Programming for Data Science with R

Part - III

Mohammad Wasiq

11/2/22

DSM - 1005

Programming for Data Science with R - III
DSM-1005

Table of Contents

I follow the book named **Statistical Inference via Data Science** *A ModernDive into R and the Tidyvers* by **Chester Ismay** and **Albert Y. Kim**

Teacher : **Prof. Athar Ali Khan Sir**

Writer : **Mohammad Wasiq** , $MS(Data\ Science)$

1 R programing fo Data Science

In this Script we learn the R programming for Data Science at intermediate level . We learn the following Topics

1. **Tidyverse**
- **Data Visualization** Using **ggplot2**
- **Data Wrangling** Using **dplyr**
- **Data Importing** & **Tidy Data**

2. **Data Modelling** with **moderndive**
- **Simple Regression**
- **Multiple Regression**

3. **Statistical Inference** with **infer**
- **Sampling**
- **Confidence Intervals**
- **Hypothesis Testing**
- **Inference for Regression**

Note :- We have already discuss the 1^{st} and II^{nd} Chapter in $Part - I$ and $Part - II$ respectively.

2 Statistical Inference with infer

3 Sampling

3.1 Sampling Theory

Some Important Defination :

1. **Population:** The aggrete of specified and well defined similar object. Ex- No. of students in AMU .
2. **Elementary Unit:** A well defined and identifiable object in the population on which some measurement are obtained . Ex- Every student in the population of students of AMU .

3. **Observation:** Any measurement made on an elementary unit. Ex- Age of Students in the population of student of AMU.
4. **Finite Population:** A population with countable no. of elementary unit. Ex- No. of stdents in AMU.
5. **Infinite Population:** A population with uncountable no. of elementary unit. Ex- No. of real values b/w 0 to 1.
6. **Parameter:** Any Characteristic of a population or numerical measurement based on all the unit of a population. Ex- Age of students of a class.
7. **Sample:** A finite subset of obsrevation drawn from a population . Ex- Age of 10 students out of a class of 100 students.
8. **Sampling Unit:** Each identifiable object in a sample. Ex- Each student in a sample of 10 students.
9. **Statistic:** Any numerical measurement based only on the sample unit.
10. **Sample Size:** No. of unit in a sample .
11. **Sampling distribution:** The probability distribution of a statistic .
12. **Statndard Error:** The standard deviation of the sampling distribution of a Statistic .
13. **Estimator:** A statistic or a function of random variable that estimates some unknown parameters. Ex- Sample mean ($\bar{x}$) is an estimator of population mean (μ).
14. **Estimate:** An estiamte is a single numerical value of a population calculation from a single sample.
15. **Sampling Frame:** A complete list of sampling units or a group of elementary units in the population which is used as the basis of section of a sample. Ex- A list of voters, A List of house-holders, etc.
16. **Sample Space:** Collection of all the possibles samples. It's denoted by 'S'.
17. **Sampling Error:** The error which arises due to only a sample being used to estimate the population parameters & draw inference about the population is termed as Sampling Error or Sampling Fluctuation. i.e. $S.E. \propto \frac{1}{\sqrt{n}}$,where n is is sample size. *Note*- i. When the sampling survey becomes a census survey, the sampling survey becomes zero(0). ii.Sampling Errors can not be controlled but it can be minimize.
18. **Non-Sampling Error:** The non-sampling error arises at follows stages: i.Failure to measure some of units in the selected sample. ii.Observational Errors due to defective measurement techniques. iii.Errors introduced in editing,coding and tabulating the result.
19. **Homogeneous Population:** Homogeneous population is one where every member has same value for the characteristic. Ex- Patients of specific disease.
20. **Hetrogeneous Population:** Hetrogeneous population is one where every member has not a same value for the characteristic. Ex- Patients in Hospital.
21. **Representative Sampling :** A sample is said to be a representative sample if it roughly looks like the population
22. **Biasd Sampling :** Biased sampling occurs if certain individuals or observations in a population have a higher chance of being included in a sample than others.

23. **Unbiased Sampling :** Sampling procedure is unbiased if every observation in a population had an equal chance of being sampled.
24. **Random Sampling :** Sampling procedure is random if we sample randomly from the population in an unbiased fashion.

3.2 Simple Random Sampling (SRS)

A procedure for selecting a sample of size n out of a finite population of size N in which each of the possible distinct samples has an equal chance of being selected is called *random sampling or Simple Random Sampling.* We may have two distinct types of simple random sampling as follows: i.Simple random sampling with replacement (SRSWR). ii.Simple random sampling without replacement(SRSWOR).

3.2.1 Simple Random Sampling with Replacement (SRSWR)

In sampling with replacement a unit is selected from the population consisting of N units, its content noted and then returned to the population before the next draw is made, and the process is repeated n times to give a sample of n units. In this method, at each draw, each of the N units of the population gets the same probability $\frac{1}{N}$ of being selected. Here the same unit of the population may occur more than once in the sample (order in which the sample units are obtained is regarded). There are N^n samples, and each has an equal probability $\frac{1}{N^n}$ being selected. **Note-** If order in which the sample units are obtained is ignored (unordered), then in such case the number of possible samples will be $\binom{N}{n} + N\left(1 + \binom{N-1}{1} + \binom{N-1}{2} + \cdots \binom{N-1}{n-2}\right)$

3.2.2 Simple Random Sampling without Replacement (SRSWOR)

Suppose the population consist of N units, then, in simple random sampling without replacement a unit is selected, its content noted and the unit is not returned to the population before next draw is made. The process is repeated n times to give a sample of n units. In this method at the r−th drawing, each of the N−r+1 units of the population gets the same probability $\frac{1}{N-r-1}$ of being included in the sample. Here any unit of the population cannot occur more than once in the sample (order is ignored). There are $\binom{N}{n}$ possible samples, and each such sample has an equal probability $\frac{1}{\binom{N}{n}}$ of being selected.

3.3 Stratified Random Sampling

The precision of an estimator of the population parameters (mean or total etc.) depends on the size of the sample and the variability or heterogeneity among the units of the population. If the population is very heterogeneous and considerations of cost limit the size of the sample, it may be found impossible to get a sufficiently precise estimate by taking a simple random sample from the entire population.

For this, one possible way to estimate the population mean or total with greater precision is to divide the population in several groups (sub-population or classes, these sub-

populations are non-overlapping) each of which is more homogenous than the entire population and draw a random sample of predetermined size from each one of the groups. The groups, into which the population is divided, are called *strata* or each group is called *stratum* and the whole procedure of dividing the population into the strata and then drawing a random sample from each one of the strata is called *stratified random sampling*. **For example**, to estimate the average income per household, it may be 8appropriate to group the households into two or more groups (strata) according to the rent paid by the households. The households in any stratum so form are likely to be more homogeneous with respect to income as compared to the whole population. Thus, the estimated income per household based on a stratified sample is likely to be more precise than that based on a simple random sample of the same size drawn from the whole population. **Ex-** To estimate the average marks of a graduate students in AMU, it may be appropriate to group the graduate into different groups according to their faculties. Thus the graduate in any stratum forms are likely to be homogeneous with respect to their scores as compare to the whole population of graduate. Thus the estimate based on stratified sampling are likely to be more precise than that based on Simple Random Sampling.

Principal reasons for stratification

- To gain in precision, divide a heterogeneous population into strata in such a way that each stratum is internally homogeneous.
- To accommodate administrative convenience (cost consideration), fieldwork is organized by strata, which usually results in saving in cost and effort.
- To obtain separate estimates for strata.
- We can accommodate different sampling plan in different strata.
- We can have data of known precision for certain subdivisions treating each subdivision as a population in its own right.

3.4 Systematic Random Sampling

A sampling technique in which only the first unit is selected with the help of random numbers and the rest get selected automatically according to some pre-determined pattern (in regular spacing pattern) is known as systematic random sampling. Suppose N units of the population are numbered from 1 to N in some order. Let N=nk , where n is the sample size, and $k = \frac{N}{n}$, being an integer and usually called the **sampling interval**. Draw a random number less than or equal to k , say i , and select the unit with the corresponding serial number and every k^th unit in the population thereafter. The resultant sample will contain the n units with serial numbers **i, i+k, i+2k,…..,i+(n−1)k** is called every k^th systematic sample and such a procedure termed linear systematic sampling. **Example:** There are 50 houses on a street. If a sample of size 5 is to be chosen, then $k = \frac{N}{n} = 10$, we select randomly one house out of the first ten, suppose we select 3rd and then take every 10th house after selected one, i.e. as $3^{rd}, 13^{th}, 23^{rd}, 33^{rd}, 43^{rd}$.

3.5 Circular Systematic Sampling

Linear systematic sampling suffers from the limitation that it cannot be used when the sampling interval $k = \frac{N}{n}$ is not an integer. The procedure to be followed is that of **circular systematic sampling**. In this method, select first item randomly out of N units and then take every k^th unit thereafter (where k is the nearest integer to N/n) in a cyclical manner until n sampling units are obtained. Example: Consider a population of size N=33.If a sample of size n=5 is to be drawn. Here the sampling interval $\frac{N}{n} = 6.6$ is a fractional number, and we have to go in for circular systematic sampling. Since the integer k nearest to 6.6 is 7, we select randomly one item out of N=33 , suppose 10th item be selected and then take every 7th item from that one, i.e. as $10^{th}, 17^{th}, 24^{th}, 31^{st}$.

3.6 Sampling of Tactile Prop Red

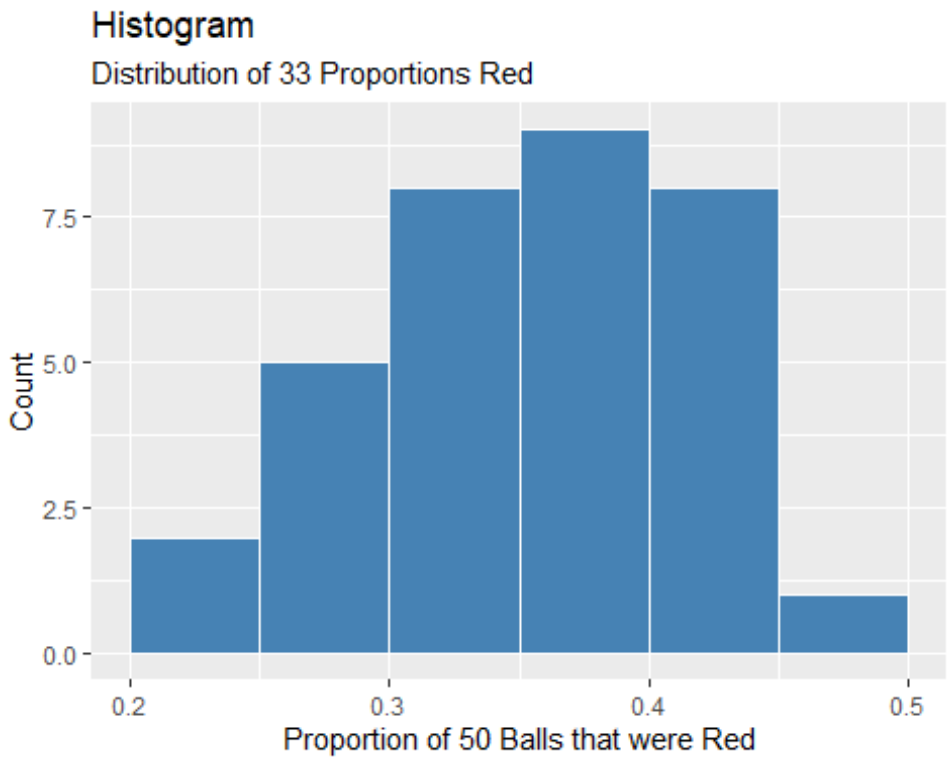
About the Data **tactile_prop_red**

A data frame of 33 rows representing different groups of students' samples of size n = 50 and 4 variables group = Group members replicate = Replicate number red_balls = Number of red balls sampled out of 50 prop_red = Proportion red balls out of 50

```
# Load the require package
library(tidyverse)
library(moderndive)

# Load the Data
data("tactile_prop_red")

# Dimension if Data
dim(tactile_prop_red)

## [1] 33  4

# Column Names of Data
names(tactile_prop_red)

## [1] "group"     "replicate" "red_balls" "prop_red"

# View(tactile_prop_red)
glimpse(tactile_prop_red)

## Rows: 33
## Columns: 4
## $ group     <chr> "Ilyas, Yohan", "Morgan, Terrance", "Martin, Thomas", ~
## $ replicate <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ red_balls <int> 21, 17, 21, 21, 18, 19, 19, 11, 15, 17, 16, 18, 17, 21~
## $ prop_red  <dbl> 0.42, 0.34, 0.42, 0.42, 0.36, 0.38, 0.38, 0.22, 0.30, ~
```

Task : Make a Histogram of red prop.

```
ggplot(tactile_prop_red , aes(x = prop_red)) +
  geom_histogram(binwidth = 0.05 , boundary = 0.4 , color = "white" , fill =
"steelblue") +
  labs(x = "Proportion of 50 Balls that were Red" , y = "Count" , title =
"Histogram" , subtitle = "Distribution of 33 Proportions Red")
```



(LC 7.1) Why was it important to mix the bowl before we sampled the ball ? **Ans :** For Random Sampling

(LC 7.2) Why is it that our 33 groups of friends did not all have the same numbers of balls that were red out of 50, and hence different proportions red? **Ans :** Because not all pairs have the same portion of the population of the balls, so each pair has a different sampled balls with different color compositions.

Task : Load the Dataset **bowl** from *moderndive* package .

```
data("bowl")
names(bowl)

## [1] "ball_ID" "color"

dim(bowl)

## [1] 2400    2

glimpse(bowl)

## Rows: 2,400
## Columns: 2
```

```
## $ ball_ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ color   <chr> "white", "white", "white", "red", "white", "white", "red~
```

Task : Take a random sample of size **50** from the *bowl* dataset.

rep_sample_n( ) function from *moderndive* allows us to take repeated, or replicated, samples of size n .

```
virtual_shovel <- bowl %>%
  rep_sample_n(size = 50)

# Show the head of sample
virtual_shovel %>% head()
```

```
## # A tibble: 6 x 3
## # Groups:   replicate [1]
##   replicate ball_ID color
##       <int>   <int> <chr>
## 1         1    1664 red
## 2         1     696 white
## 3         1    2184 white
## 4         1    1677 white
## 5         1    1601 white
## 6         1    1705 white
```

Task : Make new column and named *is_red* which contain only *red* balls .

```
virtual_shovel %>%
  mutate(is_red = (color == "red")) %>%
  head()
```

```
## # A tibble: 6 x 4
## # Groups:   replicate [1]
##   replicate ball_ID color is_red
##       <int>   <int> <chr> <lgl>
## 1         1    1664 red   TRUE
## 2         1     696 white FALSE
## 3         1    2184 white FALSE
## 4         1    1677 white FALSE
## 5         1    1601 white FALSE
## 6         1    1705 white FALSE
```

Task : Count ow many balls are *red* ?

```
virtual_shovel %>%
  mutate(is_red  = (color == "red")) %>%
  summarise(num_red = sum(is_red))
```

```
## # A tibble: 1 x 2
##   replicate num_red
##       <int>   <int>
## 1         1      13
```

Task : Make a column for *Red balls* and name that *prop_red* .

```
virtual_shovel %>%
  mutate(is_red = (color == "red")) %>%
  summarise(num_red = sum(is_red)) %>%
  mutate(prop_red = num_red / 50)
```

```
## # A tibble: 1 x 3
##   replicate num_red prop_red
##       <int>   <int>    <dbl>
## 1         1      13     0.26
```

There are *34%* red balls in ou sample (virtual_shovek)

Task : Repeat the above process **33** times from *bowl* data.

```
virtual_samples <- bowl %>%
  rep_sample_n(size = 50 , reps = 33)

head(virtual_samples)
```

```
## # A tibble: 6 x 3
## # Groups:   replicate [1]
##   replicate ball_ID color
##       <int>   <int> <chr>
## 1         1     746 white
## 2         1     181 white
## 3         1    1172 red
## 4         1    2389 white
## 5         1     257 white
## 6         1    1531 red
```

```
dim(virtual_samples)  # 50 * 33 = 1650
```

```
## [1] 1650    3
```

Task : Take virtual_samples and compute the resulting 33 proportions red and also group_by replicate .

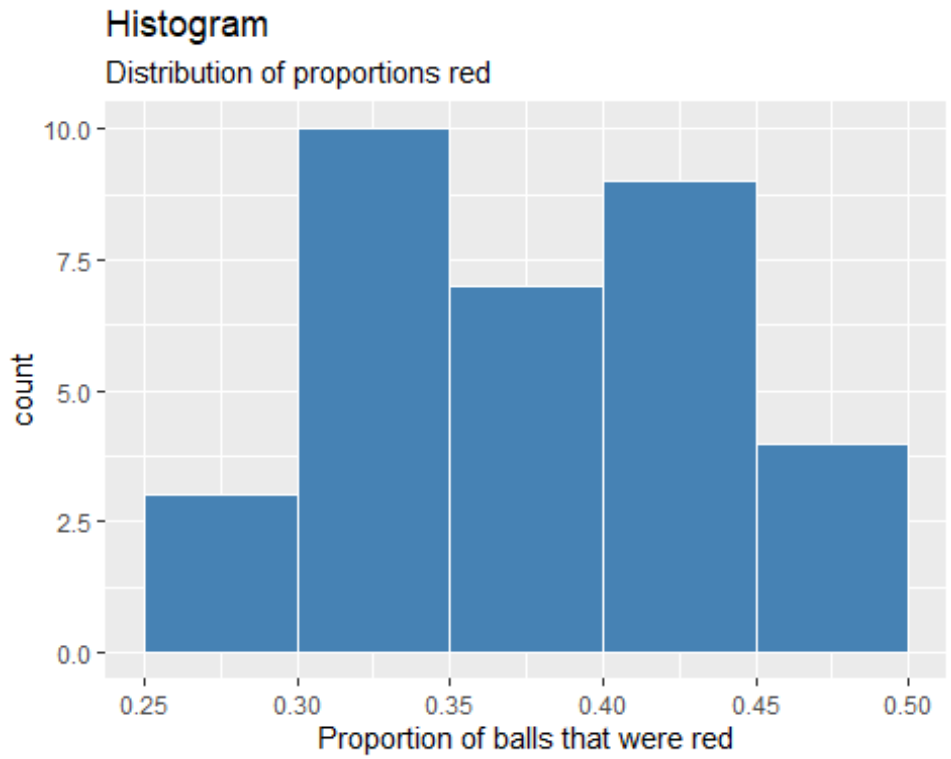
```
virtual_prop_red <- virtual_samples %>%
  group_by(replicate) %>%
  summarize(red = sum(color == "red")) %>%
  mutate(prop_red = red / 50)

virtual_prop_red %>% head()
```

```
## # A tibble: 6 x 3
##   replicate   red prop_red
##       <int> <int>    <dbl>
## 1         1    17     0.34
## 2         2    18     0.36
## 3         3    24     0.48
```

```
## 4          4     17      0.34
## 5          5     18      0.36
## 6          6     24      0.48
```

Task : Make a Histogram of prop_red.

```
ggplot(virtual_prop_red, aes(x = prop_red)) +
  geom_histogram(binwidth = 0.05, boundary = 0.4, color = "white" , fill =
"steelblue") +
  labs(x = "Proportion of balls that were red", title = "Histogram" ,
subtitle =  "Distribution of proportions red")
```



Our Sample is **-vely Skewed** .

(LC 7.3) : Why couldn't we study the effects of sampling variation when we used the virtual shovel only once? Why did we need to take more than one virtual sample (in our case 33 virtual samples)? **Ans :** If we use the virtual shovel only once, we only get one sample of the population. We need to take more than one virtual sample to get a range of proportions.

