Training Intervention Analysis

Aditi Vilas Sonkusare, id=22224352

14 November, 2022

Before you start: if you are a Mac user, you will need to install Xquartz from https://www.xquartz.org so you can use the 'tolerance' package. You can delete this line from your final report.

Context: Celtic Study

A sample of 18 full-time youth soccer players from a Youth Academy performed high intensity aerobic interval training over a 10-week in-season period in addition to usual regime of soccer training and matches.

The aim of this study to find if this extra training improves V_IFT, the maximum velocity (km/hr) achieved in an intermittent fitness test (`VIFT_Pre` vs `VIFT_Post`)?

This is a **paired design:** each player's V_IFT measured before and after the training intervention (i.e. start and after 10 weeks)

A scaffold for the analysis with the response variable VO2 max is provided below. You need to rerun the analysis using the V_IFT variables (i.e. `VIFT_Pre` vs `VIFT_Post`) to answer the question of interest: is there, on average, an improvement in V_IFT? To assess the evidence, you will provide confidence intervals, and other statistical inference, for the mean improvement of players in the population (eg of future youth soccer players under the same training intervention).

To answer the question of inter est, provide a detailed response for all of the tasks asked below using the V_IFT variables (i.e. `VIFT_Pre` vs `VIFT_Post`).

Task: State the appropriate null and alternative hypotheses for the V_IFT study. -> Null Hypotheses : Null hypotheses indicates that there is no relation between the variables and there is no significant difference in the average. Null Hypotheses is known as Ho, where μ1 = mean of population 1, and μ0 = mean of population 2. Therefore, μ1=μ0 -> Alternative Hypotheses : Alternative hypotheses indicates that there is a relation between the two variables and because their is relation, there is significant difference in the average. Alternative hypotheses is known as Ha, where μ1 = mean of population 1, and μ0 = mean of population 2. Therefore, μ1!=μ0

Here, in this study if conclusion is null hypotheses means the value of mean of VIFT_Pre and VIFT_Post are the same. And if, the conclusion is alternative hypotheses then the value of mean of VIFT_Pre and VIFT_Post are not the same i.e. there is some improvement in the mean of VIFT_Pre and VIFT_Post.

Task: Define a Type I and Type II error and discuss the implication of making these errors in this study. -> Type I : Type I error is false-positive error occurs when null hypotheses is rejected but is actually true in population. It is also known as α error. -> Type II : Type II error is false-negative error occurs when we fail to reject null hypotheses but is actually false in population. It is also known as β Error.

Here, in this study we reject null hypotheses as p-value is less than 0.05. Type I error would have occured when we would have rejected the null hypotheses but actual result would be accepting null hypotheses. Type II error would have occured when we would have accepted the null hypotheses but the actual result would be rejecting the null hypotheses.

Read in the training intervention data

Read in the data and have a look at the variable names and structure of the data.

```
train.df <- read.csv("Training_intervention_data.csv")
glimpse(train.df)
```

```
## Rows: 18
## Columns: 5
## $ ID           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17…
## $ VO2.max_Pre  <dbl> 66.4, 70.9, 64.9, 68.6, 76.7, 75.6, 78.1, 73.1, 74.4,
64.…
## $ VO2.max_Post <dbl> 67.8, 81.7, 70.1, 73.0, 84.5, 78.4, 80.5, 76.0, 78.7,
72.…
## $ VIFT_Pre     <dbl> 23.8, 28.3, 25.2, 26.9, 30.1, 29.9, 29.5, 30.2, 31.5,
22.…
## $ VIFT_Post    <dbl> 23.3, 33.3, 25.8, 30.2, 34.6, 32.7, 31.8, 30.6, 32.0,
27.…
```

Focus on the V_IFT response variables

Summary Statistics

```
train.df %>% select(VIFT_Pre,VIFT_Post) %>% summary()
```

```
##     VIFT_Pre       VIFT_Post
##  Min.   :21.40   Min.   :22.60
##  1st Qu.:23.65   1st Qu.:25.43
##  Median :25.25   Median :27.20
##  Mean   :26.02   Mean   :28.08
##  3rd Qu.:29.20   3rd Qu.:31.50
##  Max.   :31.50   Max.   :34.60
```

