Module 5 synchronous class code (complete)

Hints

- The information for the quiz intentionally includes some sample code that you might find helpful for your problem set. Don't forget about it just because the quiz is over.
- Problem set 5 has some questions that are meant to be revision for earlier modules. E.g., mean vs median for skewed data, what histograms and boxplots do and don't show. If you're having trouble with any of the early parts of question 1, make sure you look back at your notes and ask if you get stuck as this is also good revision for the midterm and future project.

Set up

- Load the `tidyverse` package.
- Load and save the two dataset you need for the problem set,
  - `ps5_sample_data.csv` (call it `orig_sample`), and
  - `ps5_census_data.csv` (call it `census`).
- Glimpse the `census` data.

```
library(tidyverse)

orig_sample <- read_csv('ps5_sample_data.csv')
census <- read_csv('ps5_census_data.csv')

glimpse(census)
```

```
## Rows: 7,601
## Columns: 9
## $ hhld_inc_binary        <chr> "$50,000+ per year", "$50,000+ per year", "$50,~
## $ hhld_inc_num           <dbl> 146000, 228000, 106000, 47000, 36000, 127000, 1~
## $ q7                     <chr> "None", "None", "None", "One or more", "None", ~
## $ q11                    <chr> "Not employed at this time", "Employed full tim~
## $ q13                    <chr> NA, "Do all of your work from home", NA, "Work ~
## $ q56                    <chr> "No", "Yes", "No", "Yes", "Yes", "No", "No", "N~
## $ lost_all_savings_q8    <chr> "No", "No", "No", "Yes", "No", "No", "No", "No"~
## $ changed_employers_q12c <chr> NA, "No", NA, "No", NA, "No", "No", "No", "No",~
## $ age_oldest             <dbl> 65, 49, 55, 70, 32, 38, 58, 80, 19, 20, 23, 68,~
```

Programming tip: How do you make a code chunk?

You can insert an R code chunk either using the RStudio toolbar (the Insert button) or the keyboard shortcut Ctrl + Alt + I (Cmd + Option + I on macOS). [1]

Stats mini-check

Proportions and probabilities come in different flavours. One important 'flavour' is **conditional probabilities**. It is the probability of one event occurring, given that another event/assumption is true.

So I might have a simple (also called marginal) probability, like "The probability of passing STA130 is 80%" but, I might want to instead focus just on students who make consistent effort in the course and make a statement like "Given a student completes all 9 problem sets, the probability they pass STA130 is 99%".

A common 'clue' word, that lets us know we might be looking for a conditional probability is 'of'. For example, "**OF** employed (`q11`) people in Representaville, USA, what proportion changed jobs (`changed_employers_q12c`)?" We'll look at this in the next part.

[1] Source: *R Markdown: The Definitive Guide.* (2021-04-09). Yihui Xie, J. J. Allaire, Garrett Grolemund, https://bookdown.org/yihui/rmarkdown/r-code.html

Teaching world

Suppose that **Dataset 1 (census)** is a complete census (survey of the entire population) of people aged 18 and over in Representaville, USA.

Looking at our census data, calculate the proportion of employed people that changed jobs over the pandemic.

- Start by inserting a chunk.
- Filter appropriately.
- Do the calculation.
- Save as an atomic variable with as.numeric().

```
# There are a couple ways to do this, two examples below

# The most manual way
census %>%
  filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
  group_by(changed_employers_q12c) %>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   changed_employers_q12c [2]
##   changed_employers_q12c     n
##   <chr>                  <int>
## 1 No                      3561
## 2 Yes                      948
```

```
parameter <- 948/(948+3561)
parameter
```

```
## [1] 0.2102462
```

```
# My favourite way, because I am less likely to make typos with the numbers
parameter <- census %>%
  filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
  mutate(changed = changed_employers_q12c == "Yes") %>%
  summarise(prop = mean(changed)) %>%
  as.numeric()
parameter
```

```
## [1] 0.2102462
```

