BUSN 5000

Beginning to Learn

Chris Cornwell

Terry College of Business

Fall 2023

Section 1

Preliminaries

Week 2 :: Reading and listening

Readings: Bekes and Kedzi, ch 2-3

IRL: "What Can Uber Teach Us About the Gender Pay Gap?", *Freakonomics Radio*, Episode 317.

R highlights

Packages

- `dplyr` part of the `tidyverse` collection of packages that provides functions for data manipulation (e.g. `mutate`, `group_by` and `summarise`)

Functions

- `read_xlsx` for reading .xlsx files
- `stat_function` for plotting statistics functions
- `plnorm` for computing cumulative probabilities using a lognormal distribution
- `stat_ecdf` for computing the empirical CDF
- `mutate` for creating new variables
- `group_by` for performing operations by group
- `summarise` creates data frame with a single row summarizing all observations in the data set or group

The main things

1. Frequentism
2. Law of large numbers
3. Estimands, estimators and estimates
4. Sources of bias
5. Conditional expectations function (CEF)
6. Law of iterated expectations (LIE)

Average hourly earnings of Uber drivers, men vs women

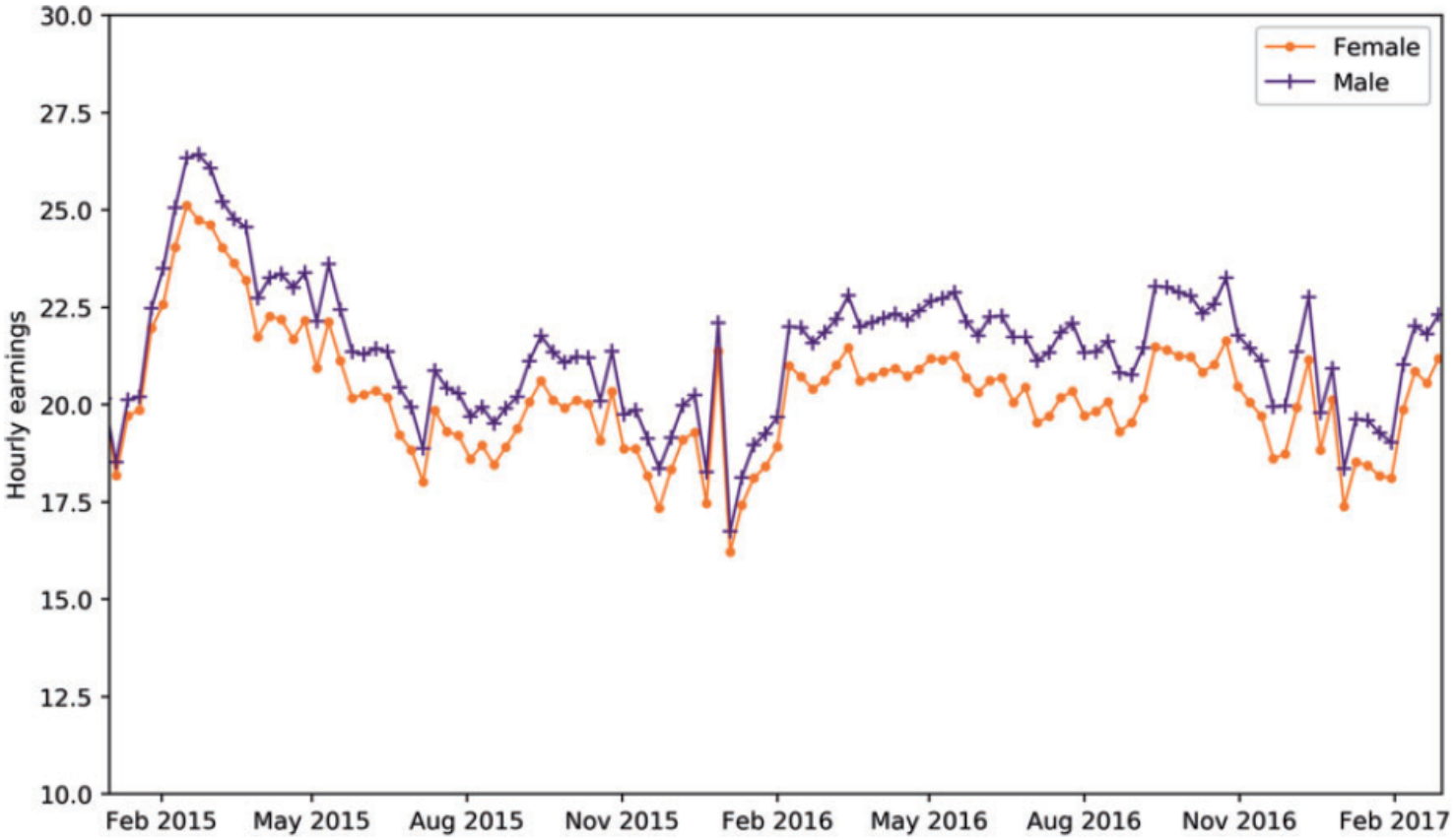


FIGURE 1

Average hourly earnings, U.S.

Notes: Data based on hourly earnings averaged across all UberX and UberPOOL drivers who worked in a given week. The percent of drivers who are female varies across city; to mitigate composition effects, we weight averages at the city level by *total* number of drivers in a city, rather than by number of male (or female) drivers. Earnings are gross; costs such as the Uber commission or gas are not subtracted.

Source: Cook et al., REStud (2021)

Section 2

The Distribution of Earnings

What we want to learn about

We want to learn how earnings are distributed among workers in the population.

What do you think the overall earnings distribution looks like? Does it look different for women and men?

What is the likelihood of earning at least $100,000? No more than $40,000? How do these likelihoods vary by gender?

Is there a gender pay gap, and if so, how big is it?

March 2009 CPS extract

We are going to try to learn about the distribution of earnings using a 2009 March CPS extract acquired from Bruce Hansen (The link takes you to a landing page where you can access this extract and documents related thereto.).

The extract contains individuals who were full-time employed (defined as those who had worked at least 36 hours per week for at least 48 weeks the past year) and excluded those in the military. This sample has 50,742 individuals. For each individual the extract contains data on 12 variables: `age`, `female`, `hisp`, `education`, `earnings`, `hours`, `week`, `union`, `uncov`, `region`, `race`, and `marital`.

