Confidence Intervals II

Nate Wells

Math 141, 3/11/21

Outline

In this lecture, we will...

Outline

In this lecture, we will. . .

- Use bootstrapping as means of creating confidence intervals
- Interpret confidence intervals
- Implement the `infer` package to automate bootstrapped confidence intervals

Section 1

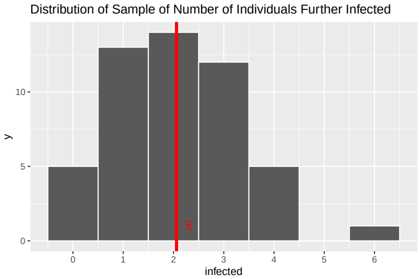
Bootstrapping Confidence Intervals

Reproduction Rate for Covid-19

Researchers are interested in the COVID-19 **reproduction rate** (the average number of individuals each infected person further infects)

- We have a sample of 50 infected individuals and perform contract tracing to determine how many other individuals each infects.

```
##   infected  n
## 1        0  5
## 2        1 13
## 3        2 14
## 4        3 12
## 5        4  5
## 6        6  1
##   mean_infected
## 1          2.06
```



- Goal: Create an interval of plausible values for the reproduction rate.

Bootstrap Reproduction Rate

- To create a 95% confidence interval:

Bootstrap Reproduction Rate

- To create a 95% confidence interval:

1. Create the bootstrap samples:

```
set.seed(121)
bootstrap_samples <- covid %>%
  rep_sample_n(size = 50, replace = TRUE, reps = 5000)
```

Bootstrap Reproduction Rate

- To create a 95% confidence interval:

1. Create the bootstrap samples:

```
set.seed(121)
bootstrap_samples <- covid %>%
  rep_sample_n(size = 50, replace = TRUE, reps =
```

2. Compute bootstrap statistics:

```
bootstrap_stats <- bootstrap_samples %>%
  group_by(replicate) %>%
  summarize(x_bar = mean(infected))
```

Bootstrap Reproduction Rate

- To create a 95% confidence interval:

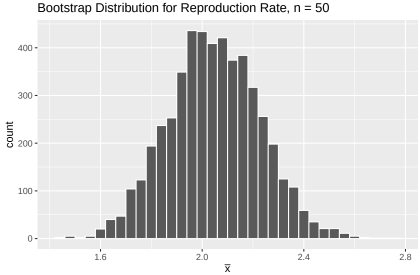
1. Create the bootstrap samples:

```
set.seed(121)
bootstrap_samples <- covid %>%
  rep_sample_n(size = 50, replace = TRUE, reps =
```

2. Compute bootstrap statistics:

```
bootstrap_stats <- bootstrap_samples %>%
  group_by(replicate) %>%
  summarize(x_bar = mean(infected))
```

3. Graph the bootstrap distribution:



Bootstrap Reproduction Rate

- To create a 95% confidence interval:

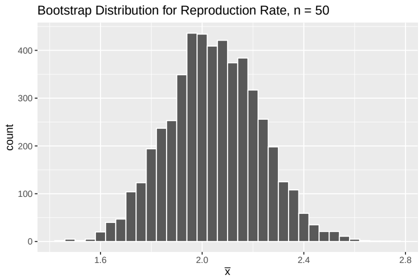
1. Create the bootstrap samples:

```
set.seed(121)
bootstrap_samples <- covid %>%
  rep_sample_n(size = 50, replace = TRUE, reps =
```

2. Compute bootstrap statistics:

```
bootstrap_stats <- bootstrap_samples %>%
  group_by(replicate) %>%
  summarize(x_bar = mean(infected))
```

3. Graph the bootstrap distribution:



4. Estimate the standard error

```
bootstrap_stats %>% summarize(SE = sd(x_bar))
```

```
## # A tibble: 1 x 1
##       SE
##    <dbl>
## 1 0.181
```

Bootstrap Reproduction Rate

- To create a 95% confidence interval:

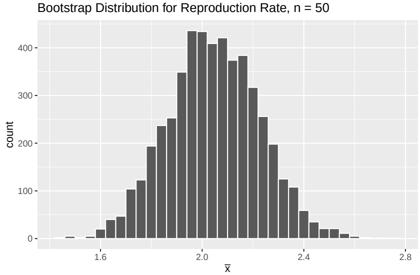
1. Create the bootstrap samples:

```
set.seed(121)
bootstrap_samples <- covid %>%
  rep_sample_n(size = 50, replace = TRUE, reps =
```

2. Compute bootstrap statistics:

```
bootstrap_stats <- bootstrap_samples %>%
  group_by(replicate) %>%
  summarize(x_bar = mean(infected))
```

3. Graph the bootstrap distribution:



4. Estimate the standard error

```
bootstrap_stats %>% summarize(SE = sd(x_bar))
```

```
## # A tibble: 1 x 1
##       SE
##    <dbl>
## 1 0.181
```

5. The 95% confidence interval is

$$\bar{x} \pm 2 \cdot SE \qquad 2.06 \pm 2 \cdot 0.181$$

Generalized Confidence Intervals

- In the previous example, we used the fact that for approximately bell-shaped sampling distributions, 95% of of sample statistics are within 2 SE of the population parameter

Generalized Confidence Intervals

- In the previous example, we used the fact that for approximately bell-shaped sampling distributions, 95% of of sample statistics are within 2 SE of the population parameter
  - But suppose we instead want a different success rate for our estimation method