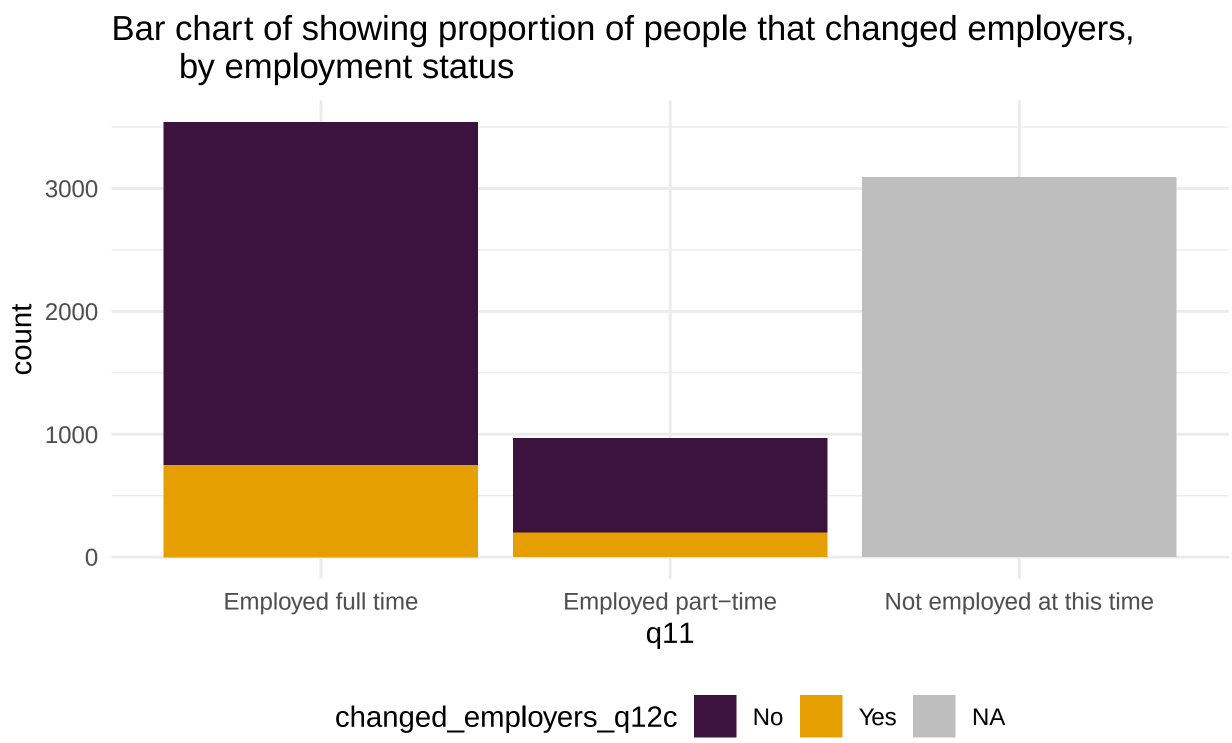
21% of employed people, 18+ in Representaville, USA changed their employers over the course of the pandemic. This is our parameter, because in **teaching world** we have stats super powers!

Using the census data set, produce (i) a table of counts for every grouping of the levels of `q11` and `changed_employers_q12c`(ii) a relevant visualization (with an appropriate title). DON'T filter the data first.

