DEPARTMENT OF DATA SCIENCE

DATA 5300 - APPLIED STAT INFER & EXP DES

United Airlines: NYC Flight Delays

Author:
Akanksha Sharma

Date : 29 October 2022

Table of Contents

List of Figures

List of Tables

1 Introduction

The analysis will attempt to identify the core cause of departure delays for United Airlines aircraft departing from New York City in 2013. The goal is to decrease flight delay time by determining what variables are causing the delays.This will enable United Airlines customers to improve those aspects, hence increasing customer happiness and flying effectively. The study focuses on environmental characteristics such as the time of day, year, temperature, wind speed, precipitation and visibility.

2 About Dataset

Our analysis will be carried out by leveraging the nycflights13 dataset. This dataset contains the departure timings for all flights departing from New York City's three airports - La Guardia (LGA), John F. Kennedy (JFK), and Newark Liberty International Airport (EWR) in 2013. For the scope of this project, we will be focusing on the data related to United Airlines.

3 Analysis of UA flight delays

Let's look at United Airlines' departure delays.

United Airlines has an average delay of 12.09 minutes. However, we must determine whether any outliers in the dataset affect the typical flight delay. We can tell through close examination that the outlier with the 483-minute or 8.04 hours delay is affecting the average.

We should consider some other central tendency metric to understand the delays. The median of the UA airline delay is 0 minutes. Which is a positive sign for the United Airlines.

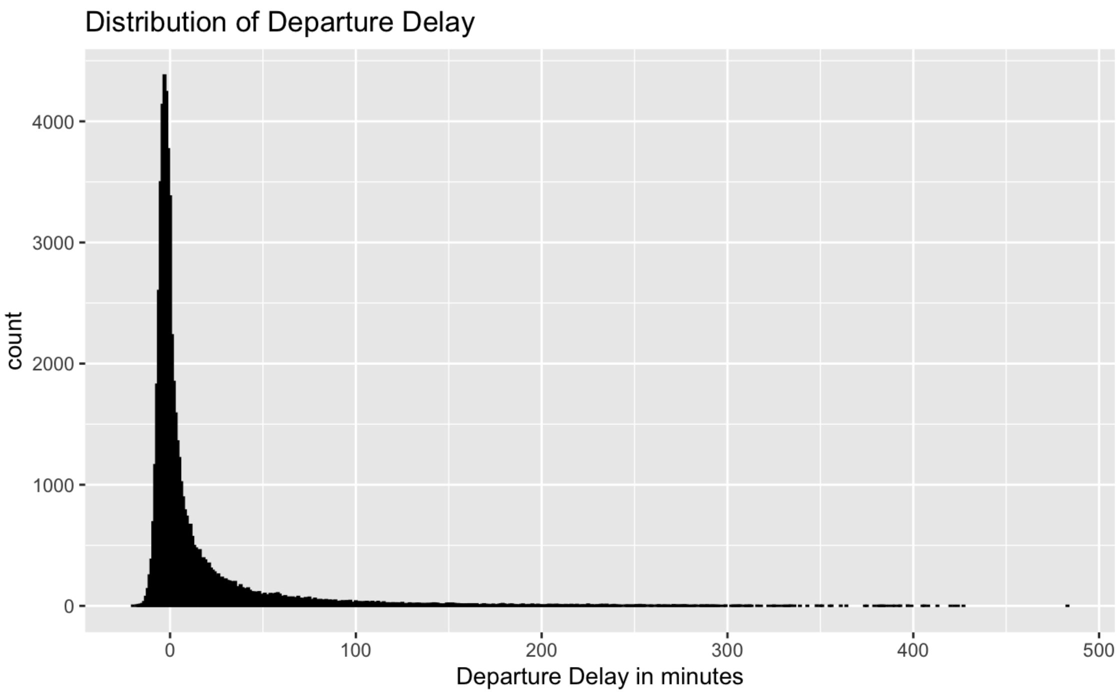


Figure 1: Histogram of departure delay

As seen in Figure 1, the departure delay data is extremely right skewed, with most flights departing with no or very small delays. However, some flights have extremely high departure delay values.

Because of this striking discrepancy, I have decided to create two variable.

Derived Variable	Condition
Very Late	If delay $\geq 30 \rightarrow True$ else False
Late	If Delay $>0 \rightarrow True$ else False

Table 1: Late and Very Late : Derived Columns

Observations: The percentage of missing values for departure delay is 1.15 %. For the remainder of this project's study, we shall impute a mean departure delay for the missing values.

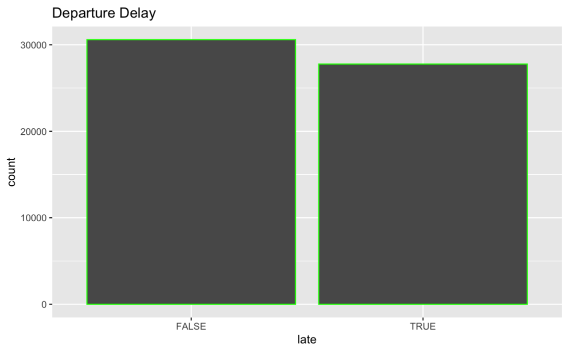


Figure 2: Late flight count

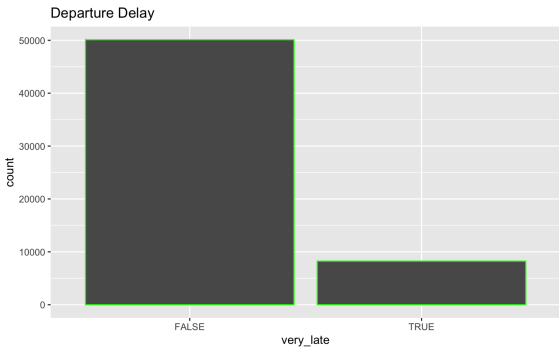


Figure 3: Very late flight count

Flights are delayed 47.57 % of the time as you can also see in Figure 2, and 14.12 % of the time they are very late (Figure 3). It is critical to determine the cause of flight delays in order to ensure customer satisfaction and increase efficiency.

NOTE : We will be using permutation tests to determine whether results are significant or not. Permutation test

4 Time of the hour

Let's try to respond to general inquiries concerning delays based on the hour.

1. What time of day is busiest for UA carrier flights?
According to Figure 4, the busiest time for United Airlines is around 6 AM.

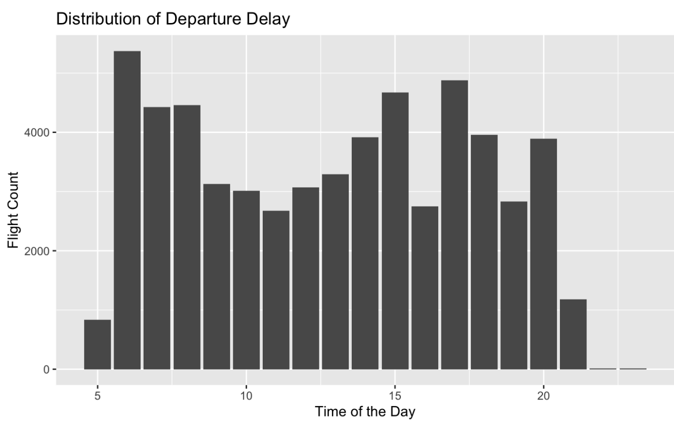


Figure 4: Departure delay based on hour

We can also see how many planes operate each hour, and there are no flights operating from midnight to 5 a.m.

Let's examine whether there's any correlation between the flying time and the departure delays. The mean, standard deviation, and median of flight delay time increase as the day progresses. We can also see that there's a sudden spike in the mean of the flight delay after 8 p.m.

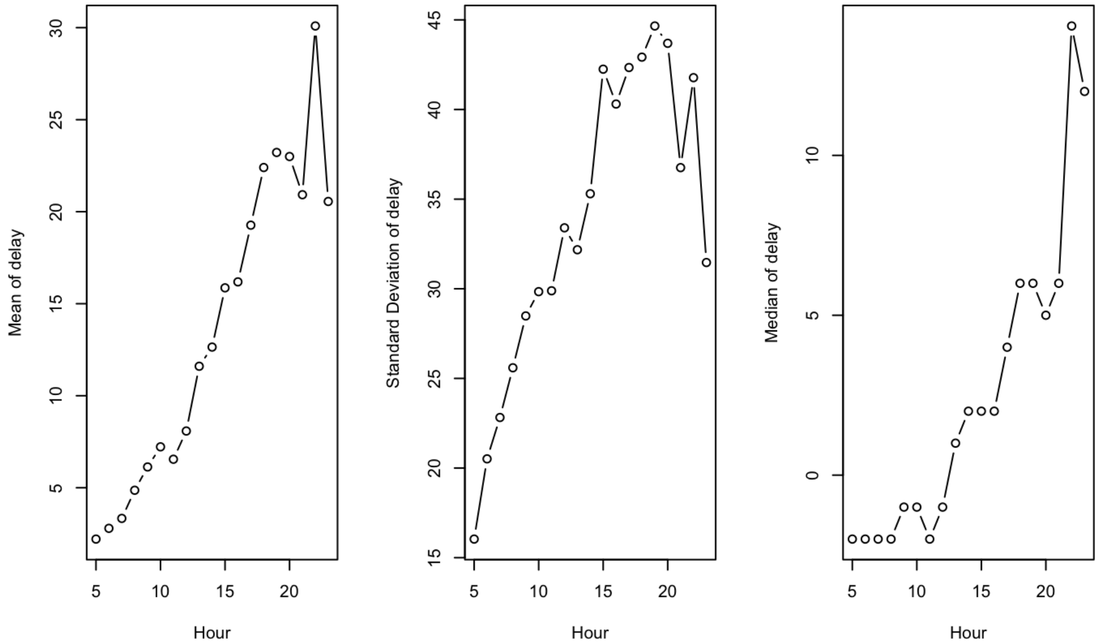


Figure 5: Progression of Mean, Standard Deviation and Median for each hour

However, we will pay close attention to the 23-hour data points. For that hour, there are just 9 data points. As a result, we can't make many conclusions. The mean delay for the 23 hour is on the higher end of the spectrum because there are two flights with departure delay of 66 and 80 minutes.
These might be the same flights that were delayed earlier in the day, eventually cause the flights to be delayed.

Let's do the analysis for JFK airport based on Departure delay. Observations:
There's a fluctuation in the mean delays for each hour. We can see that most of the flights were on time at 7PM.
JFK airport is busiest during Noon time. United Airlines can see what's causing the delay in JFK airport and try to reduce the waiting time.

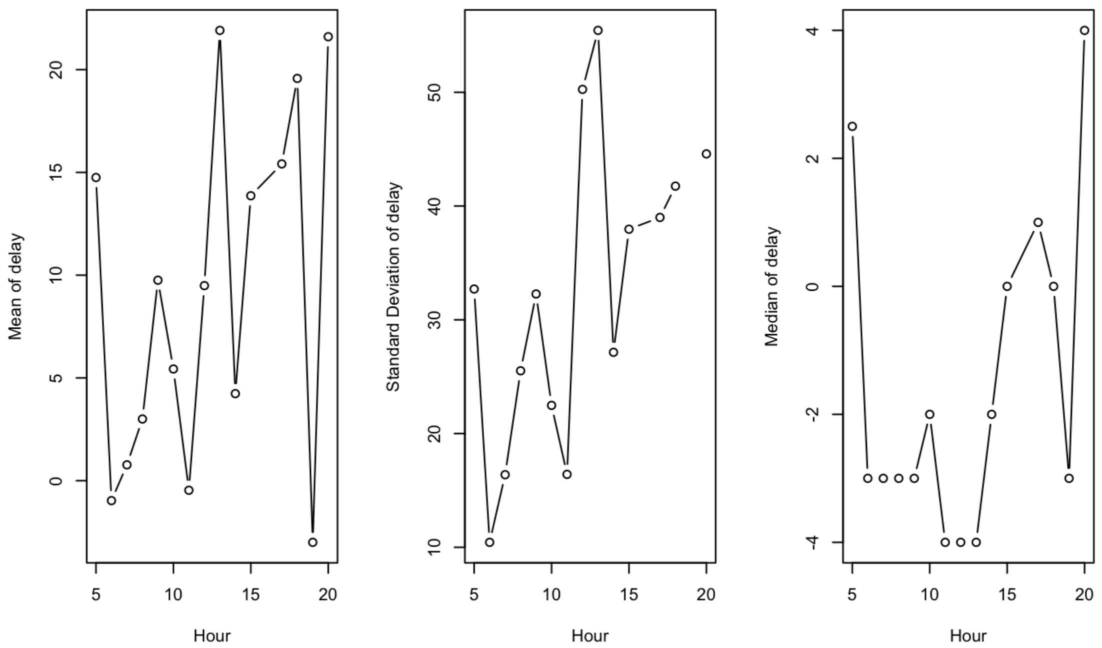


Figure 6: JFK : Progression of Mean, Standard Deviation and Median for each hour

Let's do the analysis for EWR airport based on Departure delay. Observations:
The average time delay is increasing as the day progress.
We can see sudden spike at 11 PM because these are the outliers in our dataset.

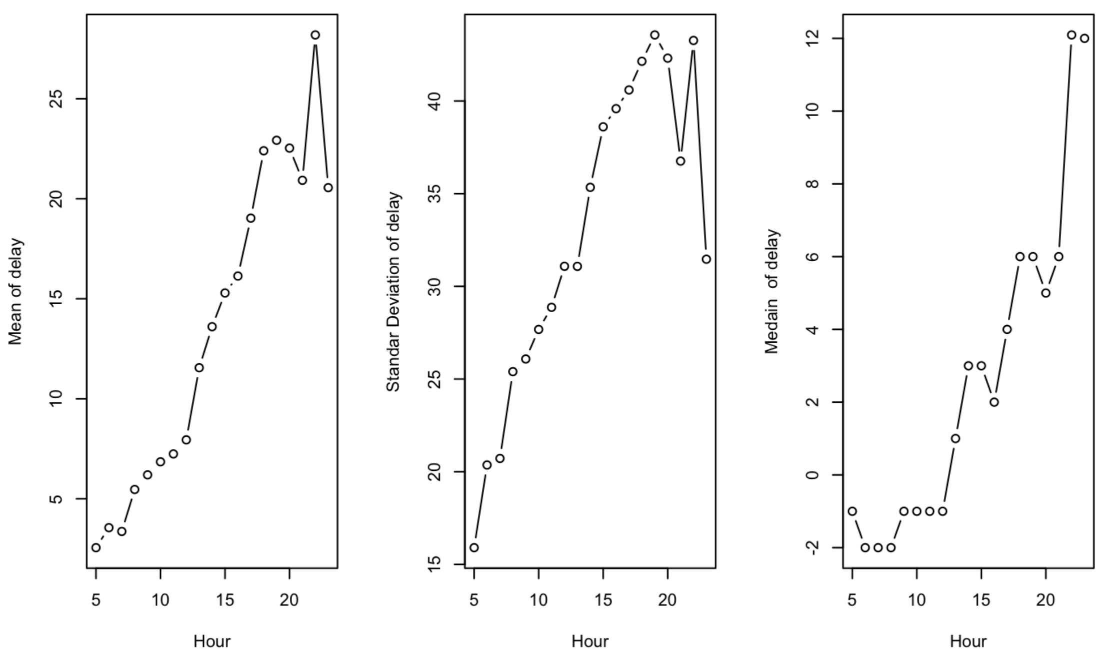


Figure 7: EWR: Progression of Mean, Standard Deviation and Median for each hour

Let's do the analysis for LGA airport based on Departure delay. Observations:
The average time delay is increasing as the day progress.

There's a fluctuation in the standard deviation for the LGA airport.
If we carefully observe the median it's more static.

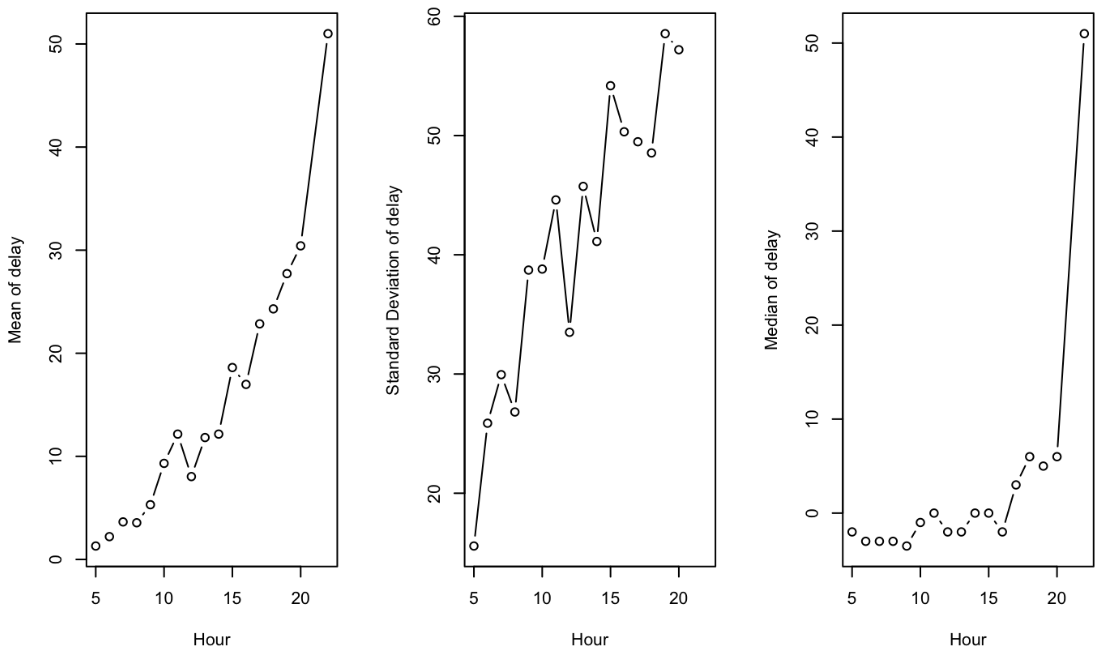


Figure 8: LGA : Progression of Mean, Standard Deviation and Median for each hour

4.1 Based on day

Let's categorize the day into 4 parts based on the following condition:

Derived Day Variable	Condition
morning	If hour <11
afternoon	If hour $\geq$ 11&<16
evening	If hour $\geq$ 16&<20
night	If hour >20

Table 2: Time of the Day : Derived Column

After creating the new variable Day in dataset, let's analyse the delay based on the time of the day. According to Figure 9, we can see that there are some flights which are having delay greater than 60 minutes.

