Problem Set 08

WRITE YOUR NAME HERE

WRITE DATE HERE

Contents

Collaboration

Please indicate who you collaborated with on this problem set:

Background

In this problem set we will use a small **sample** of data from the General Social Survey. The survey is designed to monitor changes in both social characteristics and attitudes. You will work with a **sample** from one neighborhood. The full neighborhood of **ALL individuals** is the population. For this problem set we do **not** know the **true population parameters** for any of the variables, because we do not have data on every person in the neighborhood.

Setup

First load the necessary packages

```
# Recall that loading the tidyverse "umbrella" package loads ggplot2, dplyr, and
# readr all at once. Feel free to load these packages any way you choose.
library(tidyverse)
library(moderndive)
```

Next load the data set, from where it is stored on the web:

```
gss_sample <- read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vSypSoDCMH2N76Vo2dZRPkw2q3t1mbv
```

Be sure to take a look at the data in the **viewer**. Each row in the data set is a person that was surveyed (100 rows or cases in total). The variables in the data set include each respondent's age, `age`, `race`, and number of hours of TV watched a day `tvhours`.

Setting a seed: We will take some random samples and build sampling distributions in this lab. In order to make sure R takes the same random sample every time you run your code, you can do what is called "setting a seed". Do this in any code chunk that you take a random sample!

You can set a seed like so. Any number will do. (You do not need to run this right now…just showing you how)

```
set.seed(45)
```

Confidence intervals from a bootstrap resample

Step 1: Take 1000 bootstrap resamples

The following code tells R to take 1000 bootstrap resamples from the `gss_sample` data. You can set the seed to whatever value you like!

```
set.seed(42)

boot_samp_1000 <- gss_sample %>%
  rep_sample_n(size = 100, reps = 1000, replace = TRUE)
```

Note a few important details about the `rep_sample_n` function, and bootstrap sampling in general:

- `size = 100` tells R that each bootstrap resample we take has 100 cases… the size of the original sample
- `reps = 1000` tells R to take 1000 bootstrap resamples (each of size 100).
- The `replace = TRUE` argument tells R that in each bootstrap resample, we can include a row from `gss_sample` multiple times. So if for instance, respondent # 12 is the first random resample taken here, respondent 12 is still available to be resampled **again** at random. Thus, some people may appear **multiple times** in our bootstrap resample, and some people from the original data set may not appear at all.
- We save the results in a data frame `boot_samp_1000`.

Take a look at the `boot_samp_1000` data frame we just generated in RStudio's data viewer. Note that the `replicate` column labels each bootstrap resample (the first 100 rows are labeled 1, the next 100 rows are labeled 2, etc.)

Question 1

How many rows does `boot_samp_1000` have? **Why?**

Answer:

Step 2: Calculate the bootstrap statistic

Let's say we want to use the bootstrap resample that we just generated to calculate a confidence interval for the population mean μ_{tv} of `tvhours`. To do so, we need to know the sample mean $\bar{x}$ of `tvhours` **for each of the 1000 bootstrap resamples**. In this case, the sample mean $\bar{x}$ of `tvhours` for **each bootstrap resample** is our **BOOTSTRAP STATISTIC**. We can calculate that with two lines of code, like so:

```
boot_distrib_tv <- boot_samp_1000 %>%
  group_by(replicate) %>%
  summarize(boot_stat = mean(tvhours))
```

Note that:

- The `group_by()` argument tells R to take the sample mean of `tvhours` **separately** for each different `replicate` in the bootstrap resample.
- We put the sample mean for each bootstrap resample in a column called `boot_stat`

  This is the bootstrap distribution for the mean of `tvhours`!

Take a look at the `boot_distrib_tv` we just created in RStudio's data viewer.

