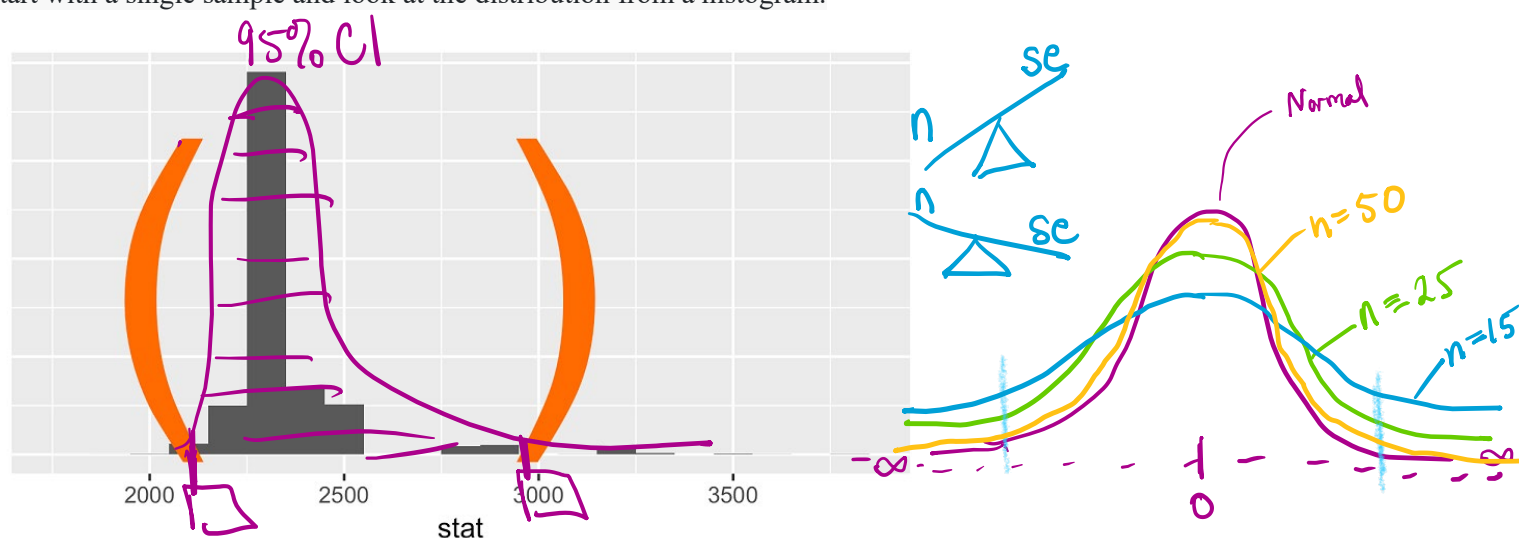


19.1 Bootstrap confidence interval for a mean

The inferential analysis methods in this chapter are grounded in quantifying how one dataset differs from another when they are both taken from the same population, but we aren't ever going to take more than one sample of observations. It does not make sense to take repeated samples from the same population because if you have the ability to take more samples, a larger sample size will benefit you more than taking two samples from the population.

Instead, of taking repeated samples from the actual population, we use bootstrapping to measure how the samples behave under an estimate of the population.

Start with a single sample and look at the distribution from a histogram.



The larger the area in the tails, the larger the potential error in our inferential calculations.

SE for confidence interval for t -distribution

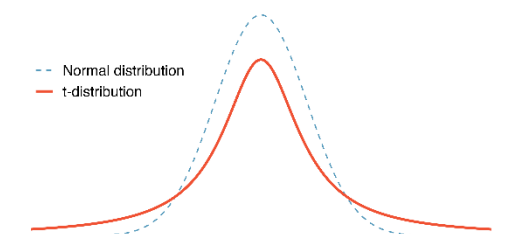
t -dist'n, $df = n - 1$

The amount of discrepancy between \bar{y} and μ is described using probability terms with the SAMPLING distribution of \bar{Y} , and the standard deviation of \bar{Y} is given as the **standard error of the mean**, which is $SE_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \approx \frac{s}{\sqrt{n}}$ (this is sometimes called the **estimated standard error**, since s is already an estimate of σ).