Task: Interpret! -> Min.: This is the minimum value from the data. The difference between VIFT_Pre and VIFT_Post is +1.2. Hence, Minimum value is improved 1st Qu.: The first quartile i.e. 25% of values. The difference between VIFT_Pre and VIFT_Post is +1.78. Hence 1st Qu is improved Median: The median value i.e. 50% of the values. The difference between VIFT_Pre and VIFT_Post is +1.95. Hence 1st Median is improved Mean: The average value. The difference between VIFT_Pre and VIFT_Post is +2.06. Hence Mean is

improved 3rd Qu.: the third quartile i.e. 75% of values are higher than this. The difference between VIFT_Pre and VIFT_Post is +2.3. Hence 3rd Qu is improved Max.: the maximum value from the data. The difference between VIFT_Pre and VIFT_Post is +3.10. Hence Maximum value is improved To conclude, all the pre and post values here are increasing gradually.

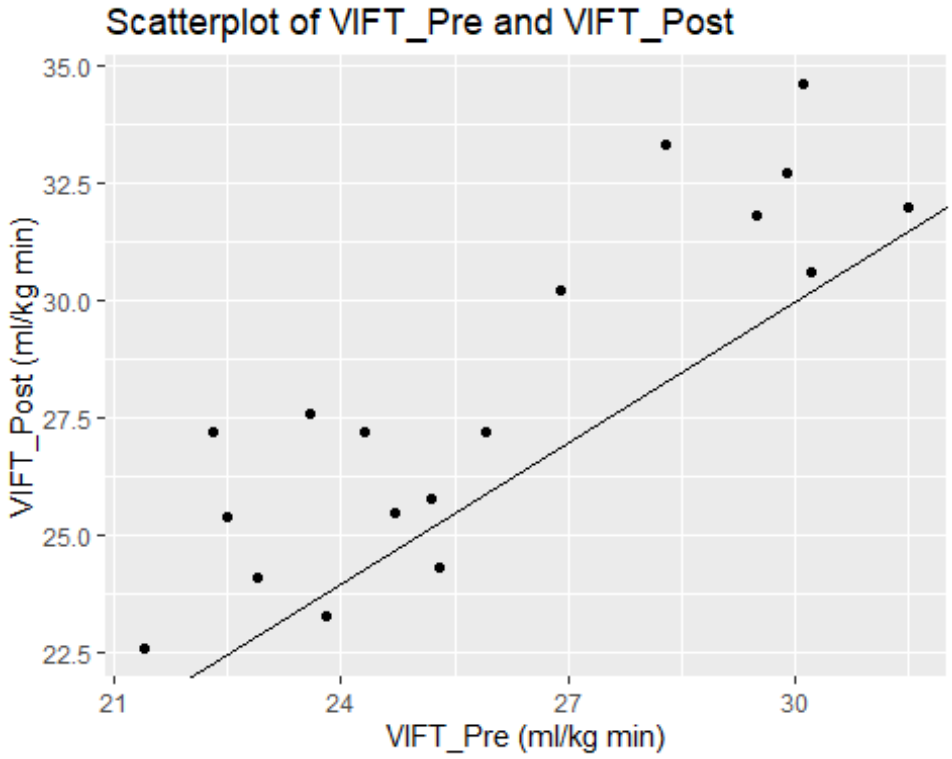
Mean and Standard Deviation

```
train.df %>% select(VIFT_Pre,VIFT_Post) %>%
            summarize(Pre_Mean=mean(VIFT_Pre), Pre_SD= sd(VIFT_Pre),
                      Post_Mean=mean(VIFT_Post), Post_SD= sd(VIFT_Post))

##   Pre_Mean   Pre_SD Post_Mean  Post_SD
## 1 26.01667 3.174207  28.07778 3.723938
```

Task: Interpret! -> Here, difference between Pre_Mean and Post_Mean is +2.061111 and difference between Pre_SD and Post_SD is +0.545731. To conclude, the pre value and post value have a positive difference in the improvement.

Scatterplot of Pre and Post with line of equality

```
train.df %>% ggplot(aes(x = VIFT_Pre, y = VIFT_Post)) +
        geom_point() +
  ggtitle("Scatterplot of VIFT_Pre and VIFT_Post") +
  ylab("VIFT_Post (ml/kg min)") +
  xlab("VIFT_Pre (ml/kg min)") +
  geom_abline(slope=1, intercept=0)
```



Task: Interpret! -> From this scatterplot, the data points forms a straight line going from near to the origin out

to high y-values. Hence, the variables here are said to be low positively correlated as there data points are scattered and are away from the straight line(if the data points would be close to the straight line, we could have said that the variables are highly positively correlated). Hence, we can conclude that as the VIFT_Pre increases the VIFT_Post is also increasing.