Generalized Confidence Intervals

- In the previous example, we used the fact that for approximately bell-shaped sampling distributions, 95% of of sample statistics are within 2 SE of the population parameter
    - But suppose we instead want a different success rate for our estimation method
    - Or suppose we want to create interval estimates for sampling distributions that are not bell-shaped

Generalized Confidence Intervals

- In the previous example, we used the fact that for approximately bell-shaped sampling distributions, 95% of of sample statistics are within 2 SE of the population parameter
  - But suppose we instead want a different success rate for our estimation method
  - Or suppose we want to create interval estimates for sampling distributions that are not bell-shaped
- We can make these modifications again using the bootstrap approximation to the sampling distribution

Generalized Confidence Intervals

- In the previous example, we used the fact that for approximately bell-shaped sampling distributions, 95% of of sample statistics are within 2 SE of the population parameter
  - But suppose we instead want a different success rate for our estimation method
  - Or suppose we want to create interval estimates for sampling distributions that are not bell-shaped
- We can make these modifications again using the bootstrap approximation to the sampling distribution

General Confidence Intervals

The $C\%$ confidence interval for a parameter is an interval estimate that is computed from sample data by a method that captures the parameter for $C\%$ of all samples.

Review: Percentiles and Quantiles

- For a number k between 0 and 100, the kth **percentile** of a distribution is the value so that 5% of the data is less than or equal to that value.

Review: Percentiles and Quantiles

- For a number k between 0 and 100, the kth **percentile** of a distribution is the value so that 5% of the data is less than or equal to that value.
  - The median is the 50th percentile of a distribution, and the 1st/3rd quartiles are the 25th and 75th percentiles, respectively.

Review: Percentiles and Quantiles

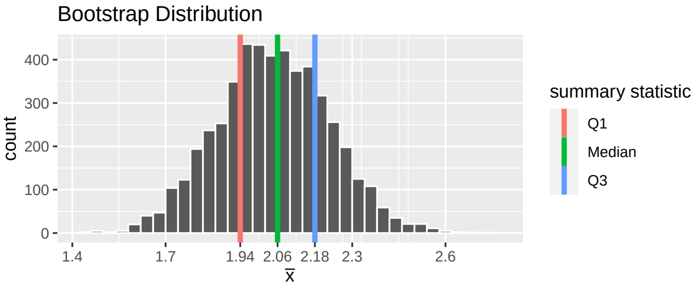
- For a number k between 0 and 100, the kth **percentile** of a distribution is the value so that 5% of the data is less than or equal to that value.
  - The median is the 50th percentile of a distribution, and the 1st/3rd quartiles are the 25th and 75th percentiles, respectively.
- For a number p between 0 and 1, the p **quantile** of a distribution is the value so that a proportion p of the data is less than or equal to that value.

Review: Percentiles and Quantiles

- For a number k between 0 and 100, the kth **percentile** of a distribution is the value so that 5% of the data is less than or equal to that value.
  - The median is the 50th percentile of a distribution, and the 1st/3rd quartiles are the 25th and 75th percentiles, respectively.
- For a number p between 0 and 1, the p **quantile** of a distribution is the value so that a proportion p of the data is less than or equal to that value.
  - The median is the 0.5 quantile of a distribution, and the 1st/3rd quartiles are the 0.25 and 0.75 quantiles, respectively.

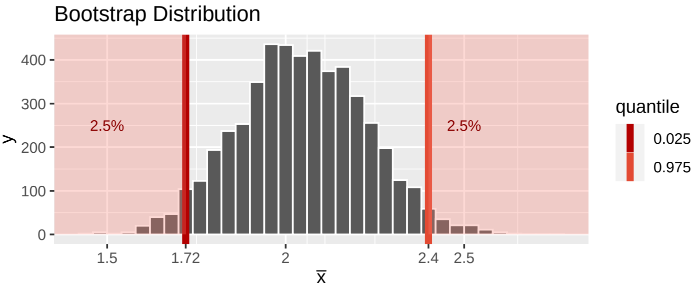
Review: Percentiles and Quantiles

- For a number k between 0 and 100, the kth **percentile** of a distribution is the value so that 5% of the data is less than or equal to that value.
  - The median is the 50th percentile of a distribution, and the 1st/3rd quartiles are the 25th and 75th percentiles, respectively.
- For a number p between 0 and 1, the p **quantile** of a distribution is the value so that a proportion p of the data is less than or equal to that value.
  - The median is the 0.5 quantile of a distribution, and the 1st/3rd quartiles are the 0.25 and 0.75 quantiles, respectively.



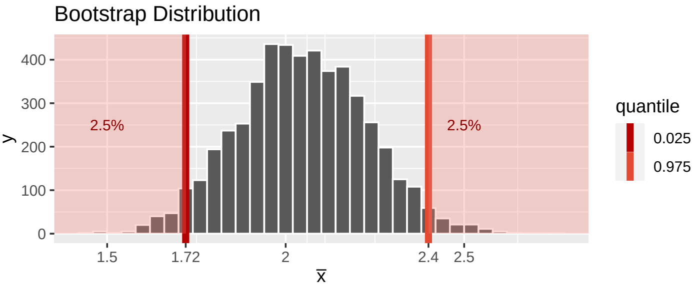
Quantiles and Percentiles

- By definition, 2.5% of the data is less than the .025 quantile, and 2.5% of the data is greater than the .975 quantile