As with the sample proportion, the variability of the sample mean is well described by the mathematical theory given by the Central Limit Theorem. However, because of missing information about the inherent variability in the population (σ), a **t-distribution is used** in place of the standard normal when performing hypothesis test or confidence interval analyses.

The t -distribution is symmetric and bell curved, but has bigger tails than the normal curve, because of the larger standard deviation. It is dependent on the sample size, because the distribution requires the **degrees of freedom (df)**, which is $n-1$.

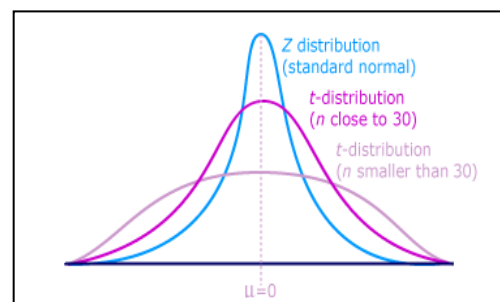
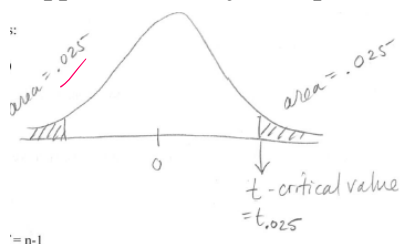
The extra thick tails of the t -distribution are exactly the correction needed to resolve the problem (due to extra variability of the T score) of using s in place of σ in the SE calculation.



Comparison of a t -distribution and a normal distribution

Notation

$t_{0.025}$ - two-tailed 5% critical value means:
(use the upper tail area of 2.5% probability with $df = n-1$)



Recall $CI_{\hat{p}} \rightarrow \hat{p} \pm z^*(se_{\hat{p}})$

Confidence Interval for μ

$$\bar{x} \pm \left(t_{0.025}^* SE_{\bar{x}} \right) = \bar{x} \pm t_{0.025}^* \left(\frac{s}{\sqrt{n}} \right)$$

** $df = n - 1$

** $t_{0.025}^*$ comes from qt(.975, df = degrees of freedom)

$$\left(\bar{x} - t_{n-1, \alpha/2} \cdot \frac{s}{\sqrt{n}}, \bar{x} + t_{n-1, \alpha/2} \cdot \frac{s}{\sqrt{n}} \right)$$



19.2.2 Evaluating the two conditions required for modeling \bar{x}

Two conditions are required to apply the Central Limit Theorem for a sample mean \bar{x} :

- **Independence.** The sample observations must be independent. The most common way to satisfy this condition is when the sample is a simple random sample from the population.
- **Normality.** When a sample is small, we also require that the sample observations come from a normally distributed population. We can relax this condition more and more for larger and larger sample sizes. This condition is obviously vague, making it difficult to evaluate, so next we introduce a couple rules of thumb to make checking this condition easier.

General rule for performing the normality check.

$$\bar{x} \sim N\left(\text{mean} = \mu, SE = \frac{\sigma}{\sqrt{n}}\right)$$

There is no perfect way to check the normality condition, so instead we use two general rules based on the number and magnitude of extreme observations. Note, it often takes practice to get a sense for whether a normal approximation is appropriate.

- **Small n:** If the sample size n is small and there are **no clear outliers** in the data, then we typically assume the data come from a nearly normal distribution to satisfy the condition.
- **Large n:** If the sample size n is large and there are no **particularly extreme** outliers, then we typically assume the sampling distribution of \bar{x} is nearly normal, even if the underlying distribution of individual observations is not.

If n is small ($n < 15$), the population must be basically normally distributed (no skewing)

If $15 < n < 30$, the population distribution should not have any strong skewing or outliers

If n is large ($n \geq 30$), the population may have some skewing or outliers (This is basically the CLT).

When n is 30 or more, you could use normal distribution calculations rather than t-distribution calculations BUT for our class, from now on, we will use t-distributions for quantitative variables.

Numerical variables \longrightarrow t-distribution calculations

Every data analysis should begin with an inspection of the graph of the data to see the shape of the distribution.