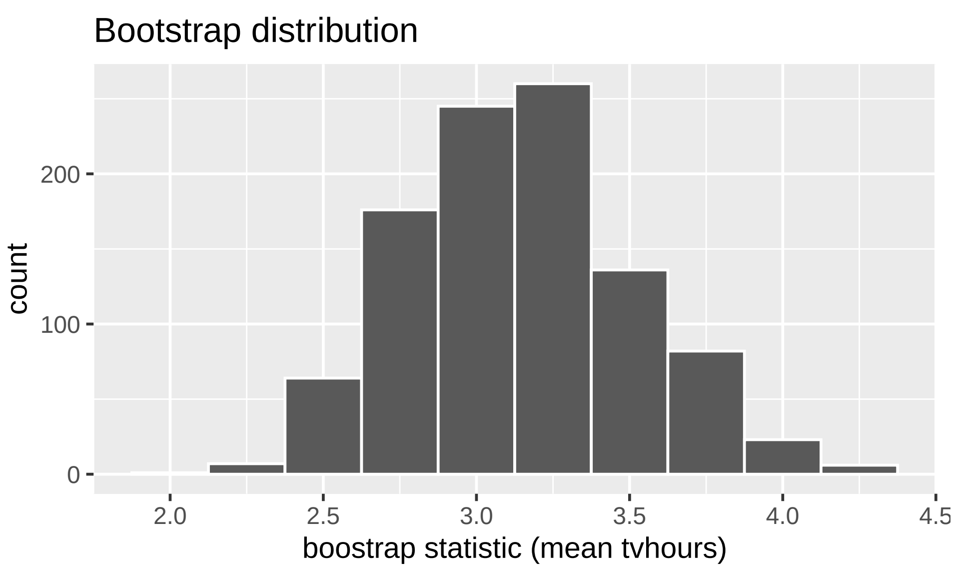
Question 2

How many values of the bootstrap statistic `boot_stat` are there? Please explain **why** there are this many values of the bootstrap statistic.

Answer:.

Visualizing the bootstrap distribution

The bootstrap distribution is shown in the figure below. This is a histogram of the `boot_stat` values from `boot_distrib_tv`.



Step 3: CI from a bootstrap resample

a) CI using the 95% rule

We can now use the bootstrap distribution for the sample mean `tvhours` $\bar{x}$ to calculate a 95% confidence interval for the population mean `tvhours` μ_{tv}, using the "95% rule for bell shaped distributions", which states that the middle 95% of values of a bell/normal shaped distribution are between

$$\text{mean} \pm 1.96 \cdot SD$$

- the mean here would be the mean of the bootstrap distribution

- the SD here is the standard deviation of the bootstrap distribution, which recall has a special name: the **standard error**.

We can thus apply the 95% rule, like so:

```
boot_distrib_tv %>%
  summarize(mean = mean(boot_stat),
            se = sd(boot_stat),
            lower_ci = mean - (1.96 * se),
            upper_ci = mean + (1.96 * se))
```

```
## # A tibble: 1 x 4
##    mean    se lower_ci upper_ci
##   <dbl> <dbl>    <dbl>    <dbl>
## 1  3.14 0.365     2.43     3.86
```

b) CI using percentile method

You can also calculate a 95% confidence interval using the percentile method. The logic goes like this:

Since our bootstrap resample had 1000 values of `boot_stat`:

1. 950 of the `boot_stat` values fall **inside** this 95% confidence interval, i.e. 95%
2. 25 values fall **below** it. i.e. the lower 2.5%
3. 25 values fall **above** it. i.e. the higher 2.5%

totaling 100%. We can use **the quantiles** of the bootstrap distribution to find these values like so:

```
prct_ci_tv <- boot_distrib_tv %>%
  summarize(lower_ci = quantile(boot_stat, 0.025),
            upper_ci = quantile(boot_stat, 0.975))

prct_ci_tv
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.51     3.89
```