Full ASEC supplement document

Visualizing the earnings distribution

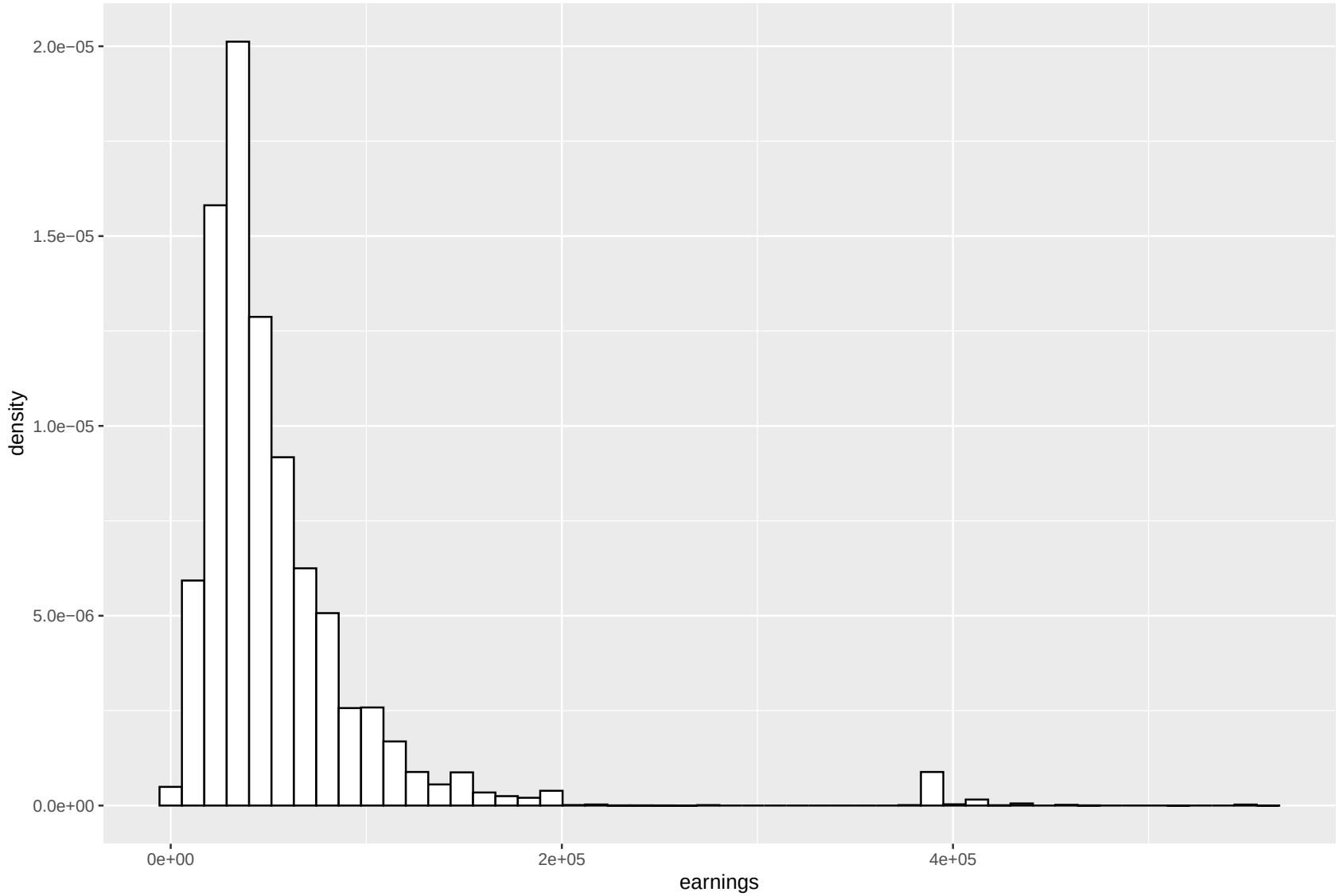
There is no ready-made `earnings` variable in the CPS. What we are using is a combination of several variables depending on, among other things, whether the worker was salaried or paid hourly.

Let's create a histogram of the constructed `earnings` variable.

```
cps_mar <- read_xlsx("data/cps09mar.xlsx")
earnings_dist <- ggplot(cps_mar, aes(x=earnings)) +
  geom_histogram(aes(y = ..density..), color=1,
                 fill="white", bins=50) +
  labs(title="Figure 1. Distribution of earnings, March 2009 CPS")
```

The distribution of earnings

Figure 1. Distribution of earnings, March 2009 CPS



What do you see?

Well, it's definitely *skewed right*, i.e. it is has long right tail. This is a feature of most all earnings distributions. What if we could find a *model* to capture this general shape?

Also, you can see set of observations clustered around $400,000. This is the result of *topcoding*, which is done to protect high-earning respondents' privacy. For the details on the 2009 top codes, consult section 5 of the 2009 ASEC Supplement. We will have more to say about this later.

One implication of the long right tail is that average earnings will be larger than median earnings.

```
mean(cps_mar$earnings)
```

```
## [1] 55091.53
```

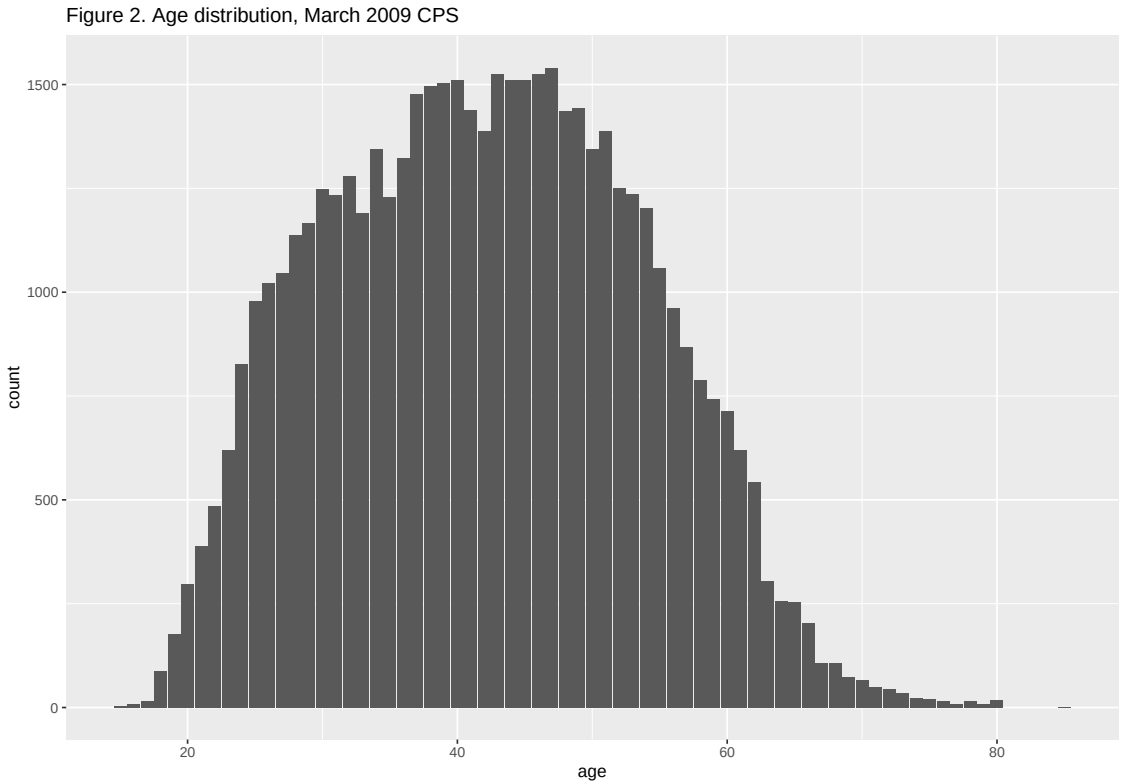
```
median(cps_mar$earnings)
```

```
## [1] 42000
```

Back to what we want to learn about

Our first take on the earnings distribution has included everyone in the CPS with earnings data, from the very young to the very old. Is this the population we want to learn about? We won't leave anybody out for now, but we probably need to think about this.

```
ggplot(cps_mar, aes(age)) +  geom_bar() +
  labs(title="Figure 2. Age distribution, March 2009 CPS")
```