Task : Using the virtual shovel 1000 times

```
virtual_samples <- bowl %>%
  rep_sample_n(size = 50 , reps = 1000)

virtual_samples %>% head() # 50 * 1000 = 50000

## # A tibble: 6 x 3
## # Groups:   replicate [1]
```

```
##   replicate ball_ID color
##       <int>   <int> <chr>
## 1         1    1201 white
## 2         1    1628 red
## 3         1    1555 red
## 4         1     704 red
## 5         1    1820 red
## 6         1     158 red
```

Task : Take virtual_samples and compute the resulting 33 proportions red and also group_by replicate .

```
virtual_prop_red <- virtual_samples %>%
  group_by(replicate) %>%
  summarise(red = sum(color == "red")) %>%
  mutate(prop_red = red / 50)

virtual_prop_red %>% head()
```

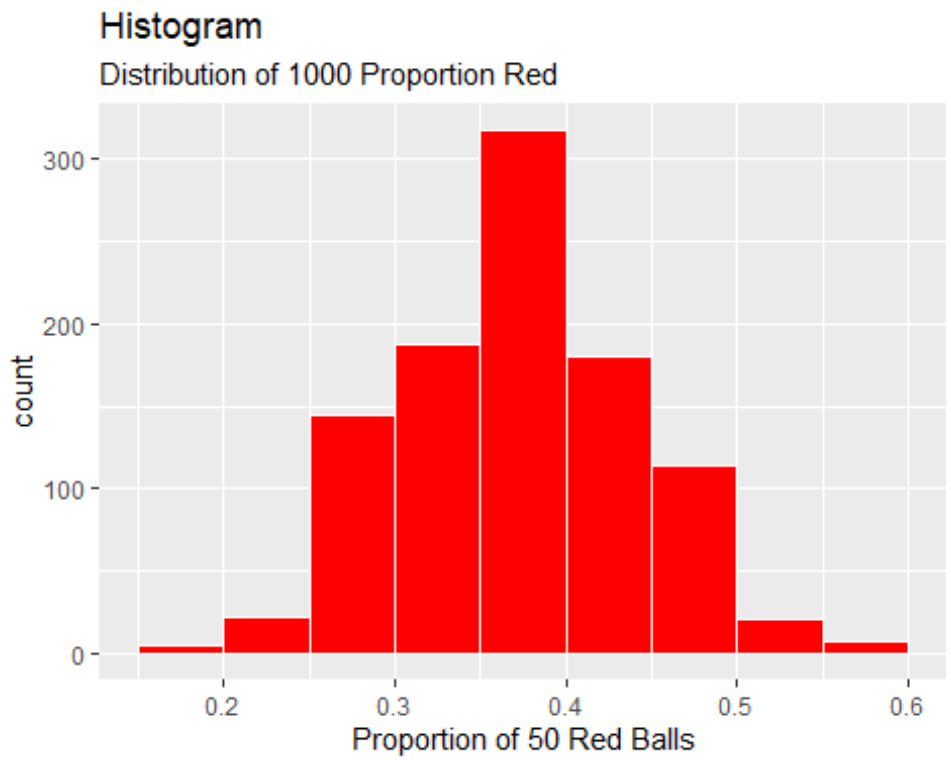
```
## # A tibble: 6 x 3
##   replicate   red prop_red
##       <int> <int>    <dbl>
## 1         1    19     0.38
## 2         2    22     0.44
## 3         3    22     0.44
## 4         4    24     0.48
## 5         5    17     0.34
## 6         6    20     0.4
```

```
virtual_prop_red %>% dim()
```

```
## [1] 1000    3
```

Task : Make a Histogram

```
ggplot(virtual_prop_red , aes(x = prop_red)) +
  geom_histogram(binwidth = 0.05 , boundary = 0.4 , col = "white" , fill =
"red") + labs(x = "Proportion of 50 Red Balls" , title = "Histogram" ,
subtitle = "Distribution of 1000 Proportion Red")
```



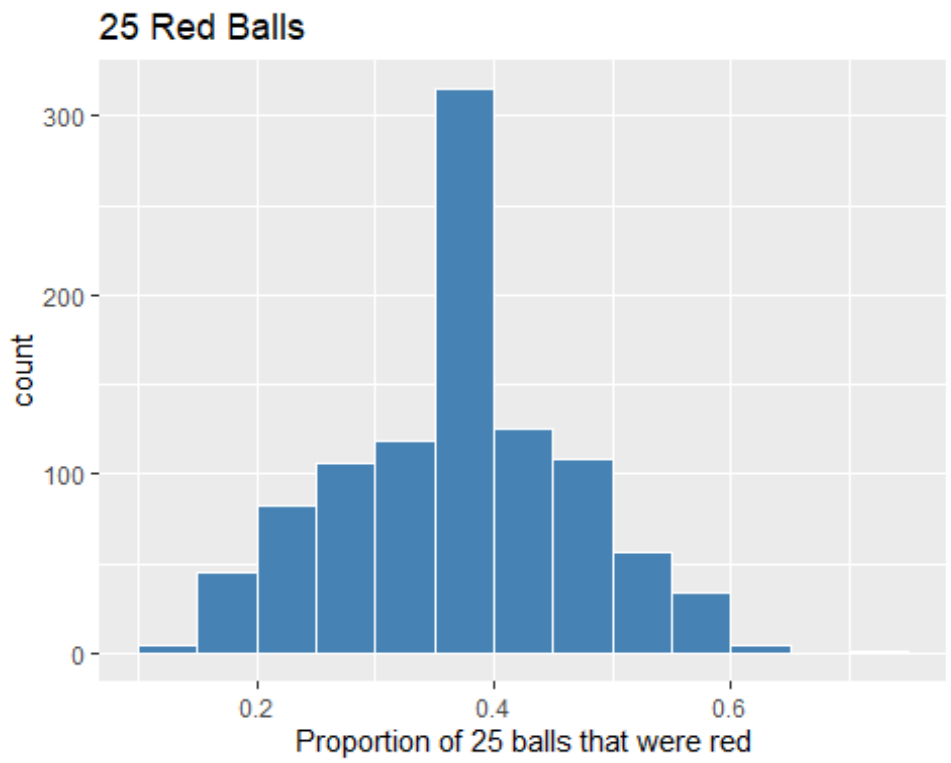
(LC 7.4) : Why did we not take 1000 "tactile" samples of 50 balls by hand?
Ans : That would be way too much repeated work.

(LC 7.5) : Looking at Figure ??, would you say that sampling 50 balls where 30% of them were red is likely or not? What about sampling 50 balls where 10% of them were red?
Ans : According to the Figure, less than 150 out of the 1000 counts were 30% red. So I would say that sampling 50 balls where 30% of them were red is not very likely. Almost no count was only 10% red, so sampling 50 balls where 10% of them were red is extremely unlikely.

Task : Virtually use the appropriate shovel to generate 1000 samples with size balls with size set to 25 .

```
# Segment 1: sample size = 25

# 1.a) Virtually use shovel 1000 times
virtual_samples_25 <- bowl %>%
  rep_sample_n(size = 25, reps = 1000)

# 1.b) Compute resulting 1000 replicates of proportion red
virtual_prop_red_25 <- virtual_samples_25 %>%
  group_by(replicate) %>%
  summarize(red = sum(color == "red")) %>%
  mutate(prop_red = red / 25)
```

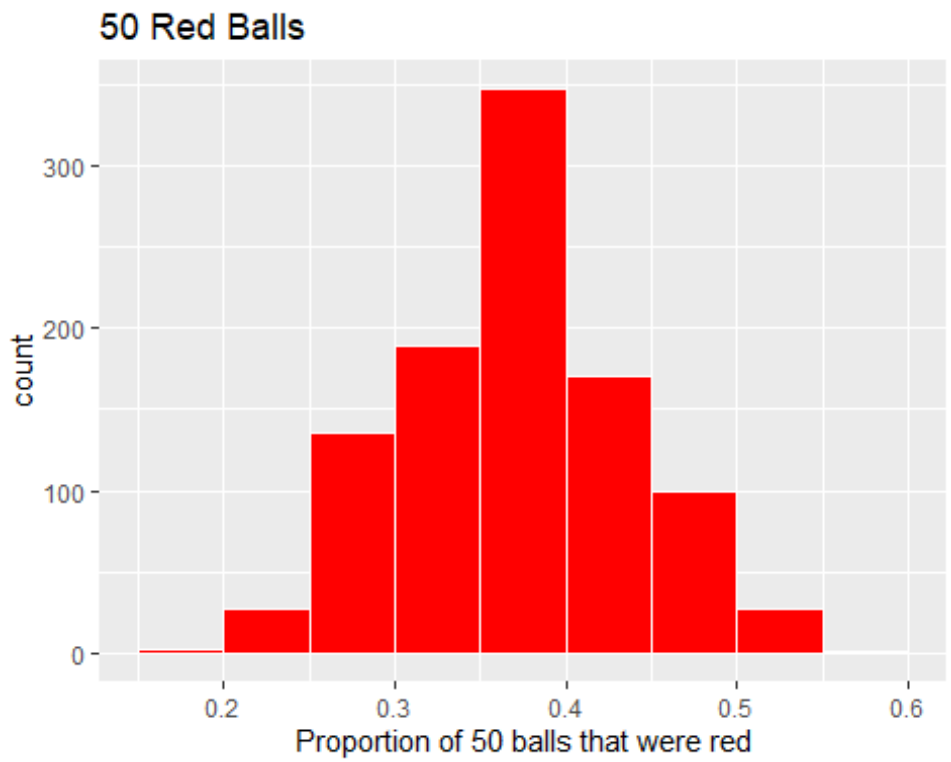
```r
# 1.c) Plot distribution via a histogram
p1 <- ggplot(virtual_prop_red_25, aes(x = prop_red)) +
  geom_histogram(binwidth = 0.05, boundary = 0.4, color = "white" , fill =
"steelblue") +
  labs(x = "Proportion of 25 balls that were red", title = "25 Red Balls")

p1
```



Task : Compute the resulting 1000 replicates of the proportion of the shovel's balls that are red with set of *50* balls.

```r
# Segment 2: sample size = 50

# 2.a) Virtually use shovel 1000 times
virtual_samples_50 <- bowl %>%
  rep_sample_n(size = 50, reps = 1000)

# 2.b) Compute resulting 1000 replicates of proportion red
virtual_prop_red_50 <- virtual_samples_50 %>%
  group_by(replicate) %>%
  summarize(red = sum(color == "red")) %>%
  mutate(prop_red = red / 50)



# 2.c) Plot distribution via a histogram
```

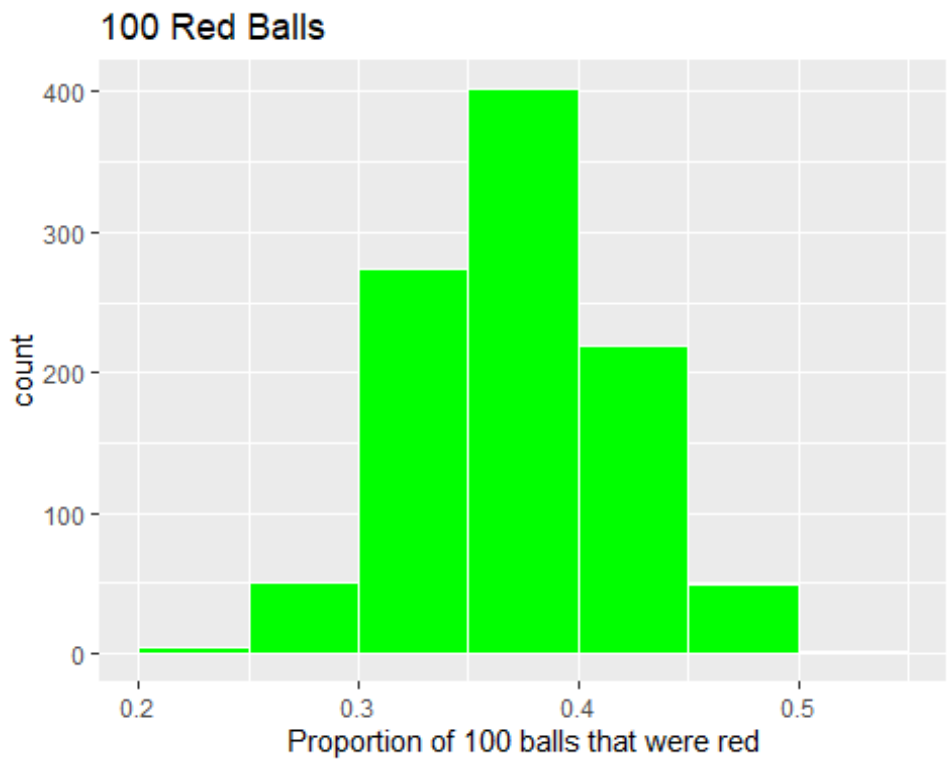
```r
p2 <- ggplot(virtual_prop_red_50, aes(x = prop_red)) +
  geom_histogram(binwidth = 0.05, boundary = 0.4, color = "white" , fill =
"red") +
  labs(x = "Proportion of 50 balls that were red", title = "50 Red Balls")

p2
```



Task : Visualize the distribution of these 1000 proportions red using a histogram with the set of *100* balls.

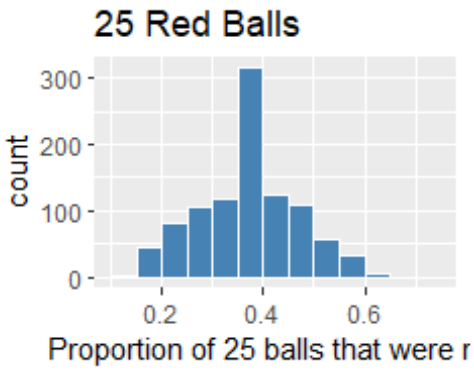
```r
# 3.a) Virtually using shovel with 100 slots 1000 times
virtual_samples_100 <- bowl %>%
  rep_sample_n(size = 100, reps = 1000)

# 3.b) Compute resulting 1000 replicates of proportion red
virtual_prop_red_100 <- virtual_samples_100 %>%
  group_by(replicate) %>%
  summarize(red = sum(color == "red")) %>%
  mutate(prop_red = red / 100)

# 3.c) Plot distribution via a histogram
p3 <- ggplot(virtual_prop_red_100, aes(x = prop_red)) +
  geom_histogram(binwidth = 0.05, boundary = 0.4, color = "white" , fill =
"green") +
  labs(x = "Proportion of 100 balls that were red", title = "100 Red Balls")
```

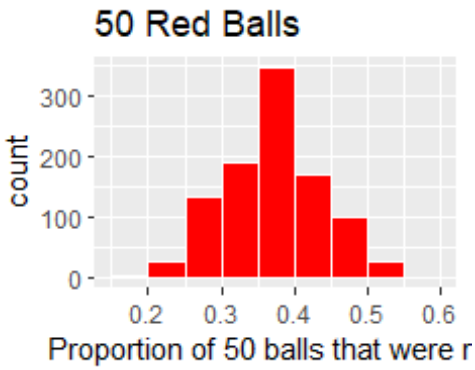
```
p3
```

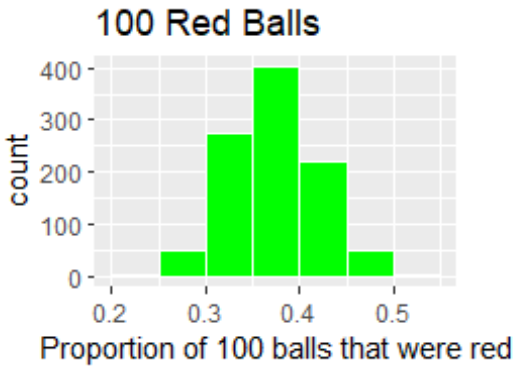


Task : Plot all three Histogeam togeter .

```
library(cowplot)
plot_grid(p1 , p2 , p3)
```







Task : Find the *sd* above three samples .

```
# n = 25
virtual_prop_red_25 %>%
  summarize(sd = sd(prop_red))
```

```
## # A tibble: 1 x 1
##       sd
##    <dbl>
## 1 0.0977
```

```
# n = 50
virtual_prop_red_50 %>%
  summarize(sd = sd(prop_red))
```

```
## # A tibble: 1 x 1
##       sd
##    <dbl>
## 1 0.0663
```

```
# n = 100
virtual_prop_red_100 %>%
summarize(sd = sd(prop_red))
```

```
## # A tibble: 1 x 1
##       sd
##    <dbl>
## 1 0.0467
```

```
Sample <- c(25 , 50 , 100)
SD_of_Balls <- c(0.094 , 0.068 , 0.048)
data.frame(Sample , SD_of_Balls)
```

```
##   Sample SD_of_Balls
## 1     25       0.094
## 2     50       0.068
## 3    100       0.048
```

We can see that as the sample size increases then the sd decrese.

$$SD \propto \frac{1}{n}$$

(LC 7.8) In the case of our bowl activity, what is the population parameter? Do we know its value?
Ans : The population parameter in the case of our bowl activity is the population proportion of the red balls in the bowl. Unless we know the exact number of red balls in the bowl, we won't know the value of this population proportion.

Task : Count the Red an Non - ed balls in Bowl data .

```
bowl %>%
summarize(sum_red = sum(color == "red"),
sum_not_red = sum(color != "red"))
```

```
## # A tibble: 1 x 2
##   sum_red sum_not_red
##     <int>       <int>
## 1     900        1500
```

3.6.1 Cental Limit Theorem

when sample means are based on larger and larger sample sizes, the sampling distribution of these sample means becomes both more and more normally shaped and more and more narrow.

In other words, their sampling distribution increasingly follows a normal distribution and the variation of these sampling distributions gets smaller, as quantified by their standard errors.

i.e. If $x_i \sim (\mu, \sigma^2)$ for $i = 1, \cdots, n \Rightarrow \bar{x} \xrightarrow{L} N\left(\mu, \frac{\sigma^2}{n}\right)$. This is known as . Note that in even if parent population is not known the sampling distribution of mean can be approximated by a normal distribution $N(\mu, \sigma^2/n)$.

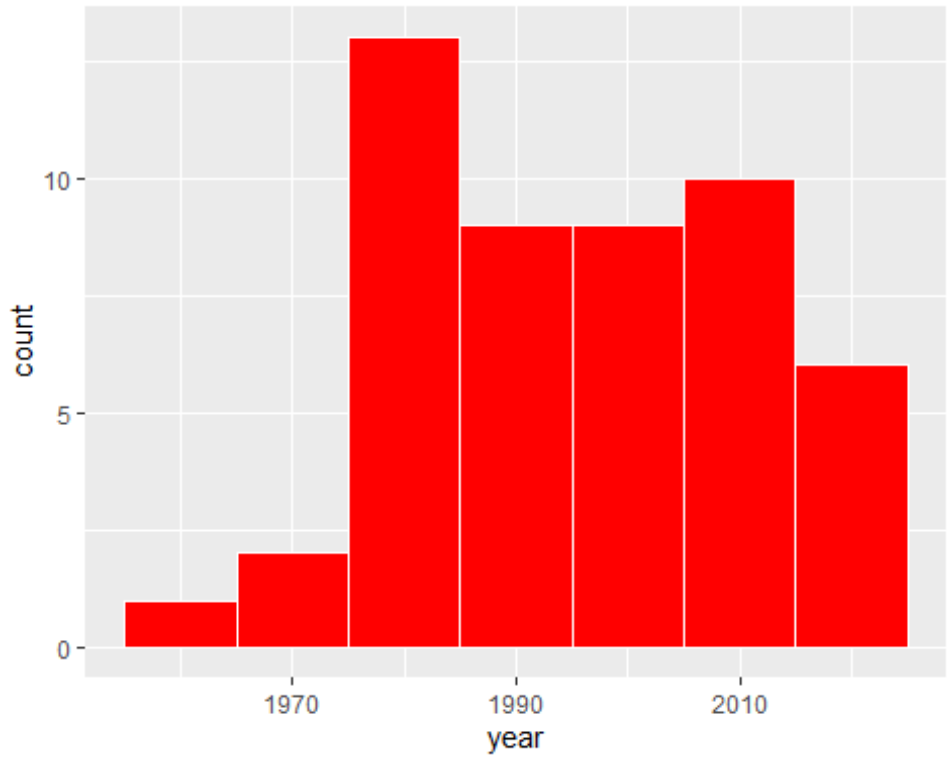
4 Bootstrapping and Confidence Intervals

We are working with **pennies_sample** .

```
library(tidyverse)
library(moderndive)
library(infer)

# Load the Data
data("pennies_sample")
dim(pennies_sample)

## [1] 50  2

names(pennies_sample)

## [1] "ID"   "year"

pennies_sample %>% head()

## # A tibble: 6 x 2
##      ID  year
##   <int> <dbl>
## 1     1  2002
## 2     2  1986
## 3     3  2017
## 4     4  1988
## 5     5  2008
## 6     6  1983
```

Task : Make a Histogram of year.

```
ggplot(pennies_sample , aes(x = year)) +
  geom_histogram(binwidth = 10 , col = "white" , fill = "red")
```



Task : Find the Mean of Year .

