to pp

## Does spending have an effect overall?
## H_0: log_inc_spend_capita = log_ch_spend_capita = 0
linearHypothesis(model.ols, c("log_inc_spend_capita=0", "log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_inc_spend_capita = 0
## log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    294
## 2    292  2 76.777 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## Are challenger and incumbent spending equally effective?
## H_0: log_inc_spend_capita + log_ch_spend_capita = 0
linearHypothesis(model.ols, c("log_inc_spend_capita+log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_inc_spend_capita  + log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    293
## 2    292  1 80.147 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## In this basic regression (no controls) we see that challenger
## spending is still more effective, but both are effective

# We have an effort to measure of challenger quality let's include that
# and some other controls
model.ols2 <- lm(inc_2p_share~log_inc_spend_capita + log_ch_spend_capita
                 +factor(ch_qual)+inc_tenure+st_uemp,
                 data=senate.data)
coeftest(model.ols2, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                          Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)           0.65792887  0.01902152  34.5887 < 2.2e-16 ***
## log_inc_spend_capita  0.05683223  0.01534803   3.7029 0.0002557 ***
## log_ch_spend_capita  -0.17249895  0.01646537 -10.4765 < 2.2e-16 ***
## factor(ch_qual)1     -0.02655789  0.01093663  -2.4283 0.0157847 *
## factor(ch_qual)2     -0.04630408  0.01514575  -3.0572 0.0024452 **
## factor(ch_qual)3     -0.05034232  0.01370930  -3.6721 0.0002870 ***
## factor(ch_qual)4     -0.05195259  0.01232827  -4.2141 3.366e-05 ***
## inc_tenure            0.00162089  0.00067379   2.4056 0.0167801 *
## st_uemp              -0.00296628  0.00200365  -1.4804 0.1398565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model.ols2,
                 c("log_inc_spend_capita=0", "log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_inc_spend_capita = 0
## log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita + factor(ch_qual)
##     inc_tenure + st_uemp
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    288
## 2    286  2 64.036 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model.ols2,
                 c("log_inc_spend_capita+log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_inc_spend_capita  + log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita + factor(ch_qual)
##     inc_tenure + st_uemp
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    287
## 2    286  1 77.351 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## In this second regression with controls we see that challenger
## spending is more effective, but both are still effective
## Effects are slightly smaller though

## Gerber instruments for incumbent and challenger spending using
# Challenger Wealth
# State population
# incumbent spending in the previous election.
## Let's try it with the first two.

# With ivreg you specify the formulas as
##    y  ~  X | Z
## outcome ~ exogenous + endogenous  | exogenous + instruments
model.2sls <- ivreg(inc_2p_share~
                      log_inc_spend_capita + log_ch_spend_capita|
                      ch_wealthy+ st_pop,
                    data=senate.data,
                    x=T, y=T)

#Unlike lm, ivreg allows you to pop a robust covariance matrix into summary
```

```r
summary(model.2sls, vcov=vcovHC)
```

```
##
## Call:
## ivreg(formula = inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita |
##     ch_wealthy + st_pop, data = senate.data, y = T, x = T)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.204309 -0.066126  0.001046  0.049266  0.403047
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.56409    0.01441  39.147  < 2e-16 ***
## log_inc_spend_capita  0.17092    0.02694   6.345 8.79e-10 ***
## log_ch_spend_capita  -0.20001    0.04420  -4.525 8.90e-06 ***
##
## Diagnostic tests:
##                                           df1 df2 statistic  p-value
## Weak instruments (log_inc_spend_capita)     2 283     25.89 4.74e-11 ***
## Weak instruments (log_ch_spend_capita)      2 283     21.79 1.58e-09 ***
## Wu-Hausman                                  2 281     11.79 1.21e-05 ***
## Sargan                                      0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08465 on 283 degrees of freedom
## Multiple R-Squared: 0.2442,  Adjusted R-squared: 0.2388
## Wald test: 20.17 on 2 and 283 DF,  p-value: 6.445e-09
```

```r
linearHypothesis(model.2sls,
                 c("log_inc_spend_capita=0", "log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
```

```
## log_inc_spend_capita = 0
## log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita | ch_wealthy +
##     st_pop
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    285
## 2    283  2 20.175 6.445e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model.2sls,
                 c("log_inc_spend_capita+log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_inc_spend_capita  + log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita | ch_wealthy +
##     st_pop
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1    284
## 2    283  1 0.7958 0.3731
```

```r
## Now things have changed?
compareCoefs(model.ols, model.2sls)
```

```
## Warning in compareCoefs(model.ols, model.2sls): models to be compared are of
## different classes
```

```
## Calls:
## 1: lm(formula = inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita,
##   data = senate.data)
## 2: ivreg(formula = inc_2p_share ~ log_inc_spend_capita +
##   log_ch_spend_capita | ch_wealthy + st_pop, data = senate.data, y = T, x = T)
##
##                       Model 1 Model 2
## (Intercept)           0.63328 0.56409
## SE                    0.00936 0.02034
##
## log_inc_spend_capita   0.0769  0.1709
## SE                     0.0154  0.0380
##
## log_ch_spend_capita   -0.2050 -0.2000
## SE                     0.0162  0.0626
##
```

```
## Both are still effective, but the effects
## are nearly identical (big jump in the effect of inc spending)

# On average, a 1% increase in challenger spending
# is associated with a 0.20 percentage point
# decrease in incumbent party
# vote share, holding everything else constant
#   (-0.2000/100)                 * 100
# Interpret logged X (1%) | From 0-1 scale to pp
####################################################
# On average, a 1% increase in incumbent spending
# is associated with a 0.17 percentage point
# decrease in incumbent party
# vote share, holding everything else constant
#   (0.1709/100)                 * 100
# Interpret logged X (1%) | From 0-1 scale to pp
```

```r
## We should always check the "first stage" regression to see if we can
## find relevance
first.stage.ch <- lm(log_inc_spend_capita~ ch_wealthy+ st_pop,
                     data=model.2sls$model)
first.stage.inc <- lm(log_ch_spend_capita~ ch_wealthy+ st_pop,
                      data=model.2sls$model)
coeftest(first.stage.ch, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  8.5887e-01  3.4619e-02 24.8093 < 2.2e-16 ***
## ch_wealthy   1.8778e-02  5.0697e-02  0.3704    0.7114
## st_pop      -3.3400e-08  4.6577e-09 -7.1710 6.516e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(first.stage.inc, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  4.1839e-01  3.2842e-02 12.7393 < 2.2e-16 ***
## ch_wealthy   2.5888e-01  5.7743e-02  4.4834 1.069e-05 ***
## st_pop      -1.4806e-08  3.5459e-09 -4.1754 3.964e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## Check the joint test of the instruments
linearHypothesis(first.stage.ch,
                 c("ch_wealthy=0", "st_pop=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
```

```
## ch_wealthy = 0
## st_pop = 0
##
## Model 1: restricted model
## Model 2: log_inc_spend_capita ~ ch_wealthy + st_pop
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    285
## 2    283  2 25.886 4.741e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(first.stage.inc,
                 c("ch_wealthy=0", "st_pop=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## ch_wealthy = 0
## st_pop = 0
##
## Model 1: restricted model
## Model 2: log_ch_spend_capita ~ ch_wealthy + st_pop
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    285
## 2    283  2 21.788 1.581e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model.2sls, vcovHC)
```

```
##
```

```
## Call:
## ivreg(formula = inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita |
##     ch_wealthy + st_pop, data = senate.data, y = T, x = T)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.204309 -0.066126  0.001046  0.049266  0.403047
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.56409    0.01441  39.147  < 2e-16 ***
## log_inc_spend_capita  0.17092    0.02694   6.345 8.79e-10 ***
## log_ch_spend_capita  -0.20001    0.04420  -4.525 8.90e-06 ***
##
## Diagnostic tests:
##                                          df1 df2 statistic  p-value
## Weak instruments (log_inc_spend_capita)   2 283     25.89 4.74e-11 ***
## Weak instruments (log_ch_spend_capita)    2 283     21.79 1.58e-09 ***
## Wu-Hausman                                2 281     11.79 1.21e-05 ***
## Sargan                                    0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08465 on 283 degrees of freedom
## Multiple R-Squared: 0.2442,  Adjusted R-squared: 0.2388
## Wald test: 20.17 on 2 and 283 DF,  p-value: 6.445e-09
```

```
## IV-REG provides some diagnostics for us:
#' 1. Weak instruments tests are the first stage Fs (want at least 10)
#' 2. Wu-Hausman is a test of similarity between the OLS and the
#'    2SLS estimates are.
#'    If we can't reject the null that they are same then
#'    that means that our instrument hasn't changed the results
#'    from the possibly inconsistent OLS.
#'    Another way to think about this:
#'    If we reject then we have
#'    evidence that X is endogenous and the 2SLS is helping.
#'    If we fail to reject then either:
#'    X is actually exogenous or the instrument is too weak to make a diff.
#' 3. Sargan's test is only for over identified models.
#'    You want to **not** reject here as the null is that all
#'    instruments are valid: cor(epsilon, Z)=0.

## Here we find evidence that the instruments are relevant.
## Sadly you can never know about validity.
## You'll always have to argue that your instruments are valid

#ASIDE: See why they call it "Two-Stage"
second.stage <- lm(model.2sls$y~first.stage.ch$fitted.values+
                     first.stage.inc$fitted.values)
cbind(second.stage$coefficients,
      model.2sls$coefficients)
```

```
##                                      [,1]       [,2]
## (Intercept)                     0.5640934  0.5640934
## first.stage.ch$fitted.values   0.1709181  0.1709181
## first.stage.inc$fitted.values -0.2000134 -0.2000134
```

```
## Let's try a model with controls
model.2sls.full <- ivreg(inc_2p_share~
                           log_inc_spend_capita + log_ch_spend_capita
                         +factor(ch_qual)+inc_tenure+st_uemp|
                           ch_wealthy+ st_pop
                         +factor(ch_qual)+inc_tenure+st_uemp,
                         data=senate.data,
                         x=TRUE,y=TRUE)
summary(model.2sls.full, vcov=vcovHC)
```

```
##
## Call:
## ivreg(formula = inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita +
##     factor(ch_qual) + inc_tenure + st_uemp | ch_wealthy + st_pop +
##     factor(ch_qual) + inc_tenure + st_uemp, data = senate.data,
##     y = TRUE, x = TRUE)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.201643 -0.050693 -0.002702  0.047004  0.358452
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.5973961  0.0234696  25.454  < 2e-16 ***
## log_inc_spend_capita  0.1630486  0.0299795   5.439 1.18e-07 ***
## log_ch_spend_capita  -0.2376369  0.0391035  -6.077 4.04e-09 ***
## factor(ch_qual)1     -0.0241064  0.0125168  -1.926  0.05514 .
## factor(ch_qual)2     -0.0350977  0.0161891  -2.168  0.03101 *
## factor(ch_qual)3     -0.0481496  0.0161158  -2.988  0.00306 **
## factor(ch_qual)4     -0.0438525  0.0139455  -3.145  0.00184 **
## inc_tenure            0.0009106  0.0006520   1.397  0.16362
## st_uemp              -0.0005050  0.0022878  -0.221  0.82547
##
## Diagnostic tests:
##                                          df1 df2 statistic  p-value
## Weak instruments (log_inc_spend_capita)   2 277    38.712 1.49e-15 ***
## Weak instruments (log_ch_spend_capita)    2 277    39.518 7.97e-16 ***
## Wu-Hausman                                2 275     6.708  0.00143 **
## Sargan                                    0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07813 on 277 degrees of freedom
## Multiple R-Squared: 0.3697,  Adjusted R-squared: 0.3515
## Wald test: 17.45 on 8 and 277 DF,  p-value: < 2.2e-16
```

```
linearHypothesis(model.2sls.full,
                 c("log_inc_spend_capita=0", "log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
```

```
## log_inc_spend_capita = 0
## log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita + factor(ch_qual)
##     inc_tenure + st_uemp | ch_wealthy + st_pop + factor(ch_qual) +
##     inc_tenure + st_uemp
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    279
## 2    277  2 19.322 1.395e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(model.2sls.full,
                 c("log_inc_spend_capita+log_ch_spend_capita=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## log_inc_spend_capita  + log_ch_spend_capita = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita + factor(ch_qual)
##     inc_tenure + st_uemp | ch_wealthy + st_pop + factor(ch_qual) +
##     inc_tenure + st_uemp
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F   Pr(>F)
## 1    278
## 2    277  1 8.4079 0.004035 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Now things are somewhat back to reduced form version
## We find that spending is effective on both sides
## We find that challenger spending is more effective
## BUT the effect of incumbent spending  is much larger than reduced
## form suggested

model.ols2$coefficients[2] # A 1% increase in incumbent spending is associated
```

```
## log_inc_spend_capita
##           0.05683223
```

```
# with only a .06 percentage point gain in vote share holding other variables constant

model.2sls.full$coefficients[2] # A 1% increase in incumbent spending is associated
```

```
## log_inc_spend_capita
##            0.1630486
```

```
# with a 0.16 percentage point gain, holding all else equal
# We can make a causal argument if we believe the instruments
# If you take the causal inference class latter in your time here,
# you'll discuss this interpretation in more detail.



## Let's check the first stages with the fuller (but not Fuller; that's an IV joke) mo
ivreg.data <- model.2sls.full$model
colnames(ivreg.data)[colnames(ivreg.data)=="factor(ch_qual)"] <- "ch_qual"
first.stage.inc.full <- lm(log_inc_spend_capita~
                             ch_wealthy+ st_pop
                           +factor(ch_qual)+inc_tenure+st_uemp,
                           data=ivreg.data, x=T)
first.stage.ch.full <- lm(log_ch_spend_capita~
                            ch_wealthy+ st_pop
                          +factor(ch_qual)+inc_tenure+st_uemp,
                          data=ivreg.data, x=T)
coeftest(first.stage.ch.full, vcov=vcovHC)
```

```
##
```

```
## t test of coefficients:
##
##                     Estimate  Std. Error t value  Pr(>|t|)    
## (Intercept)       5.3267e-01  6.7920e-02  7.8426 9.602e-14 ***
## ch_wealthy        3.7330e-01  6.1089e-02  6.1107 3.354e-09 ***
## st_pop           -1.8939e-08  2.9563e-09 -6.4063 6.376e-10 ***
## factor(ch_qual)1  4.8560e-02  4.3358e-02  1.1200   0.26369    
## factor(ch_qual)2  1.9098e-01  8.0541e-02  2.3711   0.01842 *  
## factor(ch_qual)3  4.3483e-01  6.3821e-02  6.8132 5.938e-11 ***
## factor(ch_qual)4  3.5328e-01  6.1118e-02  5.7802 2.007e-08 ***
## inc_tenure       -1.5646e-03  2.3523e-03 -0.6651   0.50652    
## st_uemp          -3.9387e-02  8.6037e-03 -4.5779 7.100e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(first.stage.inc.full, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                     Estimate  Std. Error t value  Pr(>|t|)    
## (Intercept)       1.0164e+00  7.6248e-02 13.3306 < 2.2e-16 ***
## ch_wealthy        1.0092e-01  4.8227e-02  2.0925 0.0373009 *  
## st_pop           -3.3933e-08  4.0639e-09 -8.3500 3.300e-15 ***
## factor(ch_qual)1  6.3122e-02  5.1035e-02  1.2368 0.2171934    
## factor(ch_qual)2  7.6604e-02  8.0646e-02  0.9499 0.3429991    
## factor(ch_qual)3  2.8242e-01  6.7532e-02  4.1820 3.880e-05 ***
## factor(ch_qual)4  1.8869e-01  5.6038e-02  3.3673 0.0008668 ***
## inc_tenure        1.9347e-03  2.6003e-03  0.7440 0.4574988    
## st_uemp          -4.6545e-02  8.9267e-03 -5.2142 3.618e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Check the joint test of the instruments
linearHypothesis(first.stage.inc.full,
                 c("ch_wealthy=0", "st_pop=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## ch_wealthy = 0
## st_pop = 0
##
## Model 1: restricted model
## Model 2: log_inc_spend_capita ~ ch_wealthy + st_pop + factor(ch_qual) +
##     inc_tenure + st_uemp
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    279
## 2    277  2 38.712 1.495e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linearHypothesis(first.stage.ch.full,
                 c("ch_wealthy=0", "st_pop=0"),
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## ch_wealthy = 0
## st_pop = 0
##
## Model 1: restricted model
## Model 2: log_ch_spend_capita ~ ch_wealthy + st_pop + factor(ch_qual) +
##     inc_tenure + st_uemp
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1    279
## 2    277  2 39.518 7.971e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Check the reduced form
model.reduced <- lm(inc_2p_share~ ch_wealthy+ st_pop
                    +factor(ch_qual)+inc_tenure+st_uemp,
                    data=senate.data)
coeftest(model.reduced, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                     Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)       6.3654e-01  1.9909e-02 31.9730 < 2.2e-16 ***
## ch_wealthy       -7.2255e-02  1.6113e-02 -4.4844 1.072e-05 ***
## st_pop           -1.0322e-09  7.7332e-10 -1.3348  0.183030
## factor(ch_qual)1 -2.5354e-02  1.2930e-02 -1.9608  0.050903 .
## factor(ch_qual)2 -6.7990e-02  2.1112e-02 -3.2205  0.001432 **
## factor(ch_qual)3 -1.0543e-01  1.5184e-02 -6.9437 2.717e-11 ***
## factor(ch_qual)4 -9.7038e-02  1.4656e-02 -6.6212 1.844e-10 ***
## inc_tenure        1.5978e-03  6.9047e-04  2.3141  0.021392 *
## st_uemp           1.2657e-03  2.4603e-03  0.5144  0.607359
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Another type of Reduced form (Wu-Hausman)
model.wu <- lm(inc_2p_share~ log_inc_spend_capita+ log_ch_spend_capita
                    +factor(ch_qual)+inc_tenure+st_uemp
                      +first.stage.inc.full$residuals
                    +first.stage.ch.full$residuals,
                    data=ivreg.data)
# What are these residuals? They're the part of the endogeneous regressors with the
# variance from the instrument removed (i.e., just the endogenous parts)
# So here we control for endogeneity using these residuals
# Wu-Hausman checks if these are significant. If so then evidence of endogeneity
# If not; than the other estimates should match the OLS estimates
linearHypothesis(model.wu,
                  c("first.stage.inc.full$residuals=0", "first.stage.ch.full$residuals=0"
                 vcov=vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## first.stage.inc.full$residuals = 0
## first.stage.ch.full$residuals = 0
##
## Model 1: restricted model
## Model 2: inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita + factor(ch_qual)
##     inc_tenure + st_uemp + first.stage.inc.full$residuals + first.stage.ch.full$resid
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F   Pr(>F)
## 1    277
## 2    275  2 6.7079 0.001431 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#############Over-Identified example##################################
## I want to add the total amount spend by all sides
## First we need to generate our new instrument
## I'm going to use dplyr's lag function and data manipulation tools
## because it's easier than figuring it out one state at a time
#################Aside to generate a "lag" ###########################
senate.data <- senate.data %>%
  arrange(st_name, year) %>%
  group_by(st_name) %>%
  mutate(ll_total_spending_captia = log(lag((ch_spend+inc_spend)/st_pop))) %>%
  ungroup()
#####################################################################

model.2sls.over <- ivreg(inc_2p_share~
                           log_inc_spend_capita + log_ch_spend_capita
                          +factor(ch_qual)+inc_tenure+st_uemp|
                           ch_wealthy+ st_pop +ll_total_spending_captia
```

```
                         +factor(ch_qual)+inc_tenure+st_uemp,
                         data=senate.data, x=T, y=T)
summary(model.2sls.over, vcov=vcovHC)
```

