Econometrics Models I-II
Supplementary Material

James Bland

August 27, 2019

Contents

Chapter 1

Introduction

Before getting into the more interesting material, it is important that we are all on the same page in terms of base knowledge. While I hope that there are parts of the course coming up that you can blast through with great understanding and intuition, the reality is that there are a few things that you will need to be able to do in your sleep very early on in the course. I *may* cover these in class, but please make sure you can do them as soon as possible.

1.1 Summation

[See Appendix A of Bailey (2016)]

In Econometrics, we can't avoid adding things up. For example, when we compute a sample mean, we add up all the values in the sample, and divide by the sample size. In practice, we will get our computer to do the heavy lifting for us, but we need to understand what it's doing, and have some notation to make this more compact. For one thing, id we are computing a sample mean, our hope is that we have lots of observations, and so we will need to add a lot of things up. It is cumbersome, for example, to write the sum of integers between 1 and 10 (inclusive), as:

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55 \tag{1.1}$$

Alternatively, we can write:

$$\sum_{k=1}^{10} k = 55 \tag{1.2}$$

We can read (1.2) as follows:

- The summation symbol "$\sum$" (Greek capital sigma) tells us that we are adding things.
- The "k" to the right of the summation symbol is the thing we are adding.
- The "$k = 1$" underneath the summation symbol tells us that we are using k as an index, and we start with $k = 1$.

- The "10" above the summation symbol tells us that we stop summing when we get to 10 (inclusive).

We can, and will, get more sophisticated than this. For example:

$$\sum_{k=1}^{4} 2k^2 = 2 \times 1^2 + 2 \times 2^2 + 2 \times 3^2 + 4 \times 2^2 \tag{1.3}$$

$$\sum_{k=1}^{12} \frac{k-1}{k} = \frac{0}{1} + \frac{1}{2} + \frac{2}{3} + \ldots + \frac{11}{12} \tag{1.4}$$

$$\sum_{l=1}^{3} \sum_{k=1}^{l} lk = 1 \times 1 + 2 \times 1 + 2 \times 2 + 3 \times 1 + 3 \times 2 + 3 \times 3 \tag{1.5}$$

Note that in the last equation, this is a double summation. The index of the leftmost summation (l) appears in the rightmost summation as the stopping point for index k.

The above examples are instructive, but not particularly useful. We are usually interested in adding up a whole lot of things, so we need some notation for "a whole lot of numbers." To do this, let's start with some notation for an arbitrary, indexed number, x_k. You can interpret this as the kth number in a set of numbers. We can denote "a whole lot of numbers", formally a "set of numbers", as:

$$\{x_k\}_{k=1}^{K} = \{x_1, x_2, \ldots, x_K\} \tag{1.6}$$

That is, we have a set of K (a positive integer) numbers, which we index by $k = 1, 2, \ldots, K$.

Here's an example. My drive to work involves driving the following distances, in miles (each distance is the drive distance between turns on Google Maps):

$$\begin{array}{llllll} x_1 = 0.3, & x_2 = 1.4, & x_3 = 0.1, & x_4 = 0.5, & x_5 = 0.0, \\ x_6 = 1.8, & x_7 = 4.2, & x_8 = 0.3, & x_9 = 3.4, & x_{10} = 1.8, \\ x_{11} = 0.2, & x_{12} = 0.8, & x_{13} = 0.2, & x_{14} = 0.1, & x_{15} = 0.1 \end{array}$$

We could denote this dataset as $\{x_t\}_{t=1}^{15}$, and then calculate:

$$\sum_{t=1}^{15} x_t = 15.2 \text{ miles} \qquad \text{total distance}$$

$$\sum_{t=1}^{3} x_t = 1.8 \text{ miles} \qquad \text{distance to 3rd turn}$$

$$\sum_{t=10}^{15} x_t = 3.2 \text{ miles} \qquad \text{distance left after making 9 turns}$$

$$\sum_{t=1}^{15} x_t^2 = 38.8 \text{ miles}^2 \qquad \text{total squared distance (because why not?)}$$

Note the change in units in the las summation.

There are some useful properties of sums. To begin with, multiplying every component of a sum by a constant is the same as multiplying the final value by the constant. That is (see Bailey, Appendix A):

$$\sum_{i=1}^{N} \beta X_i = \beta \sum_{i=1}^{N} X_i \tag{1.7}$$

Suppose, for example, that we wanted to report the total distance above, but in a more widely-accepted unit of measurement. Knowing that 1 mile = 1.6 km (accurate to 1 decimal place), we could do this by computing:

$$\sum_{t=1}^{15} x_t \text{ miles} \times 1.6 \frac{\text{km}}{\text{mile}} = 0.3 \times 1.6 + 1.4 \times 1.6 + \ldots$$

alternatively, we would be smarter and use this result:

$$\sum_{t=1}^{15} x_t \text{ miles} \times 1.6 \frac{\text{km}}{\text{mile}} = 1.6 \frac{\text{km}}{\text{mile}} \sum_{t=1}^{15} x_t \text{ miles} = 1.6 \frac{\text{km}}{\text{mile}} \times 15.4 \text{miles} = 24.3 \text{ km}$$

i.e. $\beta = 1.6 \frac{\text{km}}{\text{mile}}$. Note here that I made sure the unit conversion was correct by writing down the units as well as the numbers. If it's all done correctly, the units you are trying to get rid of should cancel.

Of course, I would *never* want you to waste your time doing such a menial task by hand. If you really wanted to compute these, let your computer do it!

```
clear all // Clear everything from memory
set more off // tell Stata to not stop everytime there is more than one screen of output
use DrBlandsCommute.dta // Load the dataset
list // display tha dataset

quietly summarize x
display r(sum)

quietly summarize x if t<=3
display r(sum)

quietly summarize x if t>=10
display r(sum)

generate x2 = x^2
quietly summarize x2
display r(sum)

generate xkm = x*1.6
quietly summarize xkm
display r(sum)
```

See Chapter 5 to better understand this code.

1.2 Types of random variables

There are many ways to categorize random variables, and we'll get in to a lot of them during this course. For the moment, we will begin by introducing two important types of random variables: discrete and continuous. These do not constitute an exhaustive set (i.e. I could show you some pathological examples that are neither discrete nor continuous), but cover pretty much everything we will be interested in. The distinction between discrete and continuous is important because it tells us how we will (or at least should) analyze our data (see for example Bailey, 2016, chapter 12).

We can tell these types apart by the random variable's *support*. This, loosely, is the set of values that the random variable could possibly take on. That is, let $\mathcal{S}$ be the support of a random variable X. If, say, $3 \in \mathcal{S}$, then it is possible that X could take on the value 3. If $3 \notin \mathcal{S}$, then X could *never* be equal to 3.

1.2.1 Discrete random variables

Discrete random variables have countable supports. Formally, this means that we can assign an integer to every value that the random variable could take on. In fact, discrete random numbers are often stored as integers, even if assigning them a number does not add any value to the problem. Here are some examples of discrete random variables:[1]

- The outcome of a coin toss. We could record this as Heads = 1 and Tails = 0. Hence the support is $\{0, 1\}$
- The number of days with rain in Toledo, OH in 2018. As 2018 is not a leap year, the support is $\{0, 1, 2, \ldots, 365\}$.
- The number of days between Dec 31st, 2017 and when cockroaches become extinct. Note here that, in principal at least, cockroaches could remain extant forever, and so there is no upper bound on the support, hence: $\{0, 1, 2, \ldots, \infty\}$
- The number of coin tosses made until four heads have been observed. It would be impossible to toss the coin fewer than 4 before this event occurs, and a particularly unlucky individual could potentially end up doing this for ever, so the support is $\{4, 5, 6, \ldots, \infty\}$.
- The name of the first player in the Collingwood Football Club to kick a goal in the last round of the Australian Football League 2017 season. Since at the time of writing, the 2017 AFL season was well underway, the support for this would be the list of players on the roster: $\{\text{Travis Cloke}, \text{Dane Swan}, \text{Scott Pendlebury}, \ldots\}$. However one may find it practical to handle data by assigning integers to these names (basically,

[1] I apologize that in some of these examples, the events in question are in the past (and hence they are not really random anymore, their value has been *realized*). Please put yourselves in my *ex-ante* shoes of August 2017, when I was writing these examples.

computers like numbers more than strings): $\{1 = \text{Travis Cloke}, 2 = \text{Dane Swan}, 3 = \text{Scott Pendlebury}, \ldots\}$. Additionally, since it is possible that Collingwood will have a particularly terrible game, one should also assign "nobody" to this support.

1.2.2 Continuous random variables

Continuous random variables have interval supports (or collections of intervals). Note that all of the above examples of discrete random variables fail this test. Examples of continuous random numbers include:

- The total precipitation in Toledo, OH between Jan 1st 2018 and Dec 31st 2018, in millimeters.
- The time between now and when the next asteroid hits earth.
- Your bank balance (note that strictly speaking, this is discrete random variable, because it is an integer multiple of \$0.01. however at some point it is reasonable to claim that a variable is approximately continuous and hence can be treated as continuous).

1.3 Describing one random variable

If all we had in our toolbox was "discrete" and "continuous", we would not be able to describe random variables very well. Fortunately, we can do much better than this. To begin with, there is the cumulative density function, which completely captures anything you may want to know about a single random variable. If we know that the variable is either continuous or discrete, we can use either a probability mass function or probability density function, which also characterize the variable completely (once we know that it is either continuous or discrete). Finally, we can summarize particular aspects of the random variable with quantities such as mean, variance, median, etc.. These quantities don't fully characterize the distribution, but are sometimes the most important quantities for our analysis.

1.3.1 Cumulative density function

Suppose that a random variable X has support S, which is a subset of the real number line (formally: $S \subseteq \mathbb{R}$). For any particular value of $x \in \mathbb{R}$ (i.e. pick any x on the real number line), it must be that either $X \leq x$, or $X > x$. Hence, no matter whether X is discrete or continuous, we can report the probability that X is less than or equal to any particular value of x on the real number line. Hence, we define the cumulative density function (cdf) of X as follows:

$$F_X(x) = \Pr(X \leq x) \tag{1.8}$$

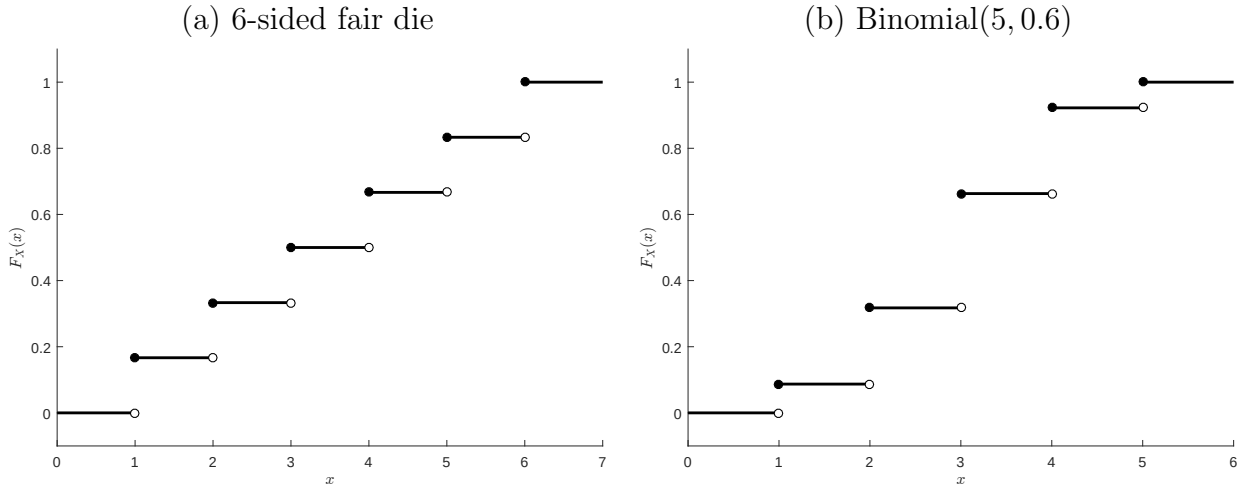


Figure 1.2: Cumulative density functions for some discrete random variables.

Note that the cdf is a function of x, a particular value, and not the random variable itself. Since $\Pr(x \leq X)$ is something that we can compute for *any* $x \in \mathbb{R}$, we must make sure to specify it for the whole real number line, and not just the support of X. For example, if U is a standard uniform random variable (i.e. U is equally likely to be drawn anywhere on the unit interval), then the support of X is the unit interval $(0, 1)$. However we can still assign a probability to U being (say) less than zero, or less than three (which would be equal to 0 and 1 respectively). This cdf would therefore be:

$$F_U(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x \geq 1 \end{cases} \tag{1.9}$$

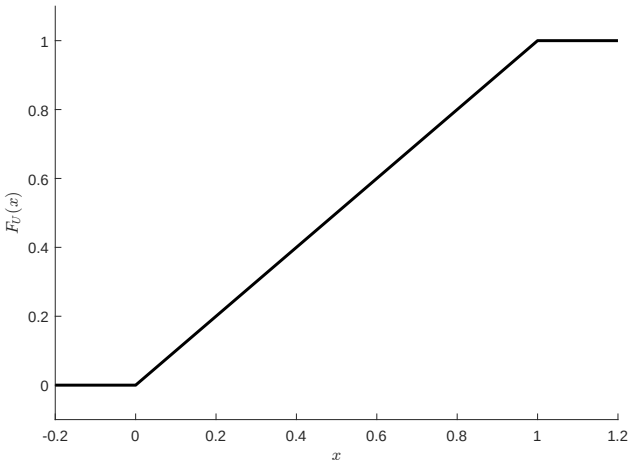


Figure 1.1: Cumulative density function of standard uniform random variable

This is shown graphically in Figure 1.1.

For discrete random variables, the cdf is defined exactly the same, but we need to take special care of the inequality. For example, consider a 6-sided fair die roll. The support of this random variable is $\{1, 2, 3, 4, 5, 6\}$, the probability of rolling any of these is $\frac{1}{6}$, but the probability of getting anything other than these is zero. Therefore, for example, the probabilities of rolling a number less than or equal to 3.01, π, 3.6, and 3.99 are all the same (i.e. they are all equal to $\frac{1}{2}$). Then, as the function gets to $x = 4$, it jumps up to $\frac{2}{3}$. Therefore, at every x in the support of a discrete random variable, the cdf jumps up, and

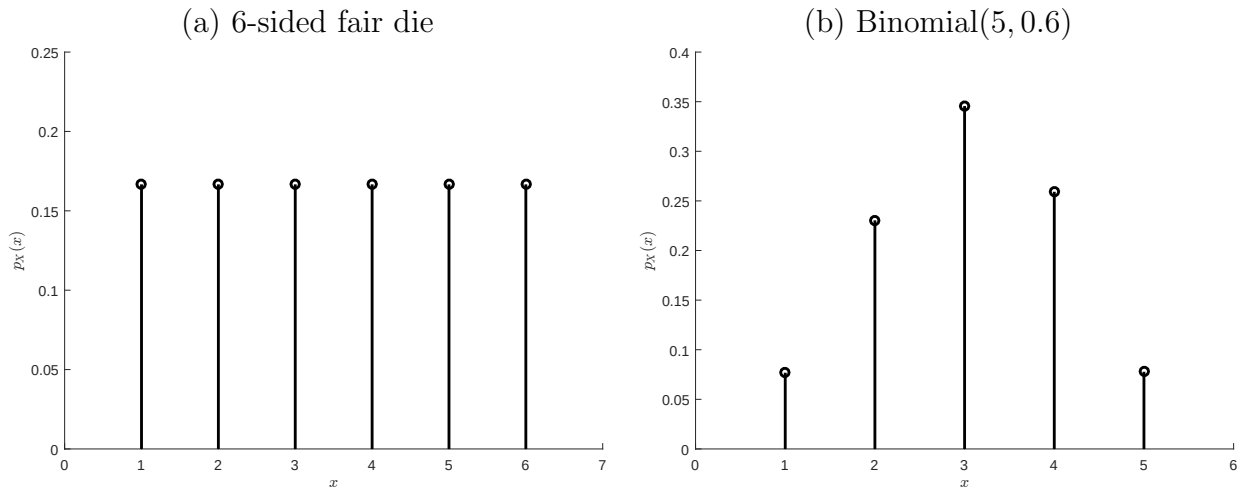


Figure 1.3: probability mass functions for some discrete random variables.

it is flat everywhere else. For example, Figure 1.2a shows the cdf of a fair, 6-sided die roll. Figure 1.2b shows the cdf of the Binomial$(5, 0.6)$ distribution, which can be constructed by flipping five coins coins, each with a probability of 0.6 of coming up heads, and then counting the number of heads.

1.3.2 Probability mass function

We can describe discrete random variables using a *probability mass function* (pmf). These take a number on the real number line, and return the probability that the random variable is equal to it. Going back to our fair die and Binomial examples in Figure 1.2, the pmf of these are:

$$\text{Fair die roll}: \quad p(x) = \begin{cases} \frac{1}{6} & \text{if } x \in \{1,2,3,4,5,6\} \\ 0 & \text{otherwise} \end{cases} \tag{1.10}$$

$$\text{Binomial}(5, 0.6): \quad p(x) = \begin{cases} \frac{5!}{x!(5-x)!} 0.6^x 0.4^{5-x} & \text{if } x \in \{0,1,2,3,4,5\} \\ 0 & \text{otherwise} \end{cases} \tag{1.11}$$

These are shown graphically in Figure 1.3. Note that we can find the height of the cdf at the values in the support by adding up all of the values of the pmf between $-\infty$ and x.

Any pmf $p(x)$ must only return non-negative numbers (because negative probability does not make sense), and must sum to 1 (because this is the probability of drawing an x inside the support).

1.3.3 Probability density function

We cannot use a pmf to describe continuous random variables. To see this, note that for a continuous random variable X, the probability that X is equal to a particular value is zero.

For example, the probability that we will get *exactly* half an inch of rain tomorrow is zero. Not because half an inch of rain is not in the support of rainfall that we could get tomorrow, but because rain does not fall in discrete chunks. Instead, we use a probability *density* function (pdf) to describe how likely drawing particular values are. If we integrate this thing over a region, we get the probability that the random variable is drawn within this region. For example, while the probability of exactly half an inch of rain is zero, the probability of getting between 1/4 and 3/4 inches of rain is not, and also quite a meaningful number (and useful, depending on how much you care about rainfall). The pdf $f_X(x)$ therefore has the following properties:

$$\Pr[X \in (a,b)] = \int_a^b f_X(x)\mathrm{d}x \tag{1.12}$$

$$\Pr[X \leq x] = \int_{-\infty}^{x} f_X(\tilde{x})\mathrm{d}\tilde{x} = F_x(x) \tag{1.13}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}F_x(x) = f_x(x) \tag{1.14}$$

While the 2nd and 3rd lines of equations here are implied by the first, I feel that they are worth pointing out: know how to go between pdf and cdf, and know how they relate to the pmf of a discrete variable. Like pmfs, and pdf must never return negative numbers, and must integrate to 1.

1.3.4 Mean, variance

While a cdf, pmf, or pdf will completely characterize a distribution, they sometimes require a bit of work to find the economically relevant values associated with this distribution. For example, a risk-neutral person cares only about the expected value of a distribution over money, and hence what we would really want to know is:

Definition 1. *The* **mean** *(alternatively* **expected value** *or* **expectation***) of random variable X with support S_X and cdf $F_X(x)$ is equal to:*

$$E[X] \equiv \int_{S_X} x\mathrm{d}F_X(x) \tag{1.15}$$

If X is a continuous random variable with pdf $f_x(x) = F'_X(x)$, then (1.15) can be expressed as:[2]

$$E[X] = \int_{S_X} x f_x(x)\mathrm{d}x \tag{1.16}$$

If X is a discrete random variable with pmf $p_x(x)$, then (1.15) can be expressed as:

$$E[X] = \sum_{x \in S_X} x p_x(x) \tag{1.17}$$

[2]If you are struggling to see this step, note the following for a continuous random variable: $\frac{\mathrm{d}F_X(x)}{\mathrm{d}x} = f_X(x)$. Then, multiplying both sides by a fancy $1 = \frac{\mathrm{d}x}{\mathrm{d}x}$ yields $\mathrm{d}F_X(x) = f_X(x)\mathrm{d}x$.

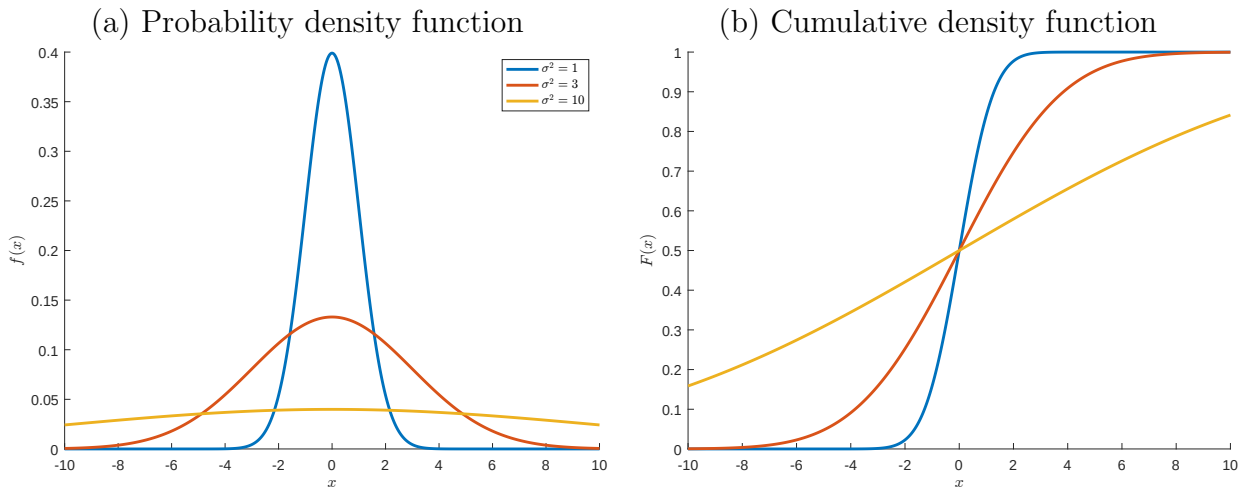


Figure 1.4: Normal distributions with different variances

Hence, for the standard uniform random variable (see (1.9)):

$$E[U] = \int_0^1 x \times 1 \mathrm{d}x = \frac{1}{2}x\Big|_0^1 = \frac{1}{2} - 0 = \frac{1}{2} \tag{1.18}$$

and for a fair die roll (see (1.10)):

$$E[X] = \sum_{k=1}^{6} k\frac{1}{6} = \frac{7 \times 3}{6} = 3.5 \tag{1.19}$$

A useful property of means is that one can add them up. For example, if we wanted to determine the expected value of the sum of two fair die rolls, say X_1 and X_2, then we could use our answer in (1.19) as follows:

$$E[X_1] = E[X_2] = 3.5 \implies E[X_1 + X_2] = 3.5 + 3.5 = 7 \tag{1.20}$$

This also means that if $Y = cX$ for some constant $c \in \mathbb{R}$, then $E[Y] = E[cX] = cE[X]$. However we need to be careful about non-linear functions of random variables. If $h(x)$ is a non-linear function, then in general $E[h(X)] \neq h(E[X])$.

The mean gives us an idea of what we might, quite literally, "expect" X to be. However it gives us no idea about how likely we are to be "close" to this value. For example, Figure 1.4 shows the pdf and cdf of three distributions, all have the same mean, but some are more spread out than others. One measure of this is:

Definition 2. *The* **variance** *of random variable X is equal to:*

$$V[X] \equiv E\left[(X - E[X])^2\right] \tag{1.21}$$

In words, $V[X]$ is X's "expected squared distance" from its mean. For example, for the uniform distribution in (1.9), the variance of X is:

$$V[X] = \int_{S_X} (x - E[X])^2 \mathrm{d}F_X(x) \tag{1.22}$$

$$= \int_0^1 \left(x - \frac{1}{2}\right)^2 \times 1 \mathrm{d}x \tag{1.23}$$

$$= \int_0^1 \left(x^2 - x + \frac{1}{4}\right) \mathrm{d}x \tag{1.24}$$

$$= \frac{1}{3}x^3 - \frac{1}{2}x^2 + \frac{1}{4}x \Big|_0^1 \tag{1.25}$$

$$= \frac{1}{3} - \frac{1}{2} + \frac{1}{4} = \frac{4 - 6 + 3}{12} = \frac{1}{12} \tag{1.26}$$

One can use our knowledge of expectations to further simplify (1.21) as follows:

$$V[X] = E\left[(X - E[X])^2\right] \tag{1.27}$$

$$= E\left[X^2 - 2XE[X] + E[X]^2\right] \tag{1.28}$$

$$= E[X^2] - E\left[2XE[X]\right] + E\left[E[X]^2\right] \tag{1.29}$$

$$= E[X^2] - 2E[X]^2 + E[X]^2 \tag{1.30}$$

$$= E[X^2] - E[X]^2 \tag{1.31}$$

where the 2nd row expands the squared term, the third recognizes that this is the expectation of the sum of some random variables, and the fourth recognizes that 2 and $E[X]$ are constants. Since we have to compute $E[X]$ to get to $V[X]$ anyway, it is sometimes easier to compute $E[X^2]$ first, rather than $E[(X - E[X])^2]$ directly. For example, with the fair die roll:

$$E[X^2] = \sum_{k=1}^{6} k^2 \frac{1}{6} = \frac{1 + 4 + 9 + 16 + 25 + 36}{6} = \frac{91}{6} \tag{1.32}$$

$$V[X] = \frac{91}{6} - \left(\frac{21}{6}\right)^2 = \frac{546 - 441}{36} = \frac{105}{36} \approx 2.92 \tag{1.33}$$

Note that variance and expectation are indifferent units. For example, if X is the height of a human in meters, then $E[x]$ has units of meters, and $V[X]$ is in square meters, an area! To express spread in the same units and the mean, we therefore sometimes take the square root of this, which we call standard deviation.

Like means, we can add the variances of two random variables, but only if they are not

correlated. To see this, we go back to our definition of variance:

$$
\begin{aligned}
V[X+Y] &= E\left[(X+Y-E[X+Y])^2\right] && (1.34)\\
&= E\left[(X+Y-E[X]-E[Y])^2\right] && (1.35)\\
&= E\left[((X-E[X])+(Y-E[Y]))^2\right] && (1.36)\\
&= E\left[(X-E[X])^2+(Y-E[Y])^2+2(X-E[X])(Y-E[Y])\right] && (1.37)\\
&= E\left[(X-E[X])^2\right]+E\left[(Y-E[Y])^2\right]+2E\left[(X-E[X])(Y-E[Y])\right] && (1.38)\\
&= V[X]+V[Y]+\underbrace{2E\left[(X-E[X])(Y-E[Y])\right]}_{\mathrm{cov}(X,Y)} && (1.39)
\end{aligned}
$$

Where the last term $E\left[(X-E[X])(Y-E[Y])\right] = \mathrm{cov}(X,Y)$ is the *covariance* of X and Y. This is a measure of how much X and Y move together in a linear way. Loosely, if $\mathrm{cov}(X,Y) > 0$, then a particularly large X means that Y is also likely to be large; conversely, $\mathrm{cov}(X,Y) < 0$ tells us that a particularly large X means that Y is likely to be small. There are many cases in econometrics where we assume (perhaps to our own peril) that a covariance is zero. In fact, a lot of this course will be devoted to what goes wrong when $\mathrm{cov}(X,Y) \neq 0$. When working through a derivation, therefore, please think carefully about why this thing migh or might not be equal to zero. Ideally, have a good story to back up your decision!

This leads us perfectly in to ...

1.4 Describing the relationship between two or more random variables

If our toolbox could only analyze one random variable at a time, econometrics would not be very interesting.Most of the empirical questions in economics boil down to "what is the causal effect of X on Y". So we had better have a way of describing the relationship between (at least) two random variables. While we are mostly interested in linear correlations, this by no means is the be all and end all of the way X and Y could be related to each other. This section shall proceed with describing the relationship between two random variables, X and Y; however all of this generalizes reasonably easily to more than two random variables.

Up to this point, we have been describing *marginal* probability density/mass functions, *marginal* expectations, and *marginal* variances. For example, $E[Y]$ tells us our expected value of Y, *if we were to have absolutely no information* about Y. For example, Y might be the height of a newborn baby. $E[Y]$ would give us a point prediction of this. Could we do better if we observed something else? Almost certainly yes![3] Suppose that we observed X, the height of the baby's mother. Then we could incorporate this information into our expectation, which we will notate as: $E[Y \mid X]$. This is called a *conditional* expectation, or more precisely: the expectation of Y conditional on X. If X tells us nothing about Y, then the conditional and unconditional expectations are equal: $E[Y \mid X] = E[Y]$. On the other hand, if X helps us improve this point prediction, then $E[Y \mid X] \neq E[Y]$.

[3] In fact, we *can't* do any worse: we can always use $E[Y]$.

1.4.1 Joint distribution functions

But to fully understand the relationship between two random variables, we need to look at things that fully characterize their joint distribution. For any two random variables, we can always use a joint cdf, which tells us the probability that both X and Y are below particular values:

$$F_{X,Y}(x,y) = \Pr[(X \leq x) \cap (Y \leq y)] \tag{1.40}$$

where "$\cap$" is the set notation for "intersection", meaning that we are asking when both $X \leq x$ *and* $Y \leq y$. $F_{X,Y}(x,y)$ is referred to as the multivariate (or joint) cumulative density (or distribution) function. This thing has some analogous properties to single-variable cdfs introduced earlier:

- $F_{x,y}(x,y) \to 0$ as x and y both $\to -\infty$. That is, the probability of X and Y being less than arbitrarily large negative numbers is zero.
- By the same reasoning: $F_{x,y}(x,y) \to 1$ as x and y both $\to \infty$
- For any x, y, x', y' such that $x' \geq x$ and $y' \geq y$, $F_{X,Y}(x',y') \geq F_{X,Y}(x,y)$. That is, if you increase any of the functions arguments, then you are relaxing the requirements for X and Y to be less than their specified values.

In addition to these:

$$\lim_{y\to\infty} F_{X,Y}(x,y) = F_X(x), \quad \lim_{x\to\infty} F_{X,Y}(x,y) = F_Y(y) \tag{1.41}$$

That is, if you make one of these cutoffs arbitrarily large, then the random variable corresponding to that cutoff will almost certainly be below it, hence all that is left to check is whether the other random variable is less than its cutoff, which is the same criterion for the univariate cdf introduced earlier.

From here, we can define joint pdfs and pmfs analogously. For continuous variables, the joint pdf is:

$$f_{X,Y}(x,y) = \frac{\mathrm{d}^2}{\mathrm{d}x\mathrm{d}y} F_{X,Y}(x,y) \tag{1.42}$$

and for discrete random variables, the joint pmf is:

$$p_{X,Y}(x,y) = \Pr[(X = x) \cap (Y = y)] \tag{1.43}$$

To get the marginal (univariate) pdf (cdf) of X, we integrate (sum) out Y:

$$f_X(x) = \int_{S_Y} f_{X,Y}(x,y)\mathrm{d}y \tag{1.44}$$

The relationship between these quantities is shown in Figure 1.5.

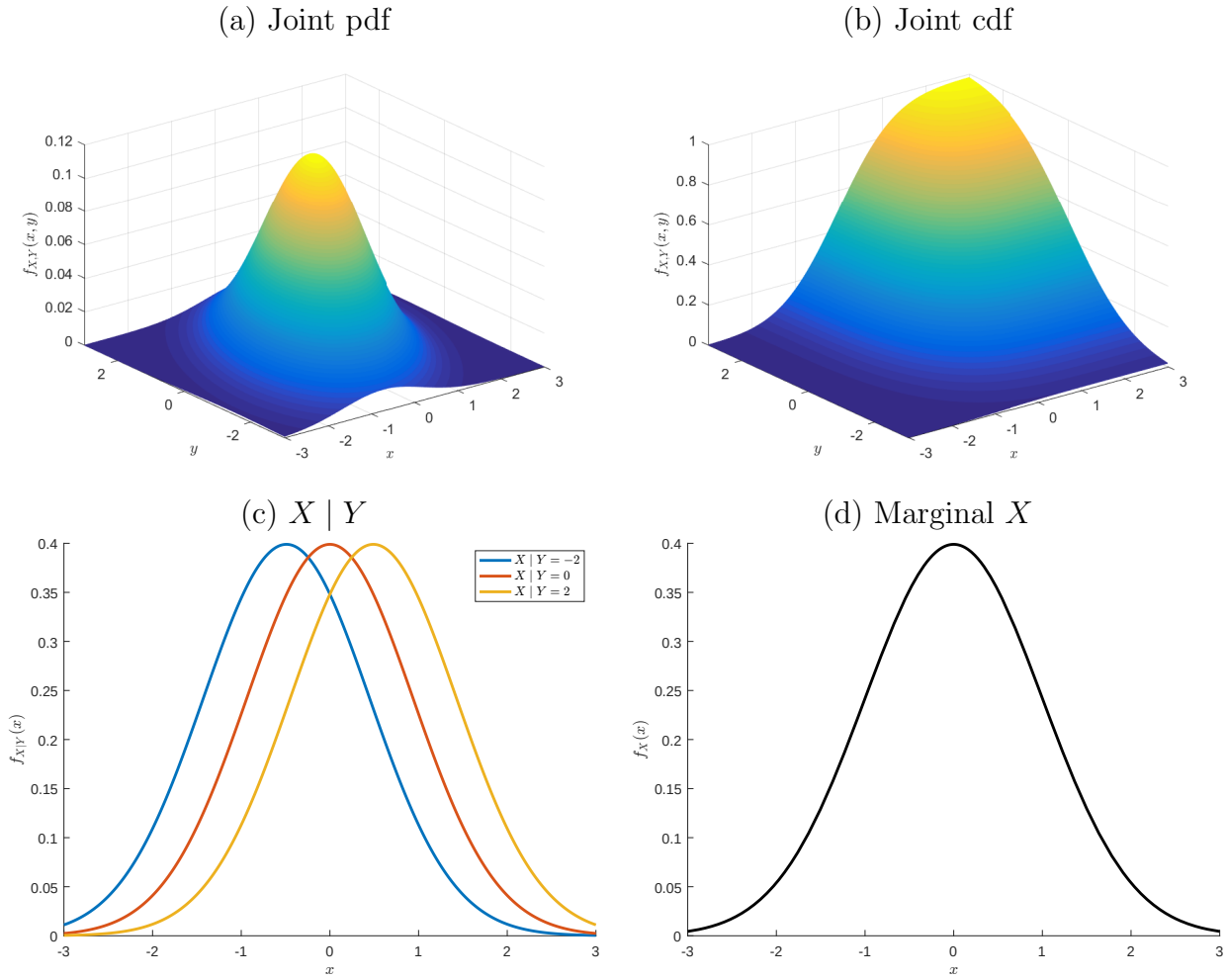


Figure 1.5: Properties of the multivariate normal distribution with $E[X] = E[Y] = 0$, $V[X] = 1$, $V[Y] = 0.7$, $\text{corr}(X, Y) = 0.7$

1.4.2 Conditional probability

So things like $p_{X,Y}(x, y)$ can tell us the likelihood of paired events (x and y) occurring. What if we already knew that one of them was. Could we refine our idea about the distribution of the other? Yes! This is where we introduce Bayes' Theorem. It tells us how we should incorporate some information Y about our beliefs (i.e. distribution) about some other variable X. Given a joint pdf $f_{X,Y}(x, y)$, suppose that we know that X takes on a particular value x', then we can use the two following observations:

1. X can now be treated as a constant, because we know that $X = x'$, and
2. Unless Y is a deterministic function of Y, then there is still some uncertainty about X.

These observations mean that the density of Y conditional on $X = x'$ must be proportional to $f_{X,Y}(x', y)$. Alternatively put, this density must have the same shape as the cross-section of joint pdf that we would slice out if we took a machete to the $y = y'$ plane of the joint pdf plot. But once we know the pdf of something is proportional to something, then we can work out the actual pdf becauese it must integrate to one. Hence:

$$f_{Y|X}(y; x) \propto f_{X,Y}(x, y) \tag{1.45}$$

$$\implies f_{Y|x}(y; x) = \frac{f_{X,Y}(x, y)}{\int_{S_X} f_{X,Y}(x, y)\mathrm{d}y} \tag{1.46}$$

The above ramblings were formalized much more eloquently by Thomas Bayes in the 1700s:

Theorem 1 (Bayes' Theorem)**.** *Let X and Y be random variables, and $p(X)$ and $p(Y)$ denote the marginal probability (density) of events X and Y occurring respectively, and denote the probability of X (Y) occurring* **conditional** *on a particular realization of Y (X) as $p(X \mid Y)$ ($p(Y \mid X)$), then:*

$$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)} \tag{1.47}$$

In terms of a joint pdf, this equation becomes:

$$f_{X|Y}(x; y) = \frac{f_{Y|X}(y; x)f_Y(y)}{f_X(x)} \tag{1.48}$$

Example: At this point, you would probably like to see an example, so here one is. Suppose that in a population, 1/3 of people have a particular disease. There is a test for the disease, but it is not perfect. If the person has the disease, then it returns a "positive" result with probability 5/6. If the person does not have the disease, then it returns a positive result with probability 2/3. Hence, the test is more likely to return a positive result if the person has the disease, but there will be some people without the disease who get a positive result

(i.e. false positive), and some people with the disease who do not get a positive result (i.e. false negative). *What is the probability that a person has the disease if they received a positive test for the disease?* In the notation of (1.47), let $P = 1$ if the person received a positive test result, $P = 0$ otherwise, and $D = 1$ if the person has the disease, $D = 0$ otherwise. We need to compute the conditional probability $p(D = 1 \mid P = 1)$. The probability that a person has the disease, conditional on receiving a positive test result. The above description gives us the following:

- $p(P = 1 \mid D = 1) = 5/6$ i.e. if a person has the disease, they test positive with probability $5/6$
- $p(P = 1 \mid D = 0) = 1/3$ i.e. if a person does not have the disease, they test positive with probability $1/3$
- $p(D = 1) = 1/3$, the fraction of people who have the disease.
- $p(D = 0) = 2/3$, the fraction of people who do not have the disease.

We can use Bayes' Theorem to calculate $p(D = 1 \mid P = 1)$:

$$p(D = 1 \mid P = 1) = \frac{p(P = 1 \mid D = 1)p(D = 1)}{p(P = 1)} \tag{1.49}$$

We know everything on the right-hand side of this except for the denominator, which is equal to the probability someone tests positive for the disease, without knowing whether or not they have it. There are (at least) two solutions to this. The first is to explicitly compute it:

$$p(P = 1) = p(P = 1 \mid D = 1)p(D = 1) + p(P = 1 \mid D = 0)p(D = 0) \tag{1.50}$$

$$= \frac{5}{6} \times \frac{1}{3} + \frac{1}{3} \times \frac{2}{3} = \frac{5 + 4}{18} = \frac{9}{18} = \frac{1}{2} \tag{1.51}$$

OK, so if we test the entire population, 50% of people will be testing positive. If that seems worrysome to you, then good! We can now substitute the other things we know into our equation:

$$p(D = 1 \mid P = 1) = \frac{5/6 \times 1/3}{0.5} = \frac{5}{18} \times 2 = \frac{5}{9} \approx 0.56 \tag{1.52}$$

So a little over half the people who test positive will have the disease. The worrying part is that a little under half of the people who test positive will *not* have the disease. Depending on if there is any stigma associated with the disease, it may not be a good idea to test everyone. A more palatable solution for this would be to only test people who are suspected (either by themselves or their doctor) of having the disease. Note that this would be represented in our problem as an increase in $p(D = 1)$, and hence a decrease in $p(D = 0)$.

The other method of solving this (alluded to above) is to recognize that $p(D = 1 \mid P = 1) = 1 - p(D = 0 \mid P = 1)$. If we take the ratio of these two conditional probabilities, the expression simplifies to something without $p(P = 1)$:

$$\frac{p(D = 1 \mid P = 1)}{p(D = 0 \mid P = 1)} = \frac{p(P = 1 \mid D = 1)p(D = 1)}{p(P = 1)} \times \frac{p(P = 1)}{p(P = 1 \mid D = 0)p(D = 1)} \tag{1.53}$$

$$= \frac{p(P = 1 \mid D = 1)}{p(P = 1 \mid D = 0)p(D = 1)} \tag{1.54}$$

$$= \frac{5/6 \times 1/3}{1/3 \times 2/3} = 5/4 = 1.25 \tag{1.55}$$

This is (almost) the answer expressed in odds ratio form: people in the group who tested positive are 1.25 times as likely to have the disease than not. But these fractions need to add to 1, so letting $q = p(D = 1 \mid P = 1)$:

$$\frac{q}{1 - q} = 5/4, \quad q = 5/9 \tag{1.56}$$

Exercises

Exercise 1.1.
Let X be the sum of two fair, four-sided die rolls. That is, each die has four faces, with numbers 1, 2, 3, 4.

1. What is the support of X?
2. Is X a discrete or continuous random variable? Explain
3. Based on your answer to question 2, construct the pdf or pmf of X.
4. Construct the cdf of X
5. Compute $E[X]$ and $V[X]$.

Exercise 1.2.
The exponential distribution has probability density function:

$$f_X(x) = \begin{cases} c \exp(-\lambda x) & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.57}$$

where c is a positive constant, and λ is a scale parameter.

1. What is the support of X?
2. What must the positive constant c be equal to?

3. What is the cdf of X?

4. Determine $E[X]$ and $V[X]$. To do this, you will need to use integration by parts:

$$\int_a^b u'(x)v(x)\mathrm{d}x = [u(x)v(x)]_a^b - \int_a^b u(x)v'(x)\mathrm{d}x \tag{1.58}$$

where $u(x)$ an $v(x)$ are both differentiable functions. You won't need to remember this, because you can always work it out from the product rule:

$$[u(x)v(x)]' = u'(x)v(x) + v'(x)u(x) \tag{1.59}$$
$$u'(x)v(x) = [u(x)v(x)]' - v'(x)u(x) \tag{1.60}$$

then just integrate both sides.

Exercise 1.3.
The cumulative density function for random variable X is:

$$F(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x^\alpha & \text{if } 0 < x < 1 \\ 1 & \text{if } x \geq 1 \end{cases}$$

where $\alpha > 0$ is a parameter of the distribution.

1. What is the support of X, and is X a discrete or continuous random variable?

2. Calculate the pdf, $f(x)$

3. Calculate $E[X]$ and $V[X]$.

Exercise 1.4.
An unfair coin has a probability of $\frac{1}{3}$ coming up heads, tails otherwise. You keep flipping it until the first time it comes up heads. Let X be the number of times you have to flip the coin.

1. What is the support of X?

2. What is the probability that $X = 10$? *Hint: Since each coin flip is an independent event you can multiply the probability of each coin flip that needs to occur together. For example, the probability that $X = 4$ is equal to:*

$$\begin{aligned} &\Pr[\text{1st flip is tails}] \times \Pr[\text{3nd flip is tails}] \times \Pr[\text{3rd flip is tails}] \times \Pr[\text{4th flip is heads}] \\ &= \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} \times \frac{1}{3} = \frac{2^3}{3^4} \end{aligned}$$

3. What is the probability that $X \leq 4$?

4. What is the probability mass function for X?

5. Verify that your pmf sums to 1. You can use the results that for $-1 < p < 1$:

$$\sum_{n=0}^{\infty} p^n = \frac{1}{1-p}, \quad \text{and } \sum_{n=1}^{\infty} p^n = 1 + \sum_{n=0}^{\infty} p^n$$

Exercise 1.5.
Let X be the sum of two fair, four-sided die rolls. That is, each die has four faces, with numbers 1, 2, 3, and 4. For the purposes of this exercise, let Z_1 and Z_2 be the die rolls themselves, hence $X = Z_1 + Z_2$.

1. What is the pmf/pdf of the distribution of X, given that X is an even number?

2. Given that the first die roll was a 2, what is the expected value of X?

3. What is the variance of X, given that $Z_2 = 3$?

4. What is the expected value of X, given that $Z_2 \leq 3$?

Part I

Estimating one parameter

Chapter 2

Estimators

An *estimator* is a mathematical function that takes data and gives you an *estimate* of something. While this course will mainly focus on numerical examples, I would like you to remember that we take in information, and use this information to make educated guesses about things *all the time*. For example, before I go grocery shopping, I have a peek into the fridge and decide how much food I will need to buy this week: an estimate. When I drive to work, I pay attention to the traffic conditions (i.e. data), in part so I have an estimate of how fast I should be driving. In what will follow, our analysis will look somewhat more formal than this, but be mindful that there are similarities: (i) we gather information and use it to make a prediction or guess about something, (ii) there are some ways of using the information that are more useful than others, (iii) some types of information are better than other types, (iv) if we know more about how we are gathering our information, we can sometimes use this to make a better guess, and (v) more information usually makes our guess more accurate.

2.1 Populations and samples

The Frequentist approach in econometrics[1] starts with the premise that there is a *population parameter* (or collection of parameters, the distinction is not important at all), say θ, that determines how we observe data. We observe a *sample* of data, say $\{x_i\}_{i=1}^N$, and use this sample to produce an *estimate* of θ, the property of the population that we would like to know about. Our prime objective in econometrics is to estimate properties of the population, using a sample and an estimator.

As an example, suppose that you wish to estimate the probability that tossing a particular coin results in it landing heads up.[2] You decide to model the data-generating process as

[1] Statistics and econometrics can be divided into two philosophies: Frequentist and Bayesian. This is a course in Frequentist econometrics. For a good reference on Bayesian econometrics, see Koop et al. (2007). Whenever econometrics is mentioned without clarifying whether it is Frequentist or Bayesian, it is usually safe to assume Frequentist.

[2] It is at this point that I feel some need to apologize for a seemingly endless string of examples involving coin flipping. There are many reasons for my choice of this type of example. Most importantly, coin-flip

follows:

$$H_i = \begin{cases} 1 & \text{if coin flip } i \text{ lands heads up} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

$$\Pr[H_i = 1] = \theta, \quad \theta \in [0, 1] \tag{2.2}$$

$$H_i \sim iid\text{Bernoulli}(\theta) \tag{2.3}$$

Here (2.1) defines a random variable that can take on two values, 0 and 1 (we implicitly assume that the coin has exactly two sides, and so $H_i \neq 1 \iff H_i = 0 \iff$ tails). (2.2) tells us that the probability of the coin flip coming up heads is equal to θ: this is the population parameter that we want to estimate. (2.3) formalizes this further, by (i) using the formal name for a coin-flip variable (Bernoulli), and (ii) formally stating the assumption that each coin flip is an independent draw from the same distribution.[3] These are all statements about the population.

Now we collect a *sample* from this population. In this example, this could involve flipping the coin (say) 100 times, and recording the result of each coin flip. Let's denote this sample as $\{h_i\}_{i=1}^{100}$. This last bit of notation denotes the collection of coin flip outcomes for 100 coin flips. We could write this out long-hand as:

$$\{h_i\}_{i=1}^{100} = \{h_1, h_2, h_3, \ldots, h_{99}, h_{100}\} \tag{2.4}$$

but we have better things to do (or at least I do).

At this point (and forever into the future) it is very important to be clear about when we are talking about properties of the sample and properties of the population. We will be using the former to tell us something about the latter, but they are two different things. $\{h_i\}_{i=1}^{N}$, the sample, is the thing we will be importing into our statistical package, then calculating means, variances, etc. of. We know the sample mean *because we can calculate it*. We can never know the population mean: that's why the sample is useful!

2.2 Estimators and the sampling distribution

We now take our sample, and stick it into our estimator. Out comes an estimate. Here's the thing: our sample is random. If we put something random into a function, in general we should expect to get something random out. In the context of our coin-flipping example starting in Section 2.1, we have a sample of 100 coin flips, and wish to use this to estimate

random variables (formally: *Bernoulli* random variables), are very simple to understand. Because of this, they allow for introduction of simple concepts, without the need for you to get your head around anything else that could be complicated. Additionally, coin-flip variables show up *everywhere*: get used to it.

[3]The "independent" part of this means (among other things) that if I toss two coins, knowing the outcome of one tells me *nothing* about whether the other outcome is heads or tails. While for practical reasons we might not give a hoot about whether the H_is are independent, we usually need to make an assumption about this when we estimate things. It is best to state it (and any other assumption we make) formally. That way, it is more obvious when we are doing something stupid.

θ, the probability that our coin comes up heads. For a lot of reasons that we will get to later on, a good estimator to use in this situation is:

$$\hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} h_i \tag{2.5}$$

which happens to be the sample mean.[4] Although it is not stated explicitly on the left-hand side of (2.5), $\hat{\theta}$ is a function of the sample: $\hat{\theta} = f\left(\{h_i\}_{i=1}^{N}\right) = \frac{1}{N}\sum_{i=1}^{N} h_i$, but that is overly cumbersome, so we will stick with the notation in (2.5). We have also gone a bit more general, and written this for an arbitrary sample size N, rather than our $N = 100$ observations in the above example.