To be **more accurate:** 1. Widen the interval by

- Increasing the confidence level
- Decreasing the sample size

To be **more precise:** 2. Narrow the interval by

- Decreasing the confidence level
- Increasing the sample size

19.2.5 One sample t-tests

The test statistic for assessing a single mean is a T.

The t test statistic is a ratio of how the sample mean differs from the hypothesized mean as compared to how the observations vary.

$$t_s = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}}$$

When the null hypothesis is true and the conditions are met, it has a t-distribution with $df=n-1$.

Conditions:

- Independent observations.
- Large samples and no extreme outliers.

Example:

Is the typical US runner getting faster or slower over time? We consider this question in the context of the Cherry Blossom Race, which is a 10-mile race in Washington, DC each spring. **The average time for all runners who finished the Cherry Blossom Race in 2006 was 93.29 minutes.** We want to determine using data from 100 participants in the 2017 Cherry Blossom Race whether runners in this race are getting faster or slower, versus the other possibility that there has been no change.

different from $\mu_{2006} = 93.26 \text{ min}$

The `run17` data can be found in the `cherryblossom` R package.

1. What are appropriate hypotheses for this context?

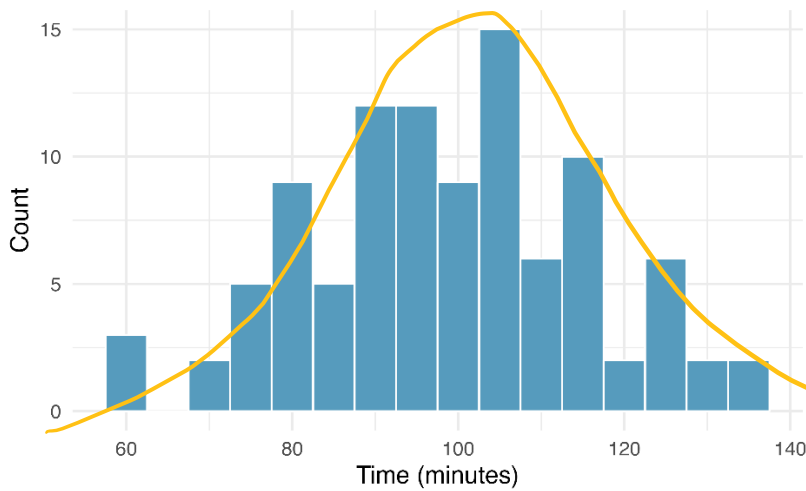
H_0 : There is no difference in finishing times for the Cherry Blossom 10K in 2017 from 2006. $H_0: \mu = 93.29$
 H_a : There is a difference " " " " $H_a: \mu \neq 93.29$

2. Check conditions for normality:

- a. Independent observations? ✓

- b. Check shape of distribution (histogram below) and sample size

finishing times histogram



1) relatively symmetric
AND
2) $n=100$
"sufficiently large"

If conditions are met, perform the computations for a t test comparing the null (from 2006) of a mean of 93.29 minutes to the 2017 mean.

Chapter 19 notes - single mean

R Saidi

Load the libraries and dataset

```
library(tidyverse)
library(openintro)
library(tidymodels)
```

T test

Use the Cherry Blossom dataset from 2017

```
library(openintro)
data("run17")
unique(run17$event)
[1] "10 Mile" "5K"
```

Mutate the variable `net_sec` to convert to minutes

First, filter out for 5K event times

```
run17_1 <- run17 |>
  filter(event != "5K") |>
  mutate(minutes = net_sec/60)
head(run17_1)

# A tibble: 6 × 10
  bib name      sex   age city  net_sec clock_sec pace_sec event minutes
  <int> <chr>    <chr> <int> <chr>   <dbl>   <dbl>   <dbl> <chr>   <dbl>
```

1	6	Hiwot G.	F	21	Ethio...	3217	3217	321	10	M...	53.6
2	22	Buze D.	F	22	Ethio...	3232	3232	323	10	M...	53.9
3	16	Gladys K.	F	31	Kenya	3276	3276	327	10	M...	54.6
4	4	Mamitu D.	F	33	Ethio...	3285	3285	328	10	M...	54.8
5	20	Karolina N.	F	35	Poland	3288	3288	328	10	M...	54.8
6	8	Firehiwot D.	F	33	Ethio...	3316	3316	331	10	M...	55.3

Generate a sample of size 100 from the population

```
# create a vector of size 100 for minutes from 2017
run17_samp <- sample(run17_1$minutes, size = 100, replace = FALSE)
summary(run17_samp)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 63.33   84.55   99.44   98.58  108.90  138.05
```

Perform the t test comparing the mean for 2017 to the mean from 2006

The mean from 2006 was 93.29 minutes

Here is the format for the single mean t test

```
t.test(x, mu = 0, alternative = "two.sided") t.test(x, mu = 0, alternative = "less") t.test(x, mu = 0, alternative = "greater")
```

```
# compare the mean from the samples from 2017 to the mean in 2006
t.test(run17_samp, mu = 93.29, alternative = "two.sided")
```

One Sample t-test

```
data: run17_samp
t = 3.2786, df = 99, p-value = 0.00144
alternative hypothesis: true mean is not equal to 93.29
95 percent confidence interval:
 95.37717 101.77616
sample estimates:
mean of x
 98.57667
```

$t = 2.1057$, $df = 99$, $p\text{-value} = 0.03776$

Reject the null. There is compelling evidence that the mean in 2017 is different from the mean in 2006.

95% CI mean finishings times in 2017 using tidymodels

create a bootstrap ci

```
boot_dist <- run17_1 |>
  specify(response = minutes) |>
  generate(reps = 1000, type = "bootstrap") |>
  calculate(stat = "mean")
head(boot_dist)

Response: minutes (numeric)
# A tibble: 6 × 2
```

```

  replicate stat
    <int> <dbl>
1         1  97.7
2         2  97.9
3         3  97.7
4         4  97.9
5         5  97.6
6         6  97.8

ci <- get_ci(boot_dist, level = .95)
ci

# A tibble: 1 × 2
  lower_ci upper_ci
    <dbl>    <dbl>
1    97.5    98.0

```

We are 95% confident that the true mean finishing time for the 10K Cherry Blossom race in 2017 is between 97.559 and 98.060 minutes.

Now work with the Ames, IA home sale price variable in ames dataset

```

data("ames")
head(ames)

# A tibble: 6 × 74
  MS_SubClass MS_Zoning Lot_Frontage Lot_Area Street Alley Lot_Shape
  <fct>       <fct>      <dbl>    <int> <fct>  <fct> <fct>
1 One_Story_1946_and_New... Resident...    141    31770 Pave   No_A... Slightly...
2 One_Story_1946_and_New... Resident...     80    11622 Pave   No_A... Regular
3 One_Story_1946_and_New... Resident...     81    14267 Pave   No_A... Slightly...
4 One_Story_1946_and_New... Resident...     93    11160 Pave   No_A... Regular
5 Two_Story_1946_and_New... Resident...     74    13830 Pave   No_A... Slightly...
6 Two_Story_1946_and_New... Resident...     78     9978 Pave   No_A... Slightly...
# [i] 67 more variables: Land_Contour <fct>, Utilities <fct>, Lot_Config <fct>,
#   Land_Slope <fct>, Neighborhood <fct>, Condition_1 <fct>, Condition_2 <fct>,
#   Bldg_Type <fct>, House_Style <fct>, Overall_Cond <fct>, Year_Built <int>,
#   Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>,
#   Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>,
#   Mas_Vnr_Area <dbl>, Exter_Cond <fct>, Foundation <fct>, Bsmt_Cond <fct>,
#   Bsmt_Exposure <fct>, BsmtFin_Type_1 <fct>, BsmtFin_SF_1 <dbl>, ...