```
## 
## Call:
## ivreg(formula = inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita + 
##     factor(ch_qual) + inc_tenure + st_uemp | ch_wealthy + st_pop + 
##     ll_total_spending_captia + factor(ch_qual) + inc_tenure + 
##     st_uemp, data = senate.data, y = T, x = T)
## 
## Residuals:
##       Min       1Q   Median       3Q      Max 
## -0.21473 -0.05286 -0.00016  0.04456  0.26800 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            0.6059085  0.0285627  21.213  < 2e-16 ***
## log_inc_spend_capita   0.1957676  0.0365430   5.357 2.44e-07 ***
## log_ch_spend_capita   -0.3076223  0.0462674  -6.649 3.08e-10 ***
## factor(ch_qual)1      -0.0189750  0.0160959  -1.179   0.2399    
## factor(ch_qual)2      -0.0134656  0.0212452  -0.634   0.5270    
## factor(ch_qual)3      -0.0090129  0.0231686  -0.389   0.6977    
## factor(ch_qual)4      -0.0198663  0.0197072  -1.008   0.3147    
## inc_tenure             0.0011473  0.0007464   1.537   0.1259    
## st_uemp               -0.0047359  0.0028343  -1.671   0.0964 .  
## 
## Diagnostic tests:
##                                          df1 df2 statistic  p-value    
## Weak instruments (log_inc_spend_capita)   3 188    31.730  < 2e-16 ***
## Weak instruments (log_ch_spend_capita)    3 188    25.973 4.21e-14 ***
## Wu-Hausman                                2 187    18.966 3.17e-08 ***
## Sargan                                    1  NA     0.708      0.4    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

```
## Residual standard error: 0.08288 on 189 degrees of freedom
## Multiple R-Squared:  0.29,   Adjusted R-squared: 0.2599
## Wald test: 14.67 on 8 and 189 DF,  p-value: < 2.2e-16
```

```
# check that Sargin test, we fail to reject the null of valid instruments
# this is a good thing

# check the reduced form too
model.reduced.over<- lm(inc_2p_share~ ch_wealthy+ st_pop +ll_total_spending_captia
                        +factor(ch_qual)+inc_tenure+st_uemp,
                        data=senate.data)
coeftest(model.reduced.over, vcov=vcovHC)
```

```
##
## t test of coefficients:
##
##                             Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)               6.6578e-01  2.3281e-02 28.5982 < 2.2e-16 ***
## ch_wealthy               -1.0379e-01  2.0201e-02 -5.1380 6.917e-07 ***
## st_pop                   -1.0314e-09  1.0538e-09 -0.9787   0.32897
## ll_total_spending_captia  1.2392e-03  7.7499e-03  0.1599   0.87313
## factor(ch_qual)1         -3.4579e-02  1.4777e-02 -2.3401   0.02033 *
## factor(ch_qual)2         -8.2156e-02  2.6487e-02 -3.1018   0.00222 **
## factor(ch_qual)3         -1.1320e-01  1.7415e-02 -6.5002 7.020e-10 ***
## factor(ch_qual)4         -1.1303e-01  1.6351e-02 -6.9125 7.195e-11 ***
## inc_tenure                1.6468e-03  7.1245e-04  2.3114   0.02190 *
## st_uemp                  -1.6670e-03  2.9317e-03 -0.5686   0.57029
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let us now briefly consider the validity of the instruments. For challenger wealth to be relevant and valid it must influence spending but only influence vote share through spending. The former makes sense to me, after all challengers who are wealthy can spend their own money so their spending rises and incumbents rise to match it (this relationship was fairly strong in the first stage for challenger spending and sorta strong for incumbent spending). The latter is trickier, does wealth affect vote share outside of money? Wealth might correlate with other components of quality that aren't captured in the dummies. After all wealthy people might just be seen as "winners" which can affect votes beyond the spending effect.

Regarding state population it must be that it affects per capita spending but only affects vote share through its affect on spending. The argument for its useful is that there is evidence that per capita spending is lower in more populous states (relevance) and that being in a small state attracts outside investment (people see their money as more impactful), which is then rolled into spending and into votes. However, what if it's also easier to just do door-to-door canvasing in small states? Iowa and New Hampshire go first in presidential races because their smallness is seen as leveling the playing field for less wealthy candidates and as such population may influence a candidates ability to drum up votes the old-fashioned way.

What about the amount spent in the previous race? To be relevant and valid it must be that this affects current spending but only affects vote share through its affect on spending. Relevance makes sense to me, but validity? What do you think?

Overall final advice:

1. The best thing to do is to just use the best instruments you have with just identified 2SLS ($\dim(z_i) = \dim(x_i)$. Just identified 2SLS has very good properties even under slight weakness.
2. Check the first stages on everything. You probably want the F test (with robust standard errors if appropriate) on the instruments to be at least 10 to proceed with 2SLS. Otherwise you should look for a new instrument, or you can look up methods that are more robust to weak instruments.
3. If you are overidentified, check the 2SLS against the LIML (see next section). If there are big differences you might have a problem with weakness.
4. Always check your reduced form model using OLS. If the relationship you expect for your instrument doesn't appear in reduced form, it might not be a good choice.

6.4 Bonus Topic: Working with many weak instruments: Limited Information Maximum Likelihood

If you have many weak instruments you may want to consider using limited information maximum likelihood (LIML). In the just identified case these will be exactly the same as 2SLS, but in the over identified case LIML is more robust to weak instruments. The LIML has a closed form solution based where

$$\hat{\beta}_{LIML} = \left[X'\left(I - kM_Z\right)X\right]^{-1} X'\left(I - kM_Z\right)y.$$

Note that M_Z is the first stage "residual maker" $(I - H_Z)$. This format is called a k class estimator.

When $k = 1$ we have 2SLS. For LIML we first partition $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$ where X_1 are exogenous and X_2 are engodenous, then set k to the minimum eigenvalue of

$$A' \left(\tilde{Y}' M_{X1} \tilde{Y} \right) A,$$

where

$$\begin{aligned} \tilde{Y} &= \begin{bmatrix} y & X_2 \end{bmatrix} \\ A &= \left(\tilde{Y}' M_Z \tilde{Y} \right)^{-1/2} \end{aligned}$$

I provide R code online to handle this for you.

```
## Picking up from the over identified case above
source("limlfit.r")

# We can use the 2SLS fit to get the data we want
X <- model.2sls.over$x$regressors
Z <- model.2sls.over$x$instruments
y <- model.2sls.over$y
liml.out <- liml.fit(X1=X[, -c(2:3)], #Exogenous X
                     X2=X[,2:3], #Endogenous X
                     Z=Z, #Instruments
                     y=y, #outcome
                     vcov = "robust") #robust to heteroskedasticity and weak instrument
signif(liml.out$coef.table,2)
```

```
##                        liml.est se.liml z.tests   p.val
## (Intercept)              0.6000 0.02900   21.00 5.3e-95
## log_inc_spend_capita     0.2000 0.04400    4.50 5.4e-06
## log_ch_spend_capita     -0.3100 0.06100   -5.10 4.0e-07
## factor(ch_qual)1        -0.0190 0.01600   -1.20 2.4e-01
## factor(ch_qual)2        -0.0130 0.02200   -0.59 5.6e-01
## factor(ch_qual)3        -0.0081 0.02700   -0.30 7.6e-01
## factor(ch_qual)4        -0.0190 0.02300   -0.84 4.0e-01
## inc_tenure               0.0011 0.00073    1.60 1.2e-01
## st_uemp                 -0.0047 0.00290   -1.70 9.8e-02
```

```
liml.out$k
```

```
## [1] 1.003567
```

```
# k is pretty close to 1 that's a good sign for these instruments
# this means that LIML and 2SLS mostly agree
# When they disagree that can be a sign of weak instruments




#With the just identified case LIML and 2SLS are always identical
X <- model.2sls.full$x$regressors
Z <- model.2sls.full$x$instruments
y <- model.2sls.full$y
liml.just <- liml.fit(X1=X[, -c(2:3)], #Exogenous X
                      X2=X[,2:3], #Endogenous X
                      Z=Z, #Instruments
                      y=y, #outcome
                      vcov = "robust") #robust to heteroskedasticity and weak instrument
signif(liml.just$coef.table,2)
```

```
##                        liml.est se.liml z.tests    p.val
## (Intercept)             0.60000 0.02400   25.00 4.5e-139
## log_inc_spend_capita    0.16000 0.03400    4.80  1.4e-06
## log_ch_spend_capita    -0.24000 0.04600   -5.20  2.2e-07
## factor(ch_qual)1       -0.02400 0.01200   -1.90  5.3e-02
## factor(ch_qual)2       -0.03500 0.01600   -2.20  2.5e-02
## factor(ch_qual)3       -0.04800 0.01700   -2.80  4.4e-03
## factor(ch_qual)4       -0.04400 0.01500   -3.00  2.8e-03
## inc_tenure              0.00091 0.00065    1.40  1.6e-01
## st_uemp                -0.00050 0.00230   -0.22  8.2e-01
```

```
liml.just$k #exactly 1 when just identified
```

```
## [1] 1
```

```
signif(coeftest(model.2sls.full, vcov=vcovHC),2)
```

```
##
```

```
## t test of coefficients:
##
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.60000    0.02300   25.00  < 2e-16 ***
## log_inc_spend_capita  0.16000    0.03000    5.40  1.2e-07 ***
## log_ch_spend_capita  -0.24000    0.03900   -6.10  4.0e-09 ***
## factor(ch_qual)1     -0.02400    0.01300   -1.90   0.0550 .
## factor(ch_qual)2     -0.03500    0.01600   -2.20   0.0310 *
## factor(ch_qual)3     -0.04800    0.01600   -3.00   0.0031 **
## factor(ch_qual)4     -0.04400    0.01400   -3.10   0.0018 **
## inc_tenure            0.00091    0.00065    1.40   0.1600
## st_uemp              -0.00050    0.00230   -0.22   0.8300
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6.5 Bonus Topic: Moment estimators for instrumental variables

A third way to consider the instrumental variable problem is with a (generalized) method of moments estimator. This approach can lead to efficiency gains over the the above. Recall from math camp, that the method of moments works by equating population moments to sample moments. Suppose that we have data y that is drawn from unknown normal distribution with parameters θ. We define the rth moment of y as

$$\mu_k(\theta) = \mathrm{E}\left[y^k\right].$$

For a sample of size N we can build empirical moments as

$$\hat{\mu}_k(\theta) = \frac{1}{N}\sum_{i=1}^{N} y_i^k.$$

The method of moments estimates θ by equating these sample moments to population moments. In this case we need as many sample moments as we have empirical moments. Let

$\theta = (\mu, \sigma^2)$ then we need the first two population and the first two empirical moments:

$$\begin{aligned}
\mu_1(\theta) &= \mathrm{E}[y] = \mu & \hat{\mu}_1(\theta) &= \frac{1}{N}\sum_{i=1}^N y_i \\
\mu_2(\theta) &= \mathrm{E}[y^2] = \mathrm{Var}(y) + \mathrm{E}[y]^2 = \sigma^2 + \mu^2 & \hat{\mu}_2(\theta) &= \frac{1}{N}\sum_{i=1}^N y_i^2.
\end{aligned}$$

To find $\hat{\theta}_{MoM}$ we set these equal and solve

$$\begin{aligned}
\hat{\mu}_{MoM} &= \frac{1}{N}\sum_{i=1}^N y_i \\
\hat{\sigma}^2_{MoM} &= \frac{1}{N}\sum_{i=1}^N (y_i - \mu)^2.
\end{aligned}$$

We can make this a little more formal. Let $g(x_i, y_i; \beta)$ be an objective function that takes in data (x_i, y_i) and a guess at β and measures how close the population moments to the sample moments are for this value of β. The population version of our conditions $\mathrm{E}[g(x_i, y_i; \beta)] = 0$ will hold only if we guess the true β (for exposition call it β^* for the moment). For the linear model, we can use the moment condition $\mathrm{E}[x_i\varepsilon_i] = 0$, this makes g:

$$\begin{aligned}
g(x_i, y_i; \beta) &= x_i(y_i - x_i'\beta) - \mathrm{E}[x_i\varepsilon_i] \\
&= x_i(y_i - x_i'\beta)
\end{aligned}$$

We then find the expectation

$$\begin{aligned}
\mathrm{E}[g(x_i, y_i; \beta)] &= \mathrm{E}[x_i(y_i - x_i'\beta)] = 0 \\
&= \mathrm{E}_x[\mathrm{E}[x_i(y_i - x_i'\beta)|x_i]] \\
&= \mathrm{E}_x[x_i\,\mathrm{E}[y_i|x_i]] - \mathrm{E}[x_ix_i']\beta \\
&= \mathrm{E}[x_ix_i'\beta^*] - \mathrm{E}[x_ix_i']\beta \\
&= \mathrm{E}[x_ix_i']\beta^* - \mathrm{E}[x_ix_i']\beta \\
&= \mathrm{E}[x_ix_i'](\beta^* - \beta)
\end{aligned}$$

Here, the first line is the population equivilent of the OLS first order conditions from way back when. Additionally, if $\mathrm{E}[x_ix_i']$ has full rank then $\mathrm{E}[g(x_i, y_i; \beta)] = 0$ only if $\beta = \beta^*$. This tells us that these moment conditions will identify the parameters only if $\mathrm{E}[x_ix_i']$ has full rank.

Once we have found g then the GMM estimator is defined as value of β that satisfies the sample version of the population moments:

$$\frac{1}{N}\sum_{i=1}^{N} g(x_i, y_i; \beta) = 0$$

which is the sample version of our $\mathrm{E}[g(x_i, y_i; \beta)]$. Again for OLS this becomes

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N} g(x_i, y_i; \beta) &= 0 \\
\frac{1}{N}\sum_{i=1}^{N} x_i(y_i - x_i'\beta) &= 0 \\
\hat{\beta}_{MoM} &= \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i'\right)^{-1} \frac{1}{N}\sum_{i=1}^{N} x_i y_i \\
\hat{\beta}_{MoM} &= \hat{\beta}_{OLS}
\end{aligned}$$

OLS is the MoM. Any MLE can also be written as a moment problem where the moments are the FOC.

$$g(x_i, y_i; \theta) = D_\theta \log(L(\theta; x_i, y_i))$$

With regular MoM we had as many moments as parameters (OLS is exactly identified $\dim(g(x_i, y_i; \beta)) = K = \dim(\beta)$). With generalized MoM (GMM) we need to have at least as many moments as parameters, but we can have more $\dim(g(x_i, y_i; \theta)) \geq \dim(\theta)$.

The GMM estimator is defined as

$$\hat{\theta}_{GMM} = \underset{\theta}{\operatorname{argmin}} \left(\frac{1}{N}\sum_{i=1}^{N} g(x_i, y_i; \theta)\right) W \left(\frac{1}{N}\sum_{i=1}^{N} g(x_i, y_i; \theta)\right).$$

Here W is a positive definite weighting matrix. For example, let

$$\begin{aligned}
y_i &= \beta' x_i + \varepsilon_i \\
\mathrm{E}[\varepsilon_i | z_i] &= 0 \\
g(x_i, y_i, z_i; \beta) &= z_i(y_i - \beta' x_i) \\
W &= \left(\frac{1}{N}\sum_{i=1}^{N} z_i z_i'\right)^{-1}.
\end{aligned}$$

Then we have

$$\begin{aligned}
\mathrm{E}[g(x_i, y_i, z_i; \beta)] &= \mathrm{E}_z[\mathrm{E}[g(x_i, y_i, z_i; \beta)|z_i]] \\
&= \mathrm{E}_z[\mathrm{E}[z_i(y_i - \beta' x_i)|z_i]] \\
&= \mathrm{E}_z[\mathrm{E}[z_i(\beta^{*\prime} x_i + \varepsilon_i - \beta' x_i)|z_i]] \\
&= \mathrm{E}_z[\mathrm{E}[z_i x_i' \beta^* + \varepsilon_i - z_i x_i' \beta | z_i]] \\
&= \mathrm{E}_z[\mathrm{E}[z_i x_i' \beta^* - z_i x_i' \beta | z_i] + \mathrm{E}[z_i \varepsilon_i | z_i]] \\
&= \mathrm{E}_z[\mathrm{E}[z_i x_i' | z_i](\beta^* - \beta) \\
&= \mathrm{E}[z_i x_i'](\beta^* - \beta) = 0
\end{aligned}$$

This gives us our identification condition. If the rank $(\mathrm{E}[z_i x_i']) \geq \dim(\beta)$ then this will only equal 0 if $\beta = \beta^*$.

Okay, let's plug this into the GMM estimator

$$\hat{\beta}_{GMM} = \underset{\beta}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{i=1}^{N} z_i(y_i - \beta' x_i) \right) \left(\frac{1}{N} \sum_{i=1}^{N} z_i z_i' \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} z_i(y_i - \beta' x_i) \right).$$

We can take the FOC to see (trust me)

$$0 = \left(\frac{1}{N} \sum_{i=1}^{N} z_i z_i' \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} z_i x_i' \right) \left(\frac{1}{N} \sum_{i=1}^{N} z_i(y_i - \hat{\beta}_{GMM}' x_i) \right).$$

Solving for $\hat{\beta}_{GMM}$

$$\hat{\beta}_{GMM} = \left[\left(\frac{1}{N} \sum_{i=1}^{N} z_i x_i' \right) \left(\frac{1}{N} \sum_{i=1}^{N} z_i z_i' \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} z_i x_i' \right) \right]^{-1} \left[\left(\frac{1}{N} \sum_{i=1}^{N} z_i x_i' \right) \left(\frac{1}{N} \sum_{i=1}^{N} z_i z_i' \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} z_i y_i \right) \right].$$

This is the 2SLS estimator! Why have we done this? We can get efficiency gains over 2SLS by choosing different values of W. If can be shown that the most efficient estimates can be gained by

$$W = \mathrm{Var}(g(\cdot))^{-1}.$$

In this case (assuming homoskedasticity) that becomes

$$\begin{aligned}
\operatorname{Var}(g(x_i, y_i, z_i; \beta)) &= \operatorname{Var}(z_i y_i - z_i x_i' \beta) \\
&= \operatorname{Var}(z_i x_i' \beta + z_i \varepsilon_i - z_i x_i' \beta) \\
&= \operatorname{Var}(z_i \varepsilon_i) \\
&= \sigma_\varepsilon^2 \operatorname{E}(z_i z_i') \\
&\approx \frac{1}{N} \hat{\sigma}_\varepsilon^2 Z' Z \\
W &= \left[\frac{1}{N} \hat{\sigma}_\varepsilon^2 Z' Z \right]^{-1}
\end{aligned}$$

If homoskedasticity doesn't suit our data, this becomes the robust version

$$\begin{aligned}
W &= \operatorname{Var}(z_i \varepsilon_i)^{-1} \\
\hat{W} &= \left[\frac{1}{N} \sum_{i=1}^{N} \hat{\varepsilon}_i^2 z_i z_i' \right]^{-1}.
\end{aligned}$$

You can use the 2SLS estimates for $\hat{\varepsilon}_i$ because they are consistent (if inefficient).

```r
library(gmm)
model.gmm <- gmm(inc_2p_share~
                   log_inc_spend_capita + log_ch_spend_capita
                 +factor(ch_qual)+inc_tenure+st_uemp,
                 ~ ch_wealthy+ st_pop +ll_total_spending_captia
                 +factor(ch_qual)+inc_tenure+st_uemp,
                 vcov="HAC",  data=senate.data,
                 model=T, X=T, Y=T)
```

```
## Warning in getDat(object$g, object$x, data = object$data): There are missing
## values. Associated observations have been removed
```

```r
summary(model.gmm)
```

```
##
## Call:
## gmm(g = inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita +
##     factor(ch_qual) + inc_tenure + st_uemp, x = ~ch_wealthy +
##     st_pop + ll_total_spending_captia + factor(ch_qual) + inc_tenure +
##     st_uemp, vcov = "HAC", model = T, X = T, Y = T, data = senate.data)
```

```
##
##
## Method:  twoStep
##
## Kernel:  Quadratic Spectral(with bw =  0.9253 )
##
## Coefficients:
##                       Estimate      Std. Error    t value       Pr(>|t|)
## (Intercept)             5.9677e-01    2.6030e-02    2.2926e+01   2.5348e-116
## log_inc_spend_capita    2.0882e-01    3.6319e-02    5.7496e+00    8.9469e-09
## log_ch_spend_capita    -3.1810e-01    4.2720e-02   -7.4462e+00    9.6040e-14
## factor(ch_qual)1       -2.1241e-02    1.2899e-02   -1.6468e+00    9.9603e-02
## factor(ch_qual)2       -1.3847e-02    1.9445e-02   -7.1212e-01    4.7639e-01
## factor(ch_qual)3       -1.0653e-02    1.9635e-02   -5.4255e-01    5.8744e-01
## factor(ch_qual)4       -1.8675e-02    1.8097e-02   -1.0319e+00    3.0210e-01
## inc_tenure              1.0945e-03    8.8314e-04    1.2394e+00    2.1521e-01
## st_uemp                -3.8998e-03    2.6353e-03   -1.4799e+00    1.3891e-01
##
## J-Test: degrees of freedom is 1
##                 J-test   P-value
## Test E(g)=0:    1.00808  0.31536
##
## Initial values of the coefficients
##          (Intercept) log_inc_spend_capita  log_ch_spend_capita
##          0.605908450          0.195767593         -0.307622338
##     factor(ch_qual)1     factor(ch_qual)2     factor(ch_qual)3
##         -0.018975009         -0.013465599         -0.009012941
##     factor(ch_qual)4           inc_tenure              st_uemp
##         -0.019866276          0.001147343         -0.004735887
```

```
summary(model.2sls.over, vcov=vcovHC)
```

```
##
## Call:
## ivreg(formula = inc_2p_share ~ log_inc_spend_capita + log_ch_spend_capita +
##     factor(ch_qual) + inc_tenure + st_uemp | ch_wealthy + st_pop +
##     ll_total_spending_captia + factor(ch_qual) + inc_tenure +
##     st_uemp, data = senate.data, y = T, x = T)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21473 -0.05286 -0.00016  0.04456  0.26800
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           0.6059085  0.0285627  21.213  < 2e-16 ***
## log_inc_spend_capita  0.1957676  0.0365430   5.357 2.44e-07 ***
## log_ch_spend_capita  -0.3076223  0.0462674  -6.649 3.08e-10 ***
## factor(ch_qual)1     -0.0189750  0.0160959  -1.179   0.2399    
## factor(ch_qual)2     -0.0134656  0.0212452  -0.634   0.5270    
## factor(ch_qual)3     -0.0090129  0.0231686  -0.389   0.6977    
## factor(ch_qual)4     -0.0198663  0.0197072  -1.008   0.3147    
## inc_tenure            0.0011473  0.0007464   1.537   0.1259    
## st_uemp              -0.0047359  0.0028343  -1.671   0.0964 .  
##
## Diagnostic tests:
##                                           df1 df2 statistic  p-value    
## Weak instruments (log_inc_spend_capita)    3 188    31.730  < 2e-16 ***
## Weak instruments (log_ch_spend_capita)     3 188    25.973 4.21e-14 ***
## Wu-Hausman                                 2 187    18.966 3.17e-08 ***
## Sargan                                     1  NA     0.708      0.4    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08288 on 189 degrees of freedom
## Multiple R-Squared:  0.29,   Adjusted R-squared: 0.2599
## Wald test: 14.67 on 8 and 189 DF,  p-value: < 2.2e-16
```

```
signif(liml.out$coef.table,2)
```

```
##                       liml.est se.liml z.tests   p.val
## (Intercept)             0.6000 0.02900   21.00 5.3e-95
## log_inc_spend_capita    0.2000 0.04400    4.50 5.4e-06
## log_ch_spend_capita    -0.3100 0.06100   -5.10 4.0e-07
## factor(ch_qual)1      -0.0190 0.01600   -1.20 2.4e-01
## factor(ch_qual)2      -0.0130 0.02200   -0.59 5.6e-01
## factor(ch_qual)3      -0.0081 0.02700   -0.30 7.6e-01
## factor(ch_qual)4      -0.0190 0.02300   -0.84 4.0e-01
## inc_tenure             0.0011 0.00073    1.60 1.2e-01
## st_uemp               -0.0047 0.00290   -1.70 9.8e-02
```

7 Time series (the quick and dirty)

Here we will very briefly consider OLS with dependent observations in the hope that it is helpful to your future. With time series data we will rewrite our assumptions
Assumption D1 *The data generating process is* $y_t = \beta' x_t + \varepsilon_t$.