Before exploring the properties of $\hat{\theta}$ when $N = 100$, it is instructive to understand how $\hat{\theta}$ behaves with stupidly small samples. Each row of Table 2.1 shows a possible sample that we could have observed, if we only tossed the coin $N = 3$ times. The rightmost column shows the probability of observing that sample, as a function of the population parameter θ. Note that we pay attention to the order of coin flips, and hence we don't think of the sample $\{1, 0, 1\}$ as being the same as $\{1, 1, 0\}$, even though they both have 2 heads and 1 tail. This is an important distinction for later on, but at the moment just be aware that there are $2^3 = 8$ possible samples that we could have observed, with varying probability of being observed. That said, the sample mean doesn't give a hoot about which order in which the heads and tails came, so we can add up the cells in the "Probability" column of this Table to get the probability mass function of the sample mean, as a function of θ:

h_1	h_2	h_3	$\hat{\theta}$	Probability
0	0	0	0	$(1-\theta)^3$
0	0	1	1/3	$\theta(1-\theta)^2$
0	1	0	1/3	$\theta(1-\theta)^2$
0	1	1	2/3	$\theta^2(1-\theta)$
1	0	0	1/3	$\theta(1-\theta)^2$
1	0	1	2/3	$\theta^2(1-\theta)$
1	1	0	2/3	$\theta^2(1-\theta)$
1	1	1	1	θ^3

Table 2.1: Sampling distribution of $\hat{\theta}$ when $N = 3$

$$\Pr[\hat{\theta} = x] = \begin{cases} (1-\theta)^3 & \text{if } x = 0 \\ 3\theta(1-\theta)^2 & \text{if } x = 1/3 \\ 3\theta^2(1-\theta) & \text{if } x = 2/3 \\ \theta^3 & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

(2.6) characterizes the distribution of our estimator $\hat{\theta}$, as a function of the population parameter θ. We call this the *sampling distribution* of $\hat{\theta}$.

If one can see the matrix when it comes to probability mass functions, one may realize

[4] ...and here is one of the first good reasons to use this: we know a lot about sample means, so we can use them to derive properties of this estimator. More on this later.

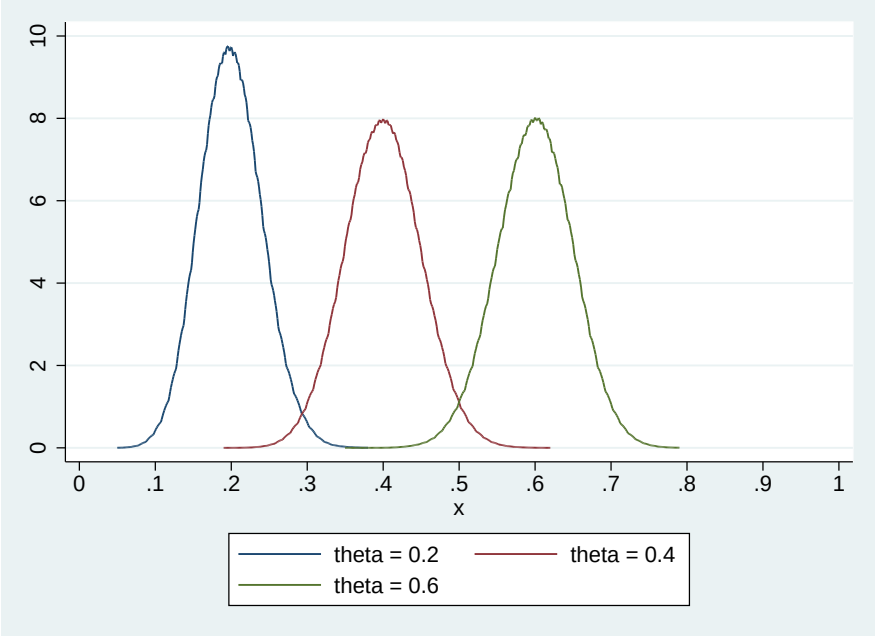


Figure 2.1: Sampling distribution of Bernoulli random variable, $N = 100$.

this that we can write (2.6) more compactly as:

$$\Pr[\hat{\theta} = x] = \begin{cases} \binom{3}{3x}\theta^{3x}(1-\theta)^{3(1-x)} & \text{if } x \in \{0, 1/3, 2/3, 1\} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

which if you squint hard enough, looks *almost* like the Binomial distribution. In fact, if you substitute in $k = 3x$, this is exactly what you get: the number of heads for this sampling process is distributed Binomial$(3, \theta)$. For a sample size of N, this generalizes to Binomial(N, θ). To illustrate this, Figure 2.1 shows the sampling distribution for the same estimator for a much more reasonable sample of $N = 100$ coin flips. As the sample size gets larger, there are more values that $\hat{\theta}$ can take on. For example, when $N = 4$ we will get one of $\hat{\theta}$ =0, 1/4, 2/4, 3/4, 4/4, and as N gets *really* large, we struggle to see that the distribution is still discrete. However note that since we are always taking a ratio of two integers, the number of heads divided by the sample size, there are some values of $\hat{\theta}$ that we could *never* get: $\frac{\pi}{4}$, for example.

Unfortunately, in general, the sampling distribution of an estimator is not easy to work out. Most of the time, however, we can determine a few properties of this distribution. In particular, we may be interested in knowing the expected value of $\hat{\theta}$ (i.e.: "on average, do I get right number?"), the variance (i.e.: "how precise is my estimator?"), and how these things change with sample size (i.e.: "if my sample size gets bigger, how much better is my estimator?"). We explore some of these properties in the next section.

2.3 Small-sample properties of estimators

While it is usually infeasible to determine the exact sampling distribution of our estimator, we can usually derive, or at least approximate (more on this later), some properties of its distribution. That is, we might not be able to write down the cdf of $\hat{\theta}$, but we may be able to work out a few things, like its mean and variance. This is especially easy when our estimator is a sample mean, because we know a lot about sample means, and is particularly useful when there is more than one estimator that could do the job for you. If there is more than one option, it usually pays to think at least a bit about which one will work best for you.

To illustrate this, suppose for example that a friend of yours was rolling a fair[5] die, and calling out the numbers. Your problem is that you don't know how many sides the die has, and you would like to estimate it. Let η be the number of sides on the die. Your data $\{k_i\}_{i=1}^N$ consists of the outcomes of the N die rolls that your friend has called out.[6] Here are two estimators that you may want to consider:

1. Noting that for an η-sided die, the expected value of a roll is $E[k_i] = \frac{\eta+1}{2}$, you replace $E[k_i]$ with its sample analog, $\bar{k} = \frac{1}{N}\sum_i k_i$ and solve for η:

$$\hat{\eta} = 2\bar{k} - 1, \quad \text{i.e.: } \bar{k} = \frac{\hat{\eta}+1}{2} \tag{2.8}$$

2. Noting that the highest possible value of k_i that you could observe is $k_i = \eta$, you use the maximum:

$$\tilde{\eta} = \max_i\{k_i\} \tag{2.9}$$

Both of these take a property of the population (the mean and maximum respectively), and then use the *sample analog* of this. Unsurprisingly, this is often referred to as an *analogy* estimation strategy.[7] We will be introduced to some important properties of estimators below, in the context of $\hat{\eta}$ and $\tilde{\eta}$. Neither will come out as unambiguously better. Get used to it! If we (economists) assume people can make trade-offs, we'd better be able to make them ourselves. But before getting into this, suppose that you observed the following sample:

$$\{1,1,1,1,1,6\} \implies \hat{\theta} = 2\frac{11}{6} - 1 \approx 2.7, \quad \tilde{\eta} = 6$$

One alarming property of $\hat{\theta}$ (that is not discussed below) is that our estimate of 2.7, even if we round it up to the nearest integer, could not *possibly* be believable, because we observe a 6 in our sample! We would never run into this problem for $\tilde{\theta}$.

[5] That is: each number is equally likely to be the outcome.

[6] At this point, you may be telling me "But James, almost all dice are 6-sided, and there are a few 20-sided dice out there, but it's really hard to get your hands on a 42-sided die. Isn't this some information that we shouldn't be ignoring?" To which my response would be: "Yes, go and learn Bayesian econometrics."

[7] Analogy estimators are reasonably easy to come up with, but there is no guarantee that they have any nice properties. Later on you will learn about maximum likelihood (ML) estimation, which is a systematic way to come up with an estimator that has some very nice properties. Another example of this is a Generalized Method of Moments (GMM) estimator.

2.3.1 Bias

While we have no guarantee that our estimator gives us the right number (i.e. $\hat{\theta} = \theta$) *for sure*, we can assess whether we get the right number *on average*. Specifically, we can compare $E[\hat{\theta}]$ to θ. If they are equal, i.e. $E[\hat{\theta}] = \theta$, then we say that our estimator is *unbiased*. On the other hand, if $E[\hat{\theta}] \neq \theta$, then our estimator is *biased*, and we might want to worry.

Now let's evaluate the properties of our estimators $\hat{\eta}$ and $\tilde{\eta}$ described earlier. For the estimator based on the population mean:

$$\begin{aligned}
E[\hat{\eta}] &= E\left[2\bar{k} - 1\right] && (2.10)\\
&= 2E[\bar{k}] - 1 && (2.11)\\
&= 2E\left[\frac{1}{N}\sum_i k_i\right] - 1 && (2.12)\\
&= \frac{2}{N}\sum_i E[k_i] - 1 && (2.13)\\
&= \frac{2}{N}NE[k_i] - 1 && (2.14)\\
&= 2E[k_i] - 1 && (2.15)\\
&= 2\frac{\eta + 1}{2} - 1 && (2.16)\\
&= \eta && (2.17)
\end{aligned}$$

In short, $E[\hat{\eta}] = \eta \iff \hat{\eta}$ is unbiased. Good! In expectation (loosely: "on average") we get the right number.

Now let's look at the estimator based on the maximum:

$$E[\tilde{\eta}] = E[\max_i k_i] \tag{2.18}$$

This opens up a bit more of a can of worms, because now we need to take an expectation over the maximum of the k_is in our sample. Welcome to the wonderful world of *order statistics*! Specifically, the first order statistic (i.e. the maximum of a sample). We first need to derive the distribution of $\max_i k_i$. Let M be this maximum, to make notation easier. What is the probability that M is equal to a particular value m? That is, what is the pmf of M?

$$\begin{aligned}
p(m) = \Pr[&N-1 \text{ observations are less than or equal to } m\\
&\text{and at least one is equal to } m] && (2.19)\\
= &\binom{N}{1}\left(\frac{m}{\eta}\right)^{N-1}\frac{1}{\eta} = \frac{Nm^{N-1}}{\eta^N} && (2.20)
\end{aligned}$$

if $m = 1, 2, 3, \ldots, \eta$, and zero otherwise. We can now take the expectation of M:

$$E[\tilde{\eta}] = E[M] = \sum_{m=1}^{\eta} \left[m \frac{N m^{N-1}}{\eta^N} \right] \tag{2.21}$$

$$= \frac{N\eta}{N+1} \sum_{m=1}^{\eta} \left[\frac{(N+1)m^{N+1-1}}{\eta^{N+1}} \right] \tag{2.22}$$

$$= \eta \frac{N}{N+1} \tag{2.23}$$

Where the last line follows by noting that the thingy that we are summing is the pmf of M, if we have one extra observation in our sample, and so it must sum to 1. Remember this monkey trick, it will come in handy! Inspection of (2.23) yields some sad news: $\tilde{\eta}$ is biased. On average we will under-estimate η by the fraction $\frac{N}{N+1}$. This should not be too surprising: For any sample size, there is a non-zero probability that the maximum is not equal to η, and so some of the terms in the above expectation calculation put positive weight on outcomes that are less than η. Further inspection of (2.23) and a bit of thinking (!), however, shows that all is not lost. Firstly, as our sample size gets large, $\frac{N}{N+1} \to 1$, and so the bias disappears. This is a common property of many (but by no means all) biased estimators, and may be why we might prefer one to an unbiased estimator if we think our sample size is large enough. Secondly, since the bias is only a function of N, we can easily correct for this by multiplying the maximum by $\frac{N+1}{N}$, that is:

$$\check{\eta} = \frac{N+1}{N} \max_i k_i = \frac{N+1}{N} \tilde{\eta} \tag{2.24}$$

$$E[\check{\eta}] = E\left[\frac{N+1}{N} \tilde{\eta} \right] = \frac{N+1}{N} E[\tilde{\eta}] = \frac{N+1}{N} \frac{N}{N+1} \eta = \eta \tag{2.25}$$

By deriving the bias of this estimator we were not only able to say something about the direction of the bias (i.e. $\tilde{\eta}$ under-estimates the population parameter on average), but we also came up with another one that was unbiased! You should probably remember that.

2.3.2 Variance

If our estimator is unbiased, or at least if the bias is something that we can cope with, the next question we might ask is: how *precise* is our estimator? That is, through the sampling process, do our estimates all fall nice and close to their mean, or are they all over the place? We typically use variance to evaluate this. After checking bias, we found that while $\hat{\eta}$ was the only unbiased estimator in consideration, we could easily correct the bias in $\tilde{\eta}$. Therefore

this should be the next thing to check in our die-rolling example. Firstly, for $\hat{\eta}$:

$$V[\hat{\eta}] = V\left[2\bar{k} - 1\right] \tag{2.26}$$

$$= \frac{4}{N} V[k_i], \qquad \text{(assumed independence here)} \tag{2.27}$$

$$= \frac{4}{N} \frac{\eta^2 - 1}{12} \tag{2.28}$$

$$= \frac{\eta^2 - 1}{3N} \tag{2.29}$$

Inspection of (2.29) tells us a few things. Firstly, if $\eta = 1$, then $V[\hat{\eta}] = 0$. This should not be surprising, but it is comforting: if we have a one-sided "die", then we will always get the same number, and hence have zero variance. Perhaps of more use is that the variance is (i) increasing in η, and (ii) decreasing in N. (ii) is typical of almost anything you will end up using, and loosely can be interpreted as "bigger samples are better".

Now let's move to our second estimator, $\tilde{\eta}$. For reasons that should become obvious after you do this over and over again, we are going to use the relationship $V[X] = E[X^2] - E[X]^2$. In case they are not obvious now, these reasons are (i) we already know $E[X]$, and (ii) it is easier to evaluate $E[X^2]$ on its own than try to do $V[X]$ in one fell swoop.

$$E[\tilde{\eta}^2] = E\left[M^2\right] \tag{2.30}$$

$$= \sum_{m=1}^{\eta} \left[m^2 \frac{N m^{N-1}}{\eta^N} \right] \tag{2.31}$$

$$= \sum_{m=1}^{\eta} \left[\frac{N m^{N+1}}{\eta^N} \right] \tag{2.32}$$

$$= \frac{N\eta^{N+2}}{\eta^N (N+2)} \sum_{m=1}^{N} \left[\frac{(N+2) m^{N+1}}{\eta^{N+2}} \right] \tag{2.33}$$

$$= \eta^2 \frac{N}{N+2} \tag{2.34}$$

$$V[\tilde{\eta}] = \eta^2 \left[\frac{N}{N+2} - \left(\frac{N}{N+1} \right)^2 \right] \tag{2.35}$$

$$= \eta^2 \frac{N^3 + 2N^2 + N - N^3 - 2N^2}{(N+2)(N+1)^2} \tag{2.36}$$

$$= \frac{N\eta^2}{(N+2)(N+1)^2} \tag{2.37}$$

as with $\hat{\eta}$, the variance of $\tilde{\eta}$ decreases as sample size increases. To see this, note that we have N in the numerator, and a cubic in the denominator, so the denominator grows much faster.

But which of $\hat{\eta}$ and $\tilde{\eta}$ is better based on variance? It turns out that for almost all

reasonable values of η and N, $V[\tilde{\eta}] < V[\hat{\eta}]$, which can be shown as follows:

$$\frac{V[\tilde{\eta}]}{V[\hat{\eta}]} = \frac{N\eta^2}{(N+2)(N+1)^2} \times \frac{3N}{\eta^2-1} \tag{2.38}$$

$$= \frac{N\eta^2}{(N+2)(N^2+2N+1)} \times \frac{3N}{\eta^2-1} \tag{2.39}$$

$$= \frac{N\eta^2}{N^3+2N^2+N+2N^2+4N+2} \times \frac{3N}{\eta^2-1} \tag{2.40}$$

$$= \frac{N\eta^2}{N^3+4N^2+5N+2} \times \frac{3N}{\eta^2-1} \tag{2.41}$$

$$= \frac{\eta^2}{N+4+5/N+2/N^2} \times \frac{3}{\eta^2-1} \tag{2.42}$$

which $\to 0$ as $N \to \infty$.[8] Furthermore, the denominator in the first fraction is at least 5,[9] so we can say that:

$$\frac{V[\tilde{\eta}]}{V[\hat{\eta}]} < \frac{3\eta^2}{5(\eta^2-1)} \tag{2.44}$$

and when is this fraction less than one?

$$\frac{3\eta^2}{5(\eta^2-1)} \leq 1 \tag{2.45}$$

$$3\eta^2 \leq 5\eta^2 - 5 \tag{2.46}$$

$$5 \leq 2\eta^2 \tag{2.47}$$

$$\eta \geq 2 \tag{2.48}$$

Note that the *mathematical* solution to (2.47) is $\eta \in (-\infty, -\sqrt{2.5}] \cup [\sqrt{2.5}, \infty)$, however we can discard the negative part of this because our die can only take on positive numbers, and we can round up the lower bound of $\sqrt{2.5}$ to 2 because our die has an integer number of sides. Hence $\eta \geq 2$ is the *econometric* solution to the problem. In short, $\tilde{\eta}$ has a smaller variance than $\hat{\eta}$, as long as we don't have a one-sided die.

2.3.3 Mean squared error

A small variance is a good thing, but only if the estimator's distribution is centered (at least roughly) around the true value. That is, if the estimator is substantially biased, why should

[8] Alternatively, note that:

$$V[\tilde{\eta}] = \frac{N\eta^2}{(N+2)(N+1)^2} < \frac{N\eta^2}{(N+0)(N+0)^2} = \frac{N\eta^2}{N^3} = \frac{\eta^2}{N^2} \tag{2.43}$$

Which is only a little bit more than $3/N \times V[\hat{\eta}] = \frac{\eta^2-1}{N^2}$, so for even very small sample sizes (say $N > 4$), using the maximum is better.

[9] I.e.: ignore the $5/N$ and $2/N^2$ terms, and set $N = 1$.

be care that the variance is small? We shouldn't! To see this, let's construct a silly but illustrative example. Suppose that you can use estimators for population parameter θ with the following sampling distributions:[10]

$$\Pr[\hat{\theta} = x] = \begin{cases} 1 & \text{if } x = 2 \\ 0 & \text{otherwise} \end{cases} \tag{2.49}$$

$$\Pr[\tilde{\theta} = x] = \begin{cases} \frac{1}{5} & \text{if } x \in \{\theta - 2, \theta - 1, \theta, \theta + 1, \theta + 2\} \\ 0 & \text{otherwise} \end{cases} \tag{2.50}$$

The first estimator $\hat{\theta}$ returns an estimate of 2 no matter what data we get. This should look like a silly choice, but $\hat{\theta}$ *does* have the following desirable property: $V[\hat{\theta}] = 0$. If we were to judge estimators based only on their variance, we could do no better than $\hat{\theta}$! The problem with this estimator is that it is biased: $E[\hat{\theta}] = 2 \neq \theta$ (unless the true value θ is also equal to 2, but we can't know that). $\tilde{\theta}$, on the other hand, is unbiased, because:

$$E[\tilde{\theta}] = \sum_{k=-2}^{2} \frac{\theta + k}{5} = \theta \tag{2.51}$$

but has a non-zero (hence realistic) variance of:

$$V[\tilde{\theta}] = \sum_{k=-2}^{2} \frac{1}{5}(\theta + k - \theta)^2 = \frac{1}{5}(2^2 + 1^2 + 0^2 + 1^2 + 2^2) = \frac{10}{2} = 5 \tag{2.52}$$

One useful measure to use in these cases is *mean squared error* (MSE). Unlike variance, which asks how far away (in terms of squared distance) on average is an estimator from its *expected* value, MSE asks how far away our estimator is from its *true* value. Let's put these side-by-side to see the difference:

$$V[\hat{\theta}] = E\left[\left(\hat{\theta} - E[\hat{\theta}]\right)^2\right] \tag{2.53}$$

$$MSE[\hat{\theta}] = E\left[\left(\hat{\theta} - \theta\right)^2\right] \tag{2.54}$$

Comparing (2.53) and (2.54), note the only difference is that for the MSE equation, we replace the expected value of the estimator, $E[\hat{\theta}]$, with population parameter that we are trying to estimate, θ. Hence, these two things will be equal if and only if $E[\hat{\theta}] = \theta$. To see

[10] Note here that I have abstracted away from the sampling process and just written down a probability distribution for each estimator. In the background, there may be a function taking data and returning an estimate, but this is unnecessary for the example. If you prefer, you can think about these as *signals* containing information about θ (which is pretty much what an estimator is, anyway).

the "only if" part of this, we can decompose (2.54) as follows:

$$MSE[\hat{\theta}] = E\left[\left(\hat{\theta} - \theta + E[\hat{\theta}] - E[\hat{\theta}]\right)^2\right] \tag{2.55}$$

$$= E\left[\left(\left(\hat{\theta} - E[\hat{\theta}]\right) + \left(E[\hat{\theta}] - \theta\right)\right)^2\right] \tag{2.56}$$

$$= E\left[\left(\hat{\theta} - E[\hat{\theta}]\right)^2\right] + E\left[\left(E[\hat{\theta}] - \theta\right)^2\right] + 2E\left[\left(\hat{\theta} - E[\hat{\theta}]\right)\left(E[\hat{\theta}] - \theta\right)\right] \tag{2.57}$$

$$= V[\hat{\theta}] + \text{Bias}^2[\hat{\theta}] + 0 \tag{2.58}$$

The third term in (2.57) is equal to zero because $E[\hat{\theta}]$ and θ are constants, and $E\left[\hat{\theta} - E[\hat{\theta}]\right] = 0$. Hence the MSE of an estimator is equal to the estimator's bias squared plus its variance. How well do our estimators $\hat{\eta}$ and $\tilde{\eta}$ from the previous section stack up based on MSE? Using this formula, we already have the hard part done:

$$MSE[\hat{\eta}] = \frac{\eta^2 - 1}{3N} + 0 \tag{2.59}$$

$$MSE[\tilde{\eta}] = \frac{N\eta^2}{(N+2)(N+1)^2} + \eta^2\left(\frac{1}{N+1}\right)^2 \tag{2.60}$$

$$= \eta^2\frac{2N+2}{(N+2)(N+1)^2} \tag{2.61}$$

$$= \eta^2\frac{2}{(N+2)(N+1)} \tag{2.62}$$

$$\frac{MSE[\tilde{\eta}]}{MSE[\hat{\eta}]} = \eta^2\frac{2}{(N+2)(N+1)} \times \frac{3N}{\eta^2 - 1} \tag{2.63}$$

$$= \frac{\eta^2}{\eta^2 - 1} \times \frac{6}{(N+2)(1+1/N)} \tag{2.64}$$

This leads to some ambiguity, but none that can't be dealt with with a bit of thinking. To begin with, the fraction $\frac{6}{(N+2)(1+1/N)} \to 0$ as $N \to \infty$, so as long as our sample is large enough we should probably use $\tilde{\eta}$. Additionally, the first term $\frac{\eta^2}{\eta^2-1}$ is reasonably close to 1 for any integer greater than about $\eta = 3$,[11] so we need not fret too much about it.

Activities

Activity 2.1.
[A low-tech Monte Carlo simulation] Take a 20-sided die and roll it 10 times, recording the numbers in a spreadsheet. This is your sample, which we shall denote $\{x_i\}_{i=1}^{10}$.

1. Calculate the sample mean of the following derivatives of your sample:

[11] "Close" is a judgment call on my part, but plug in some numbers an see for yourself.

(a) x_i (i.e. the untransformed variable itself)

(b) $y_i = x_i^2$

(c) $z_i = 1$ if $x_i \geq 17$, $z_i = 0$ otherwise

2. What are the *population* means of the variables X, Y and Z?

3. Are any of your sample means surprising (when you compare them to your population means)?

4. Report your three sample means to the class, so we have a dataset of N sample means, where N is the class size.

5. Produce histograms of these sample means

Exercises

Exercise 2.1.
Consider the distribution studied in Exercise 1.3. We derived the following properties:

$$f(x) = \begin{cases} \alpha x^{\alpha-1} & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$
$$E[X] = \frac{\alpha}{\alpha + 1}$$

This motivates the following estimator:

$$\hat{\alpha} = \frac{\frac{1}{N}\sum_{i=1}^{N} X_i}{1 - \frac{1}{N}\sum_{i=1}^{N} X_i}$$

which is the sample analog of:

$$\alpha = \frac{E[X]}{1 - E[X]}$$

An alternative estimator for $\tilde{\alpha}$ is:[12]

$$\tilde{\alpha} = -\frac{N}{\sum_{i=1}^{N} \log(X_i)}$$

Note that $\hat{\alpha}$ is a function of the sample mean of X, and $\tilde{\alpha}$ is a function of the sample mean of $\log(X)$

[12]This the *maximum likelihood* estimator of α, which is not important here, but we may cover this later in the year.

1. Download the dataset `ExBetaSim_1.csv` from Blackboard, which contains a simulated sample from this distribution. Use both estimators to estimate α.

2. (Simulation exercise) Fix $\alpha = 0.7$. Simulate some properties of these estimators for a sample size of $N = 30$. Are the estimators biased? Does one stand out as better than the other?

   Hint: You can simulate the distribution of X by transforming uniform random numbers. Specifically, if $U \sim \mathrm{U}[0, 1]$, then:

   $$X = U^{\frac{1}{\alpha}}$$

   will have the correct distribution.

Exercise 2.2 (Solutions provided)**.**
Consider the uniform distribution with unknown upper support. That This random variable can be characterized by the pdf:

$$f_X(x) = \frac{1}{\gamma} I(0 < x < \gamma) \tag{2.65}$$

We wish to estimate γ using an iid sample $\{X_i\}_{i=1}^N$ from this distribution, and the estimator:

$$\hat{\gamma} = \max_i \{X_i\} \tag{2.66}$$

That is, we use the sample maximum as an estimator for γ, the maximum value X could take on.

Find the following:

1. The cdf of $\hat{\gamma}$
2. The pdf of $\hat{\gamma}$
3. $E[\hat{\gamma}]$. is it biased? If so, can you correct it?
4. $V[\hat{\gamma}]$
5. Compare your last two answers to an alternative estimator:

   $$\tilde{\gamma} = \frac{2}{N} \sum_{i=1}^{N} X_i \tag{2.67}$$

   I.e.: twice the sample mean.

Exercise 2.3 (Solutions provided)**.**
The exponential distribution can be characterized by the pdf:

$$f_X(x) = \begin{cases} \mu^{-1} \exp(-x/\mu) & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.68}$$

where $\mu > 0$ is a scale parameter. This distribution has the following properties:

$$E[X^k] = k!\mu^k \tag{2.69}$$

$$X_i \sim iid\text{Exponential}(\mu) \implies \min\{X_1, X_2, X_3, \ldots, X_N\} \sim \text{Exponential}(\mu/N) \tag{2.70}$$

We could plug in $k = 1$ to the first property, and use:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} X_i, \qquad \text{analogy of: } \mu = E[X] \tag{2.71}$$

Alternatively, we could use the second property to construct the estimator:

$$\tilde{\mu} = N \min_i\{X_i\}, \qquad \text{analogy of: } \frac{\mu}{N} = E\left[\min_i\{X_i\}\right] \tag{2.72}$$

1. Derive the bias, variance, and MSE of these two estimators. Which one would you prefer to use? *Hint:* use the first property extensively!

2. (Simulation exercise) Simulate the properties of both estimators when $\mu = 1$ and $N = 30$. Your answer should include your approximation of the bias, variance, and MSE of the estimators, as well as a plot showing the pdfs of both estimators.

   Hint: If $U \sim U[0, 1]$, then $X = -\mu \log U \sim \text{Exponential}(\mu)$

3. (Simulation exercise) In principle, you could have plugged *any* k into 2.69 to get an analogy estimator. plug in $k = 1$ and derive the analogy estimator. Don't try to work out the bias, variance, and MSE of this analytically, just modify your code to also simulate the properties of this third estimator, say $\breve{\mu}$.

Exercise 2.4.
We know the following about two random variables X and Y:

- X and Y are independent
- $E[X] = E[Y] = \mu$
- $V[X] = \sigma_1^2 > 0$, $V[Y] = \sigma_2^2 > 0$.
- For some reason, we know the exact values of σ_1^2 and σ_2^2.

Suppose that we obtain a sample of one of each of these variables, i.e., x and y, and use it to construct an estimator for μ:

$$\hat{\mu} = ax + by \tag{2.73}$$

where a and b are constants chosen by the econometrician (you).

1. What is the expectation of $\hat{\mu}$?

2. For what values of a and b is $\hat{\mu}$ unbiased? Hint: We are looking for an expression of the form $b = f(a)$, find out what function f is.

3. For the moment, ignore your answer to part (2). That is, consider the original expression as a function of both a and b: $\hat{\mu} = ax + by$. What is the variance of $\hat{\mu}$?

4. Substitute the condition for unbiasedness (your answer in part (2)) to your expression for $V[\hat{\mu}]$ (your answer for part (3)). The right-hand side of this equation should contain only a, σ_1^2, and σ_2^2.

5. What value of a minimizes the variance of $\hat{\mu}$? Make sure that your answer is a global minimum. (Start with your answer to part (4))

6. Write out your expressions for a and b. In what case is $a = b$? What about $a > b$? When and why would we not want to just take the simple average $\hat{\mu} = \frac{1}{2}(x+y)$? Would we ever want to set either $a = 0$ or $b = 0$?

Exercise 2.5.
Write a script to simulate the sampling process you did in Activity 2.1, but instead of having your actual class size, suppose that you were in a class of $S = 10,000$. Write a script that simulates the sampling distribution of the sample mean of X, Y, and Z when each student has a sample of $N = 10$, $N = 100$ die rolls.

Chapter 3

Inference

In Chapter 2, we introduced the concept of an estimator, and some important properties of it. In econometrics, you will be estimating stuff all the time, but equally importantly, you will be making statements about your confidence in your estimates. Formally, such a statement will take the form of a hypothesis test, a p-value, or a confidence interval.

In many textbooks, the material that follows is presented alongside asymptotic theory, which tells us how estimators behave when the sample size approaches infinity. This is where your text will have a lot of $\xrightarrow{p}$, $\xrightarrow{d}$, and normal, F, and χ^2 distributions. These are very useful ideas, but I have found that they can be confusing when presented at the same time as the material I wish to teach you in this chapter. Therefore, what follows is a run-through of statistical inference in the absence of asymptotic theory, which we will get to in the next chapter. For the rest of this chapter, please note the absence of normal distribution tables, dividing mans by standard deviations, and the magic number 1.96.

To illustrate these concepts, let us go back to the coin-flipping example in Chapter 2. We have a data-generating process:

$$H_i \sim iid\text{Bernoulli}(\theta) \tag{3.1}$$

and wish to estimate θ, the probability that a coin flip will come up heads. Our research question is as follows: Is the coin a fair one? I hope that you never have to research a question as mundane as this, but once you're done with this chapter, go and do Exercise 3.1. Hopefully by then you can see the point of it. This research question can be formalized as $\theta = 0.5$. We collect a sample $\{H_i\}_{i=1}^N$ of N flips of the same coin, which we assume to be independent draws from 3.1. We then use the sample mean $\bar{h}$ as an estimator for θ:

$$\hat{\theta} = \frac{1}{N}\sum_{i=1}^{N} H_i \tag{3.2}$$

On its own, this gives us a point estimate of θ, which is somewhat useful, but at this point we have no idea how close our sample is to one that would come from fair coin flip. The next sections of this chapter approach the research question (is θ equal to 0.5?) three different ways.

3.1 Hypothesis tests

Loosely, this first approach asks whether our sample looks close enough to one that would come from a coin-flipping process with $\theta = 0.5$. To do this, we need a formal definition of "looks close enough to one that would come from a coin-flipping process with $\theta = 0.5$". Specifically, we ask whether our estimate of θ is close enough to the *hypothesized value* of $\frac{1}{2}$. Therefore we state the *null hypothesis*:

$$H_0 : \ \theta = 0.5 \tag{3.3}$$

and an *alternative hypothesis*:

$$H_A : \ \theta \neq 0.5 \tag{3.4}$$

This is a *two-sided* alternative hypothesis, because H_A permits θ to be either greater than or less than the value in the null. We will get to one-sided tests later.

Next we need a *test statistic*. This is a function of our sample, and should tell us something about θ. For the purposes of this application, we can just use our estimator for θ itself: $\hat{\theta} = \frac{1}{N}\sum_{i=1}^{N} H_i$. Using this, we can start to build up our definition of "looks close enough to one that would come from a coin-flipping process with $\theta = 0.5$". A natural measure of how close our sample is to one that would come from a $\theta = 0.5$ coin-flipping process is $t = \hat{\theta} - 0.5$: if t is close to zero, then this seems like good support for the coin being a fair one. On the other hand, it is unlikely that t is close to zero if the sample was generated by some other $\theta \neq 0.5$. This is illustrated in Figure 3.1. If H_0 is true, then our test statistic will have the distribution shown with the black lines. Notice here that there is a lot of probability for events where t is close to zero. On the other hand, if $\theta \neq 0.5$, then the distribution will look something like the red ($\theta = 0.2$) or blue ($\theta = 0.9$) lines. Note however, that we only get one realization of t, we next need to map this in a decision rule.

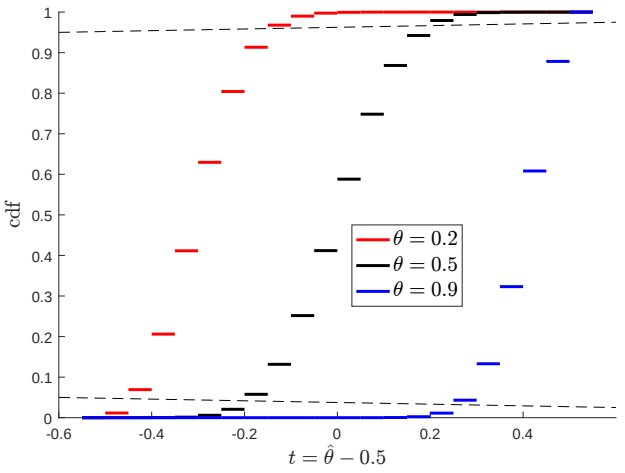


Figure 3.1: Cumulative distribution function of the test statistic for sample size $N = 20$. Dashed lines show the bounds for a two-sided, $\alpha = 5\%$ test.

t tells us "how close" the sample is, but how close is close *enough* for us to conclude that $\theta = 0.5$? To answer this, we need to make a trade-off about how often we want to be wrong, and what type of wrong that will be. Since there are two possible truths

Reject H_0?	H_0 true	H_0 not true
N	Good	Type II error
Y	Type I error	Good

Table 3.1: The four possible outcomes of a hypothesis test

(either H_0 is true or H_0 is not true), and two
possible decisions (either we reject H_0 or do
not reject H_0), then there are $2 \times 2 = 4$ possible outcomes of the test, half of which have us making the wrong conclusion. Table 3.1 summarizes the two types of wrong that we could be: we should be worried about either failing to reject H_0 when H_0 is true, a type II error, or rejecting H_0 when H_0 is true, a type I error. Since we are basing our decisions on the test statistic, which is random, there will always be a trade-off between these two errors. For practical reasons, we focus on targeting an acceptable probability of making a type I error: incorrectly rejecting the null hypothesis. One good reason for this is that this probability is a function of the null distribution (i.e. Binomial$(N, 0.5)$). In our case, we know this exactly, and in most cases, we can approximate it if N is large enough (see the next chapter to learn about this). Compare this to evaluating the probability of a type II error: if H_A is true, then all we know is that $\theta \neq 0.5$. How do we assess the distribution of t if we don't know the actual value of θ? That's a hard one, and one that we avoid entirely if we focus on targeting the probability of making a type I error.

To do this, we need to work out when we need to define a *decision rule* about rejecting H_0 based on t. As large $|t|$ is evidence against H_0, we will therefore use:

$$\text{Reject } H_0 \text{ if and only if } |t| > t_c \tag{3.5}$$

where $t_c > 0$ is a critical value. Note that as t_c gets larger, the more evidence we require against H_0 to reject it. When H_0 is true, the probability of rejecting H_0 based on this decision rule is the probability of making a type II error, and equal to:

$$\begin{aligned}
\Pr[\text{reject } H_0 \mid \theta = 0.5] &= \Pr[|t| \geq t_c \mid \theta = 0.5] && (3.6)\\
&= \Pr\left[\left|\frac{1}{N}\sum_{i=1}^{N} H_i - 0.5\right| > t_c\right] && (3.7)\\
&= \Pr\left[\left|\sum_{i=1}^{N} H_i - 0.5N\right| > Nt_c\right] && (3.8)\\
&= \Pr\left[\left(\sum_{i=1}^{N} H_i > N(t_c + 0.5)\right) \cup \left(\sum_{i=1}^{N} H_i < N(t_c - 0.5)\right)\right] && (3.9)\\
&= \Pr\left(\sum_{i=1}^{N} H_i > N(t_c + 0.5)\right) + \left(\sum_{i=1}^{N} H_i < N(t_c - 0.5)\right) && (3.10)
\end{aligned}$$

where the last line follows because $\sum_{i=1}^{N} H_i > N(t_c + 0.5)$ and $\sum_{i=1}^{N} H_i < N(t_c - 0.5)$ are mutually exclusive events. But we can simplify this further, because we know that $\sum_{i=1}^{N} H_i \sim$ Binomial$(0.5, N)$: we just need to add up all of the bits of its pmf that satisfy

$\sum_{i=1}^{N} H_i > N(t_c + 0.5)$ or $\sum_{i=1}^{N} H_i < N(t_c - 0.5)$:

$$\Pr[\text{reject } H_0 \mid \theta = 0.5] = \sum_{k:k>N(t_c+0.5)} p(k) + \sum_{k:k<N(t_c-0.5)} p(k) \tag{3.11}$$

$$= \sum_{k:k>N(t_c+0.5)} \frac{N!}{k!(N-k!)} 0.5^N + \sum_{k:k<N(t_c-0.5)} \frac{N!}{k!(N-k!)} 0.5^N \tag{3.12}$$

where the "$k : k > N(t_c+0.5)$" bit means "sum over all ks satisfying $k > N(t_c+0.5)$". We are looking to set this probability equal to $\alpha = 5\%$. α, the probability of rejecting H_0 when it is true, is referred to the test *size*. Actually, we can't in general set this probability to *exactly* 5% for the binomial distribution, because we have no guarantee that there is a place in the pmf where we can stop adding and get 5%. Let's pick smallest t_c such that this thing is less than 5%.

Table 3.2 shows the relevant pdf and cdf. Since the $Binomial(N, 0.5)$ distribution is symmetric (i.e. $p(X) = p(N - X)$), we can look for one cutoff k_c at the left tail such that $\Pr[X < k_c]$ is just less than 2.5%, and then the other cutoff will be at the corresponding point of the right tail: i.e. $N - k_c$. Looking at Table 3.2, we see that 2.1% of the samples will have 5 or fewer heads, and 5.8% of samples will have 6 or fewer heads. Therefore we can choose $k_c = 5$ and have a probability of rejecting H_0 when H_0 is true of $2 \times 2.1\% = 4.2\%$, which is reasonably close to the standard number of 5%. Hence, we will reject H_0 if and only if we observe a sample with 5 or fewer heads, or 16 or more heads. Note that we found these cutoffs from Table 3.2 by finding the (approximate) solutions to $F_X(x) = 0.025$ and $F_X(x) = 0.975$.

X	$p_X(x)$	$F_X(x)$	$\hat{\theta}$	t
0	0.0000	0.0000	0.00	-0.50
1	0.0000	0.0000	0.05	-0.45
2	0.0002	0.0002	0.10	-0.40
3	0.0011	0.0013	0.15	-0.35
4	0.0046	0.0059	0.20	-0.30
5	0.0148	0.0207	0.25	-0.25
6	0.0370	0.0577	0.30	-0.20
7	0.0739	0.1316	0.35	-0.15
8	0.1201	0.2517	0.40	-0.10
9	0.1602	0.4119	0.45	-0.05
10	0.1762	0.5881	0.50	0.00
11	0.1602	0.7483	0.55	0.05
12	0.1201	0.8684	0.60	0.10
13	0.0739	0.9423	0.65	0.15
14	0.0370	0.9793	0.70	0.20
15	0.0148	0.9941	0.75	0.25
16	0.0046	0.9987	0.80	0.30
17	0.0011	0.9998	0.85	0.35
18	0.0002	1.0000	0.90	0.40
19	0.0000	1.0000	0.95	0.45
20	0.0000	1.0000	1.00	0.50

Table 3.2: Pmf and cdf for $X = \sum_i H_i \sim \text{Binomial}(0.5, 20)$.

We're almost there. In fact, we could do the hypothesis test without going any further, but since we specified things in terms of our test statistic t instead of the sum of heads, for completeness we should work out t_c. Graphically this is exactly the same problem as solving for the rejection rule in terms of the sum of heads. To see this, have a look at Figure 3.1. The dashed lines have horizontal coordinates of 0.025 and 0.975. What we need to do is look at the cdf of t when H_0 is true (the black line), and read off these points. These points are when $t = -0.2$ and $t = 0.2$. So our rejection rule becomes:

Reject H_0 if and only if: $|t| > 0.2$

Hence, we would reject H_0 if we observed, say, 3 or 19 heads in our sample, but would not reject H_0 if we observed 8 or 11 heads in our sample.

At this point it should be pretty obvious to you that you will need to compute a lot of probabilities associated with the Binomial distribution. To learn about how to do this in *Stata*, a good place to start would be by typing the following into the command line:

```
help binomial
```

This gives you a pretty minimal description, but the blue text is a link to some more statistical functions that you may want to use later.

3.1.1 One-sided hypothesis tests

The previous section presented a *two-sided* hypothesis test: it was done under the assumption that the coin could possibly be unfair because either θ was more or less than 0.5. Sometimes, we have reason to rule out a portion of the alternative hypothesis space, and usually this means that if H_0 is not true, then we know which side of the hypothesized value (in our case 0.5) the true value of θ is. Suppose, for example, that instead of "are we are flipping a fair coin?", our research question was "is the coin biased towards heads?". Formally, we could state a null and alternative as:

$$H_0 : \ \theta = 0.5, \quad H_A : \ \theta > 0.5 \tag{3.13}$$

that is, our research question motivates a (dogmatic) belief that θ could *never* be less than 0.5. The procedure for such a test is exactly the same, but we just need to think a bit more about what realizations of $\hat{\theta}$ (or t) would provide us with support for H_A in favor of H_0. For example, observing 2 heads, and so calculating $\hat{\theta} = 0.1$ and $t = 0.4$ would not be very convincing that $\theta > 0.5$, however we would have rejected H_0 for the two-sided test outlined in the previous section. Hence, we need a rejection rule that only rejects H_0 when we observe a sufficiently large $\hat{\theta}$, or sufficiently positive t. Other than that, we approach the problem in exactly the same way.

We need to choose a critical value t_c such that the rejection rule:

$$\text{Reject } H_0 \text{ if and only if } t > t_c \tag{3.14}$$

so that the probability of this event, if H_0 was true, is equal to $\alpha = 0.05$, or at least close to 0.05, since t is a discrete random variable. We need to solve for:

$$\begin{aligned}
\alpha &= \Pr\left[t > t_c\right] && (3.15)\\
&= \Pr\left[\frac{1}{N}\sum_{i=1}^{N} H_i - 0.5 > t_c\right] && (3.16)\\
&= \Pr\left[\sum_{i=1}^{N} H_i > N(t_c + 0.5)\right] && (3.17)\\
&= \Pr\left[\text{Binomial}(\mathrm{N}, \theta) > \mathrm{N}(\mathrm{t_c} + 0.5)\right] && (3.18)\\
&= 1 - F_X\left(N(t_c + 0.5)\right) && (3.19)
\end{aligned}$$

where $F_X(\cdot)$ is the binomial cdf shown in Table 3.2. Therefore we are looking for a cell in the $F_X(x)$ column of this table corresponding to (roughly) $F_X(x) = 1 - \alpha = 0.95$. The probability of drawing 13 or fewer heads is 0.942, and the probability of drawing 14 or fewer heads is 0.979, so we can't get exactly $\alpha = 0.05$, but the following decision rule:

$$\text{Reject } H_0 \text{ if and only if } t > 0.15 \tag{3.20}$$

gets reasonably close: $\alpha = 1 - 0.9423 = 0.0577$.

3.2 p-values

Another popular way of reporting statistical significance is with a p-value. The p-value associated with a hypothesis test is defined as the probability of observing a test statistic at least as extreme as the one we actually observed, assuming that H_0 is true. If this *almost* seems like α, the test size, then you have made an important connection! p is equal to the test size α that would put you on the margin between rejecting and not rejecting H_0 for your observed sample.

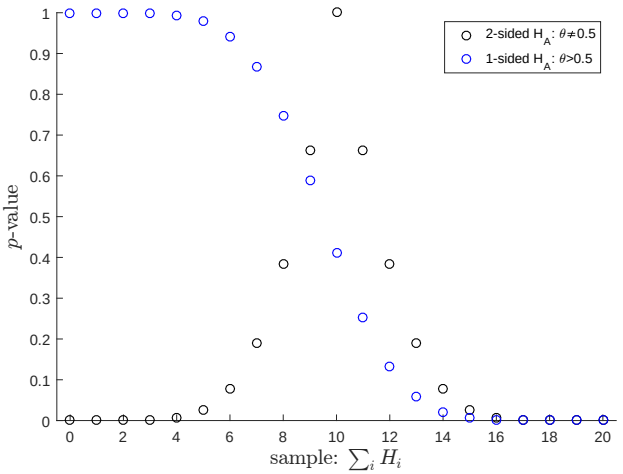


Figure 3.2: p-values associated with the 21 possible samples.

The benefit of reporting a p-value is that it allows your reader to test your hypothesis at their choice of α, rather than the one that you selected. For example, if you calculated $p = 0.03$, then you would reject H_0 if you wanted to do an $\alpha = 0.05$ test, but fail to reject H_0 for an $\alpha = 0.01$ test. A smaller p-value means that the data would pass a more stringent hypothesis test.

For example, suppose that you observed 3 heads in your sample. Since we've worked through Section 3.1, you know that you would reject the two-sided test that $\theta = 0.5$. However what if a pesky audience member at a seminar is in the mood for a more conservative test. By reporting the p-value, this may avoid an annoying question. So let's calculate it. Since we are doing a two sided test, there are eight samples that are at least as unlikely to occur when H_0 is true. These the samples that include either 0, 1, 2, 3, 17, 18, 19, or 20 heads. Hence the p-value is the sum of the probabilities of these events occurring, assuming that H_0 is true, i.e. $\theta = 0.5$:

$$p = \sum_{k=0}^{3} \frac{20!}{k!(20-k)!}\frac{1}{2^{20}} + \sum_{k=17}^{20} \frac{20!}{k!(20-k)!}\frac{1}{2^{20}} \approx 0.0015 \tag{3.21}$$

so we would reject H_0 at most reasonable levels of significance (including $\alpha = 0.05$). The black circles in Figure 3.2 show the p-values associated with the 2-sided test for all 21 possible realizations of $\sum_i H_i$.

For the one-sided test in Section 3.1.1, note that observing 3 heads is terrible support for H_A, so before we actually calculate this thing, note that it should be close to 1. We need to add up the probabilities of all the samples we could have observed, that would have supplied at least as much support for $H_A : \ \theta > 0$ as did our observed sample of 3 heads. These the samples that include either 3, 4, 5, ..., or 20 heads:

$$p = \sum_{k=3}^{20} \frac{20!}{k!(20-k)!} \frac{1}{2^{20}} \approx 0.9987 \tag{3.22}$$

The blue circles in Figure 3.2 show the p-values associated with this 1-sided test for all 21 possible realizations of $\sum_i H_i$.

For the other 1-sided test, with $H_A : \ \theta < 0.5$, we up all of the probabilities associated with getting *at most* 3 heads in our sample. In this case, observing 3 out of 20 heads is somewhat strong support for H_A in favor of H_0, because we shouldn't expect this to happen too often assuming H_0 is true. In this case:

$$p = \sum_{k=0}^{3} \frac{20!}{k!(20-k)!} \frac{1}{2^{20}} \approx 0.0013 \tag{3.23}$$

so we would reject H_0 at for any test with $\alpha > 0.0013$.

It is important to interpret p-values correctly. It is tempting to claim that p is the probability that H_0 is not true. However this is false.[1] Remember that we derived the p-value *assuming* that H_0 was true. Hence, you are permitted to interpret it in the following ways:

- p is the probability of calculating a test statistic at least as extreme as the one you actually calculated, assuming H_0 is true.
- Assuming H_0 is true (and all other distributional assumptions about the data-generating process are correct), p will be uniformly distributed (think about this when/if you learn about the method of inversion for generating random numbers).
- When H_0 is true, rejecting H_0 when $p < \alpha$ implements the same decision rule as testing H_0 at the α level of significance.

It tells you something, just be aware of what it *doesn't tell you*.

3.3 Confidence intervals

[1] Do Bayesian econometrics if you want to make claims like this.

The third way we might want to report the statistical significance of our results is a *confidence interval*. This reports all of the values of θ_0 for which we would fail to reject H_0 in the test:

$$H_0 : \ \theta = \theta_0, \quad H_A : \ \theta \neq \theta_0$$

(this also applies to one-sided hypothesis tests).

This is useful because it reports, holding α constant, *all* of the null hypotheses that would not be rejected. Figure 3.3 shows the confidence intervals we would assign after observing each of the 21 possible samples we could observe from flipping 20 coins. The black lines show the 2-sided confidence intervals. The red crosses show the lower bound of the one-sided confidence intervals with alternative $H_A : \ \theta < \theta_0$; the upper bound of all of these is $\theta = 0$. and the blue crosses show the upper bound of the one-sided confidence intervals with alternative $H_A : \ \theta > \theta_0$; the lower bound of all of these is $\theta = 1$.

