# Week 6 synchronous class and video code (complete)

Prof. Caetano

2021-02-21

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## Warning: package 'tibble' was built under R version 4.0.3
```

```
## Warning: package 'readr' was built under R version 4.0.3
```

## Synchronous class

### [Case study 1]

Consider the car insurance claims paid by an insurer over a certain time period in the `auto_claims_population.csv` data set. You will use this data for your problem set also. Assume this data set includes *ALL* claims paid (in USD) to claimants 50 years of age and older in a specific year. In other words, it represents a 'population' of car insurance claims in that year.

**(a) Select 1000 samples of size 20 from the population of claims stored in the `auto_claims_population.csv` data set (each sample is taken without replacement, so there are no repeated observations within each sample). Compute the mean age of claimants for each sample and produce appropriate summaries of the simulated sample means.**

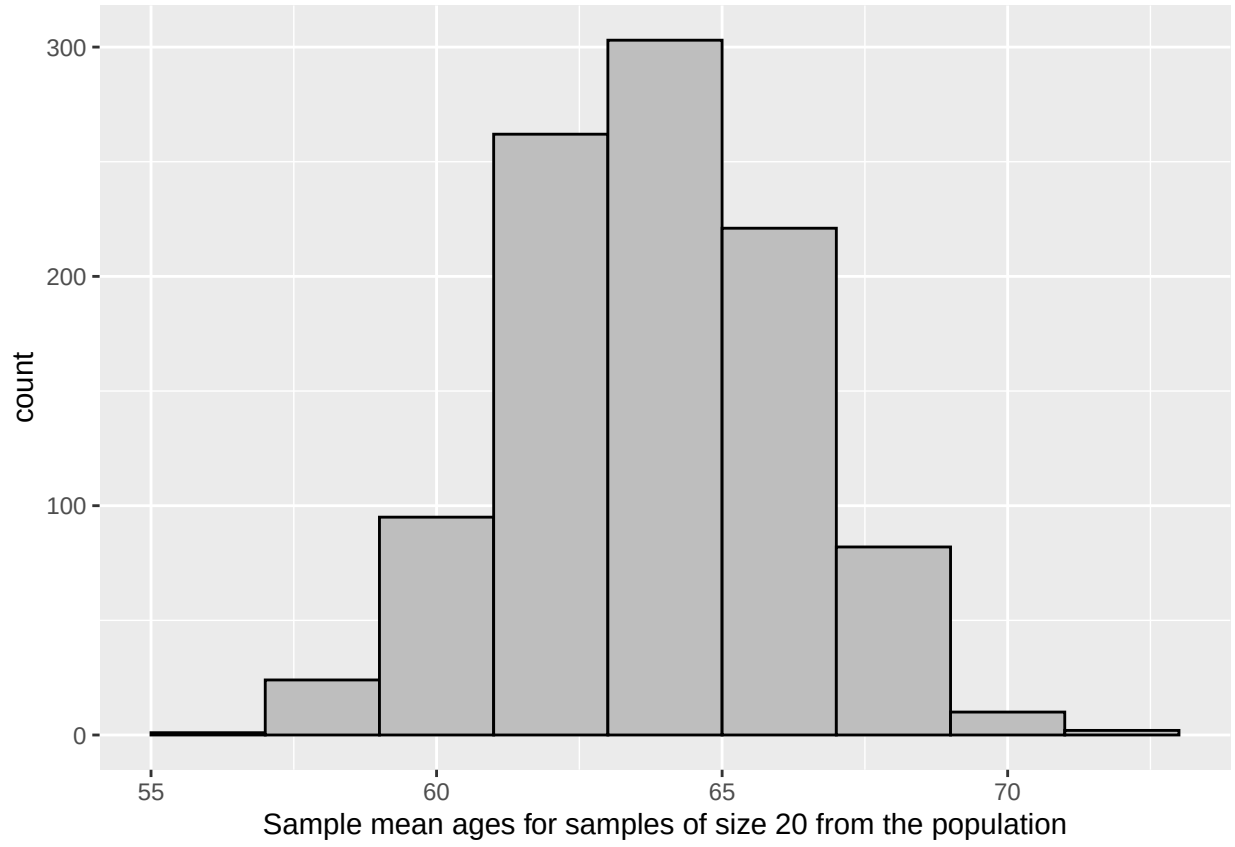
```
AutoClaimsPop <- read_csv("auto_claims_population.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   STATE = col_character(),
##   CLASS = col_character(),
##   GENDER = col_character(),
##   AGE = col_double(),
##   PAID = col_double()
## )
```

```r
set.seed(246)
n <- 20
repetitions <- 1000
sim20 <- rep(NA, repetitions)

for (i in 1:repetitions)
{
  new_sim <- AutoClaimsPop %>% sample_n(size=20, replace=FALSE)
  sim_mean <- new_sim %>%
    summarize(mean(AGE)) %>%
    as.numeric()

  sim20[i] <- sim_mean
}
sim20 <- tibble(means = sim20)
sim20 %>% ggplot(aes(x = means)) +
  geom_histogram(binwidth = 2, colour = "black", fill = "grey") +
  labs(x="Sample mean ages for samples of size 20 from the population")
```




```r
summarise(sim20,
          min=min(means),
          mean = mean(means),
          median = median(means),
          max=max(means),
          sd = sd(means),
```

```
          n=n())
```

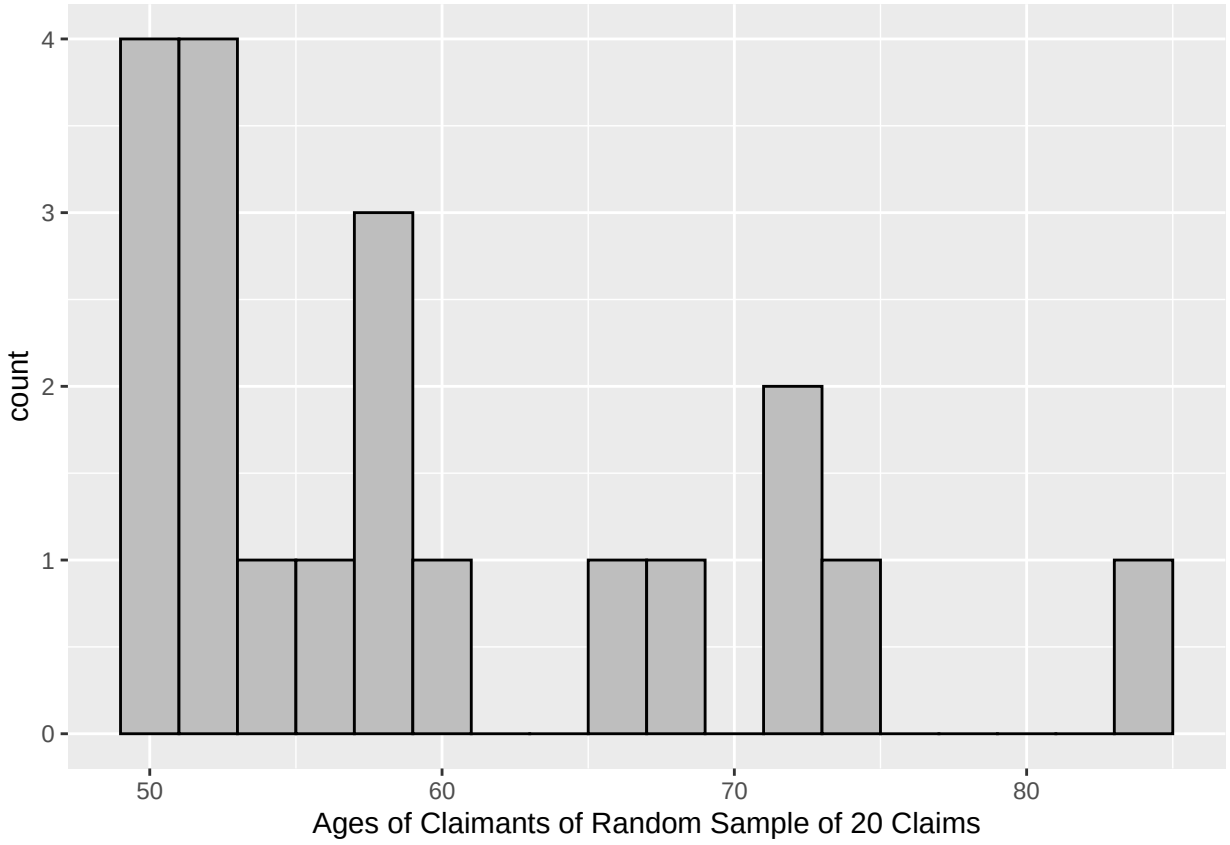
```
## # A tibble: 1 x 6
##     min  mean median   max    sd     n
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <int>
## 1  56.4  63.8   63.8  71.8  2.39  1000
```

**(b) Now suppose we only had data for ONE random sample of 20 car insurance claims, and that these 20 observations are stored in ages20.**

```
set.seed(321)
ages20 <- tibble(age=sample(AutoClaimsPop$AGE,size = 20, replace=FALSE))
glimpse(ages20)
```

```
## Rows: 20
## Columns: 1
## $ age <dbl> 75, 72, 68, 52, 50, 59, 52, 57, 53, 73, 84, 51, 51, 51, 61, 52,...
```