Note that here we may find that individual covariates x_t are possibly correlated over time and we may want to include lagged values of covariates x_{t-1} or the dependent variable y_{t-1}. As such the independence of observations (x_t, ε_t) will almost never hold (Assumption B2 is out of play). This is problematic for us as we used B2 to help us get to unbiasedness, consistency, and asymptotic normality, so now what? We will start with the easiest situation.
Assumption D2 *(Strict exogeneity)* $\mathrm{E}[\varepsilon_t | X] = 0$

Strict exogeneity says that ε_t is independent of all past, present, and future values of X. This assumption states that even if neither X nor ε_t are iid, ε_t is at least independent of all the variables in X at all times. With this we get our first property
Property D1 *OLS is unbiased if D1-D2 hold.*

In this case, the *only* concern is that the errors are "naturally" correlated over time (i.e., there is no omitted variable causing the correlation) then OLS is unbiased but inefficient under strict exogeneity. There are FGLS estimators that can be used to model the autocorrelation in ε_t (e.g., Prais-Winsten), but I don't think it's the best use of our time. Instead, let's focus on what we can do with OLS.

First, how do we know if we have autocorrelated errors? Like with heteroskedasticity, we will use the sample residuals from OLS and look for correlation. Recall that correlation is covariance divided by the product of the standard deviations.

$$\hat{\varepsilon}_t = y_t - \hat{\beta}' x_t$$
$$\hat{\rho}_\varepsilon(\ell) = \frac{\frac{1}{T-1}\sum_{t=1+\ell}^{T} \hat{\varepsilon}_t \hat{\varepsilon}_{t-\ell}}{\frac{1}{T-1}\sum_{t=1}^{T} \hat{\varepsilon}_t^2}$$

The function $\hat{\rho}_\varepsilon(\ell)$ returns the estimated correlation between ε_t and it's ℓth lag. The `acf` function in R does this for us and shows the correlation between $\hat{\varepsilon}_t$ and its first p lags. For a more formal test, the Breusch-Godfrey test is the go to test for this. The procedure will look familiar

1. Regress y_t on x_t using OLS. Save the residuals $\hat{\varepsilon}_i$.

2. Fit an AR(p) model (p lags of $\hat{\varepsilon}_i$) by fitting the model

$$\hat{\varepsilon}_t = \alpha' x_t + \sum_{\ell=1}^{p} \rho_\ell \hat{\varepsilon}_{t-\ell} + u_t,$$

   using OLS.
3. Conduct a test on $\rho_1 = \rho_2 = \ldots = \rho_p = 0$. Like the White/BP test, this test statistic is $TR^2 \sim \chi^2_p$ (asymptotically) under the null. Usually, we just check the 1st lag $p = 1$. Using both the ACF plot and Breusch-Godfrey you can a good sense of the autocorrelation in the residuals.

However, we often think that autocorrelation is rarely just a "feature" of most time series data, and assumption D2 will fail in *many* interesting cases. Instead, it is often thought that residual correlation reflects an important omitted variable and should be reduced as much as possible by using lags of y_t and x_t. Strict exogeneity fails if the DGP includes previous values of y_t within the independent variables x_t. To see this note that

$$\begin{aligned} y_t &= \beta' z_t + \gamma y_{t-1} + \varepsilon_t \\ &= \beta' z_t + \underbrace{(\beta' z_{t-1} + \varepsilon_{t-1})}_{y_{t-1}} \gamma + \varepsilon_t \end{aligned}$$

Here ε_{t-1} is included in $x_t = (z_t, y_{t-1})$, so that clearly violates strict exogeneity: OLS is biased and inefficient.

Good news, we can still get consistency of OLS without D2. To do this we need to define two new concepts **stationary** and **ergodic**. Informally, we say a series is stationary if its distribution is the same over any subseries. In other words the true mean, variance, and relative covariances are constant no matter what slice you look at of the series.

A series is ergodic (again, informal definition) if for large enough j, y_t and y_{t-j} are nearly independent. One example of stationary and ergodic process is an AR(1):

$$y_t = \rho y_{t-1} + \varepsilon_t,$$

for $|\rho| < 1$, $\mathrm{E}[\varepsilon_t] = 0$. Here, for any t we will see that y_t and y_{t-1} are correlated, while y_t and y_{t-2} will be less correlated, and so on. This series will move around $\mathrm{E}[y_t]$, but that true mean is unchanged. If $|\rho| \geq 1$ then we have a non-stationary process (more on this in a minute).

We can now impose some new assumptions.
Assumption D3 *(x_t, ε_t) is a stationary and ergodic sequence*

Note that even with strict exogeneity OLS may be unbiased but inconsistent if ε_t is explosive (non-stationary). Assumptions D1-D3 plus technical conditions can get us consistency, however, we are interested in relaxing strict exogeneity with:
Assumption D4 *(weak exogeneity)* $\mathrm{E}[\varepsilon_t | x_t] = 0$

This assumption can only be true we have *either* autocorrelated errors *or* a lagged value of y, but not both. To see this consider the following

$$\begin{aligned} y_t &= \beta' z_t + \gamma y_{t-1} + \varepsilon_t \\ \varepsilon_t &= \rho \varepsilon_{t-1} + u_t, \end{aligned}$$

rearranging we get

$$y_t = \beta' z_t + (\beta' z_{t-1} + \varepsilon_{t-1})\gamma + (\rho \varepsilon_{t-1} + u_t).$$

If $\rho = 0$ or $\gamma = 0$ then weak exogeneity holds. Otherwise, y_{t-1} is correlated with the joint error term. Let's put a pin in this for now and assume that we have D4 with (potentially) autocorrelated errors.
Assumption D5 *Other technical conditions that allow us to apply the ergodic theorem (like a LLN for dependent data) and a version of CLT for dependent observations*

Now we have a result that
Property D2 *OLS is consistent if D1, D3-D5 hold.*

This means that OLS will be be biased, but consistent if we have weak exogeneity with stationary and erogodic data (e.g., a lagged dependent variable). We can also get to asymptotic normality.
Property D3 *OLS is asymptotically normal if D1, D3-D5 hold, such that*

$$\sqrt{T}(\hat{\beta} - \beta) \xrightarrow{d} N(0, C^{-1} B C^{-1}).$$

Here $C = \mathrm{E}[x_t x_t']$ and $B = \lim_{T \to \infty} \mathrm{Var}(T^{-1/2} \sum_{t=1}^{T} x_t \varepsilon_t)$. We already know how to estimate C from our discussion on robust standard errors, but B is a new challenge.

We can start by working on the variance term

$$\begin{aligned}\mathrm{Var}(T^{-1/2}\sum_{t=1}^{T} x_t\varepsilon_t) &= \frac{1}{T}\sum_{t=1}^{T}\mathrm{Var}(x_t\varepsilon_t) + \frac{1}{T}\sum_{t\neq s}\mathrm{Cov}(x_t\varepsilon_t, x_s\varepsilon_s) \\ &= \underbrace{T^{-1}\sum_{t=1}^{T}\mathrm{E}[\varepsilon_t^2 x_t x_t']}_{\text{variance}} + \underbrace{T^{-1}\sum_{\ell=1}^{T-1}\sum_{t=\ell+1}^{T}\Big(\mathrm{E}[\varepsilon_t\varepsilon_{t-\ell}x_t x_{t-\ell}'] + \mathrm{E}[\varepsilon_t\varepsilon_{t-\ell}x_{t-\ell}x_t']\Big)}_{\text{Covariance among observations}}\end{aligned}$$

The covariance term can be simplified using stationarity of x_t and ε_t let

$$\lambda(\ell) = \mathrm{E}[\varepsilon_t\varepsilon_{t-\ell}x_t x_{t-\ell}']$$

Now because any gaps of the same size have the same covariance, same gap between them:

$$\mathrm{Var}\left(T^{-1/2}\sum_{t=1}^{T} x_t\varepsilon_t\right) = \lambda(0) + \sum_{\ell=1}^{T-1}\frac{T-\ell}{T}\left(\lambda(\ell) + \lambda(\ell)'\right)$$

By ergodicity, we know that for some lag L, $\lambda(\ell) = 0$ for all $\ell > L$. So if we know that L we can estimate B as

$$\hat{B} = \hat{\lambda}(0) + \sum_{\ell=1}^{L}\frac{T-L}{T}\left(\hat{\lambda}(\ell) + \hat{\lambda}(\ell)'\right).$$

However, we never know L in practice. Ideally, we could use $L = T-1$, but then we have to estimate $\lambda(T-1)$, which is the covariance between observations $T-1$ periods apart. We would only have two observations to estimate that covariance with…not good. Instead we will pick L in away that L increases with T and we will weight the lags such that we don't let large gaps (high variance estimates) have a lot of influence on the process.

$$\hat{B} = \hat{\lambda}(0) + \sum_{\ell=1}^{T-1} w_\ell\left(\hat{\lambda}(\ell) + \hat{\lambda}(\ell)'\right).$$

Of course now we have to pick w_ℓ *and* L, Newey and West suggest $w_\ell = 1 - \frac{\ell}{L+1}$ for $s \leq L$ and 0 otherwise. The general rule-of-thumb has settled on $L \approx T^{1/4}$, but software usually picks a good one for you.

Newey-West standard errors (`sandwich::NeweyWest`) account for both heteroskedasticity and any autocorrelation in ε_t. OLS is inefficient but consistent (so long as the autocorrelation is not the result of omitted variables), but these are the equivalent of robust standard errors for time series data. It is worth noting at this point that OLS is biased and inconsistent with autocorrelation *and* lagged variables, so if you are looking at lagged models, you want to get the autocorrelation as low as humanly possible before settling on a model.

Note, that the above only works if your data are stationary. There are tests for this and modeling approaches for non-stationary data. The main test for stationary is the (augmented) Dickey-Fuller (ADF) test. We can start with

$$y_t = \rho y_{t-1} + \varepsilon_t.$$

What we want to know if $\rho = 1$, but if it does we have a problem, so OLS is not a good choice under the null of non-stationary data. We will subtract y_{t-1} from both sides to get

$$\Delta y_t = (\rho - 1) y_{t-1} + \varepsilon_t.$$

We can also expand into a much more general null model of interest:

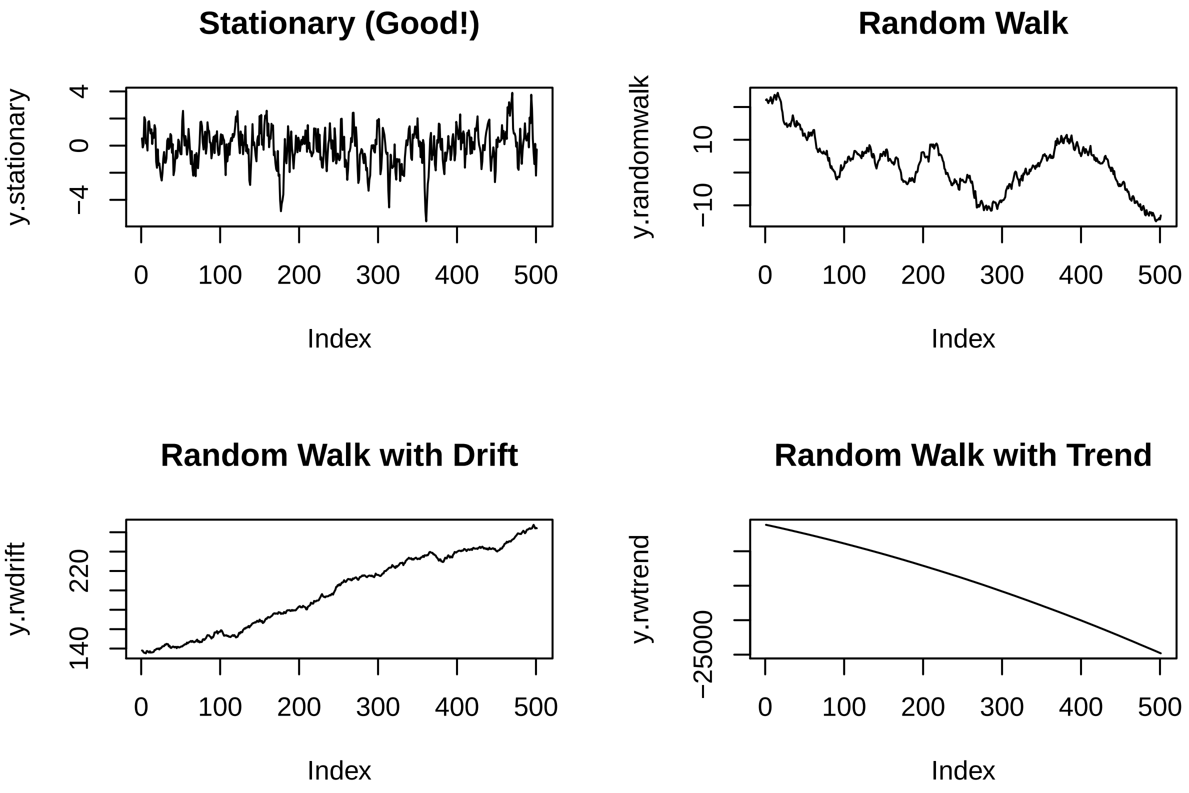
$$\Delta y_t = \beta_0 + \beta_1 t + \beta_2 y_{t-1} + \sum_{\ell=1}^{p+1} \delta_\ell \Delta y_{t-\ell} + \varepsilon_t.$$

1. Most importantly, we need to know if $\beta_2 = 0$ (unit root). This is a standard statistic

$$ADF = \frac{\hat{\beta}_2}{S.E.(\hat{\beta}_2)},$$

   but with a non-standard distribution. See the example code for more details. We want to reject the null hypothesis that our variable is non-stationary.
2. $\beta_0 = \beta_1 = 0$ but $\beta_2 \neq 0$ would tell us that y_t a random walk (a type of non-stationary data)
3. Just $\beta_0 = 0$ but $\beta_1 \neq 0$ and $\beta_2 \neq 0$ tell us that y_t a random walk with drift (also non-stationary data where the mean is moving with t)
4. Whether we choose to include β_0 and β_1 is a case-by-case choice. Rule of the thumb is to plot the series. See what it looks like

```
set.seed(123)
T <- 1001

y.stationary <-y.randomwalk <- y.rwdrift <- y.rwtrend <-  rep(0, T)
for(t in 2:T){
  y.stationary[t] <- .7*y.stationary[t-1] + rnorm(1)
  y.randomwalk[t] <- y.randomwalk[t-1] + rnorm(1)
  y.rwdrift[t] <- .25 + y.rwdrift[t-1] + rnorm(1)
  y.rwtrend[t] <- .25 - .05*t+ y.rwtrend[t-1] + rnorm(1)
}
y.stationary <- y.stationary[501:1001]
y.randomwalk <- y.randomwalk[501:1001]
y.rwdrift <- y.rwdrift[501:1001]
y.rwtrend <- y.rwtrend[501:1001]
par(mfrow=c(2,2))
plot(y.stationary, type="l", main="Stationary (Good!)")
plot(y.randomwalk, type="l", main="Random Walk")
plot(y.rwdrift, type="l", main="Random Walk with Drift")
plot(y.rwtrend, type="l", main="Random Walk with Trend")
```



Note, that we should test every variable (y and each non-constant/non-dummy variable in X) for stationarity. If we fail to reject the null, the first solution is to use the first difference $\Delta x_t = x_t - x_{t-1}$ in place of that variable (test that for stationarity, too).

Time series work flow:

1. Using your theory, build a model that may or may not include lags of covariates and the dependent variable

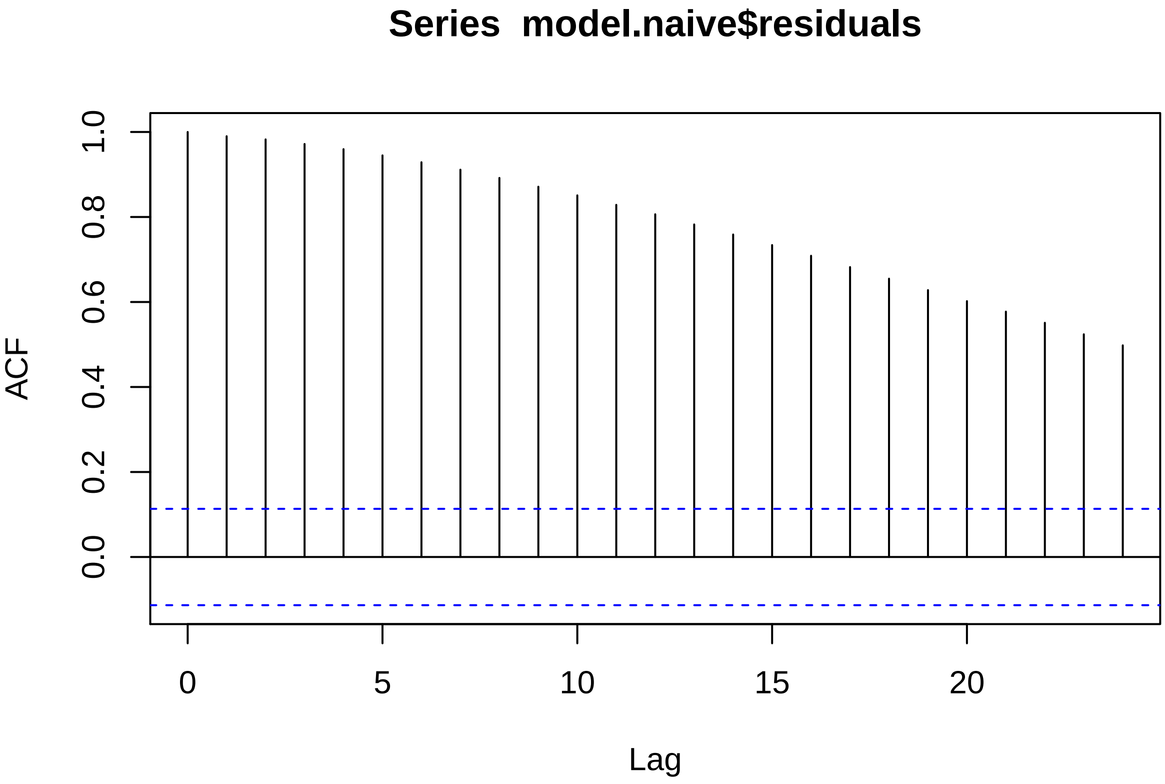
2. Test each variable for stationary. Difference variables (multiple times if necessary) until all variables are stationary. The `ur.df` function in the `urca` package, is a good implementation for the ADF.
3. Fit the model using OLS.
4. Test of autocorrelation in the residuals using a Breusch-Godfrey test and visuals like the autocorrelation function
5. If you detect autocorrelation, think carefully about whether you think more lags of the variables makes sense. Add in any additional lags (of x_t or y_t) you think make sense.
   - Usually you want to do this in some kind of balanced order. Start with y_{t-1} then x_{t-1} then y_{t-2} then x_{t-2} and so on. This is called an autoregressive distributed lag model, or ARDL(p, q) where p is the number of lags of y and q is the number of lags in x that you include.
6. Iterate steps 2 and 3 until either you eliminate autocorrelation or think you've added as many variables as makes sense.
7. Use Newey-West standard errors, especially if you have any remaining autocorrelation. You want autocorrelation as low possible to minimize the risk of bias and inconsistency, but it is not always possible to completely model away the autocorrelation.