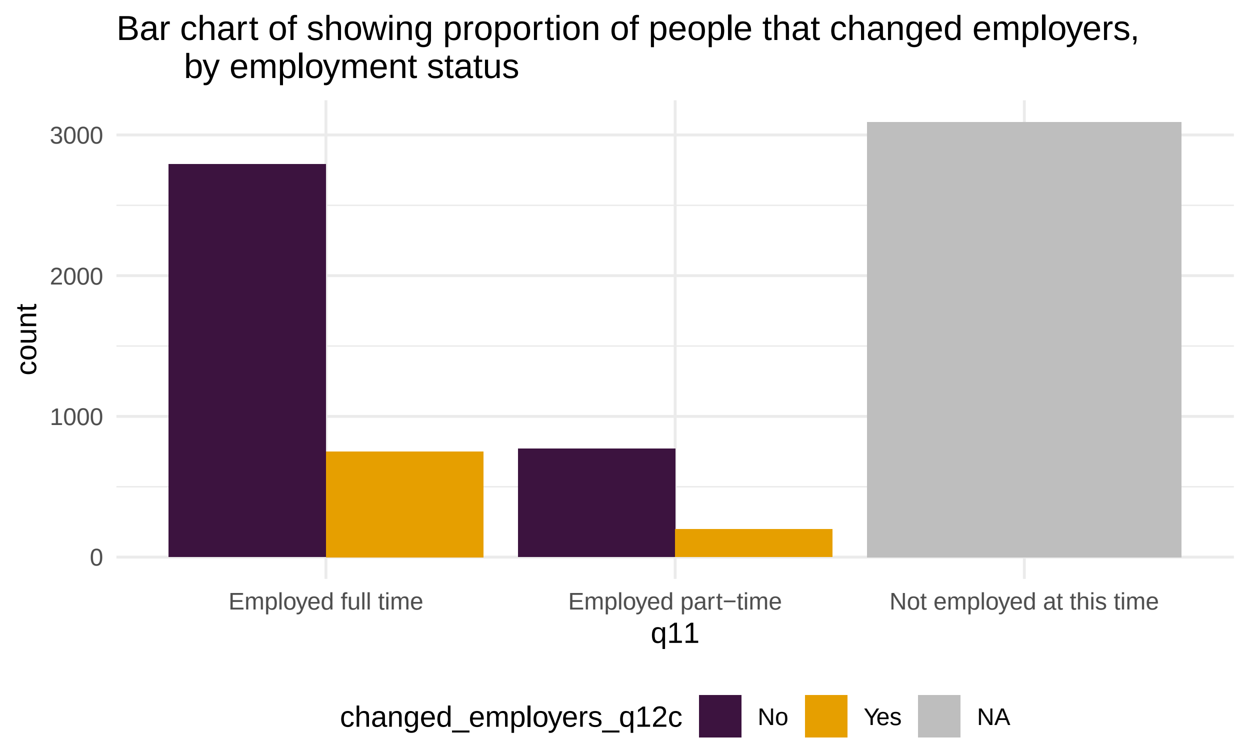
```
summary_table_census <- census %>%
  group_by(q11, changed_employers_q12c) %>%
  count()

summary_table_census
```

```
## # A tibble: 5 x 3
## # Groups:   q11, changed_employers_q12c [5]
##   q11                       changed_employers_q12c     n
##   <chr>                     <chr>                  <int>
## 1 Employed full time        No                      2791
## 2 Employed full time        Yes                      750
## 3 Employed part-time        No                       770
## 4 Employed part-time        Yes                      198
## 5 Not employed at this time <NA>                    3092
```

```
# Stacked
census %>%
  ggplot(aes(q11, fill = changed_employers_q12c)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Bar chart of showing proportion of people that changed employers,
       by employment status") +
   scale_fill_manual(values=c("No" = "#3C133F", "Yes" = "#E69F00"), na.value = "grey") +
  # look I can change the colours!
  theme(legend.position = "bottom")
```




```
# You can also force them to be side to side
census %>%
  ggplot(aes(q11, fill = changed_employers_q12c)) +
  geom_bar(position = position_dodge()) +
  theme_minimal() +
  labs(title = "Bar chart of showing proportion of people that changed employers,
       by employment status") +
   scale_fill_manual(values=c("No" = "#3C133F", "Yes" = "#E69F00"), na.value = "grey")  +
  # look I can change the colours!
  theme(legend.position = "bottom")
```