Section 3

Learning about the DGP

Models for inference

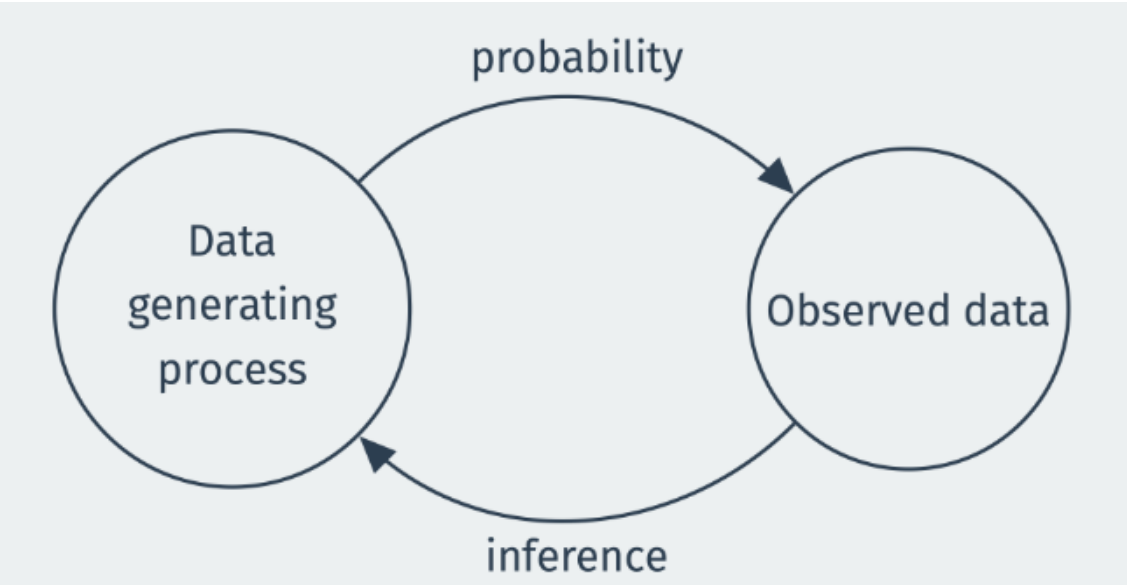
Much of the sort of learning we want to accomplish involves *random sampling* from the *population* of interest. This is what we are doing with the CPS. The goal is to use the data from CPS respondents to learn something about the *data-generating process (DGP)* for earnings.

We take earnings to be a *random variable* and the CPS sample an opportunity to *infer* features of its distribution and likelihoods of certain outcomes. Because the sample is *not* the population, this is hard.

If you need a refresher on random variables and random sampling, go here. The key ideas are *uncertainty* and that one draw from the population does not depend on another.

The opposite of probability theory

Inference is the opposite of probability theory. After all, if we know a random variable has a particular distribution, we can directly calculate the likelihood of a specific outcome. In the real world, though, we never know the distributions of random variables, so we are left with statistical inference.



Source: Matt Blackwell

Frequentism

We cannot make progress toward statistical inference without some understanding of basic probability concepts. Given our focus on learning from random sampling, we will take the *frequentist* approach, thinking about probability as *relative frequency*. Formally, we can state it like this:

Definition. Let A be some event and $S(N)$ the number of occurrences in N random trials. Then, the probability of event A is

$$P(A) = \lim_{N \to \infty} \frac{S(N)}{N}.$$

Relative frequency in the limit

If the formal definition is not your thing, let's try simulating the idea. Say the event A is "Heads" in coin flip, so that S_N is the number of heads you get from N flips of a fair coin. We'll simulate 1000 random trials (flips) of a fair ($P(Heads) = p = .5$) coin, and plot the path of the cumulative proportion of heads.

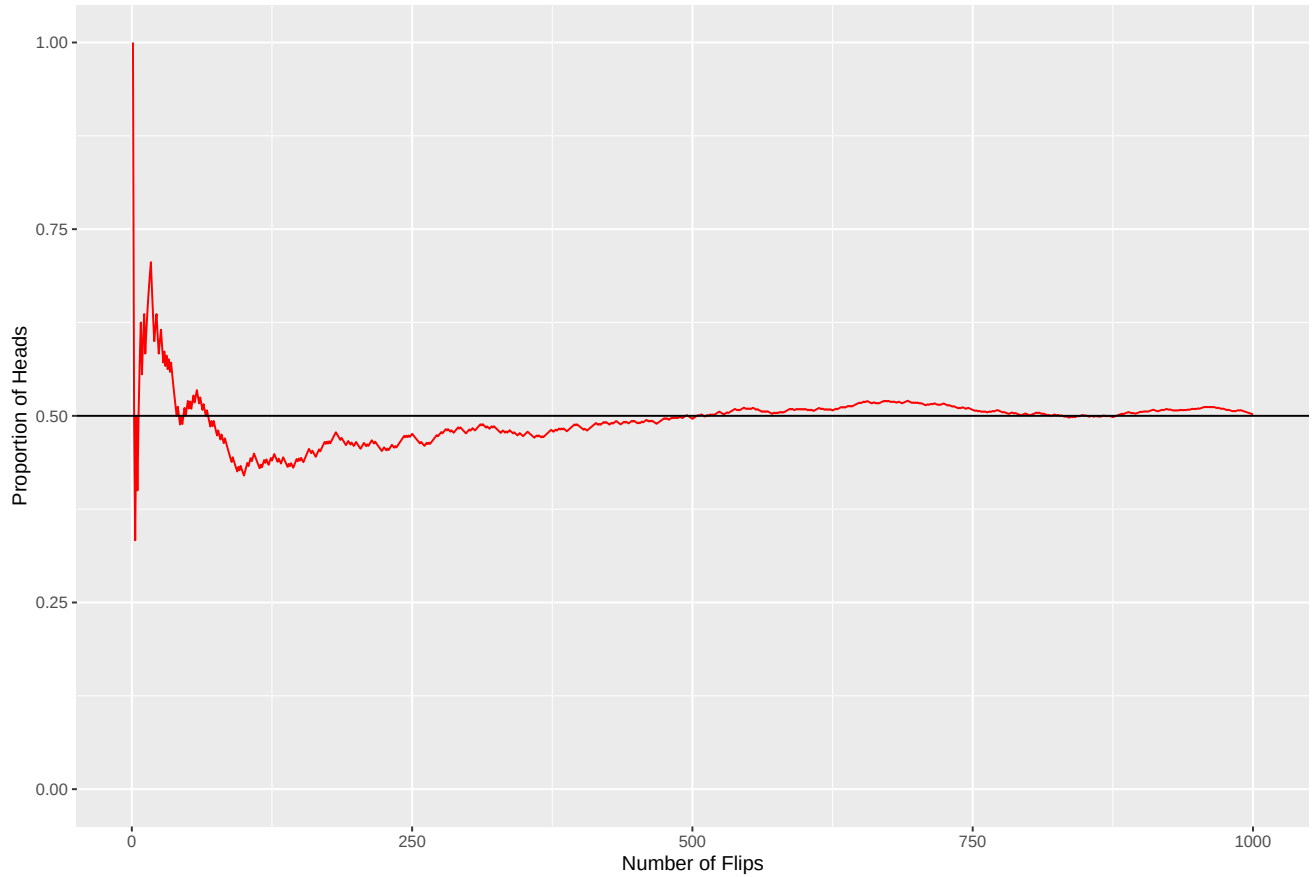
```
# set.seed(12345)
N = 1000
p = .5
flips  = sample(x=c(0,1), prob = c(1-p,p), size=N, replace=T)
S_N    = cumsum(flips)
N      = 1:N
prop_S = S_N/N

flip_data <- data.frame(run=1:1000, prop=prop_S)
flip_plot <- ggplot(flip_data, aes(x=run, y=prop, frame=run)) +
  geom_path(aes(cumulative=T), color="red") +
  xlim(1,1000) + ylim(0.0,1.0) +
  geom_hline(yintercept = 0.5) +
  labs(title="Figure 1. Flipping a fair coin") +
  ylab("Proportion of Heads") +
  xlab("Number of Flips")
```

Law of large numbers

What is formally stated in the definition and demonstrated in the simulation is an example of the *law of large numbers*, one of the most important results in statistics. We will rely on this idea over and over again.