```

There are 74 variables about house sale characteristics

Summarize sale price variable

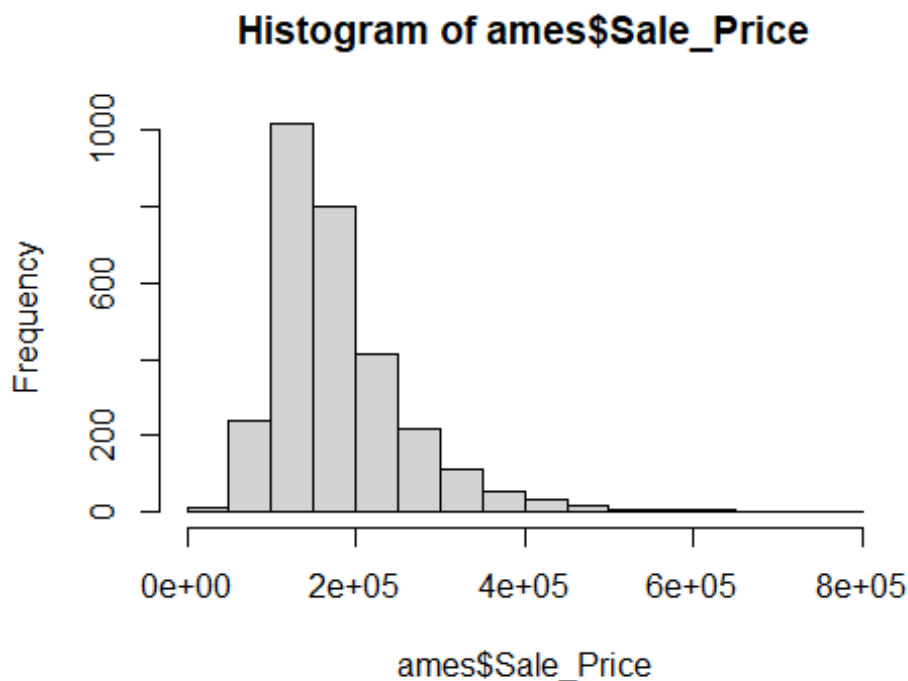
```

#options(scipen=999) # converts from scientific to standard
summary(ames$Sale_Price)

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
12789 129500 160000 180796 213500 755000

hist(ames$Sale_Price)

```



Generate a sample of size 100 from the data

Use the samp_price data for future calculations

```
samp_price <- sample_n(ames, size = 100, replace = FALSE)
head(samp_price)
```

A tibble: 6 × 74

	MS_SubClass <fct>	MS_Zoning <fct>	Lot_Frontage <dbl>	Lot_Area <int>	Street <fct>	Alley <fct>	Lot_Shape <fct>
1	One_Story_PUD_1946_and...	Resident...	41	5748	Pave	No_A...	Slightly...
2	Two_Story_PUD_1946_and...	Resident...	24	2364	Pave	No_A...	Regular
3	Two_Story_1946_and_New...	Resident...	0	15896	Pave	No_A...	Moderate...
4	Two_Story_1946_and_New...	Resident...	66	16226	Pave	No_A...	Irregular
5	One_Story_1946_and_New...	Resident...	76	10141	Pave	No_A...	Slightly...
6	Split_or_Multilevel	Resident...	80	9600	Pave	No_A...	Regular

[i] 67 more variables: Land_Contour <fct>, Utilities <fct>, Lot_Config <fct>, Land_Slope <fct>, Neighborhood <fct>, Condition_1 <fct>, Condition_2 <fct>, Bldg_Type <fct>, House_Style <fct>, Overall_Cond <fct>, Year_Built <int>, Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>, Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>, Mas_Vnr_Area <dbl>, Exter_Cond <fct>, Foundation <fct>, Bsmt_Cond <fct>, Bsmt_Exposure <fct>, BsmtFin_Type_1 <fct>, BsmtFin_SF_1 <dbl>, ...

The simple code to create a 95% CI for single mean

Use t.test(). You must indicate the variable of interest

```
t.test(samp_price$Sale_Price, conf.level = .95) # defaults to null = 0
```


One Sample t-test

```
data: samp_price$Sale_Price
t = 25.693, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 164143.5 191618.7
sample estimates:
mean of x
 177881.1
```

We are 95% confident that the mean sale price for homes in Ames is between \$155,019 and \$184,587.7.

##Answer the question:

Test to determine if the mean home sale price in Ames was different from the national average at that time from 2006-2010.

The average home price in the US in 2010 was \$222,900.