Calculate the Improvement in V_IFT

Calculate a new variable, "improvement", and have a look at the data frame to see that it has been created. High values of VO2 max are good so Post-Pre is a better measure than Pre-Post to capture this - what about V_IFT?

```
train.df <- train.df %>% mutate(Improvement = VIFT_Post-VIFT_Pre) %>%
              glimpse()

## Rows: 18
## Columns: 6
## $ ID           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17…
## $ VO2.max_Pre  <dbl> 66.4, 70.9, 64.9, 68.6, 76.7, 75.6, 78.1, 73.1, 74.4,
64.…
## $ VO2.max_Post <dbl> 67.8, 81.7, 70.1, 73.0, 84.5, 78.4, 80.5, 76.0, 78.7,
72.…
## $ VIFT_Pre     <dbl> 23.8, 28.3, 25.2, 26.9, 30.1, 29.9, 29.5, 30.2, 31.5,
22.…
## $ VIFT_Post    <dbl> 23.3, 33.3, 25.8, 30.2, 34.6, 32.7, 31.8, 30.6, 32.0,
27.…
## $ Improvement  <dbl> -0.5, 5.0, 0.6, 3.3, 4.5, 2.8, 2.3, 0.4, 0.5, 4.9, -
1.0, …
```

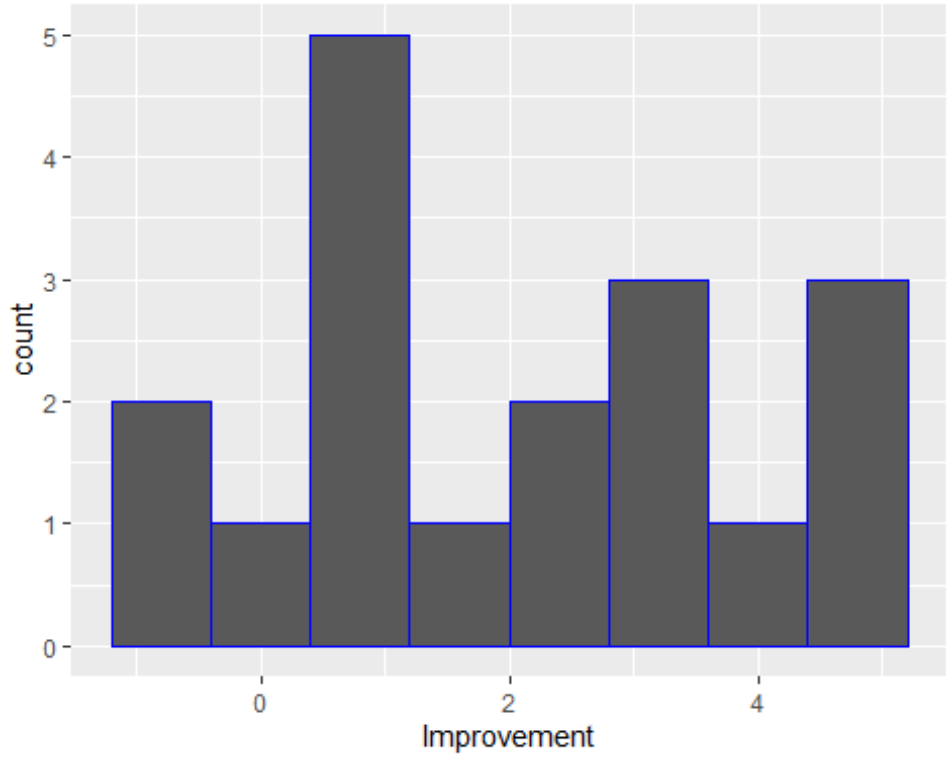
Mean and Standard Deviation of Improvement in V_IFT

```
train.df %>% select(Improvement) %>%
            summarize(Imp_Mean=mean(Improvement), Imp_SD= sd(Improvement))

##   Imp_Mean   Imp_SD
## 1 2.061111 1.828898
```