7.1 Application

```
library(lmtest)
library(sandwich)
library(car)
library(urca) #tests for stationarity
library(dplyr)
rm(list=ls())
load("Rcode/datasets/outbidding.Rdata")

model.naive <- lm(states~Hattacks.count+Fattacks.count+lag.emp,
                  data=regData)
coeftest(model.naive)
```

```
##
## t test of coefficients:
##
##                   Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)    -1.444572   0.416659  -3.4670 0.0006045 ***
## Hattacks.count -0.099619   0.122582  -0.8127 0.4170597
## Fattacks.count -0.184537   0.446252  -0.4135 0.6795216
## lag.emp        -1.084718   0.106694 -10.1666 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
acf(model.naive$residuals)
```



```
bgtest(model.naive)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model.naive
## LM test = 295.67, df = 1, p-value < 2.2e-16
```

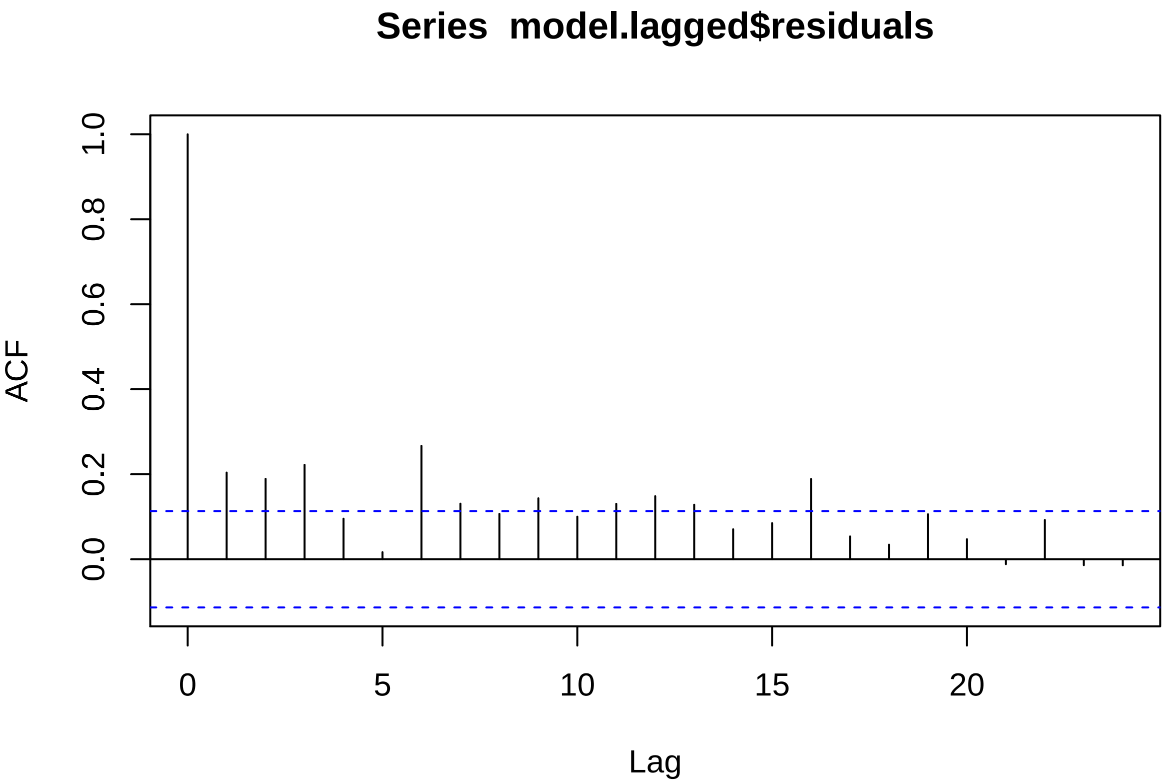
```
regData <- regData %>%
  arrange(date) %>%
  mutate(lag.states=lag(states))
head(regData[,c("date", "states", "lag.states")])
```

```
##         date   states lag.states
## 1: Jan 1994 2.893481         NA
## 2: Feb 1994 4.591593   2.893481
## 3: Mar 1994 6.215804   4.591593
## 4: Apr 1994 7.886076   6.215804
## 5: May 1994 9.618549   7.886076
## 6: Jun 1994 9.695237   9.618549
```

```
model.lagged <- lm(states~Hattacks.count+Fattacks.count+lag.emp+lag.states,
                   data=regData)
coeftest(model.lagged)
```

```
##
## t test of coefficients:
##
##                  Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)     0.0013148  0.0264057   0.0498  0.960323
## Hattacks.count -0.0497171  0.0076163  -6.5277 2.923e-10 ***
## Fattacks.count  0.3179509  0.0277787  11.4459 < 2.2e-16 ***
## lag.emp         0.0208208  0.0077439   2.6887  0.007583 **
## lag.states      1.0044683  0.0036396 275.9803 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

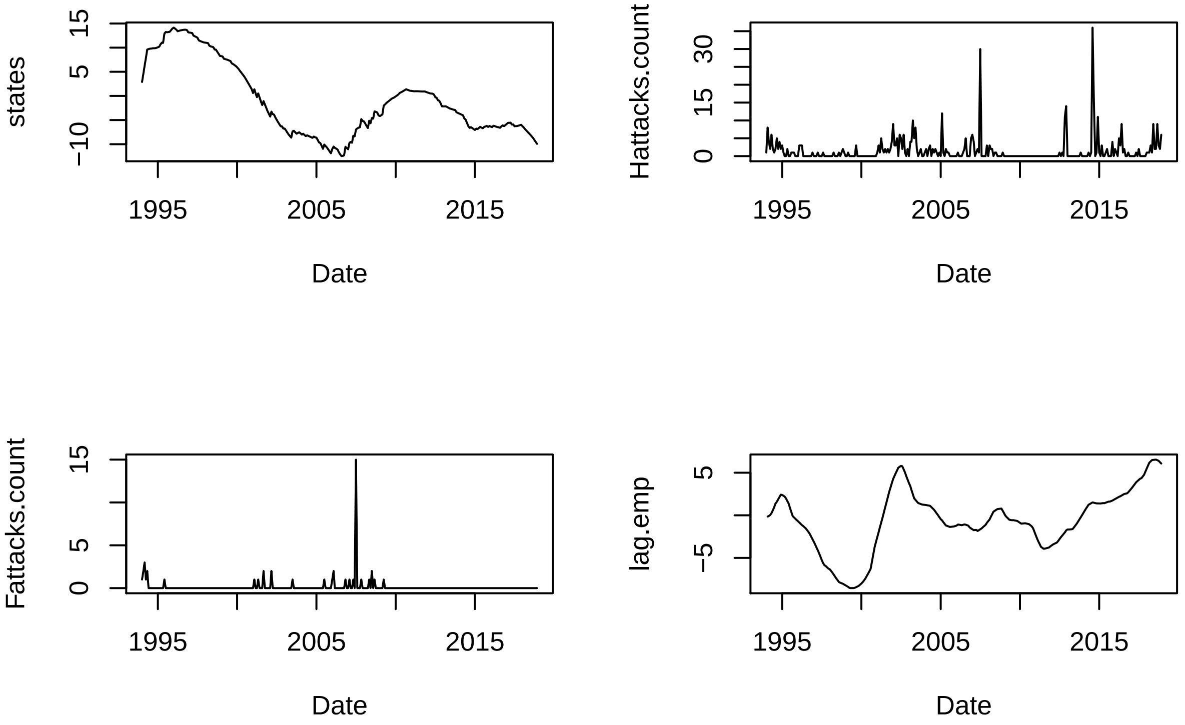
```
acf(model.lagged$residuals)
```



```
#BIG and persistent not good
bgtest(model.lagged)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model.lagged
## LM test = 12.559, df = 1, p-value = 0.0003942
```

```
#test for stationarity of y
### NOTE THAT THE t STAT IN THE REGRESSION
### IS THE ADF STAT; BUT THE p VALUE IS
### WRONG. USE THE CRITICAL VALUES AT THE BOTTOM
### WE WANT TO REJECT


par(mfrow=c(2,2))
plot(states~Date, data=regData, type="l")
plot(Hattacks.count~Date,  data=regData,type="l")
plot(Fattacks.count~Date, data=regData, type="l")
plot(lag.emp~Date, data=regData,type="l")
```



```
summary(ur.df(na.omit(regData$states),
              type="none", selectlags = "AIC")) #uh oh
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00329 -0.28790 -0.07590  0.07601  1.95070
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)    
## z.lag.1    -0.0006652  0.0035147  -0.189     0.85    
## z.diff.lag  0.2277860  0.0555417   4.101 5.32e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4588 on 296 degrees of freedom
## Multiple R-squared:  0.0538, Adjusted R-squared:  0.0474 
## F-statistic: 8.414 on 2 and 296 DF,  p-value: 0.0002791
## 
## 
## Value of test-statistic is: -0.1893 
## 
## Critical values for test statistics: 
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
regData <- regData %>%
  mutate(diff.states=states-lag.states)

summary(ur.df(na.omit(regData$diff.states),
              type="none", selectlags = "AIC")) #good
```

```
## 
## ############################################### 
## # Augmented Dickey-Fuller Test Unit Root Test # 
## ############################################### 
## 
## Test regression none 
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.91095 -0.27117 -0.07073  0.06551  1.84500 
```

```
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## z.lag.1    -0.61009    0.06819  -8.947  < 2e-16 ***
## z.diff.lag -0.25458    0.05438  -4.682 4.34e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4376 on 295 degrees of freedom
## Multiple R-squared:  0.4528, Adjusted R-squared:  0.4491 
## F-statistic:   122 on 2 and 295 DF,  p-value: < 2.2e-16
## 
## 
## Value of test-statistic is: -8.9474 
## 
## Critical values for test statistics: 
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```r
#Always good to check x too
summary(ur.df(na.omit(regData$Hattacks.count),
              type="trend",  selectlags = "AIC")) #good
```

```
## 
## ############################################### 
## # Augmented Dickey-Fuller Test Unit Root Test # 
## ############################################### 
## 
## Test regression trend 
## 
## 
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -8.401 -1.254 -0.939  0.093 34.426 
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.794512   0.407282   1.951    0.052 .
## z.lag.1     -0.764468   0.071594 -10.678   <2e-16 ***
## tt           0.002176   0.002301   0.946    0.345
## z.diff.lag   0.006405   0.058061   0.110    0.912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.411 on 294 degrees of freedom
## Multiple R-squared:  0.382,  Adjusted R-squared:  0.3757
## F-statistic: 60.57 on 3 and 294 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -10.6779 38.0412 57.0614
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
```

```
summary(ur.df(na.omit(regData$Fattacks.count),
              type="trend", selectlags = "AIC")) #good
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4070 -0.1633 -0.1271 -0.0840 14.7942
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1896422  0.1114236   1.702   0.0898 .
## z.lag.1     -0.8934342  0.0804906 -11.100   <2e-16 ***
## tt          -0.0004301  0.0006347  -0.678   0.4985
## z.diff.lag  -0.0858593  0.0577027  -1.488   0.1378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9394 on 294 degrees of freedom
## Multiple R-squared:  0.4953, Adjusted R-squared:  0.4901
## F-statistic: 96.16 on 3 and 294 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -11.0999 41.0936 61.6282
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
```

```
summary(ur.df(na.omit(regData$lag.emp),
              type="none", selectlags = "AIC")) #good
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
```

```
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28774 -0.03004 -0.00272  0.02524  0.51413
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.004421   0.001278   -3.46 0.000619 ***
## z.diff.lag  0.951303   0.019000   50.07  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07953 on 295 degrees of freedom
## Multiple R-squared:  0.8949, Adjusted R-squared:  0.8941
## F-statistic:  1255 on 2 and 295 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -3.4602
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
summary(ur.df(na.omit(regData$lag.emp),
              type="drift", selectlags = "AIC")) #good
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
```

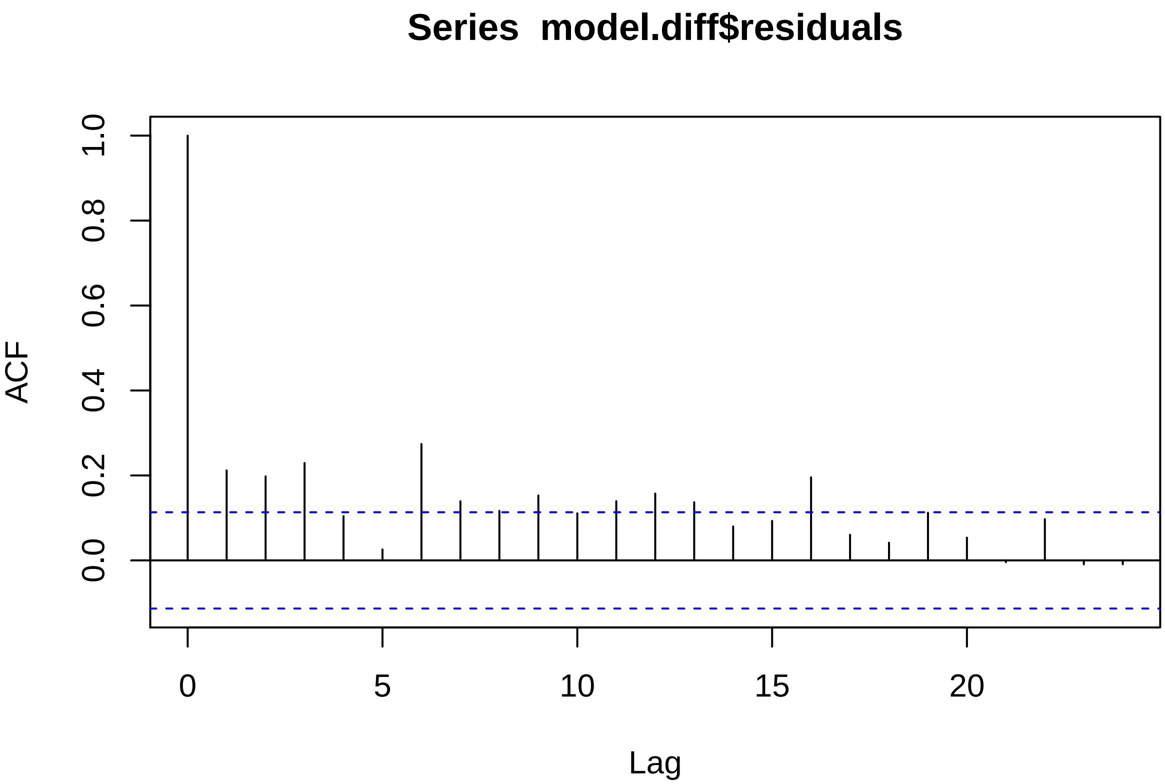
```
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28469 -0.02795 -0.00023  0.02884  0.51555
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.002644   0.004721   -0.56 0.575892
## z.lag.1     -0.004556   0.001302   -3.50 0.000537 ***
## z.diff.lag   0.952471   0.019136   49.77  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07962 on 294 degrees of freedom
## Multiple R-squared:  0.8942, Adjusted R-squared:  0.8935
## F-statistic:  1242 on 2 and 294 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -3.5 6.1294
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
```

```
#with differences
par(mfrow=c(1,1))

# no lags
model.diff <- lm(diff.states~Hattacks.count+Fattacks.count+lag.emp,
                 data=regData)
coeftest(model.diff, vcov=NeweyWest)
```

```
##
## t test of coefficients:
##
```

```
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.0051171  0.0375650 -0.1362 0.891739
## Hattacks.count -0.0499390  0.0167495 -2.9815 0.003107 **
## Fattacks.count  0.3157157  0.1048534  3.0110 0.002829 **
## lag.emp         0.0159029  0.0069924  2.2743 0.023666 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
acf(model.diff$residuals) #fairly small
```



```
bgtest(model.diff)
```

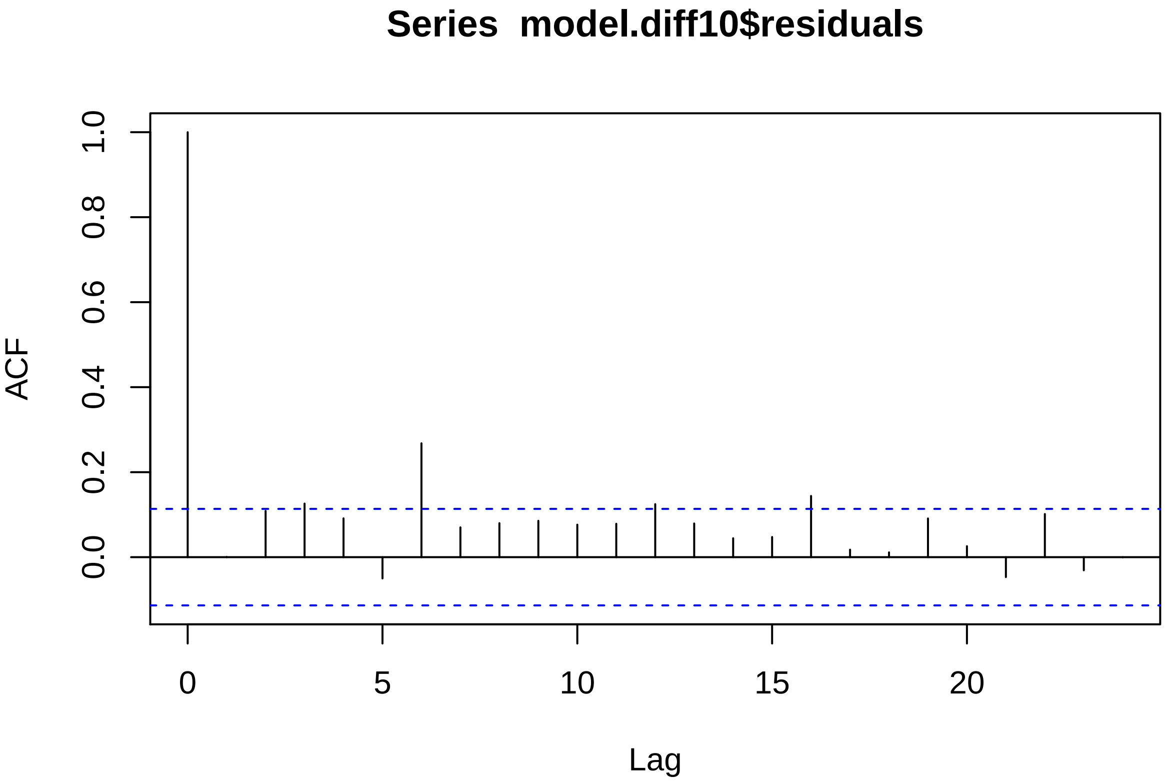
```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model.diff
## LM test = 13.462, df = 1, p-value = 0.0002434
```

```
# Lag y ARDL(1,0)
regData <- regData %>%
  mutate(L.diff.states=lag(diff.states))
```

```r
model.diff10 <- lm(diff.states~Hattacks.count+Fattacks.count+lag.emp+L.diff.states,
                   data=regData)
coeftest(model.diff10, vcov=NeweyWest)
```

```
##
## t test of coefficients:
##
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.0049230  0.0308458 -0.1596 0.873307
## Hattacks.count -0.0479644  0.0179848 -2.6669 0.008080 **
## Fattacks.count  0.2995364  0.0935361  3.2024 0.001513 **
## lag.emp         0.0144000  0.0055456  2.5967 0.009888 **
## L.diff.states   0.1604718  0.0907968  1.7674 0.078207 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
acf(model.diff10$residuals)
```



```r
bgtest(model.diff10)
```

```
##
```

```
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model.diff10
## LM test = 0.00020079, df = 1, p-value = 0.9887
```

```
# Autocorrelation is mostly gone.
# Do we have empirical or theoretical reason to add
# any additional lags? Not really.
# Let Newey-West errors take it from here
# We could also use model.diff with
# Newey-West and save an observation
# (no real differences in estimates, but good to know)

linearHypothesis(model.diff10,
                 c("Hattacks.count+Fattacks.count=0"),
                 vcov=NeweyWest)
```

```
## Linear hypothesis test
##
## Hypothesis:
## Hattacks.count  + Fattacks.count = 0
##
## Model 1: restricted model
## Model 2: diff.states ~ Hattacks.count + Fattacks.count + lag.emp + L.diff.states
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F   Pr(>F)
## 1    294
## 2    293  1 7.6843 0.005927 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model.diff,
                 c("Hattacks.count+Fattacks.count=0"),
                 vcov=NeweyWest)
```

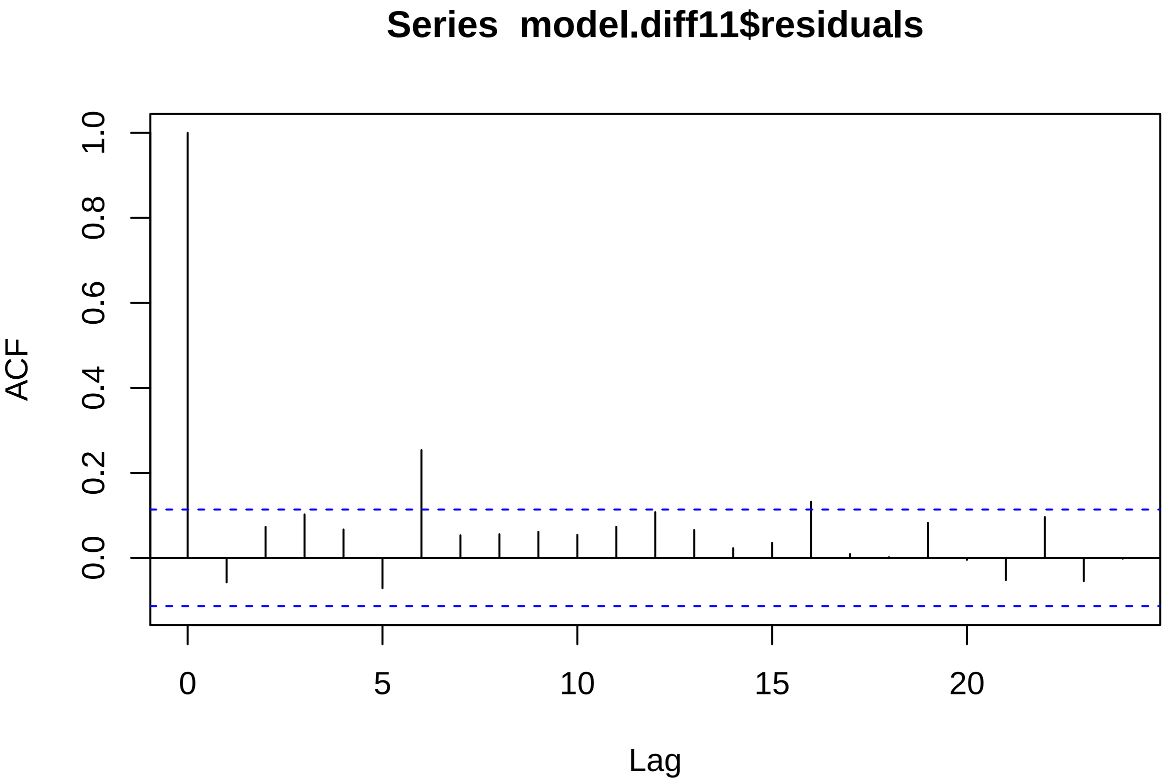
```
## Linear hypothesis test
```

```
##
## Hypothesis:
## Hattacks.count  + Fattacks.count = 0
##
## Model 1: restricted model
## Model 2: diff.states ~ Hattacks.count + Fattacks.count + lag.emp
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F  Pr(>F)
## 1    296
## 2    295  1 6.6376 0.01047 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### NOTE: If you wanted to keep going the next step would be lags of x ARDL(1,1)

regData <- regData %>%
  mutate(L.Hattacks.count=lag(Hattacks.count),
         L.Fattacks.count=lag(Fattacks.count),
         L.lag.emp=lag(lag.emp))
model.diff11 <- lm(diff.states~Hattacks.count+Fattacks.count+lag.emp
                   +L.Hattacks.count+L.Fattacks.count+L.lag.emp
                   +L.diff.states,
                   data=regData)
coeftest(model.diff11, vcov=NeweyWest)
```