Ho: The mean home sale price in Ames, IA is the same as national average. Ha: "'''''''' is different from "''''

```
t.test(samp_price$Sale_Price, mu=222900)
```

One Sample t-test

```
data: samp_price$Sale_Price
t = -6.5024, df = 99, p-value = 3.238e-09
alternative hypothesis: true mean is not equal to 222900
95 percent confidence interval:
 164143.5 191618.7
sample estimates:
mean of x
 177881.1
```

The p-value is very small. We reject the null. There is very strong evidence that the true mean home sale price in Ames, IA in 2010 is different from the national home sale price of \$222,900.

We are 95% confident that the true median home sale price in Ames is between \$156716.1 and \$184598.2

Perform the hypothesis test using tidymodels

Answer the question: Test to determine if the mean home sale price in Ames was different from the national average at that time from 2006-2010.

The average home price in the US in 2010 was \$222,900.

Calculate the observed mean

```
# Calculate observed mean
price_mean_obs <- samp_price |>
  # Summarize to calculate the mean observed sale price
  summarize(mean_price = mean(Sale_Price))
price_mean_obs
```

```
# A tibble: 1 × 1
  mean_price
    <dbl>
1    177881.
```

Use “specify”, “hypothesize”, “generate” and “calculate”

```
sale_mean_ht <- samp_price |>
  # Specify sale_price as the response
  specify(response = Sale_Price) |>
  # Set the hypothesis that national home price is 222900
  hypothesize(null = "point", mu = 222900) |>
  # Generate 10000 bootstrap replicates
  generate(reps = 1000, type = "bootstrap") |>
  # Calculate the mean
  calculate(stat = "mean")
head(sale_mean_ht)
```

Response: Sale_Price (numeric)

Null Hypothesis: point


```
# A tibble: 6 × 2
  replicate    stat
    <int>    <dbl>
1         1 231339.
2         2 219065.
3         3 224582.
4         4 214257.
5         5 229107.
6         6 221437.
```

Calculate the p-value

The p-value is 0. Reject the null. The mean sale price of homes in Ames is different (and much less than) the national home price in 2010.

```
pvalue <- get_p_value(sale_mean_ht, price_mean_obs, "two-sided")
```

Warning: Please be cautious in reporting a p-value of 0. This result is an approximation based on the number of `reps` chosen in the `generate()` step.

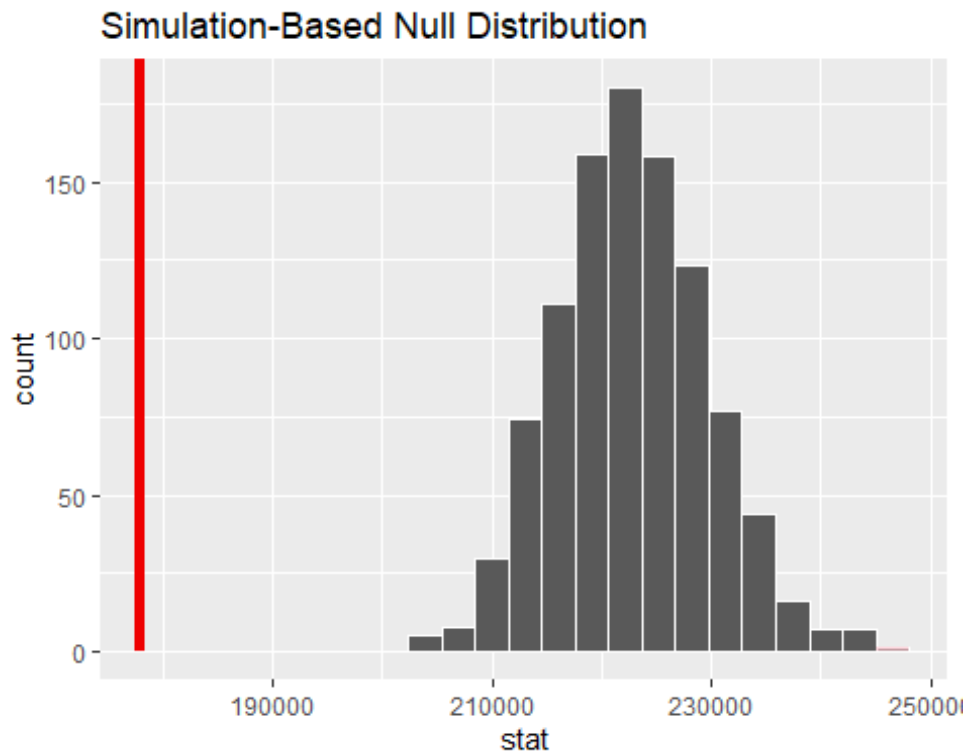
 See `get_p_value()` (`?infer::get_p_value()`) for more information.

```
pvalue
```

```
# A tibble: 1 × 1
  p_value
    <dbl>
1      0
```