Task: Interpret! -> The new mean i.e. Improved mean gives 2.06. This value is much less value than the previous mean because improved mean is the difference value between the

```
train.df %>% select(Improvement) %>% summary()

##   Improvement
##  Min.   :-1.000
##  1st Qu.: 0.650
##  Median : 1.800
##  Mean   : 2.061
##  3rd Qu.: 3.200
##  Max.   : 5.000
```

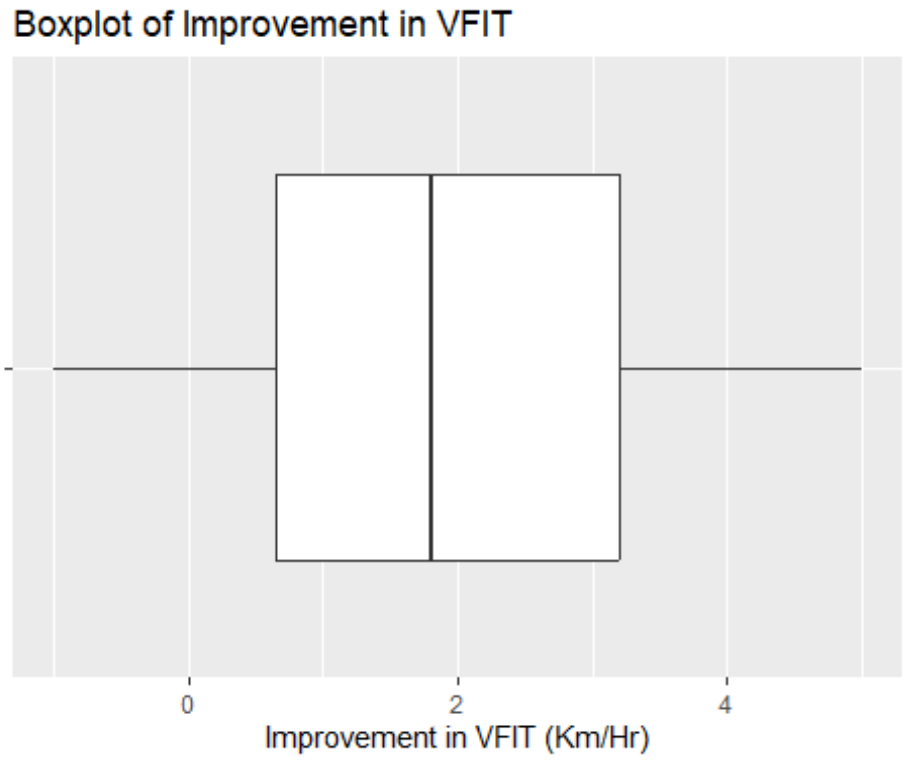
```
# Observing the summary for improvement to show more clear justification in
box plot.

ggplot(train.df, aes(x = Improvement)) +
  geom_histogram(binwidth = 0.8, color = "blue")
```




```
# Plotting histogram for Improvement to show more clear justification for box
plot.
```

Boxplot of Improvement in V_IFT

```
train.df %>% ggplot(aes(x = "", y = Improvement)) +
        geom_boxplot() +
  ggtitle("Boxplot of Improvement in VFIT") +
  ylab("Improvement in VFIT (Km/Hr)") +
  xlab("") +
  coord_flip()
```



Task: Interpret! ->
The line which is dividing the box into 2 parts is the median of the data. Here, the median is 1.80, it means that there are less number of data points below 1.80 and more number of data points above 1.80. The edges(end lines) of the box shows the lower(Q1) and upper(Q3) quartiles. If the third quartile is 3.20, it clearly states that 75% of the observations are lower than 3.20 because the value of 3rd Quartile is 3.20. The difference between Quartile1 and Quartile3 is known as the 'Inter-Quartile Range'(IQR). The flat line shows Q3+1.5xIQR to Q1-1.5xIQR which are the highest and lowest value respectively excluding outliers. Outliers are represented in the form of Dots(or other markers) beyond the extreme line shows potential outliers which are not present in this data given

95% Confidence Interval Using the t.test function

```
train.df %>% select(Improvement) %>% t.test()

##
##  One Sample t-test
##
## data:  .
## t = 4.7813, df = 17, p-value = 0.0001736
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.151621 2.970601
## sample estimates:
## mean of x
##  2.061111
```

Task: Based on the output given answer the following questions: -> In the above output, t is the t-test statistic value which is 4.7813, df is the degrees of freedom which is 17 and p-value is the level of significance of the t-test which is 0.0001736. Confidence Interval of the mean at 95% which is ([1.151621,2.970601]) and lastly, and final mean value of the sample is 2.061111.