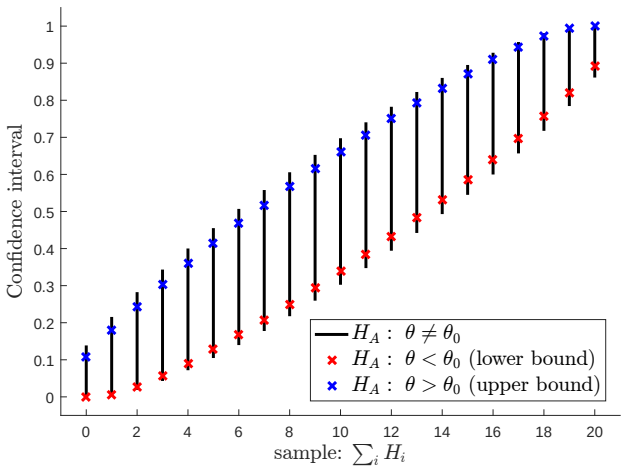


Figure 3.3: Confidence intervals ($\alpha = 0.1$) associated with the 21 possible samples. Black lines show the intervals for the 2-sided alternative. Red and blue crosses show the minimum and maximum values in the confidence intervals for the one-sided tests respectively.

3.4 Test power

Up to this point, all of our calculations were done assuming that H_0 was true (remember this, it's important). But what about H_A? Often we will be testing something with the expectation that H_0 is *not* true. For example, maybe some economic theory tells us that we are not flipping a fair coin (maybe more on this later). α tells us the probability that we are wrong when H_0 is true, but what about being wrong when H_A is true? That is, what is the probability of a Type I error? The reason it is (relatively) easy to work out things when H_0 is true is that H_0 completely pins down the distribution of our test statistic. In our case in this Chapter, we know that $\sum_i H_i \sim \text{Binomial}(N, 0.5)$. But for the alternative, all we know is that θ falls in a range: the distribution of the test statistic when H_A is true is not known! That being said, we can answer a simple question: what is the probability of rejecting the null when θ is equal to a particular value? If this seems similar to α, good! α is the answer to this question if we plug in θ equal to the value set in H_0 (in this example, $\theta = 0.5$). If we instead plugged in a value consistent with H_A, we would be calculating the test's *power*. That is, if H_0 is *not* true, how good is our test at telling us this?

To get this, let's go back to (3.6), but substitute in another value of θ, say 0.6 (i.e. the

coin comes up heads with probability 60%):

$$\Pr[\text{reject } H_0 \mid \theta] = \Pr\left(\sum_{i=1}^{N} H_i > N(t_c + 0.5) \mid \theta\right) + \left(\sum_{i=1}^{N} H_i < N(t_c - 0.5) \mid \theta\right) \tag{3.24}$$

So we know that $\sum_i H_i \sim \text{Binomial}(N, \theta)$, so this thing is equal to:

$$\Pr[\text{reject } H_0 \mid \theta] = 1 - F\left(N(t_c + 0.5); N, \theta\right) + F\left(N(t_c - 0.5) - 1; N, \theta\right) \tag{3.25}$$

where $F(\cdot; N, \theta)$ is the cdf of the Binomial(N, θ) distribution. For our 5% test calculated earlier, we had $t_c = 0.2$, so this reduces to calculating the probability of getting 6 or fewer heads, or 16 or more heads. We have already worked out that this is equal to $\alpha = 4.2\%$ when H_0 is true, but now we evaluate the same probability for a different θ. The probability of rejecting H_0 when $\theta = 0.6$ is:

$$Pr\left[\text{reject } H_0 \mid \theta = 0.6\right] = \sum_{k=0}^{6} \frac{20!}{k!(20-k)!} 0.6^k 0.4^{20-k} + \sum_{k=15}^{20} \frac{20!}{k!(20-k)!} 0.6^k 0.4^{20-k} \tag{3.26}$$

$$\approx 13\% \tag{3.27}$$

We we want this number to be as big as possible because we want to be able to reject H_0 whenever it is false. I hope 13% seems reasonably bad to you. This means that 87% of the time, we fail to reject H_0. But this is what we accept when we pin down $\alpha = 0.05$. As long as we want to do a 5% test, the only variable we can play with is the sample size, N. Unsurprisingly, the test power gets bigger (which is better) as N increases. Figure 3.4 shows this relationship. The black line shows the test size, which we have set to 5%. This and the other lines are jagged because the Binomial distribution is discrete: this wouldn't happen with a continuous distribution. The colored lines show the test power for 3 different values of θ. Looking at each line individually, it is comforting that they are upward-sloping (outside of the jagged shape due to the discrete distribution). Furthermore, as θ becomes further away from the H_0 value, the power increases: it is easier to spot the difference between $\theta 0.5$ and $\theta = 0.8$ than it is to spot the difference between $\theta 0.5$ and $\theta = 0.6$.

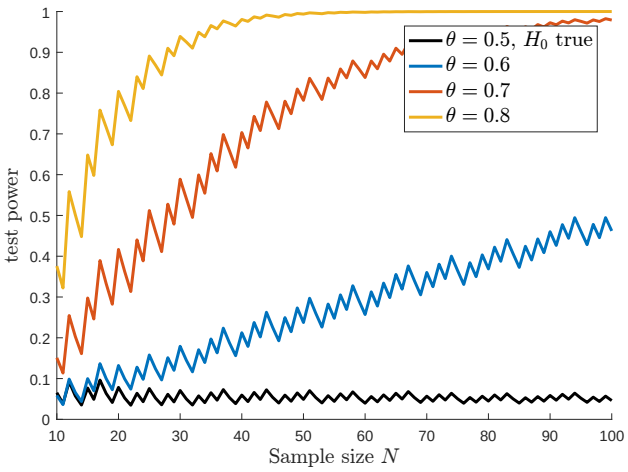


Figure 3.4: Power of the 2-sided Binomial test outlined in this chapter, targeting a test size of $\alpha = 0.05$.

3.5 The take-away

I write these notes assuming that my students have been introduced to hypothesis tests, p-values and confidence intervals at an introductory statistics/quantitative methods level. That is, you have probably seen a mechanical, plug-and-chug, explanation of *what to do*. I intend for this to be a chapter on *what you're doing*, and *why you're doing it*. So what should you take away from this?

Firstly, here's what we did:

1. Stated formal null and alternative hypotheses
2. Defined a test statistic
3. Worked out the distribution of the test statistic assuming H_0 is true
4. Used this distribution to work out a decision rule
5. Reported at least one of (i) the result and conclusion from a hypothesis test, (ii) reported and interpreted a p-value, and (iii) reported and interpreted a confidence interval.
6. Commented on our test's power of identifying the alternative hypothesis when it is true

But be aware that we did none of the following

- Divide by the sample standard deviation
- Look up a normal, χ^2, Student's t, or F distribution
- 1.96

although we *did* look up a distribution table, namely Table 3.2. Don't worry, what you were taught in the past (probably) wasn't wrong. Just be aware that hypothesis tests and the like can be done without assuming that *anything* is normally distributed. The reason that we so often do make a normal approximation is that it makes step 3 a whole lot easier, and often doesn't change things too much.

Exercises

Exercise 3.1.
Load the Galton heights dataset. Just focus on parent height. We have a working hypothesis that within a randomly selected couple, the father is more likely to be taller than the mother.

1. Formally define a population parameter that speaks to this test, and state a formal null and alternative hypothesis about this parameter that addresses the working hypothesis above.

2. Define a test statistic for this hypothesis.

3. What is the distribution of the test statistic when the null hypothesis is true?

4. Calculate the p-value associated with your null and alternative hypotheses. State any additional assumptions you needed to make for your procedure to be valid (and don't make any assumptions tht ou don't need to). Produce a graph to illustrate what you're doing. *Hint:* This is a computationally intensive exercise that will take you too long and waste too much paper if you do it by hand.

Exercise 3.2 (Assessing the performance of a "cookbook" hypothesis test)**.**
Consider the following procedure for a hypothesis test for a dataset of N iid coin flips $\{X_i\}_{i=1}^N$:

$$H_0 : \Pr[X_i = 1] = 0.5, \quad H_A : \Pr[X_i = 1] \neq 0.5$$
$$t = 2\sqrt{N}(\bar{x} - 0.5), \quad \bar{x} = \frac{1}{N}\sum_{i=1}^{N} X_i$$
$$\text{Reject } H_0 \text{ if and only if } |t| > 1.96$$

(We will understand why this might be a good approach in some circumstances in the next chapter.) Evaluate the actual size (i.e. α) of this hypothesis test. How does the actual size change with sample size N?

You may want to break this problem up into answer the following steps:

1. What component of t is random due to the random sampling procedure (*Hint:* there are two possible answers: $\bar{x}$ and N. Work out which is correct.)

2. What is the distribution of this random component when H_0 is true?

3. What values of $\bar{x}$ mean that you would reject the null? (draw a graph)

4. What is the probability that $\bar{x}$ falls in to this range?

5. How does this probability relate to α?

Exercise 3.3 (Understanding some *Stata* code)**.**
Consider the following script:

```
clear all
set obs 50
generate X = 0
replace X = 1 if runiform()<=0.5
summarize X
generate t = 2*sqrt(r(N))*(r(mean)-0.5)
summarize t
generate reject = 0
replace reject =1 if abs(t)>1.96
summarize reject
```

Note especially the line that begins `generate t =` . In relation to Exercise 3.2:

1. Provide a one-sentence description of what each line of this script does (i.e. thoroughly comment the code). Also Provide a brief description of what the entire script does.

2. Based on your answer to Exercise 3.2, what is the probability distribution of the variable `reject`? Express your answer as a function of α. (*Hint:* What values could `reject` take on, and what are the probabilities of these events?)

3. Modify this code to generate 100,000 simulated draws from t. What is the actual test size? What number should you use if you wanted to do a 5% test? *Hint:* This will take a while. Until you want to get your final answer, run your script with a smaller simulation size (e.g. 100) to make sure it works,

Exercise 3.4.
Figure 3.4 shows the power of a 2-sided Binomial test, and how it varies by sample size N. Produce a similar plot that shows the trade-off in a 1-sided test between test power and test size (α). Set $N = 100$ constant.

Exercise 3.5.
`PoissonData.dta` contains data on two variables, `X` and `Y`. For each of these variables: Use the following results to complete this exercise:

$$\text{If } X_1,\ X_2, \ldots, X_N \sim iid\text{Poisson}(\lambda),\ \text{then } \sum_{i=1}^{N} X_i \sim \text{Poisson}\left(N\lambda\right) \tag{3.28}$$

1. Perform a 2-sided hypothesis test that the data are drawn from a Poisson(1) distribution. Do this test at the 5% level of significance.

2. Perform a 1-sided hypothesis test that the data are drawn from a Poisson(1) distribution, with the alternative being that $\lambda < 1$. Do this test at the 5% level of significance.

3. Assign p values to the above hypotheses

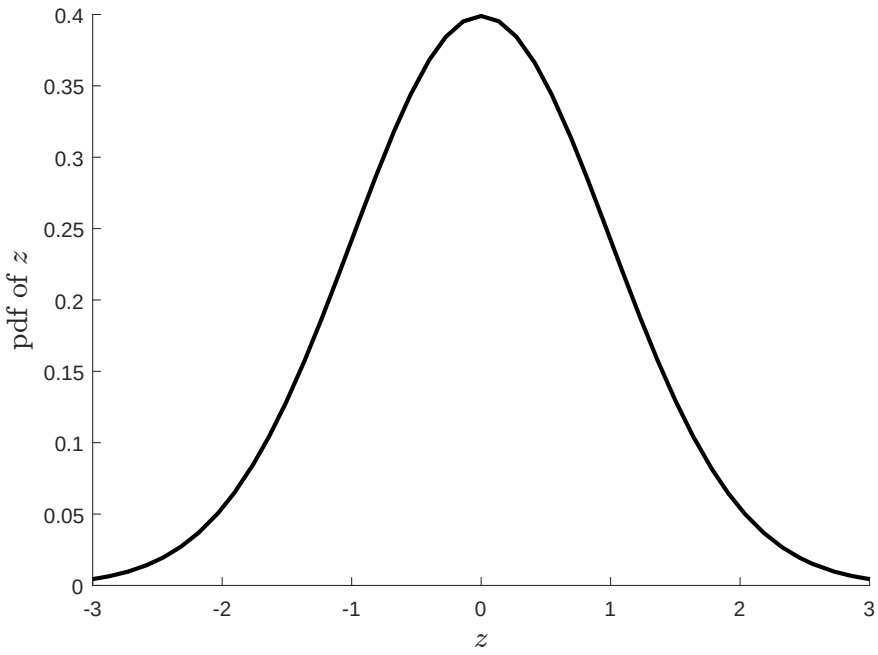
4. Construct a 95%, 2-sided confidence interval for λ (much more difficult)

Exercise 3.6.
These questions ask you to annotate figures. If you need to find an area, shade an area (and tell me what that area is equal to), if you need to find a horizontal and/or vertical coordinate, label it (and tell me what it is equal to); and so on.

1. You perform the following hypothesis test:

$$H_0:\ E[X] = 4, \quad H_A:\ E[X] > 4$$
$$\text{test size} = \alpha = 0.1$$
$$\text{test statistic: } z = \frac{1}{N}\sum_{i=1}^{N} X_i - 4$$

   The following figure shows the pdf of z when H_0 is true. Annotate this figure to show how you would find the p-value for this test if your test statistic was $z = 1$:



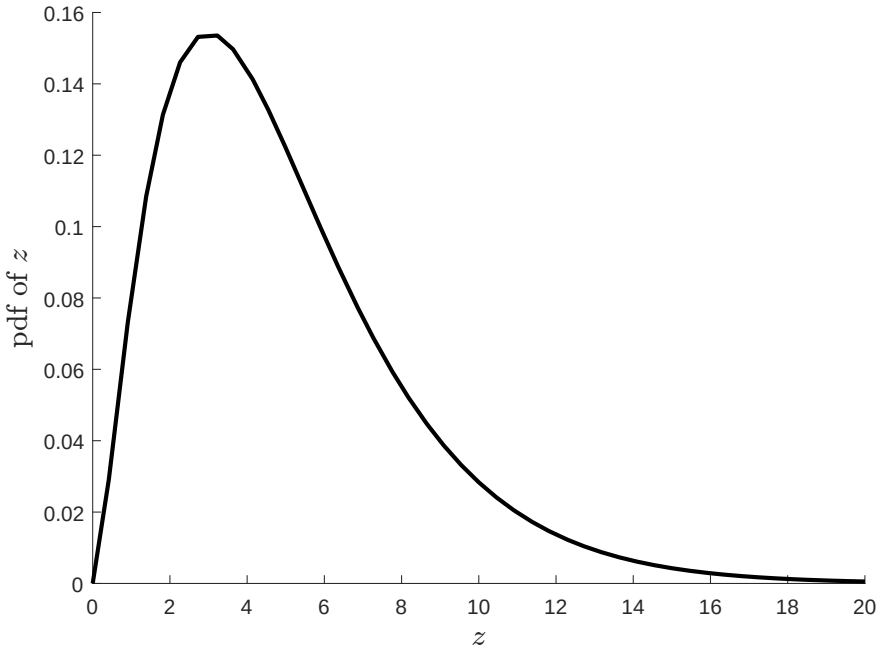
2. You perform the following hypothesis test:

$$H_0: \ E[X] = 4, \quad H_A: \ E[X] \neq 4$$

$$\text{test size} = \alpha = 0.2$$

$$\text{test statistic: } z = \left(\frac{1}{N} \sum_{i=1}^{N} X_i - 4 \right)^2$$

The following figure shows the pdf of z when H_0 is true. Annotate this figure to show how you would find the critical value (or values) for this test:



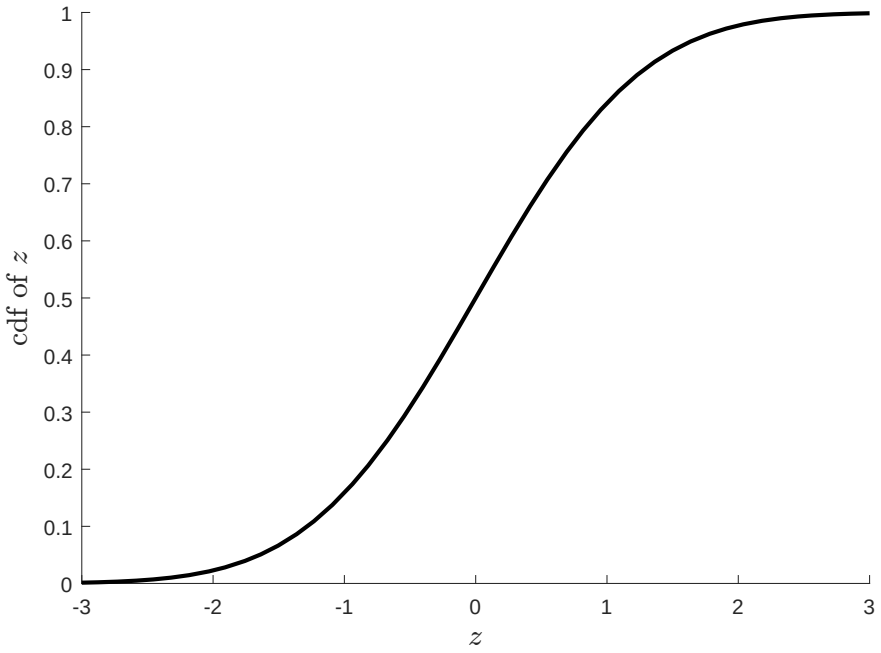
3. You perform the following hypothesis test:

$$H_0: \ E[X] = 0, \quad H_A: \ E[X] < 0$$

$$\text{test size} = \alpha = 0.1$$

$$\text{test statistic: } z = \frac{1}{N}\sum_{i=1}^{N} X_i - 0$$

The following figure shows the cdf of z when H_0 is true. Annotate this figure to show how you would find the critical value (or values) for this test:



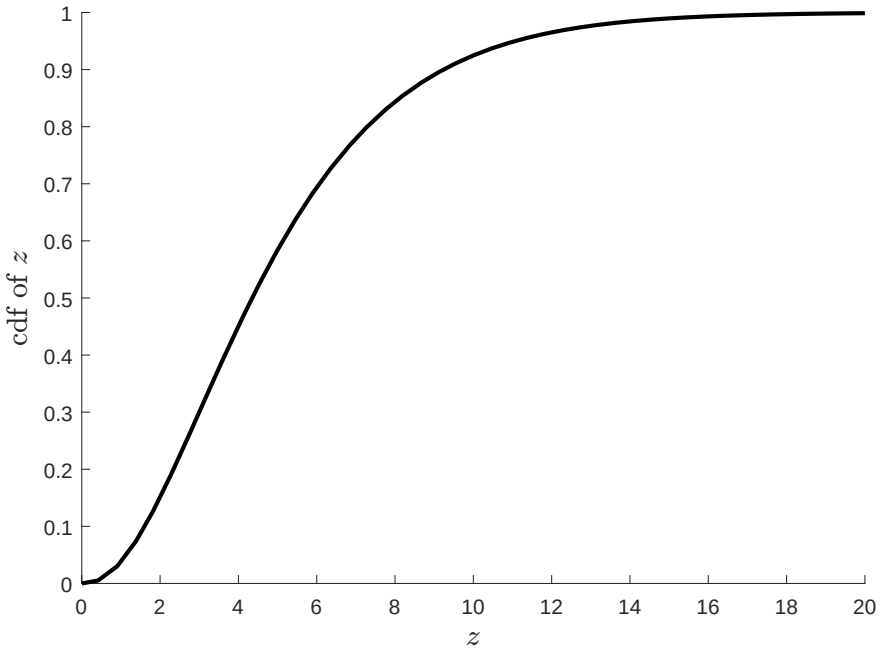
4. You perform the following test:

$$H_0: \ E[X] = 1, \quad H_A: \ E[X] \neq 1$$

$$\text{test size} = \alpha = 0.1$$

$$\text{test statistic: } z = \frac{4}{N} \sum_{i=1}^{N} X_i$$

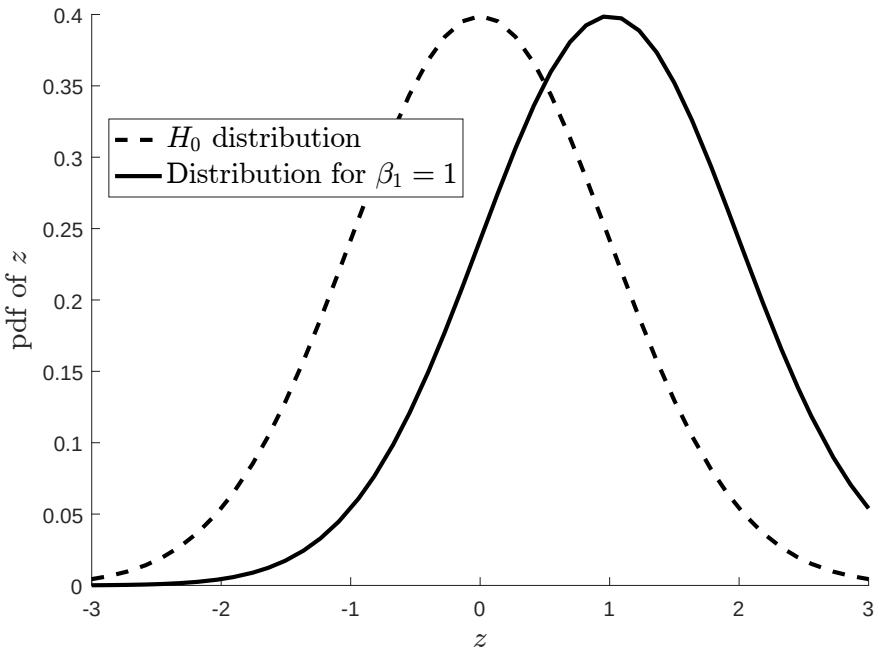
The following figure shows the cdf of z when H_0 is true. The test statistic for your sample is $z = 6$. Annotate this figure to show how you would find the p-value (or values) for this test.



5. You are testing the hypothesis:

$$H_0: \ \beta_1 = 0, \quad H_A: \ \beta_1 > 0$$
$$\text{test size} = \alpha = 0.1$$
$$\text{test statistic: } z = \hat{\beta}_1$$

The dashed line shows the distribution of $z = \hat{\beta}_1$ when H_0 is true. The solid line shows the distribution of $z = \hat{\beta}_1$ when $\beta_1 = 1$ (i.e. a special case within H_A). Annotate this figure to show how you would work out the the power of this test when $\beta_1 = 1$.


0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
pdf of z
H_0 distribution
Distribution for $\beta_1 = 1$
-3
-2
-1
0
1
2
3
z

Chapter 4

Inference with asymptotic assumptions

At this point, I am hoping that you have a rather grim view of hypothesis tests in the context of Chapter 3: we need to know a lot about our random variable in order to derive the distribution of our test statistic when H_0 is true, and even when we know all of this, the process is a difficult one. Fortunately, we have another tool that allows us to make the process much simpler: asymptotics. Even if we don't know (or don't care) enough about our random variable to derive exactly its sampling distribution, we can use this tool to work out what it would be as our sample size N approached infinity. Then we take the leap of faith that our sample size is large enough that this distribution is a good approximation for our actual sample.

To do this, we use two theorems about sample means. The Weak Law of Large Numbers tells that as our sample size $N \to \infty$, the probability that we are arbitrarily close to the population mean (i.e. $E[X]$) approaches 1. Then, central limit theorems tell us how we can appropriately scale things so that they are (usually) normally distributed as $N \to \infty$. I introduce these concepts, then outline how we can use them to derive approximate properties of our estimator and/or test statistic if we can argue that our sample size is large enough. We then learn how these are useful for doing inference, and finish with a useful approximation of transformation of sample means.

4.1 Large-sample properties of estimators

In Chapter 2, we learned that bias, variance, and mean squared error are useful quantities to summarize the performance of an estimator. These are sometimes referred to *small sample* properties of estimators, meaning that they are things that you might need to worry about if your sample size is small. You may also have to worry about them is N is large, but there are some other properties that you might need to know about your estimators that relate to large samples.

4.1.1 Consistency

Suppose that there is a population parameter θ that you would like to estimate. You have a sample $\{X_i\}_{i=1}^N$, and an estimator $\hat{\theta}$ which takes this sample and spits out a number (an estimate), which you hope is close to the true value, θ. You want to know if your estimating procedure is one in which obtaining more observations (i.e. increasing N) gets your estimate closer to θ. Unfortunately, since your sample is random, so is your estimator $\hat{\theta}$. This means that no matter how much data you collect, there is still a chance that your estimate is terrible. What we *can* work out, though, is whether increasing N will get us close enough to θ, with probability very close to 1. In math speak, what we want is:

$$\Pr\left(|\hat{\theta} - \theta| > \epsilon\right) \to 0 \text{ as } N \to \infty, \text{ for all } \epsilon > 0 \tag{4.1}$$

which we can write more compactly as $\text{plim}\hat{\theta} = \theta$, and say "the probability limit of $\hat{\theta}$ is θ," or (since we know $\hat{\theta}$ is an estimator) "$\hat{\theta}$ is a consistent estimator (of θ)." Inspecting (4.1), what is it telling us. $|\hat{\theta} - \theta|$ is the distance between our estimator and the true value, and ϵ is a positive number. So the probability that $\hat{\theta}$ is at least ϵ away from the thing we are trying to estimate goes to zero as our sample size goes to infinity. The "for all $\epsilon > 0$" means that this probability goes to zero no matter what positive number you pick for ϵ. In other words, no matter how I define "close enough" (i.e. ϵ), I can get close enough to the true value with probability 1 by sending the sample size off to infinity. Loosely speaking, if you have a consistent estimator, collecting more data means that you are more likely to have a good estimate.

4.1.2 Asymptotic distribution

In the previous chapter, probably the hardest thing to do computationally was to determine the sampling distribution of the estimator. In some special cases, such as Bernoulli (coin flip) and Normal random variables, we can work it out. For large samples, though, this is more of a classroom exercise rather than something that is done in practice (although if you deal with a lot of small samples, you may need this). Instead, much inference is based on determining the *asymptotic* distribution of an estimator. We may have no idea what the exact (small sample) distribution is, but we can work out what (a transformation of) it looks like when $N \to \infty$. We then assume that this is a good enough approximation of our actual, finite sample size. For a lot of cases, we will be using estimators that are (sometimes fancy) sample means. In this case we can use the work of others (see below) to construct something that is approximately standard normal when N is large. This is useful because we need to know *much* less about the data-generating process in order to work out the (approximate) distribution. I will leave further discussion of this to the next section.

4.2 Large-sample properties of sample means

Fortunately for us, (i) many of our estimators and test statistics are just fancy sample means, and (ii) a lot of work has gone into understanding sample means. I present some of the results of (ii) below, which will make our life a lot easier.

4.2.1 The Weak Law of Large Numbers

The weak law of large numbers tells us (loosely) that a sample mean (i.e. $\frac{1}{N}\sum_i X_i$) will get very close to the equivalent population mean (i.e. $E[X]$) as our sample size (i.e. N) becomes large. Formally:

Theorem 2 (Weak law of large numbers)**.** *Let X_i be an infinite set of iid Lebesgue integrable random numbers satisfying $E[X_i] = \mu$ for all i. Then:*

$$\lim_{N\to\infty} \Pr\left(\left|\frac{1}{N}\sum_{i=1}^{N} X_i - \mu\right| > \epsilon\right) = 0, \quad \textit{for all } \epsilon > 0 \tag{4.2}$$

The above limit can also be written as:

$$\frac{1}{N}\sum_{i=1}^{N} X_i \xrightarrow{p} \mu \tag{4.3}$$

or:

$$\operatorname{plim}\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right) = \mu \tag{4.4}$$

What does (4.2) mean in plain(er) English? Note that $\left|\frac{1}{N}\sum_{i=1}^{N} X_i - \mu\right|$ is the distance between our sample mean and the population mean. This is random because we have a random sample. Now we define ϵ as some arbitrary criterion for closeness, and ask the question: how likely are we to get a sample mean at least ϵ away from the population mean? (4.2) tells us that no matter how we define this criterion closeness, as $N \to \infty$ our sample will be close to the population mean with probability approaching 1. Basically, sample means converge to population means as $N \to \infty$. In other words: sample means are consistent estimators of population means!

4.2.2 A central limit theorem

So the Weak Law of Large Numbers is useful for point estimates: if we have a sample mean with a large sample size, we are likely to get very close to the population mean. However this is not helpful for inference. How do we put a confidence interval around an estimate if the distribution of the estimator collapses to a point? The answer is to use an appropriate

scaling of the estimator that *doesn't* collapse. To understand the problem, note that the variance of the sample mean for an iid sample is:

$$V[\bar{x}_N] = V\left[\frac{1}{N}\sum_{i=1}^{N} X_i\right] = \frac{1}{N^2}V\left[\sum_{i=1}^{N} X_i\right] = \frac{1}{N^2}\sum_{i=1}^{N} V[X] = \frac{1}{N^2}NV[X] = \frac{V[X]}{N} \tag{4.5}$$

So for finite $V[X]$, $V[\bar{x}_N] \to 0$ as $N \to \infty$. The solution to this can also be seen in Equation 4.5: we need to multiply $\bar{x}_N$ by a fudge factor $g(N)$ (actually a fudge *function* of N) that increases in such a way that $V[g(N)\bar{x}]$ is a constant. Since $g(N)$ is not random, we can do the following:

$$V[g(N)\bar{x}_N] = V[g(N)\bar{x}_N] = [g(N)]^2 V[\bar{x}_N] = [g(N)]^2 \frac{V[X]}{N} \tag{4.6}$$

So if $g(N) = \sqrt{N}$, then:

$$V[g(N)\bar{x}_N] = V[\sqrt{N}\bar{x}_N] = N\frac{V[X]}{N} = V[X] \tag{4.7}$$

a constant! That is, the variance of $\sqrt{N}\bar{x}_N$ does not depend on N. Furthermore, we can scale this a little bit more so it has zero mean:

$$E\left[\sqrt{N}\left(\bar{x}_N - E[X]\right)\right] = 0 \tag{4.8}$$

$$V\left[\sqrt{N}\left(\bar{x}_N - E[X]\right)\right] = V[X] \tag{4.9}$$

OK, so now we know that, no matter how large or small the sample size, $\sqrt{N}(\bar{x}_N - E[X])$ will always have mean zero and variance equal to $V[X]$. This is *almost* useful. What is actually useful is the following:

Theorem 3 (Central limit theorem)**.** *Let X_i be an iid random variable with mean $E[X_i] = \mu$ and variance $V[X_i] = \sigma^2 < \infty$. Let:*

$$Z_N = \frac{\sqrt{N}\left(\frac{1}{N}\sum_i X_i - \mu\right)}{\sigma} \tag{4.10}$$

Then Z_N converges in distribution to a standard normal distribution as $N \to \infty$. that is:

$$\lim_{N\to\infty} \Pr[Z_N \leq z] = \Phi(z) \tag{4.11}$$

where $\Phi(z)$ is the standard normal cdf evaluated at z. Alternatively, we could write:

$$Z_N \xrightarrow{d} N(0,1) \tag{4.12}$$

That is, *no matter what the distribution of X is*, as long as it is iid with finite variance, we know that the sampling distribution of the mean approaches a normal distribution as $N \to \infty$. We then make the leap of faith that N is "close enough" to infinity that $\Pr[Z_N \leq z]$ is "close enough" to $\Phi(z)$ that it is not too terrible to assume that it is equal to $\Phi(z)$. Why is that useful? Perhaps I should re-iterate:

no matter what the distribution of X is ...

This means that if we want to say something about the sample mean, we hardly need to know anything about the distribution of the individual Xs. Only that they are (i) independent and identically distributed, (ii) finite variance, and (iii) the sample size is sufficiently large that this is a good approximation. That's it! Think about all of the hard work we put into working out a sampling distribution in the previous chapters. We needed to know the exact distribution of our Xs, and then we had to be lucky to find a monkey trick that got the distribution of the sample mean into a recognizable form. Now we only need to be able to do hypothesis tests and calculate p-values and confidence intervals using just one distribution: the standard normal! This thing can be summarized on a single sheet of paper, and in reality you will most likely need to memorize maybe two or three numbers to never need this piece of paper again.

4.3 Using large-sample properties to make inference easier

It is quite likely that all of the work needed to do inference in Chapter 3 made you wonder whether statistics and econometrics was always this hard. Fortunately, you now have a new tool that allows you to make a very useful shortcut. In this previous chapter, we spent a lot of time deriving the sampling properties of $\sum_{i=1}^{N} H_i$, the sum of N iid unfair coin flips, which came up heads ($H_i = 1$) with probability θ, and tails ($H_i = 0$) otherwise. If you've being paying attention in this chapter so far, you would have noticed that the WLLN and CLT told us things about sample means. Unfortunately, $\sum_{i=1}^{N} H_i$ is not a sample mean. Fortunately, if we divide by N, it is *exactly* a sample mean! Let $\bar{h} = \frac{1}{N}\sum_{i=1}^{N} H_i$As we could already do in Section 1.3, we now know that:

$$E[H_i] = 1 \times \theta + 0 \times (1 - \theta) = \theta \tag{4.13}$$

$$E[H_i^2] = 1^2 \times \theta + 0^2 \times (1 - \theta) = \theta \tag{4.14}$$

$$V[H_i] = E[H_i^2] - E[H_i]^2 = \theta - \theta^2 = \theta(1 - \theta) \tag{4.15}$$

Since $\bar{h}_N$ is a sample mean, by the WLLN, we know that $\text{plim}\bar{h}_N = E[H_i] = \theta$. Awesome! the more we flip the coin, the more likely we are to have a good estimate of θ.

4.3.1 Hypothesis tests with asymptotic approximations

Suppose again that you wish to test the following hypothesis:

$$H_0 : \ \theta = \theta_0, \quad H_A : \ \theta \neq \theta_0 \tag{4.16}$$

That is, you are ding a 2-sided test, with the null being that the true value of θ is θ_0 (i.e. if you were testing for a fair coin, you would substitute $\theta_0 = 0.5$). Before we go ahead and derive the sampling distribution for $\bar{h}_N$, which is what we would have done in the previous

chapter, let's substitute some of these properties of H_i into the Central Limit Theorem as stated in Theorem 3. Specifically, for this coin flip variable, when the null hypothesis is true, we know that:

- $\frac{1}{N}\sum_{i=1}^{N} X_i$ in this case is our sample mean $\bar{h}_N$
- μ, the population mean, is equal to θ_0, and
- σ, the population standard deviation, is equal to $\sqrt{\theta_0(1-\theta_0)}$

Here comes the part that will make inference *much* easier for you! Now we can define our test statistic as:

$$Z_N = \frac{\sqrt{N}(\bar{h}_N - \theta_0)}{\sqrt{\theta_0(1-\theta_0)}} \tag{4.17}$$

and by the Theorem, we know that $Z_N \xrightarrow{d} N(0,1)$. So we know the distribution of the test statistic when the null is true. Well ... actually we don't. We know the *asymptotic* distribution of this test statistic when the null is true, and we are going to assume that our sample size N is large enough that this asymptotic distribution is a good approximation of the actual distribution. If $N = 10$, it is probably a terrible assumption. $N = 10,000$? Go for it! Actually, have a good think first, but 10,000 is certainly much better than 10. There is no real rule of thumb (forget all of this $N = 30$ stuff you may have been taught in earlier classes RIGHT NOW) for what N is large enough, because it depends on the distribution of the random variable, but as you do more of this, you will probably have some intuition about when it is a good idea and when it's not.

So how do we use this? Well, we have (i) a null hypothesis and (2-sided) alternative; (ii) a test statistic, and (iii) a distribution (approximate) of this test statistic when H_0 is true. Suppose we want to do this test at the $\alpha = 0.05$ level of significance. All we are left with is finding a rejection rule. Inspection of Equation 4.17 shows us that $Z_N \approx 0$ is (loosely speaking) support for H_0, and Z_N far away from 0 in either direction is support for H_A. Hence, qualitatively, our rejection rule must therefore look something like "reject H_0 if Z_N is large and negative, or if Z_N is large and positive". We want to find some critical values that define this rejection rule. As this is a 2-sided test, if the null is true we want to reject H_0 with probability $\alpha/2 = 0.025$ in the left tail, and the same 0.025 in the right tail (i.e. so they add up to 0.05). Fortunately for us, the normal distribution is symmetric about zero, so we only need to look up one value. Intuitively, you might want to find the left critical value, which you get by solving $0.025 = \Phi(z_{cL})$, where $\Phi(\cdot)$ is the standard normal cdf. Unfortunately, this is not provided in standard distribution tables,[1] but we solve for the right critical value $1 - 0.025 = 0.0975 = \Phi(z_{cR})$. Then we can use symmetry to get $z_{cL} = -z_{cR}$. If you go and look up your distribution tables, you will get $z_{cR} \approx 1.96$ (accurate to 2 decimal places, which

[1] Although you can always use your computer. For example, in *Stata*: `display invnormal(0.025)` returns `-1.959964`

is almost always good enough). Hence, the rejection rule is:

$$\text{Reject } H_0 \text{ if and only if } |Z_N| \geq 1.96 \tag{4.18}$$

To put this in perspective, suppose that you are doing a test for a fair coin: $\theta_0 = 0.5$, this means that your rejection region, in terms of your sample mean h_N, is equal to:

$$\frac{\sqrt{N}(\bar{h}_N - 0.5)}{\sqrt{0.5(1-0.5)}} = 2\sqrt{N}(\bar{h}_N - 0.5) \tag{4.19}$$

$$|Z_N| > 1.96 \iff 2\sqrt{N}|\bar{h}_N - 0.5| > 1.96 \tag{4.20}$$

$$\iff |\bar{h}_N - 0.5| > \frac{1.96}{2\sqrt{N}} \approx \frac{1}{\sqrt{N}} \tag{4.21}$$

4.3.2 Even more of a shortcut

At this point, you may be worried that even though getting an approximate distribution of the test statistic is really useful, you might still be stumped because you also need to know $V[X]$ to use this. What if you want to do a test about a mean, but you don't know what the variance is (or can't be bothered working it out, I won't judge), and don't want your test to based on a bad assumption about this thing $V[X]$, that is not central to your research question?

Let's define $D_i = (X_i - \mu)^2$, which is the squared deviation between our random variable X_i and its population mean μ. We could always generate this variable if we already had X, and after this, we could compute its sample mean. If we did this, we would be computing:

$$\bar{d}_N = \frac{1}{N}\sum_{i=1}^{N} D_i = \frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2 \tag{4.22}$$

which is the sample analog of $E[D_i] = E[(X_i - \mu)^2] = V[X_i]$, and hence by the WLLN $\bar{d}_N$ must converge in probability to $V[X_i]$. Since we have already assumed N is large enough that our test statistic is close enough to $N(0,1)$ that we can use it, there is not much more harm, if at all, in assuming that $\bar{d}_N$ is close enough to $V[X]$ to use it as a substitute for the denominator of the test statistic.[2] What's more, we could also replace μ with $\bar{x}_N$, the sample mean, because $\text{plim}\bar{x}_N = \mu$ (WLLN). If we make these substitutions, what we end up with is:

$$Z'_N \frac{\sqrt{N}(\bar{x}_N - \mu)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{x}_N)^2}} \xrightarrow{d} N(0,1) \tag{4.23}$$

which contains things that we either (i) can compute from the sample, or (ii) are directly making a hypothesis about. Note that the numerator is the square root of the sample

[2]There are some results about the relationship between probability limits and asymptotic distributions that I am not going in to here, but they work in our favor.

variance. Usually we would divide by $N-1$ instead of N for bias reasons. There's nothing wrong with that, but note that (i) $\frac{1}{N} \approx \frac{1}{N+1}$ for large N, which we have already assumed, and (ii) we are taking the square root of the thing, so even if you divide by $N-1$, the thing will still be biased (look up Jensen's inequality). With the formulation in (4.23), we don't even need to know the relationship between μ and $V[X]$. In practice, this is the one we will be using.

4.3.3 Confidence intervals with asymptotic approximations

Confidence intervals with asymptotic approximations are, like hypothesis tests, exactly the same as confidence intervals without asymptotic approximations, except that we use an approximate distribution of the test statistic instead of an exact distribution. For our coin-flipping example, in working through the hypothesis test example above, we have already worked out that Z_N in (4.17) is approximately distributed $N(0,1)$ when the null hypothesis is true. Going back to the previous chapter, we know that confidence intervals ask the following question: For what values of θ_0 would I fail to reject the null hypothesis? We have already worked out the rejection rule for our 5% test, so we would fail to reject the null whenever $|Z_N| \leq 1.96$, or when:

$$Z_N = \frac{\sqrt{N}|\bar{h}_N - \theta_0|}{\sqrt{\theta_0(1-\theta_0)}} \leq 1.96 \tag{4.24}$$

which is somewhat of a headache to solve. However now we will take off our statistics and econometrics hats, and put on our math hat. Note that if N is reasonably large (again, we've assumed this already, so yes, it is), we suspect that this confidence interval will be reasonably small. Therefore, we will make the same additional approximation that got us to (4.23), and instead look for solutions to:

$$Z'_N = \frac{\sqrt{N}|\bar{h}_N - \theta_0|}{\hat{\sigma}} < 1.96, \quad \text{where: } \hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(H_i - \bar{h}_N\right)^2} \tag{4.25}$$

$$|\bar{h}_N - \theta_0| < 1.96\frac{\hat{\sigma}}{\sqrt{N}} \tag{4.26}$$

$$\theta_0 \in \left[\bar{h}_N - 1.96\hat{\sigma}/\sqrt{N},\ \bar{h}_N + 1.96\hat{\sigma}/\sqrt{N}\right] \tag{4.27}$$

hopefully this is starting to become a bit more familiar.

4.3.4 p-values with asymptotic approximations

Again, we're not doing much differently with p-values in this chapter, we're just looking up a different distribution. For a p-value, our question is: what is the probability that we observed a test statistic providing at least as unfavorable support for the null hypothesis

than the one we observed in the sample? We therefore want to know the probability:

$$\Pr\left(\left|\frac{\sqrt{N}(\bar{h}_N-\theta_0)}{\sqrt{\theta_0}(1-\theta_0)}\right|\geq z_N\right) \tag{4.28}$$

$$=\Pr\left(\frac{\sqrt{N}(\bar{h}_N-\theta_0)}{\sqrt{\theta_0}(1-\theta_0)}\geq |z_N|\right)+\Pr\left(\frac{\sqrt{N}(\bar{h}_N-\theta_0)}{\sqrt{\theta_0}(1-\theta_0)}\leq -|z_N|\right) \tag{4.29}$$

$$=1-\Phi(|z_N|)+\Phi(-|z_N|) \qquad \text{note that these are standard normal cdfs} \tag{4.30}$$

$$=1-\Phi(|z_N|)+(1-\Phi(|z_N|)) \qquad \text{the standard normal is symmetric} \tag{4.31}$$

$$=2(1-\Phi(|z_N|)) \tag{4.32}$$

where z_N is the realized value of our test statistic we computed in our sample. The last few lines get the expression into something we can look up in standard probability tables. Typically you are given the cdf for positive numbers only, then you have to use symmetry to get the number you want.

4.4 Transforming variables

So now you know a lot about sample means. Great! A lot of things can be estimated using sample means. Unfortunately, sometimes the mean isn't the thing you are directly interested in. Instead, you want to report a *transform* of the sample mean. To tie things in with our coin-flipping example, suppose that instead of wanting to report an estimate of the probability of the coin coming up heads (i.e. θ), you want to report how much more likely it is to flip heads than tails. The population quantity you want to report is therefore:

$$\rho\equiv\frac{\theta}{1-\theta}=\frac{\text{probability of flipping heads}}{\text{probability of flipping tails}} \tag{4.33}$$

That is if ρ is (say) two, this means that flipping heads is twice as likely as flipping tails (i.e. $\theta=2/3$). From here it seems reasonable to use the estimator:

$$\hat{\rho}=\frac{\hat{\theta}}{1-\hat{\theta}} \tag{4.34}$$

That is, just use our estimator for θ, and transform it in the same way you transformed the population parameter θ. After all, $\hat{\theta}$ is a sample mean (earlier we worked out that it was the fraction of heads), and we know a lot about sample means. In this case:

$$E[\hat{\theta}]=\theta, \quad \text{i.e. it is unbiased} \tag{4.35}$$

$$V[\hat{\theta}]=\frac{1}{N}\theta(1-\theta) \tag{4.36}$$

$$\text{plim}\hat{\theta}=\theta \quad \text{i.e. it is consistent} \tag{4.37}$$

$$\sqrt{N}(\hat{\theta}-\theta)\xrightarrow{d}N(0,\theta(1-\theta)) \tag{4.38}$$

We know all of these things because $\hat{\theta}$ is a sample mean, but $\hat{\rho}$ is *not* a sample mean. Is it consistent? Biased? Can we put a confidence interval around it? To answer these questions, we will need the following results.

4.4.1 The continuous mapping theorem

In order to answer the consistency question, we will use the following result:

Theorem 4 (Continuous mapping theorem, loosely stated)**.** *If* $\text{plim}\hat{\theta} = \theta$*, and* $g(x)$ *is a continuous function, then* $\text{plim}g(\hat{\theta}) = g(\theta)$*.*

in words: If an estimator $\hat{\rho} = g(\hat{\theta})$ is a continuous transformation of a consistent estimator $\hat{\theta}$, then $\hat{\rho}$ is a consistent estimator of $\rho = g(\theta)$. Basically, we're done. We know that $\hat{\theta}$ is consistent (by the WLLN), and we want to report a continuous transform of it. Therefore $\hat{\rho}$ is a consistent estimator for ρ.

4.4.2 The delta method

Now we know that we have a nice, consistent point estimate of ρ. But now we want to put a confidence interval around it. To do that, we need to know, or approximate it. To do this, we will use the Delta method, which uses the following result:

Theorem 5 (The delta method, univariate case)**.** *Let*

- $\hat{\theta}$ *be a consistent estimator with a normal asymptotic distribution* $\sqrt{N}(\hat{\theta} - \theta) \xrightarrow{d} N(0, V)$*, and*
- $g(\theta)$ *be a continuous function with a continuous first derivative* $g'(\theta)$*.*

then:

$$\sqrt{N}(g(\hat{\theta}) - g(\theta)) \xrightarrow{d} N\left(0, (g'(\theta))^2 V\right) \tag{4.39}$$

Why is this useful? If $\hat{\theta}$ is asymptotically normal, we can work out the asymptotic distribution of $g(\hat{\theta})$, in our case $\hat{\rho}$. For our example, we need the derivative of g:

$$g'(\theta) = \frac{\partial}{\partial\theta}\frac{\theta}{1-\theta} \tag{4.40}$$

$$= \frac{1-\theta+\theta}{(1-\theta)^2} = \frac{1}{(1-\theta)^2} \tag{4.41}$$

And so, substituting the particulars of our coin flip estimator $\hat{\theta}$, namely $V = \theta(1-\theta)$, into (4.39):

$$\sqrt{N}(\hat{\rho} - \rho) \xrightarrow{d} N\left(0, \frac{\theta(1-\theta)}{(1-\theta)^4}\right) = N\left(0, \frac{\theta}{(1-\theta)^3}\right) \tag{4.42}$$

Hence, a 2-sided, 95% confidence interval for ρ would be:

$$\left[\frac{\hat{\theta}}{1-\hat{\theta}} - 1.96\sqrt{\frac{\theta}{(1-\theta)^3 N}},\ \frac{\hat{\theta}}{1-\hat{\theta}} + 1.96\sqrt{\frac{\theta}{(1-\theta)^3 N}} \right] \tag{4.43}$$

4.4.3 Jensen's inequality

We have established that our transformed estimator $\hat{\rho}$ is consistent, and we have worked out an approximation of its sampling distribution. These are some nice large sample properties to know, but what about bias? The original estimator $\hat{\theta}$ is unbiased because it is a sample mean. What about $\hat{\rho}$. Is it, too, unbiased? Sadly, the answer is no: no specifically in this case, and no in general. This follows from Jensen's inequality, which states that:

Theorem 6 (Jensen's inequality)**.** *If $g(x)$ is a convex function, and X is a random variable, then $g\left(E[X]\right) \leq E\left[g(X)\right]$.*

In words: the function of the expectation of a random variable is less than the expectation of the function of the random variable. Conversely, if g is a concave function, the direction of the inequality is reversed. Is our $g(x) = x/(1-x)$ concave of convex? Since it is differentiable, we can use the 2nd derivative to work it out:

$$g'(x) = \frac{1}{(1-x)^2}, \quad g''(x) = 2(1-x)^{-3} > 0 \tag{4.44}$$

So it is convex. Hence $E[\hat{\rho}] < \rho$, so the estimator is biased. It is not all lost, though. Firstly, we know the direction of the bias, so that is somewhat helpful. Also, we have already established that $\hat{\rho}$ is a consistent estimator, so for large samples this is not so much of a problem.

Exercises

Exercise 4.1.
Consider the exponential distribution, which has the following properties:

$$F_X(x) = 1 - \exp(-\lambda x) I(x > 0) \tag{4.45}$$
$$f_X(x) = \lambda \exp(-\lambda x) I(x > 0) \tag{4.46}$$
$$E[X^k] = \frac{k!}{\lambda^k} \tag{4.47}$$

It can be used to model the time until an event occurs. We will consider the following two estimators for λ:

$$\hat{\lambda} = \frac{1}{\frac{1}{N}\sum_{i=1}^{N} X_i}, \quad \tilde{\lambda} = \sqrt{\frac{2}{\frac{1}{N}\sum_{i=1}^{N} X_i^2}}$$

1. Explain how these estimators can be motivated from Equation 4.47 above.

2. Load `ExpData.csv`, which is a dataset of exponential random numbers. Estimate λ using both estimators described above.