Quantiles and Percentiles

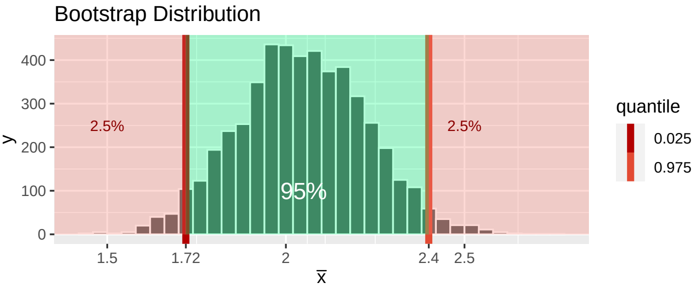
- By definition, 2.5% of the data is less than the .025 quantile, and 2.5% of the data is greater than the .975 quantile



- But this means that 95% of the data is between the .025 and the .975 quantiles

Quantiles and Percentiles

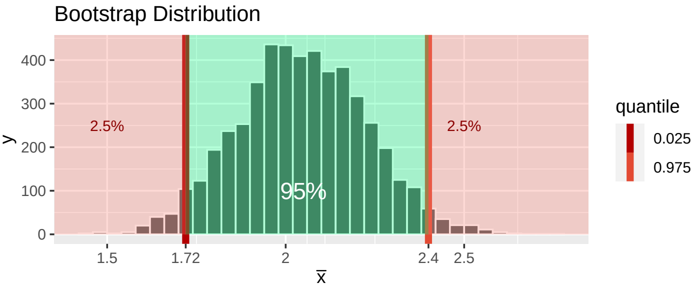
- By definition, 2.5% of the data is less than the .025 quantile, and 2.5% of the data is greater than the .975 quantile



- But this means that 95% of the data is between the .025 and the .975 quantiles

Quantiles and Percentiles

- By definition, 2.5% of the data is less than the .025 quantile, and 2.5% of the data is greater than the .975 quantile



- But this means that 95% of the data is between the .025 and the .975 quantiles
  - For a sampling distribution that is approximately bell-shaped, the .025 quantile is about $2 \cdot SE$ below the mean, and the .975 quantile is about $2 \cdot SE$ above the mean

The Percentile Method

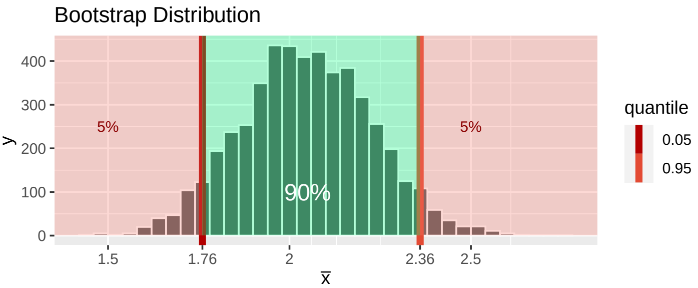
- Suppose we want to construct a 90% confidence interval for the reproduction rate

The Percentile Method

- Suppose we want to construct a 90% confidence interval for the reproduction rate
  - Instead of adding/subtracting $2 * SE$, find the 0.05 and .95 quantiles in the bootstrap distribution. Then 90% of bootstrap sample statistics will be between these values

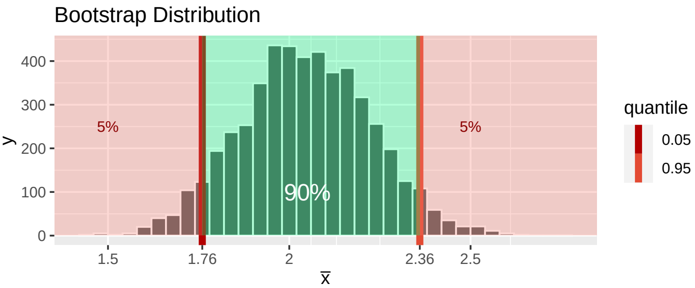
The Percentile Method

- Suppose we want to construct a 90% confidence interval for the reproduction rate
    - Instead of adding/subtracting $2 * SE$, find the 0.05 and .95 quantiles in the bootstrap distribution. Then 90% of bootstrap sample statistics will be between these values



The Percentile Method

- Suppose we want to construct a 90% confidence interval for the reproduction rate
  - Instead of adding/subtracting $2 * SE$, find the 0.05 and .95 quantiles in the bootstrap distribution. Then 90% of bootstrap sample statistics will be between these values



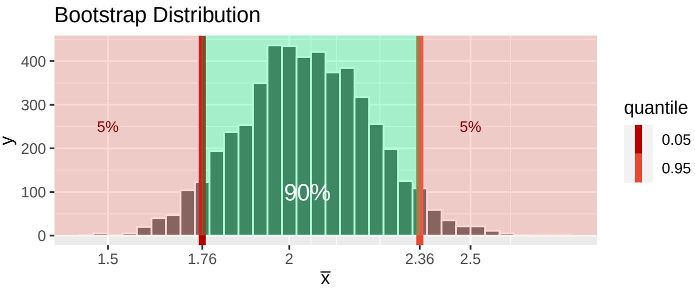
- We can use the `quantile` function in R to calculate the .05 and .95 quantiles

```
quantile(bootstrap_stats$x_bar, c(.05, .95))
```

```
##   5%  95%
## 1.76 2.36
```

The Percentile Method

- Suppose we want to construct a 90% confidence interval for the reproduction rate
    - Instead of adding/subtracting $2 * SE$, find the 0.05 and .95 quantiles in the bootstrap distribution. Then 90% of bootstrap sample statistics will be between these values



- Our 90% confidence interval is therefore 1.76 to 2.36

```
quantile(bootstrap_stats$x_bar, c(.05, .95))
```

```
##   5%  95%
## 1.76 2.36
```

Precision

How can we increase the precision of our confidence interval (i.e. decrease the margin of error)?