- What is the mean improvement in V_IFT the population of interest? Interpret the relevant 95% Confidence Interval carefully. -> As we can see in the previous improved mean of V_IFT gives value = 2.061111 and after performing the t-test, we get the mean value as 2.061111 which is same. So there is no change in the mean value after performing the t-test. Here, we get p-value as 0.0001736 which is less that 0.05. Hence, we reject the null hypotheses. So, Confidence Interval of mean is sample standard deviation(S) which is divided by square root of sample size(under-root'n') which gives standard error. We further multiply the standard error with t-value which gives margin of error and then we add and subtract from sample mean to get confidence interval. Confidence Interval gives a range of values in which we can be confident which contains the mean. We can clearly state how confident we are by the confidence interval. Therefore, in this case we can be 95% sure that the confidence interval is between 1.151621 to 2.970601 and the mean value that we get is 2.061111 which is in between the confidence interval.

- Use the relevant interval estimate and p-value to decide whether there is sufficient evidence in the sample provided to claim that there is any improvement on average in V_IFT in the population of interest. -> The relevant interval estimate shown in t-test is 1.151621 to 2.970601 and the mean shown in the output of t-test is 2.061111 and previous improved mean which we got also gives an output of 2.061111 which is same value. So, when we are getting the same values of mean there is no improvement on average in V_IFT in the population of interest. From the output, the p-value is less than the significance level 0.05, hence we can conclude that the distribution of the data are significant and indicates relation between the variables.

- What are the assumptions underlying the one sample t-test presented? -> Usually. there are two assumptions made in One-Sample t-test

1) No significant outliers in the data
2) Normality. the data should be approximately normally distributed So, from the first assumption, we can see that there no significant outliers in the data as there are no dots or other marks before or after the flat line in the Box-Plot which satisfies the assumption. From the second assumption, we can see that the data is not normally distributed as we caan see the box plot is to the left side and observing the histogram plotted we can see it is skewed on right side, this is not normally distributed i.e. from the histogram, we can see that the histogram is right-skewed, which does not satisfies the assumption.

- Explain why or why not the assumptions seem justified based on the output provided. -> As we can see in the box plot, there are no outliers before or after the flat line we can assume that 'No significant outliers in the data' satisfies and assumption on 'Normality' does not satisfy as the boxplot is towards left side and

hence histogram is right-skewed. Hence, Assumption 1 is justified by seeing the box plot and the output from one-sample t-test.

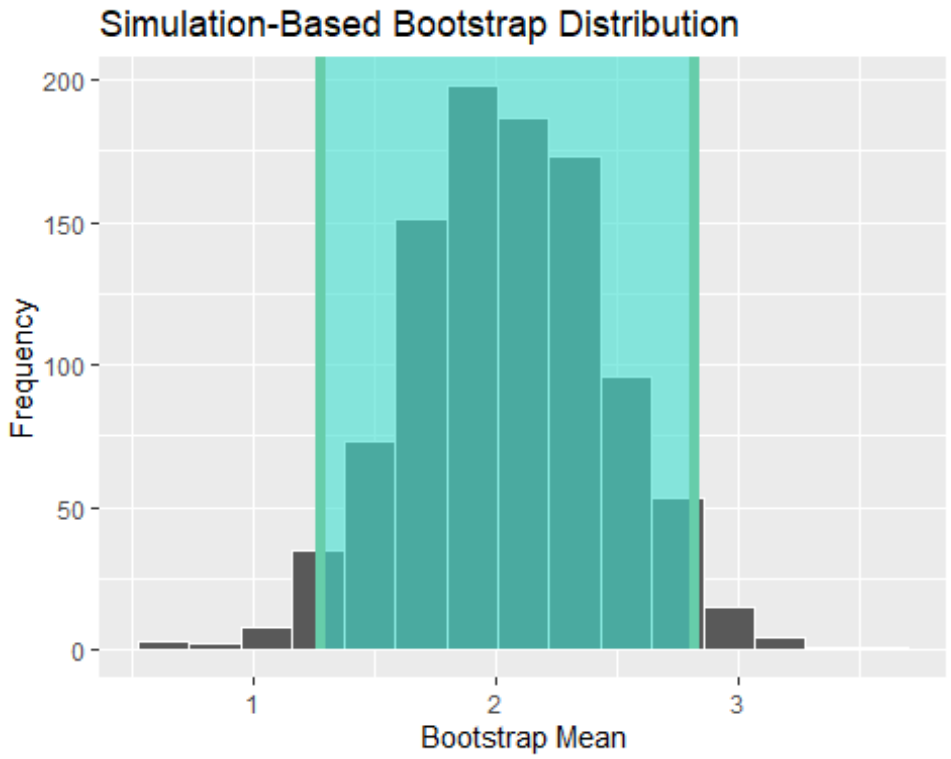
95% Bootstrap CI for the mean

```
boot <- train.df %>%
  specify(response = Improvement) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "mean")

percentile_ci <- get_ci(boot)
round(percentile_ci,2)

## # A tibble: 1 × 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1     1.28     2.82

#train.df <- train.df %>% mutate(Bootstrap_meanci = percentile_ci) %>%
#              glimpse()
#train.df %>%  summary(percentile_ci)

# Checked the overall mean for bootstrapping
```