Figure 1. Flipping a fair coin



ChatGPT and grammar of graphics ftw

ChatGPT

https://chat.openai.com/

Hadley Wickham, *Journal of Computational and Graphical Statistics* (2010)

https://www.tandfonline.com/doi/pdf/10.1198/jcgs.2009.07098

Earning six figures

What is the likelihood of earning $100,000? How can we use the CPS to infer the answer? One way is to *estimate* the likelihood by the share of respondents with incomes greater than or equal to $100,000.

```
six_figs <- subset(cps_mar, earnings >= 100000)
six_figs_shr <- nrow(six_figs)/nrow(cps_mar)
print(six_figs_shr)
```

```
## [1] 0.1002523
```

So, roughly 10% if we believe the CPS.

Modeling the earnings distribution

Because we don't know the distributions of the variables in our analyses, we infer from descriptive measures like histograms constructed from the samples we collect.

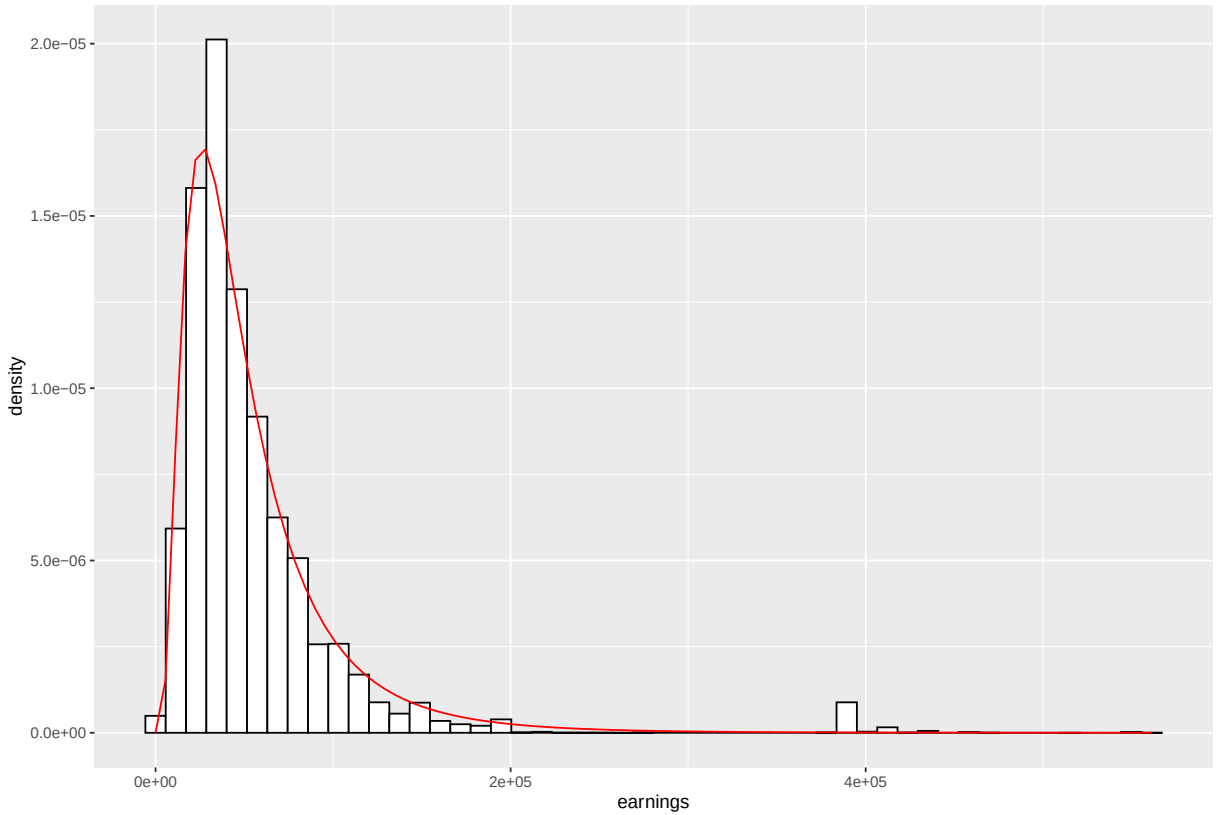
Sometimes the data appear to be well represented by a simple probability model. It turns out that earnings data can be reasonably fit with the lognormal distribution.

```
earnings_lnorm <- ggplot(cps_mar, aes(earnings)) +
  geom_histogram(aes(y = after_stat(density)), color=1,
                 fill="white", bins=50) +
  stat_function(fun = dlnorm,
                args = list(meanlog = mean(log(cps_mar$earnings)),
                            sdlog = sd(log(cps_mar$earnings))),
                colour = "red") +
  labs(title="Figure 2. Wage histogram with lognormal overlay")
```

Lognormal model of the earnings distribution

Figure 2 repeats the histogram of earnings and adds an overlay of a lognormal distribution using the sample mean and standard deviation The model looks pretty good, don't you think?

Figure 2. Wage histogram with lognormal overlay



We can use the lognormal model to estimate the likelihood of earning six figures using only the mean and standard deviation of earnings.