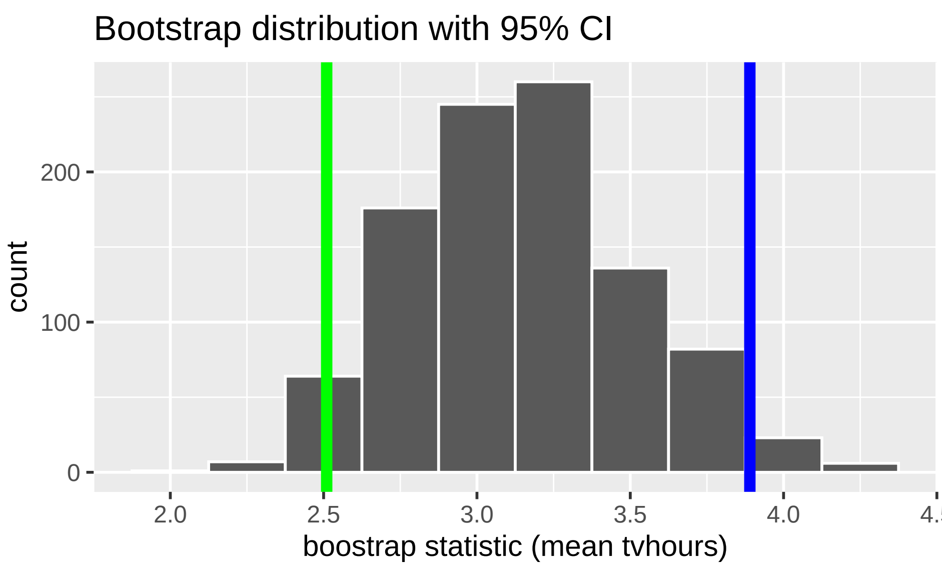
This method

- Asks R to identify the 0.025 quantile of the bootstrap sample means… this is the value **below** which **2.5% of the values of `boot_stat`** fall (or 25 cases in this example… 25/1000 = 0.025)
- Asks R to identify the 0.975 quantile for the bootstrap sample means… this is the value **above** which the other **2.5% of the values of `boot_stat`** fall (or 25 cases in this example 975/1000 = 0.975)
- The middle 95% of the values fall between these two quantiles

Based on these results, we are 95% confident that the **true mean hours of TV watched μ_{tv} in the population** is between the upper and lower CI we just calculated.

Visualizing the Confidence interval

The bootstrap distribution and the 95% confidence intervals we just calculated are shown in the figure below. This is a histogram of the `boot_stat` values from `boot_distrib_tv`. The green line is the lower bound of the 95% CI, and the blue line is the upper bound. 950 of the 1000 bootstrap resamples had a mean for `tvhours` that fell **between** the green and blue lines…25 of the samples had a mean above the blue line, and 25 of the samples had a mean below the green line.



Question 3

If we calculated a **90% confidence interval** for the mean of `tvhours` using this same bootstrap resample and the percentile method, roughly how many of the 1000 values of `tv_mean` would fall between the green and blue lines?

Answer:

Question 4

Use the bootstrap resampling distribution for `tvhours` generated above (`boot_distrib_tv`) and the **percentile** method to calculate a 99% confidence interval for the mean `tvhours`.

Question 5

Which confidence interval is **WIDER**: the 95% confidence interval or the 99% confidence interval for the population mean `tvhours` μ_{tv}? Why?

Answer:

Question 6a

Use the bootstrap resample we generated above (`boot_samp_1000`), to generate a **bootstrap distribution** for the sample mean respondent `age` $\bar{x}$ instead of `tvhours`. Please be sure to **name it something different** than the bootstrap distribution for the sample mean of `tvhours`

Question 6b

Calculate a 95% confidence interval for the population mean respondent `age` μ_{age} using the **95% rule** method.

Question 6c

Calculate a 95% confidence interval for the population mean respondent `age` μ_{age} using the **percentile** method.

Question 6d

How do the 95% confidence intervals you calculated in 6b and 6c compare? i.e. are the 95% CI values similar or are they pretty different?

Answer:

Question 7

Use the **bootstrap resampling distribution** for the sample mean respondent `age` and the percentile method to calculate an 80% confidence interval for the population mean respondent age μ_{age}.

Bootstrap sampling distribution & confidence intervals with categorical variables

The procedure for generating a bootstrap sampling distribution is VERY similar for categorical data. As an example we will generate a bootstrap sampling distribution for the proportion of respondents that identified as a Person of Color.

Step 1: Take 1000 bootstrap resamples

We already did this above! We can use the same `boot_samp_1000` as before.