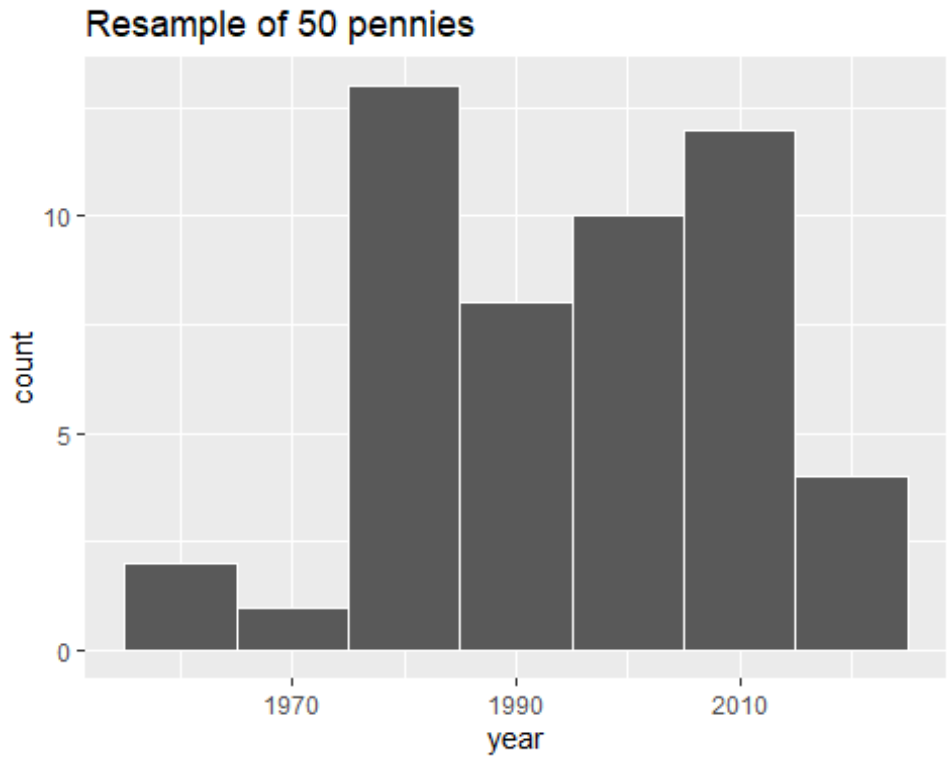
```
m <- pennies_sample %>%
  summarise(mean_year = mean(year))
m

## # A tibble: 1 x 1
##   mean_year
##       <dbl>
## 1     1995.

round(m , 0)

## # A tibble: 1 x 1
##   mean_year
##       <dbl>
## 1      1995
```

4.1 Resampling

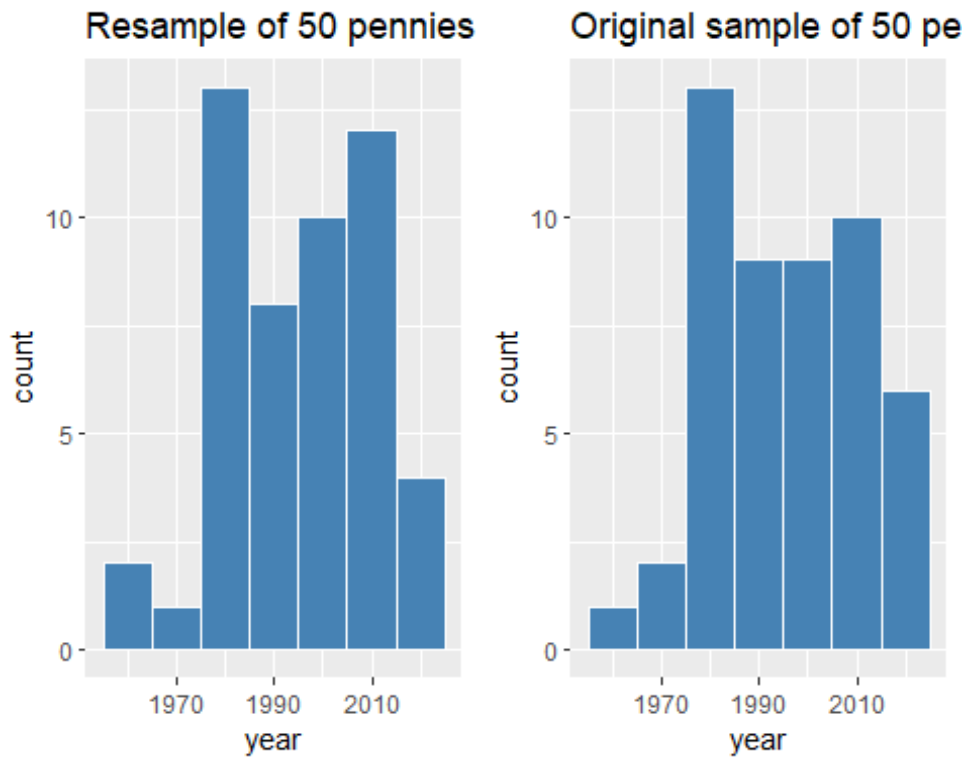
```
pennies_resample <- tibble(
year = c(1976, 1962, 1976, 1983, 2017, 2015, 2015, 1962, 2016, 1976, 2006,
1997, 1988, 2015, 2015, 1988, 2016, 1978, 1979, 1997, 1974, 2013, 1978, 2015,
2008, 1982, 1986, 1979, 1981, 2004, 2000, 1995, 1999, 2006, 1979, 2015, 1979,
1998, 1981, 2015, 2000, 1999, 1988, 2017, 1992, 1997, 1990, 1988, 2006, 2000)
)
```

Task : Compare with Resampled with Originals .

```r
ggplot(pennies_resample, aes(x = year)) +
  geom_histogram(binwidth = 10, color = "white") +
  labs(title = "Resample of 50 pennies")
```




```r
# Resampled Graph
p1 <- ggplot(pennies_resample, aes(x = year)) +
  geom_histogram(binwidth = 10, color = "white" , fill = "steelblue") +
  labs(title = "Resample of 50 pennies")

# Original graph
p2 <- ggplot(pennies_sample, aes(x = year)) +
  geom_histogram(binwidth = 10, color = "white" , fill = "steelblue") +
  labs(title = "Original sample of 50 pennies")

plot_grid(p1 , p2)
```



Task : Find the Mean of Year .

```
m <- pennies_resample %>%
  summarise(mean_year = mean(year))
m
```

```
## # A tibble: 1 x 1
##   mean_year
##       <dbl>
## 1     1995.
```

```
round(m , 0)
```

```
## # A tibble: 1 x 1
##   mean_year
##       <dbl>
## 1      1995
```

pennies_resamples Data .

```
data("pennies_resamples")
dim(pennies_resamples)
```

```
## [1] 1750    3
```

```
names(pennies_resamples)
```

```
## [1] "replicate" "name"      "year"
```

```
head(pennies_resamples)
```

```
## # A tibble: 6 x 3
## # Groups:   name [1]
##   replicate name     year
##       <int> <chr>   <dbl>
## 1         1 Arianna  1988
## 2         1 Arianna  2002
## 3         1 Arianna  2015
## 4         1 Arianna  1998
## 5         1 Arianna  1979
## 6         1 Arianna  1971
```

Task : Compute the mean of year by name .

```
resampled_means <- pennies_resamples %>%
  group_by(name) %>%
  summarise(mean_year = mean(year))

resampled_means %>% dim()
```

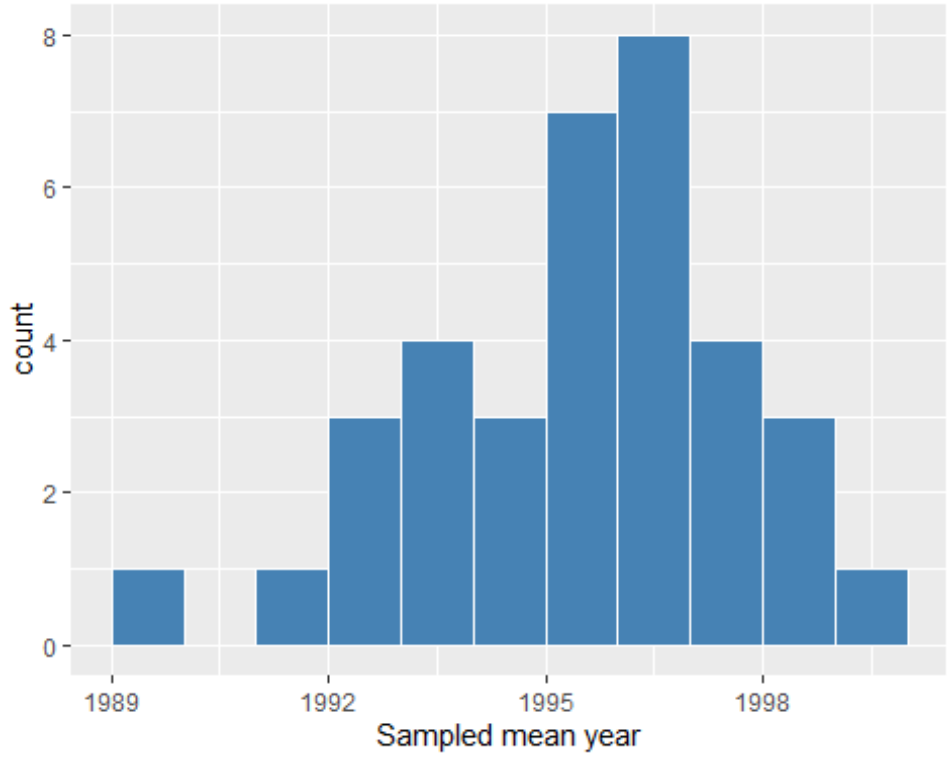
```
## [1] 35  2
```

```
resampled_means %>% head()
```

```
## # A tibble: 6 x 2
##   name      mean_year
##   <chr>         <dbl>
## 1 Arianna       1992.
## 2 Artemis       1996.
## 3 Bea           1996.
## 4 Camryn        1997.
## 5 Cassandra     1991.
## 6 Cindy         1995.
```

Task : Make a Histogram of above mean .

```
ggplot(resampled_means, aes(x = mean_year)) +
  geom_histogram(binwidth = 1, color = "white", boundary = 1990 , fill =
"steelblue") +
  labs(x = "Sampled mean year")
```



4.2 Computer Simulation of Resampling

Task : Take the sample of size *50 without replacement* and *with replacement* from the data *bowl* and *pennies_sample*.

```
# Sampling without Replacement
virtual_shovel <- bowl %>%
  rep_sample_n(size = 50)

dim(virtual_shovel)

## [1] 50  3

virtual_shovel %>% head()

## # A tibble: 6 x 3
## # Groups:   replicate [1]
##   replicate ball_ID color
##       <int>   <int> <chr>
## 1         1    1917 red
## 2         1     464 white
## 3         1     392 red
## 4         1    2081 white
## 5         1    1146 white
## 6         1    1587 white
```

```r
# Sampling with replacement
virtual_resample <- pennies_sample %>%
  rep_sample_n(size = 50 , replace = TRUE)

virtual_resample %>% dim()
```

```
## [1] 50  3
```

```r
virtual_resample %>% head(5)
```

```
## # A tibble: 5 x 3
## # Groups:   replicate [1]
##   replicate    ID  year
##       <int> <int> <dbl>
## 1         1    25  1979
## 2         1     1  2002
## 3         1     9  2004
## 4         1    13  2015
## 5         1    36  2015
```

Task : Find the Mean of *year* from above sample .

```r
# Mean of Virtual resample
virtual_resample %>%
  summarise(Mean = mean(year) ,
            Rounded_Mean = round(Mean , 0))
```

```
## # A tibble: 1 x 3
##   replicate  Mean Rounded_Mean
##       <int> <dbl>        <dbl>
## 1         1 1996.         1996
```

Virtually Resampling 35 times

```r
virtual_resamples <- pennies_resample %>%
  rep_sample_n(size = 50 , replace = T , reps = 35)

virtual_resamples %>% dim()
```

```
## [1] 1750    2
```

```r
virtual_resamples %>% head()
```

```
## # A tibble: 6 x 2
## # Groups:   replicate [1]
##   replicate  year
##       <int> <dbl>
## 1         1  1978
## 2         1  2016
## 3         1  2017
## 4         1  1978
```

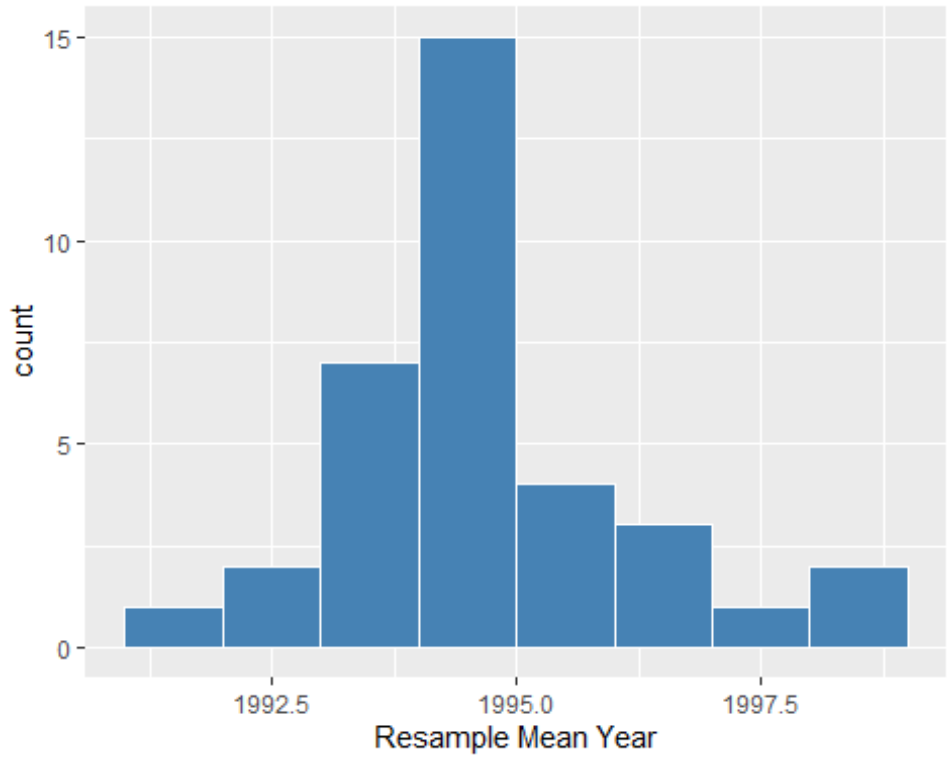
```
## 5         1  2000
## 6         1  2006
```

Task : Find the *virtual_resample* **mean** plot its **Histogram**.

```
virtual_resampled_means <- virtual_resamples %>%
  group_by(replicate) %>%
  summarise(Mean_Year = mean(year))

virtual_resampled_means %>%
  round(0) %>%
  head()
```

```
## # A tibble: 6 x 2
##   replicate Mean_Year
##       <dbl>     <dbl>
## 1         1      1993
## 2         2      1995
## 3         3      1994
## 4         4      1995
## 5         5      1995
## 6         6      1995
```

```
# Histogram
ggplot(virtual_resampled_means , aes(x = Mean_Year)) +
  geom_histogram(binwidth = 1 , col = "white" , fill = "steelblue" , boundary
= 1990) +
  labs(x = "Resample Mean Year")
```

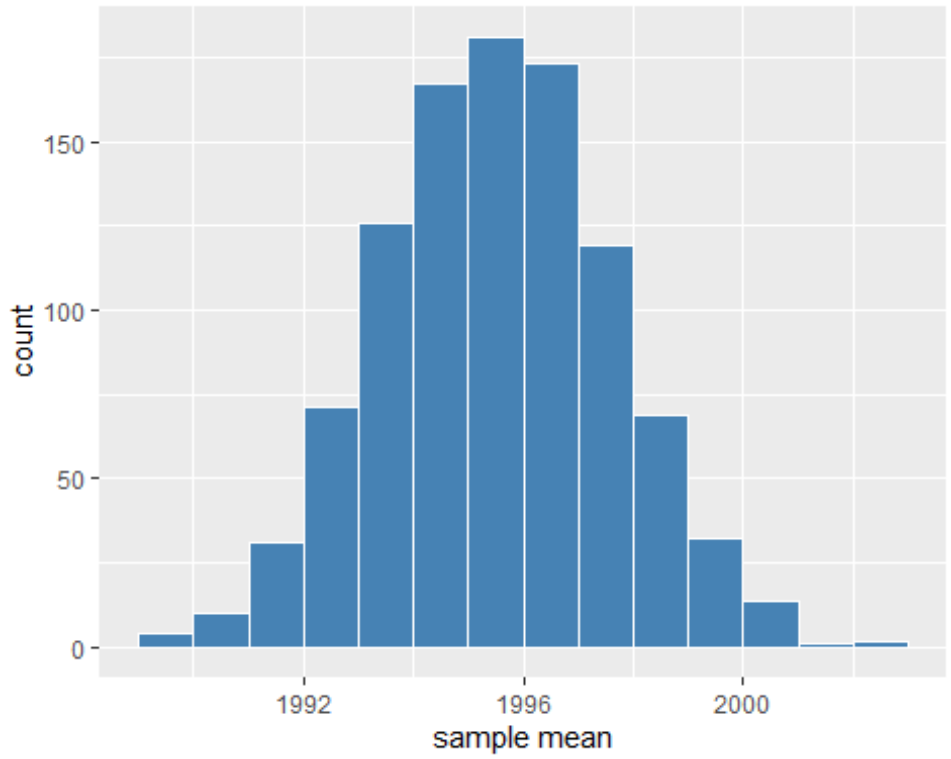


Repeat Above Process upto 1000 times :

```
virtual_resampled_means <- pennies_sample %>%
  rep_sample_n(size = 50, replace = TRUE, reps = 1000) %>%
  group_by(replicate) %>%
  summarize(mean_year = mean(year))

virtual_resampled_means %>% head()
```

```
## # A tibble: 6 x 2
##   replicate mean_year
##       <int>     <dbl>
## 1         1     1996
## 2         2     1992.
## 3         3     1998.
## 4         4     1992.
## 5         5     1997.
## 6         6     1996.
```

```
# Histogram
ggplot(virtual_resampled_means, aes(x = mean_year)) +
  geom_histogram(binwidth = 1, color = "white", boundary = 1990 , fill =
"steelblue") +
  labs(x = "sample mean")
```



Task : Find the overall mean of Year

```
virtual_resampled_means %>%
  summarise(Overall_Year_Mean = mean(mean_year)) %>% round(0)
```

```
## # A tibble: 1 x 1
##   Overall_Year_Mean
##               <dbl>
## 1              1996
```

(LC 8.1): What is the chief difference between a bootstrap distribution and a sampling distribution?
Ans : A bootstrap sample is a smaller sample that is "bootstrapped" from a larger sample. Bootstrapping is a type of resampling where large numbers of smaller samples of the same size are repeatedly drawn, with replacement, from a single original sample.

(LC 8.2) : Looking at the bootstrap distribution for the sample mean in between what two values would you say most values lie?
Ans : Most values lie in 1990 amd 2000.

4.3 Understanding Confidence Intervals

Confidence Intervals : The confidence interval is the range of values that you expect your estimate to fall between a certain percentage of the time if you run your experiment again or re-sample the population in the same way. $C.I. = (1 - \alpha)$ So if you use an α value of $p < 0.05$ for statistical significance, then your confidence level would be **1−0.05 = 0.95,** or **95%** .

$$C.I = \bar{X} \pm Z^* \frac{\sigma}{\sqrt{n}} \Rightarrow C.I. = mean \pm Crtical\_Value \frac{\sigma}{\sqrt{n}}$$

Workflow

Task 1 : Resample size = 50 pennies with replacement from the original sample of 50 *pennies_sample* data .

```
pennies_sample %>%
  rep_sample_n(size = 50 , replace = T , reps = 1000) %>%
  head()  # 50 * 1000 = 50,000  ; dim() = 50,000 * 3
```

```
## # A tibble: 6 x 3
## # Groups:   replicate [1]
##   replicate    ID  year
##       <int> <int> <dbl>
## 1         1    11  1994
## 2         1    31  2013
## 3         1     6  1983
## 4         1    39  2015
## 5         1    48  1988
## 6         1    45  1997
```

Task 2 : Group observations/rows together by the replicate variable of above data .

```
pennies_sample %>%
  rep_sample_n(size = 50 , replace = T , reps = 1000) %>%
  group_by(replicate) %>%
  head()
```

```
## # A tibble: 6 x 3
## # Groups:   replicate [1]
##   replicate    ID  year
##       <int> <int> <dbl>
## 1         1    12  1995
## 2         1    11  1994
## 3         1    22  1976
## 4         1    12  1995
## 5         1    21  1981
## 6         1    34  1985
```

Task 3 : Find the mean of year of above data .

```
pennies_sample %>%
  rep_sample_n(size = 50 , replace = T , reps = 1000) %>%
  group_by(replicate) %>%
  summarise(mean_year = mean(year) ,
            rounded_mean = round(mean_year , 0)) %>%
  head(5)
```

```
## # A tibble: 5 x 3
##   replicate mean_year rounded_mean
```

```
##       <int>     <dbl>        <dbl>
## 1         1     1990.         1990
## 2         2     1996.         1996
## 3         3     1998.         1998
## 4         4     1997.         1997
## 5         5     1996.         1996
```

4.3.0.1 infer package workflow

The **dplyr** package provides functions with verb-like names to perform *data wrangling*, the **infer** package provides functions with intuitive verb-like names to perform *statistical inference* .