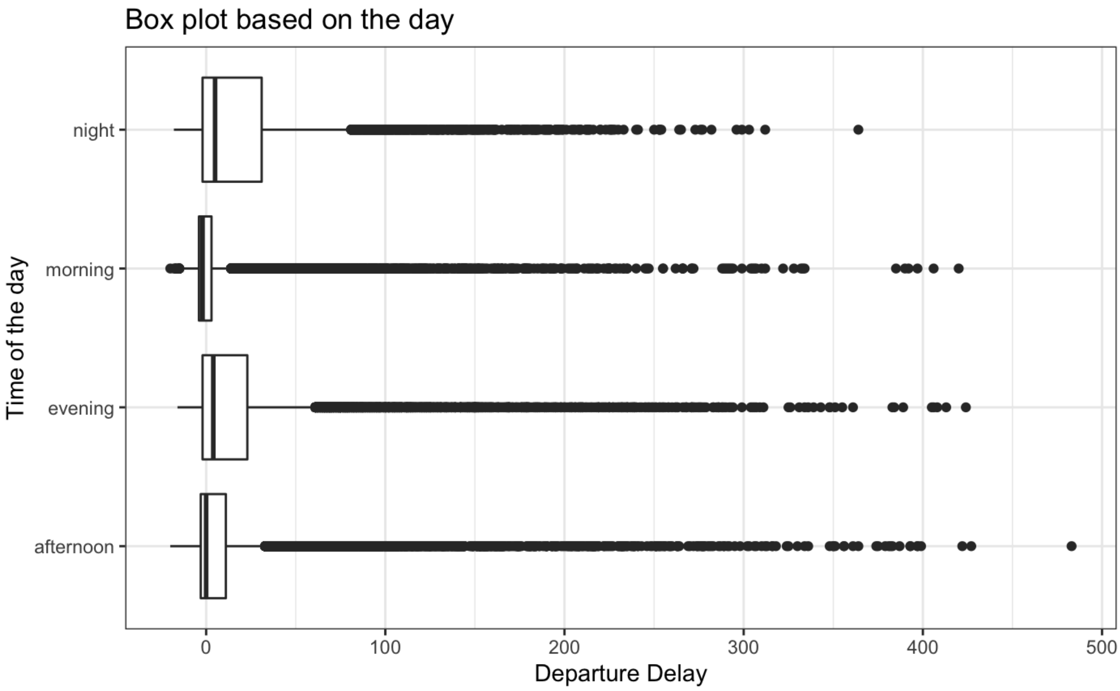
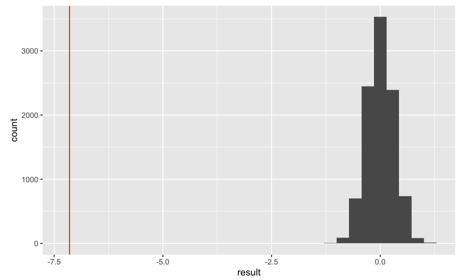


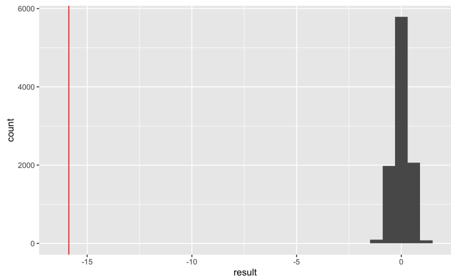
Figure 9: Boxplot of Mean, Standard Deviation and Median for each hour

The median departure delay appears to be nearly the same for morning and afternoon flights (both with a tendency for early departures — a median below zero).
As the day progresses into the evening and night, we can observe that the evening has a median slightly above zero (usually just a little late), while the night has an even higher median.

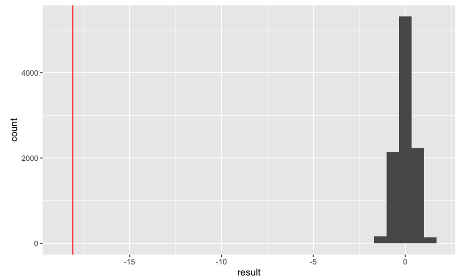
We ran permutation tests to check if there was a significant difference between the average delays at certain times of day across six alternative pairs of departure delays(Figure 10). We compared the mean departure delays across each unique pair based on the time of day.In all cases, the observed value (red line) was far from the permutation distribution, implying that a mean difference could not have happened if there was no genuine difference. All of our p values are small (0.0002), but one permutation scenario involving Evening vs Night differentiates from the others with a somewhat larger p value = 0.0014. We may infer that there is a significant difference between the means of departure delays and times of day because the p values are all less than 0.005. Overall, the data suggest that departure time is affected by the time of day.



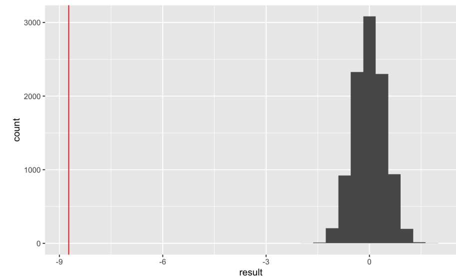
(a) Morning VS Afternoon p-value 0.0002



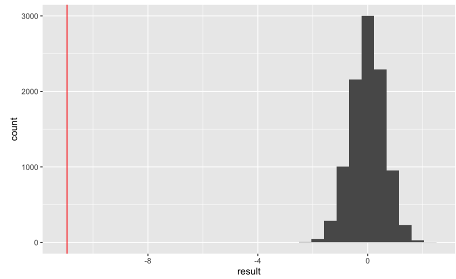
(b) Morning VS Evening p-value 0.0002



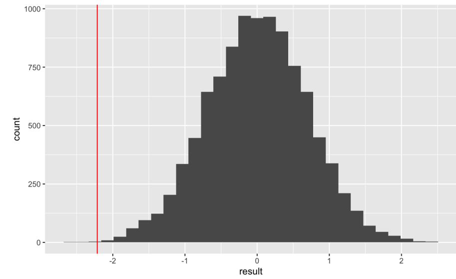
(c) Morning VS Night p-value 0.0002



(d) Afternoon VS Evening p-value 0.0002



(e) Afternoon VS Night p-value 0.0002



(f) Evening VS Night p-value 0.0014

Figure 10: Time of Day: Histograms of Permutation Tests,Difference in Mean Departure Delays

5 Time of year

Next, we'll see if departure delays vary by season. For the sake of this analysis, we divided months into following seasons:

Season	Month
Fall	September - November
Winter	December - February
Spring	March - May
Summer	June - August

Table 3: Season of the year : Derived columns

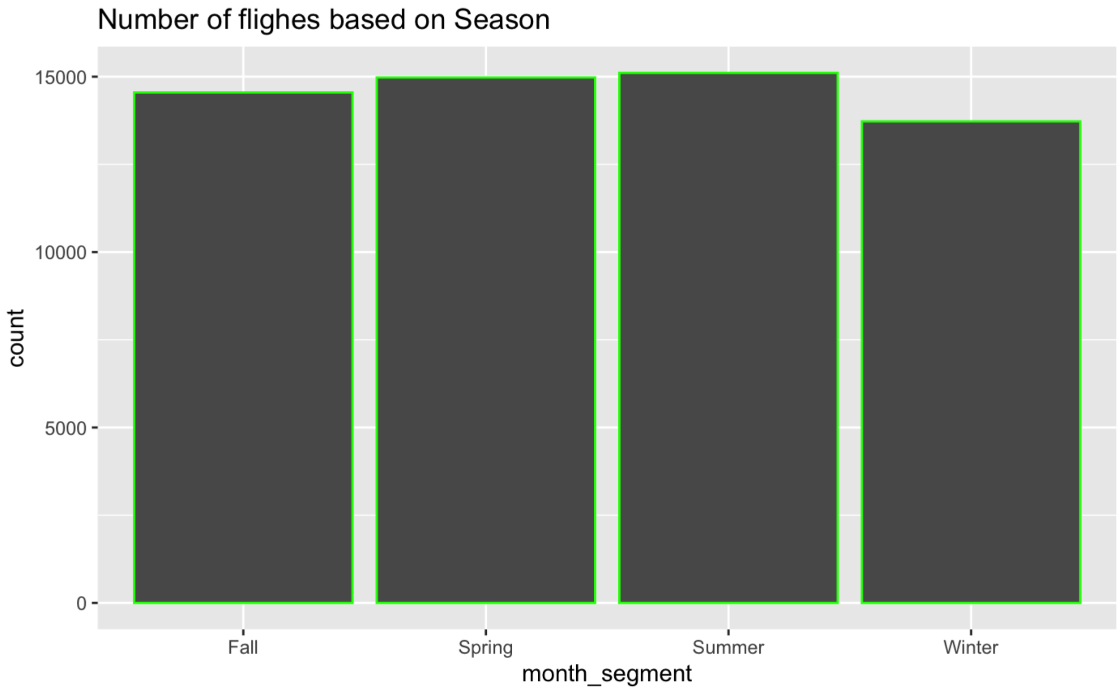


Figure 11: Box Plot of Departure delays based on each season

Based on Figure 11, it appears that, with the exception of winter, the number of flights in each season is comparable.

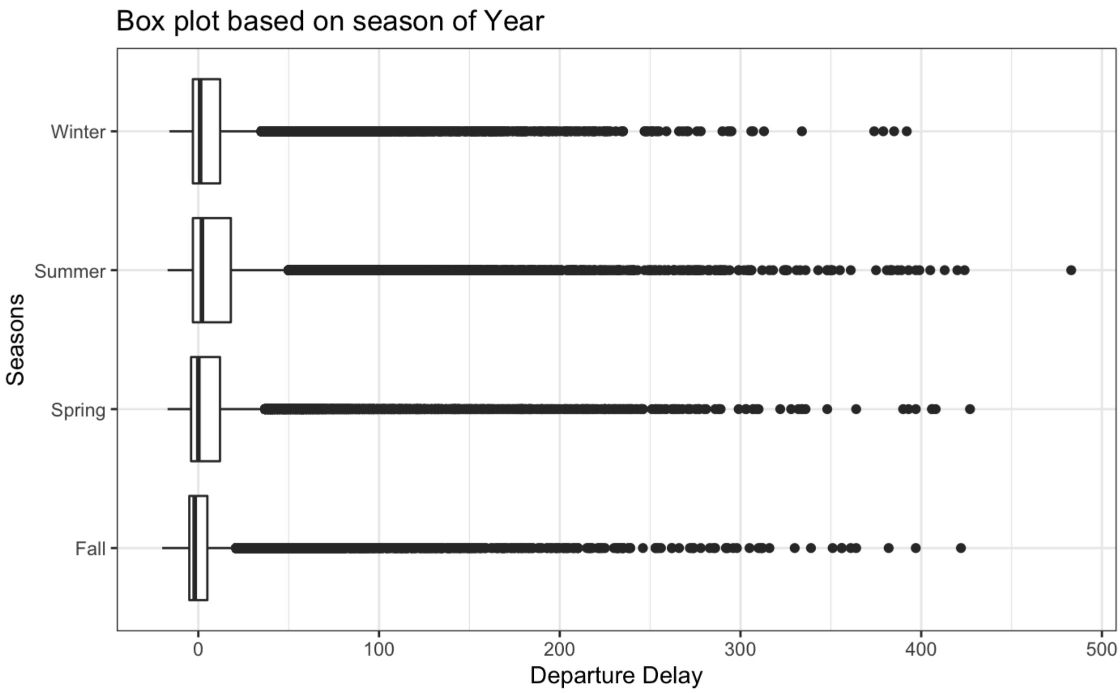
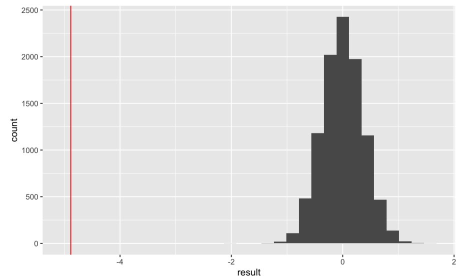
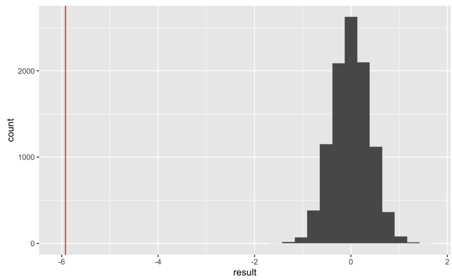


Figure 12: Box Plot of Departure delays based on each season

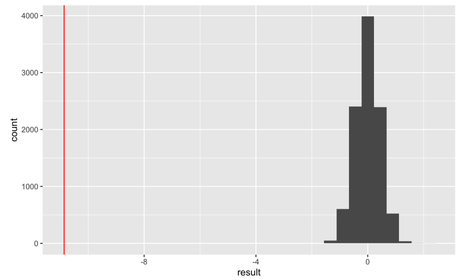
Looking at the boxplots in Figure 12, we can see that the median departure delays change somewhat across seasons. Early departures tend to be more common in the Summer and Spring than in the Summer and Winter. Performed a permutation test to see if the mean differences were significant across all seasons.



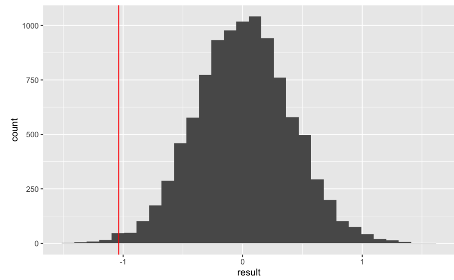
(a) Fall VS Winter p-value 0.0002



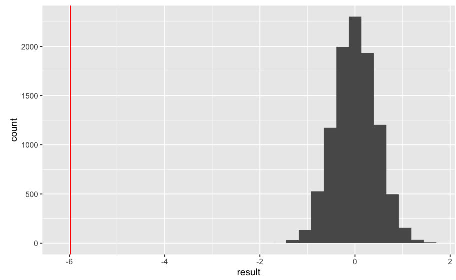
(b) Fall VS Spring p-value 0.0002



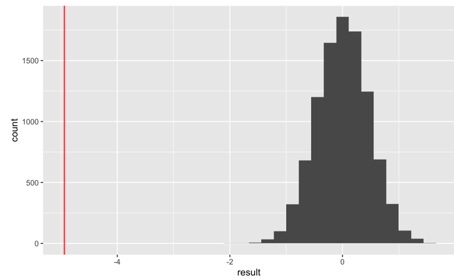
(c) Fall VS Summer p-value 0.0002



(d) Winter VS Spring p-value 0.0094



(e) Winter VS Summer p-value 0.0002



(f) Spring VS Summer p-value 0.0002

Figure 13: Seasons of Year: Histograms of Permutation Tests,Difference in Mean Departure Delays

Let's look at the difference in the mean of departure delays for each season. According to (Figure 13), the observed value (the red line) deviates significantly from the permutation distribution for each season. The majority of the p-values are small (0.0002). However, the p-value for Winter VS Spring varies from others, with a larger p-value = 0.0094. We can infer that there is a significant variation in the means of departure delays across four seasons because all of the p-values are less than 5%. [demo]graphicx subcaption

6 Temperature and Departure Delay

Let's analyse temperature variable.
NOTE: Temperature is in Fahrenheit
Minimum temperature : 10.94
Maximum temperature : 100.04
The number of recordings with missing temperature values is 7. For these 7 values, the mean temperature value has been imputed.

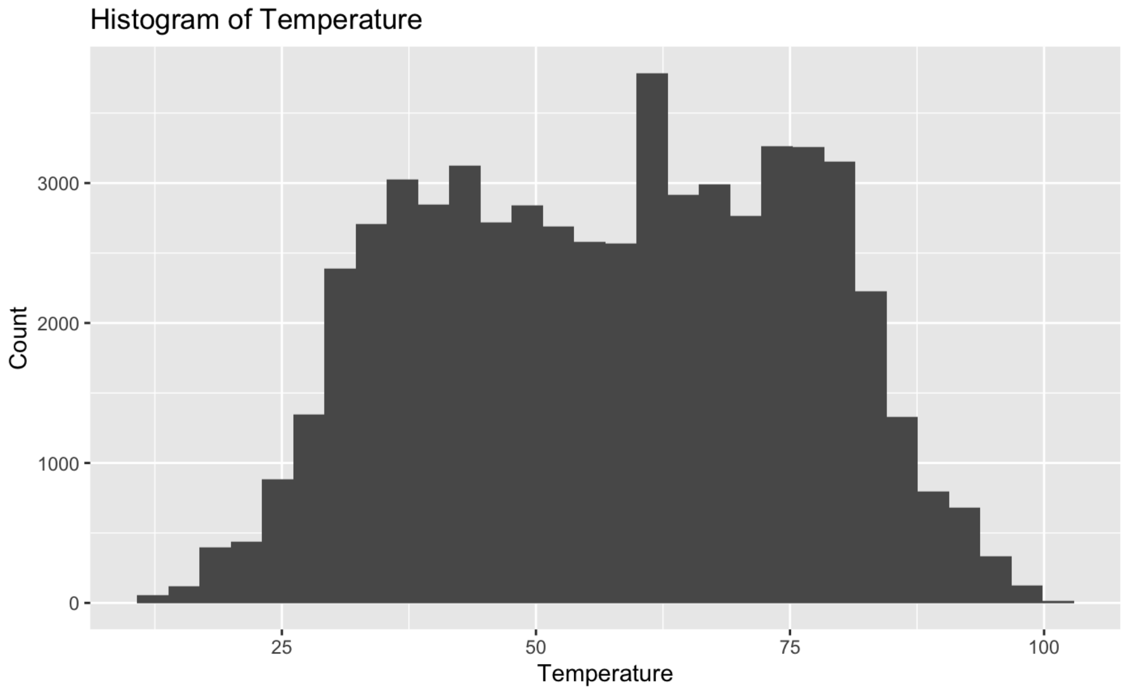
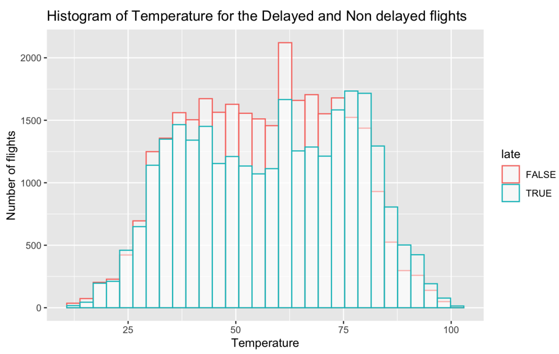


Figure 14: Histogram of temperature

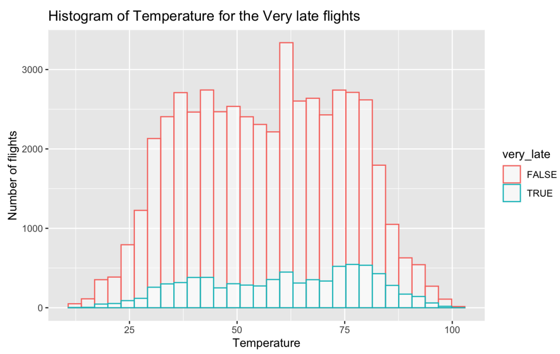
We can see that the temperature distribution is symmetric and following a normal distribution.