```
ages20 %>% ggplot(aes(x = age)) +
  geom_histogram(binwidth = 2, colour = "black", fill = "grey") +
  labs(x="Ages of Claimants of Random Sample of 20 Claims")
```

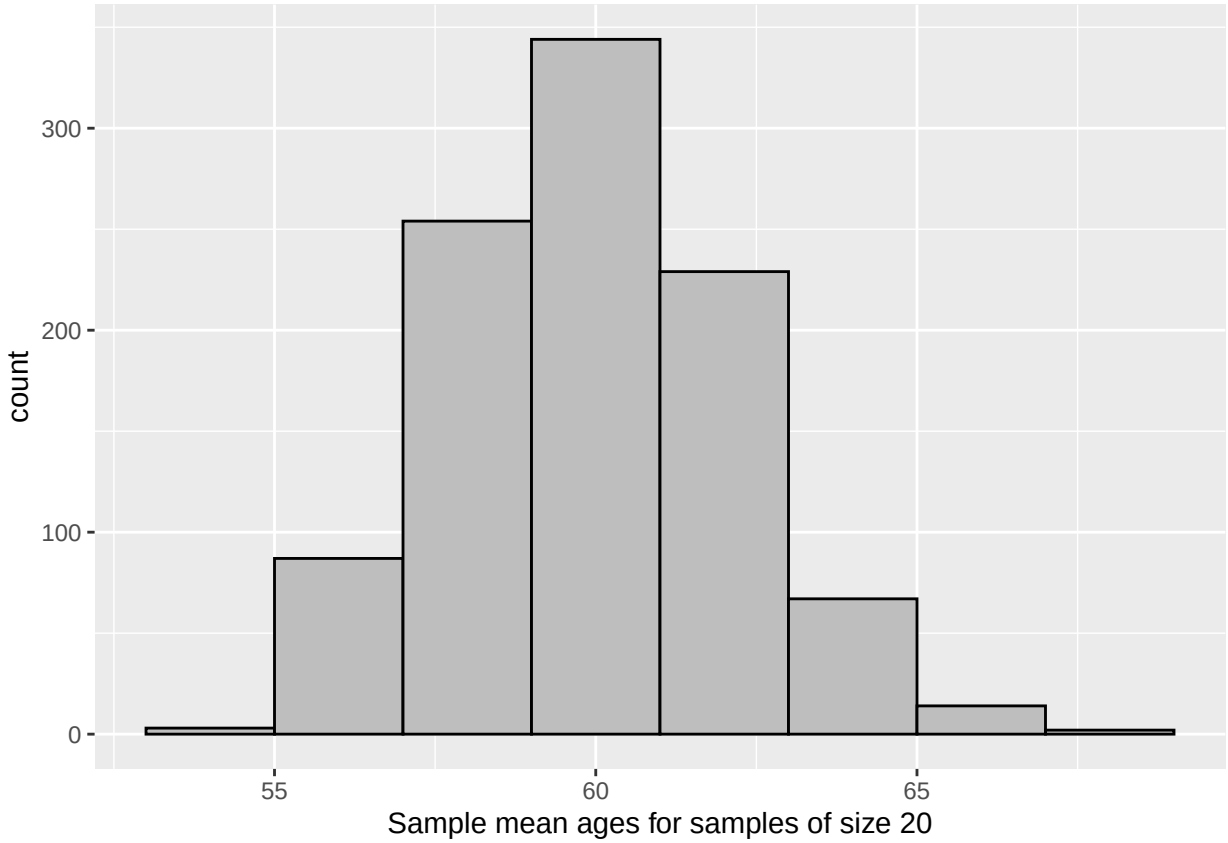


```
summarise(ages20,
          min=min(age),
          mean = mean(age),
          median = median(age),
          max=max(age),
          sd = sd(age),
          n=n())
```

```
## # A tibble: 1 x 6
##     min  mean median   max    sd     n
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <int>
## 1    50  60.0   57.5    84  9.88    20
```

**Use R to take 1000 bootstrap samples from the ages of the claimants of the claims sampled and stored in ages20. Compute the mean age of claimants for each bootstrap sample of claims and produce appropriate summaries of the bootstrap sample means.**

```
set.seed(246)
boot_means <- rep(NA, 1000)  # where we'll store the bootstrap means
sample_size <- 20
for (i in 1:1000)
{
  boot_samp <- ages20 %>% sample_n(size = sample_size, replace=TRUE)
  boot_means[i] <- as.numeric(boot_samp %>% summarize(mean(age)))
}
boot_means <- tibble(means=boot_means)
boot_means %>% ggplot(aes(x = means)) +
  geom_histogram(binwidth = 2, colour = "black", fill = "grey") +
  labs(x="Sample mean ages for samples of size 20")
```




```
summarise(boot_means,
          min=min(means),
          mean = mean(means),
          median = median(means),
          max=max(means),
          sd = sd(means),
          n=n())
```

```
## # A tibble: 1 x 6
##     min  mean median   max    sd     n
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <int>
## 1  53.8  60.0   60.0    68  2.19  1000
```

**(c) What distribution do the distributions we simulated in (a) and (b) both estimate? Comment on the similarities and differences in the estimates we obtained.**

*Both distributions estimate the same sampling distribution - the* sampling distribution of the sample mean age of claimants based on a random samples of 20 claims. *So, it is not surprising that both estimated distributions are similar in terms of shape, centre and spread. They are both approximately symmetric and unimodal, their means are relatively close (63.8 and 60 years respectively) and their standard deviations just differ a little bit (2.39 vs 2.19 years).*

*The estimate of the sampling distribution in (a) was obtained by sampling directly from the population; whereas the estimate of the sampling distribution in (b) was obtained by resampling (i.e., taking bootstrap samples) from a specific random sample of 20 claims. If the sample is not representative of the population of*

*claims, then the estimate of the sampling distribution based on bootstrap samples from that non-representative sample will not reflect the sampling distribution of mean ages very well.*

## [Case study 2]

In this question we will look at data from the Child Health and Development Studies. Our data are adapted from the `Gestation` data set in the `mosaicData` package. Birth weight, date, and gestational period were collected as part of the Child Health and Development Studies in 1961 and 1962 for a sample of 400 mothers who had babies in these two years. Information about the baby's parents—age, education, height, weight, and whether the mother smoked—was also recorded.

We will find confidence intervals for parameters related to the distribution of the mother's age, which for this sample is stored in the variable `age`.

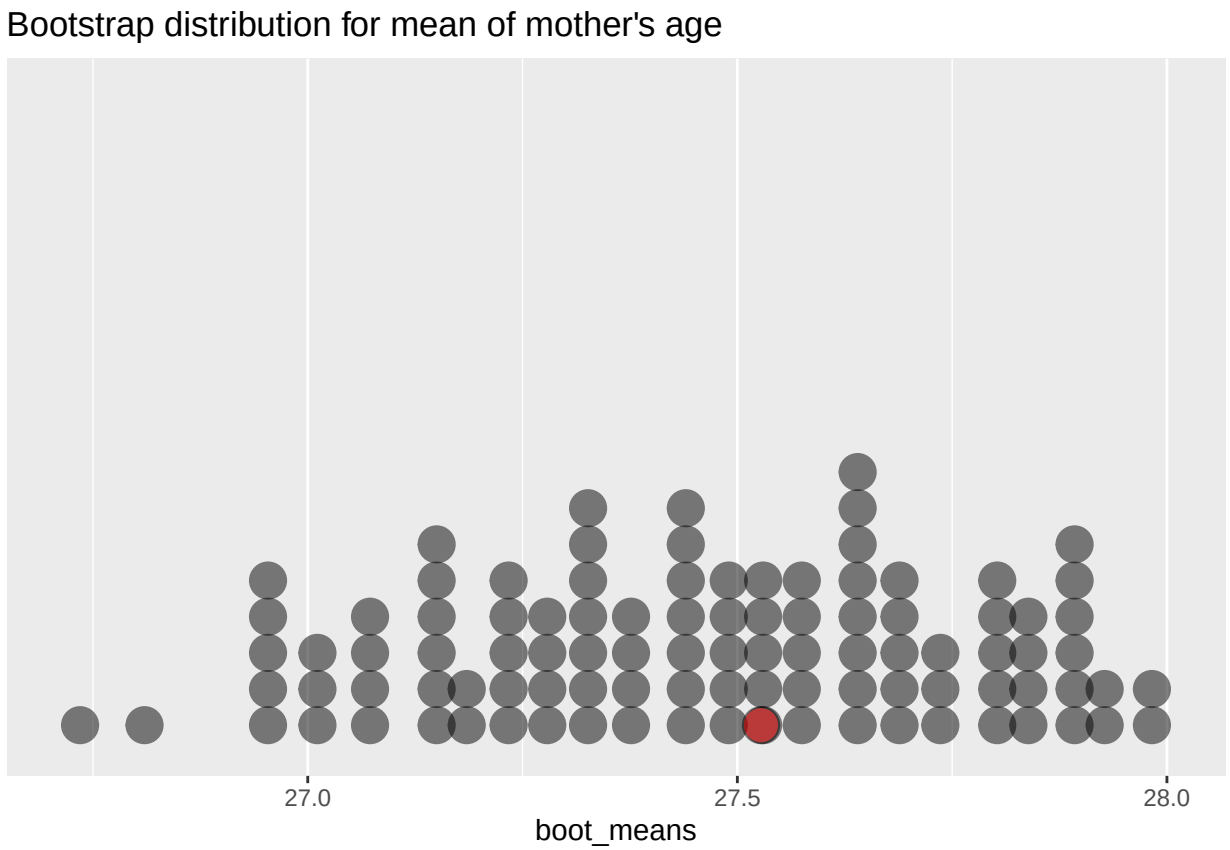
```
Gestation <- read_csv("gestation.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   drace = col_character()
## )
## i Use `spec()` for the full column specifications.
```

**(a) Suppose we are interested in how means of random samples of n=400 mothers vary across possible samples of 400 mothers we could take from the population. Explain why it is not possible to use these data (i.e., 'Gestation') to estimate this like we did in Case Study 1, question a).**

*In Case Study 1, question a, we estimated the sampling distribution of the sample mean based on samples of n=20 observations by repeatedly drawing samples of size 20 from the population of claims, which were available in the 'auto_claims_population.csv' data set. The data for this question are on a sample of mothers who participated in the Child Health and Development Studies in 1961 and 1962. The n=400 ages here, then, represent ages for a* sample *of mothers, not the entire population. Therefore, we cannot repeatedly take samples of 400 observations from the population. We do not have data on the entire population.*

**(b) The plot below shows the bootstrap distribution for the mean of mother's age for 100 bootstrap samples. The red dot is the estimate of the mean for the first bootstrap sample, and the grey dots are the estimates of the mean for the other 99 bootstrap samples.**