3. Are these consistent estimators for λ? Explain.

4. What are the asymptotic variances of these estimators? Which one would you prefer?

5. Suppose that you wanted to report the probability that the event had not occurred after 1 unit of time. Write down this expression as a function of λ.

   From now on, let's just focus on $\hat{\lambda}$, although you could do all of this with $\tilde{\lambda}$ as well.

6. Replace λ with $\hat{\lambda}$ in your answer to the previous part. This is an estimator of this probability. Is this estimator consistent?

7. What is the asymptotic distribution of the estimator of this probability?

8. Construct a 95% confidence interval for this probability.

9. Is $\hat{\lambda}$ a biased estimator for λ? Explain. If it is biased, can you say in which direction?

Exercise 4.2 (Simulation exercise – What is so magical about $N = 30$?)**.**
In an undergraduate statistics course you may have been told that you need a sample size of at least 30 to justify looking up a normal distribution table. Plot the distribution of the test statistic t, and evaluate the test size for the hypothesis test:

$$H_0: \ E[X] = 0, \quad H_A: \ E[X] \neq 0$$

$$t = \frac{\sqrt{N}\bar{X}}{\sqrt{\frac{1}{N}\sum_i (X_i - \bar{X})^2}}$$

$$\text{reject } H_0 \text{ if and only if } |t| > 1.96$$

assuming that:

1. $X_i \sim iidN(0, 1)$

2. $X_i \sim iidN(0, 4)$

3. $X_i \sim iid\text{Bernoulli}(0.5) - 0.5$ (you can draw this by generating a fair coin flip variable then subtracting 0.5 from it).

4. $X_i \sim iid\chi^2_1 - 1$. Note that if $Z \sim N(0, 1)$, then $X^2 \sim \chi^2_1$

Note that once you have generated `X` from the correct distribution, you can compute `t` as follows:

```
summarize X
display t = sqrt(_N)*r(mean)/sqrt(r(Var))
```

within a program, you will want to replace `display` with `return scalar`.

Exercise 4.3.
Consider the distribution first introduced in Exercise 1.3. We will continue analyzing the properties of the following estimators for the parameter in this distribution, which were introduced in Exercise 2.1:

$$\hat{\alpha} = \frac{\frac{1}{N}\sum_{i=1}^{N} X_i}{1 - \frac{1}{N}\sum_{i=1}^{N} X_i}$$
$$\tilde{\alpha} = -\frac{N}{\sum_{i=1}^{N} \log(X_i)}$$

1. For each of these estimators, answer do the following questions. For simplicity, I refer to everything below as $\hat{\alpha}$, but do this for both $\hat{\alpha}$ and $t\tilde{\alpha}$.

   Hint: for $\tilde{\alpha}$, you will need to do some integration by parts, then use L'Hôpital's rule. Either that or use something like this`https://www.wolframalpha.com`

   (a) Is $\hat{\alpha}$ a consistent estimator for α? Explain your answer.

   (b) What is the delta method approximation of the variance of $\hat{\alpha}$?

   (c) Suppose that you wish to test:

   $$H_0 : \ \alpha = 2$$

   against:

   $$H_A : \ \alpha \neq 2$$

   Under the null hypothesis, what is the asymptotic (large sample) distribution of $\sqrt{N}(\hat{\alpha} - 2)$? That is, complete the right-hand side of:

   $$\sqrt{N}(\hat{\alpha} - 2) \xrightarrow{d} \ ?$$

   (d) Use your answer in the previous part to propose a function of $\hat{\alpha}$ and N that at large enough samples is approximately distributed $N(0, 1)$

   (e) Suppose that you collected $N = 30$ observations and estimated $\hat{\alpha} = 2.2$. Use your answer in part 1d to test this hypothesis. Use a 5% level of significance.

   (f) What is the p-value for this test?

(g) Construct a 2-sided 90% confidence interval around this point estimate.

2. Based on their asymptotic variances alone, which estimator would you prefer to use?

3. Propose an alternative method of testing $\alpha = 2$ using the sample mean instead of our estimate of $\hat{\alpha}$ (just outline the steps). Make sure you state the distribution of the test statistic under the null, if you made a large-sample approximation to get there, and the rejection rule.

4. (Simulation exercise): In Exercise 4.3, question 1e, you constructed a rejection rule for H_0 based on a large-sample approximation of the distribution of $\hat{\alpha}$. Use a simulation to construct a rejection rule that does not need this approximation. Compare it to your large-sample approximation rejection rule. Do you think the probability of a Type II error using the large-sample approximation is close enough to 5% to be a good approximation? Briefly discuss your answer.

   Hint: To do this, you should:

   - Simulate the distribution of $\hat{\alpha}$ when the null hypothesis is true
   - Calculate two critical values (i.e. reject if $\hat{\alpha}$ is not between these critical values) from your simulated distribution. Think about how you calculated your critical values when you used the large sample approximation, and how they relate to the normal distribution.

Exercise 4.4 (A sort-of simulation exercise)**.**
When we simulate a draw from the distribution of an estimator, say $\hat{\mu}$, one thing we may want to ask is how accurate is our approximation of the bias? That is, how close is the simulated bias:

$$\text{Bias}^{\text{S}}(\hat{\mu}) = \frac{1}{S}\sum_{s=1}^{S}(\hat{\mu}^s - \mu)$$

to

$$\text{Bias}(\hat{\mu}) = E\left(\hat{\mu} - \mu\right)$$

Assume that we have correctly simulated $\{\hat{\mu}^s\}_{s=1}^{S}$, such that each $\hat{\mu}^s$ is an iid draw from the sampling distribution of $\hat{\mu}$.

1. What is the variance of our simulated bias?

2. What is the approximate distribution of:

$$\sqrt{S}\left(\text{Bias}^{\text{S}}(\hat{\mu}) - \text{Bias}(\hat{\mu})\right)$$

   for large S?

3. How can you use your previous answer to work out how accurate your simulation is?

4. Assume that $V[\hat{\mu}] = 1$. How large doe S be for your simulation to get the bias correct to the 2nd decimal place with probability 99%?

Express your answers as a function of the actual bias, the simulation size S, $E[\hat{\mu}]$ and $V[\hat{\mu}]$ (assuming these are all finite quantities).

Exercise 4.5 (One test, three ways)**.**
You wish to determine whether a randomly selected dime in the population of dimes is fair or unfair. To this end, you decide to crowd-source your coin-flipping activities. Specifically, you reach out to 10 random individuals on the internet and ask them to flip a dime 200 times, and report to you the fraction of heads that they flipped. Let $H_{i,t}$ be the binary random variable, equal to one if person i's tth flip is heads, and equal to zero otherwise. Your sample therefore consists of $\{H_i\}_{i=1}^{10}$, where $H_i = \frac{1}{200}\sum_{t=1}^{200} H_{i,t}$ is the fraction of heads that random individual i flipped. Your sample $\{H_i\}_{i=1}^{N}$ is contained in `dimeflips.csv`.

1. State a formal hypothesis that the average dime is fair, that is testable with your sample.

2. Propose a justification for why H_i is close enough to normally distributed (if the null is true) for us to reasonably assume that it is. State any other assumptions you need to get there. Derive a function of H_i and T that is approximately $N(0,1)$.

3. Use your answer to 2, and the following result: Use the following result

   If $X_i \sim iidN(\mu, \sigma^2)$, then:

   $$t = \frac{\sqrt{N}(\bar{x} - \mu)}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{x})^2}} \sim t_{N-1}$$

   where t_k is Student's t distribution with parameter k (often referred to as the "degrees of freedom").

   to suggest a suitable test statistic that has a t distribution (approximately) when the null is true. Calculate the p-value of this test. *Hint:* `help t`.

4. Use your answer to 2, and the following result:

   if $Z_1, Z_2, Z_3, \ldots, Z_T \sim iidN(\mu, \sigma^2)$, then $X = \sum_{t=1}^{T} Z_t \sim N(T\mu, T\sigma^2)$.

   to suggest a suitable test statistic that has a normal distribution (approximately) when the null is true. Perform this test at the 5% level of significance.

5. Use your answer to 2, and the following result:

if $Z_1, Z_2, Z_3, \ldots, Z_N \sim iidN(0,1)$, then $X = \sum_{i=1}^{N} Z_i^2 \sim \chi_N^2$. Where χ_k^2 is the chi-squared distribution with parameter k (often referred to as the "degrees of freedom")

to suggest a suitable test statistic that has a χ^2 distribution (approximately) when the null is true. Perform this test at the 5% level of significance. *Hint:* `help invchi2`.

Exercise 4.6.
This exercise covers a large range of things that you might want to find out about, or do with, an estimator using (mostly) large sample properties. If you are having trouble with a particular part, I suggest doing that part for all five distributions in the table below *in one go*. If you are confident with all of the steps, attempt working through all of them for one distribution.

Consider the following distributions (you are just given the mean and variance, but you won't need any more information):

Distribution	$E[X]$	$V[X]$	notes	Sample mean	H_0
Poisson	λ	λ	$\lambda > 0$	1.1	$\lambda = 1$
Exponential	λ^{-1}	λ^{-2}	$\lambda > 0$	1.1	$\lambda = 1$
χ^2	k	$2k$	$k > 0$	2.8	$k = 3$
Borel	$\frac{1}{1-\mu}$	$\frac{\mu}{(1-\mu)^3}$	$\mu \in (0,1)$	2.5	$\mu = \frac{1}{2}$
Geometric	$\frac{1}{\rho}$	$\frac{1-\rho}{\rho^2}$	$\rho \in (0,1)$	1.9	$\rho = \frac{1}{2}$

For each of these distributions:

1. Suppose you had an iid sample from this distribution, what is the asymptotic distribution of the sample mean? I.e.:

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} X_i - E[X]\right) \xrightarrow{d} ?$$

   That is, use a central limit theorem.

2. You wish to test the hypothesis in the rightmost column of the table, against a 2-sided alternative. Use your a answer to the previous question to construct a test statistic for this hypothesis that is $N(0,1)$ when the null is true. Do not use the sample variance to construct your test statistic.

3. State the rejection region for this test at the 5% level of significance.

4. Use the sample mean provided in the table to test this hypothesis.

5. Assign a p-value to this hypothesis.

   [what follows is somewhat trivial for the Poisson and χ^2 distributions]

6. Construct an estimator for the parameter in the distribution based on the relationship between the parameter and the population mean (i.e. an analogy estimator).

7. Use the sample mean provided to estimate this parameter.

8. Is this a consistent estimator for the parameter? Explain your answer.

9. What is the delta method approximation of the variance of your estimator? Note that since you don't know the parameter, you will have to use your estimate of this parameter in your expression for the variance.

10. Use this approximation to construct a 95% confidence interval for the parameter.

11. Is this estimator biased? Explain your answer. If the estimator is biased, can you work out the direction of the bias?

Part II

Basics of programming and handling data in *Stata*

Chapter 5

Getting started in *Stata*

5.1 Importing, saving, and exporting data

The most likely reason that you open up *Stata* is that you want to analyze some data. An important first step is therefore knowing how to import your file. Basically, you need to give *Stata* some instructions along the lines of "Go to this folder, and open this file." On top of this, you will also need to tell it the file format. While you can look at most of the datasets that *Stata* can open using a text editor,[1] you need to tell it the "language" it should be expecting. The bad news is that *Stata* will not be your friend if you don't get this right, the good news is that if you *do* get it right, *Stata* is able to open quite a lot of stuff (with the right instructions).

You can tell *Stata* to import data through the `File→Import` drop-down menu. In addition, `Stata` also has its own file format with the suffix `.dta`. You can import this file format using the `use` command. While this format usually preserves much more useful information than the other formats,[2] it has its drawbacks, too. First, it is difficult to read in programs other than *Stata*; and second, there are sometimes compatibility issues between versions of Stata.[3] The `.dta` format can be produced using the drop-down menus `File→Save` or `File→Save As...`.

Note that all of these functions can be accessed through the command line. For example, if you have a file `thingy.dta` in your working directory, instead of `File→Open...` you could type `use thingy`, or if you already have something in memory: `use thingy, clear`. Exercise 5.2 is there to help you get more acquainted with how *Stata* stores data in different formats.

[1] *MS Windows*' *Notepad* is a text editor, but it is not a particularly good one. I have found that *Notepad++* works pretty well for almost everything.

[2] My favorites of these are value and data labels, which make using someone else's data *much* easier.

[3] At the time of writing, I had recently encountered a problem with opening files in *Stata* 13 that were saved in *Stata* 14. This can be solved at the newer version end with the `version` command, but it is annoying, nonetheless.

5.2 Scripts

Almost everything that comes with *Stata* can be done using the drop-down menus. This seems like a great comfort at first, but I warn you against using *Stata* like this for (at least) four reasons:

1. The "almost" part: there are some things that you will not be able to access using the drop-down menus.

2. The "that comes with *Stata*" part: *Stata* has a lot of really good and free user-generated content.[4] These typically are not friendly to those who like to point and click.

3. If you always use the drop-down menus, and you ever want to change what you do (you should expect to do this *all the time*), then you will have to re-trace all of your steps. Will you remember them? Will you have time to do them? Will you get the changes right on the first try? Probably not (no offense).

4. Pedagogically, I want you to learn something about programming *in general*, as well as in *Stata* specifically. Learning about some fundamental components of programming languages (such as scripts, `for` loops, and `if` statements) will make any programming language easier to learn in the future.

In addition to not using the drop-down menus, I also encourage you to not use the command line for anything you think you might want to keep. This leaves us with the following solution:

It is with a keen sense of irony that I invite you to use the drop-down menus in *Stata* as follows: `Window` → `Do-file Editor` → `New Do-file Editor`. Alternatively, `ctrl+9` will also get you there. This opens up a new Do-file editor (duh). This is `Stata`'s in-built text editor for scripts, which are set of instructions that `Stata` follows from top to bottom. These things are extremely useful for many reasons. So much so, that from now on you should think about doing everything in *Stata exclusively* in scripts.

To demonstrate how to get started on a script, let's have a look at *Stata*'s system dataset `auto.dta`.[5] We can access this and see what's in it by typing the following lines into the command line:

```
clear
sysuse auto
describe
```

`clear` removes any data that was in memory beforehand (make sure you save regularly!), `sysuse auto` loads this dataset, and `describe` gives us a description of all of the variables held in memory. One of the variables we have is `mpg`, which is each cars' fuel efficiency, in

[4] E.g. `esttab`, which you will become familiar with shortly.

[5] *Stata* comes with a number of small, pre-loaded datasets that can be accessed with the `sysuse` command. These are frequently used in the help files as examples to demonstrate particular programs. There are also other datasets on *Stata*'s website that can be accessed using the `webuse` command.

Variable	Obs	Mean	Std. Dev.	Min	Max
mpg	74	21.2973	5.785503	12	41
Lper100km	74	11.83116	3.016803	5.749129	19.64286

Table 5.1: Resultant summary of `auto` dataset after generating `Lper100km` variable.

miles per gallon. Suppose that, for whatever reason, we wish to do our analysis in liters per 100km instead. This is easily done by generating a variable:

```
generate Lper100km = 1/(mpg*1.61/3.795)*100
```

That is, there are 1.61km in a mile, 3.795L per US gallon, and this new unit is based on 100L of fuel:

$$\frac{3.795\text{L/gallon}}{\text{mile/gallon} \times 1.61\text{km/mile}} = 2.35 \times \text{L/km} \times \frac{100}{100} = 235\text{L/100km} \tag{5.1}$$

So before we ever want to do analysis on the `auto` dataset using these units for fuel efficiency, without scripts we would have to type in all of these lines. This could become tedious if we have to do this over and over again, so alternatively, you could write the script:

```
clear                              // clear everything in memory
sysuse auto                        // load the system dataset called "auto"
describe                           // provide a description of the dataset
generate Lper100km = 1/(mpg*1.61/3.795)*100     // convert mpg into L per 100km
summarize mpg Lper100km            // provide summary statistics for mpg and Lper100km
```

which works in exactly the same way as typing stuff into the command window, but you only have to type it once. This code also produces a table of summary statistics of the old and new variable, which is shown in Table 5.1.

5.3 The working directory

In many popular MS programs, whenever you save something for the first time you are asked where you would like the file to be stored. This is a problem if you are using scripts, for at least these reasons:

1. If your script is a long one, then if you are like me you probably want to go and do something else while it runs. Go get coffee, grade a paper, mindlessly check Facebook, etc. If your script has multiple lines telling *Stata* to write files, you don't want it interrupting you whenever it needs to do this with the question "where do you want me to put this?"

2. If you are collaborating with others (which I strongly encourage), your script will need to run on many (at least two) machines, which all have different file systems.

The solution to these problems is to point *Stata* in the direction of a "working directory". This is the default folder that, if prompted to `save`, `export`, `import`, `use`, or read or write anything else for that matter, it will do it here by default.

You can ask *Stata* about the working directory through the command line by typing `dir`. This gives you the file path, e.g. `C:/users/Kryten/Metrix`, as well as information about any files and folders that are in this directory. You can go up one level by typing `cd ..`, which in this example would get you to `C:/users/Kryten`. You could go back to the original by typing either `cd Metrix`, or the whole file path: `cd C:/users/Kryten/Metrix`.[6] In fact, this second command will get you there no matter what the current working directory is.

Furthermore, by default *Stata* assumes that any incomplete file path that you give it starts at the working directory. So suppose that you had a folder called "figures" inside your working directory (I almost aways do), then you could tell *Stata* to save a histogram of variable `x` as follows:

```
hist x
graph export figures/HistogramOfX.png
```

Exercises

Exercise 5.1.

1. Create a new folder somewhere, and copy "`galton_heights.csv`" into it
2. Stata needs to know which folder you want it to look in by default. This is called the "working directory". You can set this by clicking on `File --> Change Working Directory` and following the prompts.
3. Type `ls` into the command window. Stata will tell you the contents of this folder. If you're ever unsure what your working directory is, type `pwd` (print working directory)
4. Stata needs to know a bit about the data file you want it to read. `galton_heights.csv` is a comma separated variable file. Open it with NotePad or something similar to find out what that means.
5. Today we will import our file from the command line. It *can* be done through `File --> ...` as well, but some things cannot be done with the drop-down tabs. The command we will use is `import delimited`, but we need to know the information Stata needs. To do this, type "`help import delimited`" into the command line. A help file will pop up. Get to know how to read these help files. Once you do, they are actually quite helpful! Work out what you need to type to import your data (I will run through this with you once you've tried yourself)

[6] If there is a space in the file path, you will need to put double quotes around it, e.g.: `cd "C:/this folder has spaces"`

6. In the command line, try typing (one at a time) "`describe`", "`summarize`", and "`summarize, detail`". What do these tell you? Have a look at the help files for `describe` and `summarize` Can you summarize only two of the five variables in the data set? (read the help files).[7]

7. What is wrong about the summary statistics for mothers and fathers? (think about how the data are arranged)

8. The height variables are recorded in inches (reasonable in the US), but for some reason 60 inches are subtracted. Create 3 new variables equal to the actual heights:

   `generate father_height_actual = father_height + 60` If you want to use a system of measurements that the majority of the world uses:

   `generate father_height_meters`

   `= father_height_actual / 39.37`

9. Check your work with a scatter plot. If you created these variables correctly, what would the relationship between `child_height_actual` and `child_height` be? With the drop-down menus, use: `Graphics --> Twoway graph (scatter, line, etc.)`

10. Look at the text that just appeared in the output. Copy and paste it into the command line. Same results? Good!

11. Try the following, what do they do?

    ```
    twoway (scatter child_height_actual mother_height_actual)
    ```

    ```
    twoway (scatter child_height_actual mother_height_actual if son==0)
    ```

    ```
    by son, sort:  twoway (scatter child_height_actual mother_height_actual)
    ```

12. Suppose that you wanted to get rid of some data:

    I don't need to use `family_id`, then: `drop family_id`

    I only want to focus on daughters, then: `drop if son==1`

13. Once you're done, you can save the data (but not the outputs) by typing:

    `save filename.dta`

    `dta` is a special file format for Stata that allows you to store some additional information (more on this later).

    To load a `dta` file: `use filename`

    If you already have data in the memory, you will need to clear it first:

[7] Note that Stata lets you take a few shortcuts: you could have types `desc` and `sum` respectively. I will try to not take the shortcuts, but if you use Stata enough you will probably always use them. I think this is a drawback of Stata. Having exactly one way to do things makes it easier to read others' programs. If you use Python, have a think about what a tab means.

```
clear
use filename
```

Exercise 5.2.
Load one of the system datasets (e.g. `sysuse auto, clear`), then do the following:

1. Type `describe` and `summarize` into the command line. What do these commands do?
2. Use the drop-down menus to export this dataset into the following formats:
   (a) Text data in csv format
   (b) Text data in fixed format
   (c) *Stata*'s `.dta` format.

   What commands does *Stata* write into the command line when you export these files?
3. In a good text editor (e.g. *Notepad++*). Open up the two files you created in the previous question. In each of these files, how does *Stata* know when one column ends, and another begins?
4. Import the `.csv` and `.dta` that you created, and `describe` the data. Has any information changed or been destroyed compared to the system dataset?

Exercise 5.3.

1. Find out where *Stata*'s default working directory is on your machine. I.e. what is ther working directory when you open *Stata*?
2. Create a folder somewhere. Write a script that, no matter where it is located, will export a comma-separated variable version of *Stata*'s system dataset `auto` to this folder.
3. What do you need to do to make this script run without errors twice?

Exercise 5.4.
Unpack `ExUnderstandingStata.zip` into a folder on your hard drive.[8] Within this file structure there should be a folder called `ExUnderstandingStata`, which contains three folders called `Code`, `Data`, and `Outputs`. Open up `CommentThis.do` in *Stata*'s do file editor.

1. Near the top of this script is a line that says something like:

```
cd "C:\Users\jbland\Dropbox\MetrixShare\StataLectures\CH05GettingStarted\
    ↪ ExUnderstandingStata"
```

[8] I don't need you to understand the dataset used in this exercise, but in case you are interested, there is a description of the data here: `https://vincentarelbundock.github.io/Rdatasets/doc/carData/Cowles.html`

which won't work unless you have hacked into my office machine. Change this line so that it points to the `ExUnderstandingStata` folder on *your* machine.

2. Put a comment above *every* line in this script that describes what that line does. If you are having trouble working this out, try commenting out the line, using *Stata*'s help files (e.g. for the above line, I would type `help cd`), or using an internet search (include "Stata" and the command in the search terms).

3. Answer the following questions (include them in the comments for the relevant line of code):

   (a) Explain what "storage type" is in the output that `describe` produces.

   (b) On the line that starts with `graph export`, what does `replace` do? (*hint:* comment out `replace` and try running the script twice).

   (c) What do the `jitter` and `msize` options do?

Chapter 6

For loops

Sometimes we find ourselves copying blocks of code over and over again, and only making minor changes to each copy. In econometrics, this can often involve running the same analysis on several different datasets, systematically running every possible combination of things from two or more sets of things,[1] or performing the same transformation of every variable in our dataset.[2] While there is no reason why your carefully copied and pasted script that is hundreds of lines long *could* work, it pays to take a pause whenever you get the urge to do this, and think about writing a `for loop`. Firstly, if there is an error in the block of code that you are copying, then there will be hundreds of errors after you `ctrl+v` 99 times.

The idea of a `for` loop is as follows:

```
Hey, STATA! Do this thing in the curly brackets  for x = 1, 2, 3, ..., K {
        Here is the thing that I want you to do over and over again
        I want each iteration to be a little bit different, so I can
        include ‘x’ in here to distinguish between each step
} // here is the end of the thing I want you to do
```

Of course, *Stata* isn't intelligent enough to understand this, but it can do something quite similar, if you give it the right instructions. For example:

```
1 forvalues ii = 1/5 {
2         display ‘ii’
3         }
```

which displays the numbers 1 through 5. While this example is somewhat underwhelming, note that in order to achieve this without using `forvalues`, I would have to:

```
1 display 1
2 display 2
3 display 3
4 display 4
5 display 5
```

[1] E.g.: Running an analysis for men and women separately, slicing the data by level of education.

[2] E.g.: We might want to take the natural log of all of all of our variables.

which is 2 lines longer. More to the point, if I wanted the numbers 1 through 1,000 displayed, the `for` loop would still be 3 lines long (all you would have to do is change the 5 line 1 to 1000), and the second script would be 1,000 lines long.

Listing all of the integers between 1 and 1,000 may be a special moment for you in learning about programming, but it is not particularly useful. One application that is somewhat more econometrics-y, is reporting a sample mean. In my working directory at the moment, I have 10 files (unimaginatively) called `data_1.dta`, `data_2.dta`, ..., `data_10.dta`. Each of these contains exactly one variable, called `x`. I want to report the sample mean of `x` in each dataset. Of course, I could always do this without a loop:

```
clear
use data_1.dta
quietly summarize x
display `r(mean)'

clear
use data_2.dta
quietly summarize x
display `r(mean)'

clear
use data_3.dta
quietly summarize x
display `r(mean)'

// and so on
```

But that is 4 lines of code per dataset, for 10 datasets. That's 40 lines of very repetitive code! Instead, I could do the same thing as follows:

```
forvalues dd = 1/10 {
        clear
        use data_`dd'.dta
        quietly summarize x
        display `r(mean)'
        }
```

which doesn't get any bigger if I have 100 datasets or 1,000,000 datasets.

How do you think I created these `.dta` files for this example? Hint: I'm lazy:

```
forvalues dd = 1/10 {
        clear
        set obs 1000
        gen x = rnormal()
        save data_`dd'.dta, replace
        drop x
        }
```

This chapter is a brief introduction to `for` loops without much application in econometrics. Once you understand this, go ahead and read Chapter 15, which demonstrates how this can be useful when you want to run many, similar regressions.

Exercises

Exercise 6.1.
Write one `for` loop that does all of the following

1. Computes the sum of the first 1,000 integers. That is, the final line of your script should display $\sum_{x=1}^{k} x$

2. Displays the first 1,000 Fibonacci numbers. These numbers follow the sequence $x_n = x_{n-1} + x_{n-2}$, and start with $x_1 = x_2 = 1$.

3. Puts 1,000 numbers from the Linear Congruential Generator into a column of data (this is a rather outdated method of getting a sequence of uniform pseudo-random numbers). This generator is defined by the sequence:

   $$x_n = \text{mod}(ax_{n-1} + c, m)$$

   Use the parameterization $m = 2^{32}$, $a = 1664525$, $c = 1013904223$. If you divide these by m you should get something that looks like a standard uniform. Show these numbers in a histogram.

Exercise 6.2.
Using the Galton Heights dataset, write a `for` loop that generates dummy variables:

$$\texttt{ParentHeightDifferece\_k}_i = \begin{cases} 1 & \text{if abs}(\texttt{mother\_height}_i - \texttt{father\_height}_i) \geq k \text{ inches} \\ 0 & \text{otherwise} \end{cases}$$

for $k = 1, 2, \ldots 10$. Write a few lines of code after this to check that your loop worked correctly.

Chapter 7

Types of data

7.1 How your computer thinks (or doesn't think) about data

7.2 Censored and truncated data

7.3 Categorical data

7.3.1 Unordered

7.3.2 Ordered

Chapter 8

Merging data, and wide & long formats

This chapter demonstrates some useful tools for merging data and handling panel data.

8.1 One-to-one merges

Frequently you have more than one data file that contains information you would like to study.

To begin with, we will look at two datasets:

- `USCensusTab04.xls` contains state abbreviations (i.e. AL, AK, etc), and state populations between 2000 and 1990
- `us_state.xls` contains state abbreviations and their full names

We aim to merge these two datasets so that we can have full state names and populations in the same dataset. This is a 1:1 merge because each row in the first file corresponds to exactly one row in the second file, and vice versa.

The following code imports the first dataset, appropriately labels some variables, and saves it in Stata's `.dta` format.

```
import excel "USCensusTab04.xls", sheet("Table 4") cellrange(A7:C57)
        rename A StateAbbrev
        rename B pop2000
        rename C pop1990

        // Have a look at the Tab04 dataset
        list in 1/10

save tab04.dta, replace

clear
```

The output from `list in 1/10` is

```
     +--------------------------------+
     | StateA~v    pop2000    pop1990 |
     |--------------------------------|
  1. |       AL    4447100    4040587 |
  2. |       AK    626,932    550,043 |
  3. |       AZ    5130632    3665228 |
  4. |       AR    2673400    2350725 |
  5. |       CA   33871648   29760021 |
     |--------------------------------|
  6. |       CO    4301261    3294394 |
  7. |       CT    3405565    3287116 |
  8. |       DE    783,600    666,168 |
  9. |       DC    572,059    606,900 |
 10. |       FL   15982378   12937926 |
     +--------------------------------+
```

that is, each row of data contains a state abbreviation (string variable), and that state's population in 2000 and 1990. We wish to import the actual name of the state into the dataset as well. We could, if we really wanted to waste our time, go ahead and manually code up:

```
generate State = .
replace State = "Alabama" if StateAbbrev == State = "AL"
// and so on ...
```

but we have better things to do. Fortunately, we also have the file `us_states.csv`, which has this information. First we need to get it in to a useful format:

```
import delimited "us_states.csv"
       rename v2 StateName
       rename v3 StateAbbrev
       drop v1
       // Have a look at the us_states dataset
       list in 1/10
```

which gives us the output:

```
     +------------------------+
     |   StateName   StateA~v |
     |------------------------|
  1. |     Alabama         AL |
  2. |      Alaska         AK |
  3. |     Arizona         AZ |
  4. |    Arkansas         AR |
  5. |  California         CA |
     |------------------------|
  6. |    Colorado         CO |
  7. | Connecticut         CT |
```

```
  8. |    Delaware         DE |
  9. |     Florida         FL |
 10. |     Georgia         GA |
     +------------------------+
```

So now we have a variable called `StateAbbrev` in each dataset, and we would like to have the column `StateName` alongside everything else in `tab04.dta`. To do this, all we need to do is:

```
merge 1:1 StateAbbrev using tab04.dta
```

This line gives us the output:

```
    Result                           # of obs.
    -----------------------------------------
    not matched                             1
        from master                         0  (_merge==1)
        from using                          1  (_merge==2)

    matched                                50  (_merge==3)
    -----------------------------------------
```

which tells us that we successfully merged 50 rows of each dataset, and there was one left over from the "using" dataset, `tab04.dta`. We can find out about this row by:

```
list if _merge~=3
```

```
     +-------------------------------------------------------------+
     | StateN~e   StateA~v   pop2000   pop1990           _merge |
     |-------------------------------------------------------------|
 51. |                  DC   572,059   606,900   using only (2) |
     +-------------------------------------------------------------+
```

It looks like DC did not appear in this file, but everything else worked well:

```
// Have a look at the merged dataset
list in 1/10
```

```
     +-------------------------------------------------------------+
     |   StateName   StateA~v    pop2000    pop1990        _merge |
     |-------------------------------------------------------------|
  1. |      Alaska         AK    626,932    550,043   matched (3) |
  2. |     Alabama         AL    4447100    4040587   matched (3) |
  3. |    Arkansas         AR    2673400    2350725   matched (3) |
  4. |     Arizona         AZ    5130632    3665228   matched (3) |
  5. |  California         CA   33871648   29760021   matched (3) |
```

```
     |-------------------------------------------------------------|
  6. |    Colorado         CO    4301261    3294394   matched (3) |
  7. | Connecticut         CT    3405565    3287116   matched (3) |
  8. |    Delaware         DE    783,600    666,168   matched (3) |
  9. |     Florida         FL   15982378   12937926   matched (3) |
 10. |     Georgia         GA    8186453    6478216   matched (3) |
     +-------------------------------------------------------------
```

Good!

We can fix the one row that `merge` could not help us by:

```
replace StateName = "District of Columbia" if StateAbbrev == "DC"
```

Annoying, but not as much as coding up 51 lines like this one.

If we are going to do other merges, we may want to `drop _merge` now, because it will try to generate another `_merge` variable for the next merge.

8.2 Wide and long datasets

Here we use `emp-unemployment.xls`, which can be found here: `http://www.icip.iastate.edu/tables/employment/unemployment-states`. I have modified this file slightly to make it easier to import into Stata. Specifically, I have appended the year columns with a "Y" so that the variable is preserved. Otherwise Stata just names these A, B, C, D, etc., because we aren't allowed to have variable names that start with a number.

Once we import the data, we notice a problem:

```
clear
import excel "emp-unemployment.xls", sheet("States") cellrange(B7:AL59) firstrow
desc
```

```
------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
Area            str20   %20s                  Area
Y1980           double  %10.0g                Y1980
Y1981           double  %10.0g                Y1981
Y1982           double  %10.0g                Y1982
Y1983           double  %10.0g                Y1983
```

The trouble is that we have one column for each year's unemployment rate (the rows correspond to states). This is a good way to organize things in a table, but it is terrible for organizing things in Stata. This file is in *wide format*: in terms of our panel data notation, each row corresponds to a different i subscript, and each column corresponds to a different t subscript. (e.g. one row will be for Ohio, and there will be one column for every year of data

we have). We would like to transform the data into *long format*, where each row corresponds to a different i-t pair, (e.g. one row will be Ohio in 2007). To see this:

```
list Area Y1980 Y1981 Y1982 Y1983 in 1/3
```

```
     +-----------------------------------------------+
     |          Area   Y1980   Y1981   Y1982   Y1983 |
     |-----------------------------------------------|
  1. | United States     7.1     7.6     9.7     9.6 |
  2. |       Alabama     8.9    10.6    14.1    13.8 |
  3. |        Alaska     9.6     9.4     9.9     9.9 |
     +-----------------------------------------------+
```

We fix this using the `reshape` command. The syntax on the next line is as follows:

- `reshape long` tells Stata that we want to convert a wide dataset to long,
- `Y` tells Stata that all of the columns that correspond to a different t index start with the string "Y"
- `i(Area)` tells Stata that `Area` is the i-index of the data
- `j(year)` tells stata that the (new) t-index is to be called "year", it will be equal to everything that follows the "Y" in the wide dataset. (e.g. the Y2006 column will be coded as year = 2006)

```
reshape long Y, i(Area) j(year)
// Problem solved!
 list in 1/5
```

```
     +-----------------------+
     |    Area   year      Y |
     |-----------------------|
  1. | Alabama   1980    8.9 |
  2. | Alabama   1981   10.6 |
  3. | Alabama   1982   14.1 |
  4. | Alabama   1983   13.8 |
  5. | Alabama   1984     11 |
     +-----------------------+
```

`Y` is a terrible name for unemployment, so as a good practitioner we may want to think about doing something like:

```
 rename Y unemp
```

Finally, `Area` is a string variable, which is not always easy to use in Stata. To fix this, we can use the `encode` command to assign an integer to every unique string in a variable

```
encode Area, generate(state)
```

A nice feature of this is that Stata remembers what these strings were, so you don't have to use data labels to get the state names back, even if you drop the original variable. This means that the actual state name, rather than `"state=3"` will chow up in all of your graphs, regression outputs, etc. For example:

```
quietly regress unemp i.state
coefplot, drop(_cons)
```

produces Figure 8.1, which shows the coefficients on the state dummy variables.

Of course, `encode` also allows you to do some naughty things that would get you stupid results, such as `regress state unemp`, `summarize state`, `hist state`, `kdensity state`, and so on. Stata will not give you an error here, so be careful. Go and do a Google search of "log(NAICS)" if you want a good example of what not to do with encoded variables.

we will use this dataset later, so before moving on, let's:

```
save unemp_long.dta, replace
```

And if, for whatever reason, you ever want to go back to the wide format, you can do this:

```
reshape wide unemp, i(state) j(year)
```

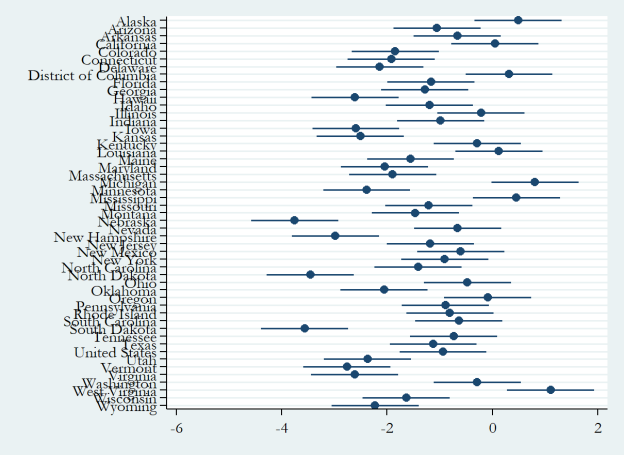


Figure 8.1:

8.3 Many-to-one and one-to-many merges

Earlier, we had two datasets, each with the same number of rows. Our expectation was that there was a on-to-one mapping between rows of these two datasets. In many cases, however, we have to implement a many-to-one or one-to-many match. This means that rows in one dataset correspond to many rows of the other. To illustrate this point, we continue with the (long format of) the dataset in Part 2, where we have yearly (t) data on US states (i). For whatever reason, we wish to include some characteristics that are in "list-state-capitals-us-764j.xlsx", which can be found here: `http://www.downloadexcelfiles.com/us_en/download-excel-file-list-state-capitals-united-states#.WHOAMBsrKUk`.

Since it is easier to do the merge for a dataset that is already in Stata's `.dta` format, let's load this in first:

```
clear
```

```
import excel "list-state-capitals-us-764j.xlsx", sheet("List of State Capitals of US")
    ↪ cellrange(A2:K52) firstrow
generate Area = State
```

The dataset we want to merge this with recode "State" as "Area". We need these to have the same name. We also need to save the dataset in Stata's `.dta` format:

```
// Save the data for merging
save list-state-capitals-us-764j.dta, replace
// Go back to the long version of the unemployment data from Part 2
use unemp_long.dta
```

We have a variable called `Area` in both datasets. We need to tell Stata which file is the "many" set, and which is the "one". For us, the data currently in memory is the "many", because we are matching on State (named Area), and there are multiple years for each state in this file. The `list-state-capitals-us-764j.dta` file contains exactly one row per state, so this is the "one".

```
merge m:1 Area using "list-state-capitals-us-764j.dta"
```

```
    Result                           # of obs.
    -----------------------------------------
    not matched                             72
        from master                         72  (_merge==1)
        from using                           0  (_merge==2)

    matched                             1,800  (_merge==3)
    -----------------------------------------
```

There are some that did not match:

```
tab Area if _merge~=3
```

```
                Area |      Freq.     Percent        Cum.
---------------------+-----------------------------------
District of Columbia |         36       50.00       50.00
       United States |         36       50.00      100.00
---------------------+-----------------------------------
               Total |         72      100.00
```

But that be expected, because DC, and the whole of the USA, does not appear in our dataset. Let's check that it worked:

```
list State unemp Capital in 1/10
sort year
list State unemp Capital in 1/10
```

which generates the output:

```
     +------------------------------+
     |   State   unemp      Capital |
     |------------------------------|
  1. | Alabama     8.9   Montgomery |
  2. | Alabama    10.6   Montgomery |
  3. | Alabama    14.1   Montgomery |
  4. | Alabama    13.8   Montgomery |
  5. | Alabama      11   Montgomery |
     |------------------------------|
  6. | Alabama     9.2   Montgomery |
  7. | Alabama     9.7   Montgomery |
  8. | Alabama     8.1   Montgomery |
  9. | Alabama     7.2   Montgomery |
 10. | Alabama       7   Montgomery |
     +------------------------------+

. sort year

. list State unemp Capital in 1/10

     +----------------------------------------+
     |          State   unemp          Capital |
     |----------------------------------------|
  1. |    Connecticut     5.8         Hartford |
  2. |   Rhode Island     7.2       Providence |
  3. |    Mississippi     7.4          Jackson |
  4. |       Maryland     6.6        Annapolis |
  5. |       Delaware     7.6            Dover |
     |----------------------------------------|
  6. |       Michigan    12.3          Lansing |
  7. |        Arizona     6.6          Phoenix |
  8. |           Utah     6.2   Salt Lake City |
  9. |        Georgia     6.3          Atlanta |
 10. | South Carolina     6.7         Columbia |
     +----------------------------------------+
```

Looks good!

Note that we can also do this as a one-to-many merge if we happened to start with `list-state-capitals-us-764j.dta` in the memory:

```
clear all
use list-state-capitals-us-764j.dta
merge 1:m Area using unemp_long.dta
```

Exercises

Exercise 8.1.
This exercise was (part of) the 4820 computational exam in 2017.

Files `auto1.csv` and `auto2.csv` contain data on cars. `auto1.csv` contains car characteristics, and `auto2.csv` contains prices.

1. Merge the two files so that we have price and characteristics in the same file
2. Briefly comment on any observations that don't match up, then drop them.
3. To check that you've merged things in correctly, produce a scatter plot of price (vertical axis) against mpg (horizontal axis). Use different colored dots for foreign and domestic cars.

Chapter 9

Non-linear models

Ai and Norton (2003)

Bland and Cook (2017)

Part III

Some common econometric techniques

Chapter 10

Ordinary Least Squares (linear regression)

This chapter is a companion to Bailey (2016) chapters 3-7. As such, it assumes that you have read them and understand them. That said, there will be some overlap. This is deliberate.

10.1 Some properties of bivariate OLS

In Bailey (2016) Chapter 3, we are introduced to bivariate OLS, or bivariate linear regression. The "bivariate" part of this means that we have two variables. These are usually notated as:

- Y_i is the *dependent*, or left-hand-side (LHS) variable, and
- X_i is the *independent*, *explanatory*, or right-hand-side (RHS) variable. I prefer not to use "independent", for reasons that will become clear when we study endogeneity.

To see how these names fit in, note that the equation that we are trying to estimate is:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{10.1}$$

where β_0 and β_1 are parameters that we are trying to estimate, and ϵ_i is an error term. Thus X_i is on the RHS, Y_i is on the LHS, X_i *explains* Y_i, and Y_i depends on X_i.

Without telling *Stata* anything else, when you ask it to `regress`, it will estimate a model that is valid if the following criteria are met:

1. $E[\epsilon_i]$=0. That is, the error term has zero mean. This is more of a normalization than a criterion, in that we can always dink with β_0 to make this true.
2. $E[X_i \epsilon_i] = 0$. In words: there is no (linear) correlation between the explanatory variable X_i and the error term ϵ_i. This could be why X_i is often referred to as the "independent variable": because it is assumed to be independent of ϵ_i. We will spend most of the remainder of this course worrying that X_i is *not* independent of ϵ_i (in specific cases), and how/if we can fix this problem.

3. $V[\epsilon_i] = \sigma^2$ for all i. In words: the variance of ϵ_i is a constant. Specifically, we might worry that the variance of ϵ_i depends on X_i. This assumption is called *homoskedasticity*.

4. $\text{cov}(\epsilon_i, \epsilon_j) = 0$ for all $i \neq j$. If this is not true, then one error will tell you something about another, and so the rows of our dataset are not independent observations.

When we `regress Y X`, we probably want to answer a question that looks like one of the following:

1. How can I predict Y_i using X_i? or

2. What is the causal effect of X_i or Y_i?

If our goal is prediction, then all we need is Assumption 1. This isn't really an assumption, so we are good to go. Specifically, if we are trying to predict Y_i without an X_i, we would just use $\bar{y}$, the sample mean of $\{y_i\}_{i=1}^N$. If we used X_i as well, then this must be at least as good as just using $\bar{y}$. On the other hand, if we want to get an unbiased estimate of the causal effect of X_i on Y_i, that is $E[\hat{\beta}_1] = \beta_1$, then we need Assumption 2 to be true.

What about the others? Well, if all we wanted was a point prediction, nobody would care. However it is `<understatement>` somewhat standard `</understatement>` to report measures of precision of your estimates, or do inference. In that case, you also need Assumptions 3 and 4.

10.1.1 Derivation of the bivariate OLS slope estimator

There are a few ways to motivate the OLS estimator. For this chapter, I will focus on minimizing the sum of squared residuals. Graphically, we seek to minimize the squared distance between the predicted value of our model, and the y-coordinate. We therefore seek the solution to:

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{\beta_0, \beta_1} \left[\sum_{i=1}^{N} (Y_i - \beta_0 - \beta_1 X_i)^2 \right] \tag{10.2}$$

In words: our estimators are the inputs to the function in the square brackets (i.e. the *arguments*) that minimize this function. Note that if we were to plot this thing as either a function of β_0 or β_1, it would look $f(x) = ax^2 + bx + c$, where $\beta_0, \beta_1 = 0$, and a is a positive constant. Hence, we are solving for the minimum of a very fancy parabola. Furthermore, this parabola is U-shaped (i.e. globally convex), so we can find the minimizers by solving the first-order conditions of the problem. Essentially, we will solve for "slope of parabola = 0" rather than "find the minimizers." Because we know the problem is convex, we know

that these two things are the same. The first-order conditions (FOC) are:

$$0 = \frac{\partial}{\partial \beta_0}\left[\sum_{i=1}^{N}(Y_i\beta_0 - \beta_1 X_i)^2\right] = -2\sum_{i=1}^{N}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i) \tag{10.3}$$

$$0 = \frac{\partial}{\partial \beta_1}\left[\sum_{i=1}^{N}(Y_i - \beta_0 - \beta_1 X_i)^2\right] = -2\sum_{i=1}^{N}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)X_i \tag{10.4}$$

We can re-arrange the FOC for β_0 as:

$$\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{\beta}_1 X_i) = \bar{y} - \hat{\beta}_1 \bar{x} \tag{10.5}$$

substituting this into the FOC for $\hat{\beta}_1$:

$$0 = -2\sum_{i=1}^{N}(Y_i - \bar{y} + \hat{\beta}_1\bar{x} - \hat{\beta}_1 X_i)X_i \tag{10.6}$$

$$0 = \sum_{i=1}^{N}\left[(Y_i - \bar{y})X_i - \hat{\beta}_1(X_i - \bar{x})X_i\right] \tag{10.7}$$

$$\hat{\beta}_1\sum_{i=1}^{N}(X_i - \bar{x})X_i = \sum_{i=1}^{N}(Y_i - \bar{y})X_i \tag{10.8}$$

$$\hat{\beta}_1\sum_{i=1}^{N}(X_i - \bar{x})(X_i - \bar{x}) = \sum_{i=1}^{N}(Y_i - \bar{y})(X_i - \bar{x}) \tag{10.9}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N}(Y_i - \bar{y})(X_i - \bar{x})}{\sum_{i=1}^{N}(X_i - \bar{x})(X_i - \bar{x})} \tag{10.10}$$

$$= \frac{\sum_{i=1}^{N}(Y_i - \bar{y})(X_i - \bar{x})}{\sum_{i=1}^{N}(X_i - \bar{x})^2} \tag{10.11}$$

where the step in line (10.9) follows by noting that $\sum_{i=1}^{N}(Z_i - \bar{z})c = 0$ for any data $\{Z_i\}_{i=1}^{N}$ and constant c. For example, data $\{X_i\}_{i=1}^{N}$ and constant $c = \bar{x}$. Here the "constant" is random, because it is a sample mean, but it is constant over all of the terms in the sum.

By multiplying the numerator and denominator by $1/N$, we can see that the estimator is a function of the sample variance of X, and the sample covariance of X and Y:

$$\hat{\beta}_1 = \frac{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \bar{y})(X_i - \bar{x})}{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{x})^2} = \frac{\widehat{\text{cov}}(X_i, Y_i)}{\hat{V}(X)} \tag{10.12}$$

This is *not* the sample correlation between X and Y.[1]

[1]To get the sample correlation, replace the denominator of this expression with $\sqrt{\hat{V}(X)\hat{V}(Y)}$

10.1.2 Unbiasedness

Ideally, we would like $\hat{\beta}_1$ to be an unbiased estimator of β_1. When is this the case? To begin with, it is useful to express $\hat{\beta}_1$ in terms of β_1 and the errors. To do this, we substitute in $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ to the numerator:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^N (Y_i - \bar{y})(X_i - \bar{x})}{\sum_{i=1}^N (X_i - \bar{x})^2} \tag{10.13}$$

$$= \frac{\sum_{i=1}^N (\beta_0 + \beta_1 X_i + \epsilon_i - \beta_0 - \beta_1 \bar{x} - \bar{\epsilon})(X_i - \bar{x})}{\sum_{i=1}^N (X_i - \bar{x})^2} \tag{10.14}$$

$$= \frac{\sum_{i=1}^N (\beta_1 (X_i - \bar{x}) + (\epsilon_i - \bar{\epsilon}))(X_i - \bar{x})}{\sum_{i=1}^N (X_i - \bar{x})^2} \tag{10.15}$$

$$= \beta_1 \frac{\sum_{i=1}^N (X_i - \bar{x})^2}{\sum_{i=1}^N (X_i - \bar{x})^2} + \frac{\sum_{i=1}^N (X_i - \bar{x})(\epsilon_i - \bar{\epsilon})}{\sum_{i=1}^N (X_i - \bar{x})^2} \tag{10.16}$$

$$= \beta_1 + \frac{\sum_{i=1}^N (X_i - \bar{x})\epsilon_i}{\sum_{i=1}^N (X_i - \bar{x})^2} \tag{10.17}$$

which is useful because now we have the thing we are tying to estimate (i.e. β_1) in the expression.