Let's look at the temperature for flights that were late or very late.

Both delayed and non-delayed flights have the same temperature distribution for Late and Very Late flights. We can't conclude that the temperature was impacting the delay in flights.

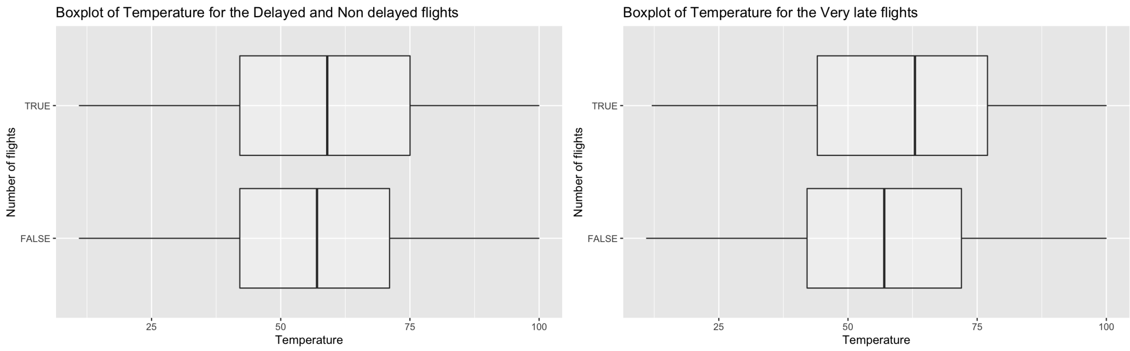


(a) Histogram of temperature late



(b) Histogram of temperature for very late flights

Figure 15: Histogram of temperature for the Late / Very Late flights



(a) Box plot of late flights

(b) Box plot of very late flights

Figure 16: Temperature: Boxplot of temperature Tests based on flights which were late and very late

According to figure 17, We can see that the median temperature for the flights which were late is more compare to the time which were having delay less than 30 minutes. We can conduct a permutation test to see if there's any impact of temperature on the flights which were late or very late.

The observed mean temperature difference is significantly distant from the permutation distribution. This test had a p-value of 0.0002. This indicates that a difference in mean temperatures as large as we discovered has only a 0.0002 chance of occurring by chance, implying that it is due to an actual difference between the two groups. As a result, we may deduce that extremely late flights and non-very late flights differ depending on temperature.

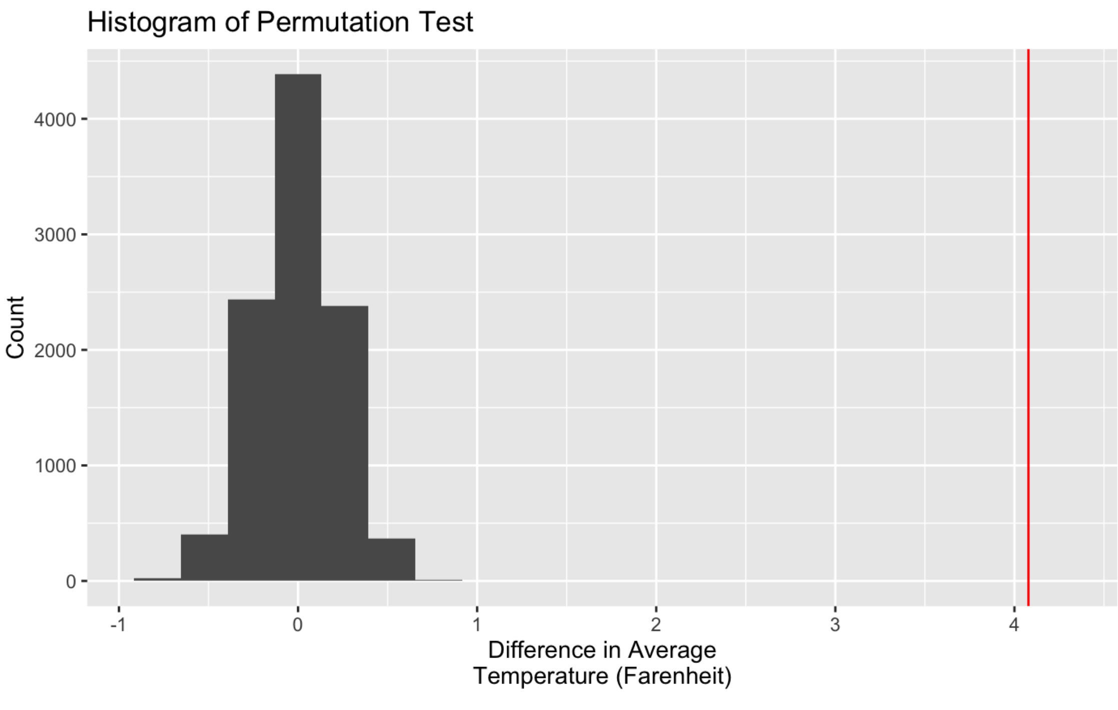


Figure 17: Permutation distribution of Mean based on temperature

24.98 and 89.06 are the 95% confidence value for the dataset. Based on these values we can find the extreme temperatures and see if there are any flights which are delayed or non delayed.

7 Wind speed and Departure Delay

Is there impact of wind speed on departure delays of flight? Let's plot at the histogram of wind speed and see how the distribution looks. We can see that the wind speed is almost normally distributed. The average wind speed across all the flights is 15 miles per hour. There are instances where we can see that there's a peak in the graph. This might indicate that most of the time the wind speed is in that range in New York city. Flights that departed "very late" had an average wind speed of 15.9 miles per hour, while flights that did not depart very late but were late had an average wind speed of 15.6 miles per hour. We used a permutation test to see if the difference between the means was significant.

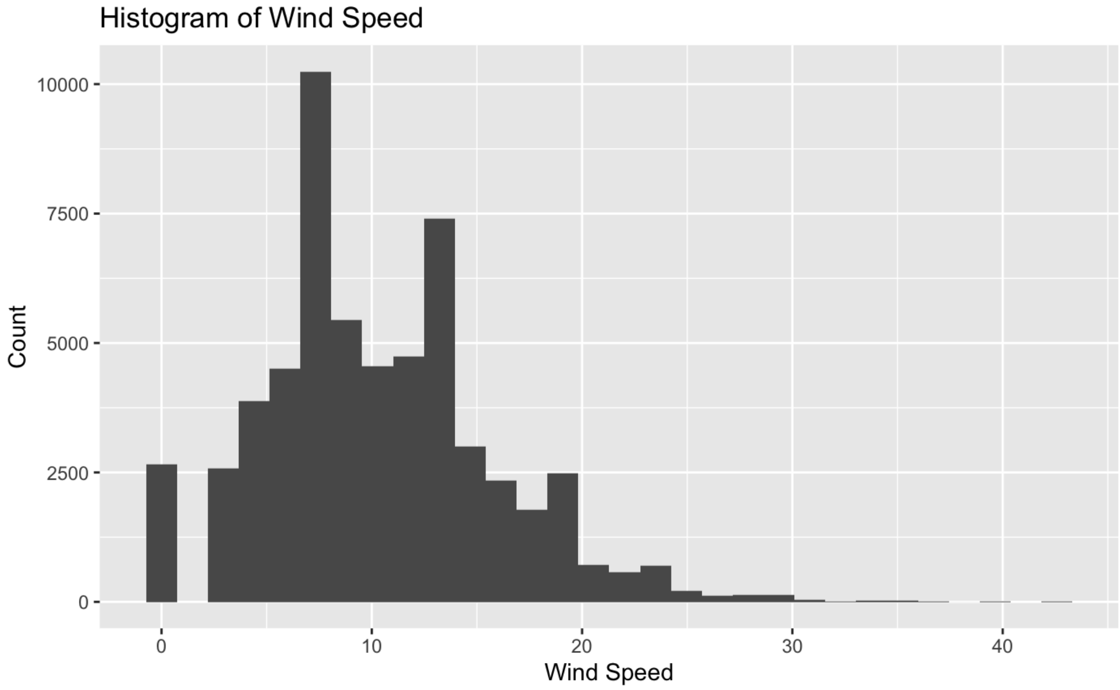


Figure 18: Histogram of Wind Speed

We can see that the observed mean value is on the right side of the distribution of the permutation test. We can see that a p-value of 0.032 less than 0.05, we may infer that the difference between the means was significant and was most likely not attributable to chance. This means that we may also assert that departure delays are affected by wind speed.

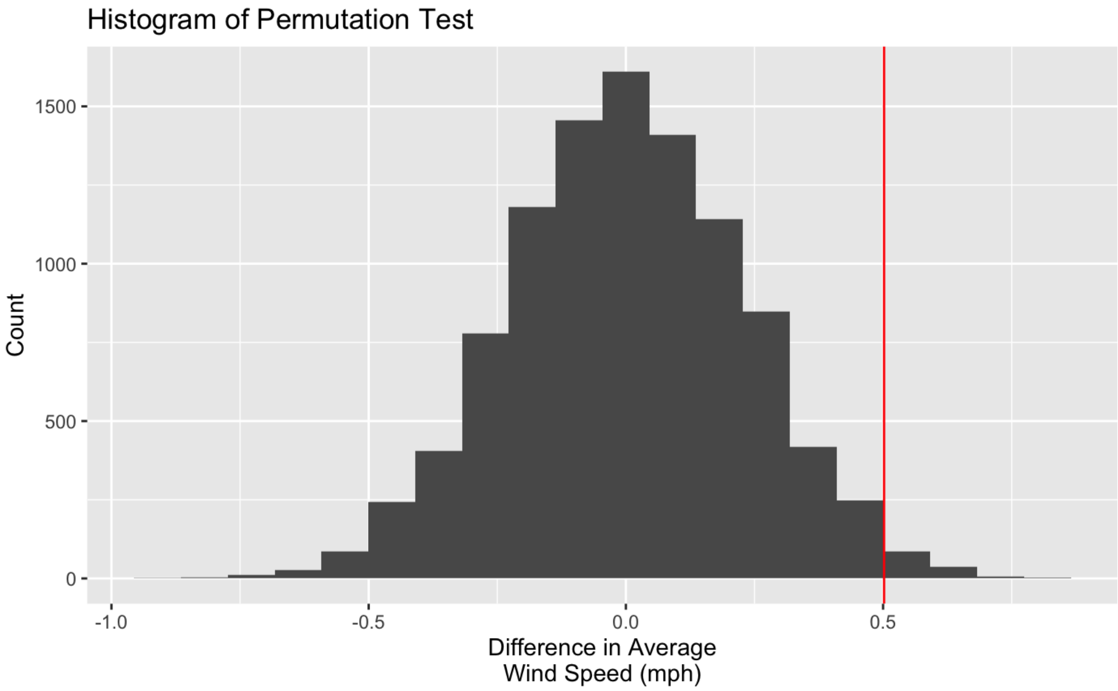


Figure 19: Permutation distribution of Mean of Wind Speed

8 Visibility and Departure Delay

Let's explore the visibility and departure delays of the flights. By looking at the data we can see that most of the time visibility is around 10 miles per hour. This distribution is considered to be substantially negatively skewed. When we compare the average visibility of the two departure delay groups, we notice that the very late group had 9.6 miles of visibility while the not very late group had visibility of 9.8 miles.

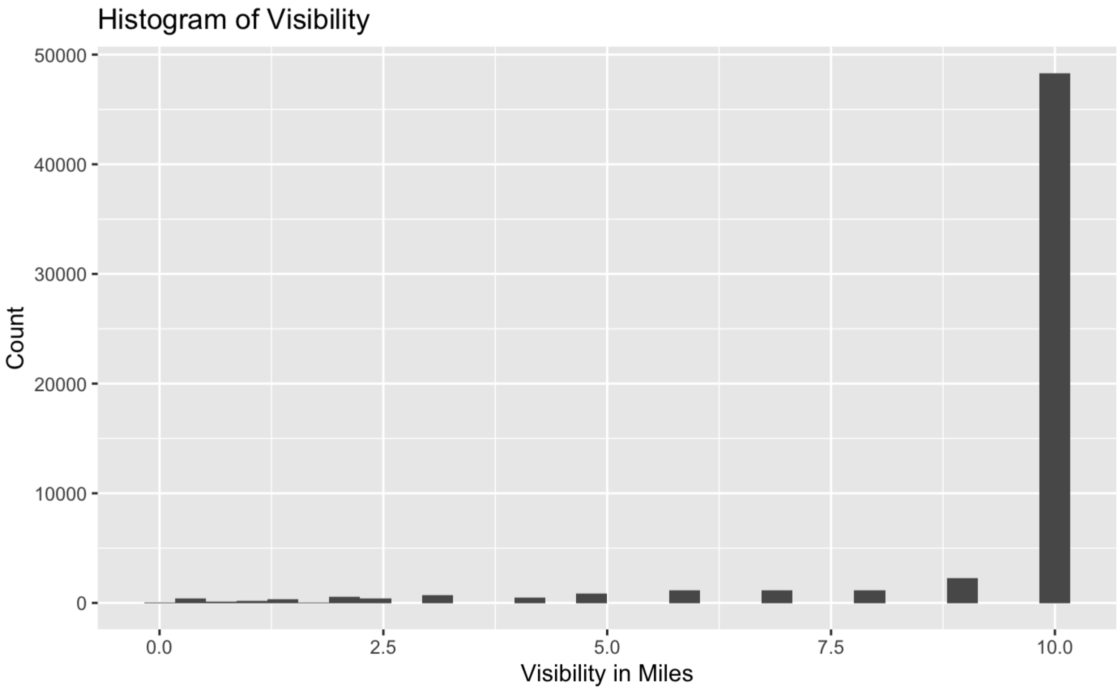


Figure 20: Histogram of Visibility

It is understandable that certain aircraft may be delayed owing to poor visibility, thus it is critical to determine if this discrepancy in averages is due to chance. We will conduct a hypothesis test to conclude our findings. In Figure 21, you can see that the difference in averages between the very late and not very late groups (denoted by the red line) deviates significantly from the permutation distribution.

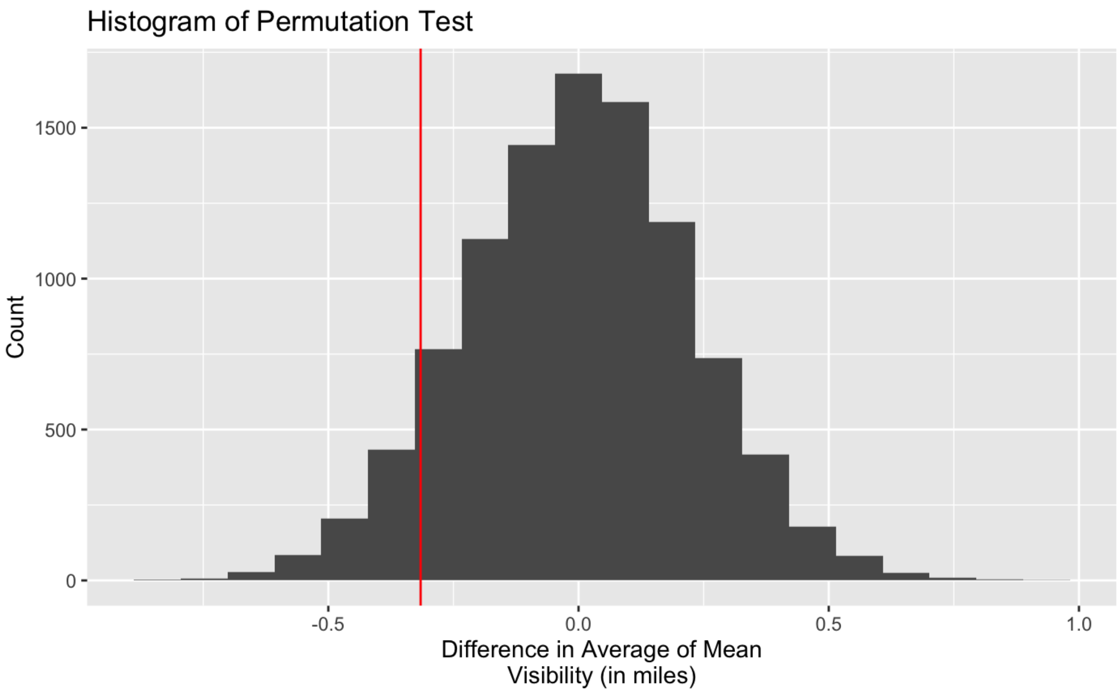


Figure 21: Permutation distribution of Mean of Visibility

The permutation test findings indicate that the difference between the two groups we observed was not just due to chance but was significant. The p-value of 0.0002 that there's a possibility that the difference we discovered was due to a chance. As a result, we may deduce that late flights and the very late flights differs depending on the visibility.

9 Precipitation and Departure Delay

Is there any impact of precipitation on departure delays? Let's plot the Histogram of precipitation for all the flights. According to Figure 22, the majority of the flights in our dataset experienced very little or no precipitation at the time of plane departure.There are few notable outliers may be due to heavy rainfall in New York city.

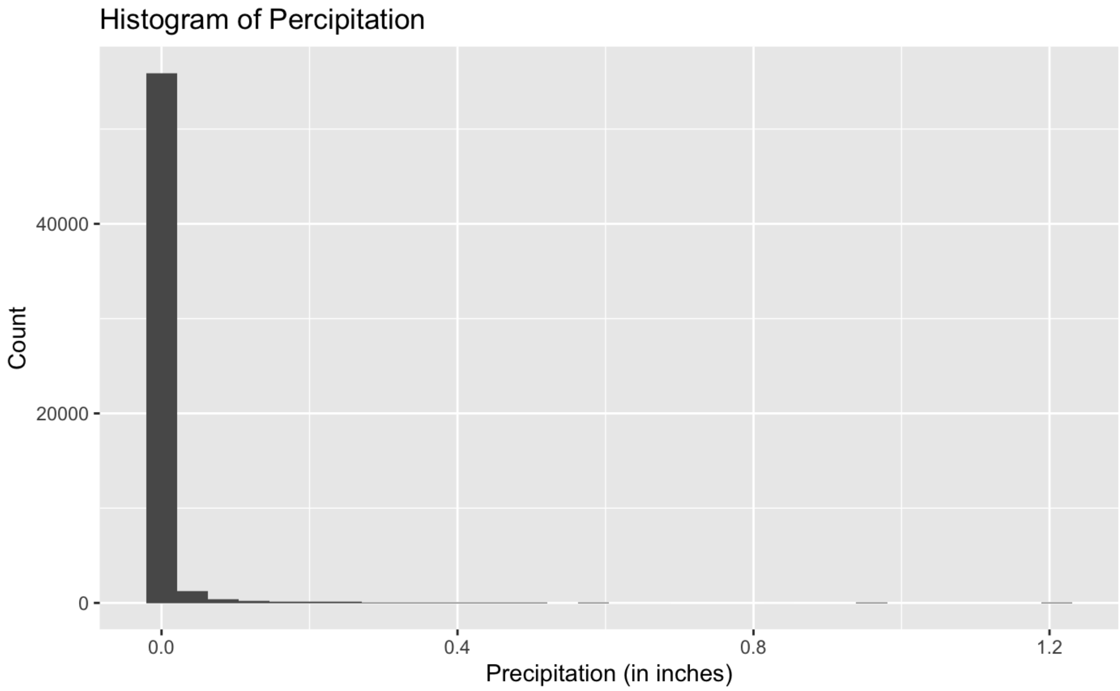


Figure 22: Histogram of Precipitation

Let's perform a permutation test to examine if there is a significant variation in rainfall between very late and not very late flights.According to Figure 23, the observed difference in precipitation is far from the permutation distribution of the mean of precipitation with the p-value of 0.0002. It suggest that the observed difference in means of very late and not very late flights is just by chance.There's extremely low possibility that the difference between the averages have occurred by chance. In conclusion, this shows that very late flights had considerably different amounts of rain (heavy rain) at intended departure time than non-very late flights, suggesting that precipitation may effect departure delays.

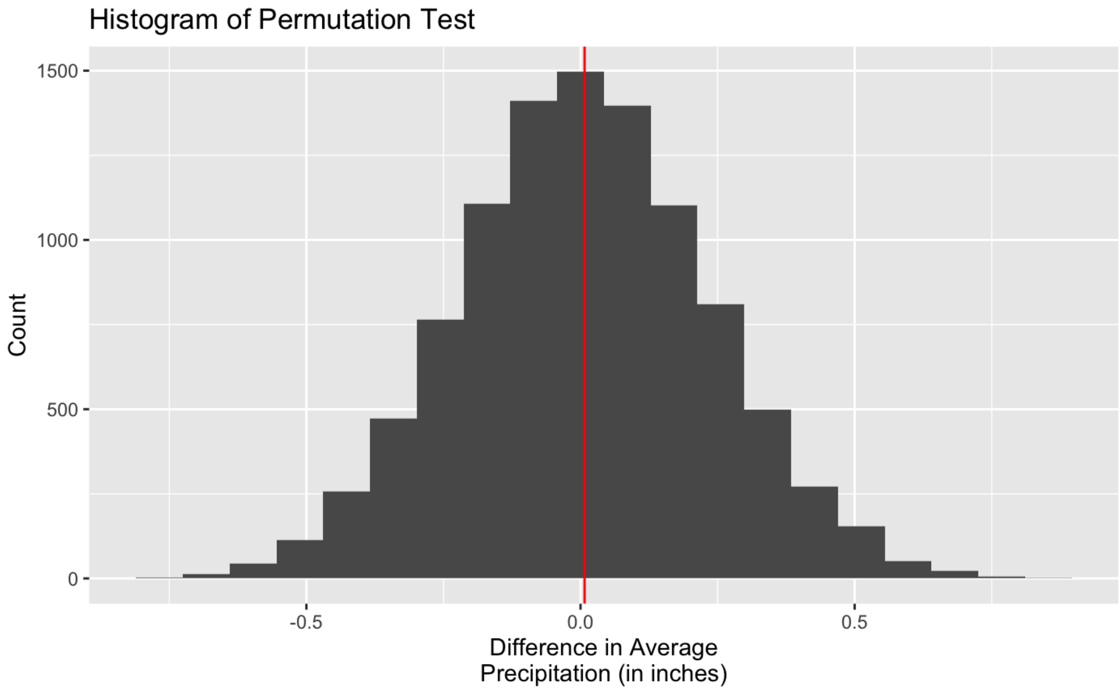


Figure 23: Permutation distribution of Mean of Precipitation

10 Conclusions

Based on the analysis we can conclude that there's impact of certain factors on the departure delay of United Airlines. Time of the day and year have huge impact on the departure delay. One must pay attention to these factors to improve the customer satisfaction and reduce delay time.

Milestone1

Akanksha Sharma

2022-10-14

##Import Libraries

```r
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
library(ggplot2)
library(nycflights13)
library(ggpubr)
library(lubridate)
```

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

```r
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

Let’s try to understand more about the data:

Filter out the data based on the United Airlines carrier

```
UA_flight = flights %>%
  filter(carrier == 'UA')
```

We are going to use UA_flight data for further analysis of this project.
How many rows are there for the United Airlines ?

```
print(paste('Size of dataset for the United Airlines', nrow(UA_flight)))
```

```
## [1] "Size of dataset for the United Airlines 58665"
```

What are the type of variables?

```
glimpse(UA_flight)
```

```
## Rows: 58,665
## Columns: 19
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ dep_time       <int> 517, 533, 554, 558, 558, 559, 607, 611, 623, 628, 643, …
## $ sched_dep_time <int> 515, 529, 558, 600, 600, 600, 607, 600, 627, 630, 646, …
## $ dep_delay      <dbl> 2, 4, -4, -2, -2, -1, 0, 11, -4, -2, -3, 8, 1, 1, -4, -…
## $ arr_time       <int> 830, 850, 740, 924, 923, 854, 858, 945, 933, 1016, 922,…
## $ sched_arr_time <int> 819, 830, 728, 917, 937, 902, 915, 931, 932, 947, 940, …
## $ arr_delay      <dbl> 11, 20, 12, 7, -14, -8, -17, 14, 1, 29, -18, -9, -6, -7…
## $ carrier        <chr> "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "…
## $ flight         <int> 1545, 1714, 1696, 194, 1124, 1187, 1077, 303, 496, 1665…
## $ tailnum        <chr> "N14228", "N24211", "N39463", "N29129", "N53441", "N765…
## $ origin         <chr> "EWR", "LGA", "EWR", "JFK", "EWR", "EWR", "EWR", "JFK",…
## $ dest           <chr> "IAH", "IAH", "ORD", "LAX", "SFO", "LAS", "MIA", "SFO",…
## $ air_time       <dbl> 227, 227, 150, 345, 361, 337, 157, 366, 229, 366, 146, …
## $ distance       <dbl> 1400, 1416, 719, 2475, 2565, 2227, 1085, 2586, 1416, 24…
## $ hour           <dbl> 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7…
## $ minute         <dbl> 15, 29, 58, 0, 0, 0, 7, 0, 27, 30, 46, 36, 45, 45, 0, 0…
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
```

With this we can see the different type of variables in the dataset.
Data type of the variables which are in scope :
1. Time of year : time_hour dttm format
2. Temperature : temp (Stored in weather dataset)
3. Wind Speed : wind_speed (Stored in weather dataset)
4. Precipitation : precip (Stored in weather dataset)
5. Visibility : visib (Stored in miles)

We need to join the dataset UA_flight with the Weather dataset.

```
glimpse(weather)
```

```
## Rows: 26,115
## Columns: 15
## $ origin     <chr> "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EW…
## $ year       <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013,…
## $ month      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
## $ day        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…
## $ hour       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, …
## $ temp       <dbl> 39.02, 39.02, 39.02, 39.92, 39.02, 37.94, 39.02, 39.92, 39.…
## $ dewp       <dbl> 26.06, 26.96, 28.04, 28.04, 28.04, 28.04, 28.04, 28.04, 28.…
## $ humid      <dbl> 59.37, 61.63, 64.43, 62.21, 64.43, 67.21, 64.43, 62.21, 62.…
## $ wind_dir   <dbl> 270, 250, 240, 250, 260, 240, 240, 250, 260, 260, 260, 330,…
## $ wind_speed <dbl> 10.35702, 8.05546, 11.50780, 12.65858, 12.65858, 11.50780, …
## $ wind_gust  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 20.…
## $ precip     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
## $ pressure   <dbl> 1012.0, 1012.3, 1012.5, 1012.2, 1011.9, 1012.4, 1012.2, 101…
## $ visib      <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…
## $ time_hour  <dttm> 2013-01-01 01:00:00, 2013-01-01 02:00:00, 2013-01-01 03:00…
```

```
UA_flight_weather = UA_flight %>%
  inner_join(weather, by = c('year','month','day','hour','origin'))
glimpse(UA_flight_weather)
```

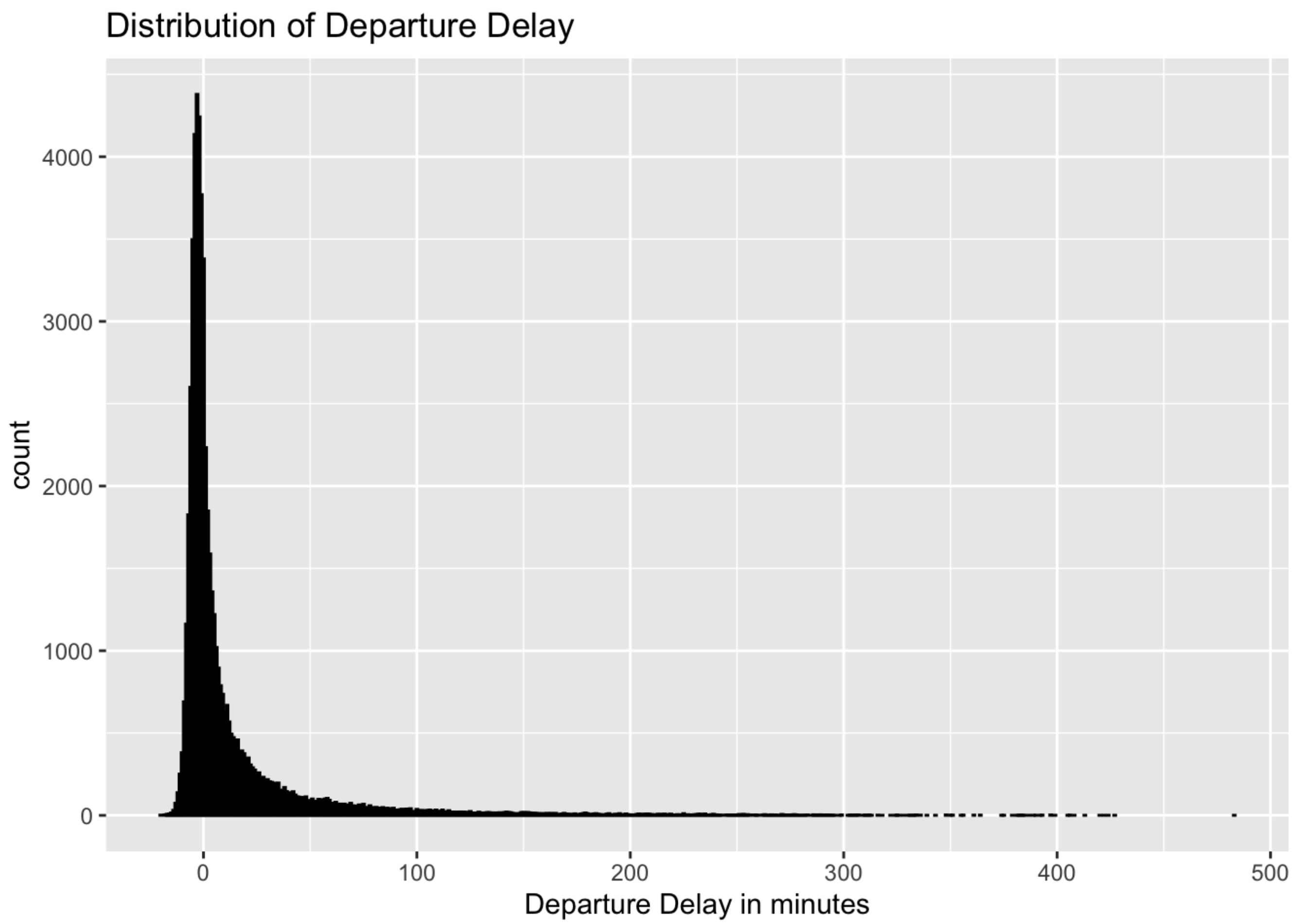
```
## Rows: 58,361
## Columns: 29
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ dep_time       <int> 517, 533, 554, 558, 558, 559, 607, 611, 623, 628, 643, …
## $ sched_dep_time <int> 515, 529, 558, 600, 600, 600, 607, 600, 627, 630, 646, …
## $ dep_delay      <dbl> 2, 4, -4, -2, -2, -1, 0, 11, -4, -2, -3, 8, 1, 1, -4, -…
## $ arr_time       <int> 830, 850, 740, 924, 923, 854, 858, 945, 933, 1016, 922,…
## $ sched_arr_time <int> 819, 830, 728, 917, 937, 902, 915, 931, 932, 947, 940, …
## $ arr_delay      <dbl> 11, 20, 12, 7, -14, -8, -17, 14, 1, 29, -18, -9, -6, -7…
## $ carrier        <chr> "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "…
## $ flight         <int> 1545, 1714, 1696, 194, 1124, 1187, 1077, 303, 496, 1665…
## $ tailnum        <chr> "N14228", "N24211", "N39463", "N29129", "N53441", "N765…
## $ origin         <chr> "EWR", "LGA", "EWR", "JFK", "EWR", "EWR", "EWR", "JFK",…
## $ dest           <chr> "IAH", "IAH", "ORD", "LAX", "SFO", "LAS", "MIA", "SFO",…
## $ air_time       <dbl> 227, 227, 150, 345, 361, 337, 157, 366, 229, 366, 146, …
## $ distance       <dbl> 1400, 1416, 719, 2475, 2565, 2227, 1085, 2586, 1416, 24…
## $ hour           <dbl> 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7…
## $ minute         <dbl> 15, 29, 58, 0, 0, 0, 7, 0, 27, 30, 46, 36, 45, 45, 0, 0…
## $ time_hour.x    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
## $ temp           <dbl> 39.02, 39.92, 39.02, 37.94, 37.94, 37.94, 37.94, 37.94,…
## $ dewp           <dbl> 28.04, 24.98, 28.04, 26.96, 28.04, 28.04, 28.04, 26.96,…
## $ humid          <dbl> 64.43, 54.81, 64.43, 64.29, 67.21, 67.21, 67.21, 64.29,…
## $ wind_dir       <dbl> 260, 250, 260, 260, 240, 240, 240, 260, 260, 240, 240, …
## $ wind_speed     <dbl> 12.65858, 14.96014, 12.65858, 13.80936, 11.50780, 11.50…
## $ wind_gust      <dbl> NA, 21.86482, NA, NA, NA, NA, NA, NA, 23.01560, NA, NA,…
## $ precip         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ pressure       <dbl> 1011.9, 1011.4, 1011.9, 1012.6, 1012.4, 1012.4, 1012.4,…
## $ visib          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…
## $ time_hour.y    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
```

Please take a note that the late and very_late variable have been added.

#Let's analyse the departure delay for the UA carrier flight

```
#Create a bar plot
ggplot(data = UA_flight_weather , aes(x= dep_delay ))+
  geom_bar(color = 'black') +
  labs(x = "Departure Delay in minutes", title = "Distribution of Departure Delay")
```

```
## Warning: Removed 675 rows containing non-finite values (stat_count).
```



Departure delay is following the log normal distribution

```
summary(UA_flight_weather$dep_delay)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  -20.00   -4.00    0.00   12.09   11.00  483.00     675
```

```
cat('Number of flights for which the departure delay is missing' , sum(is.na(UA_flight_w
eather$dep_delay)),'\n')
```

```
## Number of flights for which the departure delay is missing 675
```

```
cat('Percentage of missing data for departure delays for the UA carrier' ,sum((is.na(UA_
flight_weather$dep_delay))/nrow(UA_flight_weather))*100,'\n')
```

```
## Percentage of missing data for departure delays for the UA carrier 1.156594
```

```
perct <- c(sum(is.na(UA_flight_weather$dep_delay)),sum((is.na(UA_flight_weather$dep_dela
y))/nrow(UA_flight_weather))*100)
perct
```

```
## [1] 675.000000   1.156594
```

```
tab <- matrix(c(sum(is.na(UA_flight_weather$dep_delay)),sum((is.na(UA_flight_weather$dep
_delay))/nrow(UA_flight_weather))*100), ncol=2, byrow=TRUE)
colnames(tab) <- c('Null values in dataset','Percentage of null values')

kable(tab) %>%
  kable_styling()
```

Null values in dataset	Percentage of null values
675	1.156594

```
tab %>%
  kbl() %>%
 kable_paper("hover", full_width = F)
```

Null values in dataset	Percentage of null values
675	1.156594

```
# Impute missing values with mean in departure delay column
UA_flight_weather$dep_delay <- with(UA_flight_weather, impute(dep_delay, mean))
```

Add Late and Very_late variable in the dataset

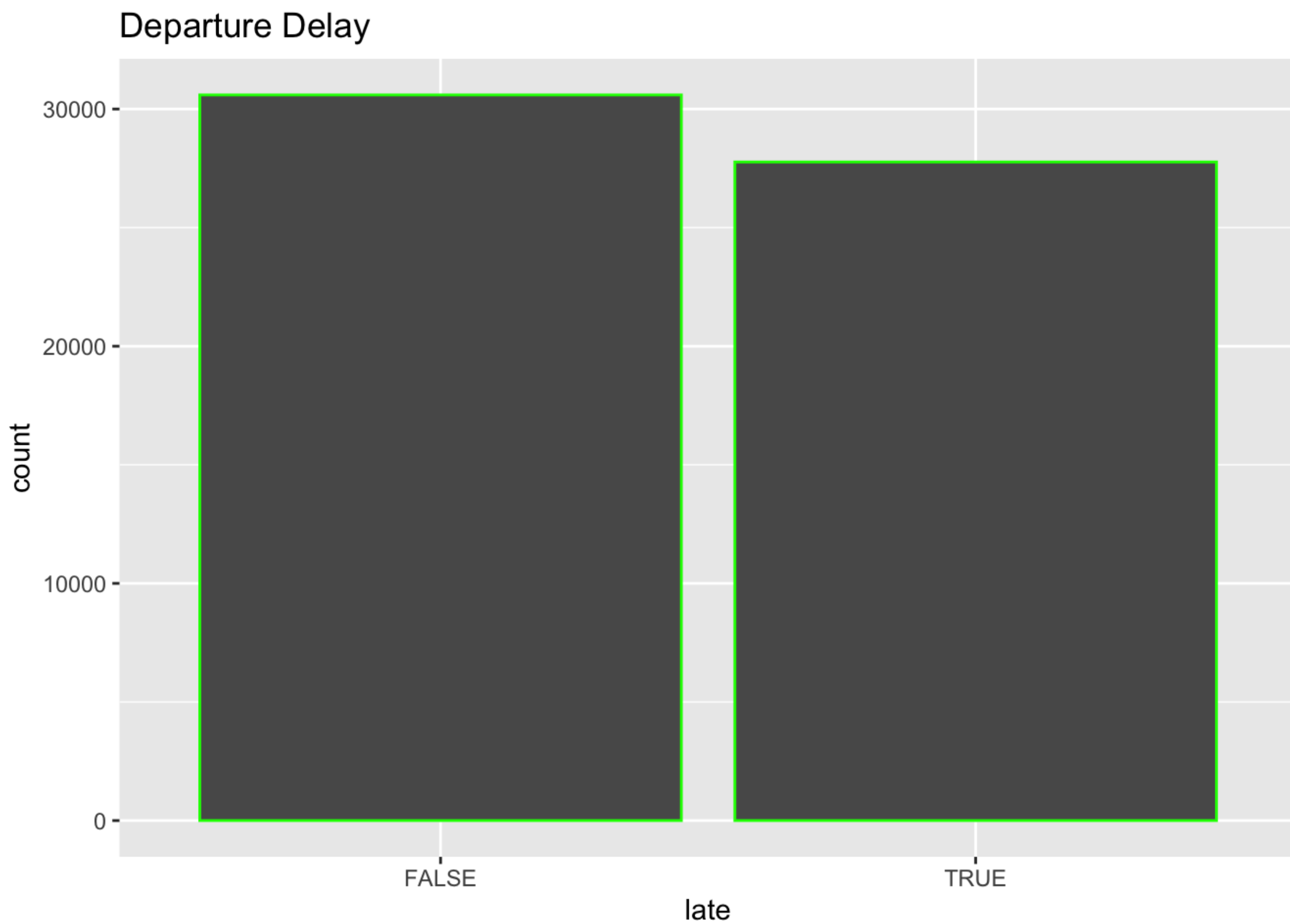
```
#Add late and Very Late columns in the dataset
UA_flight_weather <- UA_flight_weather %>%
  mutate(late = case_when(dep_delay > 0 ~ TRUE,
                          dep_delay <=0 ~ FALSE ),
         very_late = case_when(dep_delay > 30 ~ TRUE,
                          dep_delay <= 30 ~ FALSE ))
glimpse(UA_flight_weather)
```

```
## Rows: 58,361
## Columns: 31
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ dep_time       <int> 517, 533, 554, 558, 558, 559, 607, 611, 623, 628, 643, …
## $ sched_dep_time <int> 515, 529, 558, 600, 600, 600, 607, 600, 627, 630, 646, …
## $ dep_delay      <dbl> 2, 4, -4, -2, -2, -1, 0, 11, -4, -2, -3, 8, 1, 1, -4, -…
## $ arr_time       <int> 830, 850, 740, 924, 923, 854, 858, 945, 933, 1016, 922,…
## $ sched_arr_time <int> 819, 830, 728, 917, 937, 902, 915, 931, 932, 947, 940, …
## $ arr_delay      <dbl> 11, 20, 12, 7, -14, -8, -17, 14, 1, 29, -18, -9, -6, -7…
## $ carrier        <chr> "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "…
## $ flight         <int> 1545, 1714, 1696, 194, 1124, 1187, 1077, 303, 496, 1665…
## $ tailnum        <chr> "N14228", "N24211", "N39463", "N29129", "N53441", "N765…
## $ origin         <chr> "EWR", "LGA", "EWR", "JFK", "EWR", "EWR", "EWR", "JFK",…
## $ dest           <chr> "IAH", "IAH", "ORD", "LAX", "SFO", "LAS", "MIA", "SFO",…
## $ air_time       <dbl> 227, 227, 150, 345, 361, 337, 157, 366, 229, 366, 146, …
## $ distance       <dbl> 1400, 1416, 719, 2475, 2565, 2227, 1085, 2586, 1416, 24…
## $ hour           <dbl> 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7…
## $ minute         <dbl> 15, 29, 58, 0, 0, 0, 7, 0, 27, 30, 46, 36, 45, 45, 0, 0…
## $ time_hour.x    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
## $ temp           <dbl> 39.02, 39.92, 39.02, 37.94, 37.94, 37.94, 37.94, 37.94,…
## $ dewp           <dbl> 28.04, 24.98, 28.04, 26.96, 28.04, 28.04, 28.04, 26.96,…
## $ humid          <dbl> 64.43, 54.81, 64.43, 64.29, 67.21, 67.21, 67.21, 64.29,…
## $ wind_dir       <dbl> 260, 250, 260, 260, 240, 240, 240, 260, 260, 240, 240, …
## $ wind_speed     <dbl> 12.65858, 14.96014, 12.65858, 13.80936, 11.50780, 11.50…
## $ wind_gust      <dbl> NA, 21.86482, NA, NA, NA, NA, NA, NA, 23.01560, NA, NA,…
## $ precip         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ pressure       <dbl> 1011.9, 1011.4, 1011.9, 1012.6, 1012.4, 1012.4, 1012.4,…
## $ visib          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…
## $ time_hour.y    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
## $ late           <lgl> TRUE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, FA…
## $ very_late      <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,…
```

Let’s analyze the Late variable

Let’s see how many flights were late

```
# Create contigency table
flight_delay_late= table(UA_flight_weather$late)
# Create bar plot
ggplot(data = UA_flight_weather , aes(x= late))+
  geom_bar(color = 'green') +
  ggtitle('Departure Delay')
```



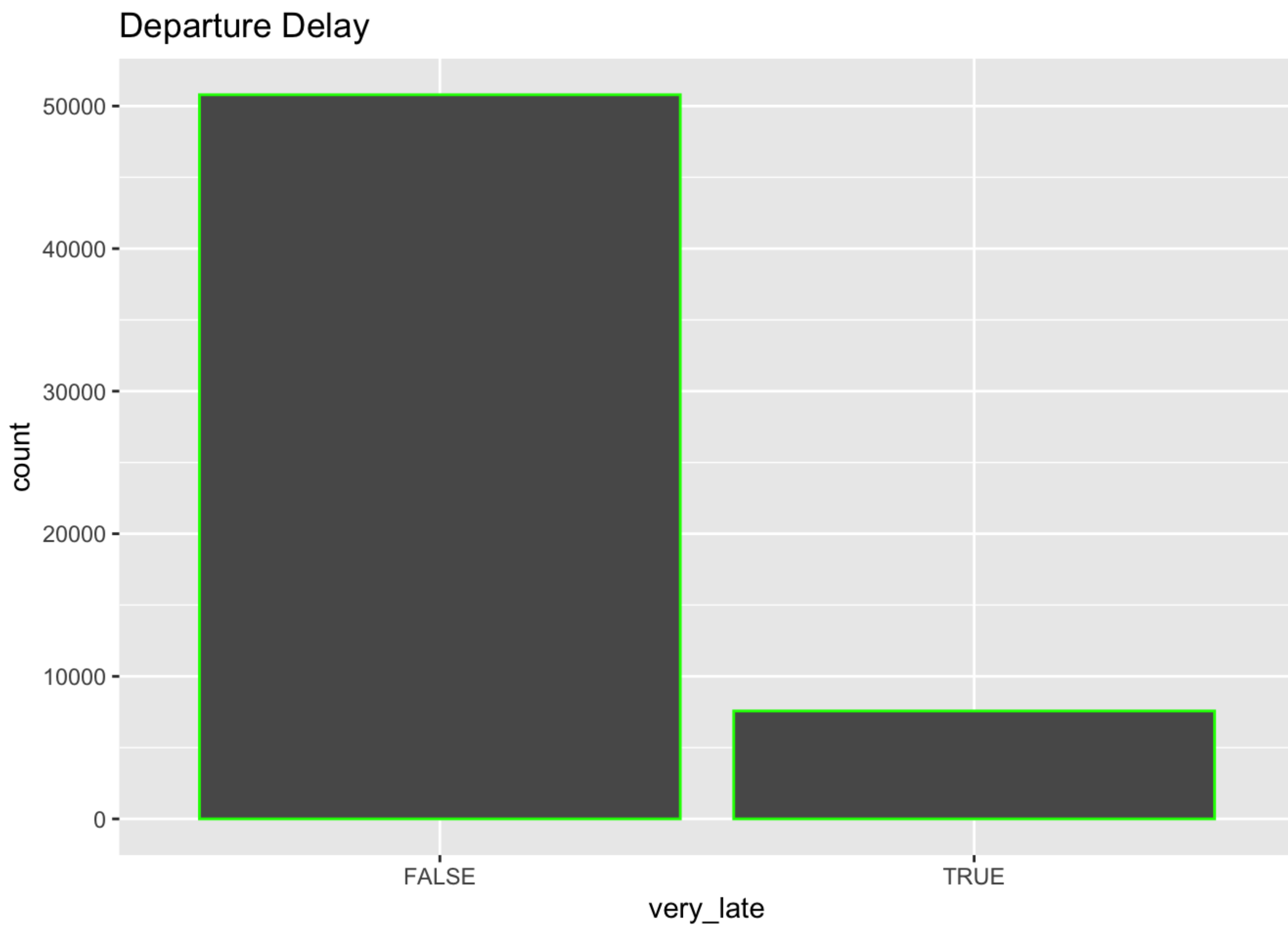

```
cat('%age of delayed flights',(flight_delay_late[2]/nrow(UA_flight_weather))*100)
```

```
## %age of delayed flights 47.57801
```

Let’s analyze the Very late variable

Let’s see how many flights were very late

```
# Create contigency table
flight_delay_very_late= table(UA_flight_weather$very_late)
# Create bar plot
ggplot(data = UA_flight_weather , aes(x= very_late))+
  geom_bar(color = 'green') +
  ggtitle('Departure Delay')
```




```
cat('%age of delayed flights',(flight_delay_very_late[2]/nrow(UA_flight_weather))*100)
```

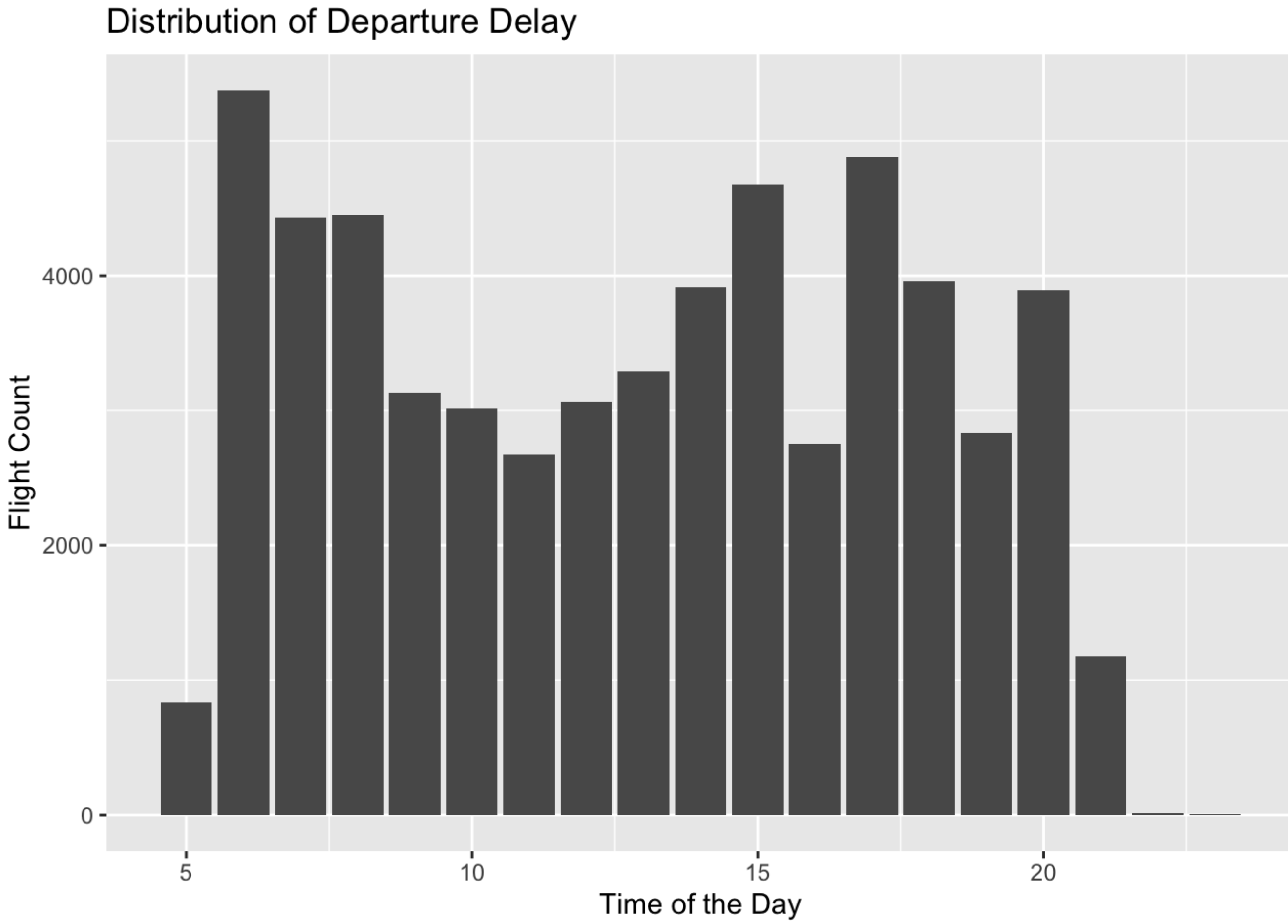
```
## %age of delayed flights 12.96928
```

Only 14.12 % flights were very_late. We need to focus more on the flights which were very late.

Time of day

Let's analyze the time of the day variable with the departure delay

```
ggplot(UA_flight_weather, aes(x= hour))+
  geom_bar()+
  labs(x = "Time of the Day", title = "Distribution of Departure Delay",y = "Flight Coun
t")
```



We

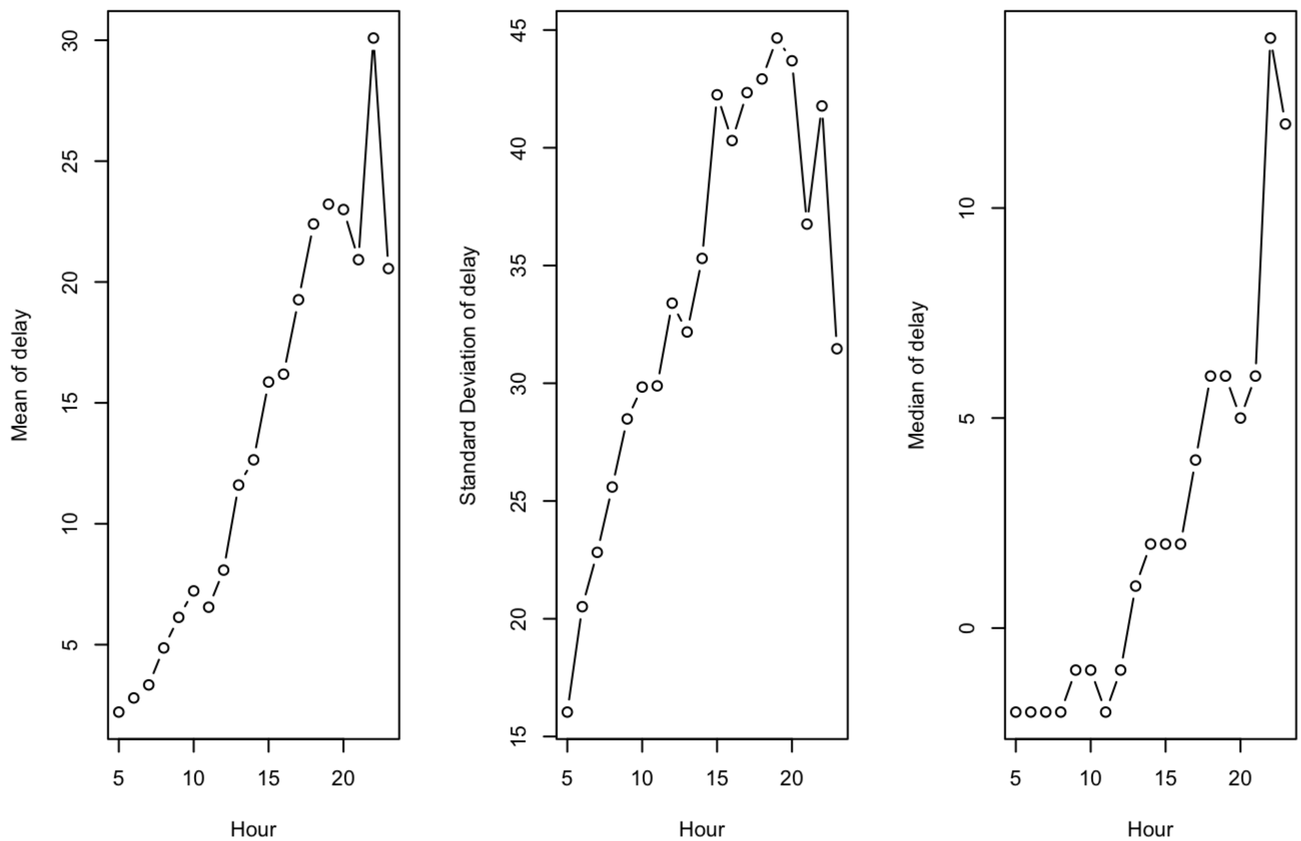
can see the number of flights for each hour and the busiest time for the UA flights is 6 AM.

We can see that there’s no flight which is operated during midnight to 5 o’clock.
Let’s see if there’s any relation between the time of hour of the flight with the delay.

```
hour_summary <- UA_flight_weather %>%
 group_by(hour) %>%
  summarise(
    mean_hour = mean(dep_delay),
    sd_hour = sd(dep_delay),
    median_hour = median(dep_delay),
    count_hour = n()
  )
hour_summary
```

```
## # A tibble: 19 × 5
##     hour mean_hour sd_hour median_hour count_hour
##    <dbl>     <dbl>   <dbl>       <dbl>      <int>
##  1     5      2.21    16.0        -2          837
##  2     6      2.80    20.5        -2         5375
##  3     7      3.33    22.8        -2         4430
##  4     8      4.86    25.6        -2         4455
##  5     9      6.13    28.5        -1         3129
##  6    10      7.22    29.8        -1         3011
##  7    11      6.55    29.9        -2         2672
##  8    12      8.08    33.4        -1         3068
##  9    13     11.6     32.2         1         3293
## 10    14     12.6     35.3         2         3916
## 11    15     15.9     42.2         2         4674
## 12    16     16.2     40.3         2         2751
## 13    17     19.3     42.3         4         4877
## 14    18     22.4     42.9         6         3956
## 15    19     23.2     44.7         6         2829
## 16    20     23.0     43.7         5         3890
## 17    21     20.9     36.8         6         1177
## 18    22     30.1     41.8        14.0         12
## 19    23     20.6     31.5        12            9
```

```
par(mfrow=c(1,3))
plot(x = hour_summary$hour,y = hour_summary$mean_hour,type = 'b',xlab = 'Hour',ylab= 'Me
an of delay')
plot(x = hour_summary$hour,y = hour_summary$sd_hour,type = 'b',xlab = 'Hour',ylab= 'Stan
dard Deviation of delay')
plot(x = hour_summary$hour,y = hour_summary$median_hour,type = 'b',xlab = 'Hour',ylab=
'Median of delay')
```



We can see that the delay keep on increasing as we progress over each hour. But we will closely look at the data points for 23 hour. We can see that there are only 9 records for that flight. Hence, we can’t conclude much . Because there are two flights which are having departure dealy of 66 and 80 minutes. It might be the case these are the same flights which got delayed during the day time hence,there’s delay for the connecting flights.

We can make a comparison with each hour of the flight and see how it’s impacting the delay.
Busiest time for the UA carrier airlines :

```
UA_flight_weather %>%
  filter(UA_flight_weather$hour==23)
```

```
## # A tibble: 9 × 31
##    year month   day dep_time sched_dep…¹ dep_d…² arr_t…³ sched…⁴ arr_d…⁵ carrier
##   <int> <int> <int>    <int>       <int>   <dbl>   <int>   <int>   <dbl> <chr>
## 1  2013    11     8     2312        2300      12      14       9       5 UA
## 2  2013    11    15     2303        2300       3       3       9      -6 UA
## 3  2013    11    19     2252        2300      -8    2341      10     -29 UA
## 4  2013    11    22        6        2300      66     113       9      64 UA
## 5  2013    11    27     2256        2300      -4       1       9      -8 UA
## 6  2013    12     1     2258        2300      -2    2350      10     -20 UA
## 7  2013    12     1     2321        2300      21      23      28      -5 UA
## 8  2013     2    14       59        2339      80     205     106      59 UA
## 9  2013     4    29        2        2345      17     222     241     -19 UA
## # … with 21 more variables: flight <int>, tailnum <chr>, origin <chr>,
## #   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour.x <dttm>, temp <dbl>, dewp <dbl>, humid <dbl>, wind_dir <dbl>,
## #   wind_speed <dbl>, wind_gust <dbl>, precip <dbl>, pressure <dbl>,
## #   visib <dbl>, time_hour.y <dttm>, late <lgl>, very_late <lgl>, and
## #   abbreviated variable names ¹sched_dep_time, ²dep_delay, ³arr_time,
## #   ⁴sched_arr_time, ⁵arr_delay
```

Let’s do the analysis based for hour based on late variable

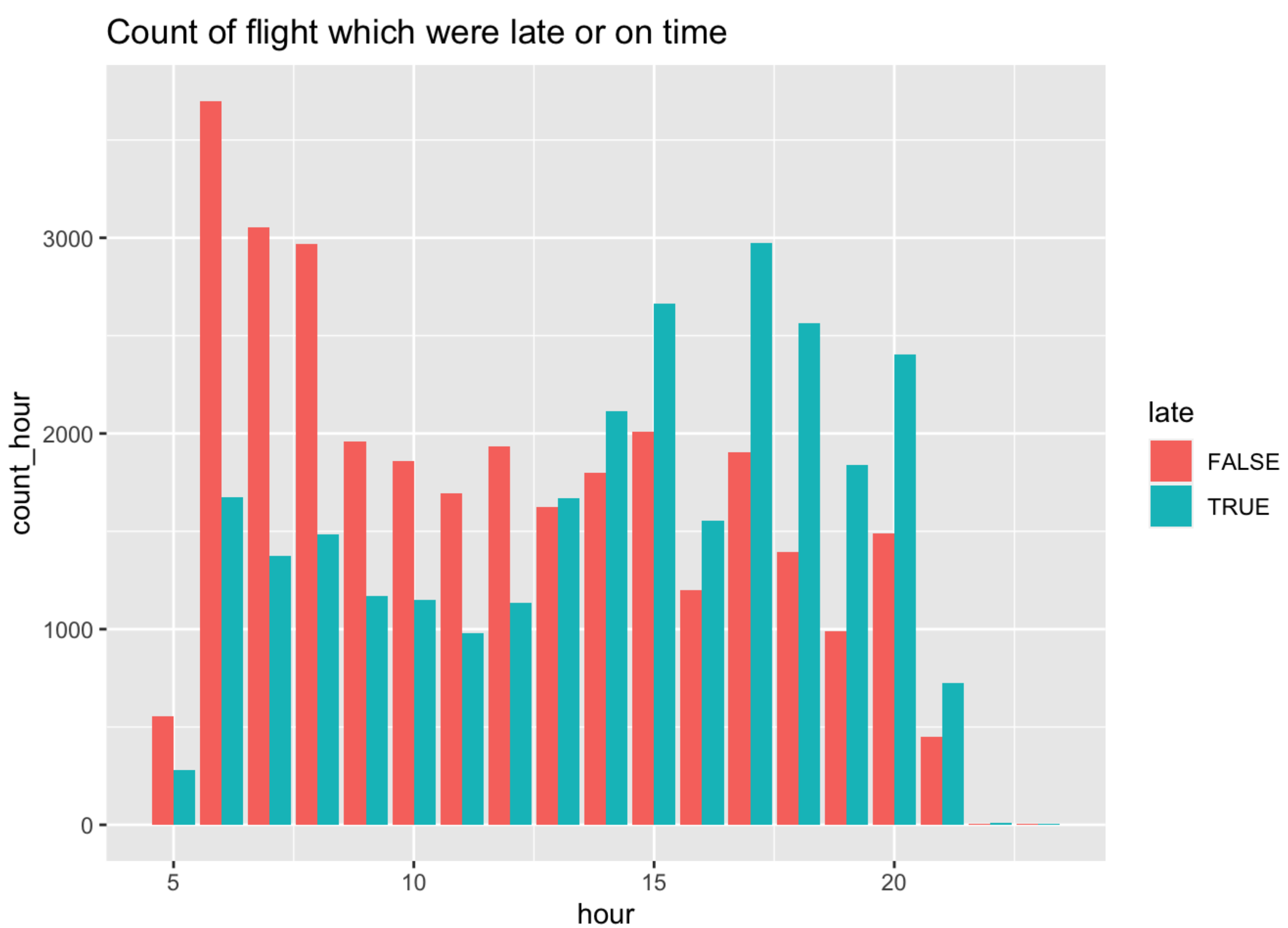
```
hour_summary <- UA_flight_weather %>%
 group_by(hour,late) %>%
  summarise(
    count_hour = n()
  )
```

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

```
hour_summary
```

```
## # A tibble: 38 × 3
## # Groups:   hour [19]
##     hour late  count_hour
##    <dbl> <lgl>      <int>
##  1     5 FALSE        556
##  2     5 TRUE         281
##  3     6 FALSE       3699
##  4     6 TRUE        1676
##  5     7 FALSE       3054
##  6     7 TRUE        1376
##  7     8 FALSE       2971
##  8     8 TRUE        1484
##  9     9 FALSE       1958
## 10     9 TRUE        1171
## # … with 28 more rows
```

```
ggplot(hour_summary,aes(hour,count_hour,fill = late))+
  geom_bar(stat = 'identity', position = 'dodge')+
  labs(title = 'Count of flight which were late or on time')
```



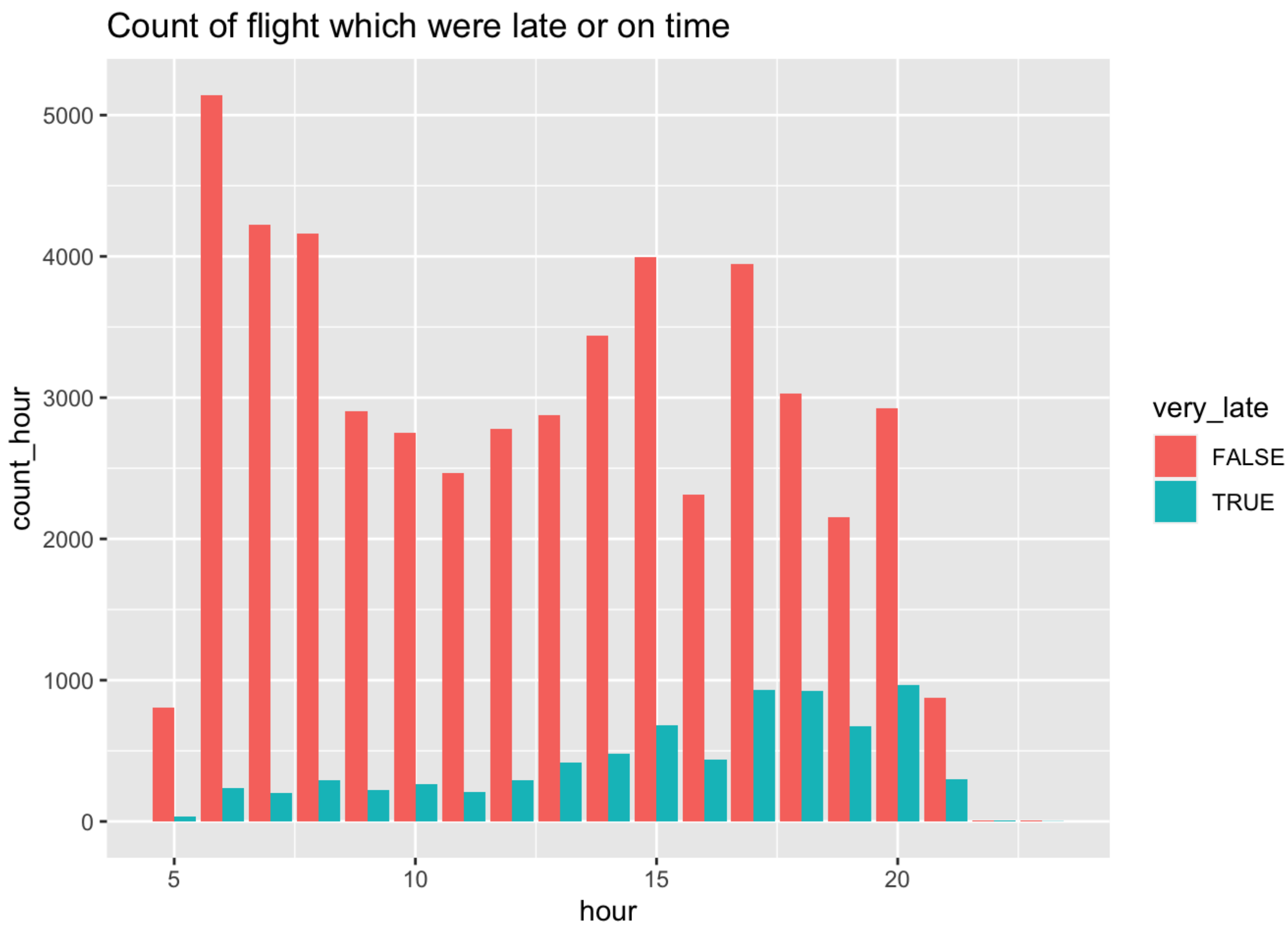
```
hour_summary <- UA_flight_weather %>%
 group_by(hour,very_late) %>%
  summarise(
    count_hour = n()
  )
```

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

```
hour_summary
```

```
## # A tibble: 38 × 3
## # Groups:   hour [19]
##     hour very_late count_hour
##    <dbl> <lgl>          <int>
##  1     5 FALSE            805
##  2     5 TRUE              32
##  3     6 FALSE           5142
##  4     6 TRUE             233
##  5     7 FALSE           4226
##  6     7 TRUE             204
##  7     8 FALSE           4161
##  8     8 TRUE             294
##  9     9 FALSE           2905
## 10     9 TRUE             224
## # … with 28 more rows
```

```
ggplot(hour_summary,aes(hour,count_hour,fill = very_late))+
  geom_bar(stat = 'identity', position = 'dodge')+
  labs(title = 'Count of flight which were late or on time')
```



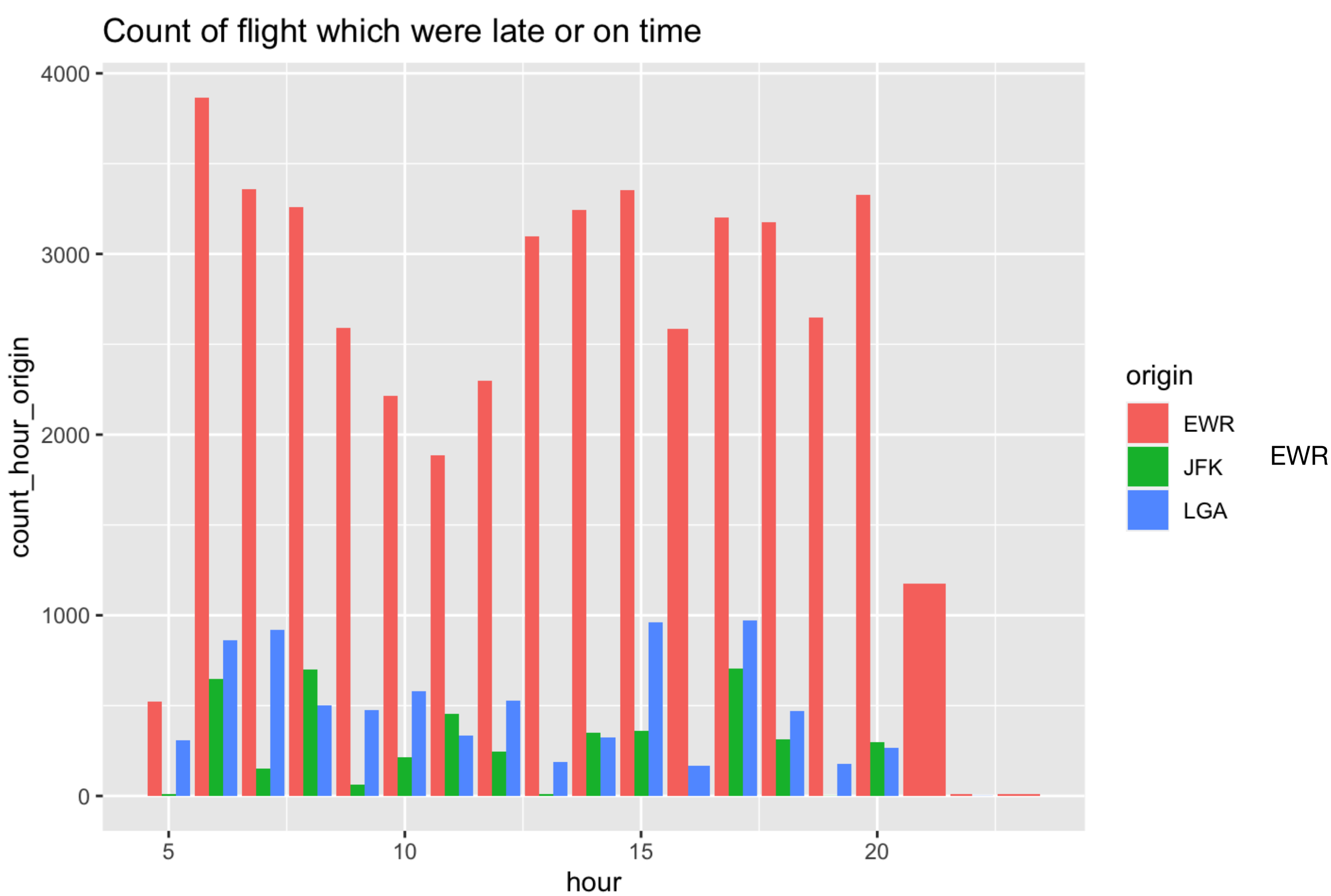

```
hour_summary <- UA_flight_weather %>%
 group_by(hour,origin) %>%
  summarise(
    mean_hour_origin = mean(dep_delay),
    sd_hour_origin = sd(dep_delay),
    median_hour_origin = median(dep_delay),
    count_hour_origin = n()
  )
```

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