Precision

How can we increase the precision of our confidence interval (i.e. decrease the margin of error)?

- Increase sample size.
  - The standard deviation of the sampling distribution decreases as sample size increases. More sample means are closer to the true parameter

Precision

How can we increase the precision of our confidence interval (i.e. decrease the margin of error)?

- Increase sample size.
  - The standard deviation of the sampling distribution decreases as sample size increases. More sample means are closer to the true parameter
- Decrease confidence level.
  - The margin of error is determined by the percentiles. A 95% confidence interval is formed by the 2.5th and 97.5th percentiles in the bootstrap distribution.
  - Decreasing confidence level brings the percentiles closer to the 50th percentile, decreasing the width of the interval.

Section 2

Confidence Interval Misunderstandings

Common Confidence Interval Misunderstandings

Suppose we wish to estimate the number of hours a Reed student sleeps on a typical night. We obtain the following 95% confidence interval: $(7.86, 8.34)$

Common Confidence Interval Misunderstandings

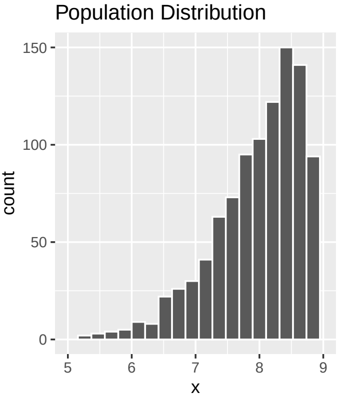
Suppose we wish to estimate the number of hours a Reed student sleeps on a typical night. We obtain the following 95% confidence interval: $(7.86, 8.34)$

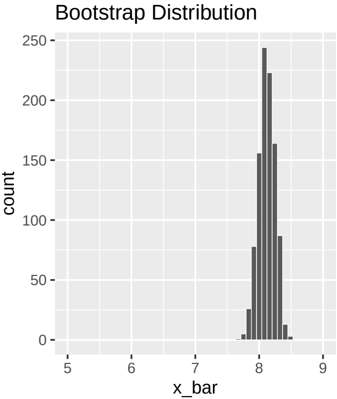
1. A 95% confidence interval **does not** contain 95% of observations in the population.

Common Confidence Interval Misunderstandings

Suppose we wish to estimate the number of hours a Reed student sleeps on a typical night. We obtain the following 95% confidence interval: $(7.86, 8.34)$

1. A 95% confidence interval **does not** contain 95% of observations in the population.

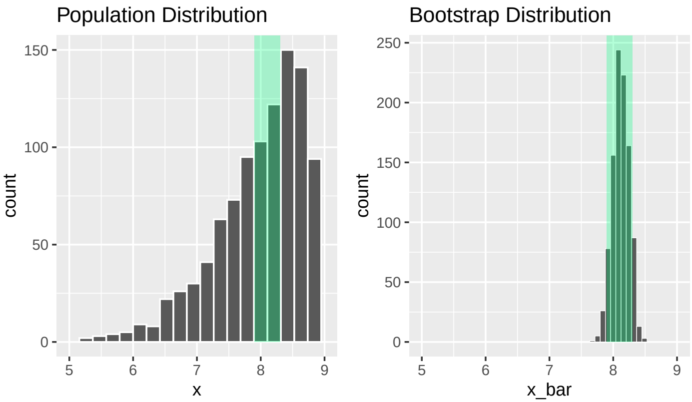




Common Confidence Interval Misunderstandings

Suppose we wish to estimate the number of hours a Reed student sleeps on a typical night. We obtain the following 95% confidence interval: $(7.86, 8.34)$

1. A 95% confidence interval **does not** contain 95% of observations in the population.

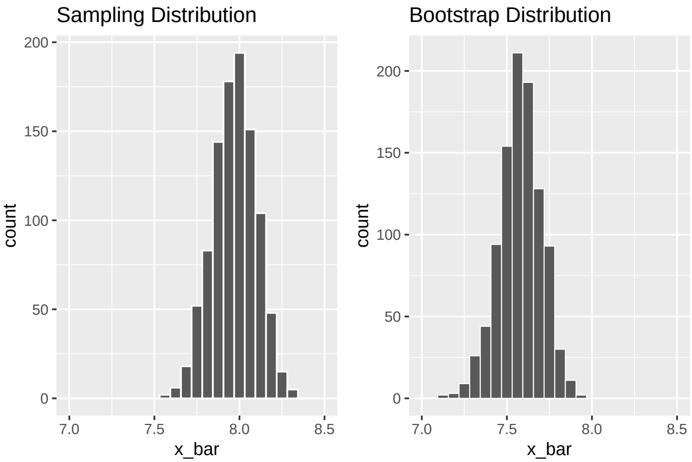


Common Confidence Interval Misunderstandings

2. A 95% confidence interval **does not** mean that 95% of all sample means fall within the given range.

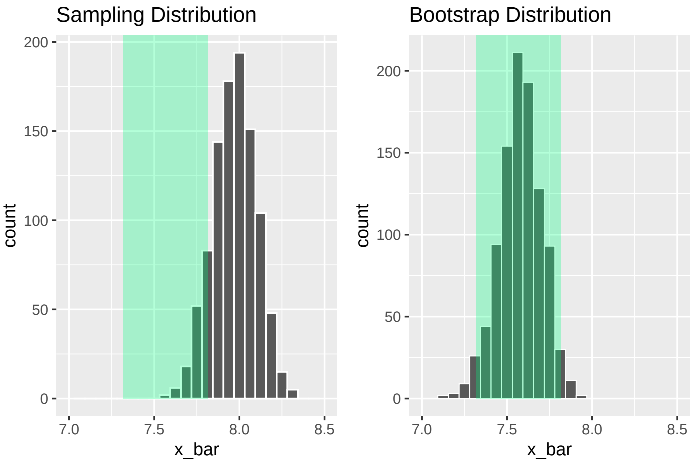
Common Confidence Interval Misunderstandings

2. A 95% confidence interval **does not** mean that 95% of all sample means fall within the given range.



Common Confidence Interval Misunderstandings

2. A 95% confidence interval **does not** mean that 95% of all sample means fall within the given range.



Common Confidence Interval Misunderstandings

3. A 95% confidence interval **does not** mean that there is a 95% chance that the true parameter falls in the given range.