Step2: Calculate the bootstrap statistic $\hat{p}$

```
boot_distrib_POC <- boot_samp_1000 %>%
  group_by(replicate) %>%
  summarize(n = n(),
            POC_count = sum(race == "POC"),
            boot_stat = POC_count/n)
```

Note that with a categorical variable, the code differs in two important respects now:

- the population parameter that we don't know, but are inferring about via sampling, is now the population proportion p that identify as a POC.
- the sample statistic AKA point estimate that we calculate with the summarize command is now the **sample proportion** $\hat{p}$ rather than a sample mean $\bar{x}$
- To get our proportion $\hat{p}$ of **ONE** of the race categories (POC), we need to **first** calculate the total sample size for each replicate and the count of how many cases are `race == "POC"` in each replicate.

Step 3: Generate the 95% Confidence Interval

a) CI using the 95% rule

The following will calculate the 95% confidence interval for the proportion of people that identified as `POC` using the 95% rule.

```
boot_distrib_POC %>%
  summarize(mean = mean(boot_stat),
            se = sd(boot_stat),
            lower_ci = mean - (1.96 * se),
            upper_ci = mean + (1.96 * se))
```

```
## # A tibble: 1 x 4
##    mean     se lower_ci upper_ci
##   <dbl>  <dbl>    <dbl>    <dbl>
## 1 0.241 0.0421    0.158    0.323
```

b) CI with the percentile method

The following will calculate the 95% confidence interval for the proportion of people that identified as "POC" using the percentile method.

```
boot_distrib_POC %>%
  summarize(lower_ci = quantile(boot_stat, 0.025),
            upper_ci = quantile(boot_stat, 0.975))
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     0.16     0.33
```

Question 8

Calculate a 95% CI for the **population proportion** of respondents p that identified as **White** using EITHER the percentile or 95% rule method. Note that you will first need to generate the bootstrap distribution for the proportion of respondents that identified as `White`.

Confidence intervals based on the theoretical normal distribution

As described in moderndive chapter 8.7.2, not only can we generate confidence intervals using a computer/resampling as we've been doing until now, in many cases there also exists a mathematical formula! This however necessitates a little mathematical/probability theory; a topic we leave to a more advanced statistics class.

To generate a 95% confidence interval based on the theoretical normal distribution, we can use the following formula:

$$\text{point estimate} \pm 1.96 \cdot \widehat{SE}$$

So, for instance if we wanted to calculate the 95% confidence interval for the population mean of `tvhours` μ_{tv} that respondents watched based on our sample:

- the **point estimate** AKA **sample statistic** in this case would be the sample mean number of `tvhours` from the sample: $\bar{x}$
- We would estimate the standard error using the formula

$$\widehat{SE} \approx \frac{s}{\sqrt{n}}$$

where s is the **sample standard deviation**, which is a point estimate of the **population standard deviation** σ.

Thus a 95% CI would be

$$\bar{x} \pm 1.96 \cdot \widehat{SE} = \bar{x} \pm 1.96 \cdot \frac{s}{\sqrt{n}}$$

We can make these calculations in R like so:

```
gss_sample %>%
  summarize(x_bar = mean(tvhours),
            sd = sd(tvhours),
            n = n(),
            se = sd/sqrt(n),
            lower_ci = x_bar - 1.96 * se,
            upper_ci = x_bar + 1.96 * se)
```

```
## # A tibble: 1 x 6
##   x_bar    sd     n    se lower_ci upper_ci
##   <dbl> <dbl> <int> <dbl>    <dbl>    <dbl>
## 1  3.14  3.59   100 0.359     2.44     3.84
```

Question 9

Write down the three 95% confidence intervals for the population mean of `tvhours` μ_{tv} you've computed in this problem set. Do this by replacing X, Y, A, B, P, and Q with the appropriate values you've computed.

When you are done, make sure all the `|` in the table still line up so your results print out in a table!

CI construction method	lower value	upper value
Using boostrap: 95% rule	X	Y
Using boostrap: percentile rule	A	B
Using mathematical formula	P	Q

Question 10

In your opinion, would you say these three confidence intervals are similar?

Answer: