Project

```r
library(binom)
library(collapsibleTree)
library(dbplyr)
library(devtools)
```

```
## Loading required package: usethis
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:dbplyr':
##
##     ident, sql
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```r
library(ggformula)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: ggstance
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```r
library(ggplot2)
library(ggpubr)
library(htmltools)
library(ISLR)
library(knitr)
library(lawstat)
library(markdown)
library(mosaic)
```

```
## Loading required package: lattice

## Loading required package: mosaicData

## Loading required package: Matrix

## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
##
## Have you tried the ggformula package for your plots?

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following object is masked from 'package:EnvStats':
##
##     iqr

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```r
library(mdsr)
library(mosaicData)
library(nycflights13)
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
## The following object is masked from 'package:mosaic':
##
##     count
## The following object is masked from 'package:ggpubr':
##
##     mutate
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(purrr)
```

```
##
## Attaching package: 'purrr'
## The following object is masked from 'package:plyr':
##
##     compact
## The following object is masked from 'package:mosaic':
##
##     cross
```

```
library(readr)
library(rmarkdown)
library(stringi)
library(tibble)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
library(tidyselect)
library(tinytex)
library(yaml)
```

```
game_teams_stats = read_csv("/home/jovyan/DataScience602/Project/game_teams_stats.csv")
```

```
## Parsed with column specification:
## cols(
##   game_id = col_double(),
##   team_id = col_double(),
##   HoA = col_character(),
##   won = col_logical(),
##   settled_in = col_character(),
##   head_coach = col_character(),
##   goals = col_double(),
##   shots = col_double(),
##   hits = col_double(),
##   pim = col_double(),
##   powerPlayOpportunities = col_double(),
##   powerPlayGoals = col_double(),
##   faceOffWinPercentage = col_double(),
##   giveaways = col_double(),
##   takeaways = col_double()
## )
```

```r
game_data = read_csv("/home/jovyan/DataScience602/Project/game.csv")
```

```
## Parsed with column specification:
## cols(
##   game_id = col_double(),
##   season = col_double(),
##   type = col_character(),
##   date_time = col_date(format = ""),
##   date_time_GMT = col_datetime(format = ""),
##   away_team_id = col_double(),
##   home_team_id = col_double(),
##   away_goals = col_double(),
##   home_goals = col_double(),
##   outcome = col_character(),
##   home_rink_side_start = col_character(),
##   venue = col_character(),
##   venue_link = col_character(),
##   venue_time_zone_id = col_character(),
##   venue_time_zone_offset = col_double(),
##   venue_time_zone_tz = col_character()
## )
```

```r
team_info = read_csv("/home/jovyan/DataScience602/Project/team_info.csv")
```

```
## Parsed with column specification:
## cols(
##   team_id = col_double(),
##   franchiseId = col_double(),
##   shortName = col_character(),
##   teamName = col_character(),
##   abbreviation = col_character(),
##   link = col_character()
## )
```

```r
# Selecting the specific columns needed from the game_team_stats data frame
game_teams_stats = game_teams_stats %>%
  select(game_id, team_id, HoA, settled_in, won, goals, shots, pim)

# Selecting the specific columns needed from the game_data data frame
game_data = game_data %>%
  select(game_id, season, date_time, home_goals, away_goals)

# Selecting the specific columns needed from the team_info data frame
team_info = team_info %>%
  select(team_id, shortName, teamName)

# Merging all rows and columns from game_team_stats and game_data by the column game_id
hockey_data = game_teams_stats %>%
  full_join(game_data, by = 'game_id')

# Cleaning team_info data frame
team_info$shortName = gsub('NY Rangers', 'NY', team_info$shortName) # replace NY Rangers with NY
team_info$shortName = gsub('NY Islanders', 'NY', team_info$shortName) # replace NY Islanders with NY
team_info$team = paste(team_info$shortName, team_info$teamName) # creating column 'team' and combining

# Merging all rows and columns from hockey_data and team_info by the column team_id
hockey_data = hockey_data %>%
  full_join(team_info, by = 'team_id')

# Selecting the specific columns needed from hockey_data data frame
game_data = hockey_data %>%
  select(game_id, season, team, HoA, won, home_goals, away_goals, shots)

# Selecting the specific columns needed from hockey_data data frame
pim_data = hockey_data %>%
  select(game_id, HoA, date_time, season, team, pim)
```

```
# Renaming the columns of both data frames
colnames(game_data) = c("Game_ID", "Season", "Team", "Home_Or_Away", "Won", "Home_Goals", "Away_Goals",
colnames(pim_data) = c("Game_ID", "Home_or_Away", "Date_Time", "Season", "Team", "PIM")

# Creating Before_or_After column where there is a 'Before' before the Wideman game and 'After' after t
pim_data$Before_or_After = ifelse(pim_data$Date_Time < '2016-01-27', 'Before', 'After')

pim_data
```

```
## # A tibble: 22,868 x 7
##       Game_ID Home_or_Away Date_Time   Season Team            PIM Before_or_After
##         <dbl> <chr>        <date>       <dbl> <chr>         <dbl> <chr>
##  1    2.01e9 away         2012-04-29  2.01e7 New Jersey D~    12 Before
##  2    2.01e9 home         2012-04-29  2.01e7 Philadelphia~     6 Before
##  3    2.01e9 away         2012-05-01  2.01e7 New Jersey D~    12 Before
##  4    2.01e9 home         2012-05-01  2.01e7 Philadelphia~    32 Before
##  5    2.01e9 away         2012-05-03  2.01e7 Philadelphia~     4 Before
##  6    2.01e9 home         2012-05-03  2.01e7 New Jersey D~    10 Before
##  7    2.01e9 away         2012-05-06  2.01e7 Philadelphia~    10 Before
##  8    2.01e9 home         2012-05-06  2.01e7 New Jersey D~     4 Before
##  9    2.01e9 away         2012-05-08  2.01e7 New Jersey D~     2 Before
## 10    2.01e9 home         2012-05-08  2.01e7 Philadelphia~     8 Before
## # ... with 22,858 more rows
```

```
game_data
```

```
## # A tibble: 22,868 x 8
##       Game_ID   Season Team         Home_Or_Away Won   Home_Goals Away_Goals Shots
##         <dbl>   <dbl> <chr>        <chr>        <lgl>      <dbl>      <dbl> <dbl>
##  1    2.01e9  2.01e7 New Jersey ~ away         FALSE          4          3    26
##  2    2.01e9  2.01e7 Philadelphi~ home         TRUE           4          3    36
##  3    2.01e9  2.01e7 New Jersey ~ away         TRUE           1          4    35
##  4    2.01e9  2.01e7 Philadelphi~ home         FALSE          1          4    20
##  5    2.01e9  2.01e7 Philadelphi~ away         FALSE          4          3    28
##  6    2.01e9  2.01e7 New Jersey ~ home         TRUE           4          3    31
##  7    2.01e9  2.01e7 Philadelphi~ away         FALSE          4          2    22
##  8    2.01e9  2.01e7 New Jersey ~ home         TRUE           4          2    43
##  9    2.01e9  2.01e7 New Jersey ~ away         TRUE           1          3    30
## 10    2.01e9  2.01e7 Philadelphi~ home         FALSE          1          3    28
## # ... with 22,858 more rows
```

Wideman Effect - Calgary Flames in the 2015-2016 season.

```
season_wideman_df = filter(pim_data, Season == '20152016', Team == 'Calgary Flames') #getting only the
season_wideman_df = arrange(season_wideman_df, Before_or_After, decreasing = FALSE)
season_wideman_df
```

```
## # A tibble: 82 x 7
##       Game_ID Home_or_Away Date_Time   Season Team           PIM Before_or_After
##         <dbl> <chr>        <date>       <dbl> <chr>        <dbl> <chr>
##  1 2015021050 home         2016-03-17  2.02e7 Calgary Fla~     9 After
##  2 2015021138 away         2016-03-29  2.02e7 Calgary Fla~    38 After
##  3 2015021087 away         2016-03-21  2.02e7 Calgary Fla~    14 After
##  4 2015020873 home         2016-02-20  2.02e7 Calgary Fla~    14 After
##  5 2015020824 away         2016-02-13  2.02e7 Calgary Fla~    20 After
##  6 2015021176 away         2016-04-03  2.02e7 Calgary Fla~    14 After
##  7 2015020742 home         2016-01-28  2.02e7 Calgary Fla~     8 After
##  8 2015021035 home         2016-03-15  2.02e7 Calgary Fla~    10 After
##  9 2015021126 home         2016-03-27  2.02e7 Calgary Fla~     8 After
## 10 2015020928 home         2016-02-28  2.02e7 Calgary Fla~     6 After
## # ... with 72 more rows
```

```
before_wideman_df = filter(season_wideman_df, Date_Time < '2016-01-27') #getting all the games in the 2
before_wideman_df
```

```
## # A tibble: 47 x 7
##       Game_ID Home_or_Away Date_Time   Season Team           PIM Before_or_After
##         <dbl> <chr>        <date>       <dbl> <chr>        <dbl> <chr>
##  1 2015020562 home         2016-01-01  2.02e7 Calgary Fla~     7 Before
##  2 2015020061 away         2015-10-17  2.02e7 Calgary Fla~     8 Before
##  3 2015020341 away         2015-11-28  2.02e7 Calgary Fla~     6 Before
##  4 2015020028 away         2015-10-11  2.02e7 Calgary Fla~    12 Before
##  5 2015020427 home         2015-12-11  2.02e7 Calgary Fla~     6 Before
##  6 2015020416 home         2015-12-09  2.02e7 Calgary Fla~     8 Before
##  7 2015020514 home         2015-12-23  2.02e7 Calgary Fla~     2 Before
##  8 2015020352 away         2015-11-29  2.02e7 Calgary Fla~     9 Before
##  9 2015020366 home         2015-12-02  2.02e7 Calgary Fla~     4 Before
## 10 2015020729 away         2016-01-26  2.02e7 Calgary Fla~     6 Before
## # ... with 37 more rows
```

```
after_wideman_df = filter(season_wideman_df, Date_Time > '2016-01-27') #getting all the games in the 20
after_wideman_df
```

```
## # A tibble: 35 x 7
##       Game_ID Home_or_Away Date_Time   Season Team           PIM Before_or_After
##         <dbl> <chr>        <date>       <dbl> <chr>        <dbl> <chr>
##  1 2015021050 home         2016-03-17  2.02e7 Calgary Fla~     9 After
##  2 2015021138 away         2016-03-29  2.02e7 Calgary Fla~    38 After
##  3 2015021087 away         2016-03-21  2.02e7 Calgary Fla~    14 After
##  4 2015020873 home         2016-02-20  2.02e7 Calgary Fla~    14 After
##  5 2015020824 away         2016-02-13  2.02e7 Calgary Fla~    20 After
##  6 2015021176 away         2016-04-03  2.02e7 Calgary Fla~    14 After
##  7 2015020742 home         2016-01-28  2.02e7 Calgary Fla~     8 After
##  8 2015021035 home         2016-03-15  2.02e7 Calgary Fla~    10 After
##  9 2015021126 home         2016-03-27  2.02e7 Calgary Fla~     8 After
## 10 2015020928 home         2016-02-28  2.02e7 Calgary Fla~     6 After
## # ... with 25 more rows
```

```
favstats(~PIM, data = before_wideman_df)
```

```
##  min Q1 median Q3 max     mean       sd  n missing
##    0  4      6  9  38 7.382979 5.780602 47       0
```

```
favstats(~PIM, data = after_wideman_df)
```

```
##  min  Q1 median Q3 max     mean       sd  n missing
##    2 6.5     10 14  38 11.62857 7.170551 35       0
```

Hypothesis Test: The null states that the average amount of penalty minutes before the game against Nashville is more than or equal to the average amount of penalty minutes after the game against Nashville, therefore stating that the Wideman effect did not exist in the 2015-2016 season for the Calgary Flames.

$$H_0 : \mu_{BeforeWideman} - \mu_{AfterWideman} \geq 0 \equiv \mu_{BeforeWideman} \geq \mu_{AfterWideman}$$

The alternative states that the average amount of penalty minutes before the game against Nashville is less than the average amount of penalty minutes after the game against Nashville, therefore stating that the Widemen effect did exist in the 2015-2016 season for the Calgary Flames.

$$H_A : \mu_{BeforeWideman} - \mu_{AfterWideman} < 0 \equiv \mu_{BeforeWideman} < \mu_{AfterWideman}$$

```
length(season_wideman_df$PIM)
```

```
## [1] 82
```

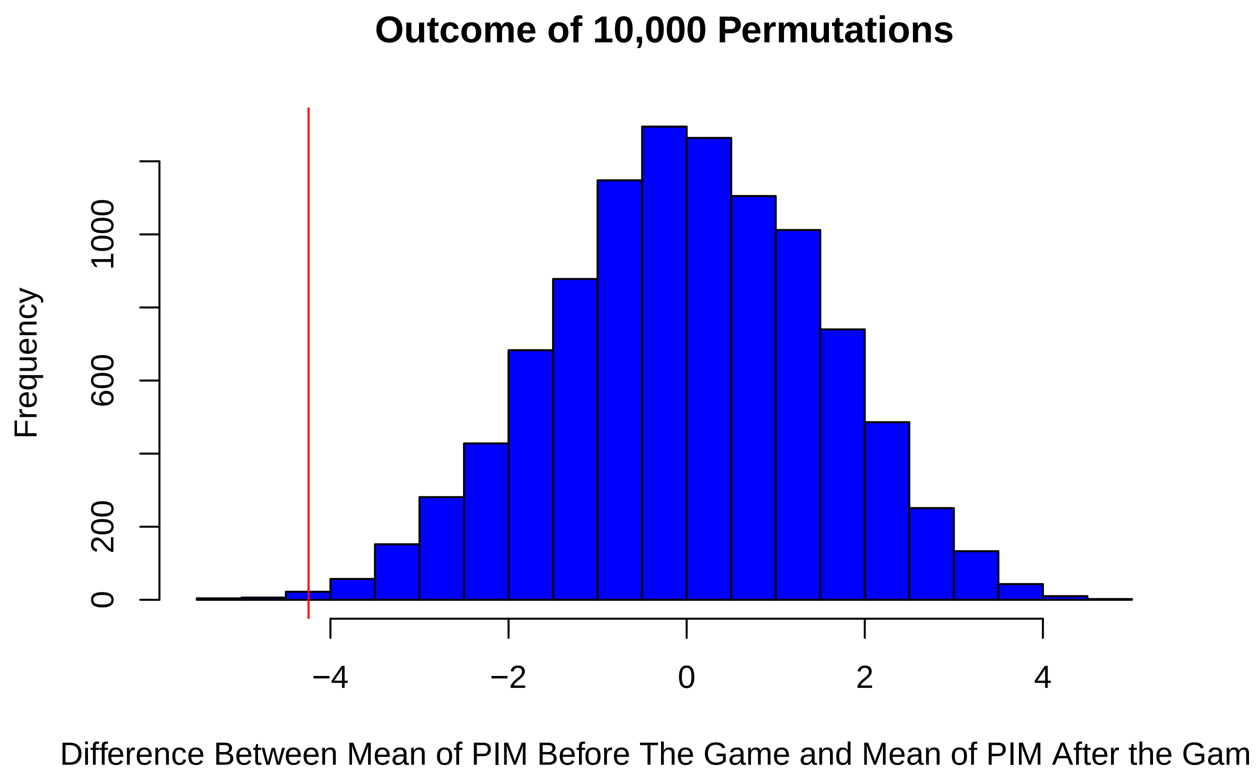
```
mean(~PIM, data = before_wideman_df) - mean(~PIM, data = after_wideman_df)
```

```
## [1] -4.245593
```

We will do a Permutation test:

```
observed_pim_diff = mean(~PIM, data = before_wideman_df) - mean(~PIM, data = after_wideman_df) # comput

N.Permutation = 10000 #doing 10000 permutations

pim_diff = numeric(N.Permutation)
n.season_wideman = length(season_wideman_df$PIM) #getting the amount of games for the 2015-2016 season
n.before_wideman = length(before_wideman_df$PIM) #getting the amount of games before the Nashville game

for(i in 1:N.Permutation){
  index = sample(n.season_wideman, n.before_wideman, replace = FALSE)
  pim_diff[i] = mean(season_wideman_df$PIM[index]) - mean(season_wideman_df$PIM[-index])
```

```
}

hist(pim_diff, xlab = 'Difference Between Mean of PIM Before The Game and Mean of PIM After the Game', 
abline(v = observed_pim_diff, col = 'red')
```



Getting the P-value

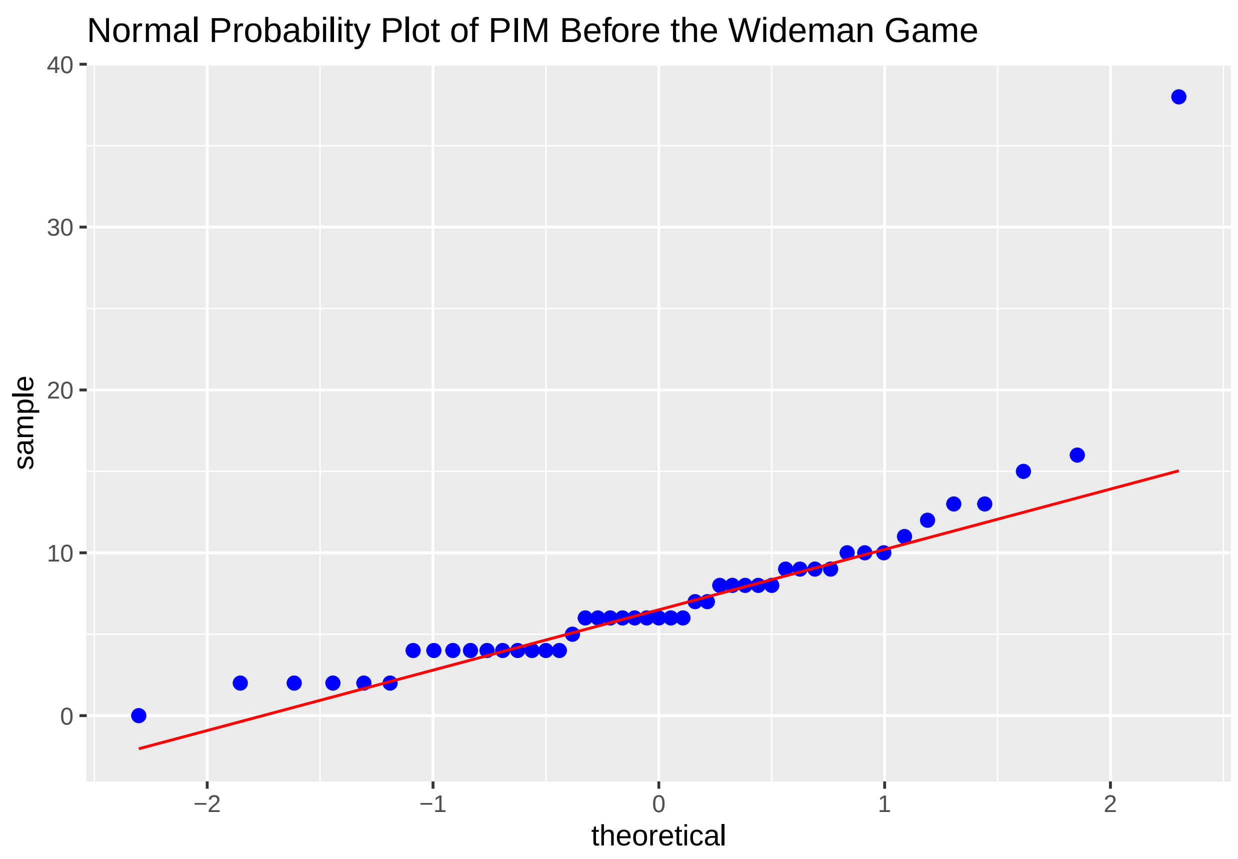
```
sum(pim_diff <= observed_pim_diff) / N.Permutation
```

```
## [1] 0.002
```

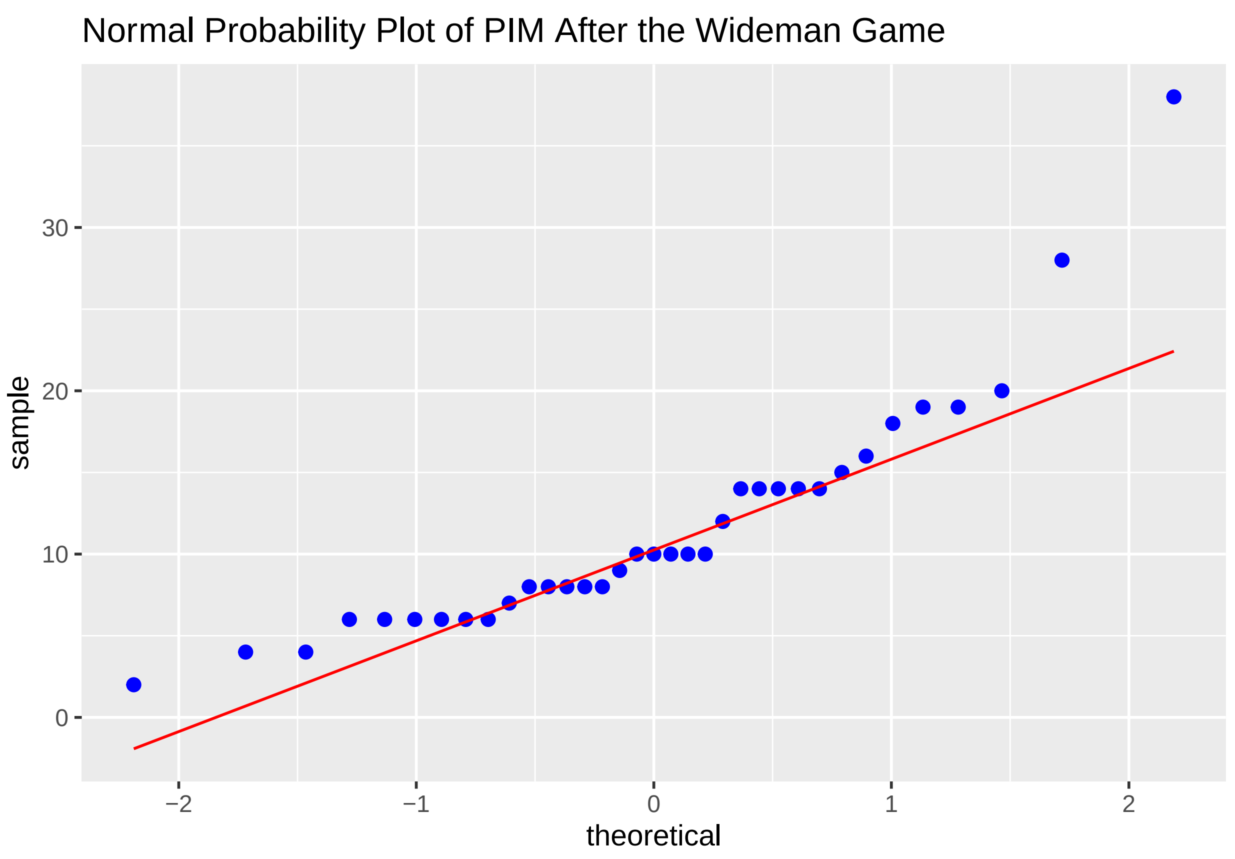
Since $0.002 < 0.05$ we reject the null hypothesis and conclude that the average penalty minutes after the Nashville game is more than the average penalty minutes before the Nashville game 2015-2016 season. Therefore, the Wideman effect existed in the 2015-2016 season for the Calgary Flames.

We can now test this same hypothesis test using a t.test. The condition of the model must be met before we use a t.test

```
ggplot(before_wideman_df, aes(sample = PIM)) + stat_qq(col = 'blue', size = 2) + stat_qqline(col = 'red
```



```
ggplot(after_wideman_df, aes(sample = PIM)) + stat_qq(col = 'blue', size = 2) + stat_qqline(col = 'red')
```



Penalty in minutes is a discrete variable and is likely not Normally distributed. However, lots of these plots lie on the red line therefore we can assume Normality for both data frames.

```
t.test(before_wideman_df$PIM, after_wideman_df$PIM, alternative = 'less')
```

```
##
##  Welch Two Sample t-test
##
## data:  before_wideman_df$PIM and after_wideman_df$PIM
## t = -2.8755, df = 63.824, p-value = 0.002738
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -1.781215
## sample estimates:
## mean of x mean of y
##  7.382979 11.628571
```

From the t.test we get a p-value of 0.002738. Therefore, we once again reject the null hypothesis and conclude that the average penalty minutes after the Nashville game is more than the average penalty minutes before the Nashville game 2015-2016 season. Therefore, the Wideman effect existed in the 2015-2016 season for the Calgary Flames.

Did the Wideman effect occur for the Calgary Flames from 2010-2019?

```
calgary_before_df = filter(pim_data, Team == 'Calgary Flames', Before_or_After == 'Before') # All Calga
calgary_after_df = filter(pim_data, Team == 'Calgary Flames', Before_or_After == 'After') # All Calgary
calgary_df = filter(pim_data, Team == 'Calgary Flames')

calgary_before_df
```

```
## # A tibble: 434 x 7
##       Game_ID Home_or_Away Date_Time   Season Team           PIM Before_or_After
##         <dbl> <chr>        <date>       <dbl> <chr>        <dbl> <chr>
##  1 2014030181 away         2015-04-16  2.01e7 Calgary Fla~     6 Before
##  2 2014030182 away         2015-04-18  2.01e7 Calgary Fla~    95 Before
##  3 2014030183 home         2015-04-20  2.01e7 Calgary Fla~    30 Before
##  4 2014030184 home         2015-04-22  2.01e7 Calgary Fla~    14 Before
##  5 2014030185 away         2015-04-24  2.01e7 Calgary Fla~     4 Before
##  6 2014030186 home         2015-04-26  2.01e7 Calgary Fla~     4 Before
##  7 2014030241 away         2015-05-01  2.01e7 Calgary Fla~    20 Before
##  8 2014030242 away         2015-05-04  2.01e7 Calgary Fla~    10 Before
##  9 2014030243 home         2015-05-06  2.01e7 Calgary Fla~     6 Before
## 10 2014030244 home         2015-05-09  2.01e7 Calgary Fla~    10 Before
## # ... with 424 more rows
```

```
calgary_after_df
```

```
## # A tibble: 290 x 7
##       Game_ID Home_or_Away Date_Time   Season Team           PIM Before_or_After
##         <dbl> <chr>        <date>       <dbl> <chr>        <dbl> <chr>
##  1 2016030171 away         2017-04-14  2.02e7 Calgary Fla~    14 After
##  2 2016030172 away         2017-04-16  2.02e7 Calgary Fla~    17 After
##  3 2016030173 home         2017-04-18  2.02e7 Calgary Fla~     2 After
##  4 2016030174 home         2017-04-20  2.02e7 Calgary Fla~     4 After
##  5 2017020586 away         2017-12-30  2.02e7 Calgary Fla~    19 After
##  6 2016021020 home         2017-03-14  2.02e7 Calgary Fla~     8 After
##  7 2017021076 home         2018-03-14  2.02e7 Calgary Fla~    11 After
##  8 2017020625 home         2018-01-05  2.02e7 Calgary Fla~     4 After
##  9 2017020506 home         2017-12-17  2.02e7 Calgary Fla~    25 After
## 10 2017020154 home         2017-10-28  2.02e7 Calgary Fla~     8 After
## # ... with 280 more rows
```

```
calgary_df
```

```
## # A tibble: 724 x 7
##       Game_ID Home_or_Away Date_Time   Season Team           PIM Before_or_After
##         <dbl> <chr>        <date>       <dbl> <chr>        <dbl> <chr>
##  1 2016030171 away         2017-04-14  2.02e7 Calgary Fla~    14 After
##  2 2016030172 away         2017-04-16  2.02e7 Calgary Fla~    17 After
##  3 2016030173 home         2017-04-18  2.02e7 Calgary Fla~     2 After
##  4 2016030174 home         2017-04-20  2.02e7 Calgary Fla~     4 After
##  5 2014030181 away         2015-04-16  2.01e7 Calgary Fla~     6 Before
##  6 2014030182 away         2015-04-18  2.01e7 Calgary Fla~    95 Before
##  7 2014030183 home         2015-04-20  2.01e7 Calgary Fla~    30 Before
##  8 2014030184 home         2015-04-22  2.01e7 Calgary Fla~    14 Before
##  9 2014030185 away         2015-04-24  2.01e7 Calgary Fla~     4 Before
## 10 2014030186 home         2015-04-26  2.01e7 Calgary Fla~     4 Before
## # ... with 714 more rows
```

Hypothesis Test: The null states that the average amount of penalty minutes for the Calgary Flames before the Nashville game is more than or equal to the average amount of penalty minutes for the Calgary Flames after the Nashville game. Therefore stating that the Wideman effect did not occur if we are looking at all seasons from 2010-2011 season to 2018-2019 season.

$$H_0 : \mu_{BeforeWideman} - \mu_{AfterWideman} \geq 0 \equiv \mu_{BeforeWideman} \geq \mu_{AfterWideman}$$

The alternative states that the average amount of penalty minutes for the Calgary Flames before the Nashville game is less than the average amount of penalty minutes for the Calgary Flames after the Nashville game. Therefore stating that the Wideman effect occur for more than just the 2015-2016 season.

$$H_A : \mu_{BeforeWideman} - \mu_{AfterWideman} < 0 \equiv \mu_{BeforeWideman} < \mu_{AfterWideman}$$

```
favstats(~PIM, data = calgary_before_df)
```

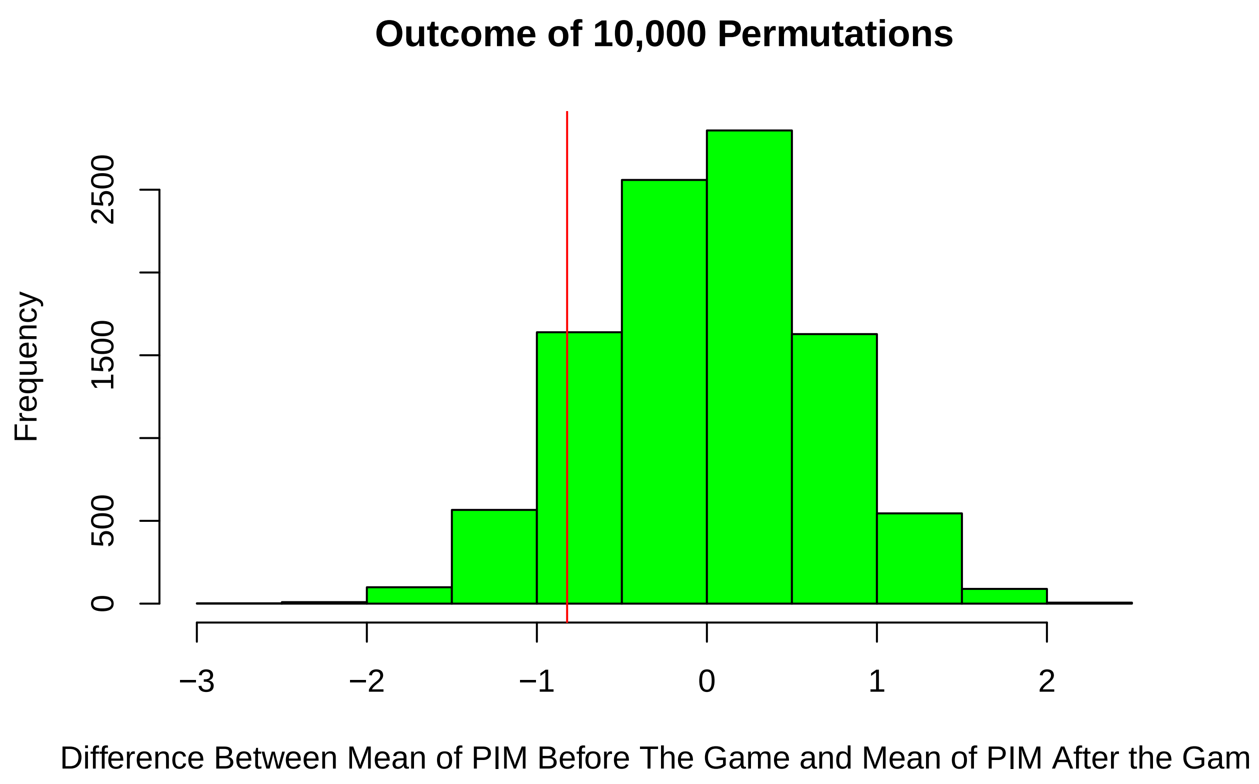
```
##  min Q1 median Q3 max     mean      sd   n missing
##    0  6      8 13 101 9.864055 8.76917 434       0
```

```
favstats(~PIM, data = calgary_after_df)
```

```
##  min Q1 median Q3 max     mean       sd   n missing
##    2  6      8 13  76 10.68621 8.708752 290       0
```

Permutation Test:

```
observed_pim_diff = mean(~PIM, data = calgary_before_df) - mean(~PIM, data = calgary_after_df) # comput

N.Permutation = 10000 #doing 10000 permutations

pim_diff = numeric(N.Permutation)
n.calgarysize = length(calgary_df$PIM)
n.calgary_before = length(calgary_before_df$PIM)

for(i in 1:N.Permutation){
  index = sample(n.calgarysize, n.calgary_before, replace = FALSE)
  pim_diff[i] = mean(calgary_df$PIM[index]) - mean(calgary_df$PIM[-index])
}

hist(pim_diff, xlab = 'Difference Between Mean of PIM Before The Game and Mean of PIM After the Game',
abline(v = observed_pim_diff, col = 'red')
```



P-value

```r
sum(pim_diff <= observed_pim_diff) / N.Permutation
```

```
## [1] 0.1109
```

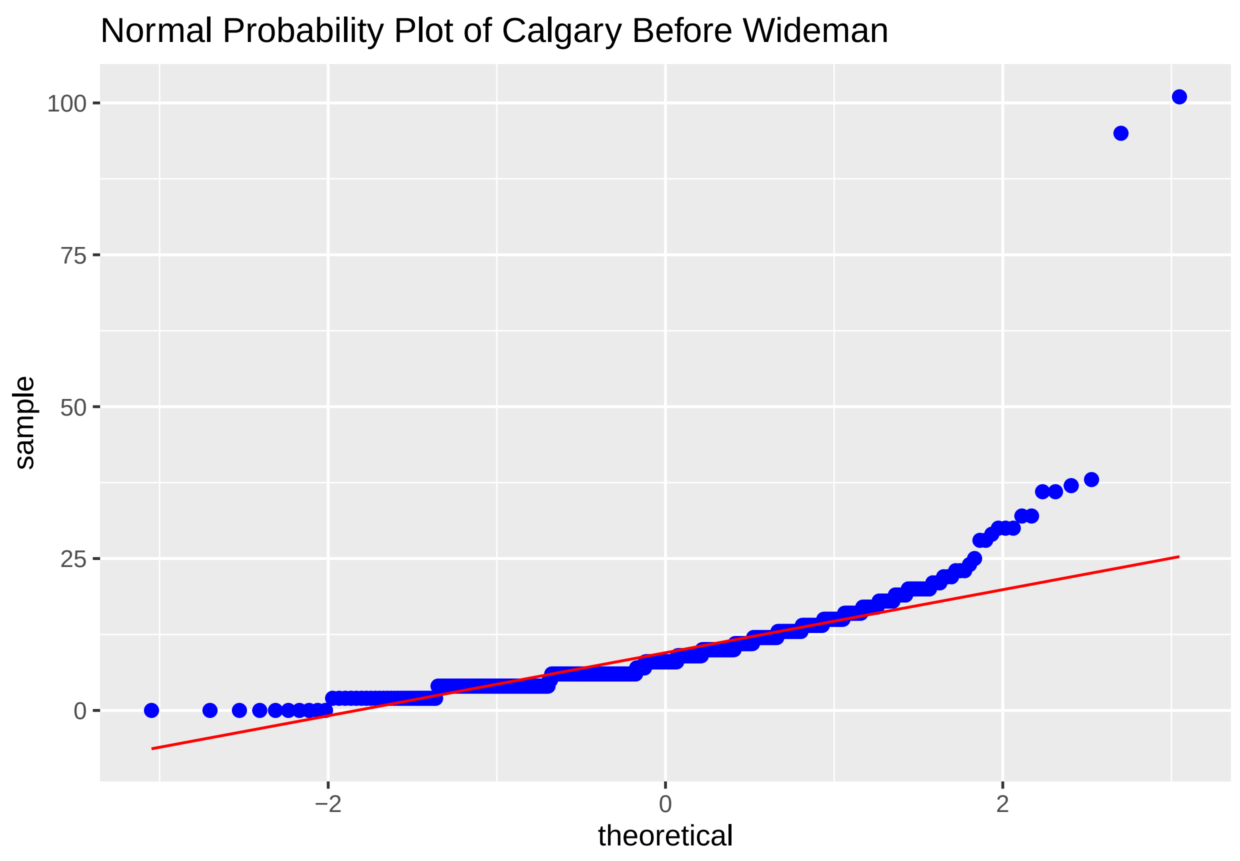
Since the p-value = 0.1106 is greater than 0.05, we fail to reject the null hypothesis and conclude that the Calgary Flames did not experience the Wideman effect if we are looking at all the seasons of the data frame.

```r
qdata(pim_diff, c(0.025, 0.975))
```

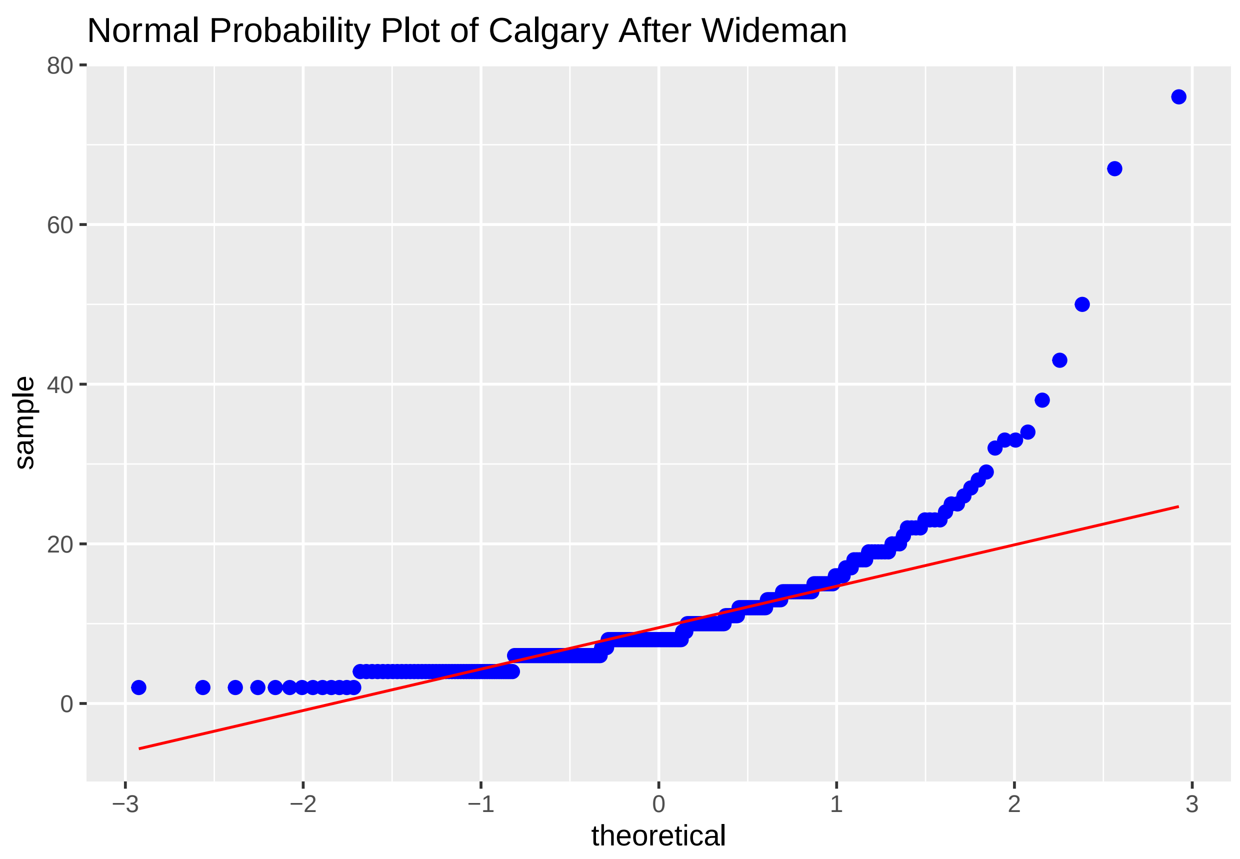
```
##      2.5%     97.5%
## -1.282345  1.260226
```

We can now do a hypothesis test with a t.test Test Normality

```r
ggplot(calgary_before_df, aes(sample = PIM)) + stat_qq(col = 'blue', size = 2) + stat_qqline(col = 'red
```



```
ggplot(calgary_after_df, aes(sample = PIM)) + stat_qq(col = 'blue', size = 2) + stat_qqline(col = 'red'
```



From this Normal Probability plots, we can assume that the penalty minutes of Calgary Flames are approximately Normally distributed.

```r
t.test(calgary_before_df$PIM, calgary_after_df$PIM, alternative = 'less')
```

```
##
##  Welch Two Sample t-test
##
## data:  calgary_before_df$PIM and calgary_after_df$PIM
## t = -1.2413, df = 622.53, p-value = 0.1075
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.2689446
## sample estimates:
## mean of x mean of y
##  9.864055 10.686207
```

Since P.value = 0.1075 is > 0.05 we fail to reject the null hypothesis and conclude that throughout all the seasons the average PIM before the wideman game is higher than the average PIM after the wideman game. Therefore, it does not exist through the 8 seasons.

Testing for all teams except Calgary in 2015-2016 season

```r
no_calgary_df = pim_data[!(pim_data$Team == 'Calgary Flames'), ] #getting all rows for all the teams ex
no_calgary_df
```

```
## # A tibble: 22,144 x 7
##      Game_ID Home_or_Away Date_Time   Season Team             PIM Before_or_After
##        <dbl> <chr>        <date>       <dbl> <chr>          <dbl> <chr>
##  1    2.01e9 away         2012-04-29  2.01e7 New Jersey D~     12 Before
##  2    2.01e9 home         2012-04-29  2.01e7 Philadelphia~      6 Before
##  3    2.01e9 away         2012-05-01  2.01e7 New Jersey D~     12 Before
##  4    2.01e9 home         2012-05-01  2.01e7 Philadelphia~     32 Before
##  5    2.01e9 away         2012-05-03  2.01e7 Philadelphia~      4 Before
##  6    2.01e9 home         2012-05-03  2.01e7 New Jersey D~     10 Before
##  7    2.01e9 away         2012-05-06  2.01e7 Philadelphia~     10 Before
##  8    2.01e9 home         2012-05-06  2.01e7 New Jersey D~      4 Before
##  9    2.01e9 away         2012-05-08  2.01e7 New Jersey D~      2 Before
## 10    2.01e9 home         2012-05-08  2.01e7 Philadelphia~      8 Before
## # ... with 22,134 more rows
```

```
no_calgary_before = filter(no_calgary_df, Before_or_After == 'Before')
no_calgary_after = filter(no_calgary_df, Before_or_After == 'After')

no_calgary_before
```

```
## # A tibble: 13,186 x 7
##      Game_ID Home_or_Away Date_Time   Season Team             PIM Before_or_After
##        <dbl> <chr>        <date>       <dbl> <chr>          <dbl> <chr>
##  1    2.01e9 away         2012-04-29  2.01e7 New Jersey D~     12 Before
##  2    2.01e9 home         2012-04-29  2.01e7 Philadelphia~      6 Before
##  3    2.01e9 away         2012-05-01  2.01e7 New Jersey D~     12 Before
##  4    2.01e9 home         2012-05-01  2.01e7 Philadelphia~     32 Before
##  5    2.01e9 away         2012-05-03  2.01e7 Philadelphia~      4 Before
##  6    2.01e9 home         2012-05-03  2.01e7 New Jersey D~     10 Before
##  7    2.01e9 away         2012-05-06  2.01e7 Philadelphia~     10 Before
##  8    2.01e9 home         2012-05-06  2.01e7 New Jersey D~      4 Before
##  9    2.01e9 away         2012-05-08  2.01e7 New Jersey D~      2 Before
## 10    2.01e9 home         2012-05-08  2.01e7 Philadelphia~      8 Before
## # ... with 13,176 more rows
```

```
no_calgary_after
```

```
## # A tibble: 8,958 x 7
##      Game_ID Home_or_Away Date_Time   Season Team             PIM Before_or_After
##        <dbl> <chr>        <date>       <dbl> <chr>          <dbl> <chr>
##  1    2.02e9 home         2017-04-14  2.02e7 Anaheim Ducks     10 After
##  2    2.02e9 home         2017-04-16  2.02e7 Anaheim Ducks     15 After
##  3    2.02e9 away         2017-04-18  2.02e7 Anaheim Ducks     12 After
##  4    2.02e9 away         2017-04-20  2.02e7 Anaheim Ducks      6 After
##  5    2.02e9 away         2016-04-14  2.02e7 NY Rangers        10 After
##  6    2.02e9 home         2016-04-14  2.02e7 Pittsburgh P~     10 After
##  7    2.02e9 away         2016-04-16  2.02e7 NY Rangers        19 After
##  8    2.02e9 home         2016-04-16  2.02e7 Pittsburgh P~     15 After
##  9    2.02e9 away         2016-04-19  2.02e7 Pittsburgh P~     12 After
## 10    2.02e9 home         2016-04-19  2.02e7 NY Rangers        10 After
## # ... with 8,948 more rows
```

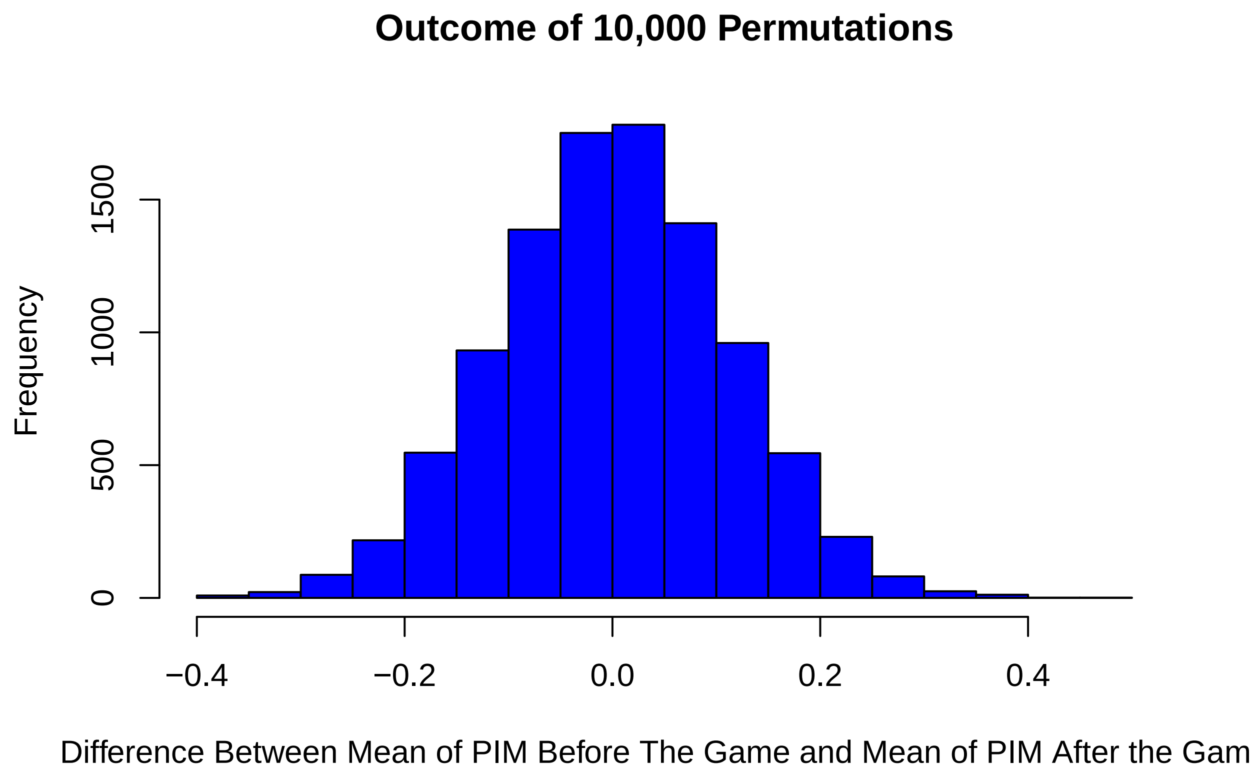
```
favstats(~PIM, data = no_calgary_before)
```

```
##  min Q1 median Q3 max     mean       sd     n missing
##    0  6      9 13 183 11.01987 8.732405 13186       0
```

```r
favstats(~PIM, data = no_calgary_after)
```

```
##  min Q1 median Q3 max     mean       sd    n missing
##    0  4      8 11  96 8.931904 6.630088 8958       0
```

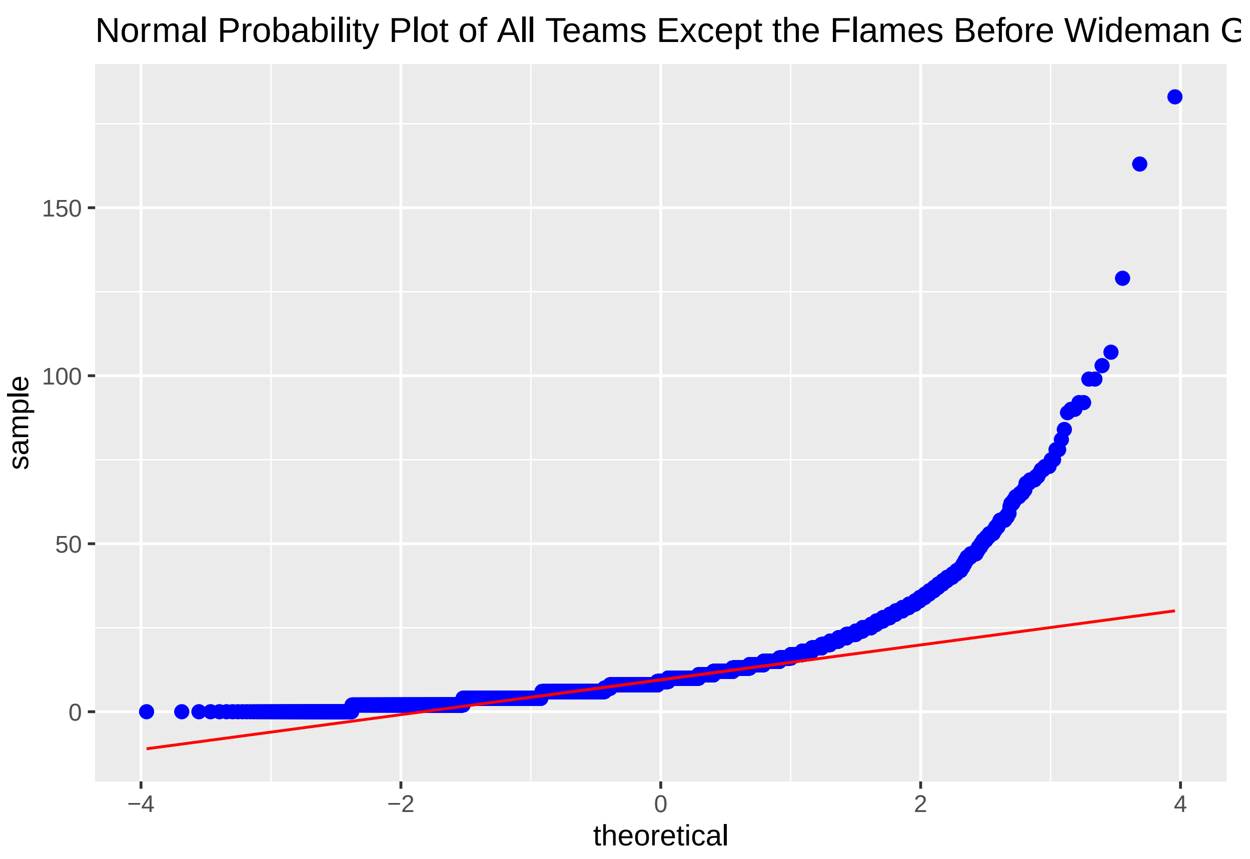
```r
observed_pim_diff = mean(~PIM, data = no_calgary_before) - mean(~PIM, data = no_calgary_after) # comput

N.Permutation = 10000
pim_diff = numeric(N.Permutation)

n.pimsize = length(no_calgary_df$PIM)
n.before_wideman = length(no_calgary_before$PIM)

for(i in 1:N.Permutation){
  index = sample(n.pimsize, n.before_wideman, replace = FALSE)
  pim_diff[i] = mean(no_calgary_df$PIM[index]) - mean(no_calgary_df$PIM[-index])
}

hist(pim_diff, xlab = 'Difference Between Mean of PIM Before The Game and Mean of PIM After the Game', 
abline(v = observed_pim_diff, col = 'red')
```



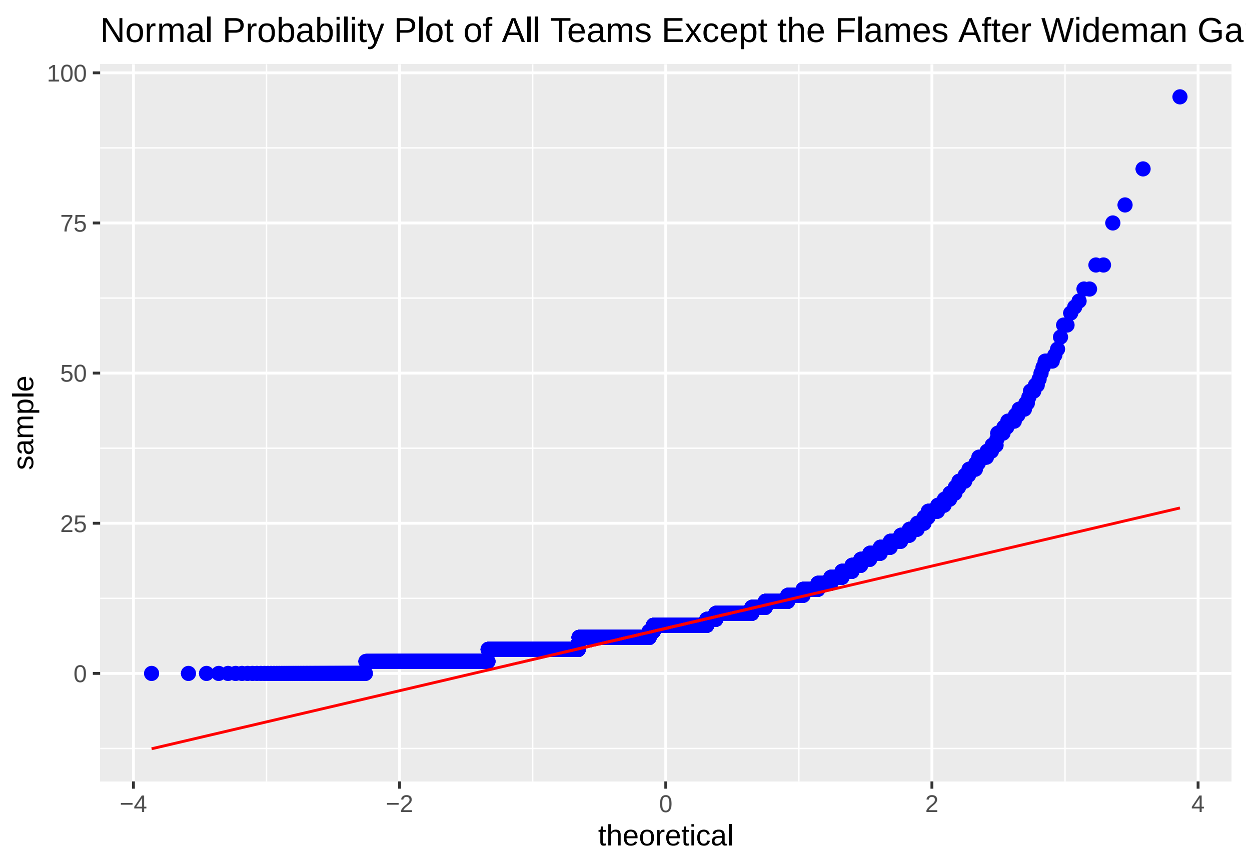
```r
sum(pim_diff <= observed_pim_diff) / N.Permutation
```

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

```r
ggplot(no_calgary_before, aes(sample = PIM)) + stat_qq(col = 'blue', size = 2) + stat_qqline(col = 'red
```



```r
ggplot(no_calgary_after, aes(sample = PIM)) + stat_qq(col = 'blue', size = 2) + stat_qqline(col = 'red')
```



```
t.test(no_calgary_before$PIM, no_calgary_after$PIM, alternative = 'less')
```

```
##
##  Welch Two Sample t-test
##
## data:  no_calgary_before$PIM and no_calgary_after$PIM
## t = 20.194, df = 21872, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 2.258039
## sample estimates:
## mean of x mean of y
## 11.019870  8.931904
```

Is there a relationship between a teams win percentage and goal differential?

```
# Getting the Goal differential of each game
game_data$Goal_Diff = ifelse((game_data$Home_Or_Away == 'home'), game_data$Home_Goals - game_data$Away_(

# Getting the Goal Differential and Win Percentage for the season 2010 - 2011
df_2010_2011 = filter(game_data, Season == '20102011')
season_2010_2011 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2010_2011, FUN = sum), aggregate(Wor
season_2010_2011$Win_Perc = season_2010_2011$Won / season_2010_2011$Won.1 #calculating win percentage: (
season_2010_2011 = season_2010_2011 %>%
  select(Team, Goal_Diff, Win_Perc)

# Getting the Goal Differential and Win Percentage for the season 2011 - 2012
df_2011_2012 = filter(game_data, Season == '20112012')
season_2011_2012 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2011_2012, FUN = sum), aggregate(Wo
season_2011_2012$Win_Perc = season_2011_2012$Won / season_2011_2012$Won.1
season_2011_2012 = season_2011_2012 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2011_2012

# Getting the Goal Differential and Win Percentage for the season 2012 - 2013
df_2012_2013 = filter(game_data, Season == '20122013')
season_2012_2013 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2012_2013, FUN = sum), aggregate(Wo
season_2012_2013$Win_Perc = season_2012_2013$Won / season_2012_2013$Won.1
season_2012_2013 = season_2012_2013 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2012_2013

# Getting the Goal Differential and Win Percentage for the season 2013 - 2014
df_2013_2014 = filter(game_data, Season == '20132014')
season_2013_2014 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2013_2014, FUN = sum), aggregate(Wo
season_2013_2014$Win_Perc = season_2013_2014$Won / season_2013_2014$Won.1
season_2013_2014 = season_2013_2014 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2013_2014

# Getting the Goal Differential and Win Percentage for the season 2014 - 2015
df_2014_2015 = filter(game_data, Season == '20142015')
season_2014_2015 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2014_2015, FUN = sum), aggregate(Wo
season_2014_2015$Win_Perc = season_2014_2015$Won / season_2014_2015$Won.1
season_2014_2015 = season_2014_2015 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2014_2015

# Getting the Goal Differential and Win Percentage for the season 2015 - 2016
df_2015_2016 = filter(game_data, Season == '20152016')
season_2015_2016 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2015_2016, FUN = sum), aggregate(Wo
season_2015_2016$Win_Perc = season_2015_2016$Won / season_2015_2016$Won.1
season_2015_2016 = season_2015_2016 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2015_2016

# Getting the Goal Differential and Win Percentage for the season 2016 - 2017
df_2016_2017 = filter(game_data, Season == '20162017')
season_2016_2017 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2016_2017, FUN = sum), aggregate(Wo
season_2016_2017$Win_Perc = season_2016_2017$Won / season_2016_2017$Won.1
season_2016_2017 = season_2016_2017 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2016_2017

# Getting the Goal Differential and Win Percentage for the season 2017 - 2018
df_2017_2018 = filter(game_data, Season == '20172018')
season_2017_2018 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2017_2018, FUN = sum), aggregate(Wo
```

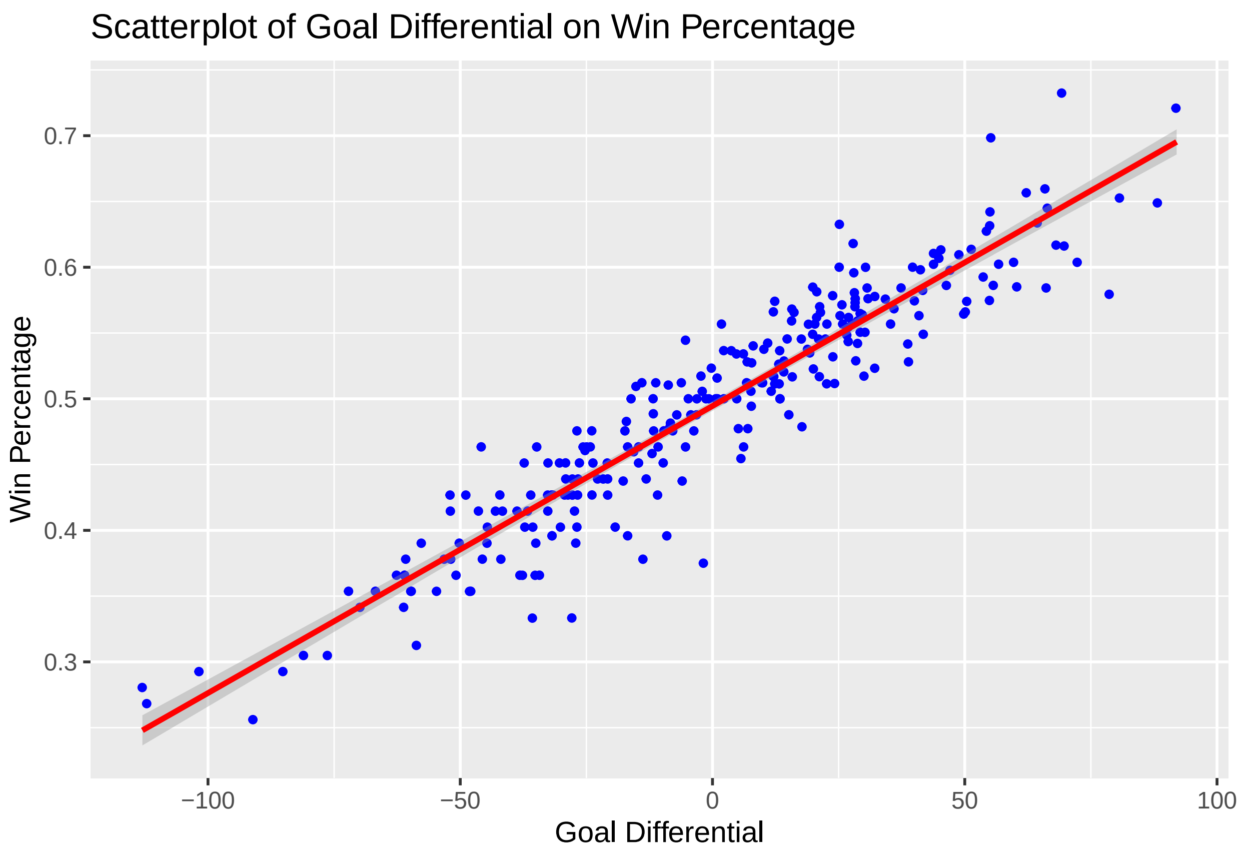
```r
season_2017_2018$Win_Perc = season_2017_2018$Won / season_2017_2018$Won.1
season_2017_2018 = season_2017_2018 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2017_2018

# Getting the Goal Differential and Win Percentage for the season 2018 - 2019
df_2018_2019 = filter(game_data, Season == '20182019')
season_2018_2019 = data.frame(aggregate(Goal_Diff ~ Team, data = df_2018_2019, FUN = sum), aggregate(Wo
season_2018_2019$Win_Perc = season_2018_2019$Won / season_2018_2019$Won.1
season_2018_2019 = season_2018_2019 %>%
  select(Team, Goal_Diff, Win_Perc)
#season_2018_2019

# Combining all the seasons dataframes into one
goaldiff_winperc_df = rbind(season_2010_2011, season_2011_2012, season_2012_2013, season_2013_2014, seas
```

Creating a linear model:

$$\text{Win Percentage} = A + B * \text{Goal Differential} + e_i$$

```r
ggplot(goaldiff_winperc_df, aes(x = Goal_Diff, y = Win_Perc)) + geom_point(size = 1,  col = 'blue', pos
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Getting the coefficients of the model:

```r
predict.winperc = lm(Win_Perc~Goal_Diff, data=goaldiff_winperc_df)
```

```
options(scipen = 999)
predict.winperc$coefficients
```

```
## (Intercept)   Goal_Diff
##   0.4945231   0.0021820
```

$$\text{Win Percentage} = 0.4945231 + 0.00221820 * \text{Goal Differential}$$

Getting the Correlation Coefficient:

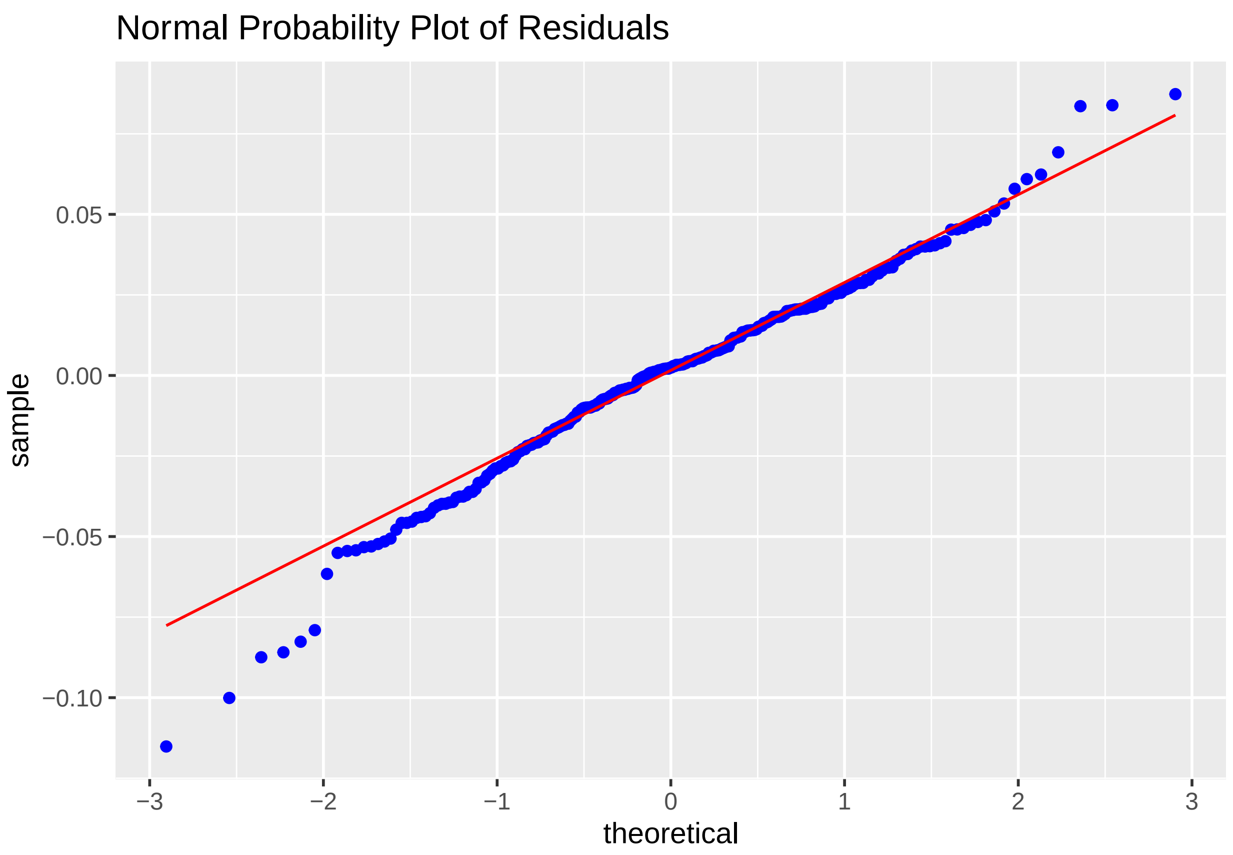
```
cor(~Win_Perc, ~Goal_Diff, data=goaldiff_winperc_df)
```

```
## [1] 0.9385296
```

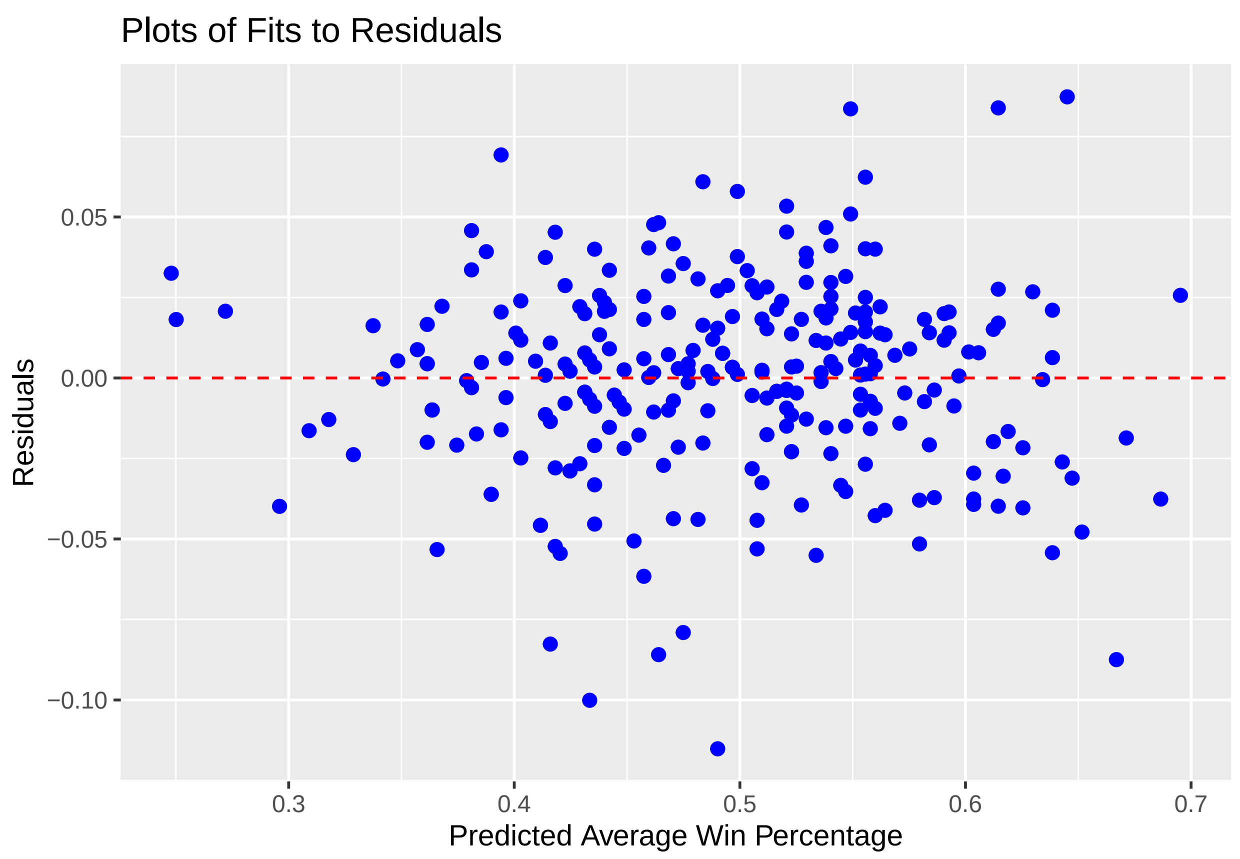
The correlation coefficient is 0.9385

Checking conditions:

```
predicts.winperc = predict.winperc$fitted.values
e.winperc = predict.winperc$residuals
diagnostic.winperc = data.frame(predicts.winperc, e.winperc)

ggplot(diagnostic.winperc, aes(sample = e.winperc)) + stat_qq(col = 'blue') + stat_qqline(col = 'red') +
```



```
ggplot(diagnostic.winperc, aes(x = predicts.winperc, y = e.winperc)) + geom_point(size = 2, col = 'blue
```



Therefore the residuals are Normal and it is homoscedastic.

$$H_0 : \text{A teams win percentage cannot be expressed as a linear function of a teams goal differential}$$

$$H_A : \text{A teams win percentage can be expressed as a linear function of a teams goal differential}$$

```r
summary(aov(predict.winperc))
```

```
##                Df Sum Sq Mean Sq F value              Pr(>F)
## Goal_Diff       1  1.833  1.8332    1996 <0.0000000000000002 ***
## Residuals     270  0.248  0.0009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test statistic is $F_{Obs} = 1996$ and the P-value $= 0.0000000000000002$. The small P-value would indicate that we reject the null hypothesis. Therefore, we can conclude from this data that a teams win percentage can be expressed as a linear function of a teams goal differential

```r
confint(predict.winperc)
```

```
##                  2.5 %      97.5 %
## (Intercept) 0.49090524 0.498141045
## Goal_Diff   0.00208584 0.002278159
```

$$0.4909 \leq A \leq 0.49810.002085 \leq B \leq 0.002278$$

With respect to A: Consider the teams where their goal differential is 0, the mean / average win percentage is somewhere between 0.4909% and 0.4981%. With respect to B: As a teams goal differential inceases by 1, a teams win percentage will increase by an average of somewhere between 0.002085% and 0.002278%.

Consider the Calgary Flames has a goal differential of 42. With 95% confidence, the average win percentage of all teams that have a 42 goal differential is:

```
predict(predict.winperc, newdata = data.frame(Goal_Diff = 42), interval = 'conf', conf.level = 0.95)
```

```
##         fit       lwr       upr
## 1 0.5861671 0.5807449 0.5915893
```

Consider the Calgary Flames has a goal differential of 42. With 95% confidence, the average win percentage of Calgary Flames with a 42 goal differential is:

```
predict(predict.winperc, newdata = data.frame(Goal_Diff = 42), interval = 'predict', conf.level = 0.95)
```

```
##         fit       lwr      upr
## 1 0.5861671 0.5262533 0.646081
```

Using the estimate of the model, we can predict that for certain teams with 30, 40 and 50 goal differential, the average win percentage for all teams that have a 30, 40 and 50 goal differential is 0.56, 0.58 and 0.60.

```
win.perc = data.frame(Goal_Diff = c(30, 40, 50))
predict(predict.winperc, newdata = win.perc)
```

```
##         1         2         3
## 0.5599831 0.5818031 0.6036231
```

Bootstrapping

```
N.Bootstrap = 3000
r_boot = numeric(N.Bootstrap)
a_boot = numeric(N.Bootstrap)
b_boot = numeric(N.Bootstrap)
r_sqrd = numeric(N.Bootstrap)
ymean_boot = numeric(N.Bootstrap)

nsize = dim(goaldiff_winperc_df)[1]
xvalue = 42

for(i in 1:N.Bootstrap){
  index = sample(nsize, replace = TRUE)
  winperc_boot = goaldiff_winperc_df[index, ]

  r_boot[i] = cor(~Win_Perc, ~Goal_Diff, data = winperc_boot)
  temp_model = lm(Win_Perc~Goal_Diff, data = winperc_boot)

  a_boot[i] = coef(temp_model)[1]
  b_boot[i] = coef(temp_model)[2]
  ymean_boot[i] = a_boot[i] + (b_boot[i] * xvalue)
  r_sqrd[i] = summary(temp_model)$r.squared
}

winperc_bootstrap_result = data.frame(r_boot, a_boot, b_boot, ymean_boot, r_sqrd)
head(winperc_bootstrap_result)
```

```
##      r_boot    a_boot      b_boot ymean_boot    r_sqrd
## 1 0.9445903 0.4929356 0.002233885  0.5867588 0.8922508
## 2 0.9378549 0.4930052 0.002232289  0.5867614 0.8795719
## 3 0.9308415 0.4942504 0.002206971  0.5869432 0.8664660
## 4 0.9429874 0.4910176 0.002130210  0.5804864 0.8892253
## 5 0.9318879 0.4962554 0.002255215  0.5909744 0.8684151
## 6 0.9333136 0.4956246 0.002108441  0.5841791 0.8710744
```

```
favstats(~a_boot, data = winperc_bootstrap_result)
```

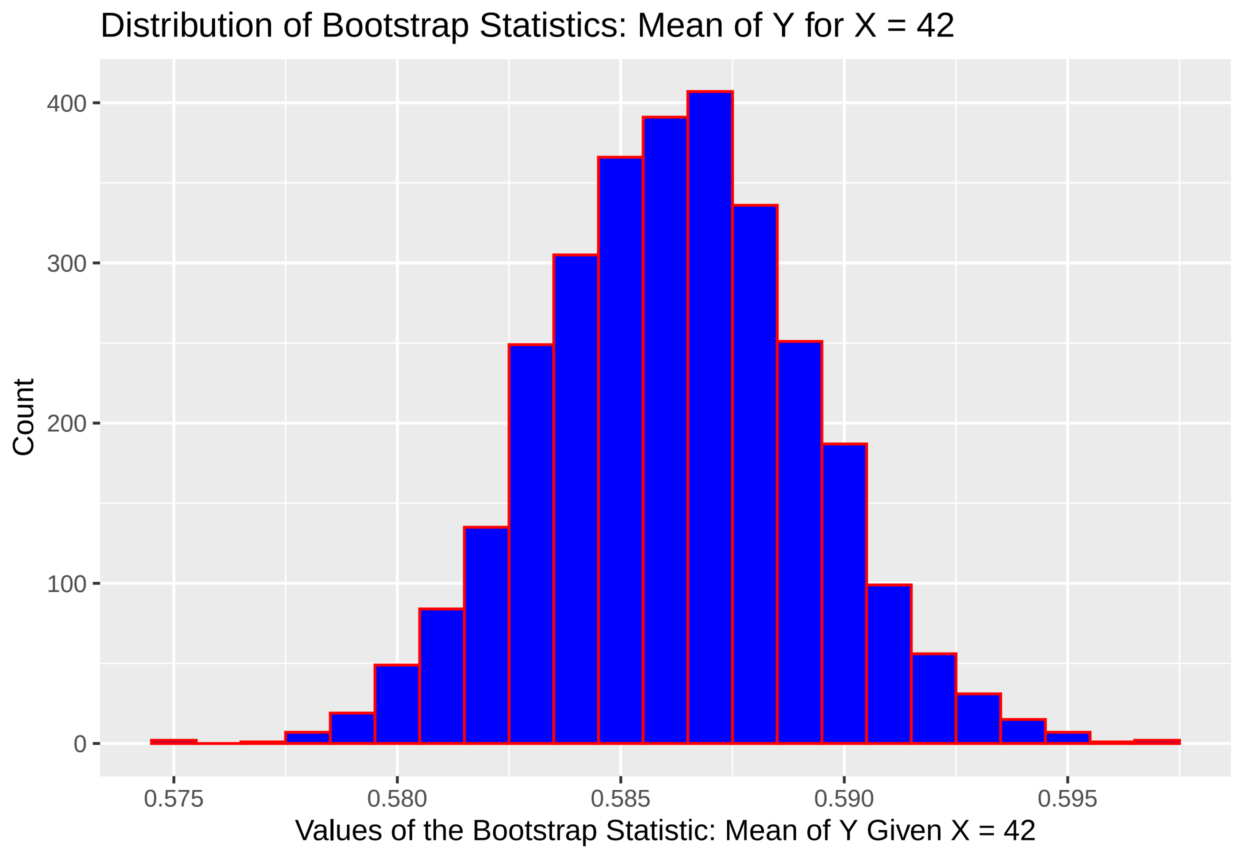
```
##        min        Q1    median        Q3       max      mean          sd    n
##  0.4882151 0.4933139 0.4945114 0.4957816 0.5011289 0.4945129 0.001849807 3000
##  missing
##        0
```

```
favstats(~b_boot, data = winperc_bootstrap_result)
```

```
##         min          Q1      median          Q3         max        mean
##  0.00200735 0.002148408 0.002184703 0.002217236 0.002359164 0.002183504
##            sd    n missing
##  0.00005028556 3000       0
```

$$\overline{a} = 0.4959\overline{b} = 0.002215$$

```
ggplot(winperc_bootstrap_result, aes(x = ymean_boot)) + geom_histogram(col = 'red', fill = 'blue', binwi
```



Using a 42 goal differential again, the 95% confidence interval we get is:

```r
qdata(~ymean_boot, c(0.025, 0.975), data = winperc_bootstrap_result)
```

```
##      2.5%     97.5%
## 0.5804644 0.5921292
```

$$0.555 \leq \mu_{y|x=42} \leq 0.565$$