Common Confidence Interval Misunderstandings

3. A 95% confidence interval **does not** mean that there is a 95% chance that the true parameter falls in the given range.

- Once a random sample has been observed and the confidence interval calculated, there is no more randomness in the process. We cannot make probabilistic statements about the outcome.

Common Confidence Interval Misunderstandings

3. A 95% confidence interval **does not** mean that there is a 95% chance that the true parameter falls in the given range.

- Once a random sample has been observed and the confidence interval calculated, there is no more randomness in the process. We cannot make probabilistic statements about the outcome.
    - At this point, the interval either does or does not contain the fixed (but unknown) parameter

Common Confidence Interval Misunderstandings

3. A 95% confidence interval **does not** mean that there is a 95% chance that the true parameter falls in the given range.

- Once a random sample has been observed and the confidence interval calculated, there is no more randomness in the process. We cannot make probabilistic statements about the outcome.
    - At this point, the interval either does or does not contain the fixed (but unknown) parameter
    - One sample happened to have a sample mean of 4.9, producing a confidence interval of $(4.6, 5.2)$.

Common Confidence Interval Misunderstandings

3. A 95% confidence interval **does not** mean that there is a 95% chance that the true parameter falls in the given range.

- Once a random sample has been observed and the confidence interval calculated, there is no more randomness in the process. We cannot make probabilistic statements about the outcome.
    - At this point, the interval either does or does not contain the fixed (but unknown) parameter
    - One sample happened to have a sample mean of 4.9, producing a confidence interval of $(4.6, 5.2)$.
    - Based on what you know about sleep patterns, do you think there is a 95% chance this interval contains the true parameter?

Common Confidence Interval Misunderstandings

3. A 95% confidence interval **does not** mean that there is a 95% chance that the true parameter falls in the given range.

- Once a random sample has been observed and the confidence interval calculated, there is no more randomness in the process. We cannot make probabilistic statements about the outcome.
  - At this point, the interval either does or does not contain the fixed (but unknown) parameter
  - One sample happened to have a sample mean of 4.9, producing a confidence interval of $(4.6, 5.2)$.
  - Based on what you know about sleep patterns, do you think there is a 95% chance this interval contains the true parameter?
  - What is a plausible alternative explanation for this interval?

Section 3

The `infer` package

The infer Package

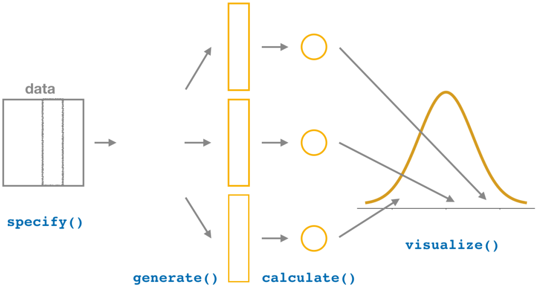
- The `infer` package makes efficient use of the `%>%` operator perform statistical inference.

The infer Package

- The `infer` package makes efficient use of the %>% operator perform statistical inference.
- The `infer` package makes use of several verbs-like functions:
  - `specify`, `generate`, `calculate`, `visualize`, `get_ci`

The infer Package

- The `infer` package makes efficient use of the %>% operator perform statistical inference.
- The `infer` package makes use of several verbs-like functions:
  - `specify`, `generate`, `calculate`, `visualize`, `get_ci`



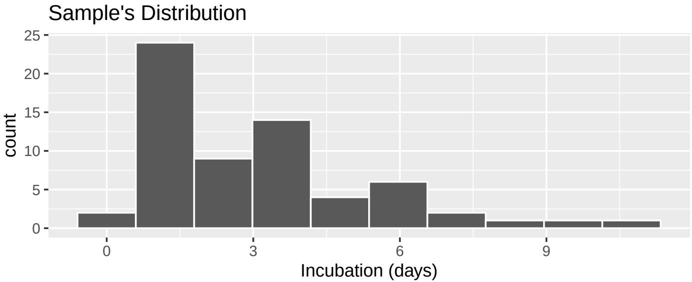
COVID Incubation Time

The Infectious Disease Dynamics Group at Johns Hopkins University collected data between Dec 2019 and Jan 2019 on exposure and symptom onset for COVID-19 in Hubei Province of China.

COVID Incubation Time

The Infectious Disease Dynamics Group at Johns Hopkins University collected data between Dec 2019 and Jan 2019 on exposure and symptom onset for COVID-19 in Hubei Province of China.

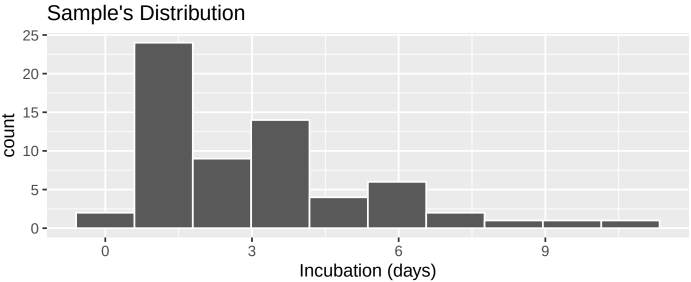
The distribution of `Incubation` times for 64 patients is shown below:



COVID Incubation Time

The Infectious Disease Dynamics Group at Johns Hopkins University collected data between Dec 2019 and Jan 2019 on exposure and symptom onset for COVID-19 in Hubei Province of China.

The distribution of `Incubation` times for 64 patients is shown below:



- What is the population of interest? What is the parameter?
- What is the sample? What is the statistic?

`specify` the variables

- Every statistical investigation begins with a sample data frame (i.e. `covid`)

specify the variables

- Every statistical investigation begins with a sample data frame (i.e. `covid`)
- The sample may contain many variables of interest

specify the variables

- Every statistical investigation begins with a sample data frame (i.e. `covid`)
- The sample may contain many variables of interest
- We must first specify which variable(s) will be the focus of our investigation by designating a response variable

specify the variables

- Every statistical investigation begins with a sample data frame (i.e. covid)
- The sample may contain many variables of interest
- We must first specify which variable(s) will be the focus of our investigation by designating a response variable
- To investigate the infection rate

```
covid %>%
  specify(response = Incubation)
```

`generate` replicates

- In order to create a bootstrap distribution, we need to resample many times from the OG sample

generate replicates

- In order to create a bootstrap distribution, we need to resample many times from the OG sample
- After selecting variables, pipe results into the generate function to create replicates

```
covid %>%
  specify(response = Incubation) %>%
  generate( reps = 2000, type = "bootstrap")
```

generate replicates

- In order to create a bootstrap distribution, we need to resample many times from the OG sample
- After selecting variables, pipe results into the generate function to create replicates

```
covid %>%
  specify(response = Incubation) %>%
  generate( reps = 2000, type = "bootstrap")
```

- We need to indicate how many replicates we want, and what type of method we'll use to create them.

generate replicates

- In order to create a bootstrap distribution, we need to resample many times from the OG sample
- After selecting variables, pipe results into the generate function to create replicates

```
covid %>%
  specify(response = Incubation) %>%
  generate( reps = 2000, type = "bootstrap")
```

- We need to indicate how many replicates we want, and what type of method we'll use to create them.
- For bootstrap confidence intervals, choose type = "bootstrap", and almost always use at least reps = 2000

generate replicates

- In order to create a bootstrap distribution, we need to resample many times from the OG sample
- After selecting variables, pipe results into the generate function to create replicates

```
covid %>%
  specify(response = Incubation) %>%
  generate( reps = 2000, type = "bootstrap")
```

- We need to indicate how many replicates we want, and what type of method we'll use to create them.
- For bootstrap confidence intervals, choose type = "bootstrap", and almost always use at least reps = 2000
- The resulting data frame has a number of rows equal reps × sample_size

`calculate` summary statistics

- Once we have our bootstrap samples, we need to compute the corresponding statistics

calculate summary statistics

- Once we have our bootstrap samples, we need to compute the corresponding statistics
- Use the `calculate` function, whose first argument is `stat`

calculate summary statistics

- Once we have our bootstrap samples, we need to compute the corresponding statistics
- Use the `calculate` function, whose first argument is `stat`
- Many statistics are available: `"mean"`, `"sum"`, `"sd"`, `"median"`, `"prop"`, `"diff in mean`, `"correlation"`, `"slope"`, and more!

calculate summary statistics

- Once we have our bootstrap samples, we need to compute the corresponding statistics
- Use the `calculate` function, whose first argument is `stat`
- Many statistics are available: `"mean"`, `"sum"`, `"sd"`, `"median"`, `"prop"`, `"diff in mean`, `"correlation"`, `"slope"`, and more!

```
covid %>%
  specify(response = Incubation) %>%
  generate( reps = 2000, type = "bootstrap") %>%
  calculate(stat = "mean")
```

calculate summary statistics

- Once we have our bootstrap samples, we need to compute the corresponding statistics
- Use the `calculate` function, whose first argument is `stat`
- Many statistics are available: `"mean"`, `"sum"`, `"sd"`, `"median"`, `"prop"`, `"diff in mean`, `"correlation"`, `"slope"`, and more!

```
covid %>%
  specify(response = Incubation) %>%
  generate( reps = 2000, type = "bootstrap") %>%
  calculate(stat = "mean")
```

- After applying `calculate` the resulting data frame consists of one bootstrap statistic for each replicate (saved to the variable `stat`)

Sample Statistic

- Suppose you want to just calculate summary statistics of the OG sample

Sample Statistic

- Suppose you want to just calculate summary statistics of the OG sample
- By using specify and calculate (and omitting generate) we can do just that, paralleling similar calculation for the bootstrap statistics

```
covid_stat<- covid %>%
  specify(response = Incubation) %>%
  calculate(stat = "mean")
covid_stat
```

```
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  3.03
```

Sample Statistic

- Suppose you want to just calculate summary statistics of the OG sample
- By using specify and calculate (and omitting generate) we can do just that, paralleling similar calculation for the bootstrap statistics

```
covid_stat<- covid %>%
  specify(response = Incubation) %>%
  calculate(stat = "mean")
covid_stat
```

```
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  3.03
```

- Note: we saved the value of this calculation as covid_stat so we could use it later

Save the bootstrap too

- Since we also will want to make frequent use of the bootstrap statistics, it's worth saving them as a variable too:

Save the bootstrap too

- Since we also will want to make frequent use of the bootstrap statistics, it's worth saving them as a variable too:

```
covid_boot<- covid %>%
  specify(response = Incubation) %>%
  generate( reps = 2000, type = "bootstrap") %>%
  calculate(stat = "mean")

head(covid_boot)
```

```
## # A tibble: 6 x 2
##   replicate  stat
##       <int> <dbl>
## 1         1  2.65
## 2         2  3.15
## 3         3  2.67
## 4         4  3.39
## 5         5  3.27
## 6         6  3.35
```

visualize Bootstrap Distribution

- In order to perform any statistical inference, we need to ensure appropriate shape conditions on bootstrap distribution are met

visualize Bootstrap Distribution

- In order to perform any statistical inference, we need to ensure appropriate shape conditions on bootstrap distribution are met
- Use the `visaulize` verb to quickly generate a reasonably nice-looking histogram of the bootstrap distribution.

visualize Bootstrap Distribution

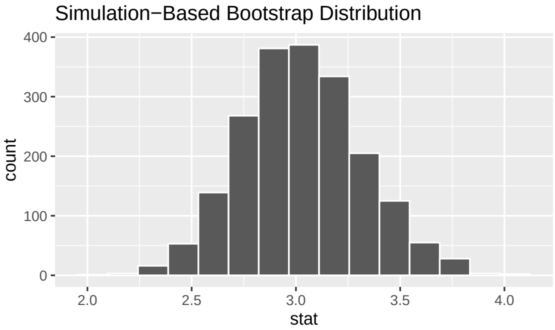
- In order to perform any statistical inference, we need to ensure appropriate shape conditions on bootstrap distribution are met
- Use the visaulize verb to quickly generate a reasonably nice-looking histogram of the bootstrap distribution.

```
covid_boot %>% visualize()
```

visualize Bootstrap Distribution

- In order to perform any statistical inference, we need to ensure appropriate shape conditions on bootstrap distribution are met
- Use the `visaulize` verb to quickly generate a reasonably nice-looking histogram of the bootstrap distribution.

```
covid_boot %>% visualize()
```



get_confidence_interval to... Get Confidence Interval

- To compute a confidence interval, pipe the calculated data frame into get_confidence_interval (you can use get_ci for brevity)

get_confidence_interval to... Get Confidence Interval

- To compute a confidence interval, pipe the calculated data frame into get_confidence_interval (you can use get_ci for brevity)
- We need to specify the type of interval we want (either "percentile" or "se"), along with the confidence level

get_confidence_interval to... Get Confidence Interval

- To compute a confidence interval, pipe the `calculated` data frame into `get_confidence_interval` (you can use `get_ci` for brevity)
- We need to specify the `type` of interval we want (either `"percentile"` or `"se"`), along with the confidence `level`
- It's useful to save the resulting data frame for later use

get_confidence_interval to... Get Confidence Interval

- To compute a confidence interval, pipe the calculated data frame into get_confidence_interval (you can use get_ci for brevity)
- We need to specify the type of interval we want (either "percentile" or "se"), along with the confidence level
- It's useful to save the resulting data frame for later use

```
percentile_ci<-covid_boot %>%
  get_ci(level = .95, type = "percentile")
percentile_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.49     3.63
```

get_confidence_interval to. . . Get Confidence Interval

- To compute a confidence interval, pipe the calculated data frame into get_confidence_interval (you can use get_ci for brevity)
- We need to specify the type of interval we want (either "percentile" or "se"), along with the confidence level
- It's useful to save the resulting data frame for later use

```
percentile_ci<-covid_boot %>%
  get_ci(level = .95, type = "percentile")
percentile_ci
```

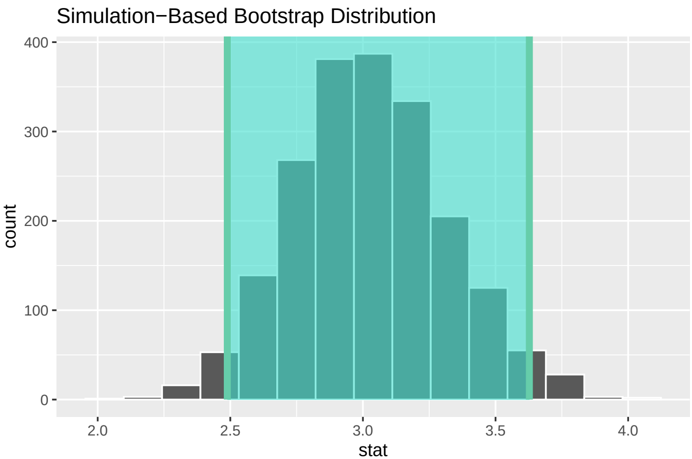
```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.49     3.63
```

- When using the percentile type, the first value printed is the lower and the second is the upper bound.