```
##
## t test of coefficients:
##
##                    Estimate Std. Error t value  Pr(>|t|)
## (Intercept)      -0.0125460  0.0313382 -0.4003 0.6891991
## Hattacks.count   -0.0520474  0.0195343 -2.6644 0.0081442 **
## Fattacks.count    0.3099712  0.0917913  3.3769 0.0008332 ***
## lag.emp          -0.2513412  0.0961282 -2.6146 0.0093988 **
## L.Hattacks.count  0.0163770  0.0082324  1.9893 0.0476028 *
## L.Fattacks.count -0.0383971  0.0203329 -1.8884 0.0599671 .
## L.lag.emp         0.2653627  0.0978218  2.7127 0.0070723 **
## L.diff.states     0.1856970  0.0843759  2.2008 0.0285361 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
acf(model.diff11$residuals)
```



```
bgtest(model.diff11) #This is a weird result and a reminder that checking the
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model.diff11
## LM test = 10.982, df = 1, p-value = 0.0009199
```

```
# ACF is very much a good idea. The first order autocorrelation is very small
# and not worth worrying about
linearHypothesis(model.diff11,
                 c("Hattacks.count+Fattacks.count=0"),
                 vcov=NeweyWest)
```

```
## Linear hypothesis test
```

```
##
## Hypothesis:
## Hattacks.count  + Fattacks.count = 0
##
## Model 1: restricted model
## Model 2: diff.states ~ Hattacks.count + Fattacks.count + lag.emp + L.Hattacks.count +
##     L.Fattacks.count + L.lag.emp + L.diff.states
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F   Pr(>F)
## 1    291
## 2    290  1 8.5269 0.003774 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.2 Bonus topic: FGLS for AR(1) errors

Suppose we have Assumptions D1-D2, then OLS is unbiased but inefficient. As always we can claw back some efficiency with an FGLS procedure. The model of interest is

$$\begin{aligned} y_t &= \beta' x_t + \varepsilon_t \\ \varepsilon_t &= \rho \varepsilon_{t-1} + u_t \\ \mathrm{Var}(u_t|X) &= \sigma_u^2 \\ \mathrm{E}[u_t|X] &= 0 \\ \mathrm{Cov}(u_t, u_{t'}|X) &= 0 \ \ \forall t \neq t'. \end{aligned}$$

We can now verify that $\mathrm{E}[\varepsilon_t|X] = 0$ in this context as

$$\mathrm{E}[\varepsilon_t|X] = \mathrm{E}[\rho\varepsilon_{t-1}+u_t|X] = \mathrm{E}[\rho(\rho\varepsilon_{t-2}+u_{t-1})+u_t|X] = \mathrm{E}[u_t|X]+\rho\,\mathrm{E}[u_{t-1}|X]+\rho^2\,\mathrm{E}[u_{t-2}|X]\ldots = 0.$$

Let's introduce D3 to get $|\rho| < 1$.

$$\mathrm{E}[\varepsilon_t^2|X] = \sigma_u^2 + \rho^2\sigma_u^2 + \rho^4\sigma_u^2 + \ldots$$

Without stationarity, this sum explodes, but with stationarity we get a geometric sum

$$\mathrm{E}[\varepsilon_t^2|X] = \frac{\sigma_u^2}{1-\rho}$$

and covariance

$$\mathrm{E}[\varepsilon_t\varepsilon_{t-1}|X] = \mathrm{E}[\rho\varepsilon_{t-1}^2 + u_t\varepsilon_{t-1}|X] = \rho\,\mathrm{E}[\varepsilon_{t-1}^2] = \frac{\rho\sigma_u^2}{1-\rho}.$$

We can now specify the full covariance matrix of ε

$$s\Omega = \frac{\sigma_u^2}{1-\rho}\begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{T-1} \\ \rho & 1 & \rho & \dots & \rho^{T-2} \\ \rho^2 & \rho & 1 & \dots & \rho^{T-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{T-1} & \rho^{T-2} & \rho^{T-3} & \dots & 1 \end{bmatrix}$$

where $s = \sigma_u^2/(1-\rho)$ The FGLS approach (as always) is based on modeling this matrix (estimating ρ) then weighting the data to get back to "spherical" errors homoskedasticity and no serial correlation.

1. *OLS Step* OLS is unbiased and consistent, so regress y_t on x_t and estimate $\hat{\rho}$ using the residuals. (sample correlation between ε_t and ε_{t-1})
2. *Transformation* Transform the data

$$\tilde{y} = \begin{bmatrix} \sqrt{1-\hat{\rho}^2}y_1 \\ y_2 - \hat{\rho}y_1 \\ y_3 - \hat{\rho}y_2 \\ \vdots \\ y_T - \hat{\rho}y_{T-1} \end{bmatrix}, \quad \tilde{X} = \begin{bmatrix} \sqrt{1-\hat{\rho}^2}x_1' \\ x_2' - \hat{\rho}x_1' \\ x_3' - \hat{\rho}x_2' \\ \vdots \\ x_T' - \hat{\rho}x_{T-1}' \end{bmatrix}$$

3. *FGLS step* Fit the model using OLS on the transformed data. The errors in this model are now

$$\varepsilon_t - \rho\varepsilon_{t-1} = u_t.$$

   As such, you can estimate σ_u^2 using the residual variance from the FGLS step. There is an R package `prais` to do these for you.

If $u_t \sim N(0, \sigma_u^2)$ then this model can be estimated in a single step using maximum likelihood with log-likelihood

$$\begin{aligned} L(\beta, \sigma_u^2, \rho | y, X) = &-\frac{T}{2}\log(2\pi) - \frac{T}{2}\log(\sigma_u^2) + \frac{1}{2}\log(1-\rho^2) \\ &- \frac{1}{\sigma_u^2}\left((1-\rho^2)(y_1 - \beta' x_t)^2 + \sum_{t=2}^{T}(y_t - \beta' x_t + \rho(y_{t-1} - \beta' x_{t-1}))^2\right) \end{aligned}$$

7.3 Bonus topic: Consistent and efficient estimation with lagged variables

Note that there is a combined FGLS and IV approach to working around the problem of inconsistency with lagged variables and autocorrelation. Let's return to the model

$$\begin{aligned} y_t &= \beta' x_t + \gamma y_{t-1} + \varepsilon_t \\ \varepsilon_t &= \rho \varepsilon_{t-1} + u_t \end{aligned}$$

where u_t is **white noise** (stationary and non-autocorrelated). As we know, this will be inconsistent if we just fit this model naively with OLS, because

$$\sum_{t=1}^{T}[x_t \; y_{t-1}]\varepsilon_t \xrightarrow{p} \mathrm{E}\left[[x_t \; y_{t-1}]\varepsilon_t\right] \neq 0,$$

due to omitted variables hiding in ε_t (i.e., even weak exogneity is violated). We could follow the ARDL steps to remove the autocorrelation to obtain consistent estimates, but we can gain some efficiency by using the following FGLS:

1. Regress y_t using the residuals of a regression of y_t on x_t and x_{t-1}. Save the residuals. By construction these true errors from this regression contain y_{t-1} and x_{t-1}, but we remove any correlation with the residuals and x_{t-1}. As such these will be related to y_{t-1} (relevant), and they will only related to y_t through this relationship (valid).
2. Estimate $\hat{\beta}$ and $\hat{\gamma}$ using 2SLS with the residuals from 1 as an instrument for y_{t-1}.
3. Use the 2SLS residuals to form an estimate of ρ.

$$\hat{\rho} = \frac{\sum_{t=3}^{T} \hat{\varepsilon}_{t,\text{2sls}} \hat{\varepsilon}_{t-1,\text{2sls}}}{\sum_{t=3}^{T} \hat{\varepsilon}_{t,\text{2sls}}^2}.$$

4. Generate data

$$\begin{aligned} y_t^* &= y_t - \hat{\rho} y_{t-1} \\ x_t^* &= x_t - \hat{\rho} x_{t-1} \\ y_{t-1}^* &= y_{t-1} - \hat{\rho} y_{t-2} \\ \hat{\varepsilon}_{t-1} &= y_{t-1} - \hat{\beta}'_{\text{2sls}} x_{t-1} - \hat{\gamma}_{\text{2sls}} y_{t-2}, \end{aligned}$$

   and regress y_t^* on x_t^*, y_{t-1}^* and $\hat{\varepsilon}_t$. Let $\hat{\delta}$ be the estimated coefficient on $\hat{\varepsilon}_t$ in this regression than the FGLS estimate of ρ is

$$\tilde{\rho} = \hat{\rho} + \hat{\delta}.$$

   The other estimates can be used straight from the regression with either classic or Newey-West standard errors.

7.4 Bonus topic : Effects on current and future values of y

Consider the ARDL(p, q) model:

$$y_t = \alpha_0 + \sum_{s=1}^{p} y_{t-s} \rho_s + \sum_{\ell=0}^{q} \left(x_{t-\ell} \beta_{1+\ell} + \gamma'_{1+\ell} z_{t-\ell} \right) + \varepsilon_t,$$

where x_t is the variable of interest and z_t are additional controls. When there is a change in x_t, there is both a contemporaneous effect on y_t, but also a future effect on $y_{t+1}, y_{t+2}, \ldots$. We can identify all of these:

$$\begin{aligned} \frac{\partial y_t}{\partial x_t} &= \beta_1 \\ \frac{\partial y_{t+1}}{\partial x_t} &= \beta_2 + \rho_1 \left(\frac{\partial y_t}{\partial x_t} \right) = \rho_1 \beta_1 + \beta_2 \\ \frac{\partial y_{t+2}}{\partial x_t} &= \beta_3 + \rho_1 \left(\frac{\partial y_{t+1}}{\partial x_t} \right) + \rho_2 \left(\frac{\partial y_t}{\partial x_t} \right) = \rho_1^2 \beta_1 + \rho_1 \beta_2 + \rho_2 \beta_1 + \beta_3 \\ &\vdots \\ \frac{\partial y_{t+s}}{\partial x_t} &= \beta_{s+1} + \sum_{\ell=1}^{s} \rho_\ell \left(\frac{\partial y_{t+s-\ell}}{\partial x_t} \right) \end{aligned}$$

By stationarity the limit of this will be 0 as $s \to \infty$.

You can also consider the total effect of a "permanent" change in x_t rather than just a one-off. In this case we would just sum these effects and using the properties of infinite geometric sums we get:

$$\sum_{s=0}^{\infty} \frac{\partial y_{t+s}}{\partial x_t} = \frac{\sum_{\ell=1}^{q} \beta_\ell}{1 - \sum_{s=1}^{p} \rho_s}.$$

For any of these effects, the delta method will provide standard errors.

8 Panel data crash course

With panel data we will start with a new set up assumptions. First, as usual, we will start with the DGP.
Assumption E1 *The data generating process is linear-in-the-parameters, such that*

$$y_{it} = \alpha_i + \beta' x_{it} + \gamma' z_i + \varepsilon_{it}.$$

Usually we think of i as "units" (individuals, states, countries, dyads, etc) and t as "within-unit" observations (typically time, but could be multiple individuals within a unit, etc). We will let $i = 1, \dots, N$, $t = 1, \dots, T$ and NT be the total number of observations. To make exposition easier, we will assume a "balanced" panel where T is the same for each i, but nothing we do will depend on that. Note that neither x_{it} nor z_i contain a constant term right now, instead α_i reflects the constant(s). More on this to come. Additionally, x_{it} is a variable that changes both across and within units, while z_i just changes within units. Until further notice assume that $\alpha_i = \alpha$ for all i (**Assumption E1.A**)

Given this setup, we are unlikely to get something like Assumption B2 (independent observations) back. Afterall, if our panel is a collection of N separate time series, then iid is pretty much gone from the start, but we will impose a similar assumption
Assumption E2 *Each unit* $(x_i, z_i, \varepsilon_i)$ *is drawn iid.*

Here we are making the assumption that each block of observations is independent of the rest and drawn from some population process. We are not currently making any assumptions about the within-unit data. We will use the notation $x_i = (x_{i1}, \dots, x_{iT})$ to refer to all observations of x_{it} within unit i.

As always, We will need to make some kind of exogeneity assumption
Assumption E3 *There is strict exogeneity within units,* $\mathrm{E}[\varepsilon_{it} | x_i, z_i] = 0$.

Let $\mathbf{X} = [1\ X\ Z]$ and let $\theta = (\alpha, \beta, \gamma)'$, then the *pooled* OLS estimator for the panel model is

$$\begin{aligned}\hat{\theta}_{\text{pooled}} &= \left(\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it} \mathbf{x}_{it}' \right)^{-1} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it} y_{it} \\ &= (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}' y,\end{aligned}$$

or (by E1.A)

$$\hat{\theta}_{\text{pooled}} = \theta + \left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}.$$

By strict exogeneity within unit and independence across units we have $\mathrm{E}[\varepsilon_{it}|\mathbf{X}] = 0$ and thus we can apply iterated expectations to get

Property E1 *Under E1.A-E3 the pooled estimator $\hat{\theta}_{pooled}$ is unbiased*

Like with time series models, we may find ourselves in a world where strict exogeneity doesn't make sense. Note that as with time series data, moving away from strict exogeneity will limit us to mostly asymptotic results, unless we want to make a within-unit iid assumption (which we probably don't).

Assumption E4 *The data are weakly exogeneous,* $\mathrm{E}[\varepsilon_{it}|\mathbf{x}_{it}] = 0$

Further, we will impose a rank condition

Assumption E5 *The matrix* $\mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right]$ *has full rank.*

In this assumption we are assert that the DGP for the T observations we observe within each unit is well behaved and of full rank. Most of the panel data results we consider will be with respect to a fixed T.

Assumption E6 *Technical and moment assumptions that allow us to apply a type of LLN and a type of CLT*

Using our usual LLN-type arguments we can verify the consistency of the pooled estimator.

$$\begin{aligned}
\hat{\theta}_{\text{pooled}} &= \theta + \left(\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right)^{-1}\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it} \\
\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right) &\xrightarrow{p} \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\mathbf{x}_{it}'\right] = \mathbf{Q} \\
\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}\right) &\xrightarrow{p} \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_{it}\varepsilon_{it}\right] = 0 \\
\hat{\theta}_{\text{pooled}} &\xrightarrow{p} \theta
\end{aligned}$$

Property E2 *Under E1.A, E2, and E4-E6 the pooled estimator $\hat{\theta}_{pooled}$ likely exists for large enough N and the estimator is consistent for θ.*

The asymptotic normality follows in the usual way with

$$\begin{aligned}
\sqrt{N}(\hat{\theta}_{\text{pooled}} - \theta) &= \left(\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it}\mathbf{x}_{it}' \right)^{-1} \frac{\sqrt{N}}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it}\varepsilon_{it} \\
\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it}\mathbf{x}_{it}' &\xrightarrow{p} \mathrm{E}\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it}\mathbf{x}_{it}' \right] = \mathbf{Q} \\
\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it}\varepsilon_{it} &\xrightarrow{d} N(0, \Sigma_N) \\
\sqrt{N}(\hat{\theta}_{\text{pooled}} - \theta) &\xrightarrow{d} N(0, \mathbf{Q}^{-1}\Sigma_N\mathbf{Q}^{-1}) \\
\Sigma &= \mathrm{Var}\left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it}\varepsilon_{it} \right)
\end{aligned}$$

Property E3 *Under E1.A, E2, and E4-E6 the pooled estimator $\hat{\theta}_{pooled}$ is asymptotically normal such that*

$$\sqrt{N}(\hat{\theta}_{pooled} - \theta) \xrightarrow{d} N(0, \mathbf{Q}^{-1}\Sigma_N\mathbf{Q}^{-1})$$

This is a format we should be used to seeing by now, but notice that we don't have within-independence and Σ is not diagonal. We can estimate this full variance of θ as

$$\begin{aligned}
\mathrm{avar}\left(\hat{\theta}_{\text{pooled}}\right) &= \frac{1}{N}\mathbf{Q}^{-1}\Sigma_N\mathbf{Q}^{-1} \\
\widehat{\mathrm{avar}}\left(\hat{\theta}_{\text{pooled}}\right) &= \frac{1}{N}\hat{\mathbf{Q}}^{-1}\widehat{\Sigma}_N\hat{\mathbf{Q}}^{-1} \\
\hat{\mathbf{Q}} &= \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T} \mathbf{x}_{it}\mathbf{x}_{it}' \\
\widehat{\Sigma}_N &= \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it}\hat{\varepsilon}_{it} \right]\left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_{it}\hat{\varepsilon}_{it} \right]'.
\end{aligned}$$

This variance matrix is called the *cluster-robust* or *clustered* variance matrix. The square root of the diagonal provides *clustered standard errors*. Note that the clustered variance matrix allows for *arbitrary* correlation among the errors (we haven't imposed any structure on them, not even stationarity). This is a powerful result that makes the clustered matrix very popular. Furthermore, the meat of this sandwich is an average over units. As such, this estimator is only asymptotically valid **in** N. If you have less than 50 units, the clustered matrix is probably not reliable and you may be better off with basic robust standard errors (if NT is large enough), or another alternative. To see one such alternative, consider a fixed N and let's allow T to increase.

With large N and small T we treated each unit as it's own independent observation. With large T and small N, we can imagine that we have a small number of long time series we want to analyze.

Assumption E7 *Within unit (x_t, ε_t) are stationary and ergodic*

Assumption E8 *The matrix* $\mathrm{E}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{it}\mathbf{x}_{it}'\right]$ *has full rank.*

These new assumptions give us the same basic results we would expect

Property E4 *Under E1.A, E4, E6, E7-E8 the pooled estimator $\hat{\theta}_{pooled}$ likely exists for large enough T and the estimator is consistent in T.*

Property E5 *Under Assumptions E1.A, E4, E6, E7-E8 the pooled estimator $\hat{\theta}_{pooled}$ is asymptotically normal such that*

$$\sqrt{T}(\hat{\theta}_{pooled} - \theta) \xrightarrow{d} N(0, \mathbf{Q}^{-1}\Sigma_T\mathbf{Q}^{-1})$$

Where the main difference is that $\Sigma_T \neq \Sigma_N$ under these new assumptions. Instead, it follow from the Newey-West results we saw above. We won't get into that, because it's not often done in most panel work. If you have enough N to use clustered standard errors instead, you're probably better off there even if T is also big. Additional alternatives exist, but we'll set those aside for now. If you have homoskedasticity and no autocorrelation, than the ordinary standard errors are still correct (but how likely is that)?

However, what about situations where we have heterogeneity across units in our model? We will consider this in two parts

8.1 Random effects

The random effects model starts with Assumption E1, but we will make some adjustments. Let's consider another adjustment (**Assumption E1.R**)

$$\begin{aligned}
y_{it} &= \alpha_i + \alpha + \beta' x_{it} + \gamma' z_i + u_{it} \\
\alpha_i &\overset{iid}{\sim} f(\alpha) \\
\mathrm{E}[\alpha_i] &= 0 \\
\mathrm{Cov}(\alpha_i, \mathbf{x}_i) = \mathrm{Cov}(\alpha_i, u_i) &= 0 \\
\mathrm{Var}(\alpha_i) &= \sigma^2_\alpha \\
\mathrm{Var}(u_{it}|\mathbf{x}_{it}) &= \sigma^2_u
\end{aligned}$$

We will also go back to an iid world (**Assumption E2.R** more restrictive than Assumption E2) by letting $(x_{it}, \varepsilon_{it})$ be iid. Finally, we will adjust the weak exogeneity assumption (**Assumption E4.R**)

$$\mathrm{E}[u_{it}|\mathbf{x}_{it}, \alpha_i] = 0.$$

Notice that we've built some homoskedasticity into Assumption E1.R as well. To see the restrictiveness of this model over the basic model, consider that all the within-unit correlation here comes from the presence of α_i. In contrast, the basic model allowed for arbitrary within-unit correlation.

Because these are more restrictive assumptions that E2 and E4 all of the above results still apply. The pooled estimator $\hat{\theta}_{\text{pooled}}$ is unbiased, consistent, and asymptotically normal for the the random effects model. However, pooled OLS will be inefficient for this model. As such, we can consider an FGLS solution (sometimes called the random effects estimator).

Steps:

1. Fit the model using pooled OLS (consistent), save the residuals ($\hat{\varepsilon}_{it} = y_{it} - \hat{\theta}'\mathbf{x}_{it}$). Note that in the pooled residuals are estimates of $\hat{\varepsilon}_{it} = \widehat{\alpha_i + u_{it}}$. This relationship implies that
$$\mathrm{Var}(\varepsilon_{it}) = \mathrm{Var}(\alpha_i) + \mathrm{Var}(u_{it}) = \sigma_\alpha^2 + \sigma_u^2,$$
while we also know that
$$\mathrm{Cov}(\varepsilon_{it}, \varepsilon_{it'}) = \mathrm{Cov}(\alpha_i + u_{it}, \alpha_i + u_{it'}) = \sigma_\alpha^2$$

2. EITHER:
   - Fit the model using the within estimator (detailed below), save the residuals $\hat{u}_{it}$ and estimate
   $$\hat{\sigma}_u^2 = \frac{\hat{u}_{it}'\hat{u}_{it}}{N(T-1)-K}.$$
   - Use the pooled residuals to estimate
   $$\hat{\sigma}_\alpha^2 = \frac{1}{N(T-1)(T/2)} \sum_{i=1}^{N} \sum_{t=2}^{T} \sum_{t'=1}^{t-1} \hat{\varepsilon}_{it}\hat{\varepsilon}_{it'},$$

3. Use the relationship
$$\sigma_\varepsilon^2 = \sigma_\alpha^2 + \sigma_u^2$$
to back out the remaining quantity of interest.