OK. Now for bias. We hope to show that $\hat{\beta}_1$ gets the right value on average. But now that we have two random variables (i.e. Y and X), we nee to be a bit more specific about what we mean by "on average". Here, we will assume that the Xs are constant. To do this, we take expectations that are conditional on the Xs (i.e. we treat them as a constant):

$$E[\hat{\beta}_1 \mid X] = E\left[\beta_1 + \frac{\sum_{i=1}^N (X_i - \bar{x})\epsilon_i}{\sum_{i=1}^N (X_i - \bar{x})^2} \mid X\right] \tag{10.18}$$

$$= E[\beta_1 \mid X] + E\left[\frac{\sum_{i=1}^N (X_i - \bar{x})\epsilon_i}{\sum_{i=1}^N (X_i - \bar{x})^2} \mid X\right] \tag{10.19}$$

$$= \beta_1 + \frac{\sum_{i=1}^N (X_i - \bar{x}) E[\epsilon_i \mid X]}{\sum_{i=1}^N (X_i - \bar{x})^2} \tag{10.20}$$

which is about as far as we can go without any more assumptions. In particular, we can only say the second term, of this expression is zero if we know that $E[\epsilon_i \mid X] = 0$. Mathematically, this means that $\hat{\beta}_1$ is unbiased if and only if the errors are uncorrelated with the Xs. In econometrics, we say "X is exogenous." When X and ϵ are correlated, we say "X is endogenous", and $\hat{\beta}_1$ is biased. This is the scourge of causal inference, and we will spend almost all of our time worrying that X is endogenous, and if it is, how (or if) we can fix the problem.

If we know whether X and ϵ are positively or negatively correlated, then we may be able

to predict the direction of the bias by noting that:

$$E[\hat{\beta}_1] = \beta_1 + \frac{\frac{1}{N}\sum_{i=1}^{N} E[(X_i - \bar{x})\epsilon_i]}{\frac{1}{N}\sum_{i=1} E[(X_i - \bar{x})^2]} = \beta_1 + \frac{\text{cov}(X, \epsilon)}{\sigma_X^2} = \beta_1 + \text{corr}(X, \epsilon)\frac{\sigma_\epsilon^2}{\sigma_X^2} \tag{10.21}$$

Since the variance terms must be positive, this tells us that the direction of the bias has the same sign as the correlation between X and ϵ.

10.1.3 Variance (in a very special case: homoskedasticity)

So we've worked out an estimator for the slope coefficient β_1, and that this estimator is unbiased in a very special set of circumstances. We can now get point estimates of causal effects (again, if we're lucky). After this, we want to do inference. That is, we want to put a standard error around our point estimate, calculate a confidence interval, a p-value, or do a hypothesis tests. All of these require the variance of $\hat{\beta}_1$. To get here, we take the variance (conditional on X) of both sides of Equation 10.17:

$$V[\hat{\beta}_1 \mid X] = V\left[\beta_1 + \frac{\sum_{i=1}^{N}(X_i - \bar{x})\epsilon_i}{\sum_{i=1}^{N}(X_i - \bar{x})^2} \mid X\right] \tag{10.22}$$

$$= V\left[\frac{\sum_{i=1}^{N}(X_i - \bar{x})\epsilon_i}{\sum_{i=1}^{N}(X_i - \bar{x})^2} \mid X\right] \quad \text{(since } \beta_1 \text{ is a constant)} \tag{10.23}$$

$$= \frac{V\left[\sum_{i=1}^{N}(X_i - \bar{x})\epsilon_i \mid X\right]}{\left(\sum_{i=1}^{N}(X_i - \bar{x})^2\right)^2} \quad \text{(Since we are treating } X \text{ as a constant)} \tag{10.24}$$

Just focusing on the numerator, we can note that each element of the sum is in expectation zero (conditional on X). That is, since we have assumed $E[\epsilon_i \mid X] = 0$, we can do the following:

$$E\left[\sum_{i=1}^{N}(X_i - \bar{x})\epsilon_i \mid X\right] = \sum_{i=1}^{N} E[(X_i - \bar{x})\epsilon_i \mid X] \tag{10.25}$$

$$= \sum_{i=1}^{N}(X_i - \bar{x})E[\epsilon_i \mid X] \tag{10.26}$$

$$= \sum_{i=1}^{N}(X_i - \bar{x}) \times 0 \tag{10.27}$$

$$= 0 \tag{10.28}$$

So going back to the numerator of Equation 10.24:

$$V\left[\sum_{i=1}^{N}(X_i-\bar{x})\epsilon_i \mid X\right] = E\left(\left(\sum_{i=1}^{N}(X_i-\bar{x})\epsilon_i - E\left[\sum_{i=1}^{N}(X_i-\bar{x})\epsilon_i \mid X\right]\right)^2 \mid X\right) \tag{10.29}$$

$$= E\left(\left(\sum_{i=1}^{N}(X_i-\bar{x})\epsilon_i - 0\right)^2 \mid X\right) \tag{10.30}$$

$$= E\left(\left(\sum_{i=1}^{N}(X_i-\bar{x})\epsilon_i\right)^2 \mid X\right) \tag{10.31}$$

$$= E\left[\sum_{i=1}^{N}\sum_{j=1}^{N}(X_i-\bar{x})\epsilon_i(X_j-\bar{x})\epsilon_j \mid X\right] \tag{10.32}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N} E\left[(X_i-\bar{x})\epsilon_i(X_j-\bar{x})\epsilon_j \mid X\right] \tag{10.33}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N}(X_i-\bar{x})(X_j-\bar{x})E\left[\epsilon_i\epsilon_j \mid X\right] \tag{10.34}$$

where the last step follows because we are conditioning on X.

Up to this point, we have made no new assumptions. Specifically, all we have assumed is that $E[\epsilon_i \mid X] = 0$, which we needed for $\hat{\beta}_1$ to be unbiased anyway.[2] But going forward, we need to assume more structure on our errors in order to get something we can work with. In particular, we need to assume something about $E[\epsilon_i\epsilon_j \mid X]$, which is the variance of ϵ_i if $i = j$, and the covariance of ϵ_i and ϵ_j otherwise. How we calculate our standard errors is a topic that deserves a whole chapter, and this is exactly what you get in Chapter 11. But for now, let's go ahead with the simplest, and therefore most dangerous assumption:

$$E[\epsilon_i\epsilon_j \mid X] = \begin{cases} \sigma^2 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} = \sigma^2 I(i=j) \tag{10.35}$$

This is telling us that:

- The variance of ϵ_i is constant for every observation in our data.
- Every possible pair of observations in our data have errors that are uncorrelated.[3]

Both of these could be wrong, and we will work on coping strategies for that later. But for the moment, let us suppose that this is a reasonable assumption to make for our application,

[2] In fact, we could have even got away with assuming $E[\epsilon_i] = 0$, which is a weaker assumption because it does not say anything about dependence between ϵ and X.

[3] My use of *uncorrelated* is very deliberate here. Specifically, I could have used the word *independent*, but I didn't.

and work out how we should calculate our standard errors. If we substitute this assumption into Equation 10.34, we get:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(X_i-\bar{x})(X_j-\bar{x})E\left[\epsilon_i\epsilon_j \mid X\right] \tag{10.36}$$

$$=\sum_{i=1}^{N}\sum_{j=1}^{N}(X_i-\bar{x})(X_j-\bar{x})I(i=j)\sigma^2 \tag{10.37}$$

$$=\sum_{i=1}^{N}(X_i-\bar{x})^2\sigma^2 \tag{10.38}$$

$$=\sigma^2\sum_{i=1}^{N}(X_i-\bar{x})^2 \tag{10.39}$$

That is, the only nonzero components of this double sum are the bits where $i=j$.

Now we can substitute this back into our expression for $V[\hat{\beta}_1 \mid X]$:

$$V[\hat{\beta}_1 \mid X]=\frac{\sigma^2\sum_{i=1}^{N}(X_i-\bar{x})^2}{\left(\sum_{i=1}^{N}(X_i-\bar{x})^2\right)^2} \tag{10.40}$$

$$=\frac{\sigma^2}{\sum_{i=1}^{N}(X_i-\bar{x})^2} \tag{10.41}$$

which is great, except we don't know what σ^2 is (it is a population parameter). Fortunately, we can replace it with an estimator of it (which happens to be both consistent and unbiased):

$$\hat{\sigma}^2=\frac{1}{N-2}\sum_{i=1}^{N}(Y_i-\hat{\beta}_0-\hat{\beta}_1X_i)^2=\frac{1}{N-2}\sum_{i=1}^{N}\hat{\epsilon}_i^2 \tag{10.42}$$

Which yields our estimator for the variance of $\hat{\beta}_1$:

$$\hat{V}[\hat{\beta}_1]=\frac{\hat{\sigma}^2}{\sum_{i=1}^{N}(X_i-\bar{x})^2} \tag{10.43}$$

10.2 `regress`: Implementing OLS in *Stata*

OK, so now that we understand what we're doing, let's actually do it. For this example, I am going to use the `galton_heights.csv` dataset to estimate the equation:

$$\texttt{child\_height}_i=\beta_0+\beta_1\texttt{av\_parent\_height}_i+\epsilon_i \tag{10.44}$$

where $\texttt{av\_parent\_height}_i$ is the average of the heights of the child's parents (in units of inches, minus sixty inches). All I have to do to estimate this is:

```
      Source |       SS           df       MS      Number of obs   =       934
-------------+----------------------------------   F(1, 932)       =    108.20
       Model |  1243.54014         1  1243.54014   Prob > F        =    0.0000
    Residual |  10711.1131       932  11.4926106   R-squared       =    0.1040
-------------+----------------------------------   Adj R-squared   =    0.1031
       Total |  11954.6533       933  12.8131332   Root MSE        =    3.3901

------------------------------------------------------------------------------
child_height |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
av_parent_~t |   .6625512   .0636941    10.40   0.000     .5375509    .7875516
       _cons |   2.342302   .4375628     5.35   0.000     1.483579    3.201025
------------------------------------------------------------------------------
```

Table 10.1: Estimation output for Equation10.44.

```
clear all
import delimited "galton_heights.csv"
desc
generate av_parent_height = (father_height+mother_height)/2

regress child_height av_parent_height
```

which gives me the output in Table 10.1. There's a lot of information here, a lot of which most people will care about, but all of it could be useful to someone, depending on the application. Of most importance are our estimates for β_0 and β_1, the constant and slope term respectively. These are $\hat{\beta}_0 = 2.34$ and $\hat{\beta}_1 = 0.66$. Immediately to the right of these numbers are the standard errors, and everything to the right of that are functions of the estimates and their standard errors. Specifically, we get t-statistics for the test that each parameter is equal to zero, the p-value associated with that (2-sided) test, ans a 2-sided 95% confidence interval for that parameter. That is (for large enough N):[4]

$$t_k = \hat{\beta}_k / \text{se}(\hat{\beta}_k) \tag{10.45}$$

$$p_k = 2\Phi(|t_k|) \tag{10.46}$$

$$CI_k = \hat{\beta}_k \pm 1.96\text{se}(\hat{\beta}) \tag{10.47}$$

10.3 Variable labels and `esttab`: Producing outputs that people actually want to look at *Stata*

So up to this point you can implement a (bivariate) linear regression, and get an output like Table 10.1. That's great! You can estimate stuff, and hopefully draw conclusions about

[4] *Stata* sometimes uses a t-distribution to calculate these values rather than a normal. If the number of degrees of freedom in the t distribution is greater than 30, these two distributions practically indistinguishable.

your data and answer your research question. However that's at most half the battle. You now have to *present* this information to an audience in a way that makes it easy for them to take in. There are several problems with presenting results in the format of Table 10.1. These include, but are not limited to:

1. It takes up a lot of space. For journals and presentations, this is bad. Furthermore, we usually want to show the results of a few regressions at once, and we can't do this with a foll regression output like the one in Table 10.1.

2. There is a lot of information that you don't want your seminar audience worrying about For example, should we always include R^2? A large R^2 is neither necessary nor sufficient to do valid inference, so why show it? Especially if it is small and you suspect there are some audience members who don't really understand R^2 and want to waste your time with stupid questions.

3. There is redundant information. For example the t, p, and CI columns are all functions of the coefficients, standard errors, and sample size. Again, this is a waste of space.

4. Sometimes the variable names are not intuitive, and we would like to display something other than an almost meaningless string. This is especially a problem if spaces help. For example, suppose that we wanted to call `av_parent_height` "`average parent height`". *Stata* won't let us do this because when you go to `summarize average parent height`, it will want to summarize three variables called `average`, `parent`, and `height`.

Let's begin with the example inBailey (2016) chapter 3 of investigating how the presidential vote changes with income changes. To begin with, we might want to show our reader (or seminar audience) a visual representation of some of our data. Some might be interested in the variable `rdi4`, which is national income growth (Bailey, quite reasonably, calls this "percent change in income"). Maybe we want to plot a histogram of this. However if we were to:

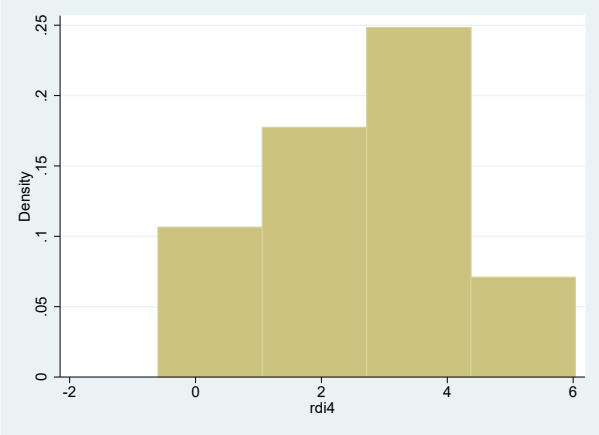
```
import excel "PresVote.xlsx", sheet("PresVote") firstrow // import the data
hist rdi4 // Produce a histogram
```

We get Figure 10.1a, which shows all of the information we need, but has a terrible horizontal axis label. What we really want to do is avoid those pesky people at the back of a seminar who are not paying attention from interrupting you to ask what variable is on the horizontal axis. What we really want is something like Figure 10.1b, which I got by typing:

```
import excel "PresVote.xlsx", sheet("PresVote") firstrow // import the data
label variable rdi4 "Income change" // Assign a variable label
hist rdi4 // Produce a histogram
```

What I did here was assign a "variable label". This does not change anything real about how *Stata* reads your code, other than telling it something like: "whenever you produce an output with the variable `rdi4`, use the variable's label, rather than its name." This label stays in the memory until you get rid of the variable, so I will get "Income change" instead

(a) Using variable names



(b) Using variable labels

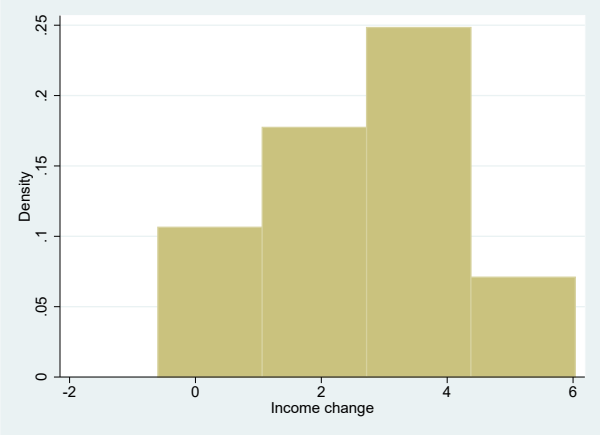


Figure 10.1: A poorly labeld figure, and a better one.

of "rdi4" in the output of everything generated below that understands variable labels as well. If, for whatever reason, we want to revert back to the stupid, unintuitive "rdi4", we can always use the `nolabel` option (after the comma).

What about regression tables? Table 10.1 is ugly and cumbersome, especially if we wanted to show more than one estimation in a small amount of space. For example, we may want also to estimate the relationship between child and parent height for just sons and just daughters, then compare the estimates side by side. Another example of this is Bailey treatment of the `PresVote` dataset in the Chapter 3 slides, which slices the data by (among other things) re-election years (e.g. Obama in 2012, GW Bush in 2000), and non re-election years (e.g. Obaba in 2008, GW Bush in 2004). The mechanics of getting these estimates is, at this point, easy:

```
regress vote rdi4
regress vote rdi4 if reelection==1
regress vote rdi4 if reelection==0
```

But this gives us three huge tables (like Table 10.1) that only render nicely in fixed-width font. What we would like to do is show all three in the same table, but only show a subset of the information, maybe just coefficients, standard errors, and something to do with p-values. Something like Bailey's Table 3.3. would be lovely (although we don't see p-values here), and fortunately *Stata*'s `esttab` command does just that. To begin, `esttab` is part of an add-on package called `estout`, so it doesn't come pre-installed. Fortunately, you can install it on your machine with the `<sarcasm>` unnecessarily cumbersome `</sarcasm>` command:

```
ssc install estout
```

You will only ever have to do this once.

Here's how it works. Firstly, you tell *Stata* to store your estimates, and give them a name. To do this, you can either do something like:

```
regress vote rdi4
estimates store reg_1
```

```
. esttab reg_*

------------------------------------------------------------
                      (1)             (2)             (3)
                     vote            vote            vote
------------------------------------------------------------
rdi4                2.291***        2.670**         1.078
                   (4.29)          (3.79)          (1.49)

_cons               45.94***        45.58***        46.94***
                   (27.15)         (18.85)         (24.91)
------------------------------------------------------------
N                       17              11               6
------------------------------------------------------------
t statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001
```

Table 10.2: `esttab output`

```
regress vote rdi4 if reelection ==1
estimates store reg_2
```

which are the first and second regressions we want to report, or we can put all of this in one line:

```
eststo reg_3: regress vote rdi4 if reelection ==0
```

if we type `esttab reg_*`, this gives us the output in Table 10.2: The `reg_*` part of this tells *Stata* to put all regressions that start with "`reg_`" into the table. Alternatively, we could have typed: `esttab reg_1 reg_2 reg_3`, but who has the time?

So this table has the right layout, but maybe we want it:

1. In a different format (maybe one that we could paste into a `MS Word` or LaTeXdocument).
2. To show standard errors, rather than p-values
3. To show the R^2
4. To use the value label for `rdi4` that we defined earlier.
5. Identify what we're doing in each column.

to achieve this, we modify the `esttab` line to the following:

```
esttab reg_* using votereg.rtf, se r2 label replace mtitles("All data" "Re-election" "Not re
    ↪ -election")
```

This gets us Table 10.3

	(1) All data	(2) Re-election	(3) Not re-election
Income change	2.291***	2.670**	1.078
	(0.534)	(0.704)	(0.724)
Constant	45.94***	45.58***	46.94***
	(1.692)	(2.418)	(1.884)
Observations	17	11	6
R^2	0.551	0.615	0.357

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 10.3: `esttab` table.

10.4 Interactions and the `margins` command

Once you get into multivariate OLS, you will probably want to include dummy variables and interactions. Once you've done this, you probably want to know the marginal effects associated with particular groups in your data. For this section, suppose that you are using the Galton heights dataset, and want to allow the relationship between parents' heights and child height to very by whether the child is a son or a daughter. One regression you might want to run would be:

```
regress  child_height i.son c.father_height##i.son c.mother_height##i.son
```

which generates the output in Table 10.4 Here I have made extensive use of Stata's # operator. Specifically, including (say) `c.father_height##i.son` tells *Stata* to include all possible interactions of `father_height` and the son dummy variable. Here `c.` tells *Stata* to treat `father_height` as a continuous variable, and the `i.` tells *Stata* to treat `son` as a categorical variable. Hence, the model we have estimated is:

$$\begin{aligned}\texttt{child\_height}_i = \beta_1 \texttt{son}_i + \beta_2 \texttt{father\_height}_i + \beta_3 \texttt{son}_i \times \texttt{father\_height}_i \\ + \beta_4 \texttt{mother\_height}_i + \beta_5 \texttt{son}_i \times \texttt{mother\_height}_i + \beta_0 + \epsilon_i\end{aligned} \tag{10.48}$$

From a quick eyeball of Table 10.4, we can see that: sons are taller than daughters ($\beta_1 > 0$), taller father and mother heights mean taller daughters ($\beta_2 > 0$, $\beta_4 > 0$), and that the slope for sons and daughters with respect for their parents' heights are approximately the same ($\beta_3 \approx 0$, $\beta_5 \approx 0$). However what is more difficult from this model is (i) accessing the predictions of child height conditional on whether the child is a son or daughter, and (ii) determining the marginal effect (i.e. causal effect) of parent height on the height of a son. Of course, while knowing the point estimates is useful, ideally we want tut standard errors around these things, too. This is where the `margins` command comes in handy. To generate the model's prediction for the average son and daughter in the sample, all we have to type is:

```
margins i.son
```

```
      Source |       SS           df       MS      Number of obs   =       934
-------------+----------------------------------   F(5, 928)       =    323.96
       Model |  7600.37366         5  1520.07473   Prob > F        =    0.0000
    Residual |  4354.27961       928  4.69211165   R-squared       =    0.6358
-------------+----------------------------------   Adj R-squared   =    0.6338
       Total |  11954.6533       933  12.8131332   Root MSE        =    2.1661

------------------------------------------------------------------------------
       child_height |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
--------------------+---------------------------------------------------------
             1.son |    4.70059   .5896612     7.97   0.000     3.543366    5.857814
     father_height |   .3708232   .0383549     9.67   0.000      .2955508    .4460955
                   |
son#c.father_height |
                 1 |   .0448174   .0574864     0.78   0.436     -.068001    .1576357
                   |
     mother_height |   .3029348   .0451339     6.71   0.000     .2143584    .3915112
                   |
son#c.mother_height |
                 1 |   .0250605   .0621703     0.40   0.687    -.0969502    .1470711
                   |
             _cons |  -.5894999   .4086313    -1.44   0.149    -1.391448    .2124486
------------------------------------------------------------------------------
```

Table 10.4: Estimation output from a regression using the Galton heights dataset, with a lot of interactions.

```
------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         son |
           0 |   4.061379   .1018357    39.88   0.000     3.861524    4.261234
           1 |   9.276831   .0988235    93.87   0.000     9.082888    9.470774
------------------------------------------------------------------------------
```

Table 10.5: `margins` output telling us our model's predictions for the heights of sons and daughters (rember the units here are inches minus 60in).

```
------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
father_height |
         son |
           0 |   .3708232   .0383549     9.67   0.000     .2955508    .4460955
           1 |   .4156406   .0428204     9.71   0.000     .3316046    .4996765
-------------+----------------------------------------------------------------
mother_height |
         son |
           0 |   .3029348   .0451339     6.71   0.000     .2143584    .3915112
           1 |   .3279953    .042756     7.67   0.000     .2440857    .4119048
------------------------------------------------------------------------------
```

Table 10.6: `margins` output telling us our model's predictions for the slopes for sons and daughters on father and mother height.

which produces the output in Table 10.5. This is the model's prediction of average son and daughter height, *conditional on all of the RHS variables* we have included in the regression. That is, for daughters, `margins` computed:

$$\frac{\sum_{i=\text{daughter}} \hat{\beta}_2 \texttt{father\_height}_i + \hat{\beta}_4 \texttt{mother\_height}_i + \hat{\beta}_0}{\text{number of daughters in sample}} \tag{10.49}$$

which is the sample mean of $\hat{y}_i$, if we restrict the sample to daughters only.

Now what about those pesky slope interactions? What if I wanted to make statements like "a son whose mother is 1in taller will on average be $x$in taller". x here is equal to $\hat{\beta}_4 + \hat{\beta}_5$, but I can't be bothered adding these up in my head (nor should you expect the person sitting in the back of your seminar to), and the standard error associated with this is a function of $\widehat{\text{cov}}(\hat{\beta}_4, \hat{\beta}_5)$ and the data, both of which we don't see in Table 10.4. Fortunately, `margins` can do this too:

```
margins i.son, dydx(father_height mother_height)
```

Which produces the output in Table 10.6 which produces the output in Table 10.6. This actually shows the four slopes we could be interested in. In order from top to bottom, they are: $\hat{\beta}_2$. $\hat{\beta}_2 + \hat{\beta}_3$, $\hat{\beta}_4$. and $\hat{\beta}_4 + \hat{\beta}_5$.

Exercises

Exercise 10.1.
Download the `galton_heights.csv` file and load it into Stata.

1. Using Stata's `esttab` function,[5] produce *one* table that can be read by your preferred

[5] If you have not already install it, execute the command `ssc install estout`.

typesetting/word processing package (e.g. LaTeX, *MS Word*, *Libre Office*, etc.) showing regressions that estimate:

   i. The effect of average parent height (i.e. `0.5×(father_height+mother_height)`) on child height
   ii. The effect of average parent height on daughters only
   iii. The effect of average parent height on sons only
   iv. The effect of mother height on daughter height

   Your table should show standard errors, not t-statistics, as is default with `esttab`. Also include the R^2 for each regression in the table, and a description of the restriction (e.g. something like "sons only", "daughters only", or "none".).

2. Interpret the slope coefficient from model (i) as if it is causal

3. Plot the squared residuals from model (i) against average parent height. Do you see evidence for heteroskedasticity? Explain. Estimate model (i) with heteroskedasticity-robust standard errors. What has changed? What has not changed?

4. Pick one model and suggest why there might be an omitted variable in it. If this is true, are we under- or over-estimating the causal effect?

Exercise 10.2.
Simulate two data-generating process, one with homoskedasticity, and one with heteroskedasticity. Specifically, contrast the distributions of the t-statistic when the null hypothesis $H_0: \beta_1 = 0$ is true, for the slope coefficient in these four cases:

1. Homoskedastic error, estimation assumes homoskedasticity

2. Homoskedastic error, estimation assumes heteroskedasticity

3. Heteroskedastic error, estimation assumes homoskedasticity

4. Heteroskedastic error, estimation assumes Heteroskedastic

Summarize your results as follows:

1. A table showing the rejection probabilities of the 5%, 2-sided test.

2. A plot of the densities of the four t-statistics, all on the same axis. For this, use the `kdensity` function. This is a kernel-smoothed density estimator, which I do not require you to understand. It is basically a fancy histogram.

Then comment on the implications of not assuming heteroskedasticity when it is present, and when it is present.

Exercise 10.3 (Simulation exercise – model fishing)**.**
Consider a dataset with a LHS variable Y, and four RHS variables X_1, and X_2. Suppose that you (were naughty and) wanted to model fish for the most statistically significant coefficient. Specifically, you execute the following commands in *Stata*:

```
regress Y X1
regress Y X2
regress Y X1 X2
```

and report the model with the smallest p-value on a slope coefficient.

Your task: Simulate the distribution of this p-value when the true data-generating process is:

$$Y_i \sim iidN(0,1) \tag{10.50}$$
$$X_{1,i},\ X_{2,i} \sim iidN(0,1) \tag{10.51}$$

with a sample size of $N = 100$. That is, there is no relationship between Y and any of the Xs, so the p-value for any of these regressions should be uniformly distributed (this uses an asymptotic assumption). The minimum of these four p-values will *not* be uniform. How frequently will we report results that are significant at the 5% level?

Exercise 10.4 (Simulation Exercise: Measurement error)**.**
Consider the following data0generating process discussed in Bailey (2016), section 5.3:

$$Y_i = \beta_0 + \beta_1 X^*_{1,i} + \epsilon_i \tag{10.52}$$
$$X_{1,i} = X^*_{1,i} + \nu_i \tag{10.53}$$

Write a simulation demonstrating that as $V[\nu_i]$ increases, $\hat{\beta}_1$ is biased toward zero.

Exercise 10.5 (Simulation exercise: regression discontinuity)**.**
Consider the following situation: At the beginning of a semester, students take a test. The test score is equal to their ability in the subject (unobservable to the econometrician), plus a random error. Those who score at or above 75 on the test are assigned to "class A", and everyone else to "class B". At the end of the semester, students take a second test. The score on this test is equal to their ability at the beginning of the semester, plus a positive amount if they were in "class A". We wish to estimate the effect of being in class A on this second test score.

Note that we can simulate this data generating process as follows (I played around with the numbers until it looked interesting:

$$\text{Ability}_i \sim N(70, \sqrt{30}) \tag{10.54}$$
$$\text{Score1}_i \mid \text{Ability}_i \sim N(\text{Ability}_i, \sqrt{5}) \tag{10.55}$$
$$\text{ClassA}_i = \begin{cases} 1 & \text{if Score1}_i \geq 75 \\ 0 & \text{otherwise} \end{cases} \tag{10.56}$$
$$\text{Score2}_i = \text{Ability}_i + 10\text{ClassA}_i \tag{10.57}$$

1. Write a .do file that simulates $N = 1,000$ draws from this distribution, then estimates the causal effect of being assigned to class A using regression discontinuity.

2. Suppose that students who score between 70 and 75 re-take the test, so that they get a second draw from 10.55. What assumption are we violating if this is happening and we estimated the causal effect using regression discontinuity?

3. Modify your .do file to simulate this process (but leave the original parts there so that you can compare them)

4. Generate a plot similar to Figure 11.10 in Bailey (2016) for both data-generating processes. Is this something you can always do? How can you tell that there is a problem with the second estimation?

5. Simulate the sampling distribution of the estimator for the causal effect of being in class A for both data generating processes. In addition to this, also simulate the distribution of the estimator for the coefficient on the dummy variable for class A in the simple bivariate regression for the following cases:

   (a) Using the entire dataset

   (b) Using only the data associated with test scores between 70 and 80.

   Why does the second model do better?

Chapter 11

Standard errors under different assumptions about ϵ

> *"He is in this, as in many other cases, led by an invisible hand to promote an end which was no part of his intention. But his standard errors, on the other hand, will always be wrong, for this is the nature of the applied economist." - Adam Smith*
>
> DeLuca (2019)

For all of Section I of Bailey (2016) (and for a lot of the following sections), we assume that the error terms of our regressions, $\{\epsilon_i\}_{i=1}^N$, (among other things):

1. Have a constant variance. That is, no matter two rows of the data we are looking at, it must be that:

$$V[\epsilon_i] = V[\epsilon_j] = \sigma^2, \quad \text{for all } i, j \in \{1, 2, \dots N\} \tag{11.1}$$

   This is the assumption of *homoskedasticity*.

2. Are uncorrelated with each other. That is, for any two rows i and j of our dataset:

$$\text{corr}(\epsilon_i, \epsilon_j) = 0, \quad \text{for all } i \neq j \tag{11.2}$$

Note that if either of these are not true, we needn't worry about *all* of the nice properties of OLS breaking down. Importantly, if these are the only problems we have, then our slope estimator is still unbiased. What we *should* worry about, however, is that our standard errors are not calculated correctly, and so without any correction for this, we report the results of hypothesis tests at our own peril. If the former assumption is violated, we refer to this as *heteroskedasticity*: the variance of the error term is not constant across observations. This is eminently fixable without having any additional insights into your data. On the other hand, if the latter is not true, then we need to know a bit more about our data to fix the problem. For a thorough run-through of these procedures, have a look at Cameron and Miller (2015).

What follows is a simplification of that work to the realm of bivariate OLS. The extension to multivariate OLS, and some non-OLS techniques, is relatively straightforward with the right matrix algebra background.

To begin with, let's see how far we can get with $V[\hat{\beta}_1]$ without making any additional assumptions about the error term. The variance of $\hat{\beta}_1$ when you `reg y x` is:

$$V[\hat{\beta}_1] = V\left[\frac{\sum_i (X_i - \bar{X})\epsilon_i}{\sum_i (X_i - \bar{X})^2}\right] \tag{11.3}$$

Noting that we are treating the Xs as fixed, without loss of generality, we can write this as:

$$V[\hat{\beta}_1] = \frac{V\left[\sum_i (X_i - \bar{X})\epsilon_i\right]}{\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.4}$$

The denominator of this is only a function of the data, so it is easily computable, and doesn't depend on any assumptions about ϵ. The numerator, however, simplifies differently depending on our understanding of ϵ. Before we make any further assumptions about ϵ, note that we can express the denominator of 11.4, without loss of generality, as follows:

$$V\left[\sum_i (X_i - \bar{X})\epsilon_i\right] = E\left[\left(\sum_i (X_i - \bar{X})\epsilon_i - E\left[\sum_j (X_j - \bar{X})\epsilon_j\right]\right)^2\right] \tag{11.5}$$

$$= E\left[\left(\sum_i (X_i - \bar{X})\epsilon_i\right)^2\right] \tag{11.6}$$

$$= E\left[\sum_i \sum_j \left((X_i - \bar{X})\epsilon_i\right)\left((X_j - \bar{X})\epsilon_j\right)\right] \tag{11.7}$$

$$= E\left[\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X})\epsilon_i\epsilon_j\right] \tag{11.8}$$

$$= \sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X})E[\epsilon_i\epsilon_j] \tag{11.9}$$

where (11.5) follows by the definition of variance, (11.6) follows because the expectation of any ϵ_i is zero, and (11.7) expands the squared term. What follows are further simplifications of (11.9), after making various assumptions about $E[\epsilon_i\epsilon_j]$.

11.1 Homoskedasticity: the *standard* standard errors

If you have been `regging y x` with free abandon up to this point, this is what you have been doing. Depending on how deep your understanding of OLS is, you would have been

implicitly, or (I really hope) explicitly, been making the assumption that the error term has constant variance, and that any two randomly selected errors are uncorrelated with each other. More formally, this means that:

Assumption 1 (Homoskedasticity)**.**

$$V[\epsilon_i] = E[\epsilon_i^2] = \sigma^2 \text{ for all } i = 1, 2, \dots N$$
$$E[\epsilon_i \epsilon_j] = 0 \text{ for all } i \neq j$$

Note that these two restrictions allow us to say something about *all* of the terms in (11.9). Specifically:

$$E[\epsilon_i \epsilon_j] = \begin{cases} \sigma^2 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{11.10}$$

This means that we can simplify (11.9) as follows:

$$\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X}) E[\epsilon_i \epsilon_j] = \sum_i (X_i - \bar{X})^2 E[\epsilon_i^2] \tag{11.11}$$
$$= \sum_i (X_i - \bar{X})^2 \sigma^2 \tag{11.12}$$
$$= \sigma^2 \sum_i (X_i - \bar{X})^2 \tag{11.13}$$

Substituting this into (11.4) yields:

$$V[\hat{\beta}_1] = \frac{V\left[\sum_i (X_i - \bar{X})\epsilon_i\right]}{\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.14}$$
$$= \frac{\sigma^2 \sum_i (X_i - \bar{X})^2}{\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.15}$$
$$= \frac{\sigma^2}{\sum_i (X_i - \bar{X})^2} \tag{11.16}$$

The denominator of this is a problem for your computer (i.e. it can always be calculated): it is N times the sample variance of X. σ^2, however, is an unknown. Fortunately we can consistently and unbiasedly estimate it using the residuals from the regression as follows:

$$\hat{\sigma}^2 = \frac{1}{N-k} \sum_i \hat{\epsilon}_i^2 \tag{11.17}$$

where k is the number of parameters in our model (for bivariate OLS, $k = 2$). And so, if we are happy with Assumption 1, we (or if we have something better than a pen and paper, our favorite statistical package) can compute our standard errors as follows:

$$\widehat{V[\hat{\beta}_1]} = \frac{\frac{1}{N-k} \sum_i \hat{\epsilon}_i^2}{\sum_i (X_i - \bar{X})^2} \tag{11.18}$$

At this point, we should make an important distinction between (11.16) (11.18). (11.16) is the *actual* variance of $\hat{\beta}_1$. However since we do not know the true value of σ^2, we must estimate this variance. Therefore (11.18) is an *estimator* of (11.16). (11.18) is what Bailey (2016) reports in his equations 3.9 and 3.10.[1]

Since we usually like to report things in the same units, we typically take the square root of this thing and report the standard error, rather than the variance:

$$\text{se}[\hat{\beta}_1] = \sqrt{\frac{\frac{1}{N-k}\sum_i \hat{\epsilon}_i^2}{\sum_i (X_i - \bar{X})^2}} \tag{11.19}$$

11.2 Heteroskedasticity: `reg y x, robust`

While Assumption 1 may seem like a reasonable restriction, there are plenty of cases where we assume homoskedasticity at our own peril. The next step is to relax the "constant variance" part of Assumption 1, while maintaining the assumption that the errors are independent. That is, we drop the "identically" from the iid assumption:

Assumption 2 (Heteroskedasticity)**.**

$$\begin{aligned} V[\epsilon_i] &= \sigma_i^2 \quad (\sigma_i^2 \text{ is not necessarily equal to } \sigma_j^2) \\ E[\epsilon_i \epsilon_j] &= 0 \text{ for all } i \neq j \end{aligned}$$

Going back to 11.9, the $E[\epsilon_i \epsilon_j]$ $(i \neq j)$ part of this, as in the previous section, means that we can set all of the $i \neq j$ components of the double summation equal to zero, leaving us just with the $i = j$ terms:

$$\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X}) E[\epsilon_i \epsilon_j] = \sum_i (X_i - \bar{X})^2 E[\epsilon_i^2] \tag{11.20}$$

However, unlike homoskedasticity, this is as far as we can get. Therefore we can simplify the expression for the variance to:

$$V[\hat{\beta}_1] = \frac{\sum_i (X_i - \bar{X})^2 E[\epsilon_i^2]}{\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.21}$$

A quick glance of (11.21) suggests that we need an estimate for $E[\epsilon_i^2]$ for every i. While $\hat{\epsilon}_i^2$ is a candidate for this, it is a terrible one because we only get one of those for each i, and so $\hat{\epsilon}_i^2$ does not plim to $E[\epsilon_i^2]$. Fortunately, *closer* inspection of (11.21) reveals that we need only estimate the numerator, specifically:

$$V[\hat{\beta}_1] = \frac{\sum_i (X_i - \bar{X})^2 E[\epsilon_i^2]}{\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.22}$$

$$= \frac{\frac{1}{N}\sum_i (X_i - \bar{X})^2 E[\epsilon_i^2]}{\frac{1}{N}\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.23}$$

[1] Put simply: $\hat{\beta}_1$ is the OLS estimator for β_1. (11.18) is the estimator of the variance of the OLS estimator for β_1. :p

and by some law of large numbers arguments:[2]

$$\frac{1}{N}\sum_i (X_i - \bar{X})^2 \hat{\epsilon}_i^2 \xrightarrow{p} \frac{1}{N}\sum_i (X_i - \bar{X})^2 E[\epsilon_i^2] \tag{11.24}$$

So we can estimate the variance of $\hat{\beta}_1$, under Assumption 2, as follows:

$$\widehat{V[\hat{\beta}_1]} = \frac{\sum_i (X_i - \bar{X})^2 \hat{\epsilon}_i^2}{\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.25}$$

$$\text{se}[\hat{\beta}_1] = \sqrt{\frac{\sum_i (X_i - \bar{X})^2 \hat{\epsilon}_i^2}{\left(\sum_i (X_i - \bar{X})^2\right)^2}} \tag{11.26}$$

Importantly, this formula requires *no additional information* about the data generating process to compute it (although it requires stronger assumptions than some of the techniques in later sections of this chapter). Contrast this to later sections of this chapter. If you can `reg y x`, you can *always* estimate standard errors that are robust to heteroskedasticity. In STATA, just `reg y x, robust` instead. These standard errors are often referred to as "heteroskedasticity-robust standard errors", or simply "robust standard errors". Try to use the former, they are not robust to everything (see, for example, the next section)!

11.3 Clustering: "I think you have 3 statistically independent observations"

In Section 11.1, we explored the implications of assuming that our errors were independently and identically distributed. In Section 11.2 we relaxed the "identically" distributed part by allowing each ϵ_i to have a different variance. In this Section, we will work to relax the "independently" part of this. In relation to (11.9), this means that we can now allow for $E[\epsilon_i \epsilon_j] \neq 0$ for some $i \neq j$.

The "some" in the previous sentence is an important one: in particular, I was very deliberate in not using the word "all". To understand this, and what is to come, it is important why we can't do this for "all" $i \neq j$. Note that the sample analog of (11.9) is:

$$\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X})\hat{\epsilon}_i \hat{\epsilon}_j \tag{11.27}$$

That is, we have replaced $E[\epsilon_i \epsilon_j]$ with $\hat{\epsilon}_i \hat{\epsilon}_j$. We can re-arrange this as follows:

$$\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X})\hat{\epsilon}_i \hat{\epsilon}_j = \sum_i \left[(X_i - \bar{X})\hat{\epsilon}_i\right] \sum_j \left[(X_j - \bar{X})\hat{\epsilon}_j\right] \tag{11.28}$$

Each one of these summation terms is the solution to the sum-of-squares minimization problem! In other words, when we do OLS, we are exactly setting these things equal to zero.

[2] I am being somewhat hand-wavy here.

Therefore, using (11.28) for the (sample equivalent of) the denominator of (11.4) means that we would compute standard errors of zero, and our t-statistics would shoot off to infinity. This is no good: we need to do better! Unlike heteroskedasticity, where we could say "we can construct standard errors that are robust to any kind of hereoskedasticity without knowing what that heterskedasticity looks like", we can't make a similar statement of the form "we can construct standard errors that are robust to any kind of correlation between the error terms, without knowing what that correlation looks like." But sometimes we *can* know a bit about the structure of this correlation, or at least have a good story about why the proposed structure is a believeable one.

One such instance of this is *clustering*. In this situation, we believ that the data are divided into distinct clusters. If two observations are not in the same cluster, then we have a good reason to believe that their errors ar uncorrelated. On the other hand, for two observations within the same cluster, then we cannot make the argument that they are uncorrelated.

An example Consider, for example, the task of estimating the mean height students on campus. The two following methods would achieve unbiased estimators of these quantities, both of which require the collection of 100 observations:

1. Randomly select N students on campus, and measure their heights $\{h_{i,1}\}_{i=1}^{N}$. Take the average of these heights. This is your estimate $\hat{\mu}^1 = \frac{1}{N}\sum_{i=1}^{N} h_{i,1}$.
2. Randomly select one student on campus. Measure his/her height on T days over the course of the academic year $\{h_{1,t}\}_{t=1}^{T}$. Take the average of these heights. This is your estimate $\hat{\mu}^2 = \frac{1}{T}\sum_{t=1}^{T} h_{1,t}$.

Suppose that each sample contains the same number of observations: $N = T = 100$. Both sampling procedures generate a point estimate using 100 observations. As (by assumption) any randomly selected student's height will on average be equal to the population mean, both procedures produce unbiased estimates. But what is generating the *variation* in measurements in these two procedures? Suppose that we can model a measurement of student i's height at time t as follows:

$$h_{i,t} = \mu + \eta_i + \epsilon_{i,t} \tag{11.29}$$

Where μ is the population mean height (the thing we are trying to estimate), η_i is student i's deviation from the mean height (i.e. how much taller/shorter is i than the average height), and $\epsilon_{i,t}$ is an iid error in measurement for student i on day i. We assume without loss of generality that $E[\eta_i] = E[\epsilon_i] = 0$. With some loss of generality, let's also assume that $V[\eta_i] < \infty$ and $V[\epsilon_{i,t}] < \infty$.

For sampling procedure 1, every row of our dataset belongs to a different student, so the variation in $h_{i,t}$ is driven by both η_i and $\epsilon_{i,t}$, so we could alternatively write this as $h_{i,t} = \mu + \psi_{i,t}$, where $\psi_{i,t}$ is the combined error term $\eta_i + \epsilon_{i,t}$. Hence $\hat{\mu}^1 \xrightarrow{p} \mu$, good! The more observations we collect in sampling procedure 1, the more likely we are to be arbitrarily close

to μ. Additionally, by standard central limit arguments: $\sqrt{N}(\hat{\mu}^1 - \mu) \xrightarrow{d} N(0, V[\eta_i + \epsilon_{i,t}])$, and so all of our inference can be done in the *usual* way.

For sampling procedure 2, things become more complicated. To see this, note that since we are repeatedly sampling the same student's height, we always get the same η_i in our equation. Therefore, instead of (loosely) converging to μ, we get a really good estimate of $\mu + \eta_1$, the single student's height. By "really good" here, I don't mean that we should be happy: we have a really good estimate of something we don't want to know, and hence a really *bad* estimate of the population mean height. While in sampling procedure 1, increasing the sample size gets us closer (in the plim sense) to μ, increasing the sample size in sampling procedure 2 gets us closer to $\mu + \eta_i$. This is *in expectation* equal to μ, but it does not have the same nice convergence properties (both $\xrightarrow{p}$ and $\xrightarrow{d}$) as $\hat{\mu}^1$. One way of looking at this problem is that sampling procedure 2 does not collect *statistically independent* observations:

$$\begin{aligned} \text{for } t \neq s: \quad \text{cov}(h_{1,t}, h_{1,s}) &= E\left[(\eta_1 + \epsilon_{1,t})(\eta_1 + \epsilon_{1,s})\right] && (11.30) \\ &= E\left[\eta_1^2 + \epsilon_{1,s}\eta_1 + \epsilon_{1,t}\eta_{1,t} + \eta_{i,t}\eta_{1,s}\right] && (11.31) \\ &= E\left[\eta_1^2\right] \neq 0 && (11.32) \end{aligned}$$

OK, so it seems reasonable, even before reading the above section, that any econometrician with half a brain should realize that procedure 2 is a terrible one for estimating μ. Why would we *ever* see such a procedure at all then? The answer is that we usually don't, but we often see things that are a mix of procedures 1 and 2. In this context, this might be because it is cheaper to sample one person N times than sample N people once (perhaps the study requires getting consent from all of the participants, but only once per participant). Clearly we would never want to just sample 1 person, but maybe we settle for sampling a few people a few times. Therefore, it is reasonably common to see a sampling procedure like the following:

3. Randomly select N students on campus, and measure their heights on T days over the course of the academic year $\{h_{i,t}\}_{i=1,t=1}^{i=N,t=T}$. Take the average of these heights. This is your estimate $\hat{\mu}^3 = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N} h_{i,t}$.

For the sake of simplicity, we have assumed that we have a *balanced panel*: each student is measured T times, hence we have NT observations. This assumption is unnecessary, and does not affect any of the discussion below.

Again, $\hat{\mu}^3$ is an unbiased estimator of μ because everything that goes in to the average is on average equal to μ. Moreover, as $N \to \infty$ (i.e. as we sample more and more students), this thing will plim to μ, and will be asymptotically normal. However, we need to be careful about how we apply this second property when doing inference. Specifically, it is reckless to think, or apply a technique that assumes, that we have NT statistically independent

observations. To see this, note the following for two arbitrary observations in our dataset:

$$\begin{aligned}
\text{cov}(h_{i,t}, h_{j,s}) &= E\left[(\eta_i + \epsilon_{i,t})(\eta_j + \epsilon_{j,s})\right] && (11.33)\\
&= E\left[\eta_i\eta_j + \eta_i\epsilon_{j,s} + \eta_j\epsilon_{i,t} + \epsilon_{i,t}\epsilon_{j,s}\right] && (11.34)\\
&= E\left[\eta_i\eta_j\right] + 0 + 0 + 0 && (11.35)\\
&= \begin{cases} E\left[\eta_i^2\right] > 0 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} && (11.36)
\end{aligned}$$

What this is telling us is that observations that correspond to the same student are not statistically independent, but observations that correspond to different students are statistically independent. Actually, 11.36 tells us more than this: observations corresponding to the same student are *correlated*, and we know that this correlation must be positive. The implications of this are as follows:

- $E[\hat{\mu}^3] = \mu$ (good)
- As $N \to \infty$, $\hat{\mu}^3 \to \mu$ (good)
- As $N \to \infty$, neither the standard, nor the heteroskedasticity-robust, standard errors approach the asymptotic standard deviation of $\hat{\mu}^3$.

The third point is really bad: we can get a good point estimate of μ quite easily, but unless you keep reading, you can't do any hypothesis tests. Please keep reading!

Formally, we have a variable c_i which identifies the cluster that observation i belongs to such that:

$$\begin{aligned}
c_i = c_j &\iff i \text{ and } j \text{ are in the same cluster} && (11.37)\\
c_i \neq c_j &\iff i \text{ and } j \text{ are not in the same cluster} && (11.38)
\end{aligned}$$

So in terms of our estimator $\hat{\mu}^3$, two rows of our dataset have the same c if and only if they correspond to the same student. In the Galton Heights dataset, we may be worried that errors within families are correlated. For example, if one child in a family is a glutton for protein, then their siblings may be protein-starved. If protein consumption positively affects height, then the errors would be negatively correlated within families.[3]

Now let's go back to Equation 11.9. Now our problem is that we have some is and js for which $E[\epsilon_i\epsilon_j] \neq 0$. Specifically, if observations i and j correspond to the same student, then $E[\epsilon_i\epsilon_j] = E[\eta_i^2] \neq 0$. Fortunately, we also have variable c in our dataset that tells us which observations belong to the same student. Our solution to this problem is remarkably similar to the heteroskedastisity problem: we suspect that some errors are correlated, so we don't assume that their correlation to zero. Specifically, note that we can (trivially) write Equation 11.9 as follows:

$$\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X})E[\epsilon_i\epsilon_j] = \sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X})\left(E[\epsilon_i\epsilon_j]I(E[\epsilon_i\epsilon_j] \neq 0)\right) \tag{11.39}$$

[3]This story is not particularly plausible to me, but if true, the errors would be negatively correlated.