Shade Confidence Intervals

- Once you've used get_ci to obtain endpoints of the confidence interval, you can shade the sampling distribution with the confidence interval region.

```
covid_boot %>% visualize()+shade_ci(endpoints = percentile_ci)
```



Standard Error Method

- The confidence interval using the standard error method will be of the form

$$\text{statistic} \pm 2 \cdot SE$$

- Here, SE is an approximation of the standard error based on the standard deviation of the bootstrap distribution

Standard Error Method

- The confidence interval using the standard error method will be of the form

$$\text{statistic} \pm 2 \cdot SE$$

- Here, *SE* is an approximation of the standard error based on the standard deviation of the bootstrap distribution
    - It is possible to use the SE method with other confidence levels too. In this case, 2 is replaced with another appropriate value (discussed later this term)

Standard Error Method

- The confidence interval using the standard error method will be of the form

$$\text{statistic} \pm 2 \cdot SE$$

- Here, *SE* is an approximation of the standard error based on the standard deviation of the bootstrap distribution
    - It is possible to use the SE method with other confidence levels too. In this case, 2 is replaced with another appropriate value (discussed later this term)

```
se_ci<-covid_boot %>%
  get_ci(level = .95, type = "se", point_estimate = covid_stat)
se_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.46     3.60
```

Standard Error Method

- The confidence interval using the standard error method will be of the form

$$\text{statistic} \pm 2 \cdot SE$$

- Here, SE is an approximation of the standard error based on the standard deviation of the bootstrap distribution
  - It is possible to use the SE method with other confidence levels too. In this case, 2 is replaced with another appropriate value (discussed later this term)

```
se_ci<-covid_boot %>%
  get_ci(level = .95, type = "se", point_estimate = covid_stat)
se_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.46     3.60
```

- Note: for the se method, we also need to specify our point estimate (which is why we saved it as a variable before)

Compare the Methods

Each method produced a different confidence interval:

Compare the Methods

Each method produced a different confidence interval:

```
percentile_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.49     3.63
```

```
se_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.46     3.60
```

Compare the Methods

Each method produced a different confidence interval:

```
percentile_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.49     3.63
```

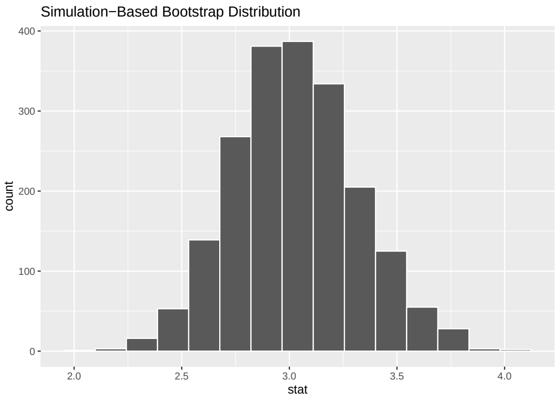
```
se_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     2.46     3.60
```

- Why?

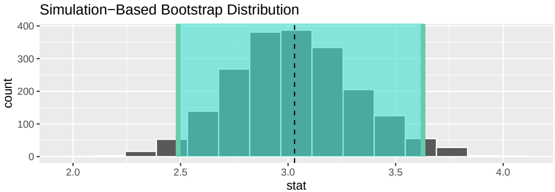
visualize Confidence Intervals

```
covid_boot %>% visualize()
```

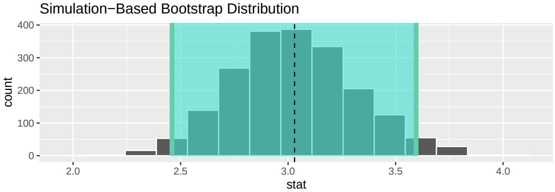


visualize Confidence Intervals

```
covid_boot %>% visualize() +
  shade_confidence_interval(endpoints = percentile_ci)+
  geom_vline(xintercept = 3.03, linetype = "dashed")
```

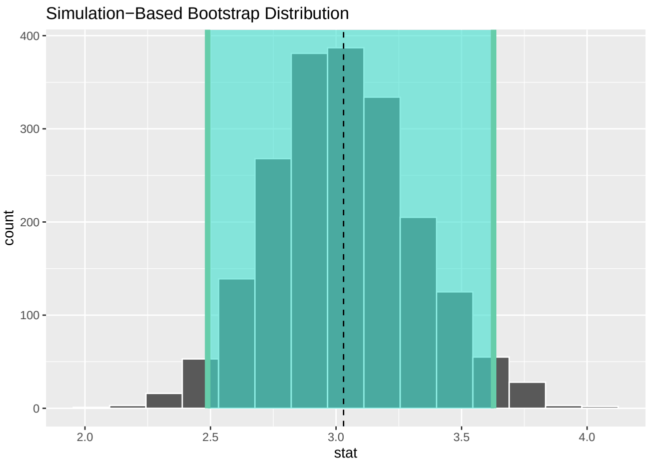



```
covid_boot %>% visualize() +
  shade_confidence_interval(endpoints = se_ci)+
  geom_vline(xintercept = 3.03, linetype = "dashed")
```



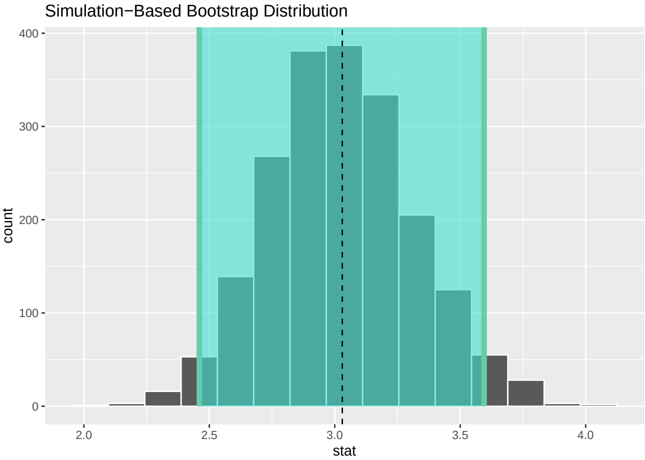
visualize Confidence Intervals

Percentile Method



visualize Confidence Intervals

SE Method



visualize Confidence Intervals

SE Method (with Percentile in blue)