```
## # A tibble: 1 x 6
##     min  mean median   max    sd     n
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <int>
## 1  26.7  27.5   27.5  28.0 0.299   100
```

i) Explain how the value of the red dot is calculated.

*The red dot is the mean of mother's age for one bootstrap sample. The bootstrap sample is obtained by taking a random sample with replacement, from the original sample data, with the same number of observations as the original sample.*

(ii) Using this plot, estimate a 90% confidence interval for the mean of mother's age.

*The 90% confidence interval ranges from approximately the 5th largest data point to the 95th largest data point. This interval will be from a value a little below 27.0 to a value a little below 28.0.*

**(c)**

(i) Use R to find a 99% bootstrap confidence interval for the mean of mother's age. Use 2000 bootstrap samples. *NOTE:* More bootstrap samples is better, but if you find your analysis times out or takes too long in RStudio , you can try using 1000 bootstrap samples instead.

```
repetitions <- 2000
boot_means <- rep(NA, repetitions)  # where we'll store the bootstrap means
sample_size <- as.numeric(Gestation %>% summarize(n()))
set.seed(50)
for (i in 1:repetitions)
{
  boot_samp <- Gestation %>% sample_n(size = sample_size, replace=TRUE)
  boot_means[i] <- as.numeric(boot_samp %>% summarize(mean(age)))
}
quantile(boot_means,c(.005,.995))
```

```
##     0.5%    99.5%
## 26.78249 28.17759
```

(ii) Explain why the interpretation *"We are 99% sure that the true mean of a mother's age at the time this sample was taken is between 26.8 and 28.2 years."* is *INCORRECT*. What is a correct interpretation?

*The true mean age of mothers in 1961/62 is unknown, but it's not random. In other words, it's a fixed, but unknown constant. Therefore it either is or isn't in this interval (i.e., the chance is either 0% or 100%). We just do not know either way.*

*We can conclude that we are 99%* confident *that the true mean mother's age in 1961/62 was between 26.8 and 28.2 years. We are confident in this because the method we used to obtain the interval will produce intervals that do include the true value of the parameter of interest for 99% of the possible samples we could take.*

**(d)**

(i) Use R to find a 95% bootstrap confidence interval for the *median* of mother's age. Use 2000 bootstrap samples. *NOTE:* More bootstrap samples is better, but if you find this times out or takes too long in RStudio , try using 1000 bootstrap samples instead.

```
repetitions <- 2000;
boot_medians <- rep(NA, repetitions)
sample_size <- as.numeric(Gestation %>% summarize(n()))
set.seed(579)
for (i in 1:repetitions)
{
  boot_samp <- Gestation %>% sample_n(size = sample_size, replace=TRUE)
  boot_medians[i] <- as.numeric(boot_samp %>% summarize(median(age)))
}
quantile(boot_medians,c(0.025,0.975))
```

```
##  2.5% 97.5%
##    26    27
```

(ii) Write an interpretation of this interval.

*The 95% bootstrap confidence interval for the median of mother's age is (26, 27). We are 95% confident that the median age of all mothers in 1961/62 is between 26 and 27 years based on these data.*

# Video code

## Setting up the flights data

```
#install.packages("nycflights13")
library(tidyverse)
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 4.0.3
```

```
# Save data in a data frame called SF
SF <- flights %>% filter(dest=="SFO" & !is.na(arr_delay))
dim(SF)
```

```
## [1] 13173    19
```

## Summarise the flights data

```
SF %>% summarise(
  mean_delay = mean(arr_delay),
  median_delay = median(arr_delay),
  max_delay = max(arr_delay))
```

```
## # A tibble: 1 x 3
##   mean_delay median_delay max_delay
##        <dbl>        <dbl>     <dbl>
## 1       2.67           -8      1007
```

```
# We'll save the population mean,
# so we can use it later on
population_mean <- SF %>%
  summarize(population_mean_delay =
              mean(arr_delay))

population_mean <-
  as.numeric(population_mean)
```

## Take a sample

```
# sample of 25 flights from our population
# by default, replace = FALSE (i.e. sampling without replacement)
sample25 <- SF %>% sample_n(size=25, replace = FALSE)
```

**What is the difference between `sample()` and `sample_n()`?**

```
sample(c("H", "T"), probs=c(0.5, 0.5),
       size=10, replace=TRUE)
sample(1:6, replace=FALSE`)
```

The `sample()` function samples elements from a **vector**, with or without replacement

```
# Create our sample
SF %>% sample_n(size=25, replace=FALSE)
```

The `sample_n()` samples rows (observations) from a data frame, with or without replacement

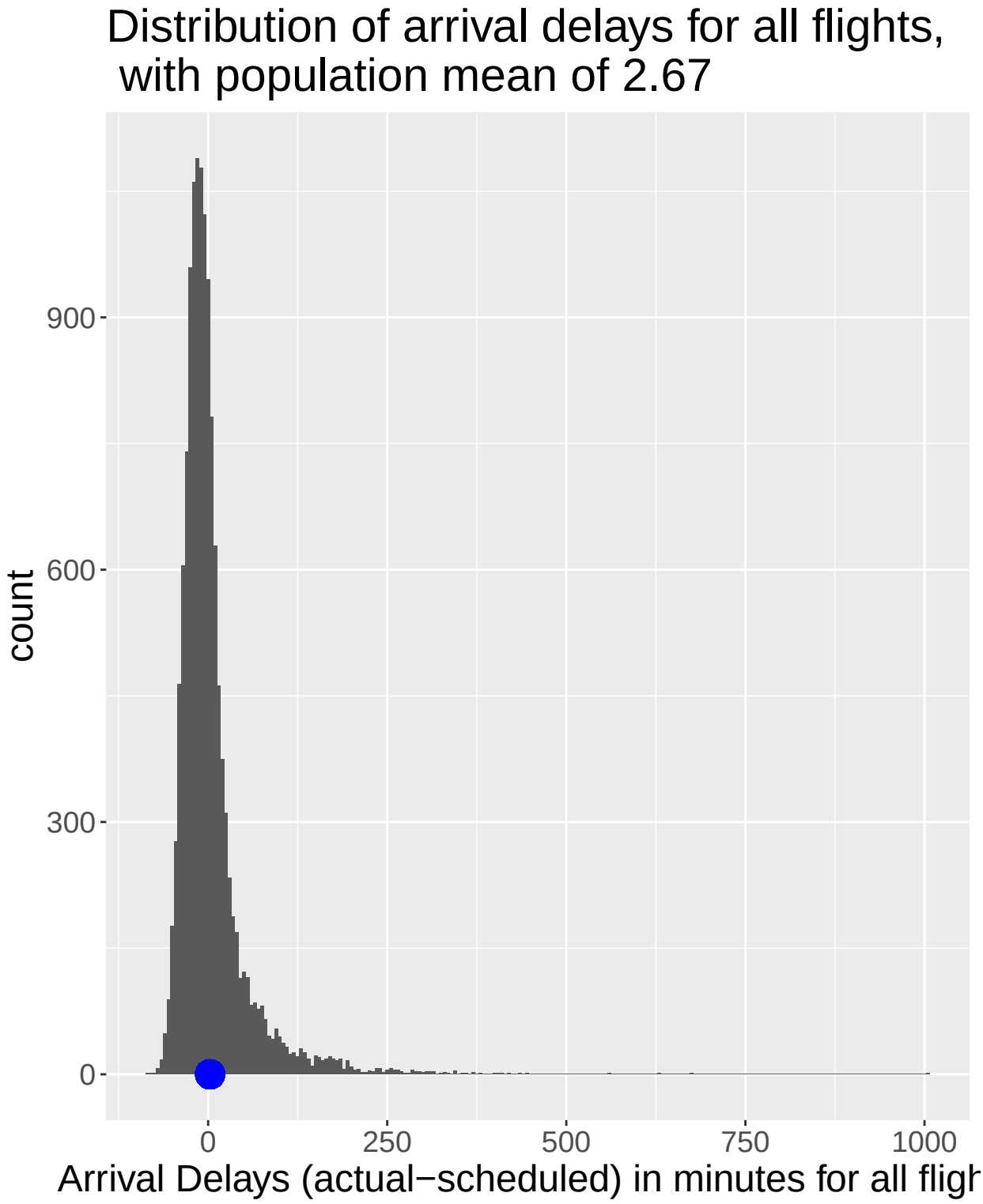
### Calculate summary values for this sample

```
sample25 %>% summarise(mean_delay = mean(arr_delay),
                median_delay = median(arr_delay),
                max_delay = max(arr_delay))
```