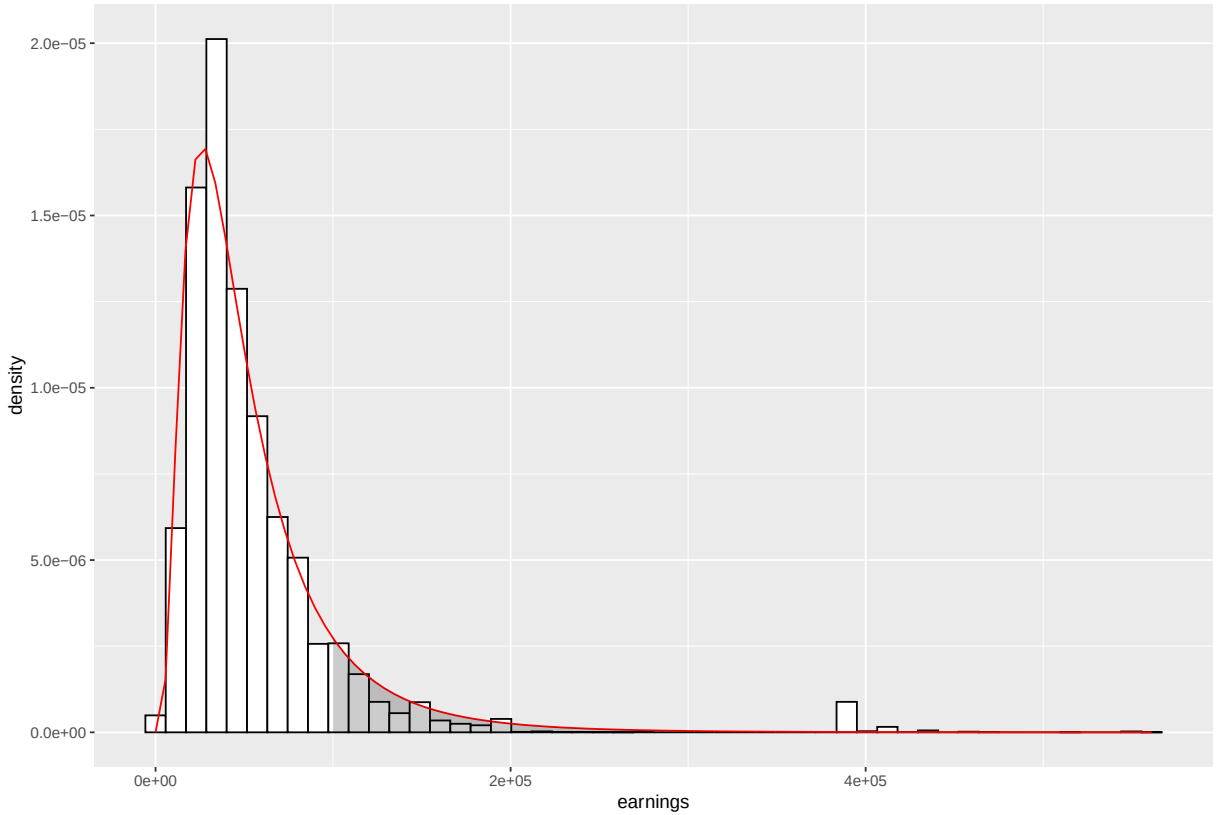
Earning six figures again

The likelihood of earning six figures based on the lognormal model is about 11% and is indicated by the area under the curve above $100,000.

```
1-plnorm(100000, meanlog =
    mean(log(cps_mar$earnings)), sdlog = sd(log(cps_mar$earnings)))
```

```
## [1] 0.1127618
```

Figure 2. Wage histogram with lognormal overlay



Estimands, estimators and estimates

Formally, what have we done?

We defined the *estimand* – the thing we want to learn about – as

$$P(\textit{earnings} \geq 100,000).$$

We used our CPS data to compute two *estimators* – (1) the share of six-figure earners and (2) the probability as implied by the lognormal distribution – and got two *estimates*:

$$(1) \quad \frac{\sum_{i=1}^{N} \mathbf{1}(\textit{earnings}_i \geq 100000)}{N} = .100$$

$$(2) \quad 1 - \int_{-\infty}^{100000} \frac{1}{y\sigma\sqrt{2\pi}} e^{-\frac{(\ln y - \mu)^2}{2\sigma^2}} dy = .112$$

Evaluating estimates

Are these estimates any good? Is one better than the other? How would we know?

A conceptual framework for answering these questions is this simple decomposition of the estimate:

$$\underbrace{\textit{Estimate}}_{\text{observed}} = \underbrace{\textit{Estimand} + \textit{Bias} + \textit{Sampling Error}}_{\text{unobserved}}$$

We'd like to think that there is no bias and the sampling error is essentially zero on average. This would mean that the estimator generating the estimate would be equal to the estimand *on average*, i.e.

$$E(\textit{estimator}) = \textit{estimand}.$$

If this is true, we say the estimator is *unbiased*.

Sources of bias

Here are several data problems that can introduce bias to our analyses:

- *Censoring*. We've already been introduced this problem in the form of earnings topcoding.
- *Sample selection*. The issue here is whether the data we have represent the population we want to learn about, and if not, whether the selection process itself can be modeled. We can't really talk about sample selection until the population of interest has been specified.
- *Measurement error*. This is just like it sounds – when a variable's empirical measurement does not accurately capture the thing we are interested in.

We will pay some attention to each over the next couple of weeks.

Section 4

Digression on logs

Why logs?

First, we are talking about *natural logs*. The natural log function is the log with base e (2.718...) or the inverse of the exponential function:

$$\log(e^y) = \log[\exp(y)] = y,$$

where e or $\exp(\cdot)$ is the exponential function. Some people use "ln" to indicate natural log. Regardless, whenever we write "log" we will mean natural log. Also, R's log function computes natural log values by default.

Ok, why logs? Sometimes the story is in percentage terms – like the percentage difference in earnings between two groups or the percentage change in earnings over time. This is true for most variables that take on large dollar values (like sales) or integer values (like subscriptions).

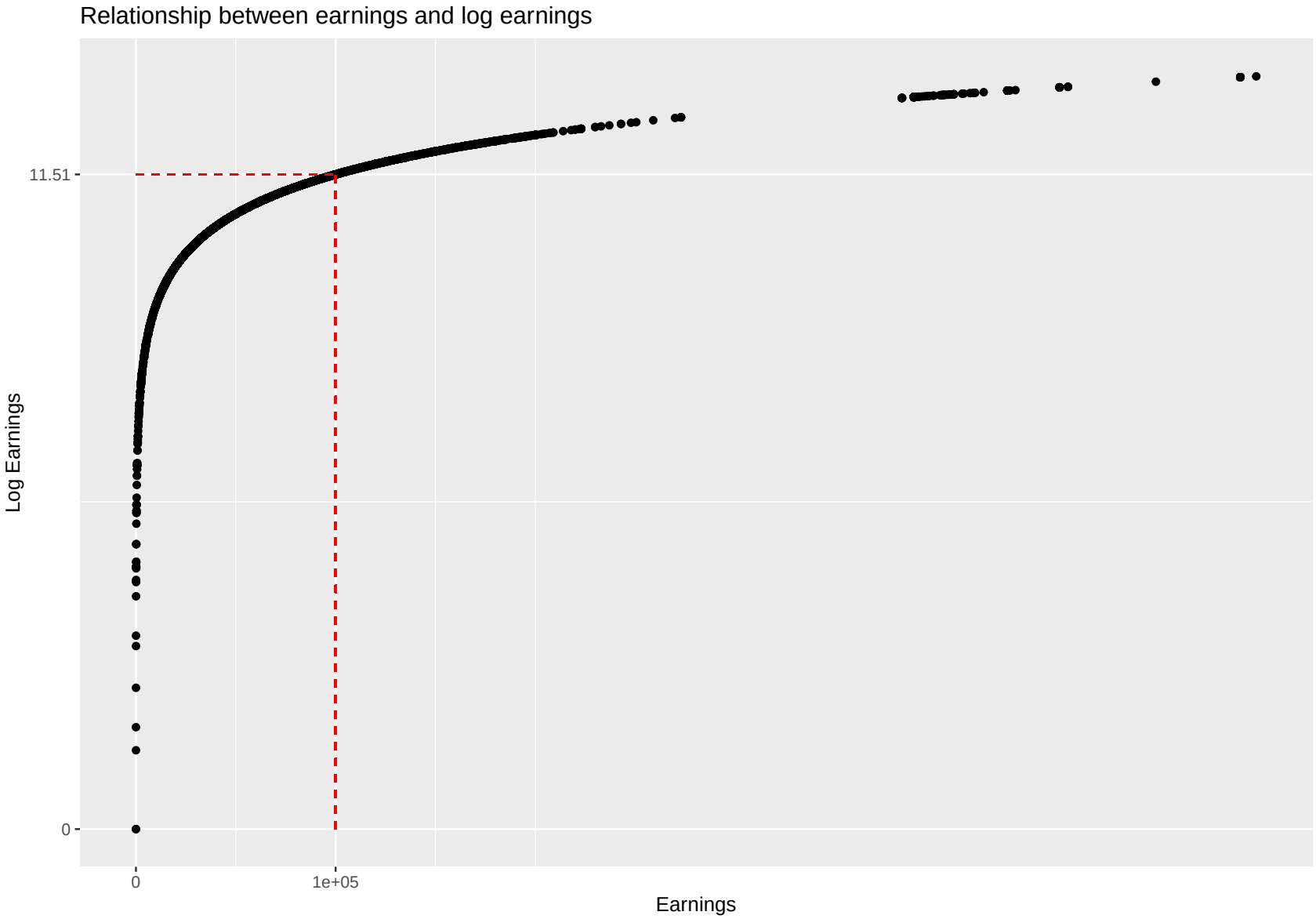
Facts about logs

Remember these?

$$\log(y) < 0, \ 0 < y < 1$$
$$\log(1) = 0$$
$$\log(y) > 0, \ y > 1$$
$$\log(xy) = \log(x) + \log(y)$$
$$\log(x/y) = \log(x) - \log(y)$$

Also, remember that the log function is defined only for positive values. This leads to a dilemma about what to do when the data show 0 earnings or sales. A common ad hoc solution is to use the approximation $\log(1 + y)$ for small values of y. Depending on the context, there are better ways to handle the problem.