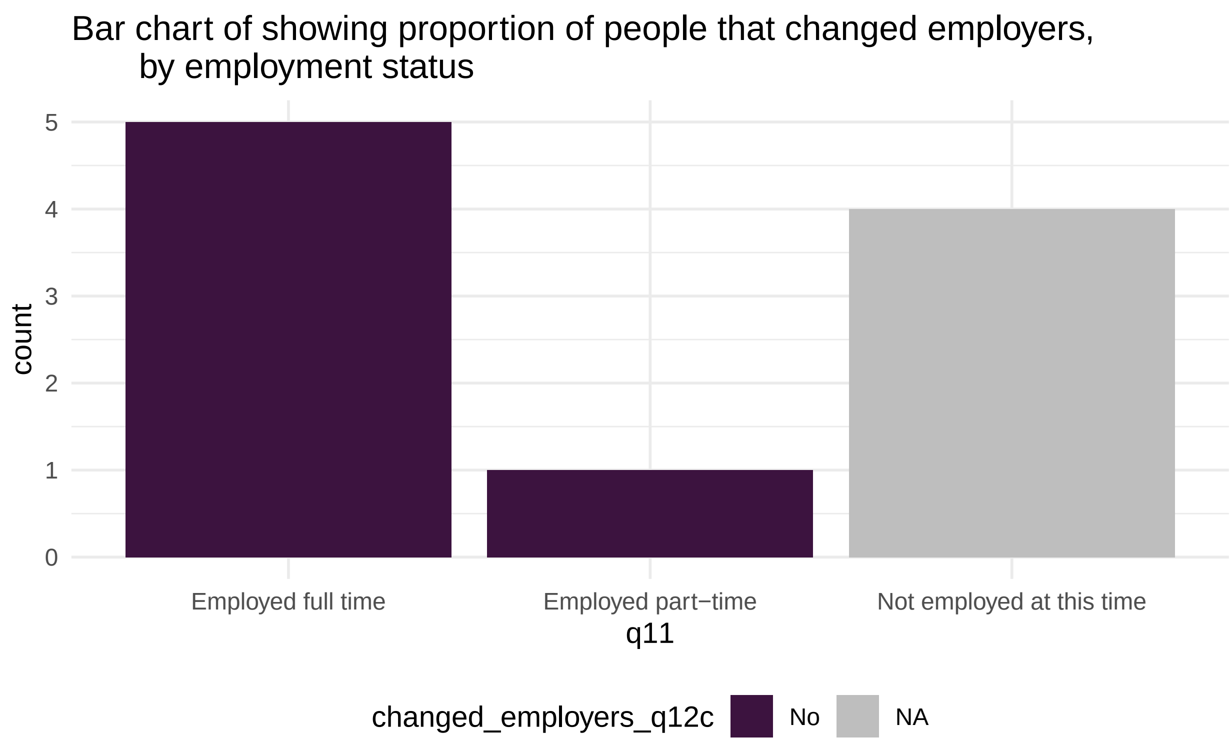


Select a random sample of size n=10 (without replacement) from the census data and calculate the proportion of employed people that changed employers. Recreate the visualization above for your new data. Set the seed as the last *three* digits of your student ID number.

```
set.seed(123) # suppose 123 are the last 3 digits of my student number

sample_10 <- census %>%
  sample_n(size = 10)

sample_10 %>%
  filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
  mutate(changed = changed_employers_q12c == "Yes") %>%
  summarise(prop = mean(changed)) %>%
  as.numeric()
```

```
## [1] 0
```

```
sample_10 %>%
  ggplot(aes(q11, fill = changed_employers_q12c)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Bar chart of showing proportion of people that changed employers,
       by employment status") +
   scale_fill_manual(values=c("No" = "#3C133F", "Yes" = "#E69F00"), na.value = "grey") +
  theme(legend.position = "bottom") # move my legend
```