```
## # A tibble: 1 x 3
##   mean_delay median_delay max_delay
##        <dbl>        <dbl>     <dbl>
## 1        1.8          -10       208
```

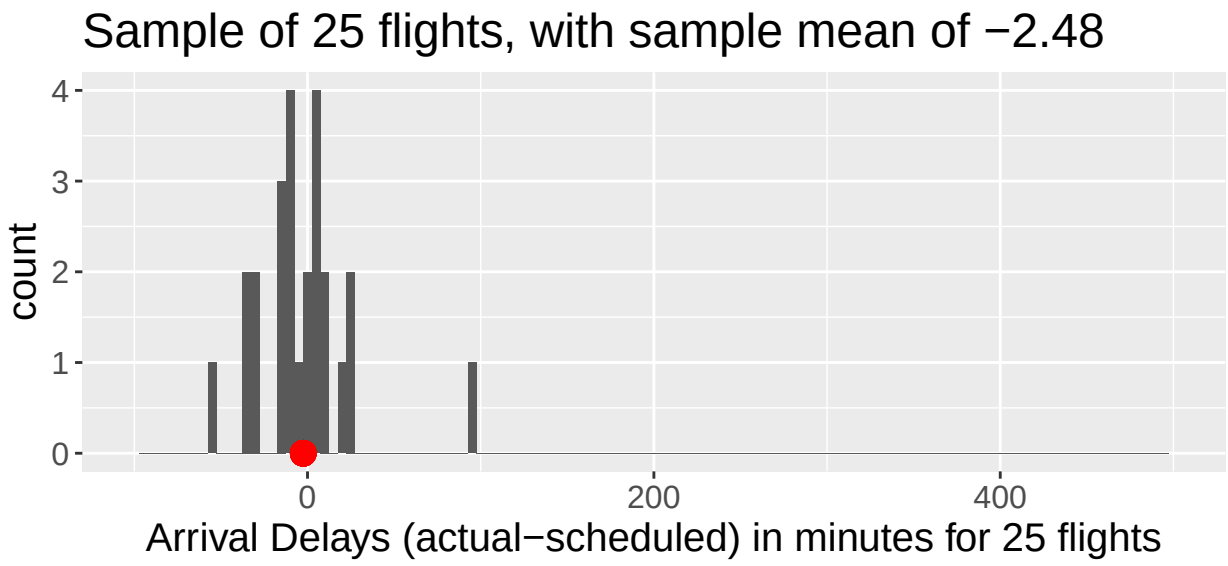
### Looking at multiple samples of size n=25

```
## Warning: Use of `SF$arr_delay` is discouraged. Use `arr_delay` instead.
```



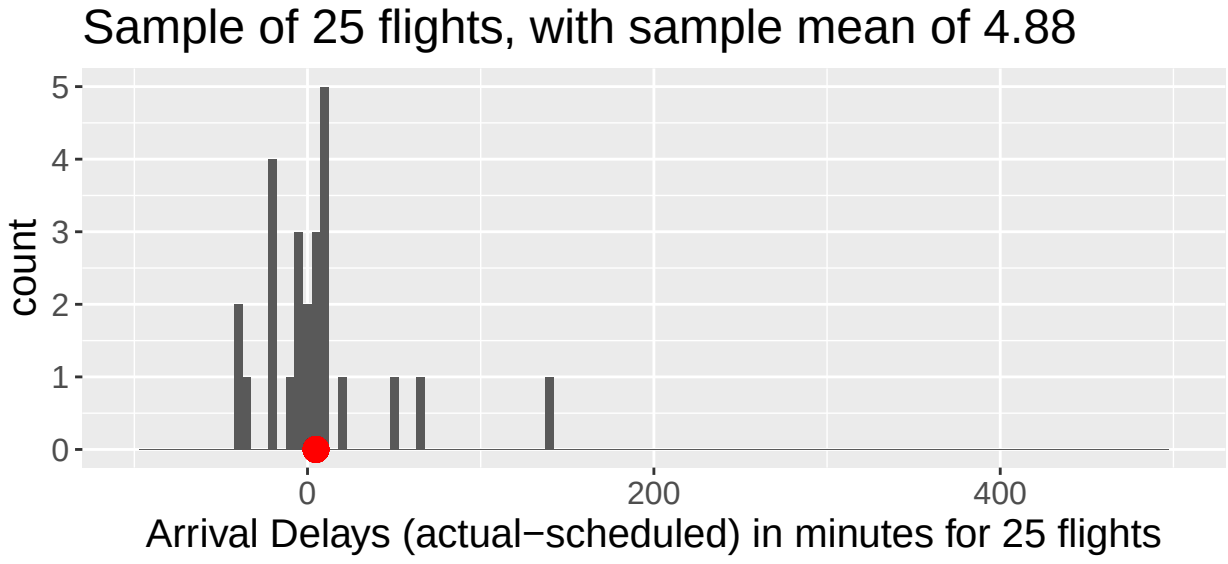

```
## Warning: Use of ‘d25$arr_delay‘ is discouraged. Use ‘arr_delay‘ instead.
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



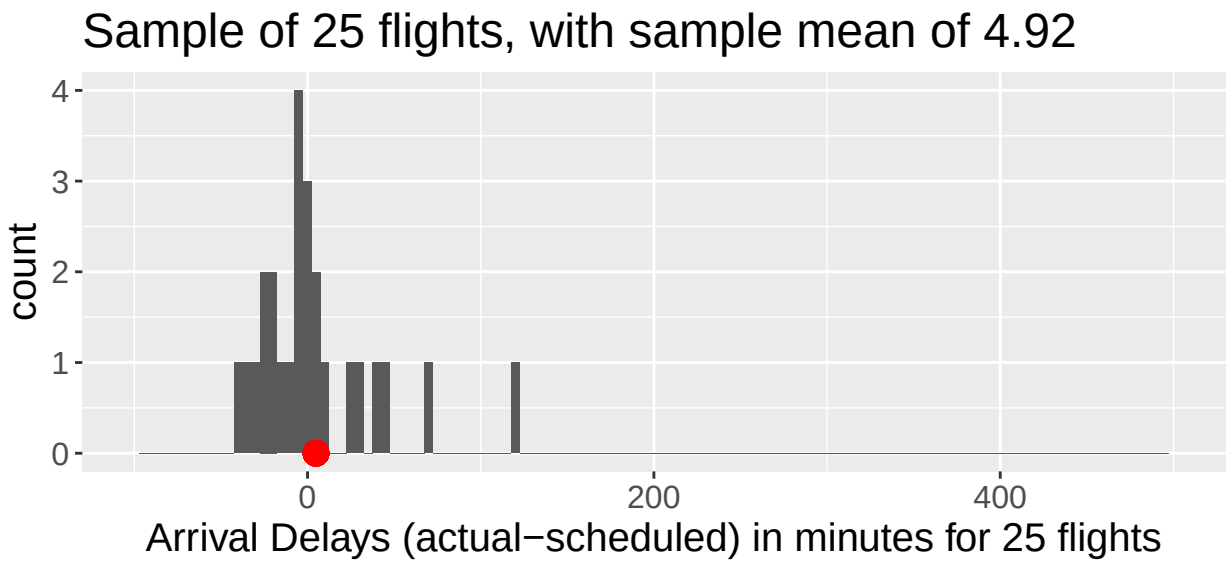

```
## Warning: Use of ‘d25$arr_delay‘ is discouraged. Use ‘arr_delay‘ instead.

## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
## Warning: Use of ‘d25$arr_delay‘ is discouraged. Use ‘arr_delay‘ instead.

## Warning: Removed 2 rows containing missing values (geom_bar).
```




### Review: Sampling distributions

Recall, the **sampling distribution** of the mean of `arr_delay` is the distribution of all the values that `mean_delay` could be for random samples of size $n = 25$

To estimate the sampling distribution, let's look at 1000 values of `mean_delay`, calculated from 1000 random samples of size $n = 25$ from our population

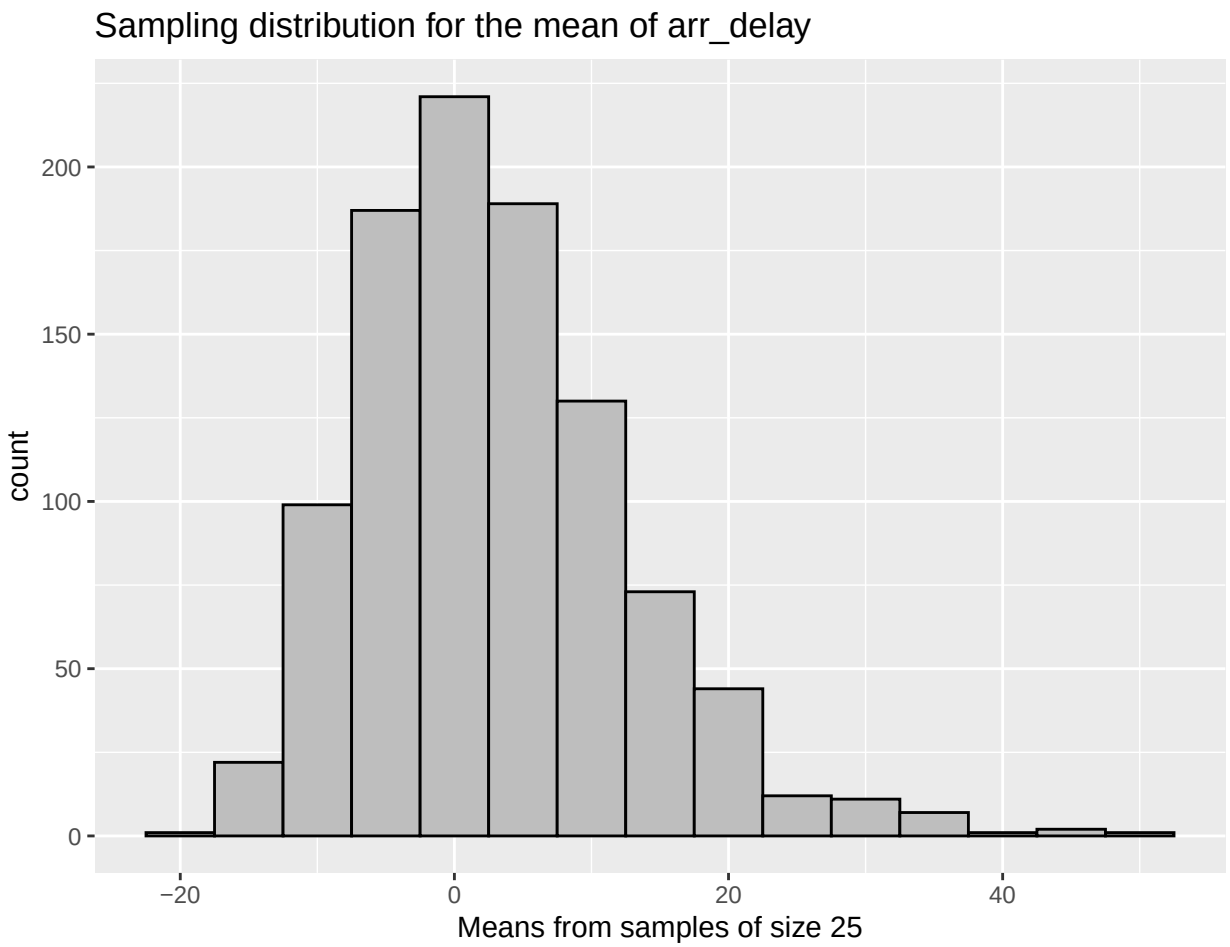
```
sample_means <- rep(NA, 1000) # where we'll store the means

for(i in 1:1000){
  sample25 <- SF %>% sample_n(size=25)
  sample_means[i] <- as.numeric(sample25 %>%
                        summarize(mean(arr_delay)))
}

sample_means <- tibble(mean_delay = sample_means)
```