The sample analog of this is:

$$\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X}) \left(\hat{\epsilon}_i \hat{\epsilon}_j I(E[\epsilon_i \epsilon_j] \neq 0)\right) \tag{11.40}$$

That is, we replace the thing we don't know, $E[\epsilon_i \epsilon_j]$, with something that we do know, $\hat{\epsilon}_i \hat{\epsilon}_j$. Note that we haven't replaced $I(E[\epsilon_i \epsilon_j] \neq 0)$ with anything. This is because we know what this is! We have made an argument that our data falls into groups, called clusters, such that if i and j are in the same cluster, their errors could be correlated, but they could not be correlated if they were not in the same cluster. Hence:

$$I(E[\epsilon_i \epsilon_j] \neq 0) = \begin{cases} 1 & \text{if } c_i = c_j \\ 0 & \text{otherwise} \end{cases} \tag{11.41}$$

Hence, we can calculate standard errors that respect this kind of dependence by substituting:

$$\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X}) \left(\hat{\epsilon}_i \hat{\epsilon}_j I(c_i = c_j)\right) \tag{11.42}$$

into the numerator of our equation for $V[\hat{\beta}]$. Hence:

$$V^{\text{clu}}[\hat{\beta}_1] = \frac{\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X}) \left(\hat{\epsilon}_i \hat{\epsilon}_j I(c_i = c_j)\right)}{\left(\sum_i (X_i - \bar{X})^2\right)^2} \tag{11.43}$$

$$\text{se}^{\text{clu}}[\hat{\beta}_1] = \sqrt{\frac{\sum_i \sum_j (X_i - \bar{X})(X_j - \bar{X}) \left(\hat{\epsilon}_i \hat{\epsilon}_j I(c_i = c_j)\right)}{\left(\sum_i (X_i - \bar{X})^2\right)^2}} \tag{11.44}$$

This is often referred to as "cluster-robust standard errors". Now compare this to (11.25), which is the estimator of the variance of $\hat{\beta}_1$ when we have heteroskedasticity (but not clustering). In particular, if there is only one observation per cluster (i.e. $c_i = i$, and hence all c_is are different), then $V^{\text{clu}}[\hat{\beta}_1]$ collapses to (11.25), because $I(c_i = c_j) = 1$ only when $i = j$. The implication of this is that the cluster-robust standard errors are also robust to heteroskedasticity.

In *Stata*, we can calculate these standard errors using the `vce` option in our estimation:

```
regress Y X, vce(cluster clusterid)
```

where `cluster` tells *Stata* that you want to calculate cluster-robust standard errors, and `clusterid` is the variable that identifies the clusters, i.e. c_i.

Further reading

1. Cameron and Miller (2015): Much more detail about cluster-robust inference. Extends this discussion to multivariate OLS (everything works the same way, just more matrix algebra).

2. Abadie et al. (2017): A working paper (i.e. as yet peer reviewed) discussing the motivation for clustering and some common misconceptions about it.

3. A blog post by Marc F. Bellemare (Metrics Monday) discussing the above working paper on clustering: `http://marcfbellemare.com/wordpress/12712#more-12712`

Exercises

Exercise 11.1.
Consider a constant-only model for Galton's data from Problem Set 5, where we just model the mean of child height:

$$\texttt{child\_height}_{i,f} = \beta_0 + \epsilon_{i,f} \tag{11.45}$$

where the "$_{i,f}$" subscript indicates child i in family f.

We will investigate the implications of error terms with the following property:

$$\epsilon_{i,f} = \eta_i + \frac{\rho}{F_i - 1} \sum_{j \in f, i \neq j} \eta_j \tag{11.46}$$

$$\eta_i \sim iidN(0, \sigma_\eta^2) \tag{11.47}$$

Where F_i is the number of children (including i) in i's family. The normal assumption for η_i is not necessary here, but we make it so it is clear what we are simulating.

Here, the notation under the sum indicates that we are summing over all other children in the same family as i. E.g. if the first 4 observations in out dataset were in the same family:

$$\epsilon_{1,1} = \eta_1 + \frac{\rho}{3}(\eta_2 + \eta_3 + \eta_4)$$
$$\epsilon_{2,1} = \eta_2 + \frac{\rho}{3}(\eta_1 + \eta_3 + \eta_4)$$

1. Calculate $E[\epsilon_{i,f}]$ and $V[\epsilon_{i,f}]$

2. Calculate $\text{cov}(\epsilon_{i,f}, \epsilon_{j,f})$, for $i \neq j$. That is, what is the correlation between child i and child j's error term if they have the same parents?

3. Interpret the role of parameter ρ in your expression for $\text{cov}(\epsilon_{i,f}, \epsilon_{j,f})$. What does it mean if $\rho = 0$? $\rho > 0$? $\rho < 0$?

4. Would there be anything wrong with using OLS if this is how ϵ behaves? Will it affect bias? consistency? standard errors? What is wrong with the usual assumptions we make to do OLS?

Exercise 11.2 (Simulation exercise)**.**
Refer to Exercise 11.1. Fix $\eta_i \sim iidN(0,1)$, assume that $\beta_0 = 0$ and investigate the role of ρ. Specifically, suppose that you have a sample of 1,000 children, each in a family of four. That is, children $i = 1, 2, 3, 4$ are in family 1, $i = 5, 6, 7, 8$ are in family 2, and so on. Simulate the test statistic of the following procedures for $\rho = 0,\ 0.5, -0.5$:

1. `regress child_height`, then test that $\beta_0 = 0$, reject H_0 if $|t| > 1.96$ (i.e. the usual way that you would test that $\beta_0 = 0$)

2. `regress child_height`, restricting your sample to only one child per family (i.e. use child 1, 5, 9, 13, ..., 997). Reject H_0 if $|t| > 1.96$.

3. `regress child_height`, restricting your sample to only the first 250 observations in your sample. Reject H_0 if $|t| > 1.96$.

Summarize your results in a table that shows how the rejection probabilities vary with these three procedures and the three values for ρ.

Given that you are simulating the distribution of the test statistic under the null, what should these rejection probabilities be equal to, and do they differ from this value? if so, how do they differ?

Exercise 11.3 (Simulation exercise)**.**
Modify your simulation from the previous question to show that appropriate clustering fixes the problem.

Chapter 12

Maximum Likelihood

12.1 How some estimators relate to maximum likelihood

12.1.1 Sample mean for a Bernoulli (coin flip) variable

In the early chapters of this material, we learned about why sample means were useful estimators. For Bernoulli random variables, we could estimate the probability of a success by taking the sample mean. Let's see how we can do it with maximum likelihood.

We start with the assumption that our data are distributed according to:

$$X_i \sim iid\text{Bernoulli}(\theta) \tag{12.1}$$

and wish to estimate θ. The probability mass function of one observation in our data is:

$$p_{X_i}(x) = \begin{cases} \theta & \text{if } x = 1 \\ 1-\theta & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{12.2}$$

$$= \theta^{I(x=1)}(1-\theta)^{I(x=0)} \tag{12.3}$$

Since we have assumed that the X_is are iid, we can multiply the probability of each observation together to get the probability mass function for all rows of our data.

$$p_{X_1,X_2,\ldots,X_N}(x_1, x_2, \ldots, x_N; \theta) = \prod_{i=1}^{N} p_{X_i}(x_i) \tag{12.4}$$

$$= \prod_{i=1}^{N} \theta^{I(x=1)}(1-\theta)^{I(x=0)} \tag{12.5}$$

which is also the likelihood function evaluated at θ. We take logs to get the log-likelihood

(because it is easier to maximize):

$$\log L(\theta) = \log \left[\prod_{i=1}^{N} \theta^{I(X_i=1)} (1-\theta)^{I(X_i=0)} \right] \tag{12.6}$$

$$= \sum_{i=1}^{N} \log \left[\theta^{I(X_i=1)} (1-\theta)^{I(X_i=0)} \right] \tag{12.7}$$

$$= \sum_{i=1}^{N} \left[I(X_i = 1) \log(\theta) + I(X_i = 0) \log(1-\theta) \right] \tag{12.8}$$

$$= N \left[\bar{X} \log(\theta) + (1 - \bar{X}) \log(1-\theta) \right] \tag{12.9}$$

We find the maximum likelihood estimator by taking the derivative and setting it equal to zero:

$$\hat{\theta} = \arg\max_{\theta} \log L(\theta) \tag{12.10}$$

$$\text{FOC:} \quad 0 = N \left[\frac{\bar{X}}{\hat{\theta}} - \frac{1-\bar{X}}{1-\hat{\theta}} \right] \tag{12.11}$$

$$\hat{\theta} = \bar{X} \tag{12.12}$$

That is, the maximum likelihood estimator of θ is also the sample mean!

Now suppose that we want to test that θ is equal to a specific value, say θ_0. Substituting $\hat{\theta}$ into the likelihood function yields:

$$L^U = N \left[\bar{X} \log(\bar{X}) + (1 - \bar{X}) \log(1 - \bar{X}) \right] \tag{12.13}$$

and our restricted likelihood is:

$$L^R = N \left[\bar{X} \log(\theta_0) + (1 - \bar{X}) \log(1 - \theta_0) \right] \tag{12.14}$$

So the likelihood ratio test statistic is:

$$LR = 2 \left[L^U - L^R \right] \tag{12.15}$$

$$= 2N \left[\bar{X} \log(\bar{X}) + (1 - \bar{X}) \log(1 - \bar{X}) - \bar{X} \log(\theta_0) - (1 - \bar{X}) \log(1 - \theta_0) \right] \tag{12.16}$$

$$= 2N \left[\bar{X} \log(\bar{X}/\theta_0) - (1 - \bar{X}) \log((1 - \bar{X})/(1 - \theta_0)) \right] \tag{12.17}$$

How do we know that this thing is distributed χ^2_1 for large N? Note that the likelihood ratio is a function of $\bar{X}$, the sample mean, and θ_0, the value of θ if H_0 is true. θ_0 is fixed for the hypothesis, so it is really only a function of $\bar{X}$. Let's make a 2nd-order Taylor series approximation of this function:

$$LR(\bar{X}) \approx LR(\theta_0) + (\bar{X} - \theta_0) \frac{\partial LR(x)}{\partial \bar{X}} \bigg|_{x=\theta_0} + \frac{1}{2} (\bar{X} - \theta_0)^2 \frac{\partial^2 LR(x)}{\partial \bar{X}^2} \bigg|_{x=\theta_0} \tag{12.18}$$

$$= 0 + (\bar{X} - \theta_0) 2N \left[\log\left(\frac{x}{\theta_0}\right) + \frac{x}{x} - \log\left(\frac{1-x}{1-\theta_0}\right) + \frac{1-x}{1-x} \right]_{x=\theta_0} \tag{12.19}$$

$$+ \frac{1}{2} (\bar{X} - \theta_0)^2 2N \left[\frac{1}{x} + \frac{1}{1-x} \right]_{x=\theta_0}$$

Noting that everything on the first line of Expression 12.19 is zero:

$$LR(\bar{X}) \approx \frac{1}{2}(\bar{X} - \theta_0)^2 \frac{2N}{\theta_0(1-\theta_0)} \tag{12.20}$$

$$= \left(\frac{\sqrt{N}(\bar{X} - \theta_0)}{\sqrt{\theta_0(1-\theta_0)}} \right)^2 \tag{12.21}$$

So when the null is true, the thing inside the parentheses is asymptotically standard normal. Since the LR is approximately this thing squared (the approximation gets better as $N \to \infty$), it follows that:

$$LR(\bar{X}) \xrightarrow{d} \chi^2_1 \tag{12.22}$$

12.1.2 Linear regression

Let's restrict our attention to the bivariate linear regression model. The data-generating process is often described as:

$$Y_i = \beta_1 + \beta_2 X_i + \epsilon_i \tag{12.23}$$

$$E[\epsilon_i \mid X] = 0 \tag{12.24}$$

$$V[\epsilon_i \mid X] = \sigma^2, \qquad \text{(homoskedasticty)} \tag{12.25}$$

$$E[\epsilon_i X_i] = 0, \qquad \text{(exogeneity)} \tag{12.26}$$

Note that we have already assumed a few things here (specifically, homoskedasticity). Now, we are going to make a *very* restrictive assumption:

$$\epsilon_i \mid X_i \sim iidN(0, \sigma^2) \tag{12.27}$$

We have seen lots of normals show up in our analysis, but this is not usually where they show up: usually we make an argument that a sample mean is approximately normal because N is large. Here, on the other hand, we have assumed that the *errors* are normal. This is therefore a much more restrictive model than the one we write down when we do OLS, but let's see where it gets us.

The parameters we wish to estimate are the intercept and slope coefficients, β_0 and β_1, as well as the variance parameter σ^2. First note that:

$$Y_i \mid X_i \sim N(\beta_0 + \beta_1 X_i, \sigma^2) \quad \text{(independent)} \tag{12.28}$$

Here I don't write "iid" because the distribution of $Y_i \mid X_i$ changes with X_i. Using the above result this information, we can construct the pdf of one observation;

$$f_{Y|X}(y; \beta_0, \beta_1, \sigma^2) = \phi(y; \beta_0 + \beta_1 X_i, \sigma^2) \tag{12.29}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \beta_0 - \beta_1 X_i)^2 \right) \tag{12.30}$$

Usually I would just leave this as the first line, with $\phi(\cdot; \mu, \sigma^2)$ representing the normal density function with mean μ and variance σ^2, however we need to use some properties of this to derive the estimator for $(\beta_0, \beta_1, \sigma^2)$. Thats the probability (density), or likelihood, of observing *one* row of the data. Now we assume that each rows are independent, so we can multiply these densities together to get the probability density function of th data when the parameters are known:

$$f_{Y_1,Y_2,\ldots,Y_N;X_1,X_2,\ldots,X_N}(y; \beta_0, \beta_1, \sigma^2) = \prod_{i=1}^{N} f_{Y_i|X_i}(Y_i; \beta_0, \beta_1, \sigma^2) \tag{12.31}$$

$$= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(Y_i - \beta_0 - \beta_1 X_i)^2\right) \tag{12.32}$$

This is the likelihood, which in principle you could go ahead and maximize, but it is much easier to maximize the log-likelihood:

$$\log L(\beta_0, \beta_1, \sigma^2) = \log\left(\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(Y_i - \beta_0 - \beta_1 X_i)^2\right)\right) \tag{12.33}$$

$$= \sum_{i=1}^{N} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(Y_i - \beta_0 - \beta_1 X_i)^2\right)\right) \tag{12.34}$$

$$= \sum_{i=1}^{N} \left[\log 0 - \frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(Y_i - \beta_0 - \beta_1 X_i)^2\right] \tag{12.35}$$

$$= -\underbrace{\frac{N}{2}\log 2\pi\sigma^2}_{A} - \underbrace{\frac{1}{2\sigma^2}\sum_{i=1}^{N}(Y_i - \beta_0 - \beta_1 X_i)^2}_{B} \tag{12.36}$$

Note that the only component of this expression that contains β_0 and β_1 is B. Therefore, we don't need to consider A if we just want to estimate β_0 and β_1. Furthermore, since $1/2\sigma^2 > 0$, we don't need to consider this constant either. So maximizing th log-likelihood with respect to the slope and intercept term is equivalent to the following optimization problems:

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\max_{\beta_0,\beta_1}\left[-\sum_{i=1}^{N}(Y_i - \beta_0 - \beta_1 X_i)^2\right] \tag{12.37}$$

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{\beta_0,\beta_1}\left[\sum_{i=1}^{N}(Y_i - \beta_0 - \beta_1 X_i)^2\right] \tag{12.38}$$

That is, $\arg\max_x g(x)$ returns the x that maximizes $g(x)$ (i.e. the *argument* of g that maximizes g), whereas $\max_x g(x)$ equals the maximum value of $g(x)$. Importantly here, the second optimization problem is one that we've seen before: $Y_i - \hat{\beta} + 0 - \hat{\beta}_1 X_i$ is the residual of observation i, and so the above minimization problem is exactly the same minimization

problem we used to derive the OLS estimator: we are minimizing the sum of squared residuals! Hence, without further derivations, we know that:

$$\hat{\beta}_1^{\text{ML}} = \hat{\beta}_1^{\text{OLS}} = \frac{\sum_{i=1}^N (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} \tag{12.39}$$

$$\hat{\beta}_0^{\text{ML}} = \hat{\beta}_0^{\text{OLS}} = \bar{Y} - \hat{\beta}_1^{\text{OLS}} \bar{X} \tag{12.40}$$

Letting SSR equal the (minimized) residual sum of squares, we can write the estimator for σ^2 as:

$$\hat{\sigma}^2 = \arg\max_{\sigma^2} \left[-\frac{N}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} SSR \right] \tag{12.41}$$

$$\text{FOC:} \quad 0 = -\frac{N}{2\hat{\sigma}^2} + \frac{SSR}{2(\hat{\sigma}^2)^2} \tag{12.42}$$

$$0 = -N\hat{\sigma}^2 + SSR \tag{12.43}$$

$$\hat{\sigma}^2 = \frac{SSR}{N} = \frac{1}{N} \sum_{i=1}^N (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2 \tag{12.44}$$

which is *almost* the equation we use for OLS (we usually divide by $N - k$ to eliminate bias).

OK, that's estimation. Now suppose that we wish to test a restriction. Note that the maximized log-likelihood can be simplified to:

$$\max \log L = -\frac{N}{2} \log(2\pi SSR/N) - \frac{N}{2SSR} SSR \tag{12.45}$$

$$= -\frac{N}{2} \left[\log(SSR) + \log(2\pi/N) + 1 \right] \tag{12.46}$$

Letting RSS and USS be the restricted and unrestricted sum of squared residuals respectively, the likelihood ratio test statistic is:

$$LR = 2 \left[\log L^U - \log L^R \right] \tag{12.47}$$

$$= -\frac{N}{2} \left[\log(USS) - \log(RSS) \right] \tag{12.48}$$

$$= \frac{N}{2} \log(RSS/USS) \tag{12.49}$$

Which is qualitatively what we're doing with an F-test in OLS: comparing how much worse our restricted model fits the data.

Exercises

Exercise 12.1.
In Game Theory, an indefinitely repeated game is one that is repeated until a random

condition is met. One way to implement this in an economic experiment is to roll a die after every repetition: if a 6-sided die roll is (say) four or less, then the game is repeated for another round, otherwise there are no more repetitions. For this particular stopping rule, what we achieve is a stopping probability of $\delta = 1/3$. That is, if we roll a 1, 2, 3, or 4, we continue, and if we roll a 5 or 6, we stop. One concern an experimenter might have is that the number of repetitions that two subsets of the sample had were very different. This might happen if one group had unusually long game lengths, and another had unusually short game lengths. This could be a problem because we want to attribute differences in participants' behavior to something else, like the different payoffs in the game. Therefore, it is common in situations like this to report the results of a hypothesis test that the two groups experienced similar game lengths.

`EndRound.csv` is a stripped-down dataset from an experiment of mine and some co-authors [note to self: insert citation when we actually publish it]. Each row of this file contains one instance of a repeated game. The file contains two variables: `EndRound` is the number of rounds that this game was played for, and `group` identifies whether this row correpsonds to Group 1 or Group 2 in the experiment. The `EndRound` variable was generated almost exactly as described above: during the experiment at the end of every repetition, I rolled a 20-sided die, and we played another one if the number was sufficiently low.

Let X_i be the number of rounds that participants play game i. Given the description above, X_i must follow a *Geometric* distribution, which has probability mass function:

$$p(x) = \begin{cases} (1-\delta)^{x-1}\delta & \text{if } x = 1, 2, 3, 4, \ldots \\ 0 & \text{otherwise} \end{cases}$$

You can think of this as X_i is the number of times you have to flip an unfair coin that comes up heads with probability δ, until you have seen one head.

1. What is the likelihood of observing a sample $\{x_i\}_{i=1}^N$?
2. What is the log-likelihood function? Express your answer as a function of δ, N, and the sample mean only.
3. What is $\hat{\delta}$, the maximum likelihood estimator for δ?
4. Use the data to estimate δ for each group individually, and for both groups pooled.
5. Report the p-value for the test that the two groups have the same δ (do the Likelihood Ratio test).
6. Suppose that you were unable to solve for $\hat{\delta}$ explicitly. Write a script that finds $\hat{\delta}$ (just for the pooled estimate) using:
   (a) Grid search. For this, use a grid of $\{0.01, 0.02, 0.03, \ldots 0.99\}$ (i.e. 99 evenly spaced points on the unit interval)
   (b) Newton's method.

Do you encounter any problems with either of these? How many iterations does it take Newton's method to converge to within 0.01 of $\hat{\delta}$? Discuss one advantage and disadvantage of using Newton's method over a grid search.

Exercise 12.2.
Download the Galton heights dataset. Create a dummy variable that is equal to one if the child is taller than *both* parents, zero otherwise. This will be our LHS variable of interest.

1. Estimate LPM, probit, and logit models using average parent height and the son dummy variable on the RHS. Include the log-likelihood of the probit and logit models in this table

2. Using the `margins` command, compute the marginal effects of average parent height and child sex on the probability of a child being taller than their parents.

3. Plot the predicted values of the OLS model and the probit model. Comment on these predictions.

4. Estimate another Probit model that tests whether the relationship between Pr[`taller`] and average parent height is different between sons and daughters. Use a likelihood ratio test.

5. Use this model to estimate (i) the probability that a son is taller than both his parents, and (ii) the probability that a daughter is taller than both of her partents. Put 95% confidence intervals around these numbers.

6. (*) Report the diffenence in these probailities, and a confidence interval for that number. Explain the interpretation of this number.

7. Suppose that you wanted to estimate the model:

$$\begin{aligned}\Pr[\texttt{taller} \mid X_i] = \Phi(&\beta_0 + \beta_1 \texttt{mother\_height}_i + \beta_2 \texttt{father\_height}_i \dots \\ &+ \beta_3 \texttt{father\_height}_i \times \texttt{mother\_height}_i + \beta_4 \texttt{son}_i)\end{aligned}$$

   Estimate this model using the following code:

```
generate MxFheight = mother_height*father_height
probit taller mother_height father_height MxFheight son
```

   Now explicitly derive the cross-partial marginal effect:

$$\frac{\partial \Pr[\texttt{taller} \mid X_i]}{\partial \texttt{father\_height}_i \partial \texttt{mother\_height}_i}$$

   Briefly explain the interpretation of this partial derivative.

   Do you think `Stata` gives you this when you type:

```
margins, dydx(FxMheight)
```

What went wrong?

Exercise 12.3.
Download and read the following paper:

> Duggan, Mark, and Steven D. Levitt. "Winning Isn't Everything: Corruption in Sumo Wrestling." The American Economic Review 92.5 (2002): 1594-1605

This is one of the papers discussed in *Freakonomics*.

1. Briefly explain why two Sumo wrestlers may face different incentives to win the same match.

2. Consider a simplified version of the econometric model in their Equation (1).

$$\text{Win}_{i,j,t,d} = \beta_0 + \beta_1 \text{Bubble}_{i,j,t,d} + \gamma \text{Rankdiff}_{i,j,t} + \epsilon_{i,j,t,d} \tag{12.50}$$

   In the paper, they estimated limited probability models. How would you interpret estimates $\hat{\beta}_1$ and $\hat{\gamma}$ from these? (assuming that the LPM is econometrically valid).

3. Suppose instead that you estimated a Probit model:

$$\Pr[\text{Win}_{i,j,t,d} = 1] = \Phi\left(\beta_0 + \beta_1 \text{Bubble}_{i,j,t,d} + \gamma \text{Rankdiff}_{i,j,t}\right) \tag{12.51}$$

   and obtain estimates $\tilde{\beta}_0$, $\tilde{\beta}_1$, and $\tilde{\gamma}$. Write down a functions of these estimates that has the same interpretations as your answer to part 2. Note how Duggan and Levitt define the variable $\text{Bubble}_{i,j,t,d}$.

4. Briefly explain what the data would look like if there was not any match fixing going on.

5. Duggan and Levitt point out that it is plausible that effort could explain the Bubble effect. Re-write equation 12.50 with an "effort" variable to reflect this. We typically don't observe effort, so how will this affect the estimate of β_1?

6. Briefly explain one of the ways that Levitt and Duggan try to convince the reader that (at least some of the) bubble effect is due to match fixing.

Chapter 13

Instrumental variables (2SLS)

13.1 Over-identification test

In the case that we are lucky enough to have more than one instrument, we can do an over-identification test. Qualitatively, what we are doing in this test is estimating $\hat{\beta}_1$ for for each of our instruments separately, and determining if this changes our answer in any appreciable way. There are two possible outcomes of this test:

1. If all of the $\hat{\beta}_1$s are close to each other, then we conclude that either all instruments satisfy the exclusion condition, or all instruments do not satisfy the exclusion condition.

2. If at least one $\hat{\beta}_1$ is very different from the others, then we conclude that at least one instrument does not satisfy the exclusion condition, and at least one instrument satisfies the exclusion condition.

Importantly, note that we can never know that all of our instruments are valid.

To demonstrate this, we need an example of using 2SLS with multiple instruments. Berry et al. (1995) is one such example. In particular, we are interested in estimating the demand curve for cars, and how this shifts with three characteristics:

- `air`: a dummy variable for whether the car has air conditioning as standard
- `weight`: The weight of the car (units are unspecified in the dataset, but are somewhat irrelevant for this example), and
- `hp`: how powerful the car is.

A cut-down version of the specification in Berry et al. (1995) is as follows:

$$\log(Q_i) - \log(Q_0) = \beta_0 + \beta_1 \mathtt{p}_i + \beta_2 \mathtt{air}_i + \beta_3 \mathtt{weight}_i + \beta_4 \mathtt{hp}_i + \epsilon_i \tag{13.1}$$

where Q_i is the quantity of car i sold in the market, and Q_0 is a measure of the potential size of the market (e.g. the population of consumers in the market). Econometrically, we worry that

	(1)	(2)	(3)
	delta	price	delta
price	-0.000361*		-0.0000753***
	(-2.15)		(-5.38)
air	0.753	3118.0	-0.480
	(0.76)	(1.36)	(-1.44)
weight	0.00154*	2.709	0.00124***
	(2.26)	(1.17)	(3.84)
hp	0.0494	169.6***	-0.000510
	(1.63)	(6.21)	(-0.12)
Z_air		-80797.3	
		(-1.26)	
Z_weight		104.0	
		(1.88)	
Z_hp		111.0	
		(0.15)	
Constant	-12.71***	-294891.0	-9.854***
	(-5.74)	(-1.87)	(-13.64)
Observations	131	131	131
F		46.73	14.98
method	2SLS	2SLS (1st stage)	OLS
p_overid	.		

t statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 13.1: 2SLS estimation of equation 13.1 using data from Berry et al. (1995).

price is endogenous. It is somewhat common in this field to use the average characteristics of all cars not produced by the firm that produces car i are valid instruments for p_i. I will ignore the justification of this here, but use theis as an example of an over-identification test. Specifically, we have one endogenous regressor (price), and three instruments (average air, weight, and hp of cars not made by manufacturer i). The results of this estimation are shown in Table 13.1. I also log price and run the estimations again in Table 13.2 so that the coefficient has an elasticity interpretation. Column 1 shows the 2SLS results, column 2 is the first stage regression, and column 3 is a naïve OLS specification that we should never take seriously. The F-statistic in column 2 provides strong support for the instruments satisfying the inclusion condition.

Now let's focus on the exclusion condition. We can never directly test that the exclusion condition is satisfied, but we can do an over-identification test. To implement this in STATA, simply code up `estat overid` below your 2SLS command. By default this displays two tests. In my code I get STATA to report the p-value of the first, the Sargan (score) test, in the `esttab` tables. In both tables, the p-value is large, so we conclude that one of the following

	(1)	(2)	(3)
	delta	log(price)	delta
log(price)	-8.031**		-2.192***
	(-3.02)		(-5.78)
air	1.625	0.257**	-0.141
	(1.68)	(3.11)	(-0.41)
weight	0.00272***	0.000273**	0.00159***
	(3.68)	(3.29)	(4.86)
hp	0.0389*	0.00626***	0.000674
	(2.11)	(6.39)	(0.15)
Z_air		-3.318	
		(-1.44)	
Z_weight		0.00514*	
		(2.60)	
Z_hp		0.00304	
		(0.12)	
Constant	55.64**	-5.996	8.574**
	(2.59)	(-1.06)	(2.73)
Observations	131	131	131
F		94.11	16.33
method	2SLS	2SLS (1st stage)	OLS
p_overid	.		

t statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 13.2: Table 13.1, but with logged price, so that we can interpret the coefficient as an elasticity.

	(1)	(2)	(3)
	delta	delta	delta
log(price)	-207.7	-110.0*	-168.4
	(-0.83)	(-2.34)	(-0.90)
air	5.629	2.602	4.414
	(0.71)	(1.59)	(0.74)
weight	0.00571	0.00357**	0.00485
	(1.00)	(2.82)	(1.13)
hp	0.122	0.0580	0.0961
	(0.74)	(1.82)	(0.78)
Constant	426.7	221.7*	344.4
	(0.81)	(2.25)	(0.87)
Observations	131	131	131
instrument	air	weight	hp

t statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 13.3: Table 13.2, using only one instrument.

is true:

- All of the instruments are valid,
- All of the instruments are equally bad,

but we don't know which.

For the sake of completeness, I also include Table 13.3, which shows the three possible 2SLS estimations using only one instrument. The coefficients on log(price) vary wildly, but not so much as to reject the null for the over-identification tests.

The code that generated these tables follows in the panel below:

```
clear all
set more off
 import excel "cars_data.xls", sheet("PS_ cars_data") cellrange(B10:H141) firstrow
 desc
global M = 100*10^6

// generate share data
qui gen share = Q/$M

egen shareCAR = total(share)
gen share0 = 1-shareCAR

// delta (estimate of quality)
gen delta = log(share)-log(share0)

// RHS characteristics
local X "air weight hp"

// Instruments of average characteristics of cars not produced by firm i
qui sum firm
```

```
local nfirms = r(max)
foreach x of local X {
        qui gen Z_‘x’ = .
        forvalues ii= 1/‘nfirms’ {
                qui sum ‘x’ if firm ~=‘ii’
                qui replace Z_‘x’ = r(mean) if firm == ‘ii’
                }
}



// Tables using all instruments
forvalues kk = 1/2 {
eststo reg_‘kk’_1: ivregress 2sls delta air weight hp ( price = Z_*)
        estadd scalar p_overid = r(p_score)
        estadd local method "2SLS"
eststo reg_‘kk’_2: regress price air weight hp Z_*
        estadd local method "2SLS (1st stage)"
eststo reg_‘kk’_3: regress delta air weight hp price
        estadd local method "OLS"

replace price = log(price)


esttab reg_‘kk’_* using cars‘kk’.tex, label scalars(F method p_overid) compress nogaps
    ↪ replace
label variable price "log(price)"
}

// Table using each instrument separately
foreach x of local X {
        eststo reg0_‘x’: ivregress 2sls delta air weight hp ( price = Z_‘x’)
        estadd local instrument "‘x’"
}
esttab reg0_* using cars0.tex, label compress nogaps replace scalars(instrument)
```

Exercises

Exercise 13.1.
Bailey (2016) claims that (see p305):

> A weak instrument does a poor job of explaining the endogenous variable (X). Weak instruments magnify the problems associated with quasi-instruments and also **can cause bias in small samples**.

We will explore this today.

Specifically, simulate the sampling distribution of $\hat{\beta}_1^{\text{2SLS}}$ from the following data-generating

process for $N = 1000$:

$$Y_i = X_i + \eta_{1,i} \tag{13.2}$$
$$X_i = 0.1Z_{1,i} + 0Z_{2,i} + 0Z_{3,i} + 0Z_{4,i} + 0Z_{5,i} + 0.1Z_{6,i} + \eta_{2,i} \tag{13.3}$$
$$Z_{i,1}, Z_{2,i}, \ldots Z_{5,i} \sim iidN(0,1) \tag{13.4}$$
$$Z_{6,i} = \eta_{3,i} \tag{13.5}$$
$$\begin{bmatrix} \eta_{1,i} \\ \eta_{2,i} \\ \eta_{3,i} \end{bmatrix} \sim iidN\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.8 & 0.2 \\ 0.8 & 1 & 0 \\ 0.2 & 0 & 1 \end{bmatrix} \right) \tag{13.6}$$

Equation 13.6 is the formal way of stating that each vector $(\eta_{1,i}, \eta_{2,i}, \eta_{3,i})'$:

- Is independent of any other vector (i.e. uncorrelated with vectors with different subscripts)
- Each $\eta_{k,i}$ has a marginal distribution of $N(0,1)$.
- $\text{corr}(\eta_{1,i}, \eta_{2,i}) = 0.8$, $\text{corr}(\eta_{1,i}, \eta_{3,i}) = 0.2$, and $\text{corr}(\eta_{2,i}, \eta_{3,i}) = 0$

The following code will simulate draws for η from this distribution:

```
matrix M = 0, 0, 0
matrix V = (1, 0.8, 0.2\ 0.8, 1, 0 \ 0.2, 0, 1)
drawnorm eta1 eta2 eta3, n(1000) cov(V) means(M)
```

Specifically, simulate the sampling distribution of $\hat{\beta}_1^{\text{2sls}}$, the estimator for the causal effect of X on Y using the following procedures:

1. Using just Z_1 as an instrument for X
2. Using Z_1, Z_2, Z_3, Z_4, and Z_5 as instruments for X
3. Using Z_1 and Z_6 as instruments for X
4. Using Z_1, Z_2, Z_3, Z_4, Z_5, and Z_6 as instruments for X

Summarize the distributions of these four estimators both graphically and in an `esttab` table. Include the mean, standard deviation, and mode in you summary table.

Explain why specification (a) allows you to estimate the causal effect of X on Y, and why you cannot make this claim with the other specifications. Discuss these in the context of your simulation results.

Exercise 13.2.
Can we include endogenous controls in our 2SLS estimation? Produce a simulation that investigates this concept. Specifically, discuss the results of two simulations showing:

1. That if controls are exogenous, the estimate of the causal effect is unbiased

2. That is controls are endogenous, the estimate of the causal effect is biased.

Before you start your coding, have a think about the *simplest* data-generating process that will prove your point.

Hint: The following code will simulate draws from a multivariate normal distribution with zero mean, unit marginal variances, and pre-defined correlations `‘c1’, ‘c2’`:

```
matrix M = 0, 0, 0
matrix V = (1, ‘c1’ ‘c2’\ ‘c1’, 1, ‘c2’ \ ‘c1’ ‘c2’ 1)
drawnorm r1 r2 r3, n(1000) cov(V) means(M)
```

FYI, if you are interested in this problem, you should read Marc Bellemare's blog post about this, and Frölich (2008).

Chapter 14

Time series

14.1 Autoregressive and moving average (ARMA) models: the basic building blocks of time series models

While there are many ways in which observations in a time series $\{Y_t\}_{t=1}^T$ could be dependent on each other, we almost always start with dependence due to an autoregressive process, a moving average process, or a combination thereof. The autoregressive model assumes that Y_t depends on the variable's lags, for example:

$$Y_t = 1 + 0.2Y_{t-1} + \epsilon_t, \quad \epsilon_t \sim iid,\ E[\epsilon_t] = 0,\ V[\epsilon_t] = \sigma^2 \tag{14.1}$$

Is an AR(1) process, which means that the deterministic component of Y_t is a (linear) function of the realization of same variable in the previous period.

In general, we can write an autoregresive process as:

$$Y_t = \alpha + \sum_{\tau=1}^{k} \phi_\tau Y_{t-\tau} + \epsilon_t \tag{14.2}$$

where k is the number of lags of Y_t included in the model, and $\{\phi_\tau\}_{\tau=1}^k$ are the coefficients on these lags. If there are k lags in the model, we call this an "AR(k)" model.

Figure 14.1 shows a white noise process (panel (a), basically just iid errors with no dependence), followed by three autoregressive processes. Panel (b) shows an AR(1) process with $\alpha = 0$ and $\phi_1 = 0.7$. Compared to panel (a), in this plot large values of Y are likely to be followed by another large value. This is because, on average, $E[Y_t \mid Y_{t-1}] = 0.7Y_{t-1} > 0$.

The other building block of time series processes is the moving average. An example of this is:

$$Y_t = \psi\epsilon_{t-1} + \epsilon_t \tag{14.3}$$

which is an MA(1) process. The "MA" part means "moving average", in that Y_t is a (weighted) average of errors that have occurred in the past. The "(1)" part means that only

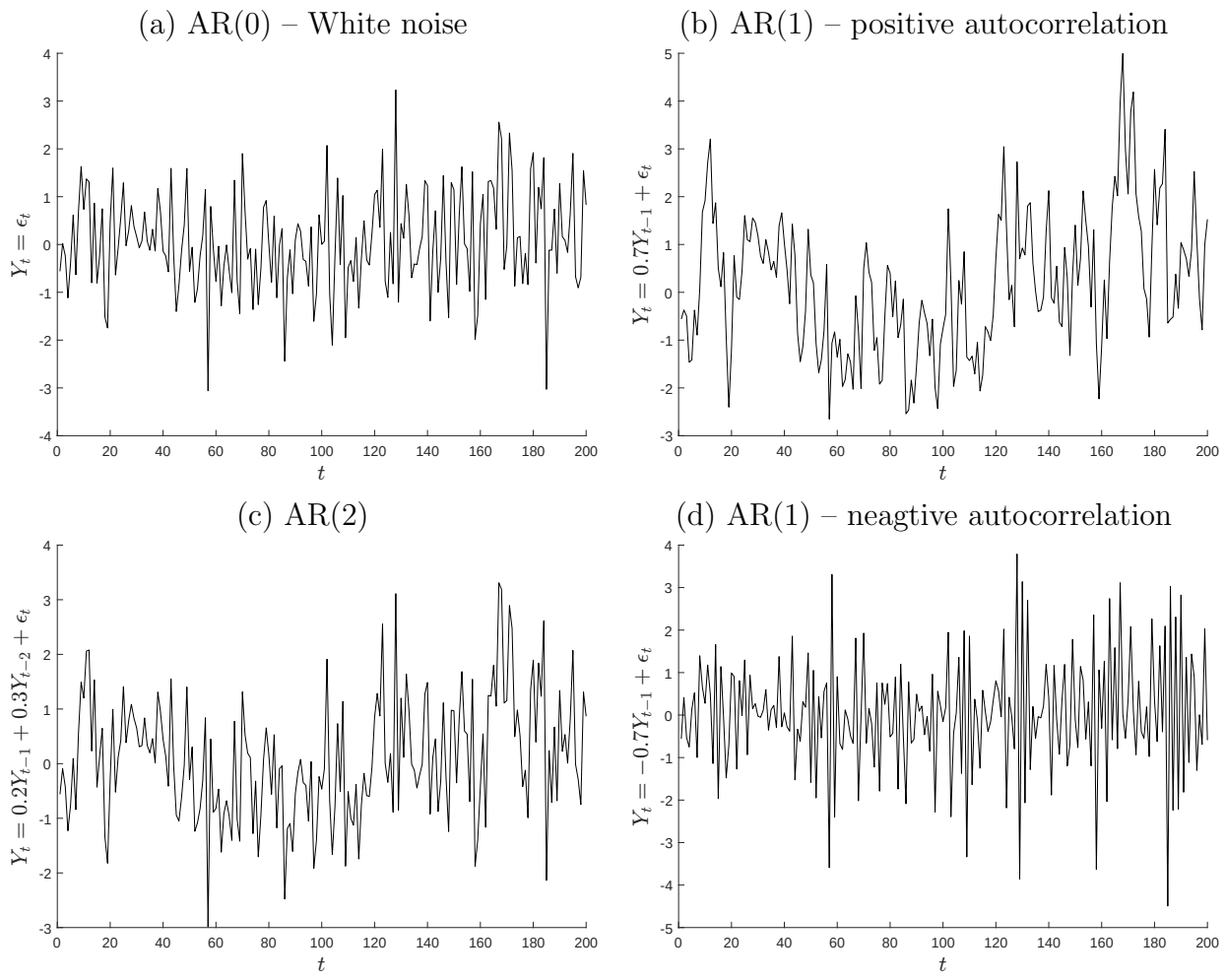


Figure 14.1: Simulated autoregressive processes.

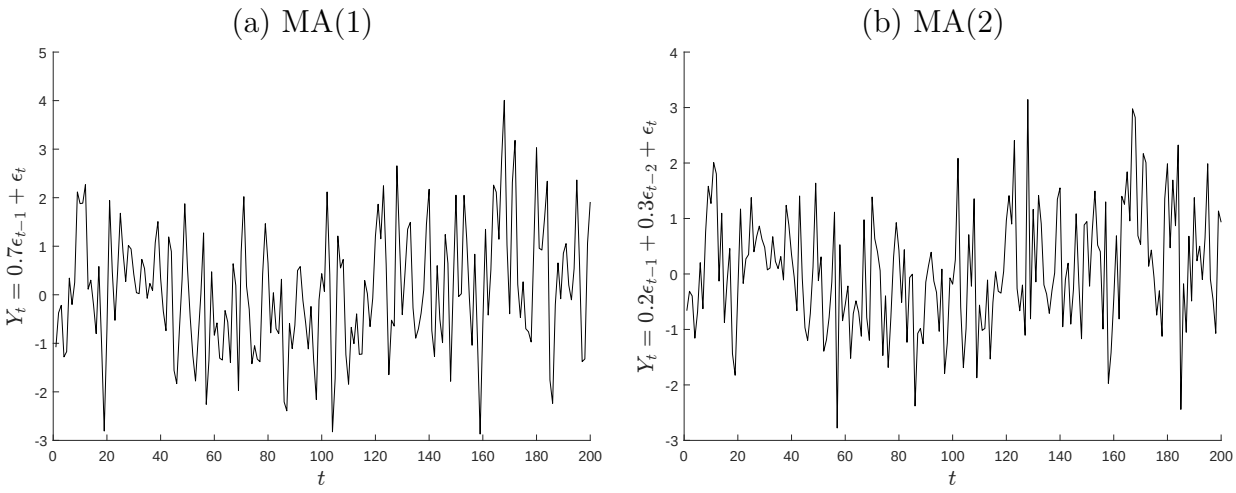


Figure 14.2: Simulated moving average processes.

the error that occurred one time period into the past (i.e. ϵ_{t-1}) shows up in this process. In general, we can write an arbitrary MA(k) process as:

$$Y_t = \alpha + \sum_{\tau=1}^{k} \psi_\tau \epsilon_{t-\tau} + \epsilon_t \tag{14.4}$$

Note that while these processes look almost exactly the same, except that for the moving average process, we are lagging the errors, instead of the Y. Some simple moving average processes are shown in Figure 14.2. It might be difficult to spot the difference between these and the autoregressive processes in Figure 14.1, this is why the autocorrelation function and partial autocorrelation function are useful.

Bailey (2016, chapter 13) also discusses at length an "autoregressive errors" (thi is what I'm calling it, not Bailey) regression model that takes the form:

$$Y_t = \beta_0 + \beta_1 X_t + \epsilon_t, \quad \epsilon_t = \rho\epsilon_{t-1} + \nu_t \tag{14.5}$$

Note that the equation for Y_t is neither an AR(1) nor MA(1) process, but the equation for ϵ_t *is* and AR(1) process.

14.2 Stationarity and properties of ARMA processes

Until we get to the unit root problem, we are going to implicitly assume that we are dealing with *stationary* processes. Intuitively, this means that if you pick two time periods, say t and s, then any *unconditional* beliefs you have about Y_t and Y_s are going to be the same. Additionally, any *unconditional* beliefs you have about Y_{t+1} and Y_{s+1} sill be the same. In fact any *unconditional* beliefs you have about $Y_{t+\tau}$ and $Y_{s+\tau}$ for any τ will be the same.

Please read through this again and underline, highlight, etc. the "unconditional" part. Here's an example. Consider the AR(1) process:

$$Y_t = 0.5Y_{t-1} + \epsilon_t \tag{14.6}$$

which happens to be stationary. If I asked you to make a forecast of Y_t at a particular time, but gave you no more information than the above equation, then you'd probably (and quite rightly) note that since you don't know Y_{t-1}, or any of the other Ys that came before it, and you know the errors are all mean zero, that a good point prediction would be to choose $E[Y_t] = 0$. But if I asked you to make a prediction for some other time period s, you would have done exactly the same thing: since I've given you nothing to refine your beliefs, this is the best you can do. Now if I told you the actual value of Y_{t-1}, you'd be able to make a better forecast of Y_t, namely $E[Y_t \mid Y_{t-1}] = 0.5Y_{t-1}$, but then you'd be conditioning on something.

For our purposes, we will think about stationary time series as follows:

Result 1. *If a process Y_t is stationary, then (among other things):*

$$E[Y_t] = E[Y_{t+1}] = E[Y_{t-1}] = E[Y_{t+\tau}] \qquad \text{for all } \tau \in \mathbb{N} \tag{14.7}$$

$$V[Y_t] = V[Y_{t+1}] = V[Y_{t-1}] = V[Y_{t+\tau}] \qquad \text{for all } \tau \in \mathbb{N} \tag{14.8}$$

$$\text{cov}(Y_t, Y_{t+1}) = \text{cov}(Y_{t+1}, Y_{t+2}) = \text{cov}(Y_{t+\tau}, Y_{t+1+\tau}) \qquad \text{for all } \tau \in \mathbb{N} \tag{14.9}$$

$$\text{cov}(Y_t, Y_{t+s}) = \text{cov}(Y_{t+\tau}, Y_{t+\tau+s}) \qquad \text{for all } \tau, s \in \mathbb{N} \tag{14.10}$$

Basically all of those things are constant. Importantly, note that in general:

$$\text{cov}(Y_t, Y_{t+s}) \neq \text{cov}(Y_t, Y_{t+\tau}) \qquad \text{for all } \tau, s \in \mathbb{N} \tag{14.11}$$

which of course isn't even true for our AR(1) process above, because the effect of Y_t on Y_{t+s} diminishes as s gets larger. In fact, this is another property of stationary processes: as we want to forecast further and further into the future, any information we have now becomes more and more useless.

14.3 Diagnostics

Like most chapters, Bailey jumps right into a concept's implication for OLS without going over some more fundamental concepts. I cover some that I think are important here. Specifically, we focus on a univariate time series $\{Y_t\}_{t=1}^t$, and the implications of autocorrelation on the properties of a sample mean. In particular, we may be worried that at least one of the following are true:

1. $\bar{y}$ is a biased estimator of $E[Y_t]$
2. The method we use for calculating standard errors for $\bar{y}$ assume that the time series is *not* serially correlated, and so we may be over- or under-stating significance.

14.3.1 Autocorrelation and partial autocorrelation functions

Autocorrelation and partial autocorrelation functions are useful ways to graphically represent the serial correlation in a time series. For stationary time series, the autocorrelation function (ACF) is defined as:

$$R(\tau) = \frac{E[(Y_t - \mu)(Y_{t-\tau} - \mu)]}{\sigma^2} \tag{14.12}$$

In words, this is the correlation between our random variable, and its value τ periods in the past, or $\text{corr}(Y_t, Y_{t-\tau})$. If Y_t is an MA(k) process, then $R(\tau) \neq 0$ for $\tau = k$, and zero for $\tau > k$.

The partial autocorrelation function is defined recursively, and involves projection matrices. In upholding my promise to not go into matrix algebra, I will hold off on the formal definition, and provide this intuition instead: You estimate the model:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta_\tau Y_{t-\tau} + \epsilon_t \tag{14.13}$$

The partial autocorrelation function is equal to:

$$\alpha(\tau) = \text{plim}\hat{\beta}_\tau \tag{14.14}$$

That is, after controlling for all lags of lower order, how much additional explanatory power does $Y_{t-\tau}$ provide for Y_t. An AR(k) process will have $\alpha(\tau) \neq 0$ for $\tau = k$, and zero for $\tau > k$.

We can therefore use the sample analog of these to diagnose the presence of autocorrelation, and maybe even the *type* of autocorrelation present (if it is not too fancy). Fortunately, Stata calculates plots of these very simply:

```
ac Y // For autocorrelation function
pac Y // For partial autocorrelation function
```

Sometimes these are reasonably easy to spot. For example Figures 14.3 and 14.4 show these functions for an AR(1) and MA(1) process respectively. For the PACF in Figure 14.3, we only identify one significant lag: the first. This is consistent with an AR(1) process. Since the higher-order lags appear to be insignificant in the PACF, this tells us that Y_{t-1} adequately characterizes the serial correlation of Y_t. For the ACF in Figure 14.4, again we only identify one significant lag: the first. This is consistent with an MA(1) process. Since the higher-order lags appear to be insignificant in the ACF, this tells us that ϵ_{t-1} adequately characterizes the serial correlation of Y_t. We may not be so lucky. For example, Figure 14.5 shows the ACF and PACF for an ARMA(1,1) process. It is not clear from the "eyeball" hypothesis test that this is indeed the case. That said, this figure is evidence for the *presence* of autocorrelation, it is just not very helpful in identifying the *type* of autocorrelation.

14.4 Declaring time series datasets and dealing with lagged variables

So you want to do some time series, and you want to do it in *Stata*? Good! However before diving in to your analysis, it may be helpful o know how *Stata* can make it easier for you.