4. Build a big $NT \times NT$ block diagonal matrix $\hat{\Omega}$

$$\hat{\Omega} = \begin{bmatrix} \hat{\Omega}_1 & 0 & \dots & 0 \\ 0 & \hat{\Omega}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\Omega}_N \end{bmatrix}$$

$$\hat{\Omega}_i = \begin{bmatrix} \hat{\sigma}^2_\alpha + \hat{\sigma}^2_u & \hat{\sigma}^2_\alpha & \dots & \hat{\sigma}^2_\alpha \\ \hat{\sigma}^2_\alpha & \hat{\sigma}^2_\alpha + \hat{\sigma}^2_u & \dots & \hat{\sigma}^2_\alpha \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\sigma}^2_\alpha & \dots & \hat{\sigma}^2_\alpha & \hat{\sigma}^2_\alpha + \hat{\sigma}^2_u \end{bmatrix}$$

More conveniently, we can build

$$\begin{aligned} \lambda &= 1 - \sqrt{\frac{\hat{\sigma}^2_u}{\hat{\sigma}^2_u + T\hat{\sigma}^2_\alpha}} \\ \tilde{\mathbf{x}}_{it} &= \mathbf{x}_{it} - \lambda \bar{\mathbf{x}}_i \\ \tilde{y}_{it} &= y_{it} - \lambda \bar{y}_i \\ \hat{\theta}_{\text{RE-FGLS}} &= \left(\tilde{\mathbf{X}}'\tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}'\tilde{y} \end{aligned}$$

Property E6 *Under Assumptions E1.R, E2.R, E4.R, and E5-E6, the random effects estimator will exist, be consistent for θ in N, be asymptotically normal, and be more efficient than pooled OLS.*

A lot of people in political science really like the random effects model and really want to lean on that efficiency gain. When $\alpha_i \sim N(\alpha, \sigma^2_\alpha)$ you can get asymptotic efficiency, but again at what cost? I don't get it. You can use the clustered variance matrix with random effects if you want to be a little more conservative, but if you do then you're also admitting that you got Ω wrong, so what was it all for?

Additional notes:

1. The random effects estimator is consistent in T, but it will not be efficient because you need large N to get a good estimate of σ^2_α.

8.2 The fixed effects model

We will now consider a more general model and then talk about how it relates to the RE model. The FE model starts with Assumption E1 with no tweaks, our DGP is

$$y_{it} = \alpha_i + \beta' x_{it} + \gamma' z_i + u_{it}.$$

Unlike with the RE model, here α_i are fixed parameters not a random variable (thus the names). We will also return to iid units (not observations) so E2 is fully back.

For the moment let's return to E3 and allow for strict exogeneity within units. Note that we have a very similar setup to the pooled model. The one tweak is that we have not said anything about the correlation between α_i and x_{it}. In the pooled model we assumed that any deviations from α could be safely let in ε, but that's no longer true. This suggests that the pooled model may be trouble. Let's break α_i into two parts $\alpha_i = \alpha + \delta_i$ so we can see what happens if we fit the fixed effects model using pooled OLS. Note that $\varepsilon_{it} = u_{it} + \delta_i$ We now find that

$$\mathrm{E}[\hat{\theta}_{\text{pooled}} | \mathbf{x}_{it}] = \theta + \left(\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it} \mathbf{x}_{it}' \right)^{-1} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{x}_{it} \delta_i,$$

which is our familiar omitted variable bias result. If there is any correlation between the variables in $\mathbf{x}_{it}$ and the unit-specific heterogeneity δ_i, then the pooled estimator (and by extension the random effects estimator) are biased and inconsistent.

We will now consider three methods for removing/accomidating this heterogeneity:

8.2.1 First differences (skip if we're short on time)

One thing we can do is take the regression in differences by subtracting y_{t-1} from both sides.

$$\begin{aligned} y_{it} - y_{t-1} &= (\alpha_i - \alpha_i) + \beta'(x_{it} - x_{it-1}) + \gamma'(z_i - z_i) + u_{it} - u_{it-1} \\ \Delta y_{it} &= \beta' \Delta x_{it} + \Delta u_{it}. \end{aligned}$$

All the time-invariant variables/heterogeneity are removed and OLS becomes a good estimator for β.

Property E7 *Under Assumptions E1-E3 the first differences estimator $\hat{\beta}_{FD}$ is unbiased for β.*

With appropriate rank and moment assumptions, we can also obtain

Property E8 *Under Assumptions E1, E2, E4, and some technical assumptions, $\hat{\beta}_{FD}$ is consistent in N and asymptotically normal with the clustered variance matrix*

The FD estimator is also consistent in T (with Assumptions E7-E8, but also E3) with the Newey-West style variance matrix. First differences are great and there are lots of reasons to like them, but a more common solution is…

8.2.2 Least squares dummy variable estimator OR within estimator

Another intuitive way to fit the fixed effects model is estimate the time invariant parameters directly for each unit. Let

$$\delta_i = \alpha_i + \gamma' z_i,$$

then the model becomes

$$y_{it} = \beta' x_{it} + \delta' d_{it} + u_{it},$$

which can be fit using OLS. Let $\theta_{\text{LSDV}} = (\beta, \delta)$ and redefine $\mathbf{X} = [X \; D]$ where D is a $NT \times N$ matrix of dummy variables where each column denotes if the observation is associated with unit i. Notice that we no longer have an overall constant, instead we have a constant for each unit δ_i. Also note that this constant controls for **all** time-invariant heterogeneity, even things we didn't think of or can't measure. In this way, the fixed effects model is a very important tool for fighting endogeneity as it eliminates any concerns about omitted variable bias from time-invariant sources.

Using the LSDV estimator to fit the fixed effects model is algebraically equivilent to fitting the following model using OLS

$$y_{it} - \bar{y}_i = \beta'(x_{it} - \bar{x}_i) + (u_{it} - \bar{u}_i),$$

this approach is called the **within estimator**. Note that $\hat{\beta}_{\text{within}}$ will always equal $\hat{\beta}_{LSDV}$, but it saves us from estimating the N δ parameters.

Property E9 *Under Assumptions E1-E3 the LSDV/within estimators for β and δ are unbiased*

With appropriate rank and moment assumptions, we can also obtain

Property E10 *Under Assumptions E1, E2, E4 and extra technical assumptions, the LSDV/within estimators for β are consistent in N and asymptotically normal with the clustered* *variance matrix*

Likewise, we can take asymptotic wrt to T (adding Assumptions E7-E8) and get consistency and asymptotic normality as $T \to \infty$ with panel Newey-West variance asymptotically.

While the LSDV and within estimators always produce the exact same estimates for β, they will only be identical to the FD estimates when $T = 2$. The main advantage of LSDV/within over the FD estimator is when we have large T and want to include lagged variables. Both FD and LSDV are biased in this case, but the LSDV is consistent in T with a lagged dependent variable (byvirtue of weak exogeneity). The FD estimator is inconsistent as it requires strict exogeneity. Likewise, with the within/LSDV you can get estimates of δ that are consistent in T, which you can't do with FD. There are other differences regarding when one is more efficient than the other, but that's not the best use of our time. For the most part, the LSDV/within approach is probably going to be your better go to. If we have time, we can talk about dynamic panel models, but I'm not optimistic; the thing to remember is that if you want a lagged DV in your fixed effects model, you need a big T.

8.3 Correlated random effects (CRE) and model testing

The next thing you should want to know is when do you want to use the pooled, or random effects FGLS, or the LSDV/within. Here is a handy chart to help with the conditions and then we'll discuss tests

Model	Pooled	RE-FGLS	FD	LSDV/within
Basic model: $y_{it} = \alpha + \beta' x_{it} + \gamma' z_i + \varepsilon_{it}$ $(\mathbf{x}_i, \varepsilon_i)$ iid	Unbiased and consistent for θ if $\text{E}[\varepsilon_{it}\|\mathbf{x}_i] = 0$. Biased, but consistent for θ if only $\text{E}[\varepsilon_{it}\|\mathbf{x}_{it}] = 0$. Most efficient if ε_{it} are homoskedastic, otherwise cluster/Newey-West depending.	Unbiased and consistent for θ if $\text{E}[\varepsilon_{it}\|\mathbf{x}_i] = 0$. Biased, but consistent for θ if only $\text{E}[\varepsilon_{it}\|\mathbf{x}_{it}] = 0$. no efficiency gains over pooled model. Covariance matrix is incorrect b/c of iid assumptions (clustering may help)	Unbiased for β if $\text{E}[\varepsilon_{it}\|x_i] = 0$. Consistent in N for β if $\text{E}[\varepsilon_{it}\|x_{it}] = 0$. Consistent in T for β if $\text{E}[\varepsilon_{it}\|x_i] = 0$ and $(x_{it}, \varepsilon_{it})$ are stationary and ergodic	Unbiased for β and δ_i if $\text{E}[\varepsilon_{it}\|x_i] = 0$. Consistent in N for β if $\text{E}[\varepsilon_{it}\|x_{it}] = 0$. Consistent in T for β and δ_i if $\text{E}[\varepsilon_{it}\|x_{it}] = 0$ and $(x_{it}, \varepsilon_{it})$ are stationary and ergodic
RE model: $y_{it} = \alpha_i + \alpha + \beta' x_{it} + \gamma' z_i + u_{it}$ $\text{E}[\alpha_i] = \text{Cov}(\mathbf{x}_{it}, \alpha_i) = \text{Cov}(u_{it}, \alpha_i) = 0$. $(\mathbf{x}_{it}, \varepsilon_{it})$ iid	see above	Unbiased and consistent for θ if $\text{E}[\varepsilon_{it}\|\mathbf{x}_i] = 0$. Biased, but consistent for θ if only $\text{E}[\varepsilon_{it}\|\mathbf{x}_{it}] = 0$ May be more efficient than pooled OLS as N increases	see above	see above
FE model: $y_{it} = \beta' x_{it} + \delta_i + u_{it}$ (x_i, ε_i) iid	Biased and inconsistent	Biased and inconsistent	see above	see above

Okay, so now you're thinking I don't want inconsistent estimates, but efficiency is nice. How do I choose among these estimator?

Interesting fact about FGLS-RE and the within/LSDV estimators. Even if the random effects assumptions are good, the RE-FGLS estimator converges to the within-estimator (below) as T increases. So the efficiency gains are fleeting as T increases while the risk of bias and inconsistency remain. One measure of how close the two estimators are is

$$\lambda = 1 - \frac{\sigma_u}{\sqrt{\sigma_u^2 + T\sigma_\alpha^2}}.$$

The closer λ is to 1 the more similar the estimates. If the random effects assumptions are not satisfied the estimator is inconsistent. This has led to the within-estimator being considered "mostly harmless."

However, there are also tests for choosing among the approaches. The traditional method is called a Hausman test, but it's been subject to a lot of scrutiny lately for being too reject happy. Instead we'll focus on a robust form of this test based on nested models.

It turns out that there is *another* estimator for the fixed effects model that is also equivalent to the LSDV/within. This is called the **correlated random effects** estimator. The model is given by

$$\begin{aligned}
y_{it} &= \alpha_i + \alpha + \beta' x_{it} + \tilde{\beta}' \bar{x}_i + u_{it} \\
\mathrm{E}[\alpha_i] &= 0 \\
\alpha_i &\overset{iid}{\sim} f \\
\mathbf{x}_{it} &= \begin{bmatrix} 1 & x_{it} & \bar{x}_i \end{bmatrix} \\
\mathrm{Cov}(\mathbf{x}_{it}, \alpha_i) &= \mathrm{Cov}(u_{it}, \alpha_i) = 0
\end{aligned}$$

What's happening here? Well we're blending the RE and FE models a bit. Or you can blend the pooled and the FE model by not including α_i, which may or may not be better. We are directly controlling for deviations from the within means (blending the within model), while deviations of y_{it} from $\bar{y}_i$ and u_{it} from u_i are captured in α_i and α. The estimates $\hat{\beta}_{\text{CRE}}$ will match those from the LSDV/within exactly. We can use this to test the hypothesis that $\tilde{\beta} = 0$. If we reject the null, then the fixed effects model is more appropriate than either the pooled or random effects model. If we fail to reject the null, then we can use either the random effects or pooled estimators.

Panel data definitions and work flow:

1. Determine N and T (or the average T if unbalanced)
2. Fit a pooled/RE and CRE models and conduct a nested model test (with clustered or Newey-West standard errors)

3. If the nested test rejects the null, then use the CRE or LSDV/within or FD estimator (decision shouldn't really matter much here). LSDV/within is a good default here.
4. If the nested test fails to the reject the null. Use the pooled estimator or the RE-FGLS estimator with appropriate standard errors.

Recap: the type of exogeneity assumptions matter here just like time series,

- If strict within-unit exogenity holds $\mathrm{E}[\varepsilon_{it}|x_i] = 0$, then all are unbiased
- If strict within-unit exogeneity fails, but weak exogeneity holds $\mathrm{E}[\varepsilon_{it}|x_{it}] = 0$, then all are consistent estimators of β in N or T
- If exogeneity fails because of time-invariant omitted variables than the FD and LSDV/within estimators are unbiased for β, δ_i, if strict exogneity otherwise holds, and consistent in N (for β) or T (for β or δ_i), if weak exogneity otherwise holds.

As always, any exogeneity assumption you make will be violated under omitted variables. If the omitted variables are time invariant, then only FE estimators are unbiased (under strict exogeneity) and consistent (under weak or strict exogeneity). This is the main reason to use a fixed effects estimator. Also, strict exogeneity will never hold if there is a lagged dependent variable, but LSDV/within estimators are still consistent in T.

8.4 Applications

```r
library(readstata13)
library(lmtest)
library(sandwich)
library(car)
library(margins)
library(stargazer)
library(dplyr) #data manipulation
library(plm) #panel models

cw.data <- read.dta13("Rcode/datasets/civwar.dta",
                      nonint.factors = TRUE)
# create a panel data object for when we move to RE/FE
cw.panel <- pdata.frame(cw.data, index=c("ccode", "year"))
within.var <- function(mod, data, idx.N, idx.T){
  invariant <- mod$model %>%
    cbind.data.frame(data[row.names(mod$model),c(idx.N, idx.T)],.) %>%
    select(!contains(idx.T)) %>%
    group_by(get(idx.N)) %>%
    summarize(across(.fns=~all(duplicated(.)[-1]))) %>%
    summarize(across(.cols=!1:3,.fns=all))
  return(colnames(invariant)[invariant[1,]==TRUE])
}

pdim(cw.panel) #large N and large T, we'll do clustered errors not Newey-West
```

```
## Unbalanced Panel: n = 161, T = 7-55, N = 6610
```

```
## let's start basic (pooled)
model1 <- lm(war~lpopl1+polity2l+I(polity2l^2)+lgdpenl1+nwstate
             +ncontig+oil
             +ethfrac+relfrac
             +colbrit+colfra+lmtnest+
               + factor(region) ,
             data=cw.data)
signif(coeftest(model1, vcov=vcovHC)[1:12,],2)
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.08800    0.06000   -1.50  1.5e-01
## lpopl1         0.05100    0.00330   16.00  5.0e-54
## polity2l       0.00620    0.00066    9.40  8.9e-21
## I(polity2l^2) -0.00083    0.00016   -5.10  4.5e-07
## lgdpenl1      -0.05100    0.00570   -8.90  6.5e-19
## nwstate       -0.02400    0.02500   -0.94  3.5e-01
## ncontig        0.13000    0.01500    8.30  1.7e-16
## oil            0.02800    0.01400    2.00  4.5e-02
## ethfrac        0.08200    0.02000    4.10  4.8e-05
## relfrac        0.00950    0.02200    0.43  6.7e-01
## colbrit       -0.02100    0.01300   -1.60  1.2e-01
## colfra        -0.02600    0.01500   -1.70  8.2e-02
```

```
# We'll make a special function for clustering
# so we don't have to keep specifying the cluster
# variable
```

```
cluster.civilwar <- function(mod){
  return(vcovCL(mod,cluster=cw.data$ccode))
}

signif(coeftest(model1, vcov=cluster.civilwar)[1:13,],2)
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.08800    0.20000   -0.44  6.6e-01
## lpopl1         0.05100    0.01300    4.10  4.9e-05
## polity2l       0.00620    0.00170    3.60  2.7e-04
## I(polity2l^2) -0.00083    0.00049   -1.70  8.9e-02
## lgdpenl1      -0.05100    0.01900   -2.60  8.5e-03
## nwstate       -0.02400    0.03200   -0.74  4.6e-01
## ncontig        0.13000    0.07800    1.60  1.1e-01
## oil            0.02800    0.04200    0.67  5.0e-01
## ethfrac        0.08200    0.07500    1.10  2.7e-01
## relfrac        0.00950    0.08300    0.11  9.1e-01
## colbrit       -0.02100    0.05600   -0.37  7.1e-01
## colfra        -0.02600    0.04900   -0.52  6.0e-01
## lmtnest        0.01000    0.01200    0.89  3.7e-01
```

```
## add in year dummies
model1.years <- lm(war~lpopl1+polity2l+I(polity2l^2)+lgdpenl1+nwstate
                   +ncontig+oil
                   +ethfrac+relfrac
                   +colbrit+colfra+lmtnest+
                     + factor(region)
                   +factor(year),
                   data=cw.data)
round(coeftest(model1.years,
               vcov=cluster.civilwar)[1:13,],4)
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.3307     0.2214  1.4936   0.1353
## lpopl1          0.0436     0.0123  3.5334   0.0004
## polity2l        0.0057     0.0019  3.0771   0.0021
## I(polity2l^2)  -0.0009     0.0005 -1.8789   0.0603
## lgdpenl1       -0.1029     0.0238 -4.3173   0.0000
## nwstate         0.0207     0.0326  0.6354   0.5252
## ncontig         0.1336     0.0773  1.7289   0.0839
## oil             0.0315     0.0402  0.7832   0.4335
## ethfrac         0.0876     0.0742  1.1795   0.2382
## relfrac         0.0309     0.0854  0.3622   0.7172
## colbrit        -0.0369     0.0550 -0.6705   0.5026
## colfra         -0.0492     0.0484 -1.0149   0.3102
## lmtnest         0.0055     0.0116  0.4715   0.6373
```

```r
model.re <- plm(war~lpopl1+polity2l+I(polity2l^2)+lgdpenl1+nwstate
                 +ncontig+oil
                 +ethfrac+relfrac
                 +colbrit+colfra+lmtnest
                 +factor(region)
                 +factor(year),
                 data=cw.panel,
                 model="random",
                random.method = "walhus")
pbgtest(model.re) #panel version of BG test for serial correlation
```

```
##
##  Breusch-Godfrey/Wooldridge test for serial correlation in panel models
##
## data:  war ~ lpopl1 + polity2l + I(polity2l^2) + lgdpenl1 + nwstate +     ncontig + o
## chisq = 3703.9, df = 3, p-value < 2.2e-16
## alternative hypothesis: serial correlation in idiosyncratic errors
```

```r
## ASIDE: Clustered standard errors vary based on if you have lm or plm
# If using lm you use vcovHC for robust and vcovCL for clustering (sandwich)
# If using plm you use vcovHC from the plm package for both
vcovRE <- vcovHC(model.re, cluster="group")
round(coeftest(model.re, vcovRE)[1:13,],4)
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.5978     0.2841  2.1042   0.0354
## lpopl1            0.0404     0.0195  2.0739   0.0381
## polity2l          0.0010     0.0022  0.4768   0.6335
## I(polity2l^2)  -0.0016     0.0005 -3.4975   0.0005
## lgdpenl1       -0.1090     0.0249 -4.3696   0.0000
## nwstate         0.0425     0.0324  1.3104   0.1901
## ncontig        -0.0444     0.1270 -0.3495   0.7268
## oil            -0.0203     0.0254 -0.7993   0.4242
## ethfrac         0.0156     0.0854  0.1829   0.8549
## relfrac         0.0483     0.0873  0.5531   0.5802
## colbrit        -0.0389     0.0604 -0.6433   0.5201
## colfra         -0.0764     0.0623 -1.2274   0.2197
## lmtnest         0.0026     0.0116  0.2255   0.8216
```

```
# moving to FE we need to first ID which variables are constant
# within units
within.var(model1, cw.data, "ccode", "year")
```

```
## [1] "relfrac"         "colbrit"         "colfra"          "lmtnest"
## [5] "factor(region)"
```

```
## Drop the constant years
model.within <- plm(war~lpopl1+polity2l+I(polity2l^2)+lgdpenl1+nwstate+ncontig
                     +oil+ethfrac-1, data=cw.panel, model="within", effect="twoways")
vcovFE <- vcovHC(model.within, cluster="group")
summary(model.within, vcov=vcovFE)
```

```
## Twoways effects Within Model
##
## Note: Coefficient variance-covariance matrix supplied: vcovFE
##
## Call:
## plm(formula = war ~ lpopl1 + polity2l + I(polity2l^2) + lgdpenl1 +
##     nwstate + ncontig + oil + ethfrac - 1, data = cw.panel, effect = "twoways",
##     model = "within")
##
## Unbalanced Panel: n = 156, T = 3-55, N = 6327
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -0.894987 -0.088025 -0.011341  0.054728  1.125709
```

```
##
## Coefficients:
##                   Estimate  Std. Error t-value  Pr(>|t|)
## lpopl1          0.03029390  0.06532354  0.4638 0.6428423
## polity2l        0.00052093  0.00227043  0.2294 0.8185350
## I(polity2l^2) -0.00162059  0.00047016 -3.4469 0.0005709 ***
## lgdpenl1      -0.11236350  0.02981168 -3.7691 0.0001654 ***
## nwstate         0.04571893  0.03311978  1.3804 0.1675105
## ncontig       -0.32228105  0.04302900 -7.4899 7.872e-14 ***
## oil           -0.03870732  0.03018082 -1.2825 0.1997112
## ethfrac       -0.71283175  0.22332463 -3.1919 0.0014205 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    366.33
## Residual Sum of Squares: 351.23
## R-Squared:      0.041203
## Adj. R-Squared: 0.0071453
## F-statistic: 50.8164 on 8 and 155 DF, p-value: < 2.22e-16
```

```r
linearHypothesis(model.within,c("polity2l=0", "I(polity2l^2)=0"), vcov=vcovFE)
```

```
## Linear hypothesis test
##
## Hypothesis:
## polity2l = 0
## I(polity2l^2) = 0
##
## Model 1: restricted model
## Model 2: war ~ lpopl1 + polity2l + I(polity2l^2) + lgdpenl1 + nwstate +
##     ncontig + oil + ethfrac - 1
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df  Chisq Pr(>Chisq)
## 1   6111
## 2   6109  2 11.888   0.002622 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# FE (LSDV)
model.LSDV <- lm(war~lpopl1+polity2l+I(polity2l^2)+lgdpenl1+nwstate+ncontig
                  +oil+ethfrac
                  +factor(ccode) + factor(year)-1,
                  data=cw.data)
signif(coeftest(model.LSDV, vcov=cluster.civilwar)[1:10,],2)
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## lpopl1              0.03000    0.06700    0.45  6.5e-01
## polity2l            0.00052    0.00230    0.22  8.2e-01
## I(polity2l^2)   -0.00160    0.00048   -3.40  7.4e-04
## lgdpenl1          -0.11000    0.03000   -3.70  2.2e-04
## nwstate             0.04600    0.03400    1.40  1.8e-01
## ncontig           -0.32000    0.04400   -7.30  2.5e-13
## oil               -0.03900    0.03100   -1.30  2.1e-01
## ethfrac           -0.71000    0.23000   -3.10  1.8e-03
## factor(ccode)2      1.30000    0.81000    1.60  1.2e-01
## factor(ccode)20  1.30000    0.63000    2.00  4.5e-02
```

```
linearHypothesis(model.LSDV,c("polity2l=0", "I(polity2l^2)=0"),
                 test="Chisq",
                 vcov=cluster.civilwar)
```

```
## Linear hypothesis test
##
## Hypothesis:
## polity2l = 0
## I(polity2l^2) = 0
##
## Model 1: restricted model
## Model 2: war ~ lpopl1 + polity2l + I(polity2l^2) + lgdpenl1 + nwstate +
##     ncontig + oil + ethfrac + factor(ccode) + factor(year) -
##     1
##
## Note: Coefficient covariance matrix supplied.
```

```
##
##   Res.Df Df  Chisq Pr(>Chisq)
## 1   6111
## 2   6109  2 11.406   0.003336 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# note we can extract the individual intercepts
head(fixef(model.within))
```

```
##         2        20        40        41        42        51
## 1.2547580 1.2628873 0.6730103 0.5903736 0.5726310 0.6394049
```

```
tail(fixef(model.within))
```

```
##      840      850      900      910      920      950
## 1.946320 2.010918 1.282981 1.582068 1.358539 1.431548
```

```
# For random effects
head(ranef(model.re)+model.re$coefficients[1])
```

```
##         2        20        40        41        42        51
## 0.5624116 0.5812645 0.5559889 0.5686911 0.5423391 0.5682329
```

```
# Mundlak version
within.data <- cw.data %>%
  slice(as.numeric(row.names(model.LSDV$model)))
cw.data <- within.data %>%
  group_by(ccode) %>%
  mutate(lpopl1.mean= mean(lpopl1, na.rm=T),
         polity2l.mean =mean(polity2l, na.rm=T),
         polity2l.sq = polity2l^2,
         polity2l.sq.mean =mean(polity2l.sq, na.rm=T),
         lgdpenl1.mean=mean(lgdpenl1,na.rm=T),
         nwstate.mean =mean(nwstate,na.rm=T),
         ncontig.mean =mean(ncontig,na.rm=T),
         oil.mean =mean(oil,na.rm=T),
         ethfrac.mean =mean(ethfrac,na.rm=T))
cw.panel <- pdata.frame(cw.data, index=c("ccode", "year"))
model.cre <- plm(war~lpopl1+polity2l+I(polity2l^2)+lgdpenl1+nwstate
                  +ncontig+oil+ethfrac
                  + lpopl1.mean+polity2l.mean
                  +  polity2l.sq.mean+lgdpenl1.mean
                  +  nwstate.mean+ncontig.mean+oil.mean
                  +ethfrac.mean
                  +relfrac
                  +colbrit+colfra+lmtnest
                  +factor(year),
                 model="random",
                 random.method = "walhus",
                  data=cw.panel)
vcovCRE <- vcovHC(model.cre, cluster="group")
signif(coeftest(model.cre, vcovCRE)[1:17,],2)
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.45000    0.25000   1.800  6.5e-02
## lpopl1            0.03100    0.06500   0.480  6.3e-01
## polity2l          0.00054    0.00230   0.240  8.1e-01
## I(polity2l^2)    -0.00160    0.00047  -3.400  5.6e-04
## lgdpenl1         -0.11000    0.02900  -3.800  1.4e-04
## nwstate           0.04600    0.03300   1.400  1.7e-01
## ncontig          -0.32000    0.04200  -7.600  3.0e-14
## oil              -0.03900    0.03000  -1.300  2.0e-01
## ethfrac          -0.71000    0.22000  -3.200  1.2e-03
## lpopl1.mean       0.01200    0.06800   0.170  8.6e-01
## polity2l.mean     0.00630    0.00360   1.800  7.5e-02
## polity2l.sq.mean  0.00041    0.00087   0.470  6.4e-01
## lgdpenl1.mean    -0.00180    0.04300  -0.042  9.7e-01
## nwstate.mean     -0.64000    0.22000  -2.800  4.5e-03
## ncontig.mean      0.46000    0.06300   7.200  6.2e-13
## oil.mean          0.17000    0.07200   2.400  1.7e-02
## ethfrac.mean      0.74000    0.24000   3.100  2.1e-03
```

```
signif(cbind(model.within$coefficients,
            model.LSDV$coef[1:8],
            model.cre$coef[2:9],
            model.re$coef[2:9]),
      2)
```

```
##                    [,1]     [,2]     [,3]    [,4]
## lpopl1          0.03000  0.03000  0.03100  0.0400
## polity2l        0.00052  0.00052  0.00054  0.0010
## I(polity2l^2) -0.00160 -0.00160 -0.00160 -0.0016
## lgdpenl1       -0.11000 -0.11000 -0.11000 -0.1100
## nwstate         0.04600  0.04600  0.04600  0.0430
## ncontig        -0.32000 -0.32000 -0.32000 -0.0440
## oil            -0.03900 -0.03900 -0.03900 -0.0200
## ethfrac        -0.71000 -0.71000 -0.71000  0.0160
```