Task: Interpret! -> Bootstrapping is an inferential statistics technique that involves repeatedly creating random samples from a single dataset. Bootstrapping enables the calculation of sampling measures such as mean, median, mode, confidence intervals, and so on. It generates a set of values or an interval in which the true value is always present. 95% bootstrapping means the lower endpoint of the confidence interval is set at the 2.5th percentile of the bootstrap distribution, and the upper endpoint is set at the 97.5th percentile.

Here, we are showing stat='mean' method for constructing 95% confidence intervals(ci). We can compute the 95% confidence interval by passing response="Improvement" because we want to observe the 95% Bootstrap CI for the mean on this variable and calculating stat="mean" we will get bootstrap for mean on the variable Improvement.The endpoints argument is set to "percentile_ci" and is rounded of up to two decimal points. To conclude, here we get the 95% Bootstrap CI for the mean between 1.28 to 2.82 by type="bootstrap"

```
boot %>% visualize()+
  shade_confidence_interval(endpoints = percentile_ci) +
                  xlab("Bootstrap Mean") + ylab("Frequency")
```



Task: Interpret! -> Here, in the simulation based Bootstrap distributions, that the Bootstrap contains 95% of the sample mean which falls between the two endpoints marked by the darker lines, with 2.5% of the sample means to the left and 2.5% to the right of the shaded area. We can also say that the 95% confidence interval for the percentile method is shaded to correspond to distributions.

95% Bootstrap CI for the median improvement

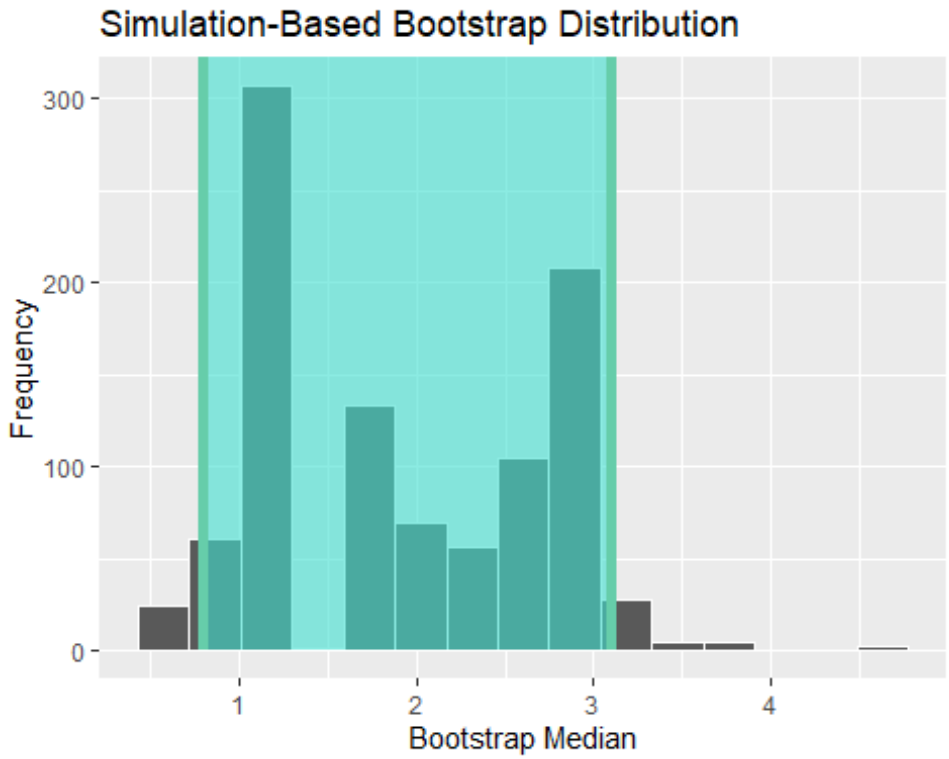
```
boot.median <- train.df %>%
  specify(response = Improvement) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "median")

percentile_ci_median <- get_ci(boot.median)
round(percentile_ci_median,2)
```

```
## # A tibble: 1 × 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1      0.8      3.1
```