Task : Computed the value of the sample mean $\bar{x}$ using the dplyr function summarize() .

```
pennies_sample %>%
  summarise(stat = mean(year) ,
            Rounded_stat = round(stat ,0))
```

```
## # A tibble: 1 x 2
##    stat Rounded_stat
##   <dbl>        <dbl>
## 1 1995.         1995
```

We can also do this using *infer* functions **specify()** and **calculate()**.

4.3.0.1.1 specify()

```
pennies_sample %>%
  specify(response = year) %>%
  calculate(stat = "mean")
```

```
## Response: year (numeric)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 1995.
```

Task : Frame of the 50 pennies sampled from the bank, the variable of interest is year.

```
pennies_sample %>%
  specify(response = year) %>% head(5)
```

```
## Response: year (numeric)
## # A tibble: 5 x 1
##    year
##   <dbl>
## 1  2002
## 2  1986
## 3  2017
## 4  1988
## 5  2008
```

Note : *specify()* is same as *group_by()* . We can also specify which variables will be the focus of our statistical inference using a *formula = y ~ x* .

```
pennies_sample %>%
  specify(formula = year ~ NULL) %>%
  head()

## Response: year (numeric)
## # A tibble: 6 x 1
##    year
##   <dbl>
## 1  2002
## 2  1986
## 3  2017
## 4  1988
## 5  2008
## 6  1983
```

There is no *explanatory* variable so, we use *NULL* above .

4.3.0.1.2 generate()

generate() is same as **rep_sample_n()**

Task : Repeat *resample* 1000 times of *pennies_sample* data .

```
library(tidyverse)
library(moderndive)
library(infer)

data(pennies_sample)
pennies_sample %>%
  specify(response = year) %>%
  generate(reps = 1000 , type = "bootstrap") %>%
  head(5)  # 50 * 1000 = 50,000

## # A tibble: 5 x 2
## # Groups:   replicate [1]
##   replicate  year
##       <int> <dbl>
## 1         1  2015
## 2         1  1990
## 3         1  2013
## 4         1  1981
## 5         1  1985
```

4.3.0.1.3 calculate()

Calculate() is same as **summarise()*

We can find **“mean”** , **“median” ,“sum”, “sd”** (standard deviation), and **“prop”** (proportion)

Task : Find the mean of above sample .

```
bootscap_distribution <- pennies_sample %>%
  specify(response = year) %>%
  generate(reps = 1000) %>%
  calculate(stat = "mean")

## Setting `type = "bootstrap"` in `generate()`.

bootscap_distribution %>% head(5)

## Response: year (numeric)
## # A tibble: 5 x 2
##   replicate  stat
##       <int> <dbl>
## 1         1 1995.
## 2         2 1993.
## 3         3 1996.
## 4         4 1994.
## 5         5 1996.
```

Comparing with Original Workflow

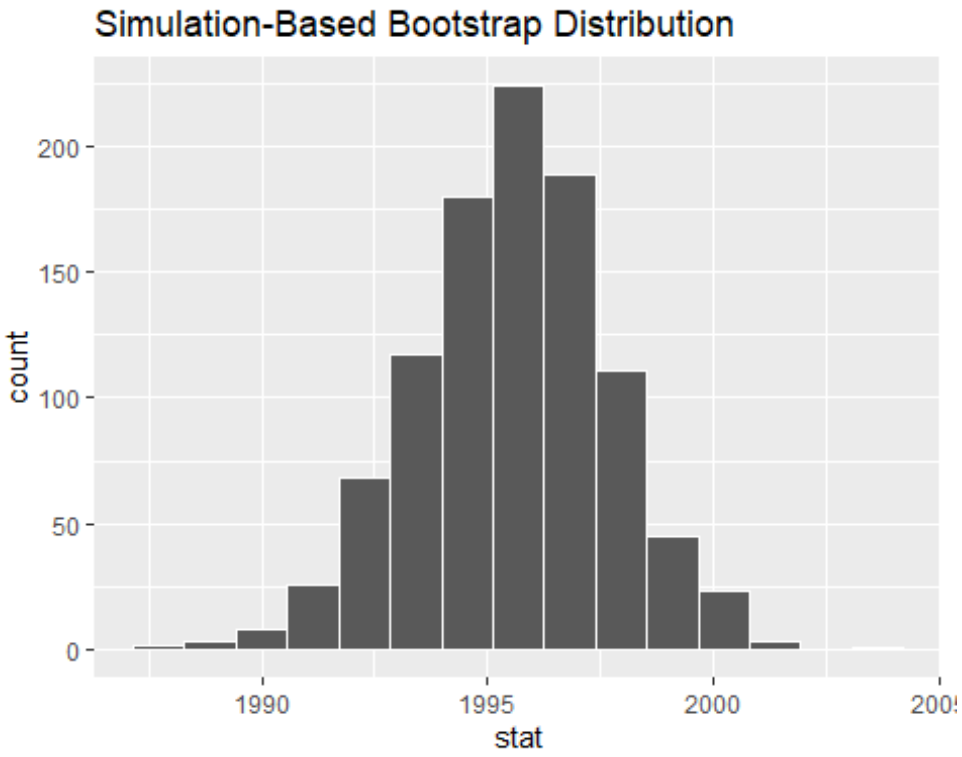
“““ infer workflow: pennies_sample %>% specify(response = year) %>% generate(reps = 1000) %>% calculate(stat = “mean”)

Original workflow: pennies_sample %>% rep_sample_n(size = 50 , replace = T , reps = 1000) %>% group_by(replicate) %>% summarize(stat = mean(year)) “““

4.3.0.1.4 visualize()

visualize()* is same as **geom_histogram()**

```
visualize(bootscap_distribution)
```



4.3.1 Calculating Confidence Interval

Task : Calculate the *C.I.* of bootscrap_distribution .

```
ci <- bootscap_distribution %>%
  get_confidence_interval(level = 0.95 , type = "percentile")

ci
```

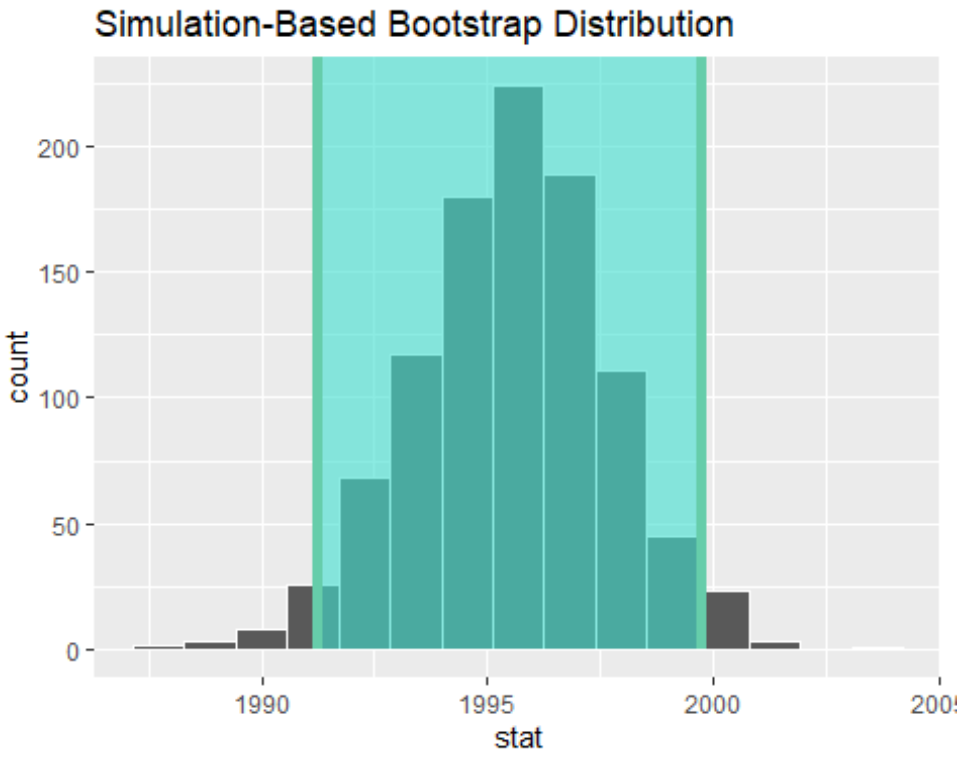
```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    1991.    2000.
```

```
round(ci , 0)
```

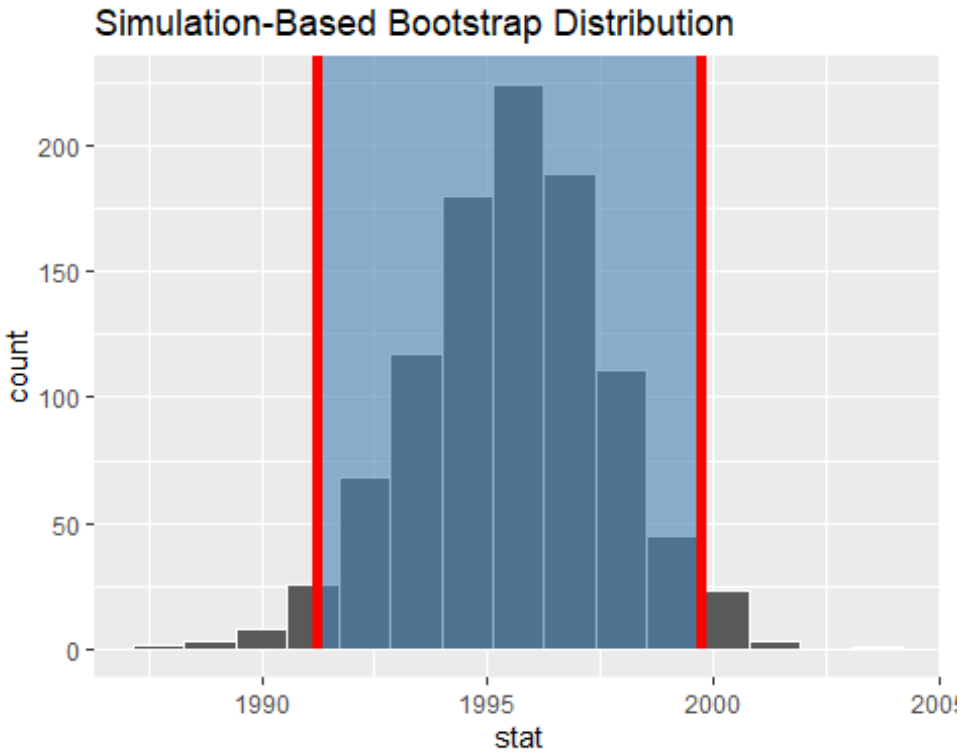
```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     1991     2000
```

Task : Visualize the above result

```
visualize(bootscap_distribution) +
  shade_confidence_interval(endpoints = ci)
```




```
visualize(bootscap_distribution) +
  shade_ci(endpoints = ci , color = "red" , fill = "steelblue")
```



4.3.1.0.1 Standard Error Method with Infer

```
# Overall mean of bootscrap_distribution
x_bar <- bootscap_distribution %>%
  summarise(x_bar = mean(stat))

x_bar
```

```
## # A tibble: 1 x 1
##   x_bar
##   <dbl>
## 1 1996.
```

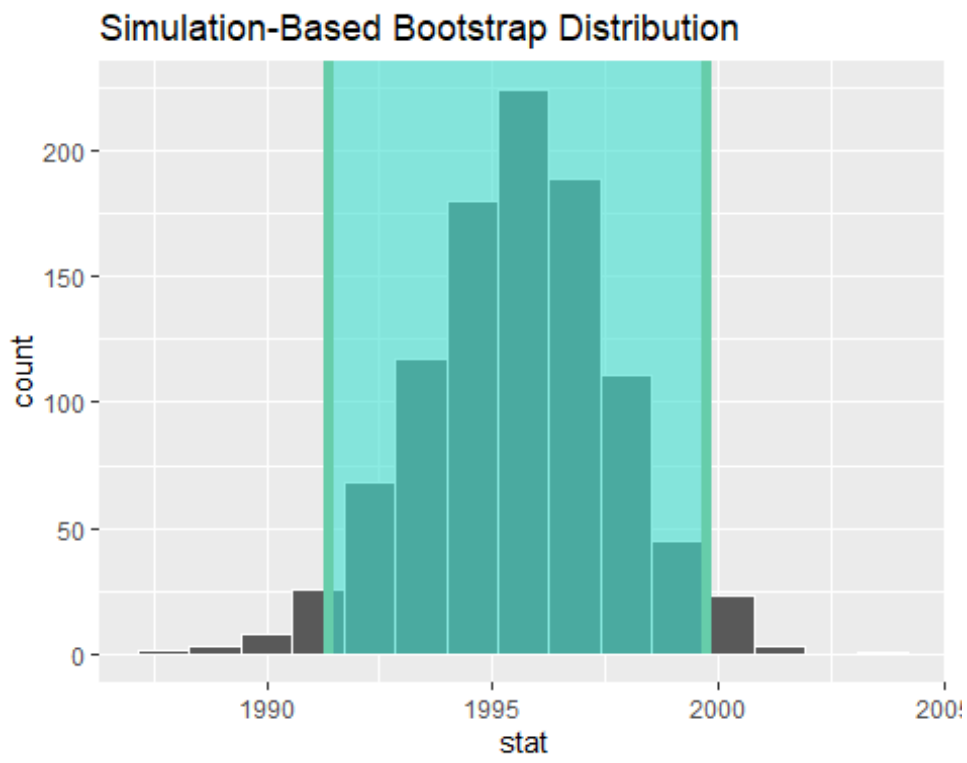
```
# Standard Error
sd_ci <- bootscap_distribution %>%
  get_confidence_interval(type = "se" , point_estimate = x_bar)
```

```
## Using `level = 0.95` to compute confidence interval.
```

```
sd_ci
```

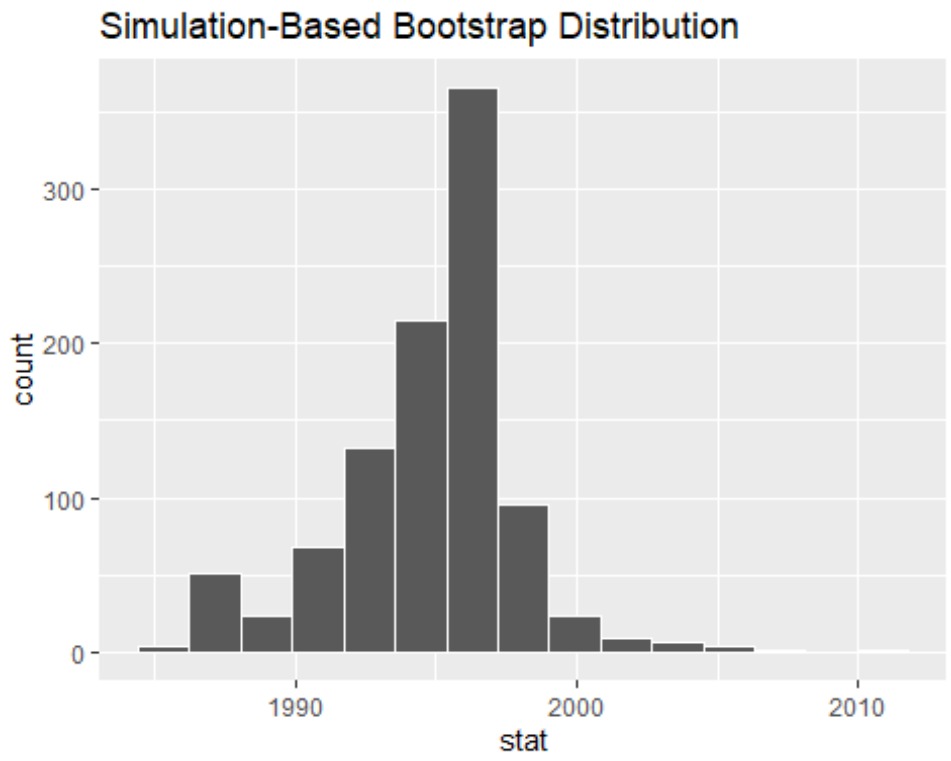
```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1    1991.    2000.
```

```
# Visualize the above result
visualize(bootscap_distribution) +
  shade_confidence_interval(endpoints = sd_ci)
```



(LC 8.5) : Construct a 95% confidence interval for the median year of minting of all US pennies? Use the percentile method and, if appropriate, then use the standard-error method.

```
bootstrap_distribution <- pennies_sample %>%
  specify(response = year) %>%
  generate(reps = 1000) %>%
  calculate(stat = "median")

## Setting `type = "bootstrap"` in `generate()`.

percentile_ci <- bootstrap_distribution %>%
  get_confidence_interval(level = 0.95, type = "percentile")
percentile_ci

## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     1988    2000.

visualize(bootstrap_distribution)
```



4.3.1.1 Interpretating Confidence Intervals

Task : Find the mean of red bowl from *bowl* dat.

```
bowl %>%
  summarize(p_red =mean(color == "red"))
```

```
## # A tibble: 1 x 1
##   p_red
##   <dbl>
## 1 0.375
```

We know that the population proportion **p** is **0.375** . In other words, we know that **37.5%** of the bowl's balls are red .

bowl_sample_1 dataset .

```
bowl_sample_1 %>% glimpse()

## Rows: 50
## Columns: 1
## $ color <chr> "white", "white", "red", "red", "white", "white", "red", "~
```

1. **Task :** We **specify()** the response variable of interest color .

```
#bowl_sample_1 %>%
#  specify(response = color)
```

Error: A level of the response variable *color* needs to be specified for the `success` argument in *specify()*.

We need to define which event is of interest! red or white balls? Since we are interested in the proportion red, let's set success to be "red".

```
bowl_sample_1 %>%
  specify(response = color , success = "red") %>%
  glimpse()

## Rows: 50
## Columns: 1
## $ color <fct> white, white, red, red, white, white, red, white, white, w~
```

2. **Task :** We **generate()** *1000* replicates of bootstrap resampling with replacement from *bowl_sample_1* .

```
bowl_sample_1 %>%
  specify(response = color , success = "red") %>%
  generate(reps = 1000 , type = "bootstrap") %>%
  glimpse()

## Rows: 50,000
## Columns: 2
## Groups: replicate [1,000]
## $ replicate <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ color     <fct> white, white, white, red, red, red, white, red, red, w~
```

3. **Task : calculate()** the proportion of the balls that are *"red"*.

```
sample_1_bootstrap <- bowl_sample_1 %>%
  specify(response = color , success = "red") %>%
  generate(reps = 1000 , type = "bootstrap") %>%
```

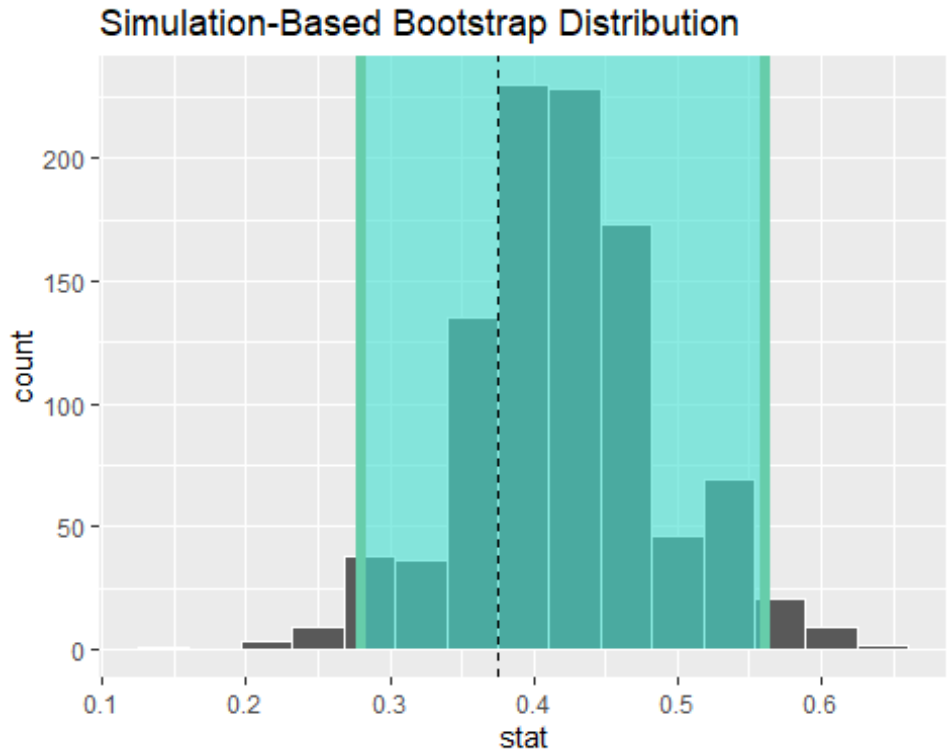
```
  calculate(stat = "prop")

sample_1_bootstrap %>% glimpse()

## Rows: 1,000
## Columns: 2
## $ replicate <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ stat      <dbl> 0.52, 0.40, 0.36, 0.28, 0.40, 0.42, 0.44, 0.38, 0.44, ~
```

4. **Task :** Calculate the C.I.

```
p_ci <- sample_1_bootstrap %>%
  get_confidence_interval(level = 0.95 , type = "percentile")

p_ci

## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     0.28     0.56
```

5. **visualize()** the above result with C.I. .