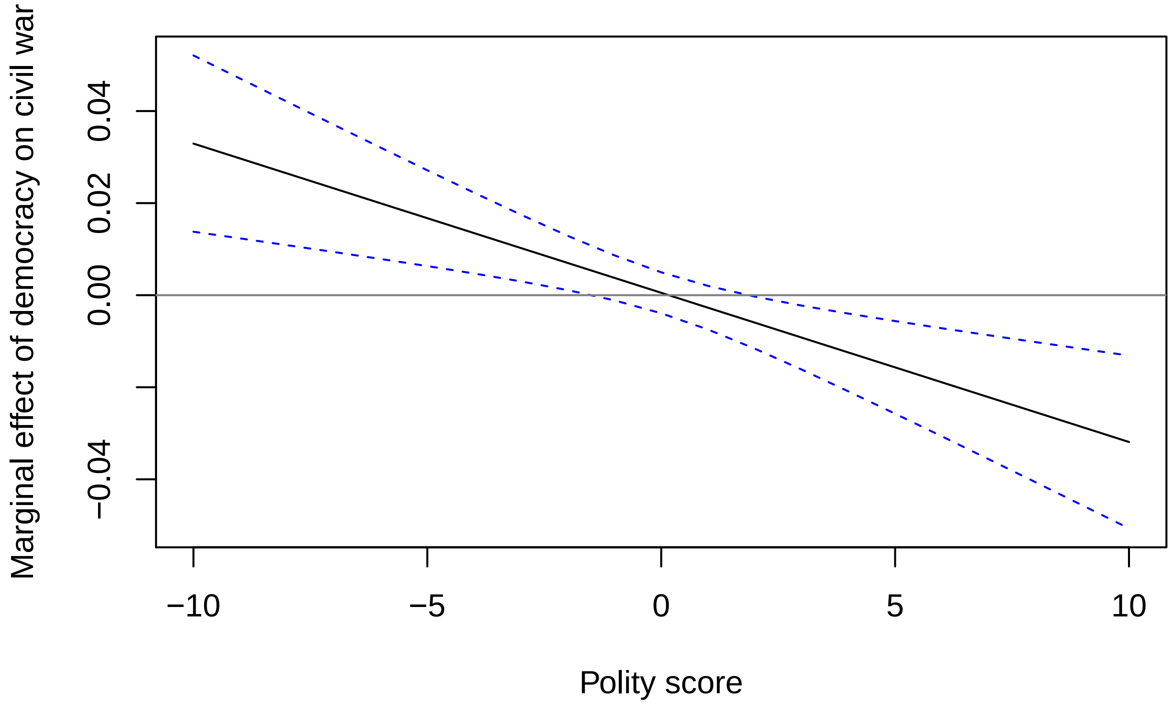
```
linearHypothesis(model.cre,
                 c("lpopl1.mean=0",
                   "polity2l.mean=0",
                   "polity2l.sq.mean=0",
                   "lgdpenl1.mean=0",
                   "nwstate.mean=0",
                   "ncontig.mean=0",
                   "oil.mean=0"),
                 vcov=vcovCRE)
```

```
## Linear hypothesis test
##
## Hypothesis:
## lpopl1.mean = 0
## polity2l.mean = 0
## polity2l.sq.mean = 0
## lgdpenl1.mean = 0
## nwstate.mean = 0
## ncontig.mean = 0
## oil.mean = 0
##
## Model 1: restricted model
## Model 2: war ~ lpopl1 + polity2l + I(polity2l^2) + lgdpenl1 + nwstate +
##     ncontig + oil + ethfrac + lpopl1.mean + polity2l.mean + polity2l.sq.mean +
##     lgdpenl1.mean + nwstate.mean + ncontig.mean + oil.mean +
##     ethfrac.mean + relfrac + colbrit + colfra + lmtnest + factor(year)
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df  Chisq Pr(>Chisq)
## 1   6259
## 2   6252  7 74.882  1.516e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### NOTES: MARGINS
# okay you guys got lucky. Margins didn't use to work on plm models
# Now it does!
ME <- margins(model.within,
              variables = "polity2l",
              at=list(polity2l=-10:10), vcov=vcovFE)
summary(ME)
```

```
##    factor polity2l     AME     SE       z      p   lower   upper
##  polity2l -10.0000  0.0329 0.0098  3.3714 0.0007  0.0138  0.0521
##  polity2l  -9.0000  0.0297 0.0089  3.3526 0.0008  0.0123  0.0470
##  polity2l  -8.0000  0.0265 0.0080  3.3267 0.0009  0.0109  0.0420
##  polity2l  -7.0000  0.0232 0.0071  3.2898 0.0010  0.0094  0.0370
##  polity2l  -6.0000  0.0200 0.0062  3.2353 0.0012  0.0079  0.0321
##  polity2l  -5.0000  0.0167 0.0053  3.1509 0.0016  0.0063  0.0271
##  polity2l  -4.0000  0.0135 0.0045  3.0125 0.0026  0.0047  0.0223
##  polity2l  -3.0000  0.0102 0.0037  2.7712 0.0056  0.0030  0.0175
##  polity2l  -2.0000  0.0070 0.0030  2.3266 0.0200  0.0011  0.0129
##  polity2l  -1.0000  0.0038 0.0025  1.5080 0.1315 -0.0011  0.0087
##  polity2l   0.0000  0.0005 0.0023  0.2294 0.8185 -0.0039  0.0050
##  polity2l   1.0000 -0.0027 0.0024 -1.1242 0.2609 -0.0075  0.0020
##  polity2l   2.0000 -0.0060 0.0029 -2.0664 0.0388 -0.0116 -0.0003
##  polity2l   3.0000 -0.0092 0.0035 -2.5967 0.0094 -0.0161 -0.0023
##  polity2l   4.0000 -0.0124 0.0043 -2.8881 0.0039 -0.0209 -0.0040
##  polity2l   5.0000 -0.0157 0.0051 -3.0563 0.0022 -0.0257 -0.0056
##  polity2l   6.0000 -0.0189 0.0060 -3.1597 0.0016 -0.0307 -0.0072
##  polity2l   7.0000 -0.0222 0.0069 -3.2269 0.0013 -0.0356 -0.0087
## polity2l   8.0000 -0.0254 0.0078 -3.2729 0.0011 -0.0406 -0.0102
## polity2l   9.0000 -0.0286 0.0087 -3.3056 0.0009 -0.0456 -0.0117
## polity2l  10.0000 -0.0319 0.0096 -3.3297 0.0009 -0.0507 -0.0131
```

```
plot.AME <- summary(ME)
plot(AME~polity2l, data=plot.AME,
     type="l",
     ylim=c(min(plot.AME$lower),
            max(plot.AME$upper)),
     ylab="Marginal effect of democracy on civil war",
     xlab="Polity score"
)
lines(lower~polity2l, data=plot.AME, type="l", lty="dashed", col="blue")
lines(upper~polity2l, data=plot.AME, type="l", lty="dashed", col="blue")
abline(h=0,col="grey50")
```



```
### linearHypothesis and deltaMethod work as expected (mostly)
#note the use ` ` when the var name has a parenthesis
# One note: the parameterNames option in deltaMethods does *not* work
# with plm objects at this time
deltaMethod(model.within,
            g="-polity2l/(2*`I(polity2l^2)`)",
            vcov=vcovFE)
```

```
##                                 Estimate       SE    2.5 % 97.5 %
## -polity2l/(2 * `I(polity2l^2)`)  0.16072  0.70003 -1.21131 1.5328
```

```
# Put it all together
model.list <- list(model1,
                   model1.years,
                   model.re,
                   model.cre,
                   model.within,
                   model.LSDV)
var.list <- list(vcovCL(model1, cluster=cw.data$ccode),
                 vcovCL(model1.years, cluster=cw.data$ccode),
                 vcovRE,
                 vcovCRE,
                 vcovFE,
                 vcovCL(model.LSDV, cluster=cw.data$ccode))
se.list <- lapply(var.list, function(x){sqrt(diag(x))})
stargazer(model.list,
          se=se.list,
          float.env = "sidewaystable",
          title="Panel models of civil war onset",
          label="tab:p.civwar",
          no.space=TRUE,
          omit=c("factor\\(c*", ".mean"),
          dep.var.labels = "Civil War Onset",
          model.names = FALSE,
          model.numbers = FALSE,
          column.labels = c("OLS", "OLS", "RE", "CRE", "FE-Within", "FE-LSDV"),
          header=FALSE,
          font.size = "small",
          add.lines=list( c("Region Dummies",
                            "No", rep("Yes", 3), "No", "No"),
                          c("Year Dummies",
                            "No", rep("Yes", 5)),
                          c("Country Dummies",
                            rep("No",4), "--", "Yes")),
          covariate.labels = c("Population (log)",
                               "Democracy",
                               "Democracy sq.",
                               "GDP pc (log)",
                               "New State",
                               "Non-continguous state",
                               "Oil exporter",
                               "Ethnic Frac.",
                               "Religious Frac",
                               "British Colony",
                               "French Colony",
                               "\\% Mountainous terrain (log)"),
          keep.stat = c("n", "adj.rsq"))
```

8.5 Bonus topic: Dynamic panel models

We mentioned above that including lagged dependent variable in a panel model violates strict exogeneity. Unless we have iid observations within units, this will induce bias in the estimates. Given that we are unlikely to have iid observations if the process is dynamic, other strategies will be needed. Here are three

1. Use the LSDV/within estimator with large T
   - The bias in the LSDV/within estimates is decreasing in T. Like with ordinary time series data, we need a long series to get consistency with weak exogeneity and dependent observations.
2. Anderson-Hsiao (AH) estimator
   - Given that the problem is endogeneity, we should instrument for it.
   - Start with the FD estimator

     $$\Delta y_{it} = \beta'(\Delta x_{it}) + \gamma(\Delta y_{it-1}) + \Delta\varepsilon_{it},$$

     and instrument for Δy_{it-1} with y_{it-2}.
3. Arellano-Bond (AB) estimator

Table 9: Panel models of civil war onset

	Dependent variable:					
	Civil War Onset					
	OLS	OLS	RE	CRE	FE-Within	FE-LSDV
Population (log)	0.051***	0.044***	0.040**	0.031	0.030	0.030
	(0.013)	(0.012)	(0.020)	(0.065)	(0.065)	(0.067)
Democracy	0.006***	0.006***	0.001	0.001	0.001	0.001
	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)	(0.002)
Democracy sq.	−0.001*	−0.001*	−0.002***	−0.002***	−0.002***	−0.002***
	(0.0005)	(0.001)	(0.0005)	(0.0005)	(0.0005)	(0.0005)
GDP pc (log)	−0.051***	−0.103***	−0.109***	−0.112***	−0.112***	−0.112***
	(0.019)	(0.024)	(0.025)	(0.029)	(0.030)	(0.030)
New State	−0.024	0.021	0.043	0.046	0.046	0.046
	(0.032)	(0.033)	(0.032)	(0.033)	(0.033)	(0.034)
Non-continguous state	0.126	0.134*	−0.044	−0.323***	−0.322***	−0.322***
	(0.078)	(0.077)	(0.127)	(0.042)	(0.043)	(0.044)
Oil exporter	0.028	0.031	−0.020	−0.039	−0.039	−0.039
	(0.042)	(0.040)	(0.025)	(0.030)	(0.030)	(0.031)
Ethnic Frac.	0.082	0.088	0.016	−0.715***	−0.713***	−0.713***
	(0.075)	(0.074)	(0.085)	(0.221)	(0.223)	(0.228)
Religious Frac	0.010	0.031	0.048	−0.043		
	(0.083)	(0.085)	(0.087)	(0.083)		
British Colony	−0.021	−0.037	−0.039	0.023		
	(0.056)	(0.055)	(0.060)	(0.046)		
French Colony	−0.026	−0.049	−0.076	−0.001		
	(0.049)	(0.048)	(0.062)	(0.055)		
% Mountainous terrain (log)	0.010	0.005	0.003	0.015		
	(0.012)	(0.012)	(0.012)	(0.012)		
Constant	−0.088	0.331	0.598**	0.454*		
	(0.197)	(0.221)	(0.284)	(0.246)		
Region Dummies	No	Yes	Yes	Yes	No	No
Year Dummies	No	Yes	Yes	Yes	Yes	Yes
Country Dummies	No	No	No	No	–	Yes
Observations	6,327	6,327	6,327	6,327	6,327	6,327
Adjusted R^2	0.182	0.213	0.086	0.092	0.007	0.586
Note:						$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

- Instrumenting is a good idea, but there are tons of potential instruments.
- Start with the FD estimator

$$\Delta y_{it} = \beta'(\Delta x_{it}) + \gamma(\Delta y_{it-1}) + \Delta\varepsilon_{it},$$

- Let $\Delta\mathbf{x}_{it} = [\Delta y_{it-1}, \Delta x_{it}]$, then the instrument matrix for the AB estimator for group i is formed by using all the available lags of y_{it}

$$Z_i = \begin{bmatrix} y_{i1} & 0 & 0 & 0 & \dots & 0 & 0 & | & \Delta x_{i3} \\ 0 & y_{i1} & y_{i2} & 0 & \dots & 0 & 0 & | & \Delta x_{i4} \\ 0 & 0 & 0 & y_{i1:3} & 0 & \dots & 0 & | & \Delta x_{i5} \\ \vdots & \vdots & \vdots & \dots & \ddots & \vdots & \vdots & | & \vdots \end{bmatrix}.$$

  Now we form full Z matrix by stacking these up $Z = [Z_i]_{i=1}^N$, and likewise to make $\mathbf{X} = \left[[\Delta\mathbf{x}_{it}]_{t=3}^T\right]_{i=1}^N$, where we lose $t = 1, 2$ to lags and differences. Other variations exist, one that makes life easier is the "collapsed" version

$$Z_i = \begin{bmatrix} y_{i1} & 0 & 0 & \dots & | & \Delta x_{i3} \\ y_{i2} & y_{i1} & 0 & \dots & | & \Delta x_{i4} \\ y_{i3} & y_{i2} & y_{i3} & \dots & | & \Delta x_{i5} \\ \vdots & \vdots & \vdots & \dots & | & \vdots \end{bmatrix}.$$

  with 0s used to fill in values lost to the lagging process. Another approach is called the "system" estimator. It uses both lagged values and lagged differences as instruments in either the collapsed or uncollapsed form.
- Fit the model using GMM (discussed above in the advanced IV topics). The function `pgmm` in the `plm` package does this for you. Note that with bigger data sets, this matrix can be quite unweldy and computationally troublesome from a memory perspective. In these cases, AH is good alternative.

The AH and AB estimators are best for small-T, big-N panels. If T is large you should be just fine with the LSDV/within. The AH estimator can be fit with `ivreg`, `plm`, or `pgmm` (probably `pgmm` is easiest, just use the right options to match what's listed). The AB estimator is fit with `pgmm`.