The p-value is 0. Reject the null. The mean sale price of homes in Ames is different (and much less than) the national home price in 2010.

```
sale_mean_ht |>
  visualize() +
  shade_p_value(obs_stat = price_mean_obs, direction = "two-sided")
```



We can see that the 95% CI is (177902, 183690).

Now using `tidymodels` code for a 95% CI for the ames house prices.

This is a two-part exercise: First, generate 15000 bootstrap distributions of `sale_price` in the ames data frame and record the median of each bootstrap distribution.

- Specify that `sale_price` is the response variable.
- Generate 15000 bootstrap replicates.
- Calculate the median of each distribution.

```
# Generate bootstrap distribution of means
sale_mean_ci <- samp_price |>
  # Specify the variable of interest
  specify(response = Sale_Price) |>
  # Generate 10000 bootstrap samples
  generate(reps = 1000, type = "bootstrap") |>
  # Calculate the mean of each bootstrap sample
  calculate(stat = "mean") # change this to "median" to see how this changes
```

```
# Take a peek
head(sale_mean_ci)
```

Response: Sale_Price (numeric)

```
# A tibble: 6 × 2
```

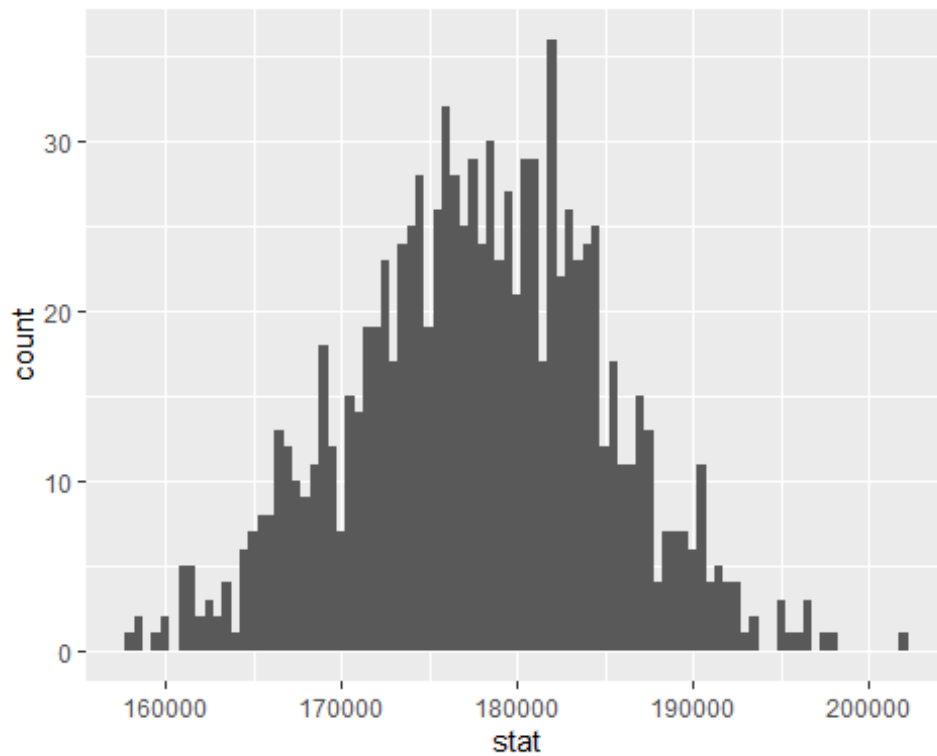
	replicate	stat
	<int>	<dbl>
1	1	181754.
2	2	171630.
3	3	181850.
4	4	187091.

```
5      5 157827.  
6      6 180795
```