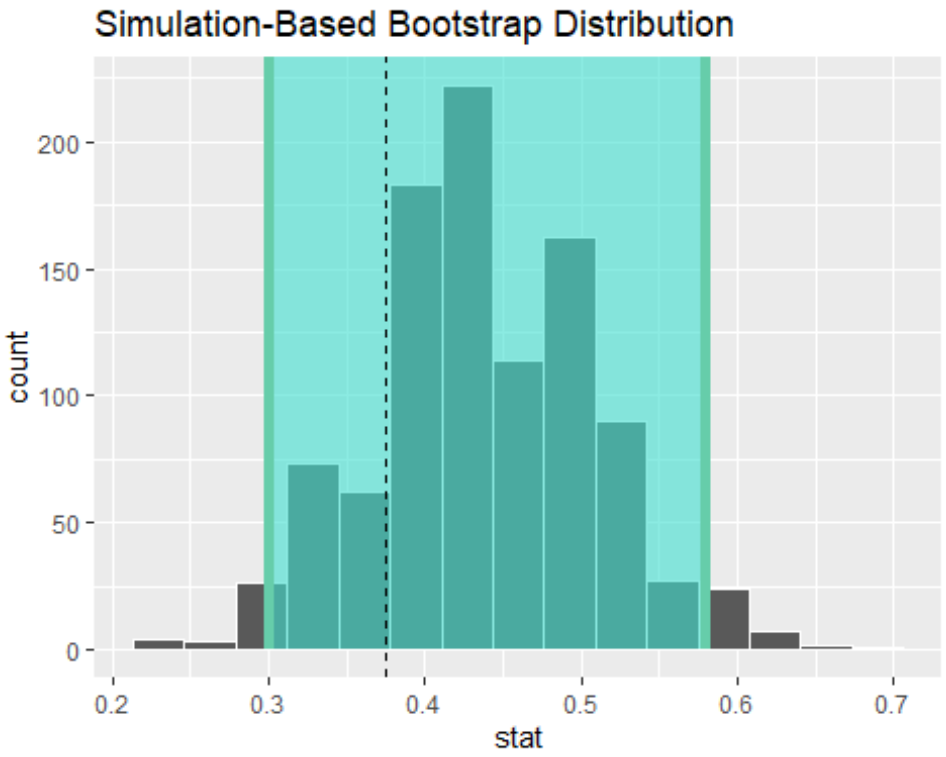
```
sample_1_bootstrap %>%
  visualise(bins = 15) +
  shade_confidence_interval(endpoints = p_ci) +
  geom_vline(xintercept = 0.375 , linetype = "dashed") # 0.375 is prop.
```



4.3.1.1.1 Work on IInd Sample of bowl dataset .

Here we are working as above (bowl_sample_1) .

```
# Take a sample of size 50 from bowl data
bowl_sample_2 <- bowl %>%
  rep_sample_n(size = 50)

bowl_sample_2 %>% glimpse()

## Rows: 50
## Columns: 3
## Groups: replicate [1]
## $ replicate <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ ball_ID   <int> 1359, 523, 2060, 1458, 473, 1128, 403, 703, 329, 2034,~
## $ color     <chr> "red", "red", "red", "white", "white", "white", "white~

# Replicate the sample upto 1000 times
sample_2_bootstrap <- bowl_sample_2 %>%
  specify(response = color,
          success = "red") %>%
  generate(reps = 1000 ,
           type = "bootstrap") %>%
  calculate(stat = "prop")

sample_2_bootstrap %>%
  glimpse()

## Rows: 1,000
## Columns: 2
## $ replicate <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ stat      <dbl> 0.40, 0.42, 0.34, 0.46, 0.50, 0.38, 0.40, 0.36, 0.56, ~

# Confidence Interval
p_ci_2 <- sample_2_bootstrap %>%
  get_confidence_interval(level = 0.95 , type = "percentile")

p_ci_2

## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1      0.3     0.58

# Visualization
sample_2_bootstrap %>%
  visualise(bins = 15) +
  shade_confidence_interval(endpoints = p_ci_2) +
  geom_vline(xintercept = 0.375 , linetype = "dashed") # 0.375 is prop.
```



Short-Hand Interpretation : We are 95% "confident" that a 95% confidence interval captures the value of the population parameter.

Not : * Higher confidence levels tend to produce wider confidence intervals. * Larger sample sizes tend to produce narrower confidence intervals. In other words, our estimates got more and more precise.

4.3.1.1.2 Mythbusters study data

Mythbusters_yawn is data from *moderndive* packages ,

```
data("mythbusters_yawn")

dim(mythbusters_yawn)

## [1] 50  3

mythbusters_yawn %>% glimpse()

## Rows: 50
## Columns: 3
## $ subj  <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,~
## $ group <chr> "seed", "control", "seed", "seed", "seed", "control", "see~
## $ yawn  <chr> "yes", "yes", "no", "yes", "no", "no", "yes", "no", "no", ~
```

Task : Make a frequency table .

```
mythbusters_yawn %>%
  group_by(group , yawn) %>%
  summarize(cout = n())
```

```
## `summarise()` has grouped output by 'group'. You can override using the
`.groups` argument.
```

```
## # A tibble: 4 x 3
## # Groups:   group [2]
##   group   yawn   cout
##   <chr>   <chr> <int>
## 1 control no       12
## 2 control yes       4
## 3 seed    no       24
## 4 seed    yes      10
```

Constructing the C.I. * *specify()* variables

```
mythbusters_yawn %>%
  specify(formula = yawn ~ group , success = "yes") %>%
  glimpse()
```

```
## Rows: 50
## Columns: 2
## $ yawn  <fct> yes, yes, no, yes, no, no, yes, no, no, no, no, no, no, ye~
## $ group <fct> seed, control, seed, seed, seed, control, seed, control, c~
```

- *generate()* replicates

```
hd <- head(mythbusters_yawn) ; hd
```

```
## # A tibble: 6 x 3
##    subj group   yawn
##   <int> <chr>   <chr>
## 1     1 seed    yes
## 2     2 control yes
## 3     3 seed    no
## 4     4 seed    yes
## 5     5 seed    no
## 6     6 control no
```

```
hd %>%
  sample_n(size = 6 , replace = T)
```

```
## # A tibble: 6 x 3
##    subj group   yawn
##   <int> <chr>   <chr>
## 1     6 control no
## 2     5 seed    no
## 3     4 seed    yes
## 4     1 seed    yes
## 5     1 seed    yes
## 6     1 seed    yes
```

```
mythbusters_yawn %>%
  specify(formula = yawn ~ group, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  glimpse()
```

```
## Rows: 50,000
## Columns: 3
## Groups: replicate [1,000]
## $ replicate <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ yawn      <fct> no, no, no, no, no, no, yes, no, yes, yes, no, no, yes~
## $ group     <fct> seed, seed, seed, control, seed, control, control, con~
```

- *calculate()* statistics .

```
mythbusters_yawn %>%
  specify(formula = yawn ~ group, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "diff in props") %>%
  glimpse()
```

```
## Warning: The statistic is based on a difference or ratio; by default, for
## difference-based statistics, the explanatory variable is subtracted in the
## order "control" - "seed", or divided in the order "control" / "seed" for
## ratio-based statistics. To specify this order yourself, supply `order =
## c("control", "seed")` to the calculate() function.
```

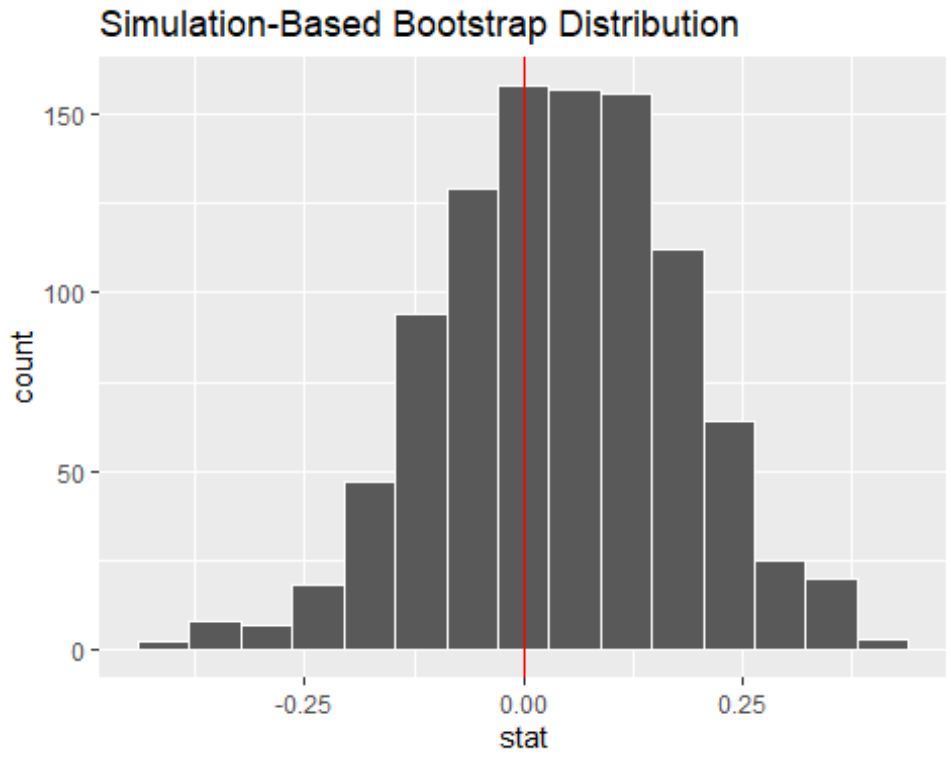
```
## Rows: 1,000
## Columns: 2
## $ replicate <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ stat      <dbl> -0.04411765, -0.12152778, 0.19542620, 0.14685315, 0.09~
```

```
# bdy = bootstrap_distribution_yawning
bdy <- mythbusters_yawn %>%
  specify(formula = yawn ~ group, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "diff in props" , order = c("seed" , "control"))

bdy %>% glimpse()
```

```
## Rows: 1,000
## Columns: 2
## $ replicate <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ stat      <dbl> 0.11226611, 0.12500000, 0.05902778, 0.26666667, 0.1120~
```

- *visualize()* the results

```
visualize(bdy) +
  geom_vline(xintercept = 0 , col = "red")
```



- *C.I.*

```
bdy %>%
  get_confidence_interval(type = "percentile", level = 0.95)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   -0.235    0.319
```

```
obs_diff_in_props <- mythbusters_yawn %>%
  specify(formula = yawn ~ group, success = "yes") %>%
  calculate(stat = "diff in props", order = c("seed", "control"))

obs_diff_in_props
```

```
## Response: yawn (factor)
## Explanatory: group (factor)
## # A tibble: 1 x 1
##     stat
##    <dbl>
## 1 0.0441
```

```
# CI
myth_ci_se <- bdy %>%
  get_confidence_interval(type = "se", point_estimate = obs_diff_in_props)
```

```
## Using `level = 0.95` to compute confidence interval.
```

```
myth_ci_se
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   -0.228    0.316
```

Interpreting he Confidence Intervals : Given that both confidence intervals are quite similar, let's focus our interpretation to only the percentile-method confidence interval of **(-0.238, 0.302)**.

we use the shorthand interpretation: we're 95% "confident" that the true difference in proportions $P_{seed} - P_{control}$ is between $(-0.238\,,\ 0.302)$.

This would suggest that $P_{seed} - P_{control} >$ or, in other words $P_{seed} > P_{control}$, and thus we'd have evidence suggesting those exposed to yawning do yawn more often.

Note :

- The bootstrap distribution will likely not have the same center as the sampling distribution. In other words, bootstrapping cannot improve the quality of an estimate.
- Even if the bootstrap distribution might not have the same center as the sampling distribution, it will likely have very similar shape and spread. In other words, bootstrapping will give you a good estimate of the standard error .

4.3.1.1.3 infer

same	infer
`summarize()`	calculate()
`group_by()`	specify()
`rep_sample_n()`	generates()

5 Hypothesis Testing

5.1 Hypothesis Testing Theory

Hypothesis testing is a statistical method that is used in making statistical decisions using experimental data. Hypothesis Testing is basically an assumption that we make about the population parameter. Hypothesis testing is an essential procedure in statistics. A hypothesis test evaluates two mutually exclusive statements about a population to determine which statement is best supported by the sample data. When we say that a finding is statistically significant, it's thanks to a hypothesis test.

1. **Hypothesis Testing in Large Sample :** Since for large sample almost all the distribution *(Binoial,Poisson ,t,F,Chi-Square)* can be appoximated very closely by

normal distribution. Hence we use normal test (Z-test) of significant for large sample.

2. **Hypothesis Testing in Small Sample :** In small samples, usually the distribution of test statistic is far away from normal distribution and hence in such case exact sampling distribution of the test statistic are used. Some well known hypothesis test for small samples are *t-test,F-test* .

3. **Hypothesis :** A definite statement about a population parameter. There are two types of Hypothesis-i.Null Hypothesis ii.Alternative Hypothesis

4. **Null Hypothesis** (H_0) : A Hypothesis about a parameter that is assumed to be true untill it is declear false. It's often, the Hypothesis of no difference (Null Difference).

5. **Alternative Hypothesis** $(H_1 \,/\, H_A)$: A Hypothesis claim about the parameter complementry to the Null Hypothesis.

6. **Type-I Error / Producer's Risk :** The type of error occur when a True null hypothesis is rejected. The Prob. of commitiy Type-I error= $\alpha = P(Reject - H_0 | H_0's\ TRUE)$

7. **Type-II Error / Cunsumer Risk :** This type of error occurs when the false H_0 is not reject. The Prob. of commitiy Type-II error=$\beta = P(Accept - H_0 | H_0's\ FALSE)$

8. **Critical Region :** It is represented by a region of the area under the probability curve of the distribution of test statistic, this region amount to the rejection of H_0.

9. **Types of Test -** Depending on the location of critical region (or depending on H_1), hypothesis test can be categorised into three types:-

Two-Tailed Test : A test of hypothesis is said to be two tailed test if the critical region lies on the both sides(tails) of the probability curve of test statistic. **OR** A test in which alternative hypothesis is two tailed is called two tailed test. i.e. $H_0: \mu = \mu_0$ VS $H_1: \mu = \mu_0$

Right-Tailed Test : A test of hypothesis is said to be right tailed test if the critical region lies on the right side(tail) of the probability curve of test statistic. **OR** A test in which alternative hypothesis is right tailed is called two tailed test. i.e. $H_0: \mu = \mu_0$ VS $H_1: \mu > \mu_0$

Left-Tailed Test : A test of hypothesis is said to be right tailed test if the critical region lies on the left side(tail) of the probability curve of test statistic. **OR** A test in which alternative hypothesis is left tailed is called two tailed test. i.e. $H_0: \mu = \mu_0$ VS $H_1: \mu < \mu_0$

10. **Critical Value :** The value of the test statitic which seperates the critical and acceptance region under the probability curve oftest statistic are called Critical Region.

11. **Level of Significance** (α) : The probability that the random variable of the test statistic belongs to the critical region is called as Level of Significance. It's the prob. of Type-I error .

12. **Degree of Freedom :** Degrees of freedom refers to the maximum number of logically independent values, which are values that have the freedom to vary, in the data sample.

13. **P-value :** The P value, or calculated probability, is the probability of finding the observed, or more extreme, results when the null hypothesis (H_0) of a study question is true — the definition of 'extreme' depends on how the hypothesis is being tested. If your P value is less than the chosen significance level then you reject the null hypothesis i.e. accept that your sample gives reasonable evidence to support the alternative hypothesis. It does NOT imply a "meaningful" or "important" difference; that is for you to decide when considering the real-world relevance of your result.

$$We\ Reject\ H_0\ if\ ;\quad \boldsymbol{P_{value} < \alpha\ (LOS)}$$

$$C.I. = (1-\alpha)\% = mean \pm CR\ value \times SE$$

14. **Steps Involving in Statistical Hypothesis :**
    i. Formulation Null Hypothesis H_0.
    ii. Formulation Alternative Hypothesis H_1/H_A.
    iii. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
    iv. Fixed the value of α (Level os Significance).
    v. Compute the test statistic, under H_0.
    vi. Taking Decision: **We Reject** H_0 if $|Z_{cal}| > Z_\alpha$ OR **We Reject** H_0 if $p < LOS$

15. **Simple Hypothesis :** If the statistical hypothesis specifies the population completely then it is termed as a Simple Hypothesis. i.e. $H_0: \mu = \mu_0$

16. **Composite Hypothesis :** If the statistical hypothesis which does not specifies the population completely then it is termed as a Composite Hypothesis. i.e. $H_1: \sigma^2 > \sigma_0^2$

5.1.1 Z-Test

Test of Significance for Large Samnples

5.1.1.1 Test for Attributes

5.1.1.1.1 Test of Significance for Single Proportion

If X is the number of successes in n independent tials with constant probability P of success for each trial then, E(x)=nP and V(X)=nPQ, where Q=1-P

$$Z = \frac{p-P}{\sqrt{\frac{PQ}{n}}} \sim N(0,1)$$

where $p = \frac{X}{n}$ called Sample proportion. $SE = \sqrt{\frac{PQ}{n}}$

Procedure -:
A. Set Null Hypothesis $H_0: P = P_0$
B. Set Alternative Hypothesis $H_1: P > P_0$, $H_1: P < P_0$, $H_1: P \neq P_0$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level os Significance).
E. Compute the test statistic, under $H0$.
F. Results - i. *For Right Tailed Test :*
$H_1: P > P_0$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
ii. *For Left Tailed Test :* For $H_1: P < P_0$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
iii. *For Two Tailed Test :* For $H_1: P \neq P_0$, We reject H_0 if $|Z_{cal}| > Z_{\alpha/2}$

5.1.1.1.2 Test of Significance for Difference of Proportions*

Suppose we want to compare two distnict population with respect to occurance of certain attributes, among their members. Let X_1, X_2 be two members of persons possessing the given attribute A in random samples of sizes n_1 and n_2 from the two populations respectively. Sample proportions are given by- $p_1 = \frac{X_1}{n_1}$ & $p_2 = \frac{X_2}{n_2}$ If P_1 and P_2 are populations then, Test Statistic under H_0

$$Z = \frac{p_1 - p_2}{\sqrt{PQ\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim N(0,1)$$

where , $SE = \sqrt{PQ\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$

REMARK -: In general we don't have information to the proportion of attribute in the population from which samples have been drawn under $H_0: P_0 = P_1$, an unbiased estimate of population P based on both samples is given by $\hat{P} = \frac{n_1 p_1 + n_2 p_2}{n_1 + n_2} = \frac{X_1 + X_2}{n_1 + n_2}$

Procedure -:
A. Set Null Hypothesis $H_0: P_0 = P_1$
B. Set Alternative Hypothesis $H_1: P_0 > P_1$ OR $H_1: P_0 < P_1$ OR $H_1: P_0 \neq P_1$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level os Significance).
E. Compute the test statistic, under $H0$.
F. *Results :-*
i. *For Right Tailed Test :* For $H_1: P > P_0$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
ii. *For Left Tailed Test :* For $H_1: P_0 < P_1$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
iii. *For Two Tailed Test :* For $H_1: P_0 \neq P_1$, We reject H_0 if $|Z_{cal}| > Z_{\alpha/2}$

5.1.1.2 Test for Variables

5.1.1.2.1 Test of Significance for Single Mean

Let $x_1, x_2, \cdots, x_n$ be a random sample of size n taken from a large population with mean μ and variance σ^2. Let $\bar{x}$ be the sample mean . Test Statistic under H_0

$$Z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \sim N(0,1)$$

Procedure -
A. Set Null Hypothesis $H_0: \mu = \mu_0$
B. Set Alternative Hypothesis $H_1: \mu > \mu_0$ OR $H_1: \mu > \mu_0$ OR $H_1: \mu \neq \mu_0$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level os Significance).
E. Compute the test statistic, under H_0.
F. *Results -*
i. *For Right Tailed Test :* $H_1: \mu_0 > \mu_1$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
ii. *For Left Tailed Test :* $H_1: \mu_0 < \mu_1$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
iii. *For Two Tailed Test :* $H_1: \mu_0 \neq \mu_0$, We reject H_0 if $|Z_{cal}| > Z_{\alpha/2}$

5.1.1.2.2 Test of Significance for Difference of Two Means

Let $\bar{x_1}$ be the mean of a sample of size n_1 from population with mean μ_1 and variance σ_1^2 and $\bar{x_2}$ be the mean of a sample of sixe n_2 from a population with mean μ_2 and variance σ_2^2. Since sample size are large.