### Sampling distribution of the mean

```
ggplot(sample_means, aes(x=mean_delay)) +
  geom_histogram(binwidth=5, color="black", fill="gray") +
  labs(x="Means from samples of size 25",
  title="Sampling distribution for the mean of arr_delay")
```

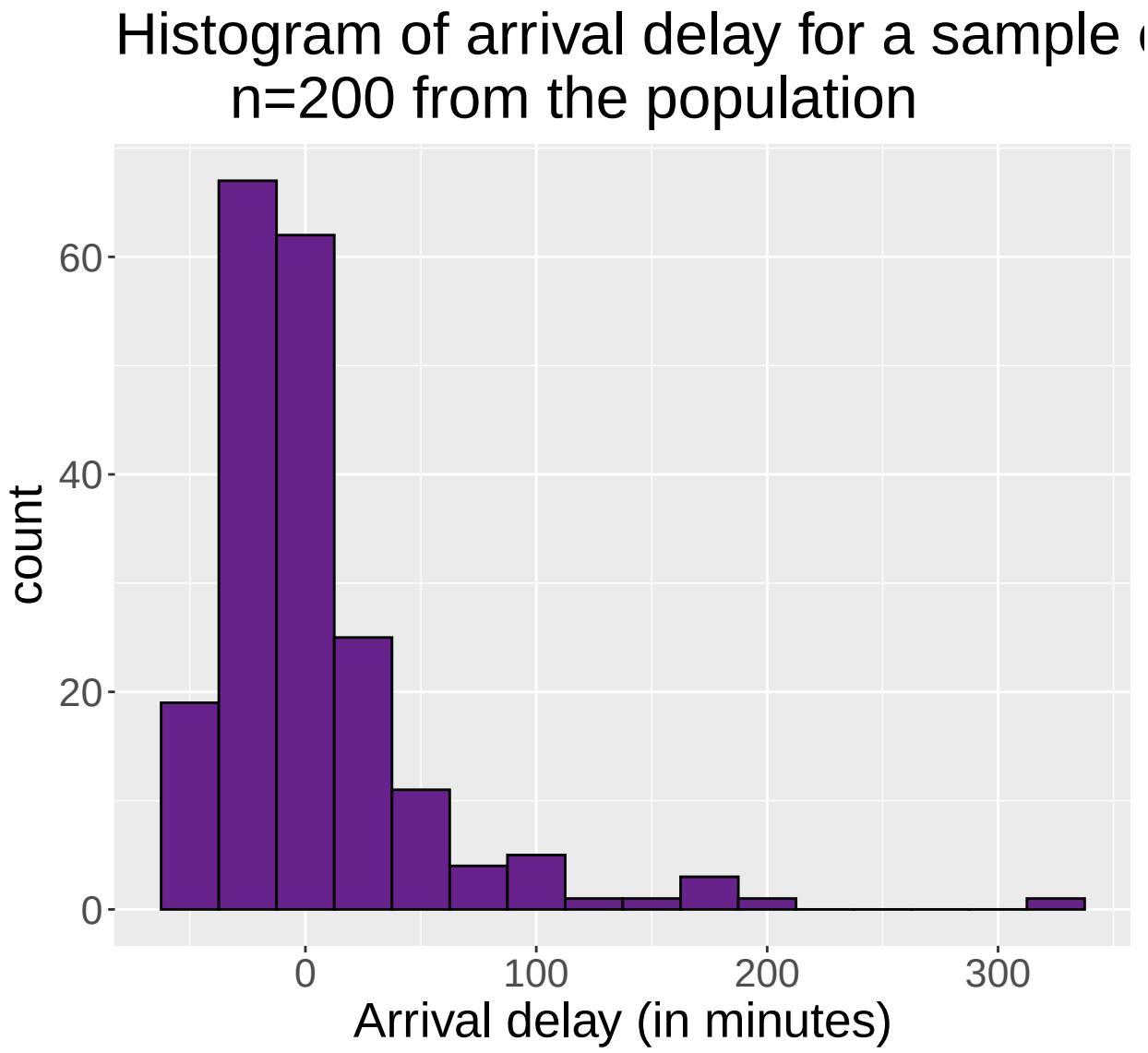



**3 histograms for question prompt**

## Bootstrapping with R

Suppose we do not observe the full population, and have only observed **one sample of size 200**

```
observed_data <- SF %>%
  sample_n(size=200)
```

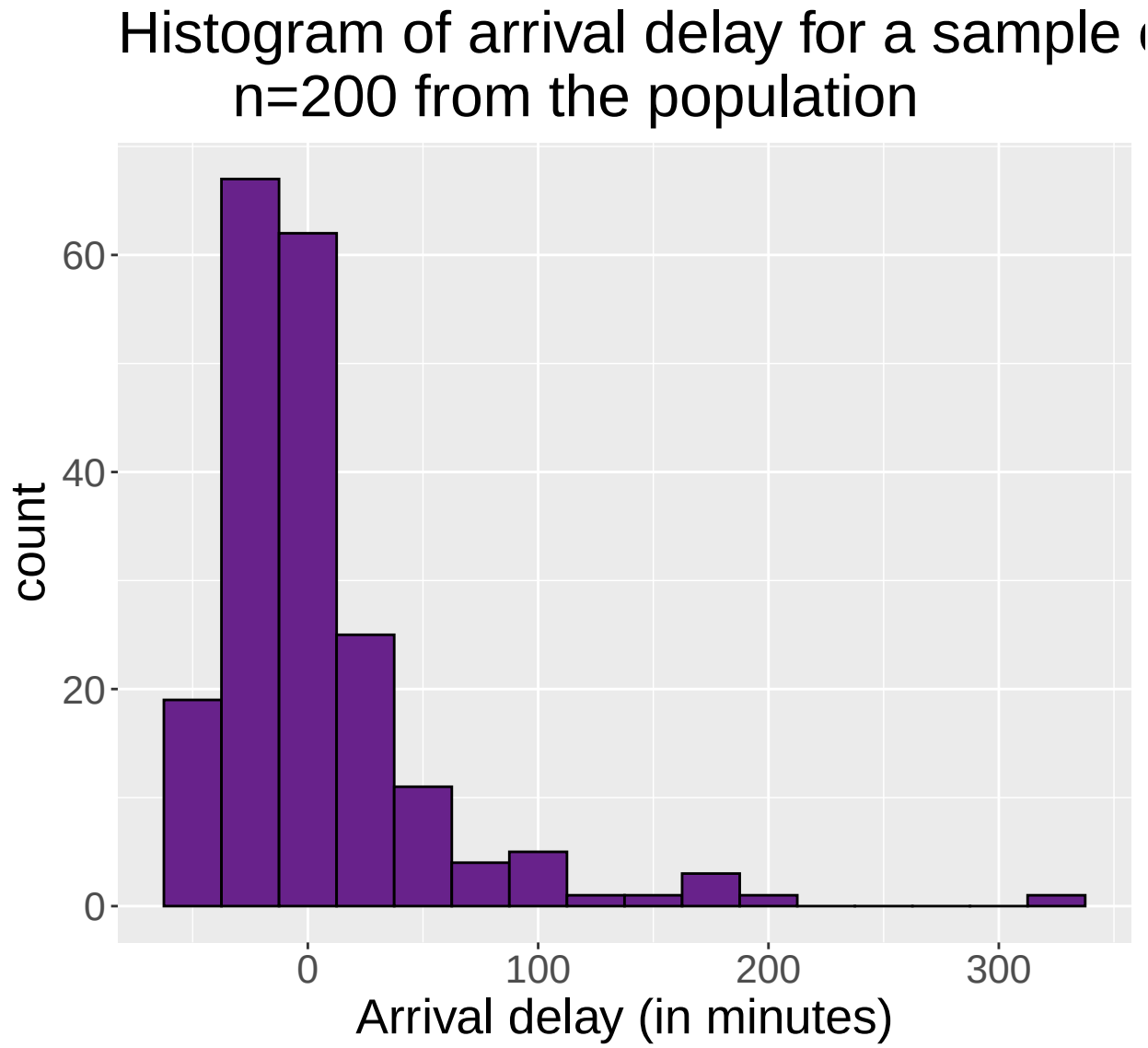



**Let's calculate the mean arrival delay for this sample**

```
obs_mean <- observed_data %>%
  summarize(mean(arr_delay))
as.numeric(obs_mean)
```

```
## [1] 4.485
```

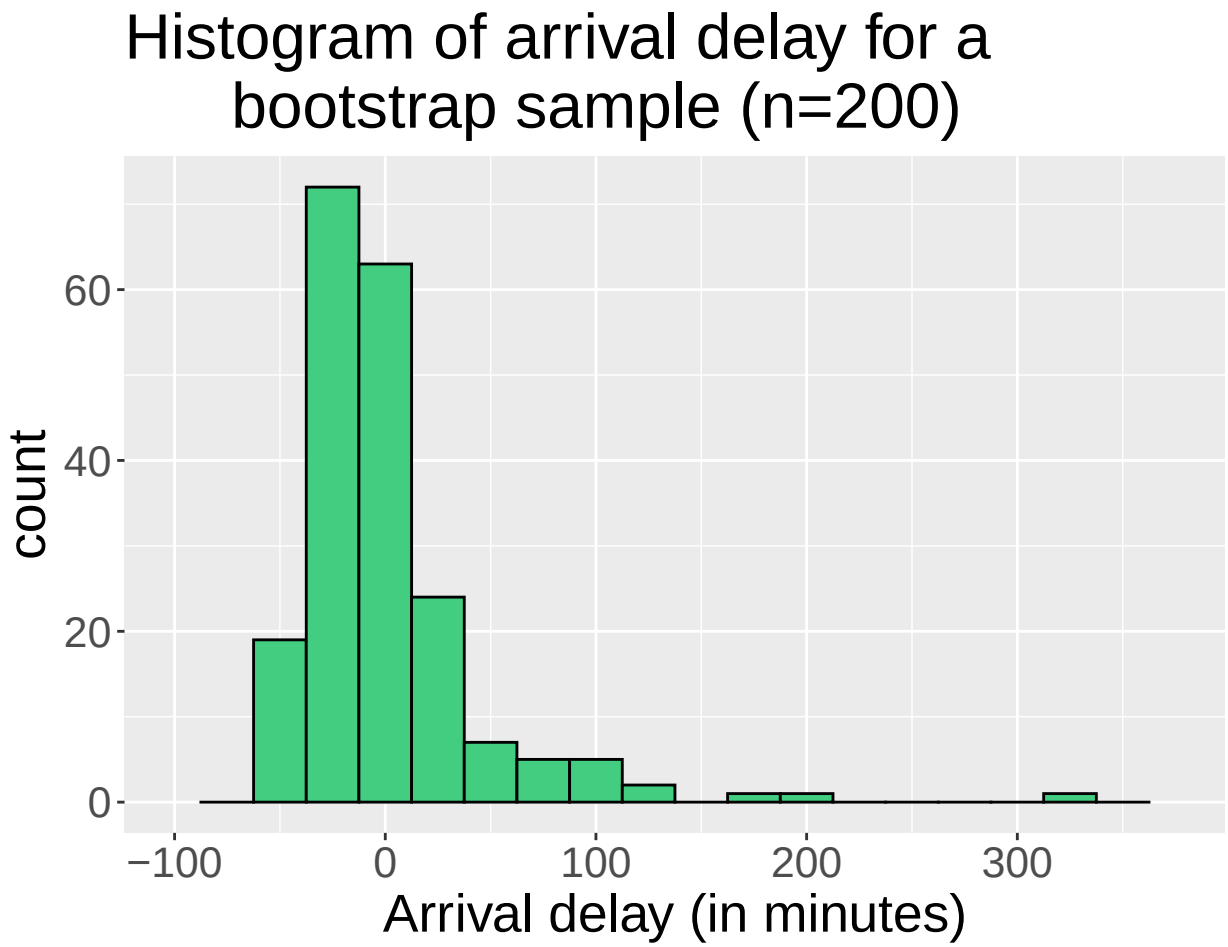
**A bootstrap sample from our observed data**




.pull-left[

```
boot_samp <- observed_data %>%
  sample_n(size=200, replace=TRUE)
```

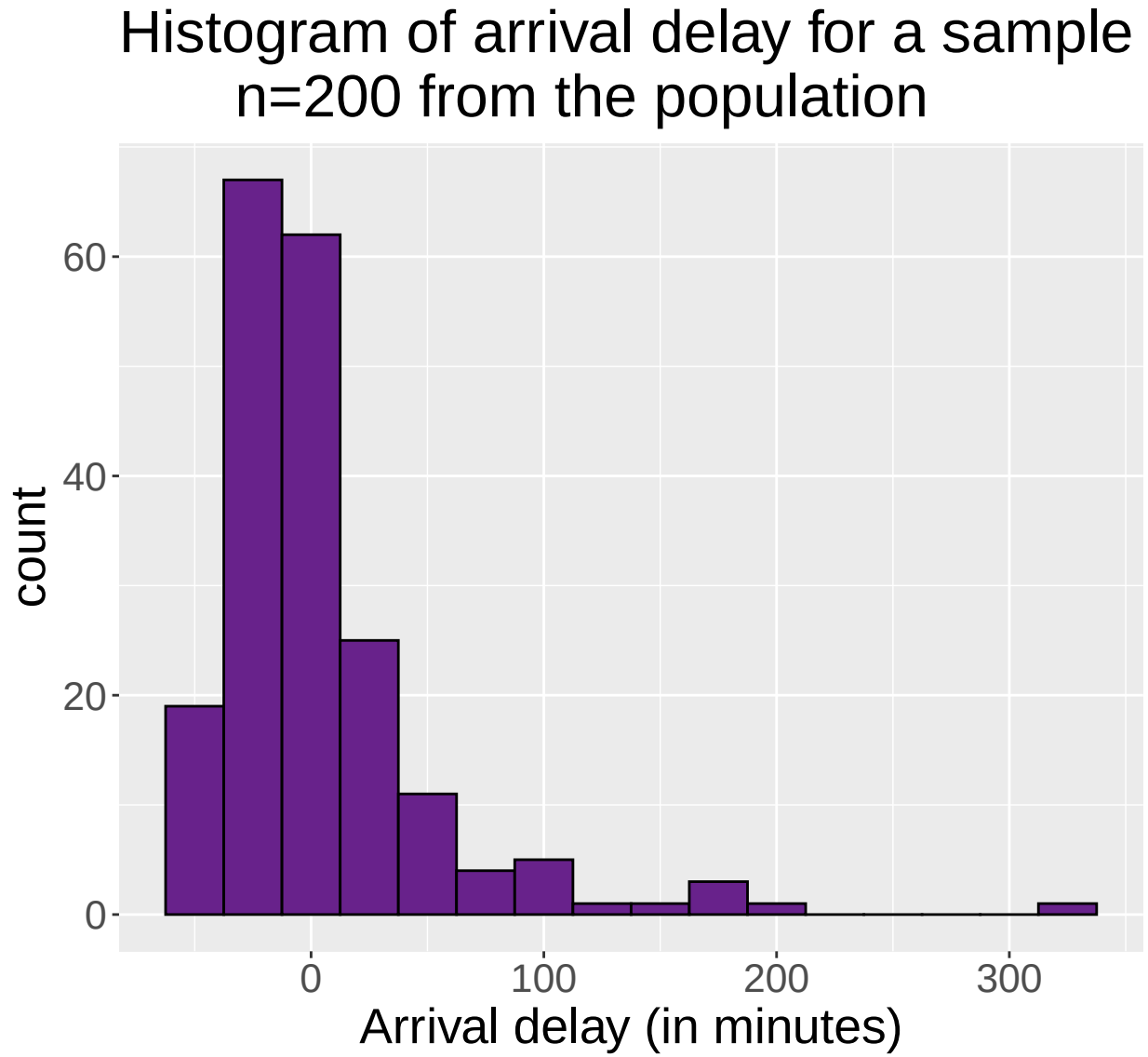
```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
boot_mean <- boot_samp %>%
  summarize(mean_delay =
              mean(arr_delay))
as.numeric(boot_mean)
```

```
## [1] 1.18
```

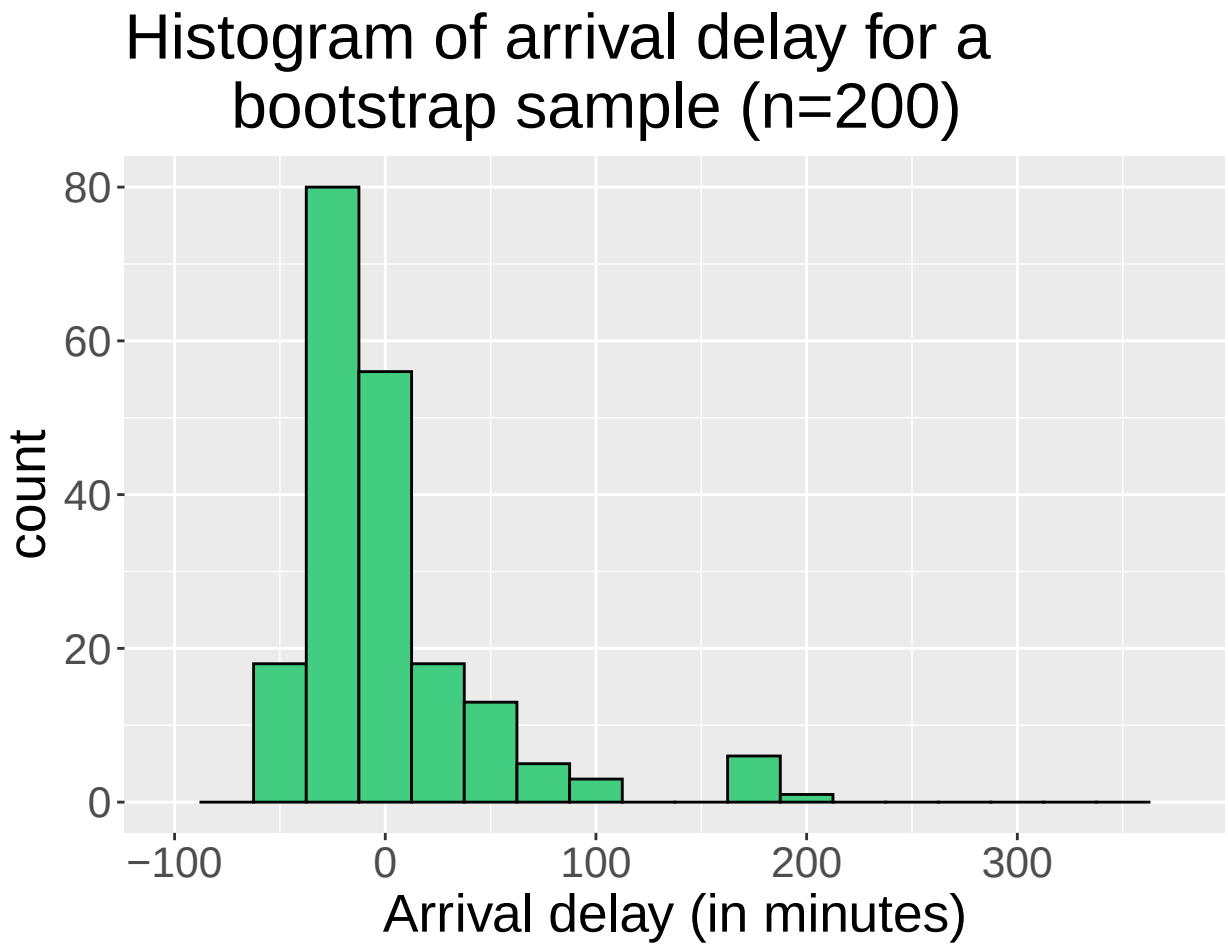
**Another bootstrap sample from our observed data**




.pull-left[

```
boot_samp <- observed_data %>%
  sample_n(size=200, replace=TRUE)
```

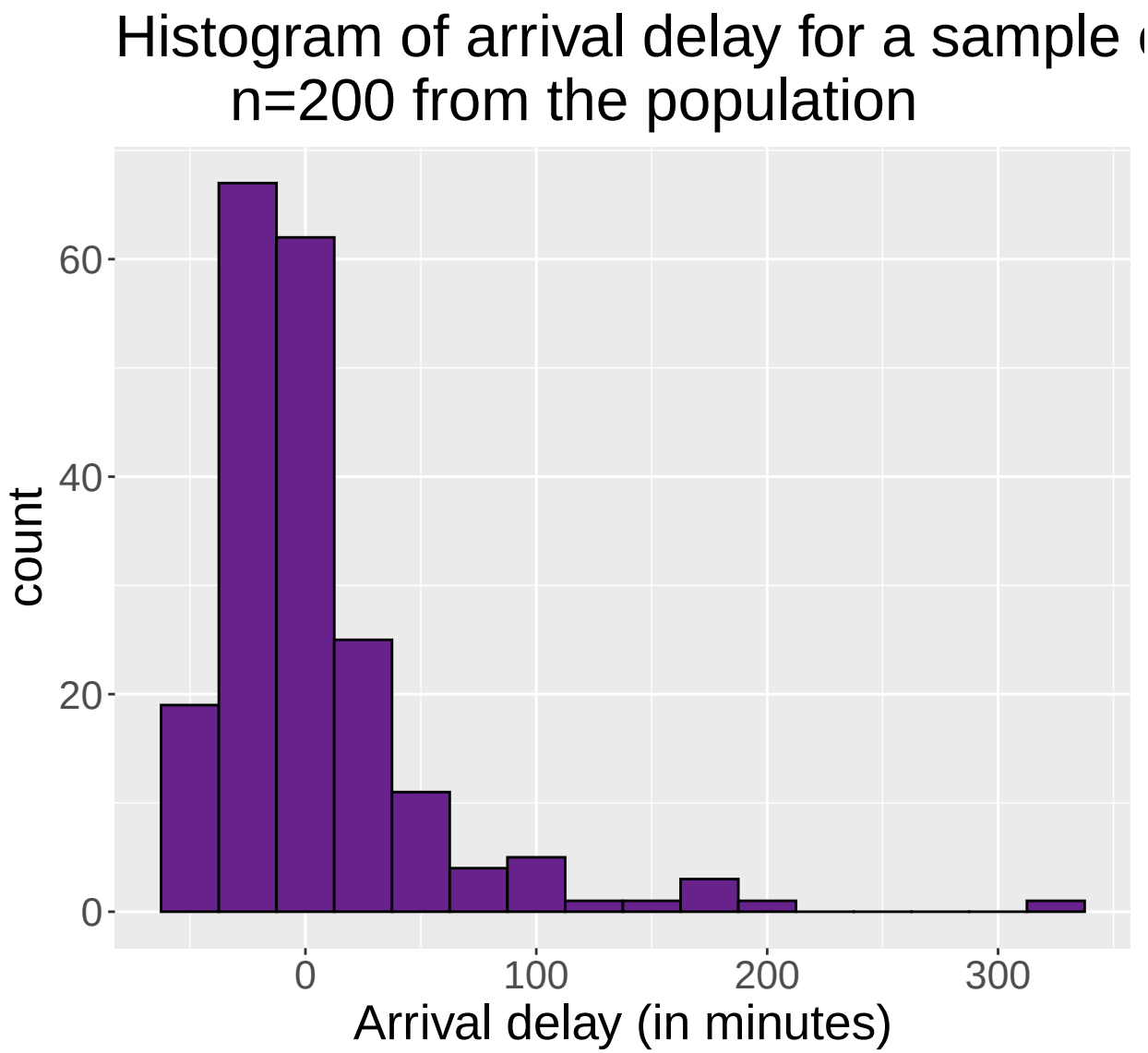
```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
boot_mean <- boot_samp %>%
  summarize(mean_delay =
              mean(arr_delay))
as.numeric(boot_mean)
```

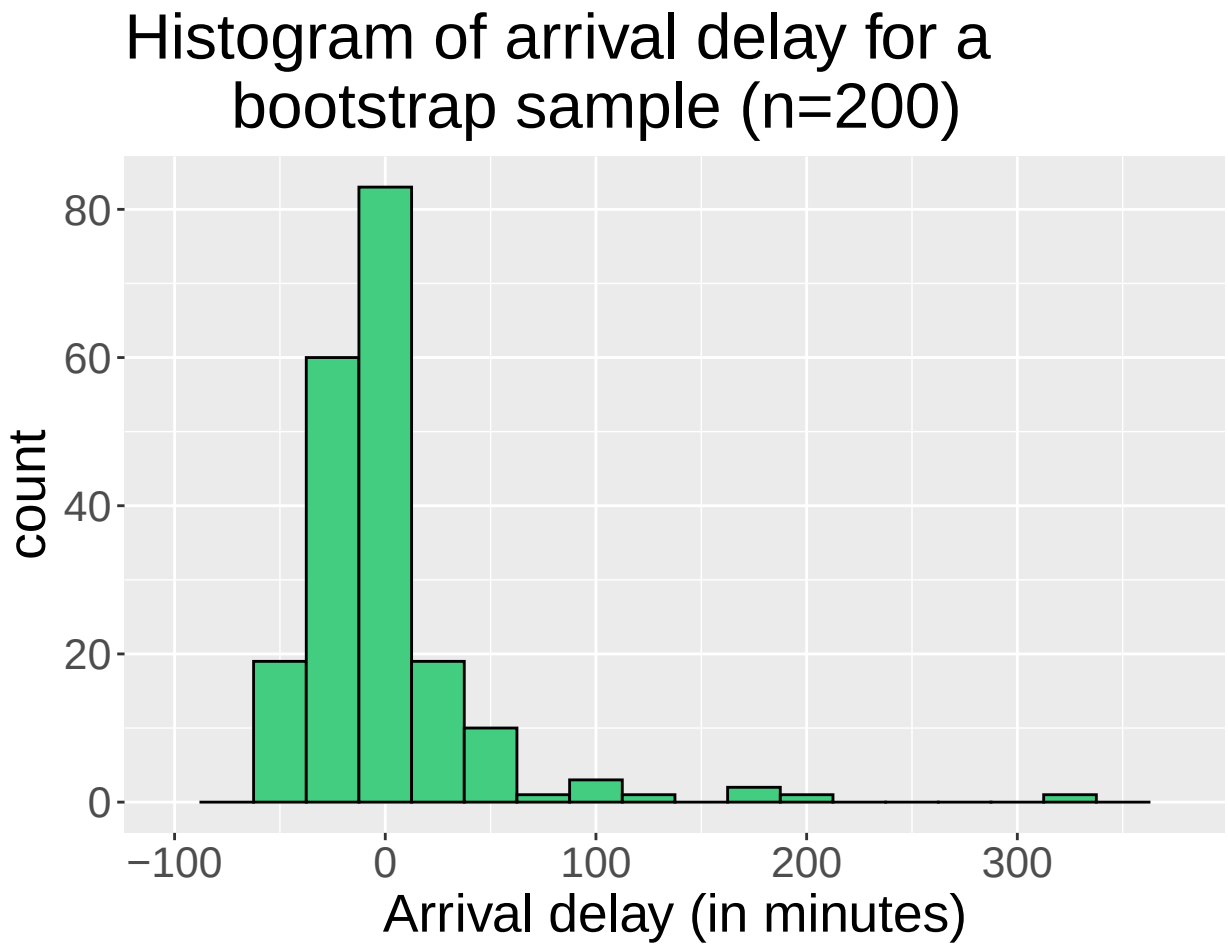
```
## [1] 2.24
```

**And another bootstrap sample...**




```
boot_samp <- observed_data %>%
  sample_n(size=200, replace=TRUE)
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

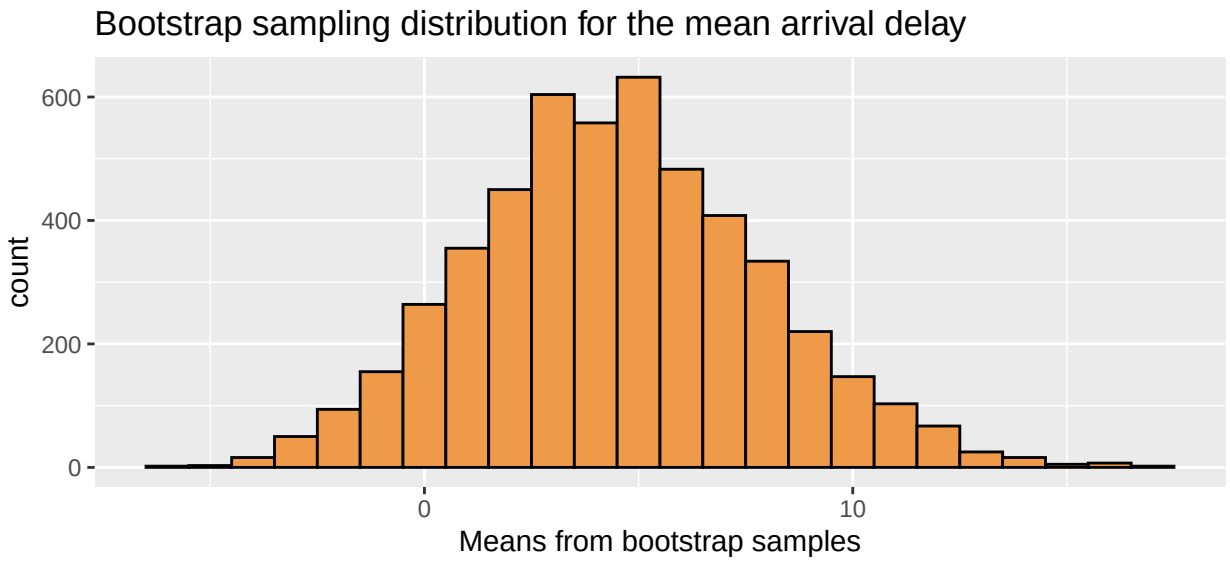



```
boot_mean <- boot_samp %>%
  summarize(mean_delay =
              mean(arr_delay))
as.numeric(boot_mean)
```

```
## [1] -0.15
```

```
boot_means <- rep(NA, 5000) # where we'll store the means
for(i in 1:5000){
  boot_samp <- observed_data %>% sample_n(size=200, replace=TRUE)
  boot_means[i] <-
    as.numeric(boot_samp %>%
                 summarize(mean_delay = mean(arr_delay)))
}
boot_means <- tibble(mean_delay = boot_means)
```

```
ggplot(boot_means, aes(x=mean_delay)) +
  geom_histogram(binwidth=1, fill="tan2", color="black") +
  labs(x="Means from bootstrap samples",
       title="Bootstrap sampling distribution for the mean arrival delay")
```




**Percentiles (quantiles): an extension of quartiles**

For a number $p$ between 0 and 100, the $p$th percentile is the smallest value that is larger or equal to $p$% of all the values

- Median ( $Q_2$ ): 50th percentile
- First quartile ( $Q_1$ ): 25th percentile
- Third quartile ( $Q_3$ ): 75th percentile

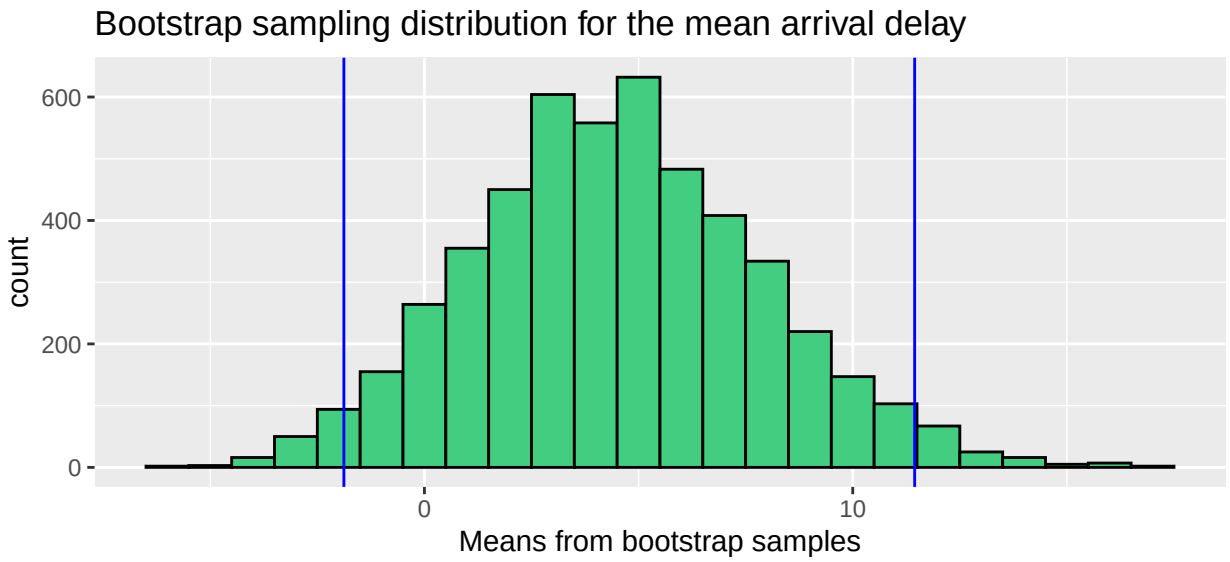
Use the `quantile()` funciton in R to calculate these:

```
# Calculate Q1, median, and Q3
quantile(boot_means$mean_delay, c(0.25, 0.5, 0.75))
```

```
##   25%   50%   75%
## 2.205 4.395 6.695
```

```
# Can also calculate any other percentiles
quantile(boot_means$mean_delay, c(0.025, 0.4, 0.57))
```

```
##      2.5%       40%       57%
## -1.880125  3.520000  4.970000
```




2.5th and 97.5th percentiles:

```
quantile(boot_means$mean_delay,
         c(0.025, 0.975))
```

```
##      2.5%     97.5%
## -1.880125 11.445625
```

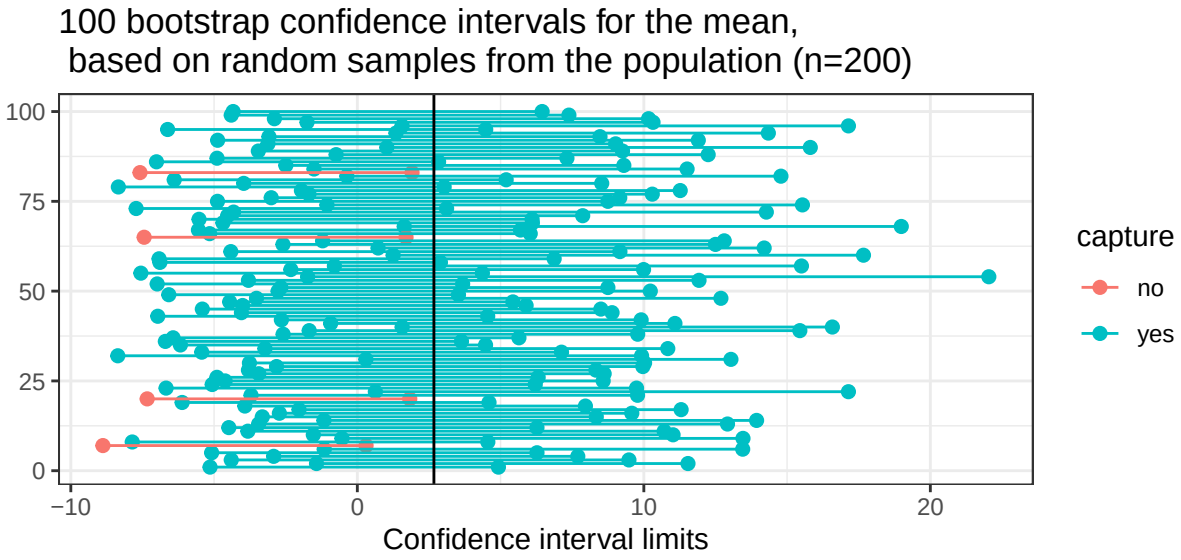
Recall true population mean:

```
as.numeric(population_mean)
```

```
## [1] 2.672892
```

### How often does this procedure give an interval that captures the population mean?

This code is for the curious but NOT something we'll ask you to be able to make yourself. It also take aaaaaages to run, so that is why we have saved the output as a csv for you.


100 bootstrap confidence intervals for the mean,
based on random samples from the population (n=200)
100
75
50
25
0
−10
0
10
20
Confidence interval limits
capture
no
yes