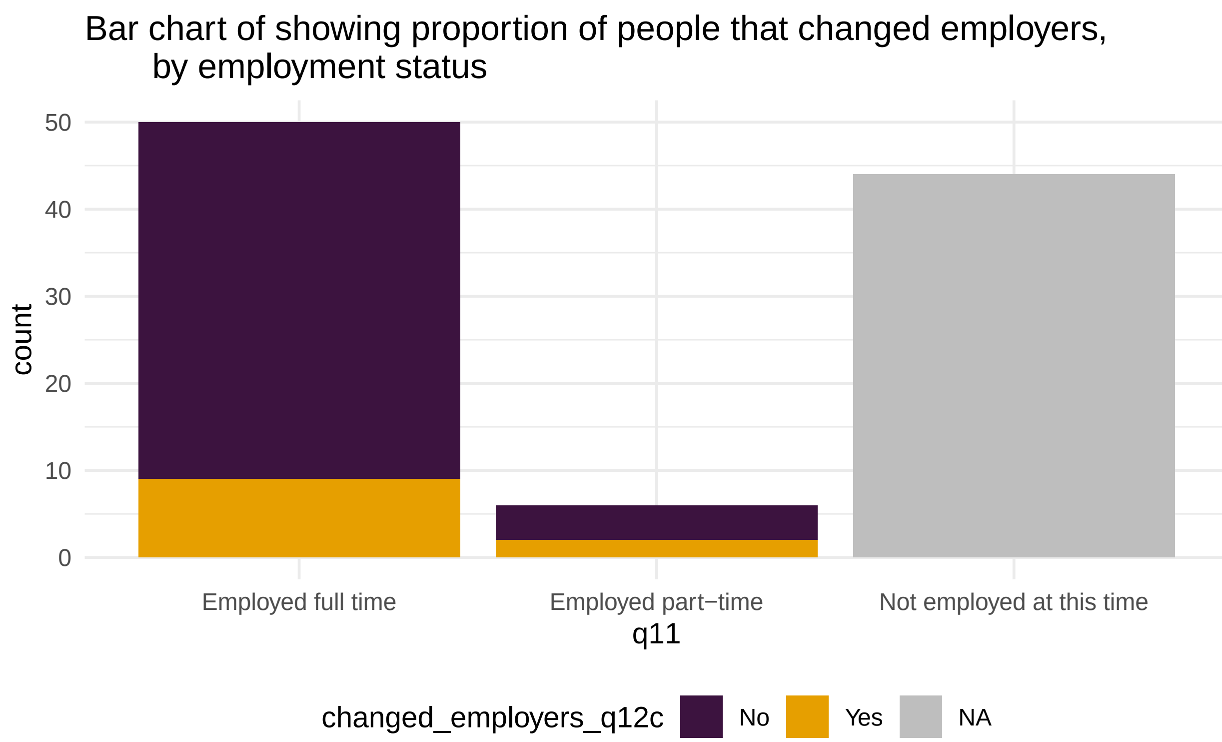


Now, select a random sample of size n=100 (without replacement) from the census data and again pand calculate the proportion of employed people that changed employers. Set the seed as the last *three* digits of your student ID number.

```
set.seed(123) # suppose 123 are the last 3 digits of my student number

sample_100 <- census %>%
  sample_n(size = 100)

sample_100 %>%
  filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
  mutate(changed = changed_employers_q12c == "Yes") %>%
  summarise(prop = mean(changed)) %>%
  as.numeric()
```

```
## [1] 0.1964286
```

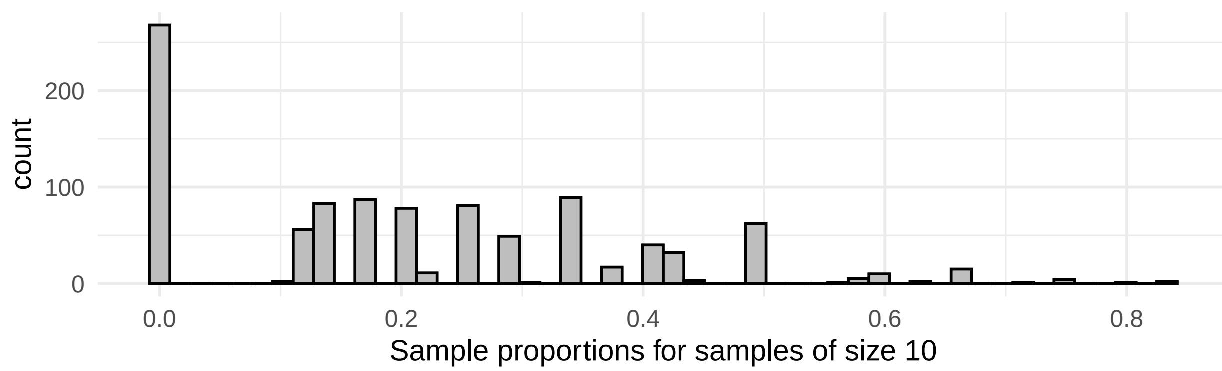
```
sample_100 %>%
  ggplot(aes(q11, fill = changed_employers_q12c)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Bar chart of showing proportion of people that changed employers,
       by employment status") +
   scale_fill_manual(values=c("No" = "#3C133F", "Yes" = "#E69F00"), na.value = "grey") +
  theme(legend.position = "bottom")
```