The Test Statistic under H_0 will be:

$$Z = \frac{(\bar{x_1} - \bar{x_2})}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0,1)$$

where , $SE = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$

Procedure -:
A. Set Null Hypothesis $H_0: \mu_1 = \mu_2$
B. Set Alternative Hypothesis $H_1: \mu_1 > \mu_2$ OR $H_1: \mu_1 > \mu_2$ OR $H_1: \mu_1 \neq \mu_2$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level os Significance).
E. Compute the test statistic, under H_0.
F. *Results -*
i. *For Right Tailed Test :* $H_1: \mu_1 > \mu_2$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
ii. *For Left Tailed Test :* $H_1: \mu_1 < \mu_2$, We reject H_0 if $|Z_{cal}| > Z_\alpha$
iii. *For Two Tailed Test :* $H_1: \mu_1 \neq \mu_2$, We reject H_0 if $|Z_{cal}| > Z_{\alpha/2}$

REMARK : i. This will be used when σ_1^2, σ_2^2 are known. ii. When σ_1^2 & σ_2^2 are not known then s_1^2 & s_2^2 i.e. Sample Variances are taken as the estimator of σ_1^2 & σ_2^2 , where $s_1^2 = \frac{1}{n_1}\sum_{j=1}^{n_1}(x_{ij} - \bar{x})^2$ then Test Statistic Under H_0 $Z = \frac{(\bar{x_1} - \bar{x_2})}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \sim N(0,1)$ iii. If $\sigma_1^2 = \sigma_2^2 = \sigma^2$

(unknown) then Test Statistic Under H_0 $Z = \frac{(\bar{x_1} - \bar{x_2})}{\sqrt{\sigma^2\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}} \sim N(0,1)$ Estimator of σ^2 is

$\widehat{\sigma^2} = s^2 = \frac{n_1 s_1^2 + n_2 s_2^2}{n_1 + n_2}$

5.1.2 χ^2 -Test

5.1.2.1 Test of Significance about a Population Variance

Inference about a Population :- Suppopse we want to test if a random sample $x_i (i = 1,2,\cdots,n)$ has been drawn from a normal population with specified. Test Statistic Under H_0 , $\chi^2_{n-1} = \sum_{i=1}^{n} (\frac{x-\bar{x}}{\sigma_0})^2$ OR $\chi^2_{n-1} = \frac{ns^2}{\sigma_0^2}$ follows χ^2 distribution with **(n-1)** degree of freedom.

Procedure -
A. Set Null Hypothesis $H_0: \sigma^2 = \sigma_0^2$
B. Set Alternative Hypothesis $H_1: \sigma^2 < \sigma_0^2$ OR $\sigma^2 > \sigma_0^2$ OR $\sigma^2 \neq \sigma_0^2$
C. Fixed the value of α (Level os Significance).
D. Compute the test statistic, under H_0.
E. *Results -:*
i. *For Right Tailed Test :* We reject H_0 if $|\chi^2_{cal}| > \chi^2_{n-1}(\alpha)$
ii. *For Left Tailed Test :* We reject H_0 if $|\chi^2_{cal}| > \chi^2_{n-1}(1-\alpha)$
iii. *For Two Tailed Test :* We reject H_0 if $|\chi^2_{cal}| > \chi^2_{n-1}(1-\alpha/2)$

Remark : If sample is large **(n>30)** then Fisher's Approximation can be used
$\sqrt{2\chi^2_{cal}} - \sqrt{2n-1} \sim N(0,1)$

5.1.2.2 Test of Significance for Goodness of Fit Test

It's given by "Prof. Karl Pearson"

1. **Observed Frequencies :** The ferquencies obtained from the performance of an experiment are called observed frequencies , denoted by O_i.

2. **Expected Frequencies :** The frequencies that we expext to obtain if the Null Hypothesis is true are called as the expected frequencies , denoted by E_i .

Assumption of Goodness of fit Test :
i. The sample observation should be independent.
ii. Constaints on the cell frequencies if any should be linear for Goodness of Fit.
i.e. $(\sum_{i=1}^{n} O_i = \sum_{i=1}^{n} E_i)$
iii. The total frequency should be larger/greate than 50. i.e.$\sum_{i=1}^{n} O_i > 50$
iv. **Pooling :** No theoritical cell frequency should be less than 5. If there is any , than the consequitive cell frequency are pooled to make them greater than 5. And finally the degree of freedom of χ^2 statistic are adjustedaccordingly. i.e. If we pooled 2 df then we less 2 df from original one. Simply If we have 7 obs.(df) and we pooled 2 df then our new df are 7-2=5. Generally we less 1 df on every calculation.

Goodness of fit Test : It helps us to find if deviation of the experiment from theory is just by chance OR The theory is not adequate to fit the observed data. If $O_i(i = 1,2,\cdots,n)$ is a set of observed (Experimented) frequencies. If $E_i(i = 1,2,\cdots,n)$ is a set of expected (theoritical/hypothetical) frequency. then The Test Statistic Under H_0 $\chi^2_{n-1} = \sum_{i=1}^{n}\{\frac{(O_i-E_i)^2}{E_i}\}$, where $\{\sum_{i=1}^{n} O_i = \sum_{i=1}^{n} E_i$ follows χ^2 with n-1 df.

Procedure -: A. Set Null Hypothesis H_0 : The given Theoritical distribution is a good fit for the experimental results B. Set Alternative Hypothesis H_1 : The given Theoritical distribution is not a good fit for the experimental results This test is always Right Tailed Test C. Fixed the value of α (Level os Significance). F. Compute the test statistic, under H_0. E. *Results -:* i. *For Right Tailed Test :* We reject H_0 if $|\chi^2_{cal}| > \chi^2_{n-1}(\alpha)$

5.1.3 t-Test

It's use when the variance σ^2 is unknown .

5.1.3.1 t-Test for Single Mean

It is use for testing if the population mean differ significantly from a hypothetical value, say μ_0 If $x_1, x_2, \cdots, x_n$ are the random sample of size n drawn from a normal population with mean μ and unknown variance σ^2 . Test Statistic Under H_0 , $t_{n-1} = \frac{\bar{x}-\mu}{s/\sqrt{n}}$, where s^2 is sample variance

Procedure -:
A. Set Null Hypothesis $H_0: \mu = \mu_0$
B. Set Alternative Hypothesis $H_1: \mu > \mu_0$ OR $H_1: \mu > \mu_0$ OR $H_1: \mu \neq \mu_0$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level of Significance).
E. Compute the test statistic, under H_0.
F. *Results -:*
i. *For Right Tailed Test :* $H_1: \mu_0 > \mu_1$, We reject H_0 if $|t_{cal}| > t_{n-1}(\alpha)$
ii. *For Left Tailed Test :* $H_1: \mu_0 < \mu_1$, We reject H_0 if $|t_{cal}| > t_{n-1}(\alpha)$
iii. *For Two Tailed Test :* $H_1: \mu_0 \neq \mu_0$, We reject H_0 if $|t_{cal}| > t_{n-1}(\alpha/2)$

5.1.3.2 t-Test for Differnce of Mean

Suppose we want to test if 2 independent samples $x_i(i = 1,2,\cdots,n_1)$ & $x_j(j = 1,2,\cdots,n_2)$ of size n_1 & n_2 have been drawn from 2 normal population with nean μ_x & μ_y respectively . Test Statistics Under H_0 ,

$$t_{n_1+n_2-2} = \frac{(\bar{x}-\bar{y})}{S\sqrt{\frac{1}{n_1}\frac{1}{n_2}}}$$

where, S is the pooled estimate of σ^2 $S^2 = \frac{n_1 s_1^2 + n_2 s_2^2}{n_1+n_2-2} = \frac{\sum_{i=1}^{n_1}(x_i-\bar{x})^2 \sum_{j=1}^{n_2}(y_i-\bar{y})^2}{n_1+n_2-2}$

Procedure -:
A. Set Null Hypothesis $H_0: \mu_1 = \mu_2$
B. Set Alternative Hypothesis $H_1: \mu_1 > \mu_2$ OR $H_1: \mu_1 > \mu_2$ OR $H_1: \mu_1 \neq \mu_2$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level os Significance).
E. Compute the test statistic, under H_0.
F. *Results -:*
i. *For Right Tailed Test :* $H_1: \mu_1 > \mu_2$, We reject H_0 if $|t_{cal}| > t_{n_1+n_2-2}(\alpha)$
ii. *For Left Tailed Test :* $H_1: \mu_1 < \mu_2$, We reject H_0 if $|t_{cal}| > t_{n_1+n_2-2}(\alpha)$
iii. *For Two Tailed Test :* $H_1: \mu_1 \neq \mu_2$, We reject H_0 if $|t_{cal}| > t_{n_1+n_2-2}(\alpha/2)$

5.1.3.3 Paired t-Test for Differnce of Mean

Let $x_1, x_2, \cdots, x_n$ & $y_1, y_2, \cdots, y_n$ are two samples of size n,which are not independent such that $(x_i, y_i), i = 1,2, \ldots, n$ is a pair of observations obtain from the i^{th} unit of sample. Test Statistic Under H_0

$$t_{n_1+n_2-2} = \frac{\bar{d}}{S/\sqrt{n}}$$

where,$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$, & $d_i = x_i - y_i$ & $S^2 = \frac{1}{n-1}\sum_{i-1}^{n}\left(d_i - \bar{d}\right)^2$

Procedure -:
A. Set Null Hypothesis $H_0: \mu_1 = \mu_2$
B. Set Alternative Hypothesis $H_1: \mu_1 > \mu_2$ OR $H_1: \mu_1 > \mu_2$ OR $H_1: \mu_1 \neq \mu_2$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level os Significance).
E. Compute the test statistic, under H_0.
F. *Results -:*
i. *For Right Tailed Test :* $H_1: \mu_1 > \mu_2$, We reject H_0 if $|t_{cal}| > t_{n_1+n_2-2}(\alpha)$
ii. *For Left Tailed Test :* $H_1: \mu_1 < \mu_2$, We reject H_0 if $|t_{cal}| > t_{n_1+n_2-2}(\alpha)$
iii. *For Two Tailed Test :* $H_1: \mu_1 \neq \mu_2$, We reject H_0 if $|t_{cal}| > t_{n_1+n_2-2}(\alpha/2)$

5.1.4 F-Test

5.1.4.1 F-Test for Equality of Two Population Variances

This test is used to test,if the two independent samples have been drawn from normal population with same variance σ^2 If $x_i (i = 1,2, \cdots, n_1)$ and $y_j (j = 1,2, \cdots, n_2)$ are the two independent sizes n_1 & n_2 respectively drawn from a normal population with means μ_1 , μ_2 and variances σ_1^2 , σ_2^2 respectively, then Test Statistic Under H_0

$$F_{(n_1-1, n_2-1)} = \frac{S_x^2}{S_y^2}$$

where, $S_x^2 = \frac{n_1 s_x^2}{n_1 - 1}$ and $S_y^2 = \frac{n_1 s_y^2}{n_2 - 1}$

Procedure -:
A. Set Null Hypothesis $H_0: \sigma_0^2 = \sigma_1^2 = \sigma_2^2$
B. Set Alternative Hypothesis $H_1: \sigma_1^2 < \sigma_2^2$ OR $H_1: \sigma_1^2 > \sigma_2^2$ OR $H_1: \sigma_1^2 \neq \sigma_2^2$
C. Identify the type of Test.(Two Tailed / Right Tailed / Left Tailed).
D. Fixed the value of α (Level os Significance).
E. Compute the test statistic, under H_0.
F. *Results -:*
i. *For Right Tailed Test :* $H_0: \sigma_1^2 > \sigma_2^2$, We reject H_0 if $|F_{cal}^2| > F_{(n_1,n_2)}(\alpha)$
ii. *For Left Tailed Test :* $H_0: \sigma_1^2 < \sigma_2^2$, We reject H_0 if $|F_{cal}| > F_{(n_1,n_2)}(1-\alpha)$
iii. *For Two Tailed Test :* $H_0: \sigma_1^2 \neq \sigma_2^2$, We reject H_0 if $|F_{cal}| > F_{(n_1,n_2)}(1-\alpha/2)$

5.2 Promotions Activity

About Promotions Data Data from a 1970's study on whether gender influences hiring recommendations. Originally used in OpenIntro.org. There are 3 variables which contains 48 obs. i. *id* , ii. *gende* : Male / Female , iii. *Decision :* Promoted / Not-Promoted .

```
library(tidyverse)
library(infer)
library(moderndive)
library(nycflights13)
library(ggplot2movies)
```

Task : Take a sample of size 5 from *pomotions* dataset

```
data("promotions")

# Sampling
promotions %>%
  sample_n(size = 6) %>%
  arrange(id)
```

```
## # A tibble: 6 x 3
##      id decision gender
##   <int> <fct>    <fct>
## 1     2 promoted male
## 2     7 promoted male
## 3    17 promoted male
## 4    18 promoted male
## 5    41 not      female
## 6    43 not      female
```

Task : Make a Stacked Barplot of promotions and fill it by decision .

```
ggplot(promotions , aes(x = gender , fill = decision)) +
  geom_bar() +
  labs(x = "Gender of Name on Resume")
```

Task : Make a frequency table by **tally()** not by **summarize(n = n())** .

```
promotions %>%
  group_by(gender , decision) %>%
  tally()
```

```
## # A tibble: 4 x 3
## # Groups:   gender [2]
##   gender decision     n
##   <fct>  <fct>    <int>
## 1 male   not          3
## 2 male   promoted    21
## 3 female not         10
## 4 female promoted    14
```

5.2.1 Conducting Hypothesis Tests

infer package workflow

1. **specify()** variables : we use the specify() verb to specify the response variable and if needed, any explanatory variables for our study.

Task : Find the proportion of résumés "promoted", and not the proportion of résumés not promoted .

```
promotions %>%
  specify(formula = decision ~ gender , success = "promoted") %>%
  glimpse()
```

```
## Rows: 48
## Columns: 2
## $ decision <fct> promoted, promoted, promoted, promoted, promoted, promo~
## $ gender   <fct> male, male, male, male, male, male, male, male, male, m~
```

2. ***Hypothesize()***
   Null Hypothesis : There was *no difference in gender-based discrimination rates* .
   i .e. $H_0: p_m - p_f = 0$
   Alternative Hypothesis : Males are more promoted than Females .
   i.e. $H_1: p_m - p_f > 0 \quad p_m > p_f$

```
promotions %>%
  specify(formula = decision ~ gender , success = "promoted") %>%
  hypothesize(null = "independence") %>%
  head(5)
```

```
## Response: decision (factor)
## Explanatory: gender (factor)
## Null Hypothesis: independence
## # A tibble: 5 x 2
##   decision gender
##   <fct>    <fct>
## 1 promoted male
## 2 promoted male
## 3 promoted male
## 4 promoted male
## 5 promoted male
```

3. **generate()** replicate .
   Repeat the process upto 1000 times.

```
promotions_generate <- promotions %>%
  specify(formula = decision ~ gender , success = "promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000 , type = "permute")

glimpse(promotions_generate)  # 48*1000 = 48000
```

```
## Rows: 48,000
## Columns: 3
## Groups: replicate [1,000]
## $ decision  <fct> promoted, promoted, promoted, promoted, promoted, prom~
## $ gender    <fct> male, male, male, male, male, male, male, male, male, ~
## $ replicate <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

Unlike for confidence intervals where we generated replicates using *type = "bootstrap"* resampling with replacement, we'll now perform shuffles/permutations by setting *type = "permute"* . Recall that shuffles/permutations are a kind of resampling, but unlike the bootstrap method, they involve resampling without replacement.

4. **calculate()** summary statistics .

   Task : Find the proportion difference of Male and female.

```
null_distribution <- promotions %>%
  specify(formula = decision ~ gender , success = "promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000 , type = "permute") %>%
  calculate(stat= "diff in props", order = c("male" , "female"))

null_distribution %>% head(5) # 1000 obs
```

```
## Response: decision (factor)
## Explanatory: gender (factor)
## Null Hypothesis: independence
## # A tibble: 5 x 2
##   replicate    stat
##       <int>   <dbl>
## 1         1  0.208
## 2         2  0.125
## 3         3 -0.208
## 4         4 -0.125
## 5         5 -0.0417
```

Observe that we have 1000 values of stat, each representing one instance of $\widehat{p_m} - \widehat{p_f}$ in a hypothesized world of no gender discrimination .

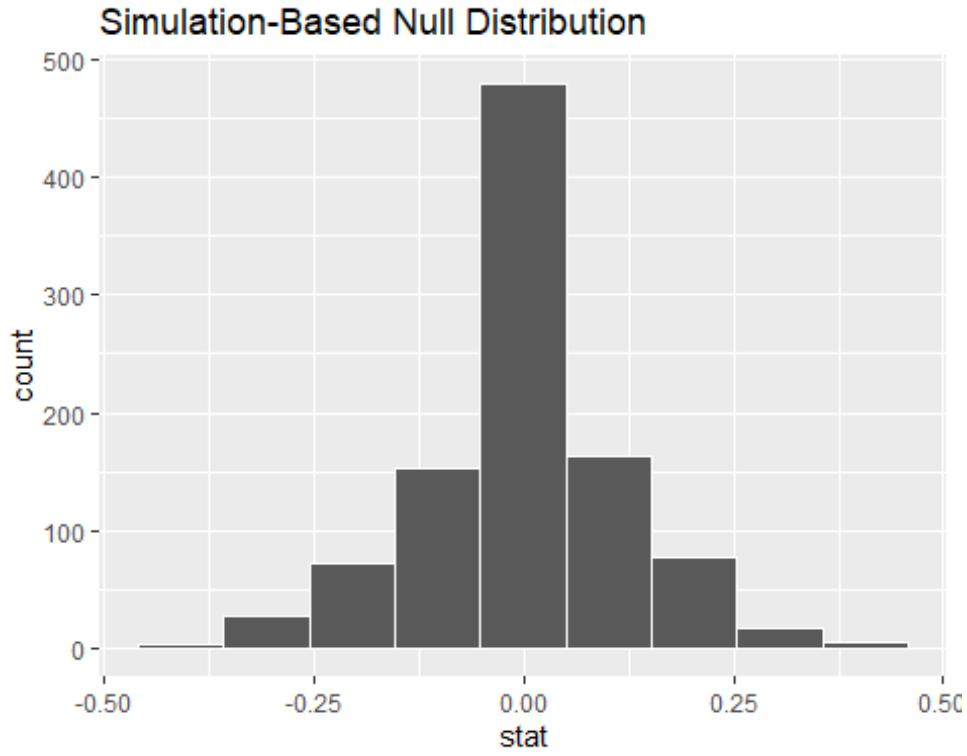
In above code if , we remove *generate()* command then ,

```
obs_diff_prop <- promotions %>%
  specify(formula = decision ~ gender , success = "promoted") %>%
  hypothesize(null = "independence") %>%
  calculate(stat= "diff in props", order = c("male" , "female"))

## Message: The independence null hypothesis does not inform calculation of
the observed statistic (a difference in proportions) and will be ignored.

obs_diff_prop

## Response: decision (factor)
## Explanatory: gender (factor)
## Null Hypothesis: independence
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 0.292

obs_diff_prop %>% round(2)

## Response: decision (factor)
## Explanatory: gender (factor)
## Null Hypothesis: independence
## # A tibble: 1 x 1
```

```
##    stat
##   <dbl>
## 1  0.29
```

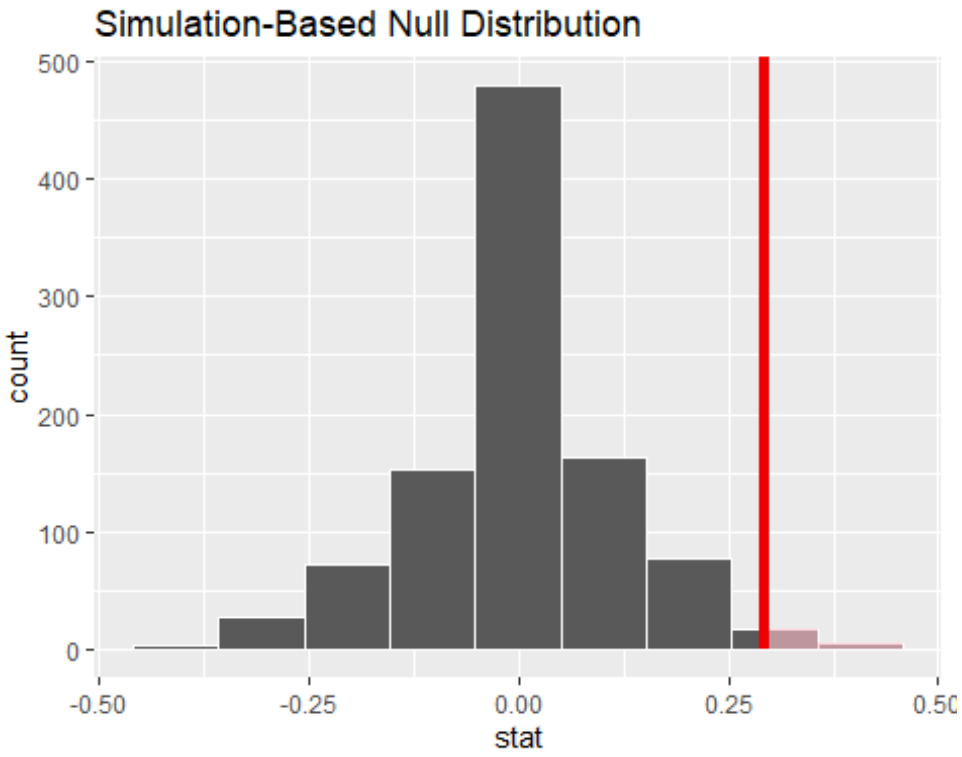
5. **visualize()** the p-valve

```
visualize(null_distribution , bins = 10)
```



We'll set the direction = "right" reflecting our alternative hypothesis $H_1: p_m - p_f > 0$ OR $H_1: p_m > p_f$.