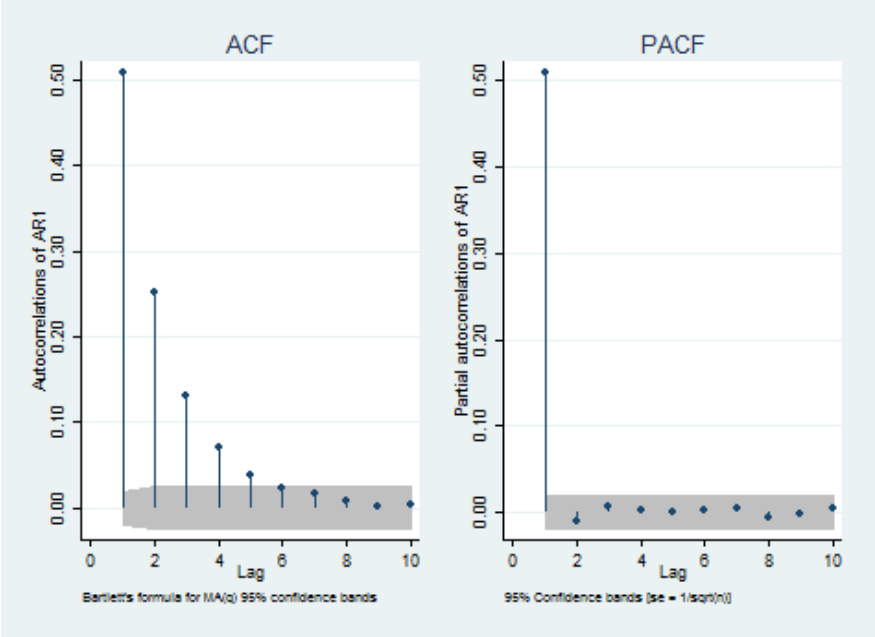


Figure 14.3: ACF and PACF of a simulated AR(1) process: $Y_t = 0.5Y_{t-1} + \epsilon_t$

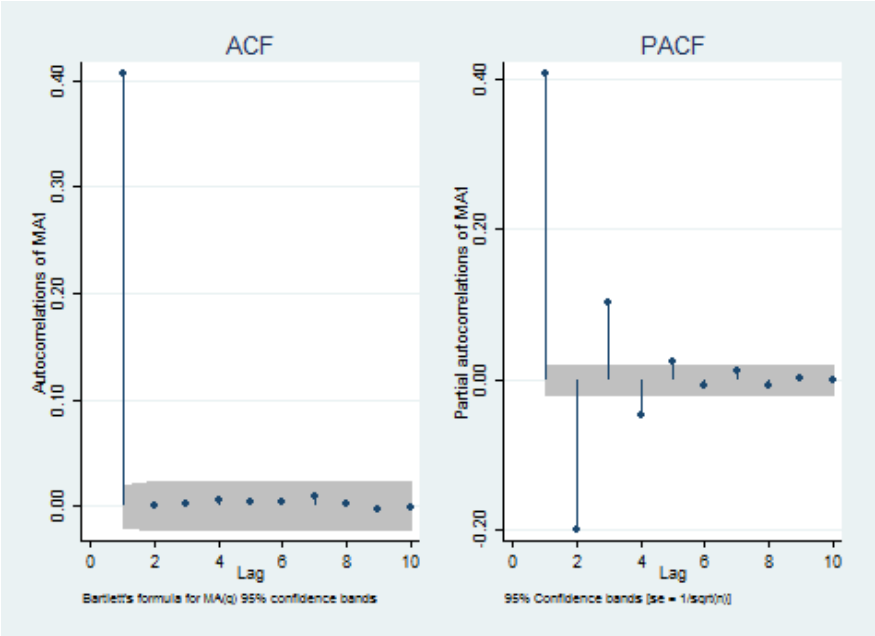


Figure 14.4: ACF and PACF of a simulated MA(1) process: $Y_t = \epsilon_t + 0.5\epsilon_{t-1}$

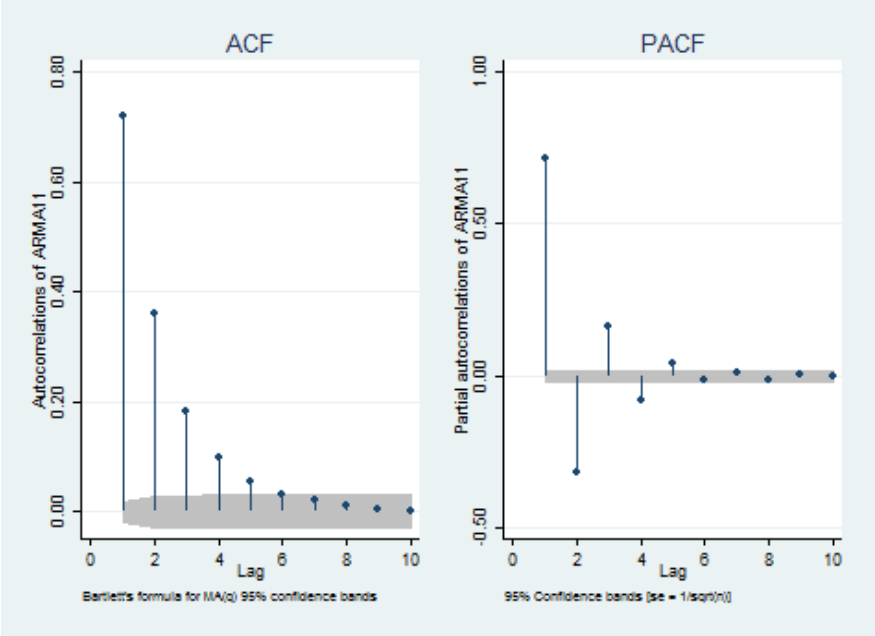


Figure 14.5: ACF and PACF of a simulated ARMA(1,1) process: $Y_t = 0.5Y_{t-1} + \epsilon_t + 0.5\epsilon_{t-1}$

For the most part, this boils down to including lagged variables in regressions, plots, and so on.

Before we learn the easy way, here's how you could do it manually. Suppose you had a single column of data `Y`, that was sorted from earliest to latest observation. If you wanted to create the lag of this variable, you could do something like:

```
generate Ylag = .
forvalues tt = 2/_N {
        replace Ylag = Y[_n] if 'tt'==_n
}
```

Furthermore, you might even realize that this works with the single line:

```
generate Ylag = [_n-1]
```

OK. Thats great. But *Stata* can help you out a bit more than this. Specifically, *Stata* has a "lag" operator that works similarly to `i.X`.[1] If you want to include the first lag of `Y`, you just `L.Y`. If you want to include its second lag, you `L2.Y`. And for the third lag, um . . . (drumroll) `L3.Y` (actually, `L1.Y` works for the first lag too). However while *Stata* is smart enough to know exactly what to do when you `i.X`, it needs to know a bit more information to use the `L.` operator. Specifically, if you have an integer variable that identifies time periods, then all you need to do it tell *Stata* what this variable is. If this variable is called `t`, then all you have to do is:

```
tsset t
```

[1]Remember that when you `i.X`, you include a dummy variable for every unique vaue of `X`.

For example, using the Toledo Airport weather dataset, the following code:

```
generate t = date(date,"YMD",1901)
tsset t
```

does this. The first line takes the string-format date, which is in ISO 8601 format (e.g. `2018-02-05`), and hence a string, into an integer equal to the number of days that date is after 1901.

14.5 Prediction and forecasting

For most of this course, we have focused on *estimating* things: sample means, probabilities, causal effects, and so on. However onve we get into modeling times series data, we might also be interested in *prediction*: or models tell us, given some information about things now, what is going to happen tomorrow, or next week, or in a year? Note that this language is *way* different from how we think about causal inference. When we seek the "right" (i.e. causal) marginal effect, we want to ask questions like "if I change X, what will this do to Y?", but with prediction, we might not actually care about the causal mechanism, we just want the number. Fortunately for us, most of our knowledge of OLS and the like follows through with *point* predictions. However we need to treat randomness slightly differently.

14.5.1 Example: Prediction with univariate problems

Suppose that we have a dataset of N iid observations of Y_i: $\{Y_i\}_{i=1}^N$. We are going to come across another Y in the future, call it Y_{N+1}, and we'd like to have some kind of idea what it will be. You might be tempted, and you'd also be on the right track, to take the sample mean of the N Ys that we already have, and use this as our *point prediction* of Y_{N+1}:

$$\hat{Y}_{N+1} = \frac{1}{N}\sum_{i=1}^{N} Y_i = \bar{Y} \tag{14.15}$$

This is a great place to start! In fact, if you were going to be making a decision based on a point prediction, and your payoff of this decision was decreasing in the mean squared error of your prediction, i.e. $E[(\hat{Y}_{N+1} - Y_{N+1})^2]$, then this would be *very* good.

But being a good econometrician, you also want to eYpress some level of uncertainty in your prediction. Again, you might be tempted to report something like the confidence interval:

$$\hat{Y}_{N+1} \pm 1.96\sqrt{\frac{1}{N}s^2}, \quad s^2 = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \bar{Y})^2 \tag{14.16}$$

which would be nice if it was correct, but you'd be vastly overstating how much you know about Y_{N+1}. The problem is that this thing is an expression of our uncertainty about the population *mean*, not an expression of our uncertainty about Y_{N+1}. If it helps, note that as N gets large, this confidence interval collapses about the point prediction, which would

be absolutely awesome: just collect a lot of data, and we'll be able to predict everything perfectly! The trouble is we really want to account for the randomness associated with getting a new draw of Y. This is a draw from the population distribution of Y, *not* a draw from the sampling distribution of $\bar{Y}$.

One fairly reasonable thing to do would therefore be to use the 2.5th and 97.5th percentiles of or sample for the prediction interval. As $N \to \infty$, these plim to the 2.5th and 97.5th percentiles of the population distribution, and so we are literally (and consistently) estimating two points for which Y has a 95% chance of falling between. Alternatively, we could get a bit more fancy and calculate the *smallest* interval that covers 95% of our data., however if the distribution is symmetric and single-peaked, we will be calculating the same thing.

Another popular way of doing this is to assume that the data come from a Normal distribution (think about whether this is a good assumption for your own application, it probably won't be). If this is the case, we can use the sample mean $\bar{Y}$ and variance s^2 to claim that:

$$Y_{N+1} \sim N(\mu, \sigma^2) \implies Y_{N+1} \overset{\text{approx}}{\sim} N\left(\bar{Y}, s^2\right) \tag{14.17}$$

Note that there are three sources of randomness here:

1. Y_{N+1} is (assumed to be) normally distributed with mean μ and variance σ^2
2. We don't know μ, but we have $\bar{Y}$, an estimate of it, which if N is large enough will be approximately $N(\mu, \sigma^2/N)$
3. We don't know σ^2, but we have s^2, an estimate of it. It can be shown that $(N-1)s^2/\sigma^2 \xrightarrow{d} \chi^2_{N-1}$. Let's assume that this process is negligible (basically you get t critical values rather than normal ones)

So we have:

$$\frac{Y_{N+1} - \bar{Y}}{\sqrt{s^2}} = \frac{(Y_{N+1} - \mu) - (\bar{Y} - \mu)}{\sqrt{s^2}} \tag{14.18}$$

$$\xrightarrow{d} \frac{(Y_{N+1} - \mu) - (\bar{Y} - \mu)}{\sqrt{\sigma^2}} \tag{14.19}$$

$$\overset{\text{approx}}{\sim} N(0,1) - N(0, 1/N) \tag{14.20}$$

$$= N\left(0, 1 + 1/N\right) \tag{14.21}$$

$$\implies 0.95 \approx \Pr\left[\left|Y_{N+1} - \bar{Y}\right| \leq 1.96\sqrt{s^2(1 + 1/N)}\right] \tag{14.22}$$

Note with the above expression, as $N \to \infty$ the $1/N$ term goes to zero, which reflects us knowing the population mean for sure.

14.5.2 Example: Prediction in bivariate OLS

OK, but what if we have some Xs as well? To begin with, our point prediction can remain the same, we simply condition on X because for bivariate OLS (hopefully the multivariate case is obvious):

$$\hat{Y_{N+1}} \mid X_{N+1} = \widehat{E[Y_{N+1} \mid X_{N+1}]} = \hat{\beta}_0 + \hat{\beta}_1 X_{N+1} \tag{14.23}$$

Again, our prediction interval needs to take into account that we are uncertain about the parameters β_0 and β_1, *and* that we are getting a new draw of Y. In the context of OLS, we are drawing a new error term.

Note that:

$$\hat{Y}_{N+1} - Y_{N+1} = \hat{\beta}_0 + \hat{\beta}_1 X_{N+1} - \beta_0 - \beta_1 X_{N+1} - \epsilon_{N+1} \tag{14.24}$$

$$= \underbrace{(\hat{\beta}_0 - \beta_0) + (\hat{\beta}_1 - \beta_1) X_{N+1}}_{B = \text{error with mean prediction}} - \epsilon_{N+1} \tag{14.25}$$

$$= \underbrace{(\hat{\alpha}_0 - \alpha_0) + (\hat{\beta}_1 - \beta_1)(X_{N+1} - \bar{X})}_{B = \text{error with mean prediction}} - \epsilon_{N+1} \tag{14.26}$$

where we make the substitution $\alpha_0 = \beta_0 + \bar{X}\beta_1$. Note that B and ϵ_{N+1} are (assumed to be) independent, we can analyze them separately. B is the component of the prediction error associated with us not knowing the conditional mean. We can calculate its variance in the same way we'd calculate the variance of a linear combination of the parameters. Noting that the population βs are constants, the estimators are unbiased, and the following results:[2]

$$\hat{\alpha}_0 - \alpha_0 \sim N(0, \sigma^2/N) \tag{14.27}$$

$$\hat{\beta}_1 - \beta_1 \sim N\left(0, \frac{\sigma^2}{\sum_i (X_i - \bar{X})^2}\right) W \tag{14.28}$$

and all of these are independent, so:

$$V[\hat{Y}_{N+1}] = V\left[\hat{\alpha}_0 + \hat{\beta}_1 (X_{N+1} - \bar{X}) - \epsilon_{N+1}\right] \tag{14.29}$$

$$= V[\hat{\alpha}_0] + V[\hat{\beta}_1](X_{N+1} - \bar{X})^2 + V[\epsilon_{N+1}] \tag{14.30}$$

$$= \frac{\sigma^2}{N} + \frac{\sigma^2}{\sum_i (X_i - \bar{X})}(X_{N+1} - \bar{X})^2 + \sigma^2 \tag{14.31}$$

$$= \sigma^2 \left[\frac{1}{N} + \frac{(X_{N+1} - \bar{X})^2}{\sum_i (X_i - \bar{X})} + 1\right] \tag{14.32}$$

which is almost like our expression for the unconditional prediction variance in the previous example. The middle term is the extra bit, which states that our prediction becomes less accurate the further away from the mean of X that we want to make predictions. Note, however, that this terms would also appear in our confidence interval for the population mean conditional on X. The first two terms will go to zero as $N \to \infty$, and we are just left with $V[\hat{Y}_{N+1}] \approx \sigma^2$ for large N.

[2]Here I just state the result for $\hat{\alpha}_0$, although we've derived the result for $\hat{\beta}_1$.

Exercises

Exercise 14.1 (Forecasting the weather, Part I. Solution provided)**.**
Use an autoregressive model with month fixed effects to forecast the maximum temperature at Toledo Airport. Estimate your model on a 70% random sample of your data, then evaluate your forecasts based on the 30% you didn't use. Generate a plot of the root-mean-squared error of your model against the number of lags you include. Show the RMSE for 1 through 100 lags.

Exercise 14.2 (Forecasting the weather, Part II)**.**
Use an autoregressive model to forecast the maximum temperature at Toledo Airport. Specifically, focus on models that include 15 RHS variables(excluding the constant) of the form:

$$T_t = \alpha_0 + \beta_1 T_{t-1} + \beta_2 T_{t-2} + \beta_3 T_{t-3} + \ldots + \gamma_1 T_{t-365\times 1} + \gamma_2 T_{t-365\times 2} + \gamma_3 T_{t-365\times 3} + \epsilon_t \tag{14.33}$$

where T_t is the maximum temperature at Toledo Airport on day t. Note that this model includes the standard lags (i.e. yesterday, 2 days ago, etc), but also includes lags going back in integer multiples of years (here we will assume away the leapyear problem).

1. Explain why including the "usual" lags (i.e. the variables on β coefficients) might be a good idea for your forecast.

2. Explain why including the yearly lags might be useful.

3. Randomly divide your sample into a *estimation* (sometimes referred to as "training") dataset (70% of observations), and a *testing* dataset (30%), and estimate all possible models like this that you could that have 15 RHS variables (i.e. 0 day lags, 15 year lags; 1 day lag, 14 year lags; 2 day lags, 13 year lags, and so on). Produce a plot of the root-mean-squared error of your forecasts against the number of day lags. *Hint:* save some time and write a `for` loop.

4. Explain why your answer is not one of the endpoints (i..e. 15 daily lags or 15 yearly lags).

Exercise 14.3.
Draw a sample of 10,000 errors $\epsilon_t \sim iidN(0,1)$, then simulate $T = 10,000$ observations of the following time series:

1. AR(1): $Y_t = 0.7Y_{t-1} + \epsilon_t$

2. AR(1): $Y_t = -0.7Y_{t-1} + \epsilon_t$

3. AR(2): $Y_t = 0.2Y_{t-1} + 0.5Y_{t-2} + \epsilon_t$

4. MA(1): $Y_t = 0.7\epsilon_{t-1} + \epsilon_t$

5. MA(2): $Y_t = 0.2\epsilon_{t-1} + 0.5\epsilon_{t-2} + \epsilon_t$
6. ARMA(2,2) $Y_t = 0.7Y_{t-1} - 0.2Y_{t-2} + \epsilon_t + 0.3\epsilon_{t-1} + 0.7\epsilon_{t-2}$
7. Noise: $Y_t = \epsilon_t$ (i.e. no lags of anything)

Then:

1. Produce a table showing summary statistics (just mean and standard deviation) of all of the above time series.
2. Produce four plots for each time series. These are (i) the empirical autocorrelation function, (ii) the empirical partial autocorrelation function, (iii) a line plot of Y_t against time, and (iv) a scatter plot of Y_t against Y_{t-1}. Combine these four plots into one figure (*Hint:* `help graph combine`). For plots (iii) and (iv), just show the last 1,000 simulated observations (otherwise it gest messy).
3. Comment on how these plots could help you identify (i) if you have autocorrelation in your data, and (ii) the type of autocorrelation in your data. Note that you can almost always do (i), but sometimes (ii) is difficult.

Exercise 14.4.
Load the provided data file `ThreeTimeSeries.dta`.

1. For each the variables `y1`, `y2`, and `y3`, plot the acf and pacf **in the same figure**. You should look up the help file on `graph combine` to see how to achieve this.
2. Use your answers to the previous part to diagnose the type of autocorrelation present in these time series. While this is generally difficult for arbitrary ARMA processes, use the following fact: (1) Each of these is either $AR(p)$ or $MA(q)$. (2) The order (i.e. p or q) is always less than 4. Your answer should include the type of autocorrelation (i.e. AR or MA), and the order (i.e. the number of lags, p or q).

Exercise 14.5.
For the following processes, calculate $E[Y_t]$, $V[Y_t]$, $\text{cov}(Y_t, Y_{t-1})$, and $\text{cov}(Y_t, Y_{t-2})$. State explicitly where you assume stationarity. ϵ_t is iid with mean 0 and variance σ^2.

1. $Y_t = \epsilon_t + 0.2\epsilon_{t-1}$
2. $Y_t = 0.8Y_{t-1} + \epsilon_t$

If you want more practice and more of a challange, try doing this for the time series variables in Exercise 14.3.

Exercise 14.6.
Consider the dynamic model:

$$Y_t = \gamma Y_{t-1} + \beta_0 + \beta_1 X_t + \epsilon_t \tag{14.34}$$

Bailey (2016) in Section 13.4 states that if:

- The errors ϵ_t are autoregressive, e.g.: $\epsilon_t = \rho\epsilon_{t-1} + \nu_t$
- $\gamma = 0$, and
- X_t is autoregressive, e.g.: $X_t = \psi X_{t-1} + \eta_t$

then $\hat{\beta}_1$ is biased.

Write a simulation that demonstrates this, and one that shows that $\hat{\beta}_1$ is not biased if $\rho = 0$ (i.e. if the errors are not autocorrelated.) Specifically, use the following parameterization:

$$\begin{gathered}\beta_0 = 0, \quad \beta_1 = 1 \\ \psi = \rho = 0.5 \\ \nu_t,\ \eta_t \sim iidN(0,1)\end{gathered}$$

Part IV

Advanced reg-monkeying

Chapter 15

Looping over variables: one `reg y x, robust`, many regressions.

In the process of doing research, unless you have perfect foresight you will be constantly updating the way you analyze your data, and how you communicate this. This could include things like:

- changing the variables that you include on the RHS of your regressions
- changing the way your regression tables are labeled
- changing the hypothesis tests that you do

All of these changes could be motivated by, for example:

- recognizing a mistake
- recognizing a better way to analyze your data or display your results
- being told by a referee or conference participant that you are doing it wrong. Especially for a referee, you should pay attention to this! (even if you think they are nuts)

To demonstrate this problem, we will investigate the determinants of speeding fines. We will use a dataset (also used in Bailey, 2016), which is a cut-down version of the dataset used in Makowsky and Stratmann (2009).

The dataset includes information about people who were pulled over for speeding. In particular, we are interested in the effect of `MPHover` (how must faster than the speed limit a person was driving) on `amount` (the fine they received). Additionally, we might be worried if the other variables in the dataset had any bearing on speeding: if they do, this could be evidence for discrimination. In order to drive home our point (that there is discrimination), we run five regressions two different ways. The five regressions are:

```
1. reg amount MPHover      //i.e. just bivariate OLS

2. reg amount MPHover age female
```

3. `reg amount MPHover Black Hispanic`

4. `reg amount MPHover StatePol OutTown OutState`

5. `reg amount MPHover age female Black Hispanic StatePol OutTown OutState`

That is, in 2-4 we control for some things that could be related, and regression 5 we put them all together. The "two different ways" are:

1. Using the entire sample

2. Only using observations of people who were actually fined (there are a lot of zeros)

We want to show 2 tables. The first for all regressions without the restriction, and the second with the restriction (only people who got fined). In addition to this, we would like to do a hypothesis test that all of the controls jointly do not affect fines (i.e. no discrimination), and include the p-value in our regression tables. By my reckoning we need about 3 lines of code per regression:

```
eststo reg_1: regress amount mphover controls...  if restriction
test controls
estadd scalar p=round(‘r(p)’,0.0001)
```

then for each restriction:

```
esttab reg_*, scalars(p) ...
```

so we're looking at $2 \times 5 \times 3 = 30$ lines of code for the regressions and hypothesis tests, and another 2 to get the table outputs. More to the point, if we want to add another set of controls, we need to code up another $3 \times 2 = 6$ additional lines, as well as change the code for the last column on the table (which would now be the 6th column). This seems tedious, and a great way to make a mistake. Instead, we are going to do the following:

1. Define our three sets of controls

2. Define the two restrictions

3. Loop over the controls and restrictions

This way, if we want to add or remove some controls, or a referee/seminar attendant says we deed to do something differently, we simply change one line of code in steps 1 or 2, and STATA will take care of the rest:

```
clear all
set more off
import delimited "M08_speeding_tickets_text.csv"

desc

summarize

// how Stata deals with strings
```

```
local strA "stringA" //here we are generating two variables, strA and strB and 'adding' them
    ↪  togehter to create a vairable called both.
local strB "stringB"

        local both "`strA' `strB'"
        disp "`both'"
/*In the coding literature this called string concatenation. We are 'adding' strings
    ↪ together so we can define our controls and then run multiple regressions adding in a
    ↪ control at a time.  Note that there are some missing values in "amount". These are
    ↪ recorded in the data editor as ".". Let's assume that in these cases no fine was
    ↪ issued. We will create a variable called "nofine", and replace the missing values
    ↪ with zeros.
*/

generate nofine = 0
replace nofine = 1 if amount==.
replace amount = 0 if nofine ==1
summarize

// Define the controls here
local control_0 ""
local controllabel_0 "-"
local control_1 "black"
local controllabel_1 "black"
local control_2 "female"
local controllabel_2 "female"
local control_3 "outtown outstate statepol"     // location controls
local controllabel_3 "location"
local control_4 "hispanic"
local controllabel_4 "hispanic"
local controllabel_5 "all"


//when we go through the outdie loop `ii' for the first time, ii=-1, which will include all
    ↪ the data since all the amounts are 0 or greater,
//the second time we go through the outside loop, since we have a strict inequality we are
    ↪ going to exclude all fines equal to 0, i.e. only run on peple who got fined.

forvalues ii = -1/0 {
        local control_all = ""
         forvalues cc = 0/4 {
                        // All regressions except the one with all controls are done here
                        quietly eststo reg_`cc': regress amount mphover `control_`cc'' if
                            ↪ amount>`ii'
                        estadd local controls `controllabel_`cc''
                        local control_all = "`control_all' `control_`cc''"
                        disp "`control_all'"
                }
                // The final column with all controls
                quietly eststo reg_all: regress amount mphover `control_all'  if amount>`ii'
                estadd local controls `controllabel_5'
                esttab reg_* using Looping_over_variables`ii'.tex, se compress nogaps
                    ↪ replace keep(mphover) scalars(controls)
                drop *reg_*
}
```

The above code produces the following Tables 15.1 and 15.2

Exercise 15.1.
You are unsure whether `amount` and/or `mphover` should be logged or not in Table 15.2.[1] Write a script that produces an `esttab` table with four columns, corresponding to the 4

[1]Note that this doesn't make much sense with Table 15.1 because we can't (or at least shouldn't) log the

	(1)	(2)	(3)	(4)	(5)	(6)
	amount	amount	amount	amount	amount	amount
mphover	8.345***	8.340***	8.273***	8.031***	8.323***	7.971***
	(0.0437)	(0.0437)	(0.0436)	(0.0402)	(0.0437)	(0.0401)
N	68357	68357	68357	68357	68357	68357
controls	-	black	female	location	hispanic	all

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 15.1: All data used

	(1)	(2)	(3)	(4)	(5)	(6)
	amount	amount	amount	amount	amount	amount
mphover	6.886***	6.889***	6.871***	6.899***	6.884***	6.887***
	(0.0385)	(0.0385)	(0.0385)	(0.0382)	(0.0385)	(0.0382)
N	31674	31674	31674	31674	31674	31674
controls	-	black	female	location	hispanic	all

Standard errors in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Table 15.2: Restricting to positive amounts.

possible combinations of logging or not logging these variables. You may use the `regress` command exactly once.

zeros. If we applied this same script to the whole dataset, the columns with logged `amount` on the LHS will be for regressions dropping all observations with no fine (i.e. the restriction in Table 15.2).

Part V

Simulation techniques

Chapter 16

An introduction to Monte Carlo techniques

16.1 Stata's (pseudo) random number generators

16.2 Using random number generators

16.3 *Stata*'s `simulate` command

Stata allows you to break up the simulation process into two steps. This is helpful because you can concentrate on getting one thing done well at a time. These steps are:

1. Write a program that simulates *one* draw from the distribution you are trying to simulate.

2. Use *Stata*'s `simulate` command to run this program over and over again lots of times. It puts a "sample" from this simulated distribution in the data editor.

To begin with, let's work through the example for the `simulate` function in Stata's help file. You can access this by typing: `help simulate`. This example simulates draws from a lognormal distribution: if $X \sim N(\mu, \sigma^2)$, then $Y = \exp(X) \sim \text{lognormal}(\mu, \sigma^2)$, i.e. $\log(Y) = X \ N(\mu, \sigma^2)$, hence if you log Y, it has a normal distribution.

We would like to simulate the distributions of the sample mean and variance of $Y \sim \text{lognormal}(0, 1)$. To do this, we will need to:

0. Set the sample size to simulate

1. draw $X \sim N(0, 1)$

2. Generate $Y = \exp(X)$

3. Summarize Y (to get the mean and variance of our simulated sample)

4. Store the mean and variance as data

5. Go back to step 1. Stop when we have done this enough that we have approximated the distribution well.

Steps 0-3 for a sample size of 20 on their own would be:

```
set obs 20                  // (0)
generate X = rnormal(0,1)   // (1)
generate Y = exp(X)         // (2)
summarize Y                 // (3)
```

Note that we can access the stored results of summarize using `‘r(.)’`, specifically:

```
display "‘r(mean)’"
display "‘r(Var)’"
```

To see what else we can access from summarize, type `help summarize` into the command line.

If we want to use this procedure for simulating data, it will take a long time. A more elegant way to set this problem up is to write a program that performs steps 0-3, then let *Stata*'s `simulate` program do steps 4 and 5. The first step is to write a program. This one is a cut and paste from the `simulate` help file, then I have commented above each line explaining what it does:

```
// Clear everything in the memory so that we know that we are starting fresh
clear all
/*Tell stata that we are writing a program. It knows that everything between
here and "end"  is part of the program  The program name is lnsim (i.e.
lognormal simulation) rclass lets us know that we can access variables generated
 by the program through ‘r(.)’  (more on this later)
*/
program define lnsim, rclass
                // Sometimes newer and older versions of Stata work slightly differently.
                    ↪ Make Stata behave as if it’s Stata 13
                // This line is not always needed
        version 13
                /*
                define the syntax of the program
                Here we let Stata know the inputs to the program
                These inputs are:
                        obs = number of observations. It must be an integer, and by default
                            ↪ is equal to 1
                        mu  = parameter mu in the lognormal distribution. It must be a real
                            ↪ number, and by default it is equal to 0
                        sigma = sigma in the lognormal distribution. It must be a real
                            ↪ number, and by default it is equal to 1

                If you don’t specify these inputs when calling the function, Stata will use
                    ↪ the default values.
                */
        syntax [, obs(integer 1) mu(real 0) sigma(real 1) ]
                // Drop all variables from the memory
        drop _all
                // Set the number of observations in the dataset (i.e. step 0)
        set obs ‘obs’
                // Define a temporary variable called z (it will be deleted when the program
                    ↪  finishes)
        tempvar z
```

```
                // generate z but taking the exponential of a normal random variable with
                    ↪ mean mu and standard deviation sigma
                // This is our random sample
        gen ‘z’ = exp(rnormal(‘mu’,‘sigma’))
                // Summarize z. This calculates the mean and variance of the random sample
        summarize ‘z’
                // Tell Stata to store the mean calcualted in summarize as a scalar called
                    ↪ mean
        return scalar mean = r(mean)
                // Tell Stata to store the Variance calcualted in summarize as a scalar
                    ↪ called Var
        return scalar Var  = r(Var)
end
```

If you run this script as is, you probably won't notice much. What is going on in the background is *Stata* adds this function `nlsim` to its memory, so you can now use it just like you would use `summarize` or `tabulate`. To see this, once you have run the above script, try typing the following into the command line (or pasting it below this script):

```
// Check that the program works by itself
lnsim , obs(20) mu(0) sigma(1)
display r(mean)
display r(Var)
```

This will display the sample mean and variance of your simulated sample of 20 observations.

However we want a "sample" of the sample mean and variance, not just one observation. If you really had nothing better to do, you could click run 1,000 times (actually this is probably not enough repetitions) and copy and paste the numbers into a spreadsheet. Fortunately, *Stata* can do this for you. Here's how:

```
simulate SampleMean = r(mean) SampleVariance = r(Var), reps(1000): lnsim, obs(20) mu(0)
    ↪ sigma(1)
```

which will give you a dataset that looks something like this (just showing the first 10 draws):

```
. list in 1/10
     +---------------------+
     | Sample~n   Sample~e |
     |---------------------|
  1. | 1.208445   .8967296 |
  2. |  1.50637   1.604306 |
  3. | 2.138609    6.13594 |
  4. | 2.690951   6.772248 |
  5. |  1.46579   1.116809 |
     |---------------------|
  6. | 1.088434   .5308753 |
  7. | 1.803651    3.82006 |
  8. | 2.250783    7.55613 |
  9. |  1.89442   6.097122 |
 10. | 1.069643    .364791 |
     +---------------------+
```

So we have a "sample" of sample means and variances. The following script will do all of this for 10,000 repetitions, then outputs some histograms of

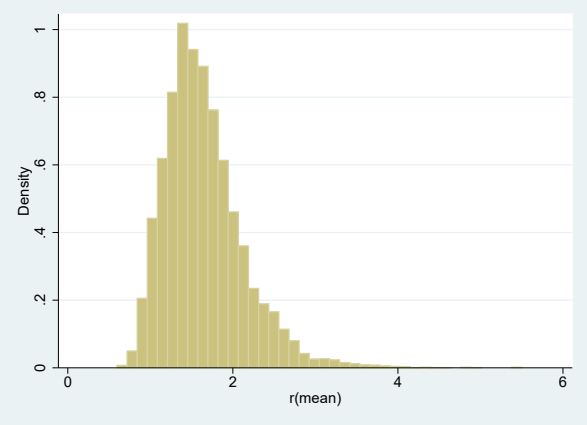
- The simulated sample mean and variance, See Figure 16.1, and
- The simulated t-statistics testing $H_0: \ E[X] = \exp(1/2)$ (this is the true population mean). Specifically:

$$t = \frac{\bar{x} - \exp(1/2)}{\sqrt{s_X^2/N}}$$

  See Figure 16.2. Alarmingly, the last 3 lines of this script calculates that about 15% of these t-statistics are greater than 1.96 in absolute value, but 1.96 is the critical value for the 5% test. We would be rejecting H_0 much too frequently!!

```
clear all

// run the script where I have defined the function lnsim
do ExampleLogNormal01

// check that the program works by itself
lnsim , obs(20) mu(0) sigma(1)
display r(mean)
display r(Var)

set more off
set seed 42
simulate SampleMean = r(mean) SampleVariance = r(Var), reps(10000): lnsim, obs(20) mu(0)
    ↪ sigma(1)
list in 1/10

// histograms of means and variances
histogram SampleMean
graph export ExampleLogNormalMean.pdf, replace
histogram SampleVariance
graph export ExampleLogNormalVariance.pdf, replace

// t-statistics
generate t = (SampleMean-exp(1/2))/sqrt(SampleVariance/20)
histogram t
graph export ExampleLogNormalT.pdf, replace

generate reject = 0
replace reject = 1 if abs(t)>1.96
summarize reject
```

(a) Sample mean



(b) Sample variance

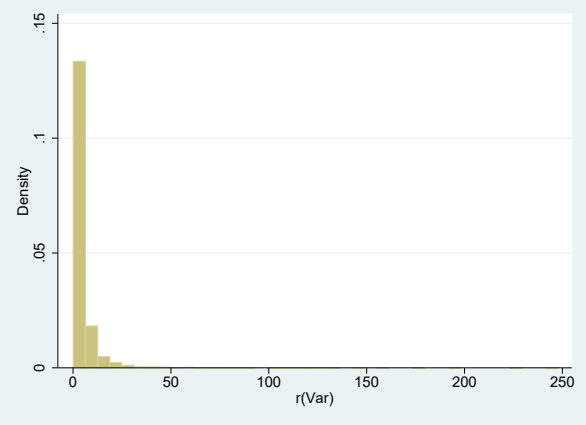


Figure 16.1: Simulated sample means and variances of 10,000 draws from $\log(X_i) \sim iidN(0,1)$.

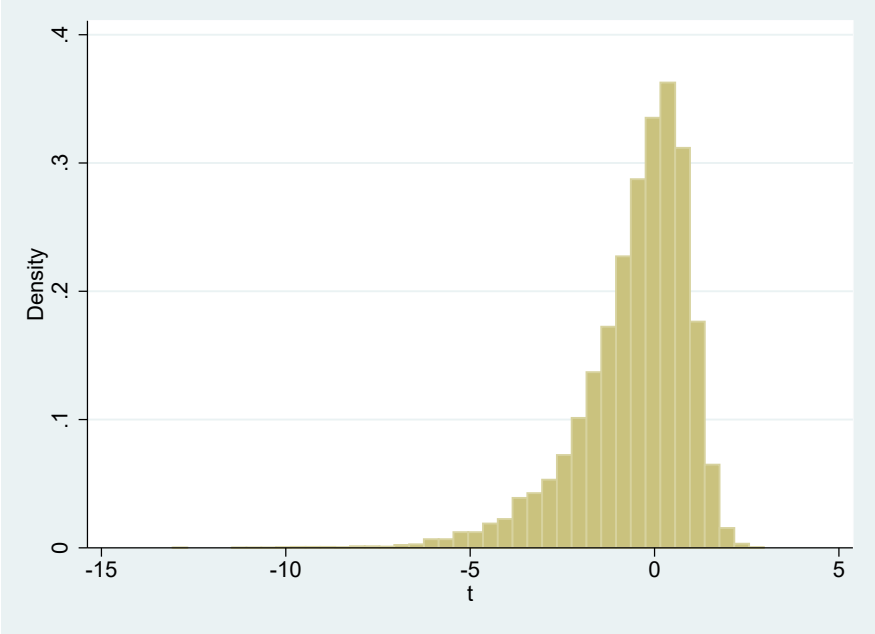


Figure 16.2: Simulated t-statistics for the hypothesis that $E[X] = \exp(1/2)$ (which is true in this case).

Exercises

Exercise 16.1 (Power calculations)**.**
In this exercise, we will investigate an application of the power calculation. This can be useful in (at least two) stages of research:

- Once your data are collected and analyzed, you can defend a null result by showing that an economically significant false null would be identified by your test a good fraction of the time.
- Before you collect your data, it may be able to help you collect a better data set.

We will focus on the second here.

Suppose that we wish to test that the means of two subsets of the population are equal. To fix ideas, consider a drug trial where we have a treated group and a control group. Since we have a finite budget, we can only collect 100 observations. How many people should be in the treatment group, and how many in the control?

Let random variable X be some measure of an individual patient health in the control group, and Y be the same measure of patient health in the treatment group. A reasonable hypothesis to test is:

$$H_0 \; E[X] = E[Y] \quad H_A : \; E[X] > E[Y]$$

To test this, we assign N_Y test subjects to the treatment, and $100 - N_Y$ to the control. We therefore have samples:

$$\{X_i\}_{i=1}^{100-N_Y}, \quad \{Y_i\}_{i=1}^{N_Y}$$

A simple test of the above hypothesis using data like this is the two-sample t-test, which is outlined here, and can be easily implemented using Stata's `ttest` command. Please read about this test, it is a very useful one, but for now we take it that it is the right one for this example. For this exercise, we ask the question:

How many observations should we assign to the treatment group?

This is going to be a function of the distributions of X and Y, and how economically significant the difference in means has to be to be excited about the new drug. For the purpose of this simulation, we want to be able to maximize the power of a 5% test when $E[Y] - E[X] = 1$. That is, if Y is (in expectation) one unit better that Y, then we want our test to be able to reject the null as frequently as possible. To further simplify things, fix the population DGP as:

$$X_i \sim iidN(0, 1), \quad Y_i \sim iidN(1, 2)$$

1. Write a program that simulates 100 draws (total) from X and Y, and outputs the p-value of the t-test assuming different variances. The program should take N_Y as an input.

2. Simulate the distribution of the p-values when the null is false. That is, when X and Y conform to the distributions above. Do this for a reasonable range of N_Y. E.g. 20, 40, 60, 80.

3. For each value of N_Y, calculate the fraction of times that you would reject the null on a 5% test (i.e. what fraction of p-values are less than 5%?)

4. Find the N_Y that maximizes this fraction.

   Extensions: The intersection of experiment design, econometrics, and economics!

   Your payoff from this trial is \$1bn times the power of this test:

5. You have a budget of \$100,000, Each control observation costs \$100, and each treatment observation costs \$200. How do you allocate your budget to maximize payoff? Is it optimal to spend the entire budget?

6. How much would you be willing to pay to reduce the variance of Y (e.g. by using better testing equipment)? How does this change your allocation of 100 test subjects between treatment and control? Express your answer as an elasticity of demand for precision ($\frac{1}{\sigma_Y^2}$). That is, report: $\frac{\partial \sigma_Y^{-2}}{\partial P}\frac{P}{\sigma^{-2}}$

Chapter 17

Simulations with OLS

When running simulations for regressions, we often want our source of randomness to be only from the error term, and not from repeated draws of right-hand side variables. Therefore when running simulations for regressions, we need to be able to keep some variables in the constant for all simulation steps, while randomly drawing the component that we are interested in: usually the error term.

To help understand this, note that frequently we derive result for OLS estimators assuming that the RHS variable(s) are constant. That is, we think about the thought experiment where we collect the same X data, but get different draws of ϵ every time.

Sadly, when we write our simulation program, we need to include a `drop _all` command at the beginning so that Stata allows us to overwrite data in the memory. Fortunately, there are two easy fixes to this. One of them is better because it uses the processor and memory less. The following discussion is motivated from Example 2 in the Stata documentation on the simulate function, which can be found here.

17.1 Method 1: Load the variables you want to keep constant when you run the program

0. Before running your program, generate the variables you want to keep constant. Them save them to the hard drive. For example, if you want to keep your RHS variable x constant. E.g.: if you want to have X distributed N(0,1), with the same draws every time:

```
clear _all
set obs 100
generate x = rnormal()
save keepx.dta
```

1. Start the program, Clear any data in the memory

```
program define myprogram // etc, put the relevant things in here
clear _all
```

2. load x from your stored file `keepx.dta`

```
use keepx.dts
```

3. generate `y`, for example if we want the true model to be $y = 1 + 2x + \epsilon$, with the error term distributed $N(0, 0.1^2)$:

```
generate y = 1+2*x + 0.1*rnormal()
```

4. Run the regression, then end

```
regress y x
end
```

   The program will automatically return anything regress stores in its results. Type `help regress` and look at what is in `e(.)`.

This method is cumbersome because it requires Stata to load your `x` variable every time. It would be faster if it did not have to (reading and writing to the hard drive takes time). Fortunately, there is:

17.2 Method 2: Keep `x` in memory

This method takes advantage of *Stata*'s `capture` command. One useful feature of the `capture` command is that it will allow your `.do` file to proceed to the next line, even if it returns an error. Here's why it is useful in our case. We need to run a script like this one (note that `drop y` is commented out here):

```
clear all
set seed 42
set obs 20

generate x = rnormal()

program define myprogram, rclass
        //drop y
        generate y = x + rnormal()*0.1
        regress y x
end

simulate b = _b[x], reps(100): myprogram
```

So the first `generate` gets us our RHS variable `x`, which we want to keep constant. The program `myprogram` will work just fine on the first simulation step, but then `Stata` will kick up a fuss on the second step because when it comes to `generate y = ...`, this variable already exists. Alternatively, if we uncomment `drop y`, then on the first step `Stata` gives us an error because `y` does not exist. We need a line just before this one to tell *Stata* something like "if `y` exists, drop it, otherwise do nothing". This is where `capture` come in. If we replaced the line `//drop y` with `capture drop y`, then this will work. Specifically, on the

first step `drop y` returns an error, but `capture` tells *Stata* to ignore it; then on subsequent steps it does not return an error, so `y` gets dropped.

Here is the step-by-step guide to this method:

0. Before running your program, generate the variables you want to keep constant. For example, if you want to keep your RHS variable `x` constant. E.g.: if you want to have x distributed $N(0,1)$, with the same draws every time:

```
clear _all
set obs 100
generate x = rnormal()
```

1. Use *Stata*'s `capture` command to drop y if is in the memory, and do nothing otherwise. Specifically for us, if we have a variable y hanging round in the memory from a previous simulation step, we want to delete it, but we don't want to drop y because Stata will kick up a fuss on the first run through because y doesn't exist yet. So we can proceed with:

```
program define myprogram // etc, put the relevant things in here
```

2. Get rid of `y` if it is in the memory, otherwise proceed without doing anything:

```
capture drop y
```

3. generate `y`, for example if we want the true model to be $y = 1 + 2x + \epsilon$, with the error term distributed $N(0, 0.1^2)$:

```
generate y = 1+2*x + 0.1*rnormal()
```

4. As before, run the regression and end the program.

```
regress y x
end
```

Finally, no matter which method we used, we can:

```
simulate _b _se, reps(10000): myprogram
```

which simulates the distribution of the estimators for the slope and intercept (stored in `e(_b)`) and the standard errors (stored in `e(_se)`).

Exercises

Exercise 17.1 (Endogeneity)**.**
Investigate what happens when X and ϵ are correlated (this is the problem we ran into with the flu shots and death example). To do this, you should compare the case where X and ϵ are uncorrelated to a case where they are. Try this:

1. Run the simulation of y step as:

```
generate y = 1 + 2*x + rnormal()*0.1
```

2. Run another simulation with:

```
generate y = 1 + 2*x + rnormal()*0.1 + 0.2*x
```

That is, in case (1) X and ϵ are uncorrelated, and in case (2) $\text{corr}(X_i, \epsilon_i) = 0.2$.

What is the bias in these cases?

Exercise 17.2 (How many observations do I need to get a good confidence interval?)**.** Bailey (2016, ch 4) States that that the slope coefficient estimator is approximately normally distributed for large samples. But how large is large enough for this to be a good assumption? The answer to this question depends on the data-generating process. If the ϵ_is are normally distributed, then this is exactly true for any sample size. For other distributions of the error term, we can use simulation to inform us.

One consequence of the normal approximation being bad is that the probability that a 95% confidence interval (constructed using a large-sample result) contains the true value of the slope coefficient is not necessarily 95%. Neither can we be sure of whether this confidence interval will contain the true value with probability greater or less than the intended value. We will use simulation to explore this. Note that such a simulation could be used to tell us how many observations we should collect, or to tell us that, for a fixed sample size, whether we should think about constructing confidence intervals without using a large-sample approximation (more on how to do this in a couple of weeks).

We will investigate the relationship between sample size and large-sample confidence intervals when the error term is uniformly distributed. Specifically, consider the true data-generating model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$
$$\epsilon_i \sim iidU[-1, 1]$$

Since the error term is iid with mean zero we know that for large samples the slope coefficient estimator is approximately normal, but we have no guarantee that this is the case for small samples.

1. Choose 3-4 sample sizes to investigate. Remember that things approach normality on a $\sqrt{N}$ scale.

2. Choose intercept and slope coefficients (this won't change your answer too much).

3. Write a program that simulates this data-generating process and the subsequent regression, holding X fixed.

4. Run a simulation for each of your chosen sample sizes, store the slope coefficients and their standard errors

5. Use these to construct 2-sided 95% confidence intervals around the slope coefficient

6. For each sample size, generate a variable that is equal to 1 if your true value is inside the confidence interval, and zero otherwise.

7. For each sample size, compare the nominal size of the confidence interval (i.e. 95%) to the actual size of your confidence interval.

Note that if $A \sim U[0,1]$, Then $B = 2A - 1 \sim U[-1,1]$.

Chapter 18

Techniques for drawing random numbers

This section of the Masters course is going to cover some common simulation techniques. Generally, we will exploit a mathematical theorem to generate random numbers in a particular way. A good reference for this material (and the basis for a lot of my notes) is Chapter 8 of Judd (1998).

18.1 Inversion

18.1.1 What inversion is and how it works

Consider a uniform random variable $U \sim U[0,1]$ and another continuous random variable X with cdf $F_X(\cdot)$. Since X is continuous, its cdf can be inverted. Therefore we can do the following:

$$\begin{aligned} \Pr\left(F_X^{-1}(U) \leq x\right) &= \Pr\left(U \leq F_X(x)\right) && (18.1)\\ &= F_U\left(F_X(x)\right) && (18.2)\\ &= \begin{cases} F_X(x) & \text{if } F_X(x) \in (0,1) \\ 0 & \text{if } F_X(x) \leq 0 \\ 1 & \text{if } F_X(x) \geq 1 \end{cases} && (18.3) \end{aligned}$$

That is, the cdf of the transformed random variable $F_X^{-1}(U)$ is identical to $F_X(\cdot)$. Therefore is we can:

1. Invert the cdf of X, and
2. generate (pseudo) random uniforms

we can draw from the distribution of X as follows:

$$X = F_X^{-1}(U) \tag{18.4}$$

18.1.2 Example

Consider an exponential random variable X with cdf:

$$F_X(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 - \exp(-x) & \text{if } x > 0 \end{cases} \tag{18.5}$$

For $x > 0$, we can invert the cdf as follows:

$$\begin{aligned} u &= F_X(x) && (18.6)\\ &= 1 - \exp(-x) && (18.7)\\ \exp(-x) &= 1 - u && (18.8)\\ x &= -\log(1-u) && (18.9) \end{aligned}$$

Therefore $X = -\log(1-U)$ has the desired distribution. To implement this in Stata:

```
clear all
set seed 42
set obs 30 // number of random values to be generated
generate U = uniform() // draw uniforms
generate X = -log(1-U) // inversion

// plot the empirical cdf against the target
generate cdf = 1-exp(-X) // target
cumul X, generate(cX) // empirical cdf
label variable cX "simulated"
sort X
twoway (line cdf cX X)
graph export inversion.png, replace
```

which generates Figure 18.1. Here I have deliberately shown an example with a small simulation size (30 observations). We should expect *any* random sample to exhibit deviations from the true cdf because it is ... well, random!

Exercises

Exercise 18.1.
Generate the equivalent of Figure 18.1 for the following:

1. The normal distribution – use Stata's inverse normal function `invnormal()`.
2. A special case of the Beta distribution:

$$F_X(x) = \begin{cases} x^\alpha & \text{if } 0 < x < 1 \\ 0 & \text{if } x \leq 0 \\ 1 & \text{if } x \geq 1 \end{cases}$$

   Do this for $\alpha = 0.5$

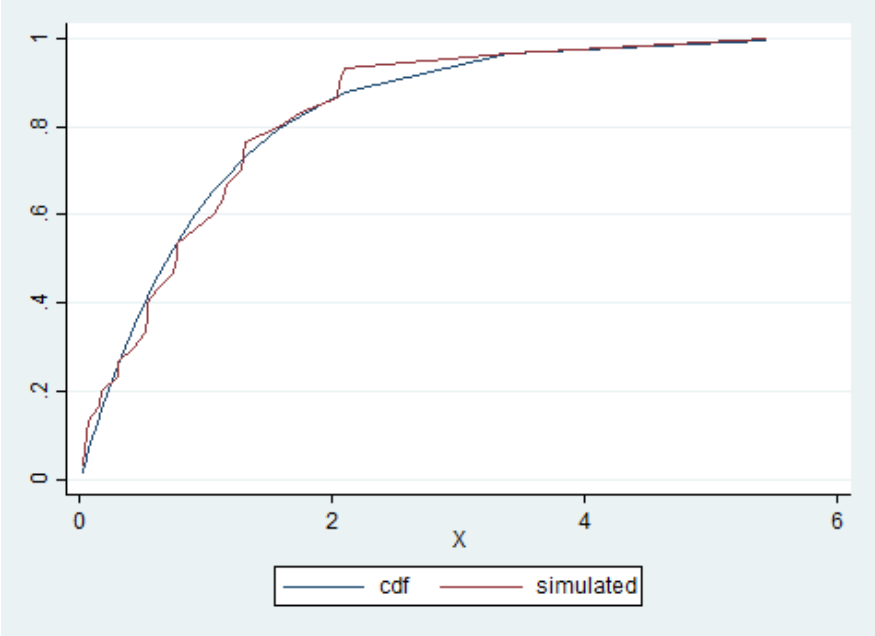


Figure 18.1: Simulation of 30 random numbers with cdf $F_X(\cdot)$ defined in Equation 18.5

3. The triangular distribution:

$$f_X(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \frac{2x}{c} & \text{if } 0 < x \leq c \\ \frac{2(1-x)}{c} & \text{if } c < x \leq 1 \\ 0 & \text{if } x > 1 \end{cases}$$

   Do this for $c = 0.2$ and $c = 0.5$. Note that you will have to integrate the cdf to get the pdf.