Relationship between earnings and log earnings



Relationship code chunk

```
ggplot(cps_mar, aes(x = earnings, y = log(earnings))) +
  geom_point() +
  ggtitle("Relationship between earnings and log earnings") +
  xlab("Earnings") +
  ylab("Log Earnings") +
  geom_segment(aes(x = 0, y = log(100000),
                   xend = 100000, yend = log(100000)),
               color = "red", linetype = "dashed") +
  geom_segment(aes(x = 100000, y = 0,
                   xend = 100000, yend = log(100000)),
               color = "red", linetype = "dashed") +
  scale_x_continuous(breaks = c(0, 100000), labels = c(0, 100000)) +
  scale_y_continuous(breaks = c(0, log(100000)),
                     labels = c(0, round(log(100000), 2)))
```

Percentage changes

As we said, the main use of the log function is to talk about percentage changes. For relatively small differences/changes,

$$\log(y_1) - \log(y_0) \approx \frac{y_1 - y_0}{y_0},$$

where the subscripts distinguish two groups or time periods. For example...

```
y1 <- 105000 # Earnings this year
y0 <- 100000 # Earnings last year
pct_raise_actual <- ((y1-y0)/y0)*100
pct_raise_diflog <- (log(y1)-log(y0))*100
print(paste("Actual percentage raise =", pct_raise_actual))
```

```
## [1] "Actual percentage raise = 5"
```

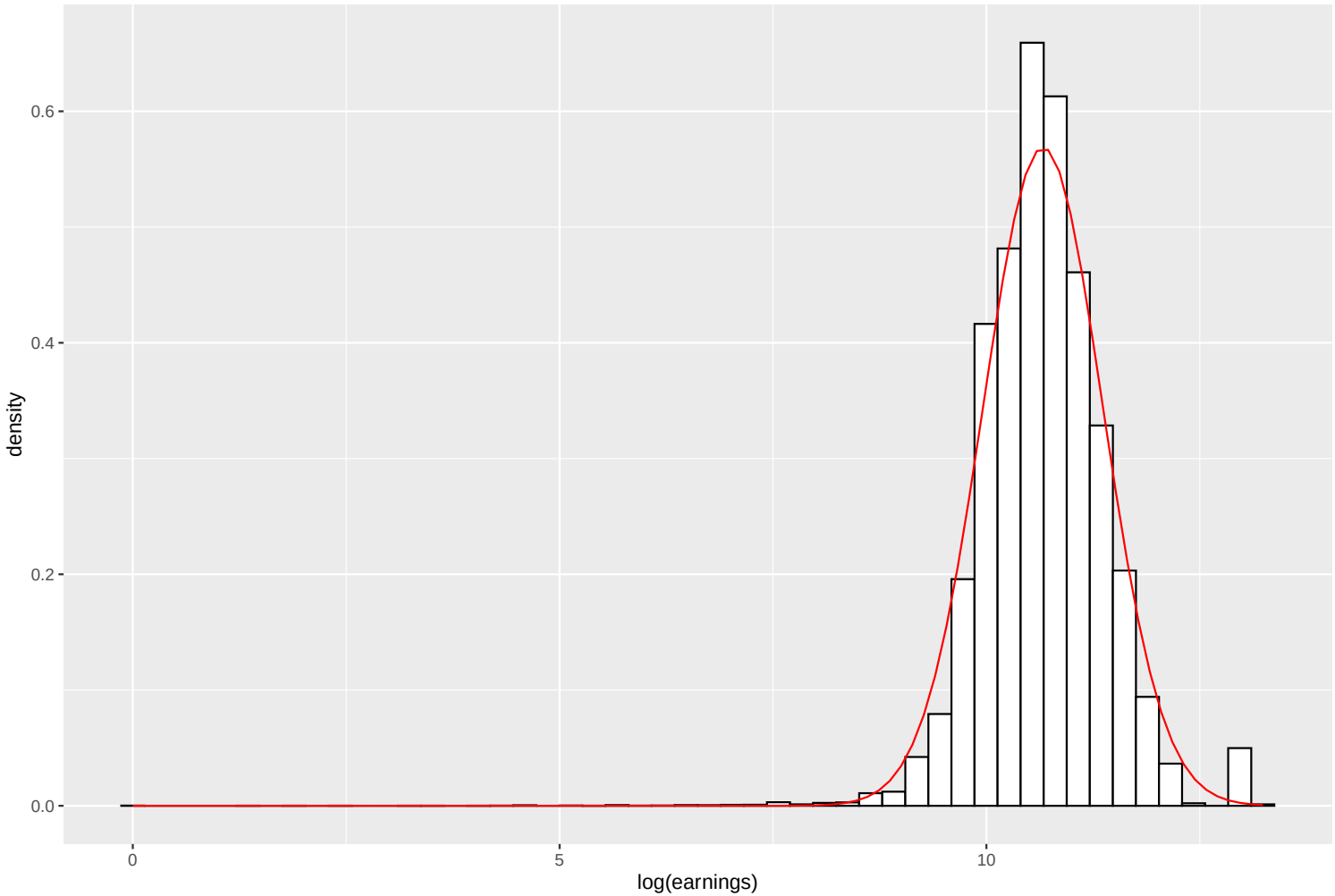
```
print(paste("Approx percentage raise =", pct_raise_diflog))
```

```
## [1] "Approx percentage raise = 4.87901641694322"
```

Distribution of log earnings

If the lognormal distribution is a good model of earnings, maybe the normal distribution is a good model of log earnings.