```
visualize(null_distribution, bins = 10) +
  shade_p_value(obs_stat = obs_diff_prop, direction = "right")
```



Note : A p-value is the probability of obtaining a test statistic just as or more extreme than the observed test statistic assuming the null hypothesis H_0 is true.

Task: Find the exact p-value.

```
null_distribution %>%
  get_p_value(obs_stat = obs_diff_prop , direction = "right")
```

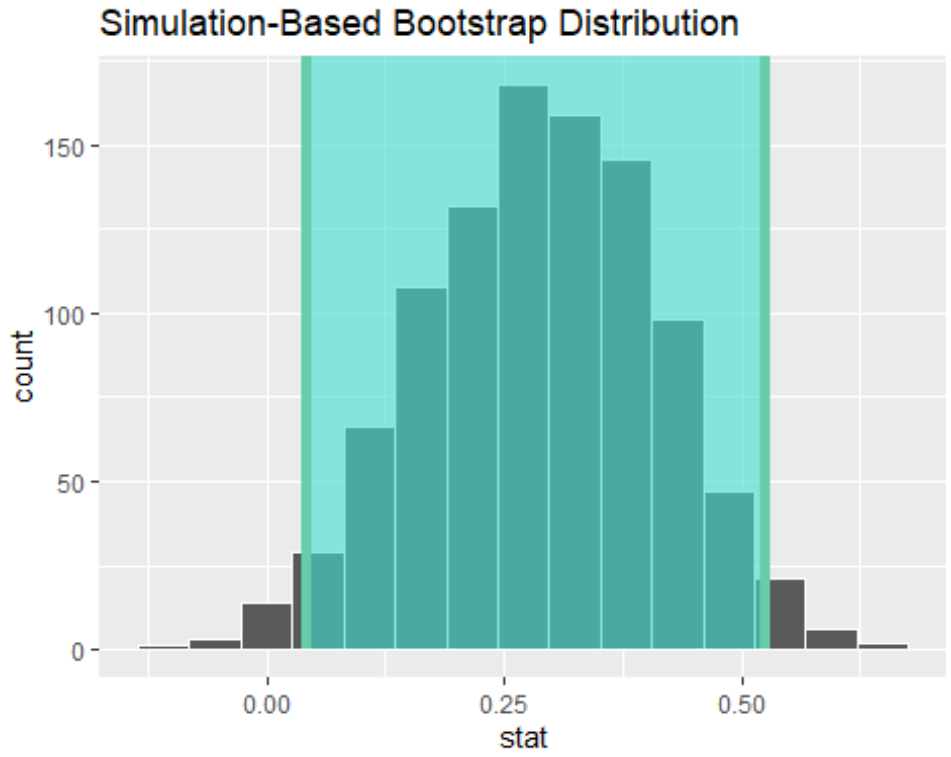
```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.023
```

Comparision with C.I.

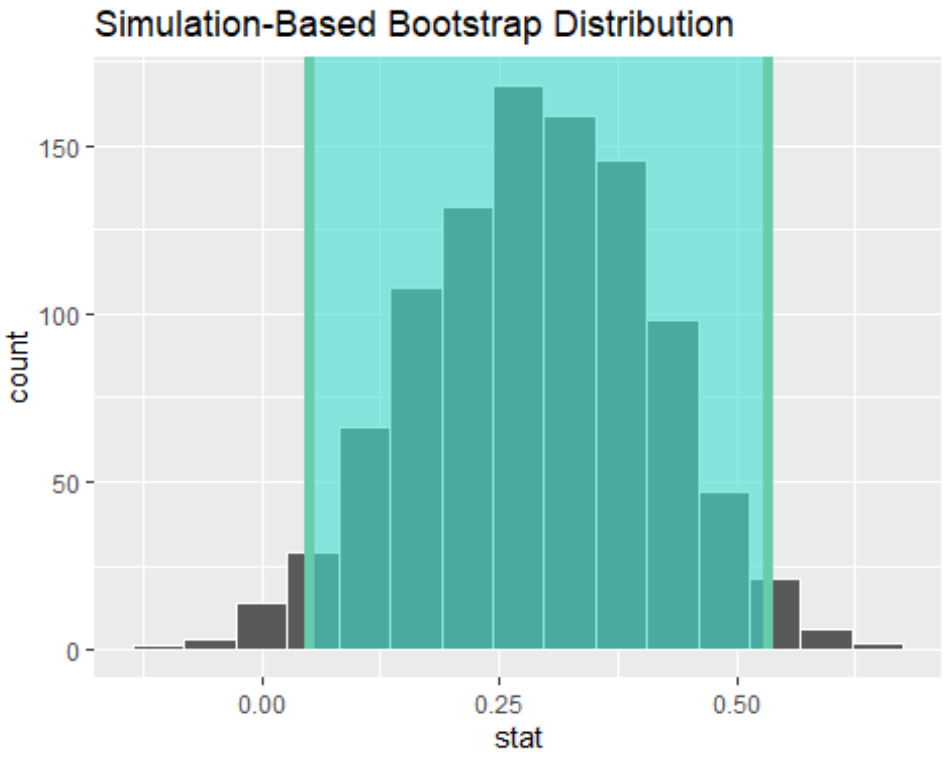
```
bootstrap_distribution <- promotions %>%
  specify(formula = decision ~ gender, success = "promoted") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "diff in props", order = c("male", "female"))

# CI
percentile_ci <- bootstrap_distribution %>%
  get_confidence_interval(level = 0.95, type = "percentile")

percentile_ci
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
```

```
##      <dbl>    <dbl>
## 1   0.0417    0.524

# Graph
visualize(bootstrap_distribution) +
  shade_confidence_interval(endpoints = percentile_ci)
```




```
se_ci <- bootstrap_distribution %>%
get_confidence_interval(level = 0.95, type = "se",
point_estimate = obs_diff_prop)

se_ci

## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   0.0500    0.533

visualize(bootstrap_distribution) +
shade_confidence_interval(endpoints = se_ci)
```



(LC 9.2) : Why are we relatively confident that the distributions of the sample proportions will be good approximations of the population distributions of promotion proportions for the two genders?
Ans : Because the sample is representative of the population.

(LC 9.3) : Using the definition of p-value, write in words what the *p-value* represents for the hypothesis test comparing the promotion rates for males and females.
Ans : The *p-value* represents for the likelihood that the true mean for the promotion rates for males and females in the population is the same.

5.2.1.1 Only One Test

Workflow : `specify()` the variables of interest in your data frame. `hypothesize()` the Null Hypothesis H_0 . `generate()` shuffles assuming H_0 is true . `calculate()` the *test statistic* of interest , both for the observed data and your *simulated* data . `visualize()` the resulting *null distribution* and computer the *p-value* by computing the null distibution to the observed test statistic .

(LC 9.4) : Describe in a paragraph how we used Allen Downey's diagram to conclude if a statistical difference existed between the promotion rate of males and females using this study.
Ans : We use the promotions dataset as the input for test statistic. The H_0 model is "there is no difference between promotion rates of males and females", and with the p-value from infer commands, we reject the H_0 model and conclude that there is a statistical difference existed between the promotion rate of males and females.

5.2.1.2 Interpreting Hypothesis Tests

- If the **p-value** is less than α , then we **Reject** the *null hypothesis* H_0 in favor of H_1 i.e. $p_{value} < \alpha$.
- If the **p-value** is greater than or equal to α , we **Fail** to **Reject** the *null hypothesis* H_0 in favour of H_1 i.e. $p_{value} > \alpha$

(LC 9.5) : What is wrong about saying, "The defendant is innocent." based on the US system of criminal trials ?
Ans : Failing to prove the defendant is guilty does not equal to proving that the defendant is innocent. There will always be the possibility of making errors in the trial.

(LC 9.6) : What is the purpose of hypothesis testing ?
Ans : The purpose of hypothesis testing is to determine whether there is enough statistical evidence in favor of a certain belief, or hypothesis, about a parameter. (source: https://personal.utdallas.edu/~scniu/OPRE-6301/documents/Hypothesis_Testing.pdf)

(LC 9.7) : What are some flaws with hypothesis testing? How could we alleviate them ?
Ans : The p-value's 0.05 threshold can be misleading researchers to conduct multiple bootstrap tests to get a smaller p-value, therefore validating their statistical results. This threshold is relatively arbitrary (if a p-value is 0.051, does it mean there is no statistical significance?), and trusting it too much may lead to imprecise conclusions. To alleviate this problem, keep in mind that having a smaller p-value can be the result of a "lucky" sampling that is not truly representative, and do multiple bootstrap samplings for hypothesis testing before concluding.

(LC 9.8) : Consider two α significance levels of **0.1** and **0.01** Of the two, which would lead to a more liberal hypothesis testing procedure ? In other words, one that will, all things being equal, lead to more rejections of the null hypothesis H_0 .
Ans : The greater α of **0.1** will lead to a more liberal hypothesis testing procedure, because the required p-value to reject the null hypothesis H_0 can be greater. There is a higher chance that the p-value will be less than α .

5.2.2 Case study: Are action or romance movies rated higher?

Here we are discussing **movies** data from *gg2movies* package .

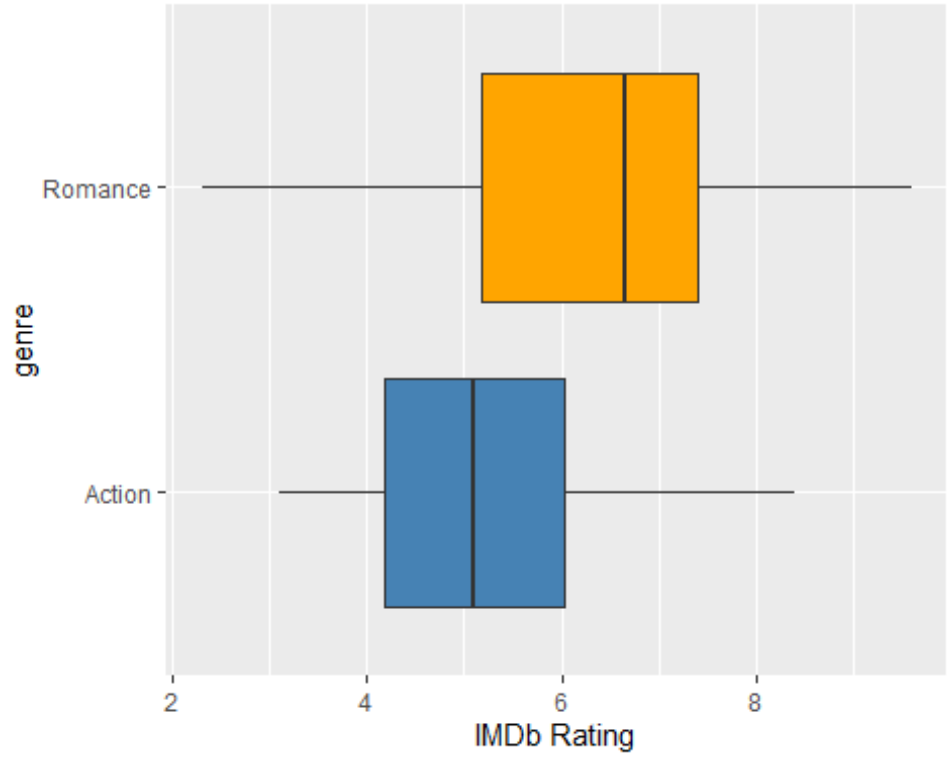
```
data("movies")
movies %>% glimpse()
```

```
## Rows: 58,788
## Columns: 24
## $ title       <chr> "$", "$1000 a Touchdown", "$21 a Day Once a Month", ~
## $ year        <int> 1971, 1939, 1941, 1996, 1975, 2000, 2002, 2002, 1987~
## $ length      <int> 121, 71, 7, 70, 71, 91, 93, 25, 97, 61, 99, 96, 10, ~
## $ budget      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ rating      <dbl> 6.4, 6.0, 8.2, 8.2, 3.4, 4.3, 5.3, 6.7, 6.6, 6.0, 5.~
## $ votes       <int> 348, 20, 5, 6, 17, 45, 200, 24, 18, 51, 23, 53, 44, ~
## $ r1          <dbl> 4.5, 0.0, 0.0, 14.5, 24.5, 4.5, 4.5, 4.5, 4.5, 4.5, ~
```

```
## $ r2          <dbl> 4.5, 14.5, 0.0, 0.0, 4.5, 4.5, 0.0, 4.5, 4.5, 0.0, 0~
## $ r3          <dbl> 4.5, 4.5, 0.0, 0.0, 0.0, 4.5, 4.5, 4.5, 4.5, 4.5, 4.~
## $ r4          <dbl> 4.5, 24.5, 0.0, 0.0, 14.5, 14.5, 4.5, 4.5, 0.0, 4.5,~
## $ r5          <dbl> 14.5, 14.5, 0.0, 0.0, 14.5, 14.5, 24.5, 4.5, 0.0, 4.~
## $ r6          <dbl> 24.5, 14.5, 24.5, 0.0, 4.5, 14.5, 24.5, 14.5, 0.0, 4~
## $ r7          <dbl> 24.5, 14.5, 0.0, 0.0, 0.0, 4.5, 14.5, 14.5, 34.5, 14~
## $ r8          <dbl> 14.5, 4.5, 44.5, 0.0, 0.0, 4.5, 4.5, 14.5, 14.5, 4.5~
## $ r9          <dbl> 4.5, 4.5, 24.5, 34.5, 0.0, 14.5, 4.5, 4.5, 4.5, 4.5,~
## $ r10         <dbl> 4.5, 14.5, 24.5, 45.5, 24.5, 14.5, 14.5, 14.5, 24.5,~
## $ mpaa        <chr> "", "", "", "", "", "", "R", "", "", "", "", "", "",~
## $ Action      <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0~
## $ Animation   <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ Comedy      <int> 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0~
## $ Drama       <int> 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0~
## $ Documentary <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1~
## $ Romance     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ Short       <int> 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1~

movies_sample %>% glimpse()

## Rows: 68
## Columns: 4
## $ title  <chr> "Underworld", "Love Affair", "Junglee", "Eversmile, New J~
## $ year   <int> 1985, 1932, 1961, 1989, 1979, 1988, 1991, 1995, 1976, 197~
## $ rating <dbl> 3.1, 6.3, 6.8, 5.0, 4.0, 4.9, 7.4, 3.5, 7.7, 5.8, 8.9, 3.~
## $ genre  <chr> "Action", "Romance", "Romance", "Romance", "Action", "Rom~

movies_sample %>% head(5)

## # A tibble: 5 x 4
##   title                 year rating genre
##   <chr>                <int>  <dbl> <chr>
## 1 Underworld            1985    3.1 Action
## 2 Love Affair           1932    6.3 Romance
## 3 Junglee               1961    6.8 Romance
## 4 Eversmile, New Jersey  1989    5   Romance
## 5 Search and Destroy     1979    4   Action
```

Task : Make a Boxplot b/w *genre* and *rating* .

```
ggplot(movies_sample , aes(x = genre , y = rating)) +
  geom_boxplot(fill =  c("steelblue" , "orange")) +
  labs(y = "IMDb Rating") + coord_flip()
```



Task : Find the *n* , *mean* , *sd* of *movies_sample* data.

```
movies_sample %>%
  group_by(genre) %>%
  summarise(n = n(),
            mean_rating = mean(rating),
            sd_rating = sd(rating))
```

```
## # A tibble: 2 x 4
##   genre       n mean_rating sd_rating
##   <chr>   <int>       <dbl>     <dbl>
## 1 Action     32        5.28      1.36
## 2 Romance    36        6.32      1.61
```

Hypothesis Testing :

$$H_0: \mu_{action} = \mu_{romance} \quad vs \quad H_A: \mu_{action} \neq \mu_{romance}$$

i. **specify()** variable :

```
movies_sample %>%
  specify(formula = rating ~ genre) %>%
  head()  # dim() = 68 x 2
```

```
## Response: rating (numeric)
## Explanatory: genre (factor)
## # A tibble: 6 x 2
##   rating genre
##    <dbl> <fct>
```

```
## 1    3.1 Action
## 2    6.3 Romance
## 3    6.8 Romance
## 4    5   Romance
## 5    4   Action
## 6    4.9 Romance
```

2. **hypothesize()** the Null

```
movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  head() # dim = 68 x 2

## Response: rating (numeric)
## Explanatory: genre (factor)
## Null Hypothesis: independence
## # A tibble: 6 x 2
##   rating genre
##    <dbl> <fct>
## 1    3.1 Action
## 2    6.3 Romance
## 3    6.8 Romance
## 4    5   Romance
## 5    4   Action
## 6    4.9 Romance
```

3. **generate()** replicate

```
movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(rep = 1000 , type = "permute") %>%
  glimpse()

## Rows: 68,000
## Columns: 3
## Groups: replicate [1,000]
## $ rating    <dbl> 2.3, 5.0, 6.8, 7.4, 6.7, 7.1, 5.0, 7.1, 7.4, 4.1, 4.8,~
## $ genre     <fct> Action, Romance, Romance, Romance, Action, Romance, Ro~
## $ replicate <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

4. **calculate()** summary statistics

```
null_distribution_movies <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(rep = 1000 , type = "permute") %>%
  calculate(stat = "diff in means" , order = c("Action" , "Romance"))

null_distribution_movies %>%
  head()  # dim = 1000 x 2
```