Bar chart of showing proportion of people that changed employers,
by employment status
count
50
40
30
20
10
0
Employed full time
Employed part−time
Not employed at this time
q11
changed_employers_q12c
No
Yes
NA

Estimate and plot the sampling distribution for the proportion of employed people who changed employers by taking 1000 samples of (i) size n=10 and (ii) size n=100 from the population data and produce appropriate data summaries for each. Set the seed as the last *THREE* digits of your student number for each set of simulations. That is, there should be two graphs and two summary tables, one for each sample size.

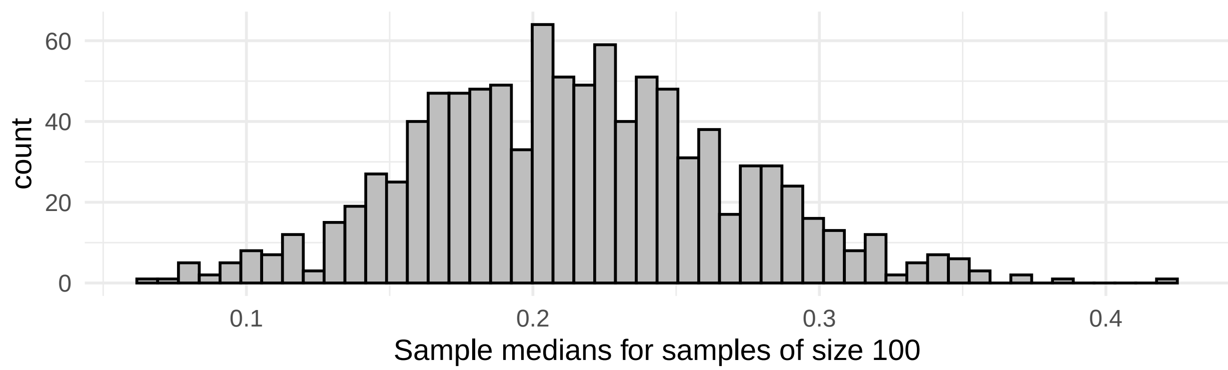
```
## (i)
n <- 10
repetitions <- 1000
set.seed(123)
sim10 <- rep(NA, repetitions)
for (i in 1:repetitions)
{
  new_sim <- census %>%
    sample_n(size = n, replace = FALSE) # TEACHING WORLD!

  sim_prop<- new_sim %>%
      filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
      mutate(changed = changed_employers_q12c == "Yes") %>%
      summarise(prop = mean(changed)) %>%
      as.numeric()
  sim10[i] <- sim_prop
}
sim10 <- tibble(prop = sim10)
sim10 %>% ggplot(aes(x = prop)) +
  geom_histogram(bins = 50, colour = "black", fill = "grey") +
  labs(x="Sample proportions for samples of size 10") +
  theme_minimal()
```




```
# this is a fast version of the tables you've learned for a singe vector
summary(sim10$prop)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.2000  0.2109  0.3333  0.8333
```

```r
## (ii)
n <- 100
repetitions <- 1000
set.seed(123)
sim100 <- rep(NA, repetitions)
for (i in 1:repetitions)
{
  new_sim <- census %>%
    sample_n(size = n, replace = FALSE) # TEACHING WORLD!

  sim_prop <- new_sim %>%
      filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
      mutate(changed = changed_employers_q12c == "Yes") %>%
      summarise(prop = mean(changed)) %>%
      as.numeric()
  sim100[i] <- sim_prop
}
sim100 <- tibble(prop = sim100)
sim100 %>% ggplot(aes(x = prop)) +
  geom_histogram(bins = 50, colour = "black", fill = "grey") +
  labs(x="Sample medians for samples of size 100") +
  theme_minimal()
```




```r
# this is a fast version of the tables you've learned for a single vector
summary(sim100$prop)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0678  0.1754  0.2131  0.2148  0.2500  0.4237
```

CLASS QUESTION: Why is a histogram appropriate here when we were using bar graphs before?