8.6 Bonus topic: A primer on difference-in-differences

Panel data provides an important avenue for fighting endogeneity. As we know, fixed effects estimators control for any invariant omitted variables. Beyond that however, we can use repeated observations over time to uncover interesting casual estimates. Before we go any further, let's define a few new concepts

8.6.1 Casual inference and potential outcomes

- *Potential outcome* We will let $y_i(x_i)$ be the potential outcome that y_i takes on when observation i receives treatment $x_i \in \{0, 1\}$
- *Treatment effect* The effect that receiving treatment has on y_i

  $$TE(x_i) = y_i(1) - y_i(0).$$

  The fundamental problem of casual inference is that each individual can only ever observe one potential outcome. We can never observe both.
- *Average treatment effect* The ATE is defined as

  $$\mathrm{E}[y_i(1) - y_i(0)].$$

- *Average treatment effect on the treated* The ATT is defined as

  $$\mathrm{E}[y_i(1) - y_i(0)|x_i = 1].$$

Let's consider the relationship among these quantities. First,let's rewrite the ATE in terms of observed data

$$\begin{aligned}\mathrm{E}[y_i|x_i = 1] - \mathrm{E}[y_i|x_i = 0] &= \mathrm{E}[y_i(1)|x_i = 1] - \mathrm{E}[y_i(0)|x_i = 0] \quad \text{by assumption}\\ &= \underbrace{\mathrm{E}[y_i(1)|x_i = 1] - \mathrm{E}[y_i(0)|x_i = 1]}_{\text{ATT}} - \underbrace{\mathrm{E}[y_i(0)|x_i = 0] + \mathrm{E}[y_i(0)|x_i = 1]}_{\text{Selection bias (endogeneity)}}.\end{aligned}$$

What does this tell us? The ATE is a combination of the ATT and bias that is introduced through non-random treatment assignment. And what is the ATT?

$$\begin{aligned}\mathrm{ATT}(x_i) &= \mathrm{E}[y_i(1) - y_i(0)|x_i = 1]\\ &= \underbrace{\mathrm{E}[y_i(1)|x_i = 1]}_{\text{observable}} - \underbrace{\mathrm{E}[y_i(0)|x_i = 1]}_{\text{counterfactual}}\end{aligned}$$

Here, we can see part of the problem, we need to figure out to estimate the counterfactual component to make advancement here. In substantive terms, we need to know what would have happened to the treated observations if they hadn't been treated. The selection bias then comes from the possibility that untreated individuals may not provide good estimates for the counterfactual quantity $\mathrm{E}[y_i(0)|x_i = 1]$. Put another way, the types of people who do an do not seek out a job training program may not be the same on qualities like ambition, connections, skills. These differences make $\mathrm{E}[y_i(0)|x_i = 0]$ and $\mathrm{E}[y_i(0)|x_i = 1]$ potentially quite different. If however, we can minimize these concerns (through random treatment assignment, say) then

$$\mathrm{E}[y_i(0)|x_i = 0] = \mathrm{E}[y_i(0)|x_i = 1]$$

and

$$\mathrm{E}[y_i|x_i = 0] - \mathrm{E}[y_i|x_i = 1] = \mathrm{E}[y_i(1) - y_i(0)|x_i = 1] = \mathrm{E}[y_i(1) - y_i(0)].$$

With observational (non-experimental) data we will often be unable to identify the ATE without an interesting theoretical model that we can rely on. However, we can sometimes get to the ATT under various design-based assumptions. One such way to do that with panel data is with a difference-in-differences (DiD) design.

8.6.2 DiD framework

Let's start with a simple case two groups $s = 1, 2$, within-group units $i = 1, \ldots, N$, where each unit is observed for two time periods $t = 1, 2$. Neither group receives treatment in period 1, but one group receives it between periods 1 and 2. This setup is called a 2×2 DiD design. Without loss of generality, suppose that $s = 1$ is the group that receives treatment. We will work backwards on this one by starting with what might strike us as an intuitive quantity and then see what it actually tells us. Let $\bar{y}(s,t)$ be the sample mean of the outcome y within group s in period t. Then we will consider the quantity

$$\hat{\beta} = \left(\bar{y}(s = 1, t = 2) - \bar{y}(s = 1, t = 1)\right) - \left(\bar{y}(s = 2, t = 2) - \bar{y}(s = 2, t = 1)\right).$$

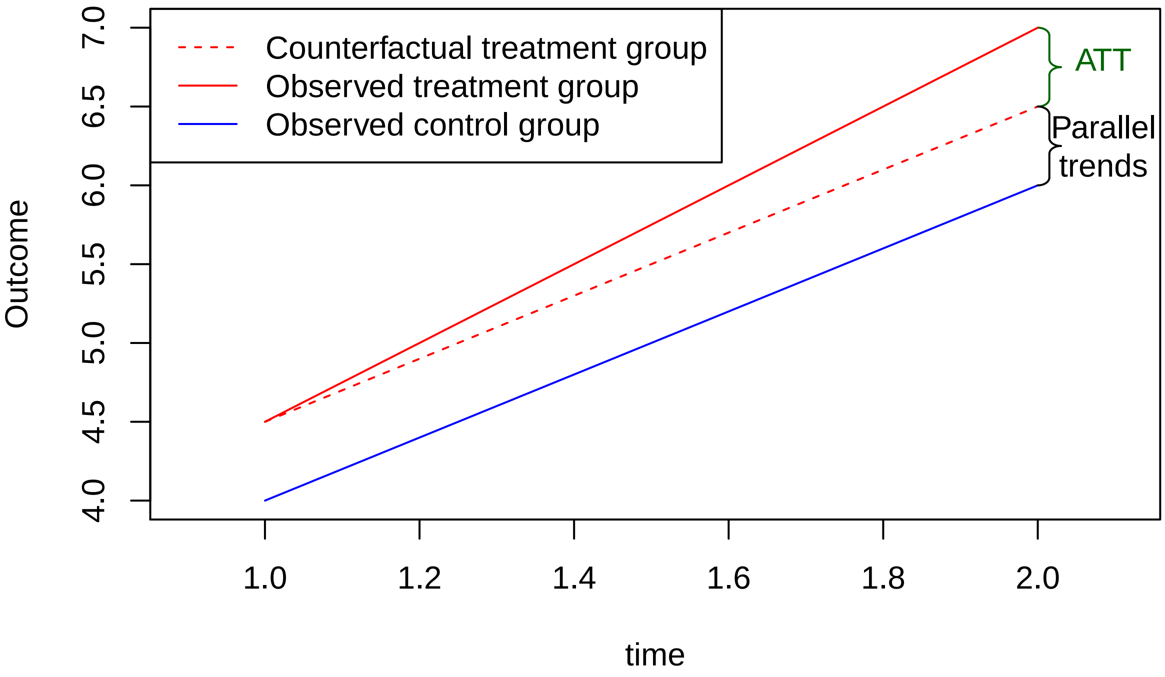
What makes this quantity intuitive? It provides us the average change in y after treatment and then subtracts out how much y changed absent treatment.

What do we actually estimate with this quantity? Let's replace these sample means with their expected values

$$\begin{aligned}
\beta &= (\mathrm{E}[y_{ist}|s_i=1,t=2]-\mathrm{E}[y_{ist}|s_i=1,t=1])-(\mathrm{E}[y_{ist}|s_i=2,t=2]-\mathrm{E}[y_{ist}|s_i=2,t=1]) \\
&= (\mathrm{E}[y_{ist}(1)|s_i=1,t=2]-\mathrm{E}[y_{ist}(0)|s_i=1,t=1]) \quad \text{by assumption} \\
&\quad -(\mathrm{E}[y_{ist}(0)|s_i=2,t=2]-\mathrm{E}[y_{ist}(0)|s_i=2,t=1]) \quad \text{by assumption} \\
&\quad +\mathrm{E}[y_{ist}(0)|s_i=1,t=2]-\mathrm{E}[y_{ist}(0)|s_i=1,t=2] \quad \text{add 0} \\
&= \underbrace{(\mathrm{E}[y_{ist}(1)|s_i=1,t=2]-\mathrm{E}[y_{ist}(0)|s_i=1,t=2])}_{\text{ATT}} \\
&\quad +\underbrace{(\mathrm{E}[y_{ist}(0)|s_i=1,t=2]-\mathrm{E}[y_{ist}(0)|s_i=1,t=1])}_{\text{Trend among treated w/o treatment}} \\
&\quad -\underbrace{(\mathrm{E}[y_{ist}(0)|s_i=2,t=2]-\mathrm{E}[y_{ist}(0)|s_i=2,t=1])}_{\text{Trend among control w/o treatment}}
\end{aligned}$$

The final quantity contains three parts. The first is the ATT. The next two lines make up what is known as the *parallel trends assumptions*. In order for us to identify the ATT, we need these two trends to be identical. The second trend is observed, but the first trends is the counterfactual "what if the treated group hadn't been treated?" We don't observe that.

Graphically, this could be represented as



All right, what's happening here? Well absent treatment we assume that both groups will exhibit parellel trends in the outcome. With this assumption we can compare the observed effect in the treatment group to the counterfactual effect absent treatment to compute the

ATT. The control group is used to establish what those parallel trends should look like. When this assumption is violated, our ATT is biased by how badly the assumption fails.

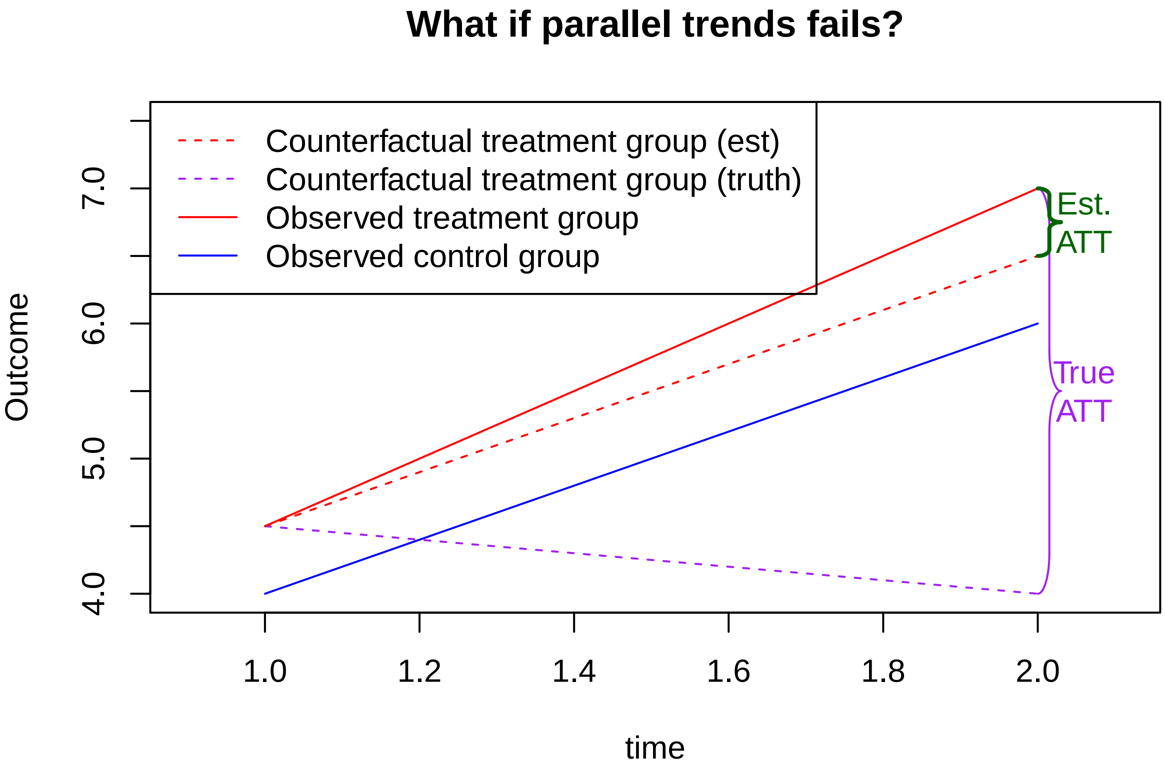
In the 2 ×2 framework, our basic model of interest takes the form

$$y_{ist} = \beta_0 + \beta_1 \underbrace{\mathbb{I}(s_i = 1, t = 2)}_{\text{Treatment group, post}} + \beta_2 \underbrace{\mathbb{I}(t = 2)}_{\text{post}} + + \beta_3 \underbrace{\mathbb{I}(s_i = 1, t = 1)}_{\text{Treatment group, pre}} \varepsilon_{ist}.$$

Turning these dummies on and off we can get some values of interest and one effect:

$$\begin{aligned}
\mathrm{E}[y_{ist}(0)\widehat{|s_i =} 2, t = 1] &= \hat{\beta}_0 \\
\mathrm{E}[y_{ist}(0)\widehat{|s_i =} 1, t = 1] &= \hat{\beta}_0 + \hat{\beta}_3 \\
\mathrm{E}[y_{ist}(1)\widehat{|s_i =} 2, t = 2] &= \hat{\beta}_0 + \hat{\beta}_2 \\
\mathrm{E}[y_{ist}(1)\widehat{|s_i =} 1, t = 2] &= \hat{\beta}_0 + \hat{\beta}_1 + \hat{\beta}_2 + \hat{\beta}_3 \\
\mathrm{E}[y_{ist}(0)\widehat{|s_i =} 1, t = 2] &= \hat{\beta}_0 + \hat{\beta}_2 + \beta_3 \quad \text{Counterfactual} \\
\widehat{\text{ATT}} &= \mathrm{E}[y_{ist}(1)\widehat{|s_i =} 1, t = 2] - \mathrm{E}[y_{ist}(0)\widehat{|s_i =} 1, t = 2] \\
&= \hat{\beta}_1
\end{aligned}$$

If parallel trends is a good assumption than the 2 × 2 DiD will give you what you want. Otherwise, the estimator will impose the control trend on the treatment group to estimate the counterfactual trend. The worse assumption parallel trends is, the worse the bias is because our counterfactual is wrong.



Consider this new figure where the purple dashed line is the true counterfactual. Fitting the DiD model will return the gap between the red solid and red dashed lines which is clearly very different from the true ATT. This occurs because the model builds in the parallel trends assumption.

Unfortunately, parallel trends is not testable. Instead we have to rely on some work-around arguments to assert that parallel trends is not an unreasonable assumption. There are two ways you can think about trying to justify this approach. One is to show that (pre-treatment) the observations in the treatment group look very similar to those in the control group on a host of relevant observables. These checks will help make the case that the two groups are comparable and should look similar on a host of issues along with way. The second approach is to consider pre-treatment placebo effects. This means that you estimate a treatment effect for q pre-treatment periods. These should all be zero. The argument here is that if there are no effects pre-treatment then the DiD lines are all parallel pre-treatment. If the assumption is good pre-treatment, why not post? It's a reach, but take what you get.

Now to fit the placebo test you need more than two-periods. This take us into a more general DiD approach with multiple groups $s = 1, \ldots, S$ and multiple time periods. Let $t = 0$ be the period where treatment is applied. Negative periods will be pre-treatment, positive post-treatment. Without loss of generality, suppose that the first s^* groups are treated and the rest are control. Our model is now

$$y_{ist} = \alpha_s + \tau_t + \sum_{t'=-q}^{-1} \beta_{t'} \mathbb{I}(s \leq s^*, t = t') + \sum_{t^*=0}^{m} \beta_{t^*} \mathbb{I}(s \leq s^*, t = t^*) + \gamma' z_{ist} + \varepsilon_{ist}.$$

Here we want $\beta_{-q}, \ldots, \beta_{-1}$ to be all 0, and $\beta_0, \ldots, \beta_m$ are the treatment effects for m periods out. Note that this model can be fit using OLS with two-way fixed effects (group and time dummies or transformations), where each time dummy is also interacted with an indicator for whether the group receives treatment. We can use `lm` to fit this without trouble.

Another type of placebo test helps provide credibility to this exercise. Here, you want to swap out the outcome with a similar outcome variable, but one where you think there should be no effect. In a minimum wage study, the outcome of interest might be low-wage employment rate, a placebo test may be high wage or prestige employment rate where we suspect that changes in the minimum wage would have little-to-no effect. These tests should be designed to rule out some common trend would effect the outcome other than the treatment of interest (like a recession between periods 1 & 2).

Finally, when treatments occur in different units at different times, things can get weird. Differential timing and right way to think about it is a growing and active field, but more than I want to get into right now, so we'll call it here.

8.7 Bonus topic: Estimating the effect of time-invariant variables with fixed effects

Both of our main estimators for the fixed-effects model remove all time-invariant variables. Most of the time, this isn't a big deal, but sometimes we may want to know the effects of a specific time-invariant variable on the outcome. The easiest method would be to just use the CRE estimator from above. But if any of your invariant variables are also endogenous there is another alternative that can be a little better: the Hausman-Taylor (HT) estimator. Start with the model

$$y_{it} = \beta_1' x_{1it} + \beta_2' x_{2it} + \gamma_1' z_{1i} + \gamma_2' z_{2i} + \alpha_i + u_{it},$$

where

- x_{1it} contains K_1 time-varying exogenous variables
- z_{1i} contains L_1 time-unvarying exogenous variables
- x_{2it} contains K_2 time-varying endogenous variables (correlated with u_i)

- z_{2i} contains L_2 time-unvarying endogenous variables (correlated with u_i).

HT impose some random effects style structure on the model.

$$\begin{aligned}
\mathrm{E}[\alpha_i] = \mathrm{E}[\alpha_i | x_{1it}, z_{1it}] &= 0 \\
\mathrm{Var}(\alpha_i | x_{1it}, x_{2it}, z_{1i}, z_{2i}) &= \sigma_\alpha^2 \\
\mathrm{Cov}(\alpha_i, u_{it} | x_{1it}, x_{2it}, z_{1i}, z_{2i}) &= 0 \\
\mathrm{Var}(u_{it} + \alpha_i | x_{1it}, x_{2it}, z_{1i}, z_{2i}) &= \sigma_\varepsilon^2 = \sigma_\alpha^2 + \sigma_u^2 \\
\mathrm{Cov}(u_{it} + \alpha_i, u_{is} + \alpha_i | x_{1it}, x_{2it}, z_{1i}, z_{2i}) &= \sigma_\alpha^2
\end{aligned}$$

Our goals is to estimate $\theta = (\beta, \gamma) = (\beta_1, \beta_2, \gamma_1, \gamma_2)$. It should be obvious how to obtain consistent estimates of β: the LSDV/within estimator. The standard within transformation removes α_i and thus both x_1 and x_2 are now exogenous for the purposes of estimation.

The HT insight is that the $K_1 + K_2$ within-transformed X's can be instruments for their untransformed counterparts. The exogenous z_1 variables can instrument for themselves and the group means of x_1 can be used as instruments for z_2 so long as $K_1 \geq L_2$. The HT estimator is an FGLS, that takes the following form:

1. Use the FE estimator to obtain estimates of β. Use the residuals from this to produce $\hat{\sigma}_u^2$.

2. Construct the within-group constants using the FE estimates of β

$$\begin{aligned}
\hat{\delta}_i &= \bar{y}_i - \bar{x}_i \hat{\beta}_{FE} \\
&\approx \gamma' z_i + \alpha_i + \bar{u}_i \text{ with abuse}
\end{aligned}$$

   Generate $\hat{\delta}_{it} = \hat{\delta}_i \otimes 1_T$, where $\otimes$ is the Kronecker product. What this means is that we just repeat the within-group mean of the residuals T times within each group.

3. Using 2SLS regress $\hat{\delta}$ on z_1 and z_2 with instruments z_1 and x_1. This will produce consistent estimates of γ. We could stop here since everything is consistent, but to clean but some inefficiency we move on.

4. The variance of the residuals from the 2SLS in the last step is a consistent estimate of $\sigma_\varepsilon^2 = \sigma_\alpha^2 + \sigma_u^2/T$. We have a consistent estimate of σ_u^2 from above, so we can use these two things to back out σ_α^2. The FGLS weights for the RE model are, as we know from above,

$$\lambda = 1 - \frac{\sigma_u}{\sqrt{\sigma_u^2 + T\sigma_\alpha^2}}.$$

5. We now have data, instruments, and weights, we're set. Let

$$\begin{aligned}\mathbf{x}_{it} &= (x_{1it}, x_{2it}, z_{1i}, z_{2i})\\ \tilde{\mathbf{x}}_{it} &= \mathbf{x}_{it} - \lambda\bar{\mathbf{x}}_i\\ \tilde{y}_{it} &= y_{it} - \lambda\bar{y}_i\\ \mathbf{z}_{it} &= (x_{1it} - \bar{x}_{1i}, x_{2it} - \bar{x}_{2i}, z_{1i}, \bar{x}_{1i}),\end{aligned}$$

then are estimates are

$$\hat{\theta}_{HT} = [\tilde{\mathbf{X}}'\mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\tilde{\mathbf{X}}]^{-1}[\tilde{\mathbf{X}}'\mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\tilde{y}],$$

which is the over-identified 2SLS with regressors $\tilde{\mathbf{X}}$ and instruments $\mathbf{Z}$. The problem with this approach is that you need to know which variables are 1s (exogenous) and which are 2s (endogenous), but that's always true in research design. This estimator can be used with `plm` by setting `model="ht"` and specifying the instruments (the help file for `plm` has an HT example).

8.8 Bonus topic: Panel Newey-West standard errors

If you have a large N you should be using clustered standard errors if you believe in Assumption E2. but if you don't have a large N and instead are relying on large-T, small-N asymptotics then you want to use the panel version of Newey West standard errors. Recall that we need to specify Σ_T in order to get to this variance matrix. We being with defining a Newey-West style function

$$\hat{\lambda}(\ell) = \frac{1}{T-\ell-1}\sum_{t=\ell+1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} x_{it}\hat{\varepsilon}_{it}\right)\left(\frac{1}{N}\sum_{i=1}^{N} x_{it-\ell}\hat{\varepsilon}_{it-\ell}\right),$$

Then we will exploit the ergodic nature of the data to get

$$\widehat{\Sigma}_T = \hat{\lambda}(0) + \sum_{\ell=1}^{L}\left(1 - \frac{\ell}{L+1}\right)\left(\hat{\lambda}(\ell) + \hat{\lambda}(\ell)'\right).$$

As before, we have to choose L and we should choose L such that it increases with T. The `plm` function `vcovNW` will implement this for us.