Plot a histogram of the bootstrap replications

Look at the shape of the distribution

```
# Plot the sale_mean_ci statistic  
ggplot(sale_mean_ci, aes(x = stat)) +  
  # Make it a histogram with a binwidth of 500  
  geom_histogram(binwidth = 500)
```



Calculate the 95% CI using the percentile method

```
# Calculate the 95% CI via percentile method  
sale_mean_ci |>  
  summarize(  
    l = quantile(stat, 0.025),  
    u = quantile(stat, 0.975)  
  )  
  
# A tibble: 1 × 2  
    l      u  
  <dbl> <dbl>  
1 163692. 191493.  
  
sale_ci <- get_ci(sale_mean_ci, level=.95)  
sale_ci  
  
# A tibble: 1 × 2  
  lower_ci upper_ci
```

```

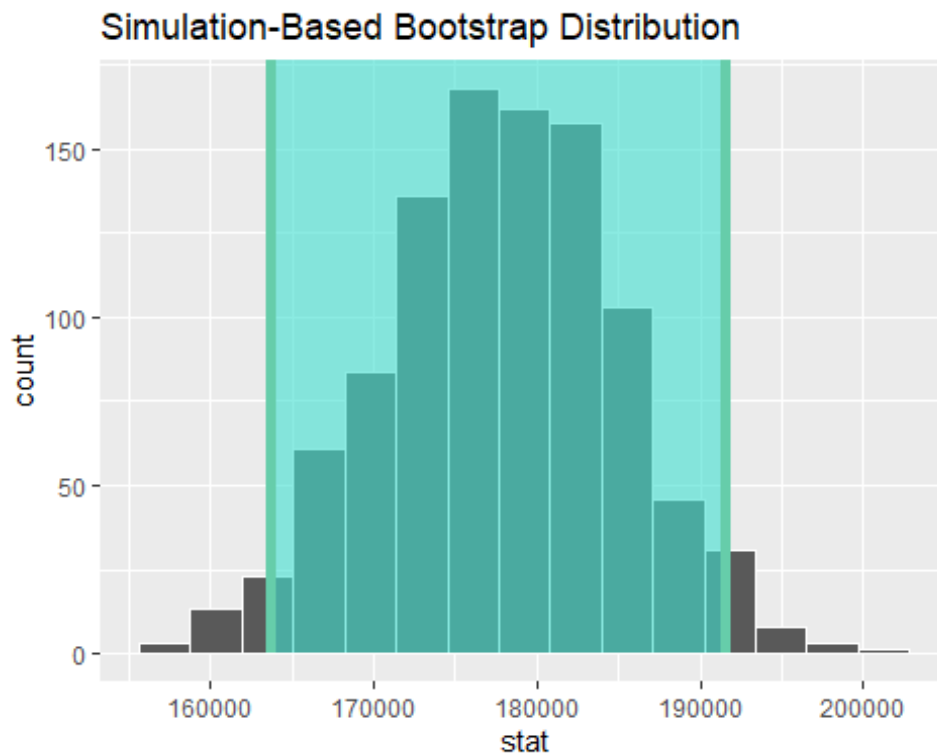
      <dbl>      <dbl>
1 163692. 191493.

sale_mean_ci %>%
  visualize() +
  shade_confidence_interval(endpoints = sale_ci, level=.95)

Warning in (function (mapping = NULL, data = NULL, stat = "identity", position
= "identity", : Ignoring unknown parameters: `level`

Warning in (function (mapping = NULL, data = NULL, stat = "identity", position
= "identity", : Ignoring unknown parameters: `level`

```



Homework Chapter 19

1. Review section 19.3 (the chapter review)
2. Suggested problems from textbook section 19.4 exercises: 3, 5, 6, 12, 15, 17, 22
3. Suggested tutorials:

[5 – Bootstrapping to estimate single means parameter](#)

[6 – Introducing the t-distribution](#)