Real world

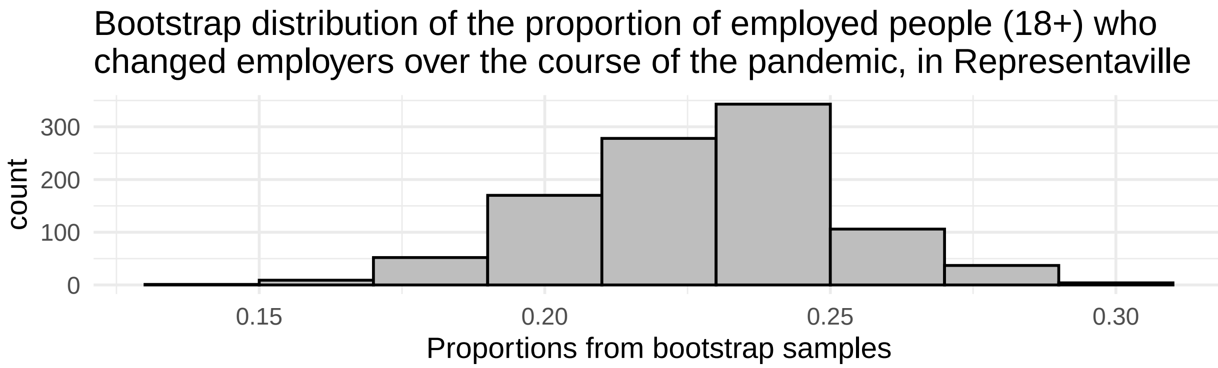
It is actually very hard to get a census of a whole town (why many countries only run their census every 10 years and why they cost so much!), so let's suppose we only have the sample of 500 (`orig_sample`). That is still a pretty big random sample!

Simulate 1000 bootstrap samples and calculate the proportion of employed people (18+) who changed employers over the course of the pandemic, in Representaville, USA. Set the seed as the last *three* digits of your student number.

```
# What is our true sample size, if we're just focusing on the employed people?
sample_size <- orig_sample %>%
  filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
  count() %>%
  as.numeric()

sample_size
```

```
## [1] 304
```

```
set.seed(123) # change to the last three digits of your student number
repetitions <- 1000

boot_p <- rep(NA, repetitions)  # where we'll store the bootstrap proportions

for (i in 1:repetitions)
{
  boot_samp <- orig_sample %>%
    filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
    sample_n(size = sample_size, replace=TRUE) # REAL WOLRD!!
  # THIS is a bootstrap sample!!!

  boot_p[i]  <- boot_samp %>%
      mutate(changed = changed_employers_q12c == "Yes") %>%
      summarise(prop = mean(changed)) %>%
      as.numeric()
}

boot_p <- tibble(boot_p)

ggplot(boot_p, aes(x=boot_p)) + geom_histogram(binwidth=0.02, fill="gray", color="black") +
  labs(x="Proportions from bootstrap samples",
    title="Bootstrap distribution of the proportion of employed people (18+) who\nchanged employers ove
  theme_minimal()
```



Calculate a 95% confidence interval for the proportion of employed people (18+) who changed employers over the course of the pandemic, in Representaville, USA.

```
quantile(boot_p$boot_p, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.1809211 0.2763158
```

With 95% confidence, we can claim that the proportion of employed people (18+) in Representaville, USA who changed employers during over the course of the pandemic was between 18 and 28%.

Does your interval capture the true value we calculated earlier?

Mine does. But we'd expect 5% of the class to get an interval that DOESN'T capture the true value.

Hypothesis test recap and connection!

Let's suppose we saw the stat in the NPR report that "21% of workers have changed employers since the COVID-19 outbreak began" and wanted to test if that was the case in Representaville, USA. (We happen to know it is, in fact, true, as we were basically all knowing stats gods in the first part of this demo.)