```
hour_summary
```

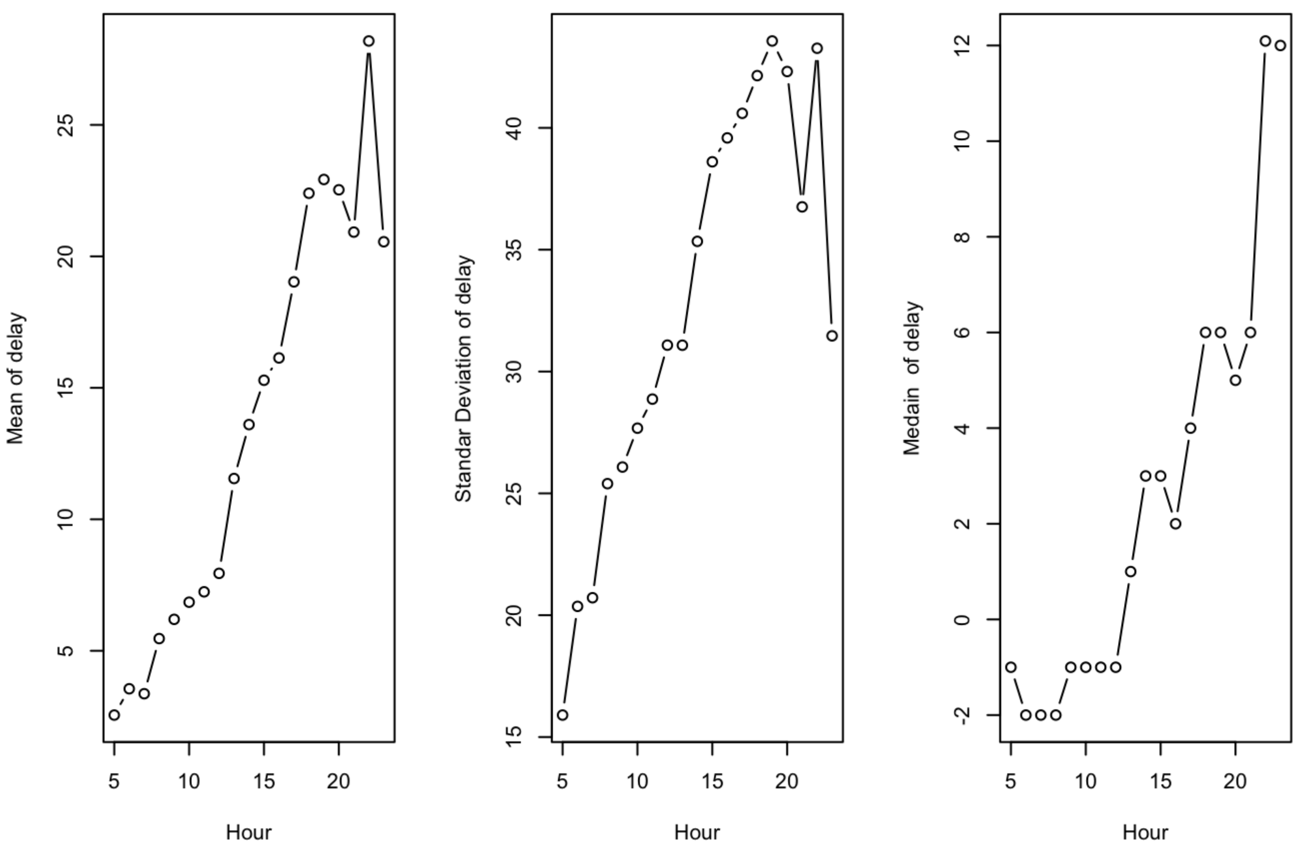
```
## # A tibble: 51 × 6
## # Groups:   hour [19]
##     hour origin mean_hour_origin sd_hour_origin median_hour_origin count_hour_…¹
##    <dbl> <chr>             <dbl>          <dbl>              <dbl>         <int>
##  1     5 EWR               2.56            15.9                 -1           521
##  2     5 JFK              14.8             32.7                  2.5           8
##  3     5 LGA               1.30            15.6                 -2           308
##  4     6 EWR               3.56            20.4                 -2          3866
##  5     6 JFK              -0.970           10.4                 -3           647
##  6     6 LGA               2.21            25.9                 -3           862
##  7     7 EWR               3.37            20.7                 -2          3358
##  8     7 JFK               0.770           16.4                 -3           152
##  9     7 LGA               3.64            29.9                 -3           920
## 10     8 EWR               5.46            25.4                 -2          3257
## # … with 41 more rows, and abbreviated variable name ¹count_hour_origin
```

```
ggplot(hour_summary,aes(hour,count_hour_origin,fill = origin))+
  geom_bar(stat = 'identity', position = 'dodge')+
  labs(title = 'Count of flight which were late or on time')
```



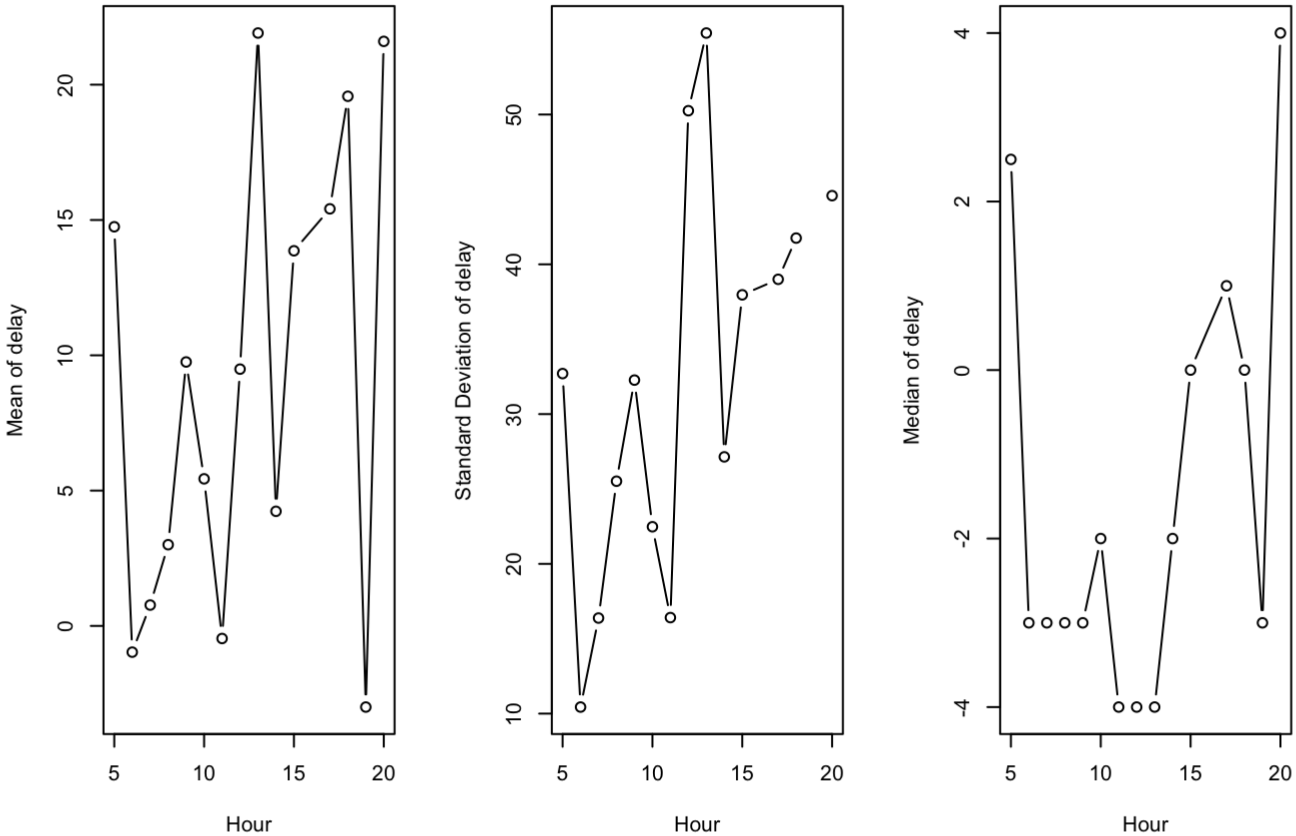
delay based on time

```
hour_summary_EWR <- hour_summary %>%
                filter(origin == 'EWR')

par(mfrow=c(1,3))
plot(x = hour_summary_EWR$hour,y = hour_summary_EWR$mean_hour_origin,type = 'b',xlab =
'Hour',ylab= 'Mean of delay')
plot(x = hour_summary_EWR$hour,y = hour_summary_EWR$sd_hour_origin,type = 'b',xlab = 'Ho
ur',ylab= 'Standar Deviation of delay')
plot(x = hour_summary_EWR$hour,y = hour_summary_EWR$median_hour_origin,type = 'b',xlab =
'Hour',ylab= 'Medain  of delay')
```

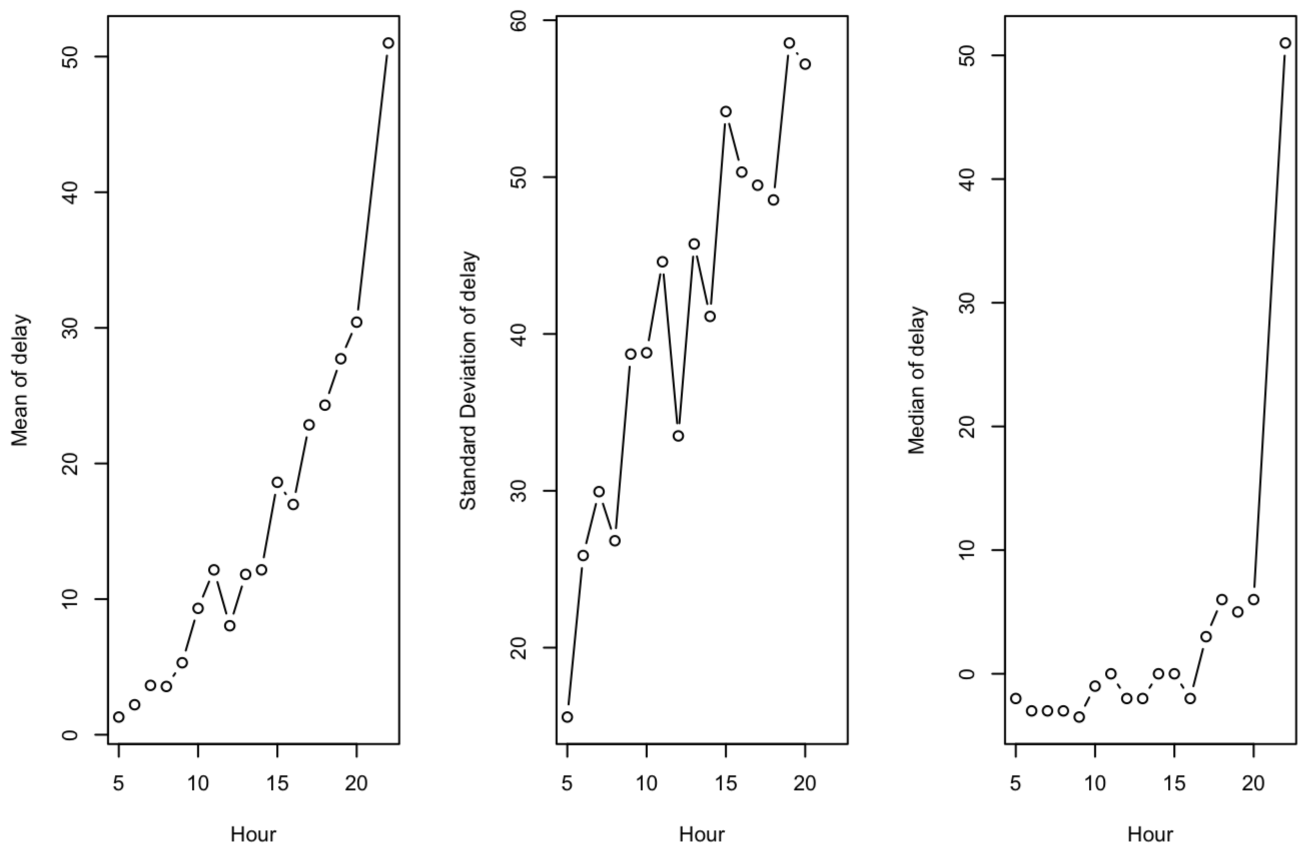


JFK delay based on time

```
hour_summary_JFK <- hour_summary %>%
                filter(origin == 'JFK')

par(mfrow=c(1,3))
plot(x = hour_summary_JFK$hour,y = hour_summary_JFK$mean_hour_origin,type = 'b',xlab =
'Hour',ylab= 'Mean of delay')
plot(x = hour_summary_JFK$hour,y = hour_summary_JFK$sd_hour_origin,type = 'b',xlab = 'Ho
ur',ylab= 'Standard Deviation of delay')
plot(x = hour_summary_JFK$hour,y = hour_summary_JFK$median_hour_origin,type = 'b',xlab =
'Hour',ylab= 'Median of delay')
```




```
hour_summary_LGA <- hour_summary %>%
                filter(origin == 'LGA')

par(mfrow=c(1,3))
plot(x = hour_summary_LGA$hour,y = hour_summary_LGA$mean_hour_origin,type = 'b',xlab =
'Hour',ylab= 'Mean of delay')
plot(x = hour_summary_LGA$hour,y = hour_summary_LGA$sd_hour_origin,type = 'b',xlab = 'Ho
ur',ylab= 'Standard Deviation of delay')
plot(x = hour_summary_LGA$hour,y = hour_summary_LGA$median_hour_origin,type = 'b',xlab =
'Hour',ylab= 'Median of delay')
```




```
total = nrow(UA_flight_weather)
total
```

```
## [1] 58361
```

```
#Find percentage share of each flight
cat('Number of flights for each flight originating from the New York airports')
```

```
## Number of flights for each flight originating from the New York airports
```

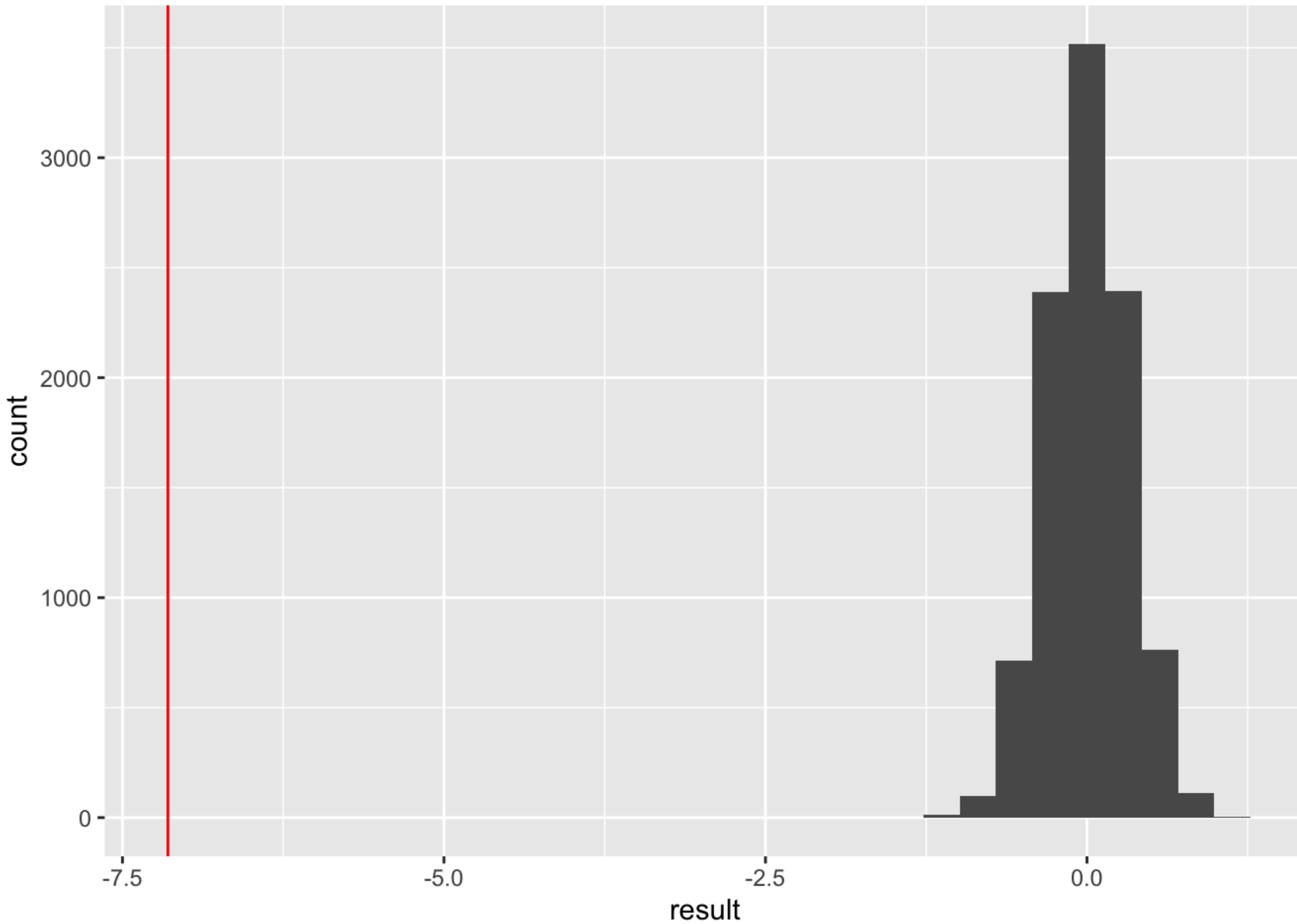
```
flight_percentage_origin <- UA_flight_weather %>%
  group_by(origin)  %>%
  summarise(
    mean_origin = mean(dep_delay),
    sd_origin = sd(dep_delay),
    median_origin = median(dep_delay),
    count_origin = n(),
    per_origin = (n()/total)*100
  )
flight_percentage_origin
```