Figure 3. Log wage histogram with normal density overlay



Section 5

Making Comparisons

Earnings conditional on gender

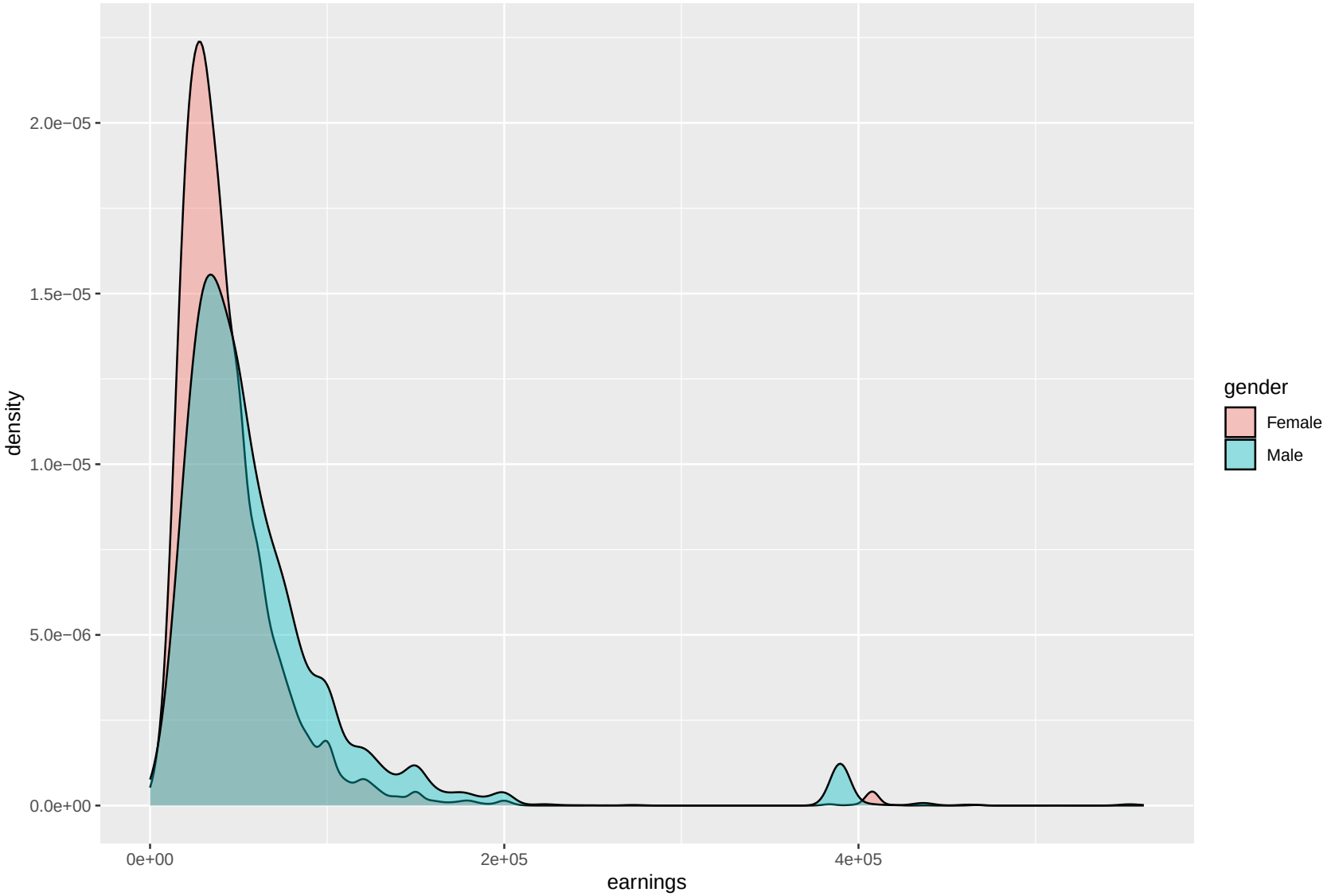
Now, let's look at the distribution of earnings *conditional* on gender:

$$f(earnings \,|\, gender) = P(earnings = amount \,|\, gender = category)$$

Instead of plotting the simple histograms, we'll smooth them out with `geom_density` to make the comparison easier to grasp.

Earnings distributions for women and men

Figure 4. Distribution of earnings by gender



Earning six figures, women vs men

Now, let's estimate the likelihood of earning at least $100,000 for women and men,

$$P(earnings \geq 100,000 \,|\, gender),$$

using the share of six-figure earners in each category as our estimates.

```
six_figs_fvm <- cps_mar %>%
  group_by(gender) %>%
  summarise(six_figs_shrs = mean(earnings >= 100000))
print(six_figs_fvm)
```

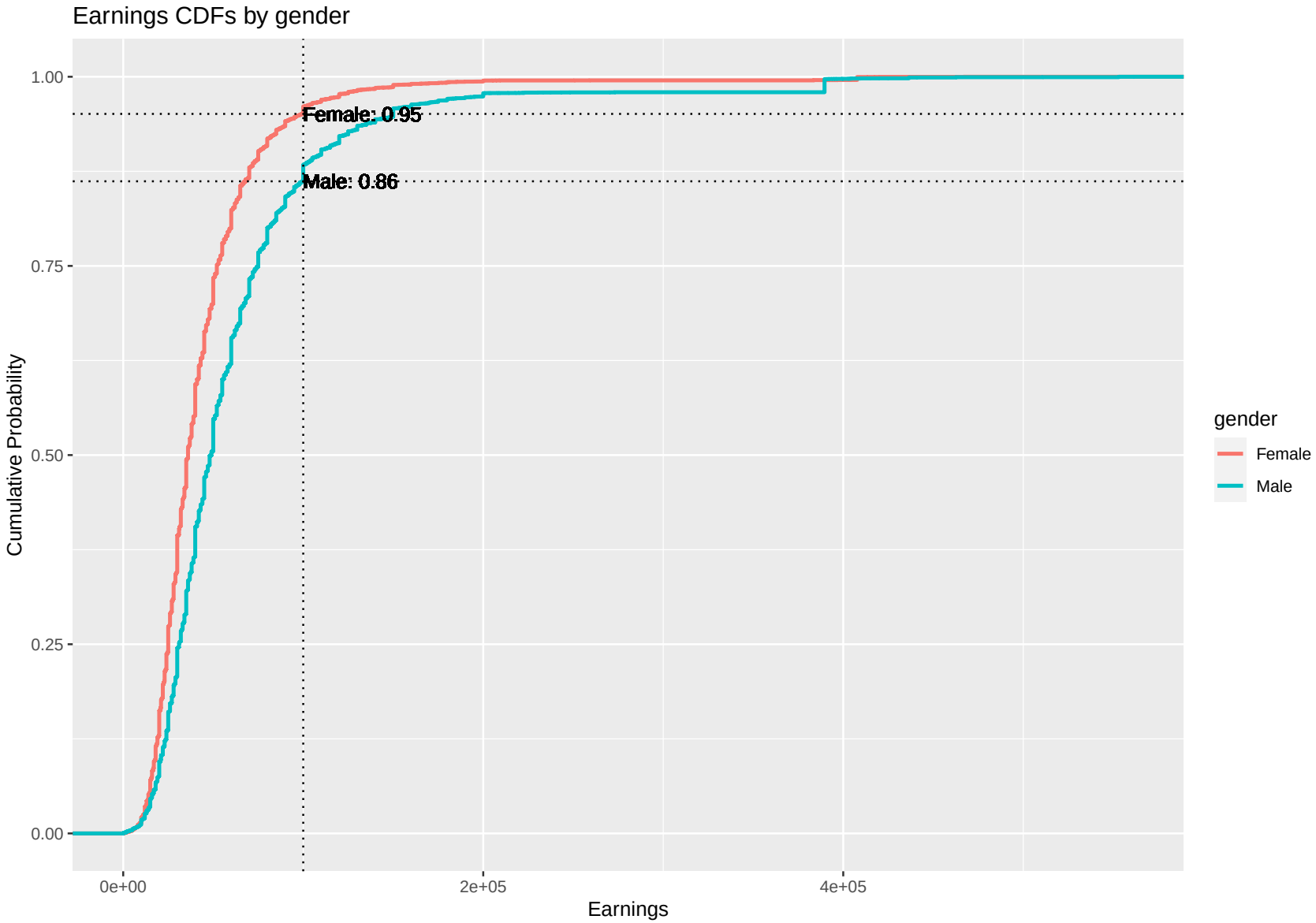
```
## # A tibble: 2 x 2
##   gender six_figs_shrs
##   <chr>          <dbl>
## 1 Female        0.0491
## 2 Male          0.138
```