Let's perform a hypothesis test (Module 4!) to look into this.

$$H_0 : p_{\text{changed} \mid \text{employed}} = 0.21$$
$$H_A : p_{\text{changed} \mid \text{employed}} \neq 0.21$$

Note: You can read changed | employed as "changed given employed". This symbol is (annoyingly?) also called a 'pipe' but is very different to the pipe we use from tidyverse %>%, and also, in the art sense, "ceci n'est pas une pipe"...

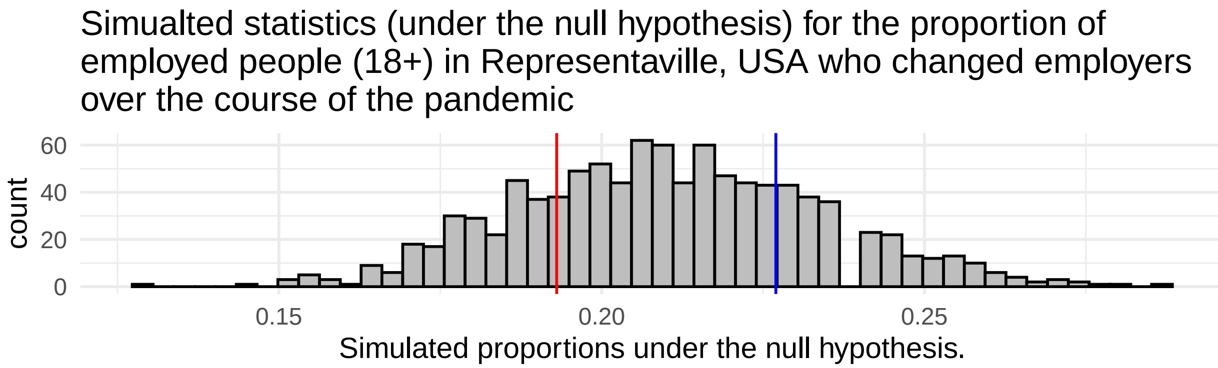
Before we do our test, based on our confidence interval, do you think we'll have evidence against your null hypothesis?

I don't think we'll have any evidence against our hypothesis as the 21% is plausible value for our parameter, based on our confidence interval. We could be wrong! But that's our best guess from our data and now we'll look at it from a different direction.

```r
# What is our true sample size, if we're just focusing on the employed people?
sample_size <- orig_sample %>%
  filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
  count() %>%
  as.numeric()

sample_size
```

```
## [1] 304
```

```r
# What is our test statistic?
test_stat <- orig_sample %>%
  filter(q11 == "Employed full time" | q11 == "Employed part-time") %>%
  mutate(changed = changed_employers_q12c == "Yes") %>%
  summarise(prop = mean(changed)) %>%
  as.numeric()

# Do the simulation
set.seed(123)
sim_stat <- rep(NA, 1000)

for(i in 1:1000){

  sim <- tibble(changed = sample(c("Yes", "No"), size = sample_size,
              prop = c(0.21, 1-0.21), replace = TRUE))
    # Why is replace TRUE here?

  sim_stat[i] <- sim %>%
    mutate(changed_logical = changed == "Yes") %>%
    summarise(prop = mean(changed_logical)) %>%
    as.numeric()

}

# Covert to tibble for easy plotting
simulated_stats <- tibble(sim_stat = sim_stat)

# Plot
simulated_stats %>%
  ggplot(aes(x = sim_stat)) +
  geom_histogram(bins=50, fill="grey", color="black") +
  theme_minimal() +
  geom_vline(xintercept = test_stat, color = "blue") +
  geom_vline(xintercept = 0.21-(test_stat-0.21), color="red") +
  labs(title = "Simualted statistics (under the null hypothesis) for the proportion of \nemployed peopl
     x = "Simulated proportions under the null hypothesis.")
```




```
# Calculate p-value
p_val <- simulated_stats %>%
  filter(sim_stat <= 0.21-(test_stat-0.21) | sim_stat >= test_stat) %>%
  summarise(n()/1000) %>%
  as.numeric()

p_val
```

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