Exercise 18.2.
For Exercise 18.1 question 2):

1. Use your simulation to approximate $E[X]$ and $V[X]$

2. Provide an estimate of the accuracy of these approximations. *Hint:* Did you use a sample mean? What do we know about the asymptotic properties of sample means?

Chapter 19

Using pseudo random numbers to calculate things

While Chapter 18 introduces methods for drawing pseudo random numbers that conform to a particular distribution. This chapter is about using them to calculate things. Again, the theory in this chapter draws heavily from Chapter 8 of:

Judd, K. L. (1998). *Numerical methods in economics.* MIT press.

19.1 Monte Carlo Integration

19.1.1 Expectations of random variables

If we can draw pseudo random numbers from the distribution of X, it is straightforward to use these to calculate the expectation of X. In particular, given a sample $\{x_{s=1}\}^S$ of S simulated draws, we can approximate $E[X]$ as follows:

$$E[X] \approx \frac{1}{S}\sum_{s=1}^{S} x_s \tag{19.1}$$

that is, we simply compute the sample mean of our random numbers. If we can further establish that each x_s is *independent*,[1], then we can use a Lindeberg-Levy Central Limit Theorem argument:

$$\sqrt{S}\left(\frac{1}{S}\sum_{s=1}^{S} x_s - E[X]\right) \xrightarrow{d} N(0, V[X]) \tag{19.2}$$

to assign a degree of accuracy to our approximation. Therefore the precision of our approximation is proportional to $\sqrt{S}$. This tells us that as we increase S, the simulation size, we get

[1]This *is* the case if, for example, we draw X using the method of inversion (see Section 18.1), and the uniform draws we use for this are independent. On the other hand, if we use Markov chain techniques, then the draws are typically *not* independent.

closer and close to the true value of $E[X]$. Unlike collecting real data, with simulation this is not so much of a thought experiment: increasing S is relatively cheap, and one is really only limited by the available RAM ($\{x_s\}_{s=1}^S$ must be held in your computer's memory),[2] and the processor speed.

19.1.2 Expectations of functions of random variables

Equation 19.2 is useful because it tells us how large of a simulation we need to achieve a desired level of accuracy. The trouble is that if we need a simulation to evaluate $E[X]$, we probably also need a simulation to evaluate $V[X]$. This is not actually too much trouble, because variance is also an expectation. That is:

$$V[X] \approx \frac{1}{S}\sum_{s=1}^{S}(x_s - E[X])^2 \tag{19.3}$$

where we can substitute in our approximation for $E[X]$. Alternatively, we can use:

$$V[X] = E[X^2] - E[X]^2 \approx \frac{1}{S}\sum_{s=1}^{S}x_s^2 - \left(\frac{1}{S}\sum_{s=1}^{S}x_s\right)^2 \tag{19.4}$$

Approximating $V(X)$ is a special case of approximating $E[g(X)]$.[3] If we can draw from X, then this is a simple extension:

$$E[g(X)] \approx \frac{1}{S}\sum_{s=1}^{S}g(x_s) \tag{19.5}$$

19.1.3 Expectations when you can't draw directly from X

For whatever reason, it might be that we can't find an appropriate way to draw pseudo random numbers from X directly. When X is a continuous random variable, this does not have to be a deal-breaker. To see this, note that for continuous X with pod $f(\cdot)$:

$$E[g(X)] = \int_{\mathbb{R}} g(x)f(x)\mathrm{d}x \tag{19.6}$$

$$= \int_{\mathbb{R}} \frac{g(x)f(x)}{h(x)}h(x)\mathrm{d}x \tag{19.7}$$

Where (19.7) makes the much celebrated algebra monkey trick of multiplying by a fancy one. In this case

$$1_{\text{fancy}} = \frac{h(x)}{h(x)}$$

[2]Even this is not so much of an issue: if a very large S is required, the simulation can be split up into blocks of smaller simulations. Form this, you will get a sample mean for each block, and it is simply a matter of taking the mean of the blocks' sample means.

[3]Here $g(x) = X^2 - E[X]$

Note here that we have implicitly assumed that $h(x) \neq 0$ for all x in the support of X. The implication of (19.7) is that if we can draw from *any* pdf $h(\cdot)$, whose support has the same support as X, then we can approximate $E[g(X)]$ as follows:

1. Draw a simulated sample $\{y_s\}_{s=1}^S$ from the pdf $h(\cdot)$.
2. Generate the transformed sample according to $z_s = g(y_s)f(y_s)/h(y_s)$
3. Compute the sample mean of z_s:

$$E[g(x)] \approx \frac{1}{S}\sum_{s=1}^{S} z_s$$

19.1.4 Example

We wish to evaluate $E[|X|]$, the expected absolute value of X, where X conforms to the logistic distribution with location and scale parameters both equal to one. That is, the pdf of X is:

$$f(x) = \frac{\exp(-(x-1))}{(1+\exp(-(x-1)))^2} \tag{19.8}$$

but we are unable to draw directly from this distribution (perhaps because (i) we have forgotten about inversion, and (ii) that you can read up on it in Section 18.1). To get around this, we use normal draws. In particular, we draw from $Z \sim N(1,1)$. The reason for this is that the logistic distribution looks a lot like the normal distribution, as long as the means and scale are similar. By drawing from $N(1,1)$ instead of the standard normal, we make $f(z)/\phi(z) \approx 1$ for most draws of z. The procedure is therefore:

$$E[|X|]\int |x|f(x)\mathrm{d}x = \int \frac{|x|f(x)}{\phi(x)}\phi(x)\mathrm{d}x \tag{19.9}$$

So the procedure is:

1. Generate $\{z_s\}_{s=1}^S$, a sample of independent normals with $\mu = \sigma^2 = 1$
2. Generate the transformed variable $y_s = \frac{|z|f(z)}{\phi(z)}$
3. $E[|X|] \approx \frac{1}{S}\sum_{s=1}^{S} y_s$

The following code implements this in Stata:

```
clear all
set seed 42
set obs 100 // number of random values to be generated

// Step 1
generate Z = rnormal()+1
```

```
// Step 2: do this in stages
generate absZ = abs(Z)
generate fZ   = exp(-(Z-1))/(1+exp(-(Z-1)))^2
generate phiZ = normalden(Z-1)
generate Y    = absZ*fZ/phiZ

// Step 3
summarize Y
// and look at the mean

// Since we can use inversion, let's try this too
generate U = runiform()
generate X = 1+log(U/(1-U)) // I looked this up on Wikipedia

generate absX = abs(X)

summarize Y absX
```

which generates the output:

```
   Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          Y |       100    1.601705    3.358623   .0113715   25.86216
       absX |       100    1.593992    1.301472   .0280923   5.829037
```

where `Y` was computed using the above method, and `absX` was computed using inversion.

Exercises

Exercise 19.1 (Solution provided)**.**
Let $X \sim \text{Beta}(3, 7)$. Use Equation 19.7 and uniform random numbers to compute $E[(X - 0.5)^2]$.

Exercise 19.2 (Solution provided)**.**
You are an expected utility maximizer with Constant Relative Risk Aversion utility function over money:

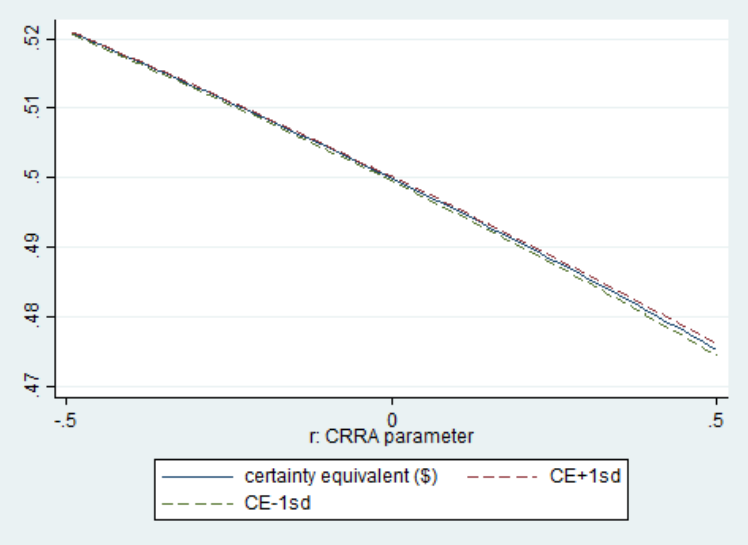
$$u(x) = \frac{x^{1-r}}{1-r}$$

where $r = -u''(x)/u'(x)$ is your coefficient of relative risk aversion. You are offered a lotter that pays out a random amount $\$X$, where X is drawn from the triangular pdf:

$$f_X(x) = \begin{cases} 4x & \text{if } 0 \leq x < 0.5 \\ 4(1-x) & \text{if } 0.5 \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

Calculate the certainty equivalent of this lottery, as a function of r, for $r \in (-0.5, 0.5)$. Summarize your answer in a plot. You have forgotten how to draw from X directly, so use draws from the uniform distribution to perform these calculations.

Your final plot should look something like this:



Note that I have also coded up a measure of the accuracy of this approximation.

Hint: You need a rather large simulation size to get an accurate approximation of the expected utility. One way to check how accurate you are is to check the accuracy of your certainty equivalent for $r = 0$. That is, if you are risk neutral, it is easy to work out what the certainty equivalent is. Once you are happy with how well your simulation approximates this, move on to coding up a `for` loop to generate the plot.

Exercise 19.3.
The game of Yatzee is a die-rolling game with a similar scoring system to Poker. In each round, a player rolls five six-sided dice (at most) three times. After the first and second rolls, the player can choose to only roll a subset of the dice. Their score for the round is a function of the numbers facing upward after the third roll. The highest possible score, called a "Yatzee" occurs when all five dice have the same number (i.e. five ones, five twos, etc.).

Consider the following strategy:

1st roll: Roll all 5 dice (there really isn't any decision to make)

2nd roll: Determine the modal outcome of the previous roll.

- If there is exacty one mode, roll only the dice that do not show this modal number.
- If there is more than one mode, pick the mode with the highest value, and roll only the dice that do not show this modal.
- If there is no mode, roll all 5 dice

3rd roll: Follow the same rule as for the 2nd roll.

Write a simulation to answer the following questions:

1. What is the probability of scoring a Yatzee by following this strategy?

2. What is the distribution of points induced by this decision rule? You can find the list of scoring rules here: `https://en.wikipedia.org/wiki/Yahtzee#Rules`. Assume that the player chooses the highest scoring option from the "upper section" only.[4] Show your answer graphically, and also report the expected score with a measure of the accuracy of this simulated moment.

3. Suppose that instead of following the above decision rule with probability one, you accidentally roll a die that you shouldn't with probability θ. That is, for every die that you shouldn't roll, you accidentally roll it with probability θ, and these accidental rolls are independent of each other. How does this change your answer to question 2.

4. *Ex-ante*, it seems reasonable that rolling all the dice in the second roll might be optimal if the modal outcome is a small number (i.e. 1 or 2). Also, it seems pretty obvious that if there is no modal outcome after the first or second roll, one should hang on to a 6. Write down a modified decision rule to reflect this, and determine whether this tweak results in a better outcome.

5. How much would a risk-neutral[5] player benefit if they were allowed to make 4, 5, or 6 rolls, instead of just 3?

Exercise 19.4.
Suppose you have the constant absolute risk aversion (CARA) utility function:

$$u(x) = -\exp(-ax), \quad a > 0 \tag{19.10}$$

Which means that if you have a choice between different lotteries (i.e. probability distributions over monetary outcomes), say random variables X_1 and X_2, you will choose the distribution that maximizes $E[u(X)]$.

Suppose that you are exposed to a risky asset X, which means that your wealth will be equal to $\$X + 100$ when you sell the asset. The distribution of X can be described by the Laplace distribution, which has pdf and cdf:

$$f(x) = \frac{1}{20}\exp\left(-\frac{|x|}{10}\right) \tag{19.11}$$

$$F(x) = \begin{cases} \frac{1}{2}\exp\left(x/10\right) & \text{if } x \leq 0 \\ 1 - \frac{1}{2}\exp\left(-x/10\right) & \text{otherwise} \end{cases} \tag{19.12}$$

Assume that $a = 1$. Approximate the certainty equivalent of this distribution of wealth for $a = 1$. That is, calculate the amount of money $\$w$ such that $u(w) = E[u(X + 100)]$. Make this calculation twice, assuming that:

[4] You could code up the lower section as well, but that would be more menial work, for a very similar (albeit longer) looking program.

[5] I.e. they only care about the mean of the distribution of points.

1. You have access to a Laplace random number generator (In *Stata*, type `help rlaplace`). This requires *Stata* 15+.
2. You can only draw standard normal random numbers
3. You can only draw uniform random numbers

Provide an estimate of the accuracy of your approximation.

Part VI

More advanced probability and statistics

Chapter 20

Exact tests

20.1 Dependence of categorical variables: The Fisher exact test

Agresti (1992)

The Fisher exact test can be used to test for association between categorical variables. In the simplest case, we might have paired observations of two binary variables $\{x_i, y_i\}_{i=1}^N$ and hypothesize that they are independent. That is, $X \perp\!\!\!\perp Y$. In the classic example, Fisher proposes an experiment to test whether is colleague can determine by taste whether milk has been added before or after tea. The null hypothesis is that she cannot, or formally that the probability that she reports "milk first" does not depend on whether the milk was poured before or after the tea [CITATION NEEDED]. The test generalizes to an arbitrary number of categories and an arbitrary number of categorical variables.

20.1.1 Test for independence of two binary variables

In the simple case that we are testing for independence of two binary variables, it is sometimes informative to visualize the data in a contingency table. This is shown in Table 20.1, where $n_{j,k} = \sum_i I(x_i = j)I(y_i = k)$. Intuitively, if $X \perp\!\!\!\perp Y$, we would expect that the empirical and marginal distributions of X and Y should be similar, which would be expressed in the data

	$Y = 0$	$Y = 1$	row total
$X = 0$	$n_{0,0}$	$n_{0,1}$	$n_{0,0} + n_{0,1}$
$X = 1$	$n_{1,0}$	$n_{1,1}$	$n_{1,0} + n_{1,1}$
column total	$n_{0,0} + n_{1,0}$	$n_{0,1} + n_{1,1}$	N

Table 20.1: A contingency table showing the joint frequencies of two binary variables

as:

$$\hat{p}(X = 0 \mid Y = 0) = \frac{n_{0,0}}{n_{0,0} + n_{1,0}} \approx \frac{n_{0,0} + n_{0,1}}{N} = \hat{p}(X = 0) \tag{20.1}$$

$$\hat{p}(Y = 0 \mid X = 0) = \frac{n_{0,0}}{n_{0,0} + n_{0,1}} \approx \frac{n_{0,0} + n_{1,0}}{N} = \hat{p}(Y = 0) \tag{20.2}$$

An "eyeball" hypothesis test therefore could be to compare these empirical frequencies. More formally, we can calculate the probability of observing data as extreme as Table 20.1 assuming that (1) X and Y are independent, and (2) the marginal frequencies are constant.

One-sided test: We wish to calculate the probability of observing data at least as extreme as those actually observed, conditional on the row and column totals. That is, we fix N, $R_1 = n_{1,0} + n_{1,1}$ and $C_1 = n_{0,0} + n_{0,1}$. Without loss of generality we can assume that $\frac{n_{1,0}}{n_{0,0}} \leq \frac{n_{1,1}}{n_{0,1}}$.[1] This inequality tells us that if the empirical frequencies were equal to the population frequencies, then $\Pr(X = 1 \mid Y = 0) \leq \Pr(X = 1 \mid Y = 1)$, with one inequality strict implying the other. If the second inequality is strict, then the null is false. Under the null hypothesis, the probability that $X = 1$ is independent of Y, let this probability be p. The relevant probability mass functions of $n_{1,0}$ and $n_{1,1}$ are therefore:

$$f(n_{1,0} \mid N, C_1, R_1) = \binom{n_{0,0} + n_{1,0}}{n_{1,0}} p^{n_{1,0}} (1 - p)^{N - C_1 - n_{1,0}} \tag{20.3}$$

$$f(n_{1,1} \mid N, C_1, R_1) = \binom{n_{0,1} + n_{1,1}}{n_{1,1}} p^{n_{1,1}} (1 - p)^{C_1 - n_{1,1}} \tag{20.4}$$

for values of $n_{1,0}$ and $n_{1,1}$ in the appropriate support. Under the null hypothesis, $n_{1,}$ and $n_{1,1}$ conditional on the row and column totals are independent, so we can multiply these probabilities together to show that the joint distribution is proportional to:

$$f(n_{1,0}, n_{1,1} \mid N, C_1, R_1) \propto \binom{N - C_1}{n_{1,0}} \binom{C_1}{n_{1,1}} p^{R_1} (1 - p)^{N - R_1} \tag{20.5}$$

$$f(n_{1,1} \mid N, C_1, R_1) \propto \binom{N - C_1}{R_1 - n_{1,1}} \binom{C_1}{n_{1,1}} p^{R_1} (1 - p)^{N - R_1} \tag{20.6}$$

where the second line substitutes in $n_{1,0} = R_1 - n_{1,1}$. Note that following this substitution, the component of $f(n_{1,1} \mid N, C_1, R_1)$ that is proportional to p does not vary with $n_{1,1}$. Therefore any ratio of these probabilities will not be a function of p. It is this step that allows us to determine the p-value exactly without knowing the marginal probability p. For

[1] We can always re-define our variables to make this hold.

the one-sided test, we seek:

$$\Pr[N_{1,1} \geq n_{1,1} \mid N, C_1, R_1] \tag{20.7}$$

$$= \frac{\sum_{m=0}^{n_{1,1}} \binom{N-C_1}{R_1-m}\binom{C_1}{m} p^{R_1}(1-p)^{N-R_1}}{\sum_{m=0}^{R_1} \binom{N-C_1}{R_1-m}\binom{C_1}{m} p^{R_1}(1-p)^{N-R_1}} \tag{20.8}$$

$$= \frac{\sum_{m=0}^{n_{1,1}} \binom{N-C_1}{R_1-m}\binom{C_1}{m}}{\sum_{m=0}^{R_1} \binom{N-C_1}{R_1-m}\binom{C_1}{m}} \tag{20.9}$$

$$= \frac{\sum_{m=0}^{n_{1,1}} \binom{N-C_1}{R_1-m}\binom{C_1}{m}}{\binom{N}{R_1}} \tag{20.10}$$

(20.10) has a reasonably straight-forward interpretation. The denominator is the number of ways that R_1 out of N observations can satisfy $X = 1$. The term inside the summation in the numerator is the number of ways m observations can satisfy $X = Y = 1$ while there still being R_1 observations satisfying $X = 1$ and C_1 observations satisfying $Y = 1$. Therefore the entire denominator is the number of ways that we can have $n_{1,1} \geq m$ without altering the row and/or column totals.

(20.10) also raises a computational issue: each binomial coefficient contains 3 factorials to be computed, and can therefore be very large if N is large. One solution is to compute this summation in logs, then exponentiate the final answer:

$$\text{Define: } \lambda(a,b) \equiv \log\binom{a}{b} = \log\left(\sum_{k=1}^{a} k\right) - \log\left(\sum_{k=1}^{a-b} k\right) - \log\left(\sum_{k=1}^{b} k\right)$$

$$\gamma(m) \equiv \lambda(N - C_1, R_1 - m) + \lambda(C_1, m) - \lambda(N, R_1)$$

$$\implies p = \sum_{m=0}^{n_{1,1}} \exp[\gamma(m)]$$

The example in Section **??** uses this method.

Chapter 21

Order statistics

In most Econometrics courses, as well as applications of Econometrics, we are obsessed with properties of the mean, and maybe if we are rigorous, the variance of a random sample. But this is not the be all and end all of summarizing data. Sometimes in this course, I get you to simulate the distribution of two estimators, one based on the sample mean, and one based (say) the minimum. The former is attractive because we know a lot about sample means. Specifically, due to the weak law of large numbers, central limit theorem, and so on, it is relatively simple to approximate properties of an estimator that is based on a sample mean. But sometimes using something like a sample minimum gets us a better result, maybe because this estimator has a smaller variance, or converges faster. The problem is that we cannot apply the standard Chapter 4 arguments, because all of that was about properties of sample *means*. We are not $\frac{1}{N}\sum_i X_i$-ing something, so we can;t use this material. Fortunately, we can derive some results about *order statistics* (i.e., minimum, maximum, median, quartiles, deciles, etc) that will help us do stuff similar to Chapters 3 and 4.

For the rest of this Chapter, suppose that we have a sample of N iid draws from a distribution with cdf $F_X(x)$, the support of X is a subset of the real number line.

21.1 Sample maximum and minimum

The sample maximum is equal to the highest number in our sample. We can denote this as

$$Y = \max_i \{X_i\} \tag{21.1}$$

Note that this is a random variable with the same support as X: Y is random because X is random, and it has the same support as X because Y is equal to one of the draws from X that we got in our sample $\{X_i\}_{i=1}^N$. In order to completely characterize the distribution of

Y, we need to work out its cdf. Let's denote this as:

$$\begin{aligned}
F_Y(y) &= \Pr[Y \leq y] && (21.2)\\
&= \Pr\left[\text{All } X\text{s} \leq y\right] && (21.3)\\
&= \Pr\left[X_1 \leq y \cap X_2 \leq y \cap \ldots X_N \leq y\right] && (21.4)\\
&= \prod_{i=1}^{N} \Pr\left[X_i \leq y\right], \qquad \text{because we assumed independence} && (21.5)\\
&= \Pr\left[X_i \leq y\right]^N, \qquad \text{because we assumed identical} && (21.6)\\
&= F_X(y)^N && (21.7)
\end{aligned}$$

This is the cdf of X raised to the power of N. If X is a continuous random variable, then we can find the pdf of the maximum by taking the derivative:

$$\begin{aligned}
f_Y(y) &= \frac{\partial F_Y(y)}{\partial y} && (21.8)\\
&= \frac{\partial}{\partial y} F_X(y)^N && (21.9)\\
&= N f_X(y) F_X(y)^{N-1} && (21.10)
\end{aligned}$$

The sample minimum is equal to the lowest number in our sample. We can denote this as

$$Y = \min_i \{X_i\} \tag{21.11}$$

The derivation of the cdf of the minimum is slightly more involved, but begins in the same place:

$$\begin{aligned}
F_Y(y) &= \Pr\left[Y \leq y\right] && (21.12)\\
&= \Pr\left[\text{at least one } X_i \leq y\right] && (21.13)\\
&= 1 - \Pr\left[\text{all } X_i\text{s} > y\right] && (21.14)\\
&= 1 - \Pr\left[X_1 > y \cap X_2 > y \cap \ldots \cap X_N > y\right] && (21.15)\\
&= 1 - \prod_{i=1}^{N} \Pr\left[X_i > y\right], \qquad \text{because we assumed independence} && (21.16)\\
&= 1 - \prod_{i=1}^{N} \left(1 - \Pr\left[X_i \leq y\right]\right) && (21.17)\\
&= 1 - \left(1 - \Pr\left[X_i \leq y\right]\right)^N, \qquad \text{because we assumed identical} && (21.18)\\
&= 1 - \left(1 - F_X(y)\right)^N && (21.19)
\end{aligned}$$

The cdf of some order statistics are shown on Figure 21.1. Note that since the uniform distribution has a finite support, the minimum and maximum will converge in probability to

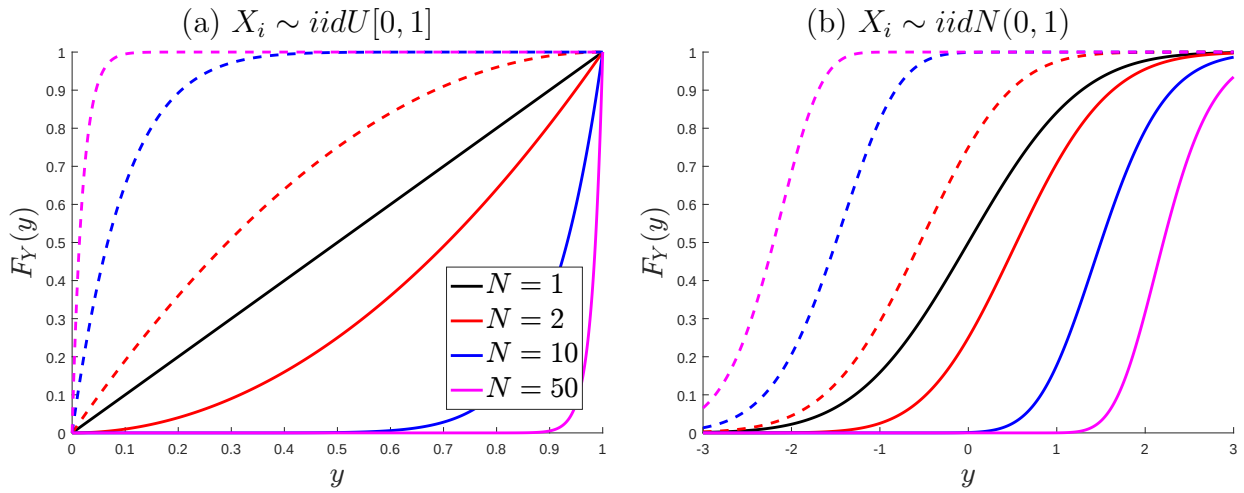


Figure 21.1: Cumulative distribution functions of minimum (dashed lines) and maximum (solid) line oder statistics for various sample sizes. Note that the black line shows the cdfs of the minimum, and maximum for $N = 1$, as well as the cdf for X itself.

the minimum and maximum[1] of the support of X, in this case 0 and 1 respectively. For the normal distribution, the support is the whole real number line: as we increase the sample size, outliers are more likely.

[1]Really the infimum and supremum.

Chapter 22

Further reading

22.1 Reference & Text books

22.1.1 General econometrics and statistics references

- Cunningham, S. (2018). *Causal Inference: The Mixtape*. `http://scunning.com/mixtape.html`
- Wackerly, D., Mendenhall, W., and Scheaffer, R. (2007). *Mathematical statistics with applications*. Nelson Education
- Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT press
- Wooldridge, J. M. (2015). *Introductory econometrics: A modern approach*. Nelson Education
- Bailey, M. (2016). Real econometrics: The right tools to answer important questions
- Koop, G., Poirier, D. J., and Tobias, J. L. (2007). *Bayesian econometric methods*. Cambridge University Press

22.1.2 Specific types of econometrics

- Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press
- Moffatt, P. G. (2015). *Experimetrics: Econometrics for experimental economics*. Palgrave Macmillan

22.1.3 Other

Computational techniques

- Judd, K. L. (1998). *Numerical methods in economics*. MIT press

22.2 Popular press

- Silver, N. (2012). *The signal and the noise: why so many predictions fail–but some don't*. Penguin
- List, J. and Gneezy, U. (2014). *The why axis: hidden motives and the undiscovered economics of everyday life*. Random House
- Pearl, J. and Mackenzie, D. (2018). *The Book of Why: The New Science of Cause and Effect*. Basic Books

Bibliography

Abadie, A., Athey, S., Imbens, G., and Wooldridge, J. (2017). When should you adjust standard errors for clustering? *arXiv preprint arXiv:1710.02926*.

Agresti, A. (1992). A survey of exact inference for contingency tables. *Statistical Science*, 7(1):pp. 131–153.

Ai, C. and Norton, E. C. (2003). Interaction terms in logit and probit models. *Economics Letters*, 80(1):123 – 129.

Bailey, M. (2016). Real econometrics: The right tools to answer important questions.

Berry, S., Levinsohn, J., and Pakes, A. (1995). Automobile prices in market equilibrium. *Econometrica*, 63(4):841–890.

Bland, J. R. and Cook, A. C. (2017). Random effects probit and logit: The right marginal effects for the right econometric specification.

Cameron, A. C. and Miller, D. L. (2015). A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, 50(2):317–372.

Cunningham, S. (2018). *Causal Inference: The Mixtape*. `http://scunning.com/mixtape.html`.

DeLuca, K. (2019). Tweet, @cantstopkevin, 11:14 pm, jul 17, 2019, `https://twitter.com/cantstopkevin/status/1151691761586229249`.

Frölich, M. (2008). Parametric and nonparametric regression in the presence of endogenous control variables. *International Statistical Review*, 76(2):214–227.

Judd, K. L. (1998). *Numerical methods in economics*. MIT press.

Koop, G., Poirier, D. J., and Tobias, J. L. (2007). *Bayesian econometric methods*. Cambridge University Press.

List, J. and Gneezy, U. (2014). *The why axis: hidden motives and the undiscovered economics of everyday life*. Random House.

Makowsky, M. D. and Stratmann, T. (2009). Political economy at any speed: what determines traffic citations? *The American Economic Review*, 99(1):509–527.

Moffatt, P. G. (2015). *Experimetrics: Econometrics for experimental economics*. Palgrave Macmillan.

Pearl, J. and Mackenzie, D. (2018). *The Book of Why: The New Science of Cause and Effect*. Basic Books.

Silver, N. (2012). *The signal and the noise: why so many predictions fail–but some don't*. Penguin.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

Wackerly, D., Mendenhall, W., and Scheaffer, R. (2007). *Mathematical statistics with applications*. Nelson Education.

Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT press.

Wooldridge, J. M. (2015). *Introductory econometrics: A modern approach*. Nelson Education.

Part VII

Appendices

Appendix A

Past exam questions

A.1 ECON5820 Final Exams

A.1.1 Computational exams

Exercise A.1 (2017 5820 Final Computational Exam)**.**
Consider the expectation:

$$E[\Phi(1+X)] = \int_0^\infty \Phi(1+x) f_X(x) \mathrm{d}x \qquad (\star)$$

where $X \sim$ Exponential(1), $f_X(x)$ is the pdf of this distribution, and $\Phi(\cdot)$ is the standard normal cdf.

1. (30 points) Approximate $(\star)$ using any functions and/or random number generators available in *Stata*.
2. (30 points) Approximate $(\star)$ using inversion.
3. (30 points) Approximate $(\star)$ using only χ^2_1 random numbers. You may *not* use the χ^2_1 cdf.
4. (10 points) Approximate $(\star)$ using only standard normal random numbers. You may *not* use the normal cdf to generate uniform random numbers.
5. (10 points) Explain (in a comment) why in question 4 you can't just replace the χ^2_1 parts of your answer to question 3 with their normal equivalents.

The correct answer is about 0.94, but it's all about how you get there.

Hint: `https://www.stata.com/manuals13/dfunctions.pdf`

A.1.2 Written exams

Exercise A.2 (2017 5820 Final Written Exam)**.**
Consider Levitt's IV approach for estimating the effect of police on crime. Specifically, he estimates the following model using 2SLS:

$$\text{crime}_i = \beta_0 + \beta_1 \text{police}_i + \epsilon_i \tag{A.1}$$

using the variable "firefighters$_i$" as an instrumental variable.[1] Remember that his justification for this is that cities that hire more firefighters will probably also hire more police officers, but the variation in firefighters is should be uncorrelated with crime.

1. In the context of this estimation and Equation A.1, explain the *inclusion condition* and how/if one could go about testing that it is satisfied.

2. In the context of this estimation and Equation A.1, explain the *exclusion condition* and how/if one could go about testing that it is satisfied.

3. One could also argue that the number of publicly funded hospitals in a city could also be correlated with the number of police officers. If the crime variable of interest is *violent* crime, explain why this variable may not be a valid instrument for the number of police officers. For this problem, assume that both police and publicly funded hospitals are funded at the local (city) level.

4. Suppose that the federal government wishes to give cities funding to expand their police force by 10%. Unfortunately, there is not enough money to do this for every city, and so the funds must be rationed. The government is considering two rationing mechanisms:

   (a) Selecting the cities with the highest crime rates, or

   (b) Randomly selecting cities (e.g.: for each city, flip a coin. If it comes up heads, the city gets the funding.)

   In the context of the inclusion and exclusion conditions, explain why `selected` is a valid instrument for mechanism (b), but not for mechanism (a).

   [For questions 5-8, assume that we are using rationing mechanism (b)]

   Suppose that the selected cities can opt in to the additional funding (i.e. they do not have to accept it). In our dataset we have the following variables:

[1] I.e.: in STATA: `ivregress 2sls crime (police = fire)`

Variable	Description
$\text{Crime}_{i,t}$	Crime rate
$\text{Police}_{i,t}$	Number of police officers (per 100,000 of city's population)
$\text{Fire}_{i,t}$	Number of fire fighters (per 100,000 of city's population)
$\text{Selected}_{i,t}$	Dummy variable = 1 if the city was selected for extra funding
$\text{Accepted}_{i,t}$	Dummy variable = 1 if the city accepted the funding

5. Write down the *intention to treat* regression equation (either write down the STATA command, or briefly and accurately describe how you would go about estimating it), and discuss how its interpretation is different from:

```
ivregress 2sls Crime (Accepted = Selected)
```

6. We have established that `selected` is a valid instrument for `police`. Therefore, we may want to estimate:

```
ivregress 2sls crime (police = fire selected)
```

   Explain why this model is over-identified, and what information rejecting and failing to reject the null hypothesis of the over-identification test gives us.

7. Suppose that you *knew for sure* that `selected` satisfied the exclusion condition. What does rejecting the null hypothesis in (6) tell you now?

8. Briefly describe a table of descriptive statistics that you may want to look at before being satisfied with your results in (5). Why would you want to do this?

   For the remaining questions: Suppose that instead of using method 4b (randomly selecting cities) outlined above, the federal government provided funding to cities with populations between 300,000 and 500,000. The funding was enough to increase this number by 1 police officer per 10k people. Your dataset now contains the following variables (and nothing else):

Variable	Description
$\text{Crime}_{i,t}$	Crime rate
$\text{Pop}_{i,t}$	Population

   This program had no issues with compliance: all cities with populations within this range received the funding and used it to increase the size of their police force.

9. Write down an econometric specification that allows you to estimate this causal effect of police on crime rates using this dataset. Explain which coefficient(s) can be interpreted as the causal effect, and how to interpret it. Restrict yourself to linear specifications with interactions. Don't worry about windows.

10. Suppose that at some time after the policy was announced, but before it was implemented, some cities had the opportunity to re-draw their boundaries (thus artificially raising or lowering their population). Explain why this may make your analysis in the previous question invalid.

Appendix B

Solutions to selected problems

Solution to Exercise 2.2

1. cdf

$$F + \hat{\gamma} = \Pr[\hat{\gamma} \leq x] = \Pr[X_1, X_2, \ldots, X_N \leq x] \tag{B.1}$$

$$= \prod_{i=1}^{N} \Pr[X_i \leq x] \quad \text{since we have iid data} \tag{B.2}$$

$$= (F_X(x))^N \tag{B.3}$$

$$= \begin{cases} 0 & \text{if } x \leq 0 \\ \left(\frac{x}{\gamma}\right)^N & \text{if } 0 < x < \gamma \\ 1 & \text{if } x \geq 1 \end{cases} \tag{B.4}$$

2. Pdf: This is equal to zero everywhere, except for in the support, where we can take the derivative of the cdf:

$$f_{\hat{\gamma}}(x) = \frac{N x^{N-1}}{\gamma^N} I(0 < x < \gamma) \tag{B.5}$$

3. Mean: We need the variance later, so it is going to be easier in the long run to work

out:

$$E[\hat{\gamma}^k] = \int_0^\gamma x^k f_{\hat{\gamma}}(x)\mathrm{d}x \tag{B.6}$$

$$= \int_0^\gamma x^k \frac{Nx^{N-1}}{\gamma^N}\mathrm{d}x \tag{B.7}$$

$$= \int_0^\gamma \frac{Nx^{k+N-1}}{\gamma^N}\mathrm{d}x \tag{B.8}$$

$$= \frac{Nx^{k+N}}{\gamma^N(k+N)}\Big|_0^\gamma \tag{B.9}$$

$$= \frac{N\gamma^{k+N}}{\gamma^N(k+N)} \tag{B.10}$$

$$= \frac{N}{N+k}\gamma^k \tag{B.11}$$

$$E[\hat{\gamma}] = \frac{N}{N+1}\gamma \tag{B.12}$$

4. Variance:

$$V[\hat{\gamma}] = E[\hat{\gamma}^2] - E[\hat{\gamma}]^2 \tag{B.13}$$

$$= \frac{N}{N+1}\gamma^2 - \left(\frac{N}{N+1}\right)^2 \gamma^2 \tag{B.14}$$

$$= \frac{N\gamma^2}{N+1}\left(1 - \frac{N}{N+1}\right) \tag{B.15}$$

Solution to Exercise 2.3
Bias:

$$E[\hat{\mu}] = E\left[\frac{1}{N}\sum_{i=1}^N X_i\right] \tag{B.16}$$

$$= \frac{1}{N}\sum_{i=1}^N E[X_i] \tag{B.17}$$

$$= \frac{1}{N}\sum_{i=1}^N \mu, \qquad \text{(iid)} \tag{B.18}$$

$$= \mu \implies \text{unbiased} \tag{B.19}$$

$$E[\tilde{\mu}] = E\left[N \min_i\{X_i\}\right] \tag{B.20}$$
$$= NE\left[\min_i\{X_i\}\right] \tag{B.21}$$
$$= N\frac{\mu}{N}, \qquad \text{(given)} \tag{B.22}$$
$$= \mu \implies \text{unbiased} \tag{B.23}$$

Variance:

$$V[\hat{\mu}] = V\left[\frac{1}{N}\sum_{i=1}^{N} X_i\right] \tag{B.24}$$
$$= N\frac{1}{N^2}V[X_i] = \frac{1}{N}V[X_i], \qquad \text{(iid)} \tag{B.25}$$
$$= \frac{1}{N}\left(E[X_i^2] - E[X_i]^2\right) \tag{B.26}$$
$$= \frac{1}{N}\left(2\mu^2 - \mu^2\right) \tag{B.27}$$
$$= \frac{\mu^2}{N} \tag{B.28}$$
$$V[\tilde{\mu}] = V\left[N \min_i\{X_i\}\right] \tag{B.29}$$
$$= N^2 V[\min_i\{X_i\}] \tag{B.30}$$
$$= N^2 V[\text{Exponential}(\mu/N)] \tag{B.31}$$
$$= N^2\frac{\mu^2}{N^2} \tag{B.32}$$
$$= \mu^2 \tag{B.33}$$

MSE: Since both are unbiased, $MSE[\hat{\mu}] = V[\hat{\mu}]$ and $MSE[\tilde{\mu}] = V[\tilde{\mu}]$.

By these measures, $\hat{\mu}$ is unambiguously better than $\tilde{\mu}$: both are unbiased, but $\hat{\mu}$ has smaller variance.

Simulation:

Let's start by working out $\check{\mu}$. This is motivated from:

$$E[X^2] = 2\mu^2 \tag{B.34}$$
$$\check{\mu} = \sqrt{\frac{1}{2N}\sum_{i=1}^{N} X_i^2} \tag{B.35}$$

So in terms of the code, once we have simulated $\{X_i\}_{i=1}^N$, we need to generate X_i^2, take its mean, then divide by 2 and take te square root.

The code below produces the following output table:

```
. summarize
```

```
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       muHat |     10,000    .9983377    .1829214    .4800103   1.714862
     muTilde |     10,000    .9992013    .9903682    .0001359   10.60011
        muC |     10,000    .9791806    .1963733    .4536264   1.972492
```

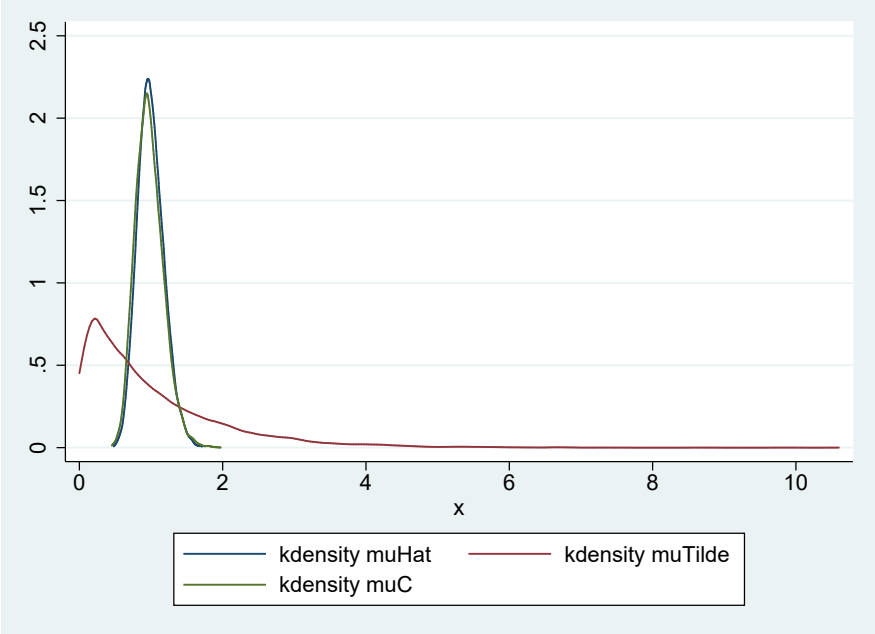
The mean values are close enough to 1 to confirm that our calculations in part 1 are correct. $\check{\mu}$ appears to be biased downward. For the standard deviation for the simulated sampling distribution, note that:

$$\sqrt{V[\hat{\mu}]} = \frac{1}{\sqrt{30}} \approx 0.18 \tag{B.36}$$

$$\sqrt{V[\hat{\mu}]} = 1 \tag{B.37}$$

Good!

The simulated sampling distributions are shown in the plot below. Again, we are much better off using the sample mean than the minimum. The sampling distribution of estimator $\check{\mu}$ is almost indistinguishable from that of $\hat{\mu}$, so it does better than $\tilde{\mu}$, but we might want to explore its properties further if we want to use it (as a general rule, estimators that are sample means are hard to beat).




```
// Clear everything from memory
clear
clear all
/* First we write a program that generates a sample conforming to N=30 and mu=1,
then applies the two estimators
*/
program define ExponentialSim, rclass
```

```
                // Here our inputs are the number of observation, and mu, the parameter we
                    ↪ want to estimate
        syntax [, obs(integer 30) mu(real 1) ]
                // Set the sample size
        set obs `obs'
                // declare some temporary variables that we will use later
        tempvar x x2
                // Generate x according to the specified distribution
        generate `x' = -`mu'*log(runiform())
                // Summarize x. We need to calcualte its mean and the minimum
        summarize `x'
                // estimators
        return scalar muHat = r(mean)
                return scalar muTilde = `obs'*r(min)
                // muC, based on 2nd raw moment
                generate `x2' = `x'^2
                summarize `x2'
                return scalar muC = sqrt(r(mean)/2)
end

// check that the program is running properly
ExponentialSim, obs(30) mu(1)
disp "`r(muHat)'"
disp "`r(muTilde)'"


// run the simulation
set seed 42
quietly simulate muHat=r(muHat) muTilde=r(muTilde) muC=r(muC), reps(10000): ExponentialSim,
    ↪ obs(30) mu(1)

summarize

twoway (kdensity muHat) (kdensity muTilde) (kdensity muC)
graph export ExExponentialSim.pdf, replace
```

Solution to Exercise 14.1

See Figure B.1.

```
clear all
set more off
set seed 42
import delimited TempToledoAirport.csv
desc
generate t = date(date,"YMD",1901)
tsset t
//tsline tmax

generate R = runiform()
sort R

generate estimation = 1
replace estimation = 0 if _n <= 0.3*_N
tab estimation


generate AllData = .
generate ModelSelection = .
generate lags = .
local RHS ""
sort t

generate month_temp = substr(date, 6,2)
tab month_temp
```

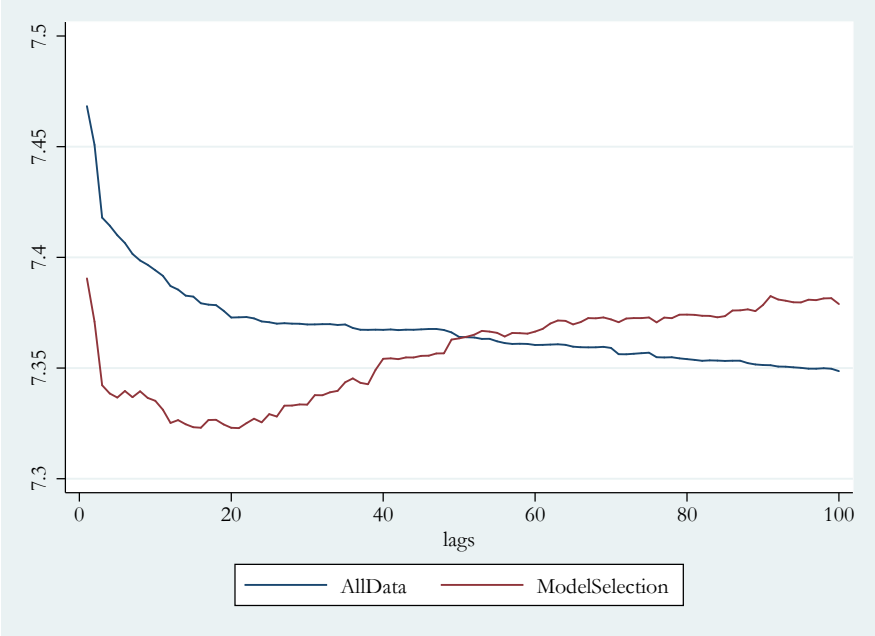


Figure B.1: Exercise 14.1: Root mean squared forecast error for the daily maximum temperature at Toledo Airport, as a function of the number of lags included in the model. It appears that the best model under consideration uses about 15 lags.

```
encode month_temp, generate(month)

forvalues ll = 0/100 {
        if `ll'>0 {
                local RHS "`RHS' L`ll'.tmax"
        }
        quietly regress tmax i.month `RHS'
        quietly replace lags = `ll' if _n==`ll'
        quietly predict tHat, xb
        quietly generate tError2 = (tmax-tHat)^2
        quietly summarize tError2
        quietly replace AllData = sqrt(`r(mean)') if _n==`ll'

        drop tHat tError2

        quietly regress tmax i.month `RHS' if estimation ==1
        if `ll' ==15 {
                estimates store modelSelected
                predict tempHat, xb
        }
        quietly predict tHat, xb
        quietly generate tError2 = (tmax-tHat)^2 if estimation ==0
        quietly summarize tError2
        quietly replace ModelSelection = sqrt(`r(mean)') if _n==`ll'

        drop tHat tError2
        display `ll'
}

twoway (line AllData ModelSelection lags)
graph export ToledoWeatherForecast1.png, replace
```

```
estimates replay modelSelected

generate tError2 = (tmax-tempHat)^2
regress tError2 i.month if estimation ==0
```

Solution to Exercise 19.1

```
clear all
set seed 42
set obs 100
generate U = runiform()
generate gU = (U-0.5)^2
generate hU = 1
generate fU = betaden(3,7,U)
generate Y = gU*fU/hU
estpost summarize Y
esttab using MCIntegrationEx1.tex, replace cells("count mean sd min max") noobs
```

Which generates the following table:

(1)

	count	mean	sd	min	max
Y	100	.0547719	.0800443	3.93e-13	.2622285

So we conclude that $E[(X-0.5)^2] \approx 0.055$ **Solution to Exercise 19.2**

```
clear all
set seed 42
set obs 1000000
generate U = runiform()

generate R = .
generate EU = .
generate Y = .
generate Yp1sd = .
generate Ym1sd = .


forvalues rr = 1/100 {
        local r = -0.5 + `rr'/100*(0.5-(-0.5))

        generate uU = 1/(1-`r')*(U)^(1-`r')             // function to evaluate

        generate fU = 4*U                                                   // pdf of X for 0<x
            ↪ <0.5
        quietly replace fU = 4*(1-U) if U>0.5   // pdf of X for 0.5<x<1
        generate hU = 1                                                     // pdf of U

        generate T = uU*fU/hU
        quietly summarize T
        local EU = r(mean)
        local std = sqrt(r(Var)/_N)

        /* Certainty equivalent calculation
        we are trying to solve:
                u(y) = EU
        after substituting in u(.) on the LHS:
```

```
                y^(1-r)/(1-r) = EU
                y^(1-r) = (1-r)*EU
                y = [(1-r)*EU]^(1/(1-r))
        */

        quietly replace EU = `EU' if `rr'==_n

        local y =  ((1-`r')*`EU')^(1/(1-`r'))
        display `r' `y'

        quietly replace R = `r' if `rr'==_n
        quietly replace Y = `y' if `rr'==_n
        quietly replace Yp1sd = `y'+`std' if `rr'==_n
        quietly replace Ym1sd = `y'-`std' if `rr'==_n
        drop uU fU hU T
}

label variable R "r: CRRA parameter"
label variable Y "certainty equivalent ($)"
label variable Yp1sd "CE+1sd"
label variable Ym1sd "CE-1sd"
twoway (line Y R) (line Yp1sd R, lpattern(dash)) (line Ym1sd R, lpattern(dash))
graph export MCIntergrationEx2.png, replace
```