As seen in a CDF comparison

The shares of women and men earning at least $100,000 can be easily extracted from the *cumulative distribution function (CDF)*. Here is how to plot the earnings CDFs.

```
cdf_values <- cps_mar %>%
  group_by(gender) %>%
  summarise(cdf_100k = mean(earnings < 100000))
female_cdf_100k <- cdf_values$cdf_100k[cdf_values$gender == "Female"]
male_cdf_100k   <- cdf_values$cdf_100k[cdf_values$gender == "Male"]
cdfs <- ggplot(cps_mar, aes(x = earnings, color = gender)) +
  stat_ecdf(size = 1) +
  labs(title = "Earnings CDFs by gender",
       x = "Earnings",
       y = "Cumulative Probability") +
  geom_vline(aes(xintercept = 100000), linetype = "dotted",
             color = "black", size = 0.5) +
  geom_hline(yintercept = female_cdf_100k, linetype = "dotted",
             color = "black", size = 0.5) +
  geom_hline(yintercept = male_cdf_100k, linetype = "dotted",
             color = "black", size = 0.5) +
  geom_text(aes(x = 100000, y = female_cdf_100k,
                label = sprintf("Female: %.2f", female_cdf_100k)),
            color = "black", hjust = "left") +
  geom_text(aes(x = 100000, y = male_cdf_100k,
                label = sprintf("Male: %.2f", male_cdf_100k)),
            color = "black", hjust = "left")
```

Earnings CDFs for women and men



Average earnings conditional on gender

Next, let's talk about how *average earnings* vary by gender. Here we will call up the conditional expectations function (CEF):

$$E(earnings \mid gender)$$

which will define our estimands.

Most of you probably know where this is going. We'll estimate $E(earnings \mid gender = Female)$ and $E(earnings \mid gender = Male)$ by *plugging in* their respective category averages for the expectations. Under random sampling, the principle of plugging in sample averages for expectations has strong justification.

What expected value means

First, we should probably remind ourselves about what *expected value* means. If you had to pick a single number to represent a random variable, what would it be? You might say whatever a rational person would *expect* the value to be when it is observed. This way of thinking leads us to the concept of a weighted average of all of the random variable's possible outcomes. For a *discrete* random variable Y, it looks like this:

$$E(Y) = \sum_{j=1}^{J} y_j f(y_j),$$

where the PDF, $f(y_j) = P(Y = y_j)$, gives the weights and J indicates the number of values. Understand that expected value, $E(\cdot)$, is an *operator* that returns a real number when applied to a random variable.

Importantly, $E(Y)$ is a *population* value and is defined without reference to a particular sample. As such, it is often called the *population mean* and denoted by the parameter μ, which is a *constant*. While there are many sample averages, there is only one $E(Y)$.

Estimating expected values

Because we never know the underlying PDF, we have to estimate $E(Y)$. The standard choice is the sample mean:

$$\hat{\mu} = \hat{E}(Y) = \overline{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i,$$

where the "hat" indicates an estimator.

As we hinted at earlier, under random sampling plugging in $\overline{Y}$ works well. It's not hard to show that $\overline{Y}$ is *unbiased* and the LLN applies $\overline{Y} \to E(Y)$ as $N \to \infty$.

Conditional expectations

The basic idea is the same, its just now the weights are given by the conditional PDF, which defines the probabilities of Y's values conditional on the value of some other random variable, X. In the discrete case, we have

$$E(Y|x) = \sum_j y_j f(y_j|x),$$

where $f(y_j|x) = P(Y = y_j|X = x)$.

It should be clear that the *CEF*, $E(Y|X)$, is a function of X, and as such, is a *random variable*. We make this explicit by writing $E(Y|X)$ as $\mu(X)$.

The plug-in principle applies to estimating the CEF as well. In this case, we plug in the sample mean of Y for each value of X, which, of course, is what we will do to estimate average earnings for women and men.

If you need an expected value refresher, check out this resource.

Expected values of indicator variables

If you didn't know this already, the expected value of an *indicator variable* (like `gender`) equals the probability that the indicator equals 1 (or "yes" or whatever it should be indicating). Let Y be an indicator (sometimes we say "dummy") variable taking on the values 1 and 0. Then,

$$E(Y) = P(Y = 1).$$

This means we can take the sample average as an estimator of $P(Y = 1)$:

$$\overline{Y} = \hat{P}(Y = 1).$$

We've been using this fact when we plugged in the gender shares for the probability that a woman or man earns six figures.

Comparing expected earnings of women and men

So, when we say we want to compare the average earnings of women and men, we have in mind estimating the CEF,

$$E(earnings|gender).$$

In terms of our CPS variables, this means estimating

$$\mu_{Female} = E(earnings \,|\, gender = Female)$$
$$\mu_{Male} = E(earnings \,|\, gender = Male).$$

As we anticipated, we will do this by plugging in the sample average earnings for each category:

$$\hat{\mu}_{Female} = \frac{\sum_{i=1}^{N} \mathbf{1}(gender_i = Female) \times earnings_i}{N_{Female}}$$
$$\hat{\mu}_{Male} = \frac{\sum_{i=1}^{N} \mathbf{1}(gender_i = Male) \times earnings_i}{N_{Male}}.$$

Law of iterated expectations

It should be intuitive that we can recover average overall earnings from the CEF. This is the implication of the *law of iterated expectations (LIE)*, which says

$$\begin{aligned} E(earnings) &= E[E(earnings|gender)] \\ &= E(earnings|gender = Female)P(Female) \\ &+ E(earnings|gender = Male)P(Male), \end{aligned}$$

as the CEF is a random variable with its own distribution and expected value.

In general, the LIE says for any two random variables Y and X,

$$E(Y) = E[E(Y|X)].$$

Section 6

The Gender Earnings Gap

How big in dollars?

How big is the gender earnings gap in dollars?

```
earnings_bar <- cps_mar%>%
  group_by(gender) %>%
  summarise(avg_earnings= mean(earnings))
earnings_bar
```

```
## # A tibble: 2 x 2
##   gender avg_earnings
##   <chr>          <dbl>
## 1 Female       44224.
## 2 Male         63148.
```

```
dollar_gap = (earnings_bar$avg_earnings[2]
              - earnings_bar$avg_earnings[1])
print(paste("Difference in average earnings =",
            round(dollar_gap,2)))
```

```
## [1] "Difference in average earnings = 18923.6"
```

How big in percentage terms?

This dollar difference does not really tell you what you want to know. The question is whether $19K a big number. So, let's calculate the percentage difference in earnings:

```
pct_gap = (earnings_bar$avg_earnings[2] -
             earnings_bar$avg_earnings[1])/
             earnings_bar$avg_earnings[1]*100
print(paste("Percentage difference in average earnings =",
            round(pct_gap,2)))
```

```
## [1] "Percentage difference in average earnings = 42.79"
```

Using the log approximation

What is the answer the difference-in-logs approximation gives here?

```
learnings_bar <- cps_mar%>%
  group_by(gender) %>%
  summarise(avg_learnings= mean(log(earnings)))
learnings_bar
```

```
## # A tibble: 2 x 2
##   gender avg_learnings
##   <chr>          <dbl>
## 1 Female          10.5
## 2 Male            10.8
```

```
approx_pct_gap = (learnings_bar$avg_learnings[2]
                  - learnings_bar$avg_learnings[1])*100
print(paste("Approximate percentage difference in average earnings =",
            round(approx_pct_gap,2)))
```

```
## [1] "Approximate percentage difference in average earnings = 28.74"
```

This is a warning that the difference in logs approximation to percentage changes gets worse as the change gets larger.

Ok, why the gap?

This is the real question, isn't it? What are some potential explanations?

- Discrimination
- Hours constraints
- Preferences
- Education
- Experience

What do the Uber data suggest?