Task: Interpret! -> A confidence interval (CI) is a type of computational value calculated on sample data. It generates a set of values or an interval in which the true value is always present. 95% bootstrapping means the lower endpoint of the confidence interval is set at the 2.5th percentile of the bootstrap distribution, and the upper endpoint is set at the 97.5th percentile. In the bootstrap distribution, the resulting interval captures the middle 95% of the sample mean values.

Here, we are showing stat='median' method for constructing 95% confidence intervals(ci). We can compute the 95% confidence interval by passing response="Improvement" because we want to observe the 95% Bootstrap CI for the median on this variable and calculating stat="median" we will get bootstrap for mean on the variable Improvement.The endpoints argument is set to "percentile_ci_median" and is rounded of up to two decimal points. To conclude, here we get the 95% Bootstrap CI for the median between 0.7 to 3.1 by type="bootstrap"

```
boot.median %>% visualize()+
  shade_confidence_interval(endpoints = percentile_ci_median) +
                    xlab("Bootstrap Median") + ylab("Frequency")
```



Task: Interpret! ->
Here, in the simulation based Bootstrap distributions, that the Bootstrap contains 95% of the sample median which falls between the two endpoints marked by the darker lines, with 2.5% of the sample means to the left and 2.5% to the right of the shaded area. We can also say that the 95% confidence interval for the percentile method is shaded to correspond to distributions.

95% Tolerance Interval (Bonus Question)

Calculate a 95% tolerance interval covering 95% of V_IFT improvement values

```
normtol.int(train.df$Improvement, alpha = 0.05, P = 0.95)
```

```
##   alpha    P    x.bar 1-sided.lower 1-sided.upper
## 1  0.05 0.95 2.061111      -2.42508      6.547302
```

Task: Interpret! -> The tolerance interval is the range of values for a distribution with confidence limits calculated to a specific percentile of the distribution. This library is installed in the beginning of the code itself. To calculate a tolerance interval for data from a normal distribution we use use the tolerance package function “normtol.int”. The function arguments contain the data in the form of a vector denoted as we want the data for Improvement in train.df. The tolerance interval’s confidence level is specified by alpha, where alpha is the difference between 100% and the confidence level - alpha is 0.05 for the 95% confidence. The argument P represents the percentage of data to be included in the tolerance interval. Since, the side arguement is not passed, it takes side=1 by default. (The side argument specifies whether the interval can be 1-sided or 2-sided. The alpha and P are as stated above, and the average of the data, as well as the lower and upper tolerance intervals, are reported in this case because we requested a 1-sided interval)

Here, we get alpha as 0.05, P as 0.95, x.bar as 2.061111, 1-sided.lower as -2.42508 and 1-sided.upper as 6.547302.

Overall Conclusion

Task: state your overall conclusion.

```
t.test(train.df$VIFT_Pre)

##
##  One Sample t-test
##
## data:  train.df$VIFT_Pre
## t = 34.774, df = 17, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  24.43817 27.59516
## sample estimates:
## mean of x
##  26.01667

t.test(train.df$VIFT_Post)

##
##  One Sample t-test
##
## data:  train.df$VIFT_Post
## t = 31.989, df = 17, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  26.22591 29.92965
## sample estimates:
## mean of x
##  28.07778
```

Here, for the final conclusion, we are checking with t.test for seperate variables for VIFT_Pre and VIFT_post we can see the improvement in VIFT_Pre and VIFT_Post, which

means that we reject the null hypotheses as there is significant difference in the means of before and after values.