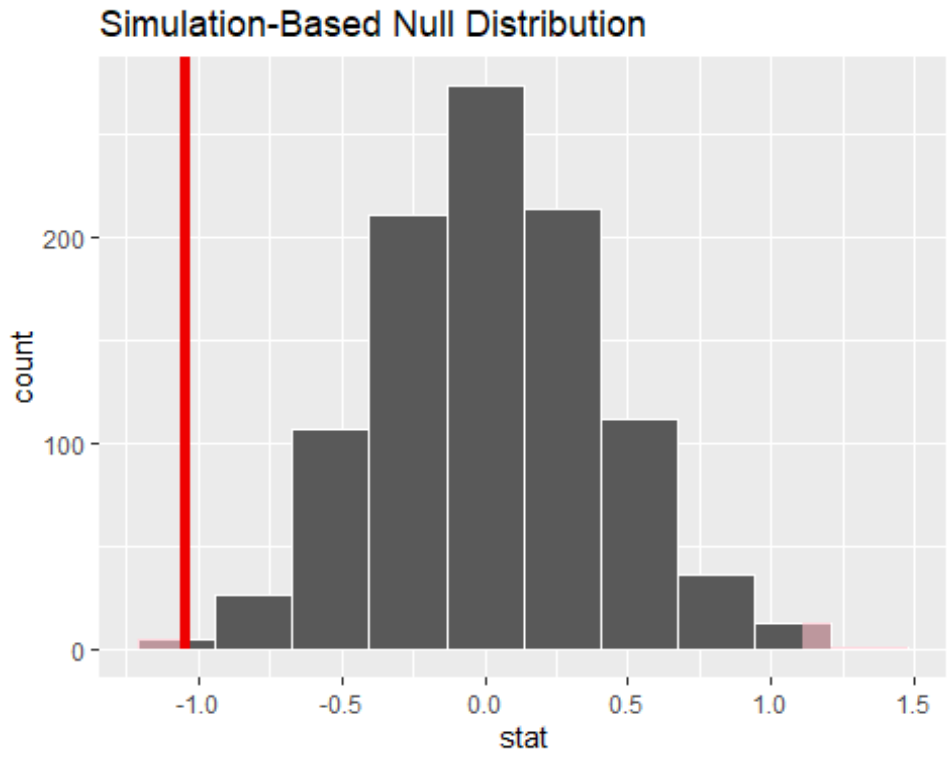
```
## Response: rating (numeric)
## Explanatory: genre (factor)
## Null Hypothesis: independence
## # A tibble: 6 x 2
##   replicate    stat
##       <int>   <dbl>
## 1         1  0.133
## 2         2  0.234
## 3         3  0.877
## 4         4  0.493
## 5         5  0.0743
## 6         6 -0.410
```

```
obs_diff_means <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
 calculate(stat = "diff in means" , order = c("Action" , "Romance"))

obs_diff_means
```

```
## Response: rating (numeric)
## Explanatory: genre (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 -1.05
```

5. **visualize()** the p-value

```
visualize(null_distribution_movies , bins = 10) +
  shade_p_value(obs_stat = obs_diff_means , direction = "both")
```




```
null_distribution_movies %>%
  get_p_value(obs_stat = obs_diff_means, direction = "both")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.004
```

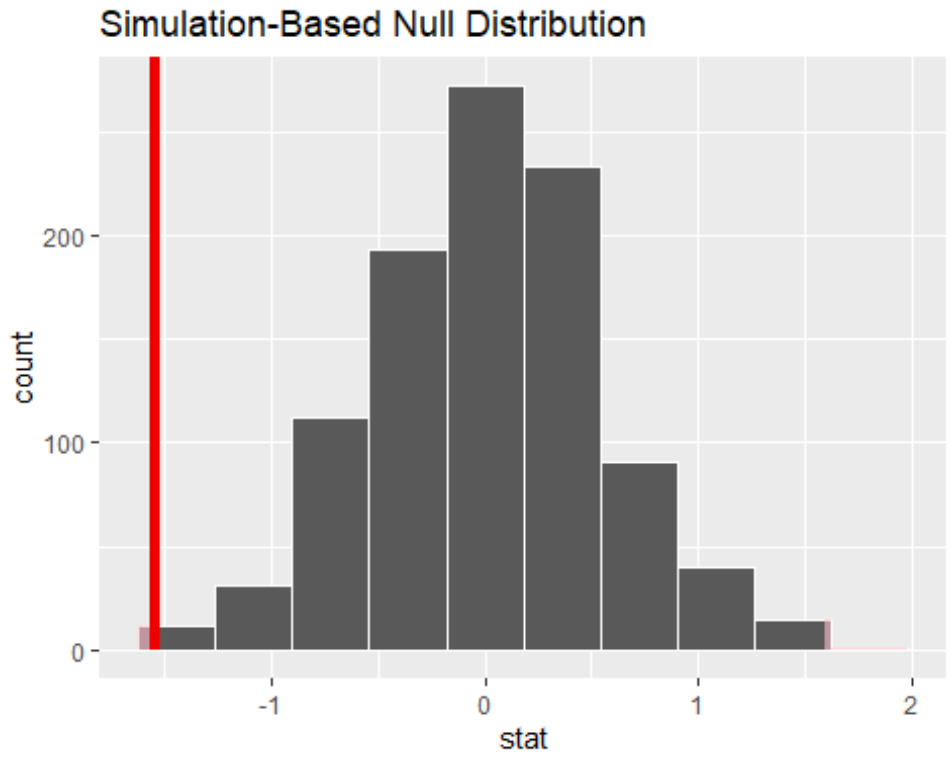
(LC 9.9) : Conduct the same analysis comparing action movies versus romantic movies using the median rating instead of the mean rating. What was different and what was the same ?

```
# In calculate() step replace "diff in means" with "diff in medians"
null_distribution_movies_median <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in medians", order = c("Action", "Romance"))

# compute observed "diff in medians"
obs_diff_medians <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  calculate(stat = "diff in medians", order = c("Action", "Romance"))
obs_diff_medians
```

```
## Response: rating (numeric)
## Explanatory: genre (factor)
```

```
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 -1.55
```

```
# Visualize p-value. Observing this difference in medians under H0
# is very unlikely! Suggesting H0 is false, similarly to when we used
# "diff in means" as the test statistic.

visualize(null_distribution_movies_median, bins = 10) +
  shade_p_value(obs_stat = obs_diff_medians, direction = "both")
```




```
# p-value is very small, just like when we used "diff in means"
# as the test statistic.
null_distribution_movies_median %>%
  get_p_value(obs_stat = obs_diff_medians, direction = "both")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.002
```

(LC 9.10) : What conclusions can you make from viewing the faceted histogram looking at rating versus genre that you couldn't see when looking at the boxplot ?
Ans : From the faceted histogram, we can also see the comparison of ratingversusgenre` over each year, but we cannot conclude them from the boxplot.

(LC 9.11) : Describe in a paragraph how we used Allen Downey's diagram to conclude if a statistical difference existed between mean movie ratings for action and romance movies.
Ans : We use the *movies_sample* dataset as the input for test statistic. The H_0 model is "there is no statistical difference existed between mean movie ratings for action and romance movies", and with the p-value from infer commands, we fail to reject the H_0 model and conclude that there is insufficient evidence to conclude that a statistical difference existed between mean movie ratings for action and romance movies.

(LC 9.12) : Why are we relatively confident that the distributions of the sample ratings will be good approximations of the population distributions of ratings for the two genres ?
Ans : Because the sample is representative of the population.

(LC 9.13) : Using the definition of p-value, write in words what the p-value represents for the hypothesis test comparing the mean rating of romance to action movies.
Ans : The p-value represent the probability that the difference between mean movie ratings for action and romance movies in the sample is natural, i.e., the probability that there is no statistical difference between mean movie ratings for action and romance movies in the population.

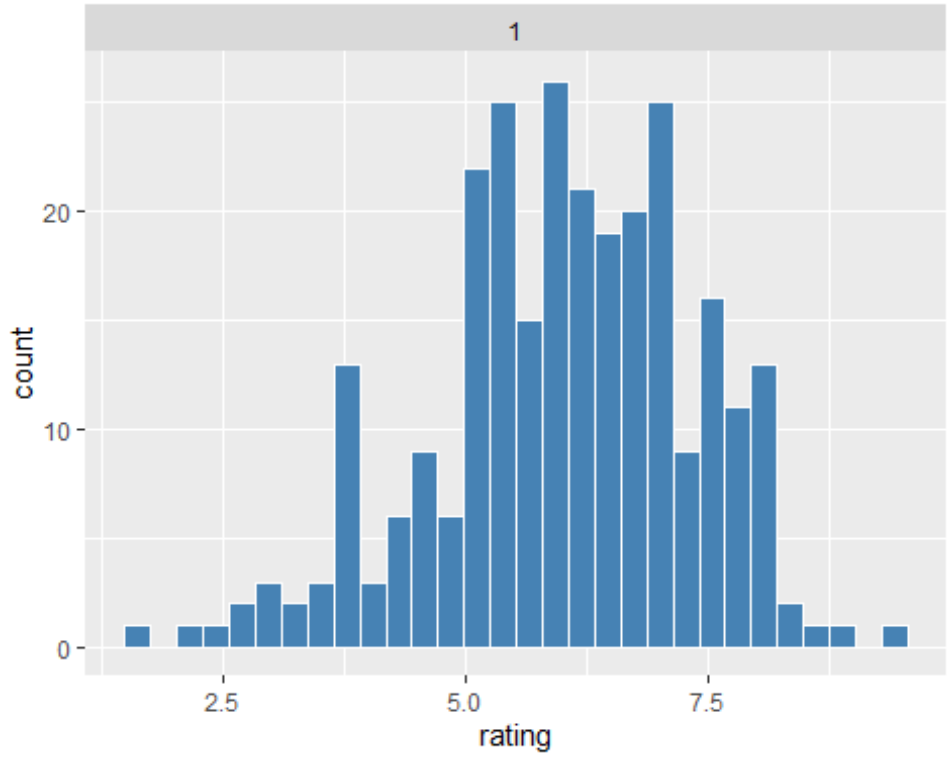
(LC 9.14) : What is the value of the p-value for the hypothesis test comparing the mean rating of romance to action movies ?
Ans : The p-value here is **0.004** .

(LC 9.15) : Test your data wrangling knowledge and EDA skills: * Use *dplyr* and *tidyr* to create the necessary data frame focused on only action and romance movies (but not both) from the movies data frame in the *ggplot2movies* package. * Make a boxplot and a faceted histogram of this population data comparing ratings of action and romance movies from IMDb. * Discuss how these plots compare to the similar plots produced for the movies_sample data.
Ans Use *dplyr* and *tidyr* to create the necessary data frame focused on only action and romance movies (but not both) from the movies data frame in the *ggplot2movies* package.

```
action_romance <- movies %>%
  filter(Action == 1 & Romance == 1)
```

Make a boxplot and a faceted histogram of this population data comparing ratings of action and romance movies from IMDb. # need a tidy dataset with genre

```
ggplot(action_romance, aes(rating)) +
  geom_histogram(col = "white" , fill = "steelblue") +
  facet_wrap(~Action)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



5.2.3 Two Sampled Test

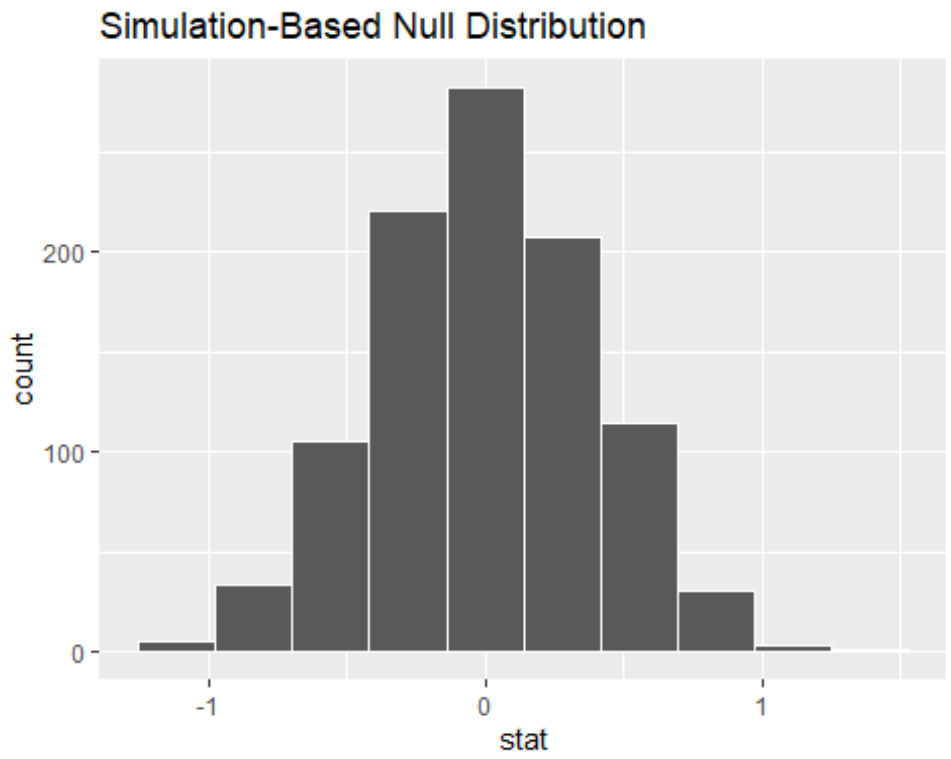
Two Sampled t- statistic

we are working on *movies_sample* data .

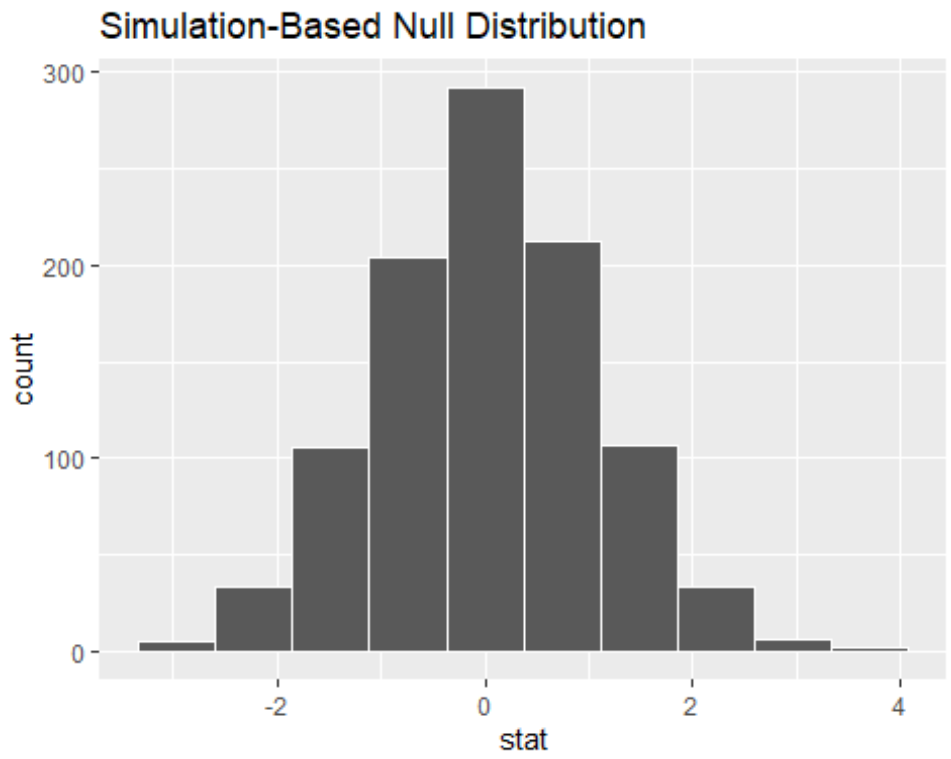
```
# Statistics summary
movies_sample %>%
  group_by(genre) %>%
  summarize(n = n(),
            mean_rating = mean(rating),
            std_dev = sd(rating))
```

```
## # A tibble: 2 x 4
##   genre       n mean_rating std_dev
##   <chr>   <int>       <dbl>   <dbl>
## 1 Action     32        5.28    1.36
## 2 Romance    36        6.32    1.61
```

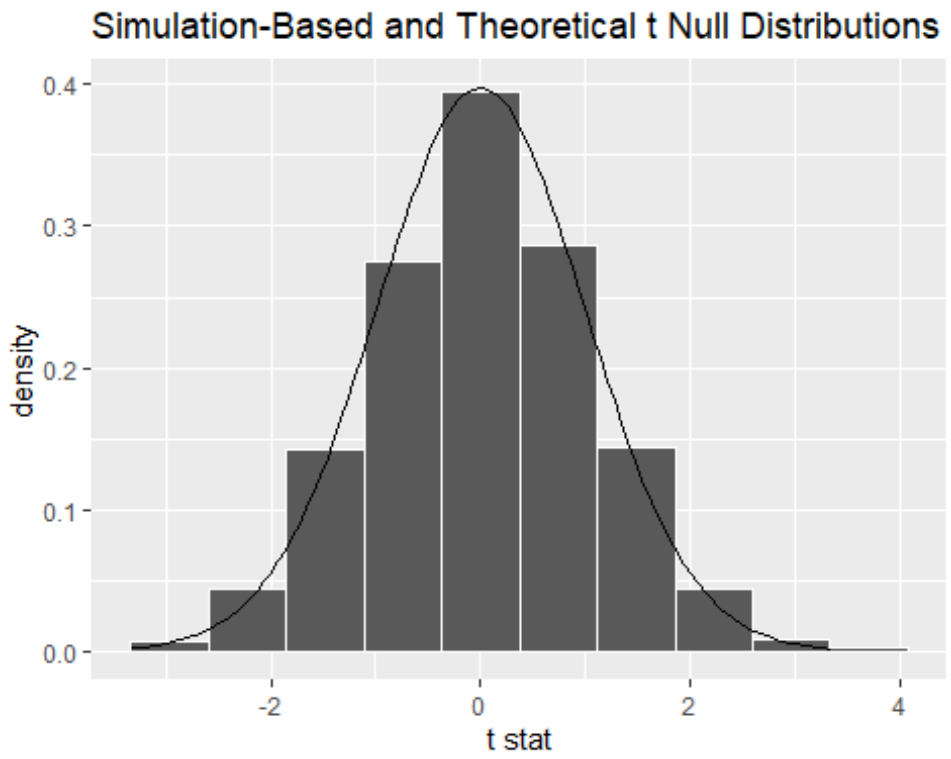
```
# Construct null distribution of xbar_a - xbar_m:
null_distribution_movies <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("Action", "Romance"))

visualize(null_distribution_movies, bins = 10)
```




```r
# Construct null distribution of t:
null_distribution_movies_t <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%

  # Notice we switched stat from "diff in means" to "t"
  calculate(stat = "t", order = c("Action", "Romance"))

visualize(null_distribution_movies_t, bins = 10)
```




```
visualize(null_distribution_movies_t, bins = 10, method = "both")

## Warning: Check to make sure the conditions have been met for the
## theoretical method. {infer} currently does not check these for you.
```

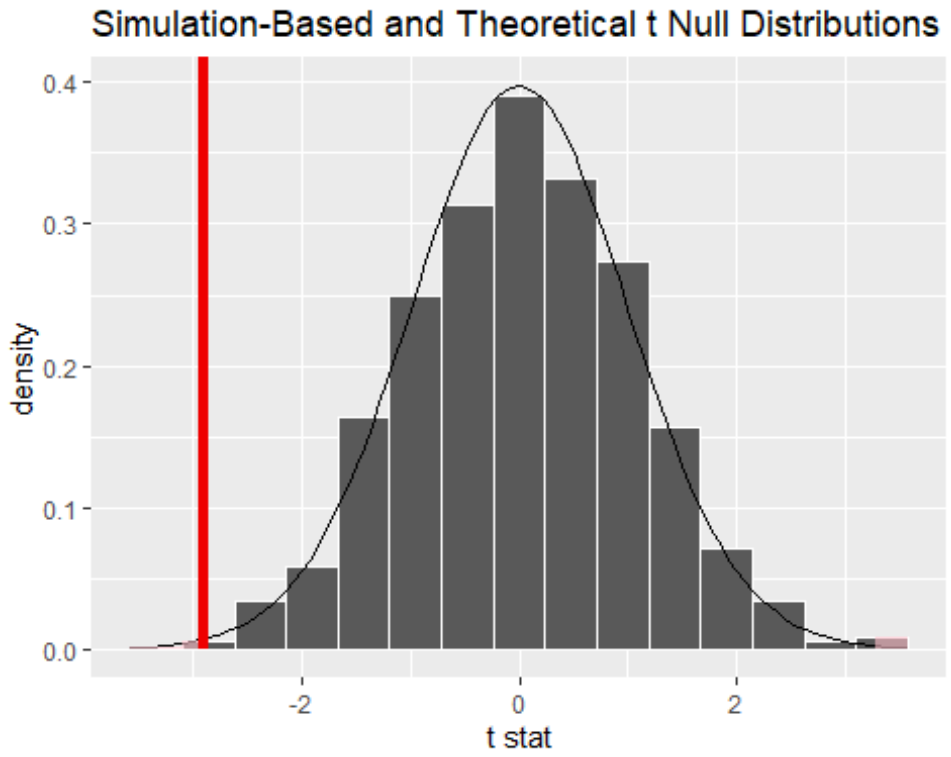



```
obs_two_sample_t <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  calculate(stat = "t", order = c("Action", "Romance"))

obs_two_sample_t
```

```
## Response: rating (numeric)
## Explanatory: genre (factor)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1 -2.91
```

```
# visualize
visualize(null_distribution_movies_t, method = "both") +
  shade_p_value(obs_stat = obs_two_sample_t, direction = "both")
```

```
## Warning: Check to make sure the conditions have been met for the
## theoretical method. {infer} currently does not check these for you.
```




```
# p-value
null_distribution_movies_t %>%
  get_p_value(obs_stat = obs_two_sample_t, direction = "both")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.004
```

Note : Next Part is on *Next* file