```
## # A tibble: 3 × 6
##   origin mean_origin sd_origin median_origin count_origin per_origin
##   <chr>        <dbl>     <dbl>         <dbl>        <int>      <dbl>
## 1 EWR          12.5       34.5             1        45820      78.5
## 2 JFK           7.91      32.5            -2         4516       7.74
## 3 LGA          12.1       42.4            -1         8025      13.8
```

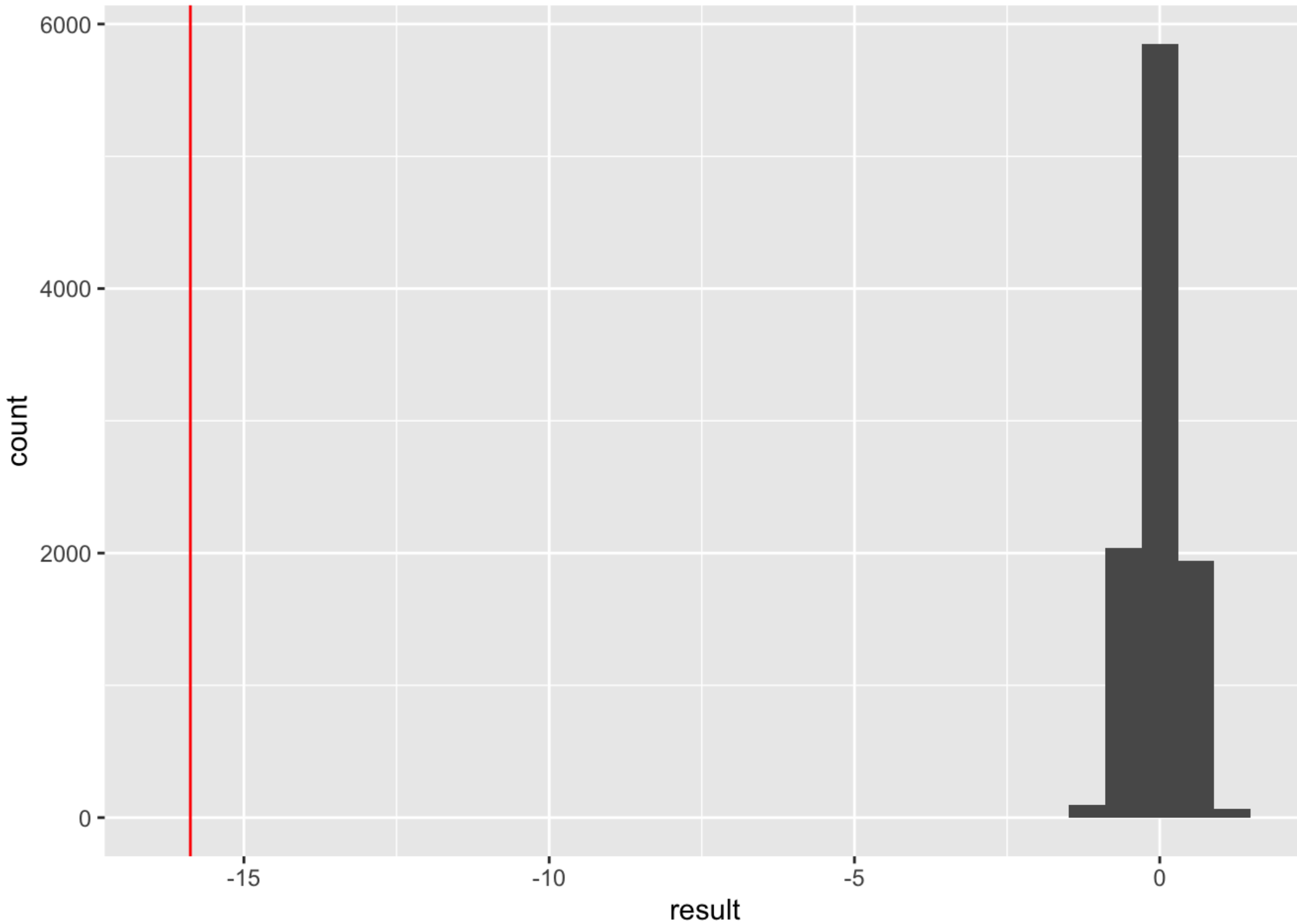
```
#Add late and Very Late columns in the dataset
UA_flight_weather <- UA_flight_weather %>%
  mutate(day_segment = case_when(hour < 11 ~ 'morning',
                         hour >= 11 & hour < 16 ~ 'afternoon',
                         hour >=16 & hour < 20 ~ 'evening',
                         hour >= 20 ~ 'night' )
         )
glimpse(UA_flight_weather)
```

```
## Rows: 58,361
## Columns: 32
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ dep_time       <int> 517, 533, 554, 558, 558, 559, 607, 611, 623, 628, 643, …
## $ sched_dep_time <int> 515, 529, 558, 600, 600, 600, 607, 600, 627, 630, 646, …
## $ dep_delay      <dbl> 2, 4, -4, -2, -2, -1, 0, 11, -4, -2, -3, 8, 1, 1, -4, -…
## $ arr_time       <int> 830, 850, 740, 924, 923, 854, 858, 945, 933, 1016, 922,…
## $ sched_arr_time <int> 819, 830, 728, 917, 937, 902, 915, 931, 932, 947, 940, …
## $ arr_delay      <dbl> 11, 20, 12, 7, -14, -8, -17, 14, 1, 29, -18, -9, -6, -7…
## $ carrier        <chr> "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "…
## $ flight         <int> 1545, 1714, 1696, 194, 1124, 1187, 1077, 303, 496, 1665…
## $ tailnum        <chr> "N14228", "N24211", "N39463", "N29129", "N53441", "N765…
## $ origin         <chr> "EWR", "LGA", "EWR", "JFK", "EWR", "EWR", "EWR", "JFK",…
## $ dest           <chr> "IAH", "IAH", "ORD", "LAX", "SFO", "LAS", "MIA", "SFO",…
## $ air_time       <dbl> 227, 227, 150, 345, 361, 337, 157, 366, 229, 366, 146, …
## $ distance       <dbl> 1400, 1416, 719, 2475, 2565, 2227, 1085, 2586, 1416, 24…
## $ hour           <dbl> 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7…
## $ minute         <dbl> 15, 29, 58, 0, 0, 0, 7, 0, 27, 30, 46, 36, 45, 45, 0, 0…
## $ time_hour.x    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
## $ temp           <dbl> 39.02, 39.92, 39.02, 37.94, 37.94, 37.94, 37.94, 37.94,…
## $ dewp           <dbl> 28.04, 24.98, 28.04, 26.96, 28.04, 28.04, 28.04, 26.96,…
## $ humid          <dbl> 64.43, 54.81, 64.43, 64.29, 67.21, 67.21, 67.21, 64.29,…
## $ wind_dir       <dbl> 260, 250, 260, 260, 240, 240, 240, 260, 260, 240, 240, …
## $ wind_speed     <dbl> 12.65858, 14.96014, 12.65858, 13.80936, 11.50780, 11.50…
## $ wind_gust      <dbl> NA, 21.86482, NA, NA, NA, NA, NA, NA, 23.01560, NA, NA,…
## $ precip         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ pressure       <dbl> 1011.9, 1011.4, 1011.9, 1012.6, 1012.4, 1012.4, 1012.4,…
## $ visib          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…
## $ time_hour.y    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
## $ late           <lgl> TRUE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, FA…
## $ very_late      <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,…
## $ day_segment    <chr> "morning", "morning", "morning", "morning", "morning", …
```

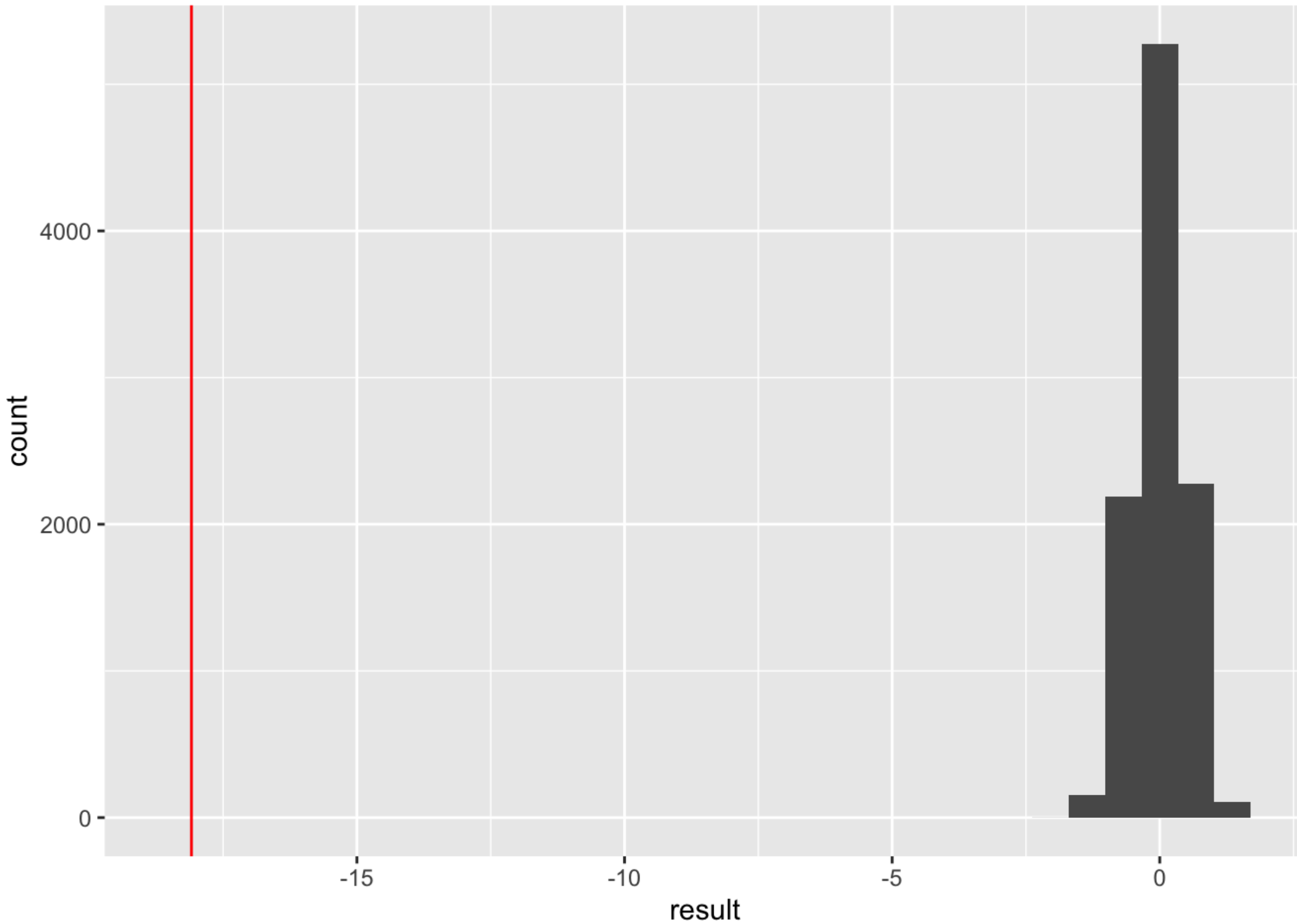
```
# number of simulations
N <- 10^4-1
# vector to store the simulations
result <- numeric(N)
# vector to store the time of the day
vectorDay = c("morning", "afternoon", "evening", "night")
# loop through the time of the day and do permutation testing
#calculate and store the observed difference in the sample
for(i in 1:length(vectorDay))
{
  for(j in 1:length(vectorDay)){
    if(j < 4 & i <= j){
      column1 = (vectorDay[i])
      column2 = (vectorDay[j+1])
      #anlyse the data based on column1 and column2
      reduced_flights <- UA_flight_weather %>%
      filter(day_segment==column1 | day_segment==column2)
      # observations in our sample
      sample.size = nrow(reduced_flights)
      # observations in one of the group
      group.1.size = nrow(reduced_flights[reduced_flights$day_segment==column1,])
      #calculate the observed value
      observed <- mean(reduced_flights$dep_delay[reduced_flights$day_segment ==column1])
-
      mean(reduced_flights$dep_delay[reduced_flights$day_segment == column2])

      for(k in 1:N)
      {
        index = sample(sample.size, size=group.1.size, replace = FALSE)
        result[k] = mean(reduced_flights$dep_delay[index])-
        mean(reduced_flights$dep_delay[-index])
      }
      #print the histograms
      print(ggplot(data=tibble(result), mapping = aes(x=result)) + geom_histogram(bins =
30) + geom_vline(xintercept = observed, color = "red"))
      #Calculate the p-value
      if(observed > 0)
      {
        cat("The permutation for ", column1, " vs ", column2, ": ")
        print(p_value <- 2 * (sum(result >= observed) + 1) / (N + 1))
      }
      else{
        cat("The permutation for ", column1, " vs ", column2, ": ")
        print(p_value <- 2 * (sum(result <= observed) + 1) / (N + 1))
        }
      }
  }
}
```



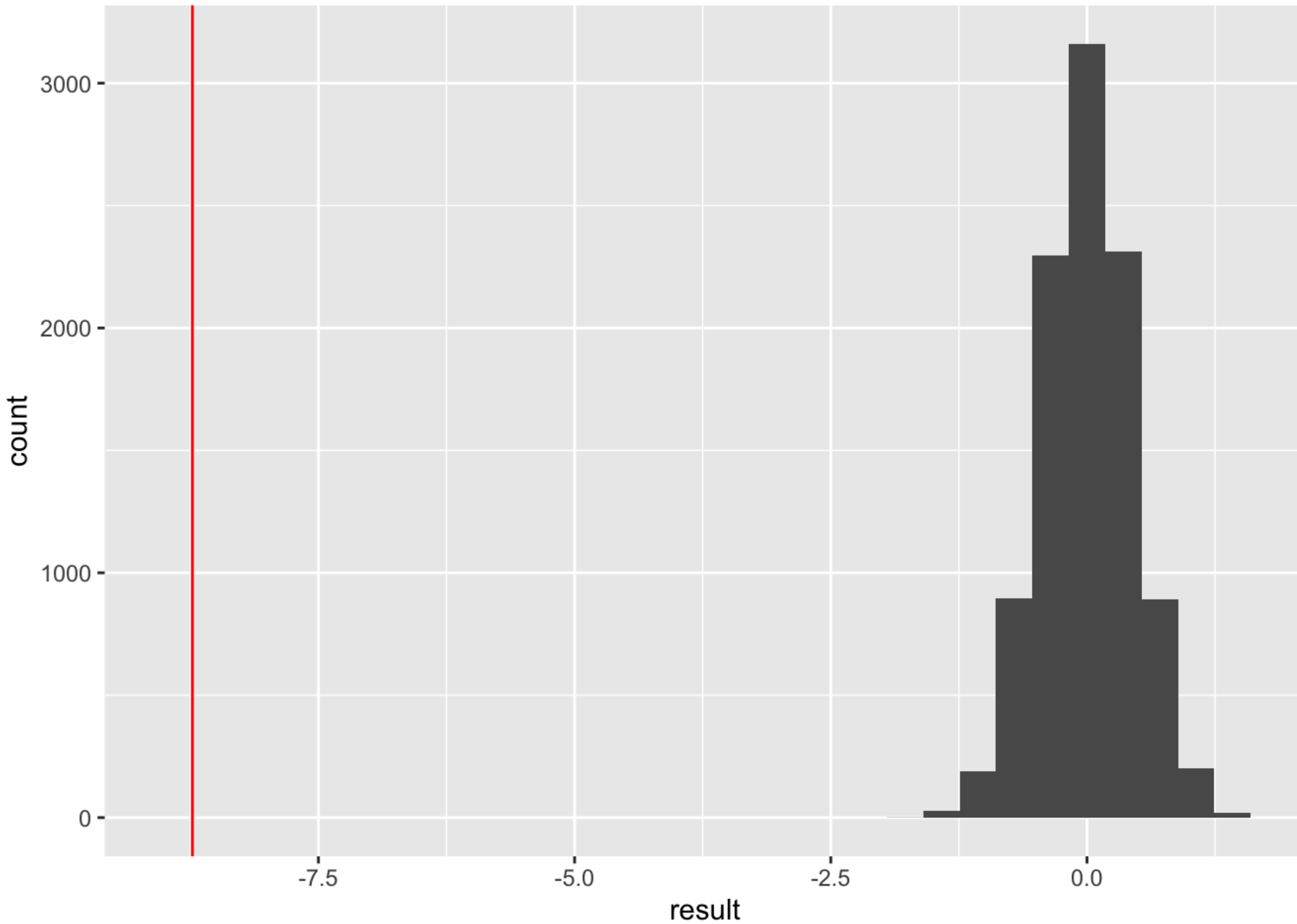

```
## The permutation for  morning  vs  afternoon : [1] 2e-04
```



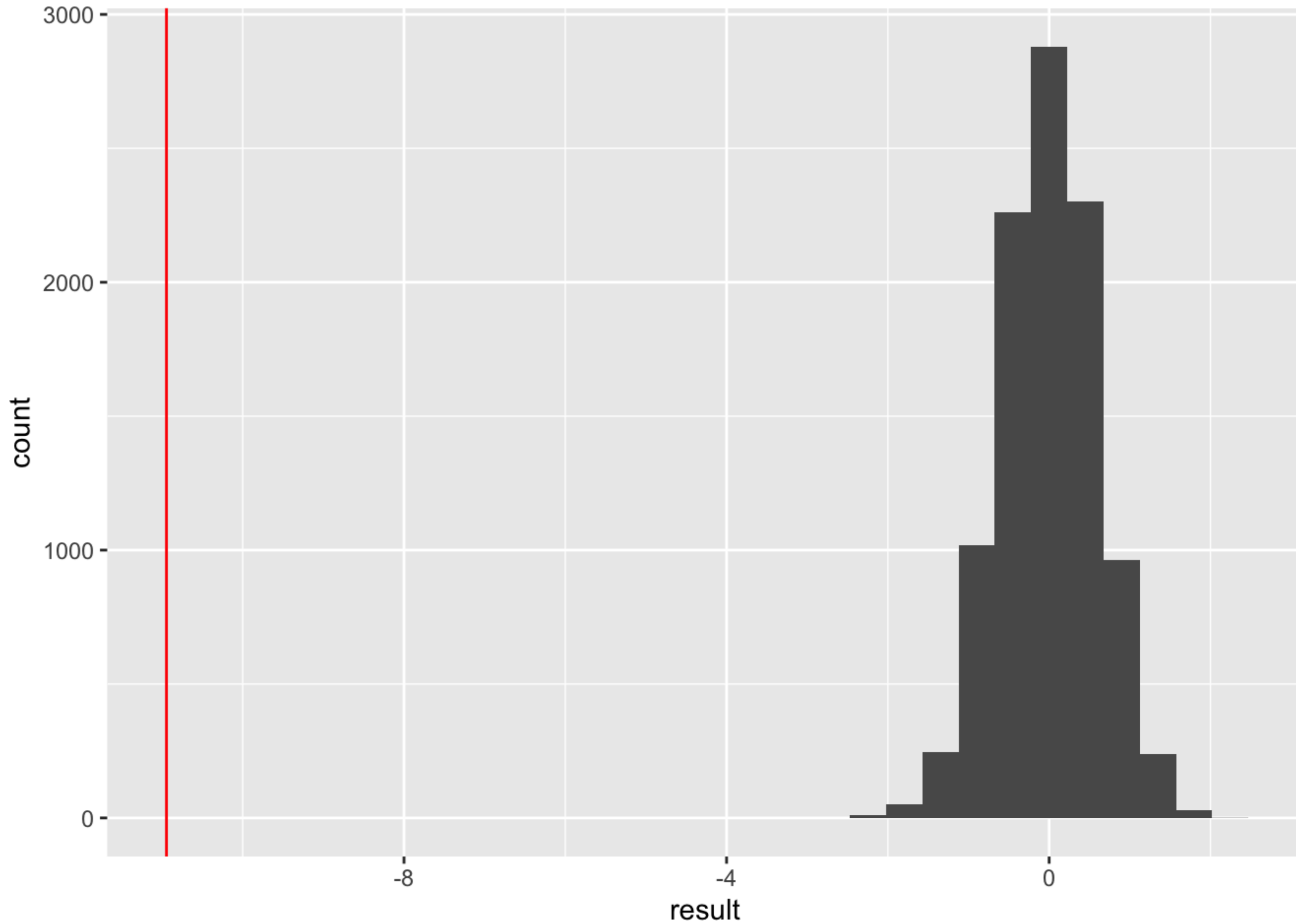

```
## The permutation for  morning  vs  evening : [1] 2e-04
```



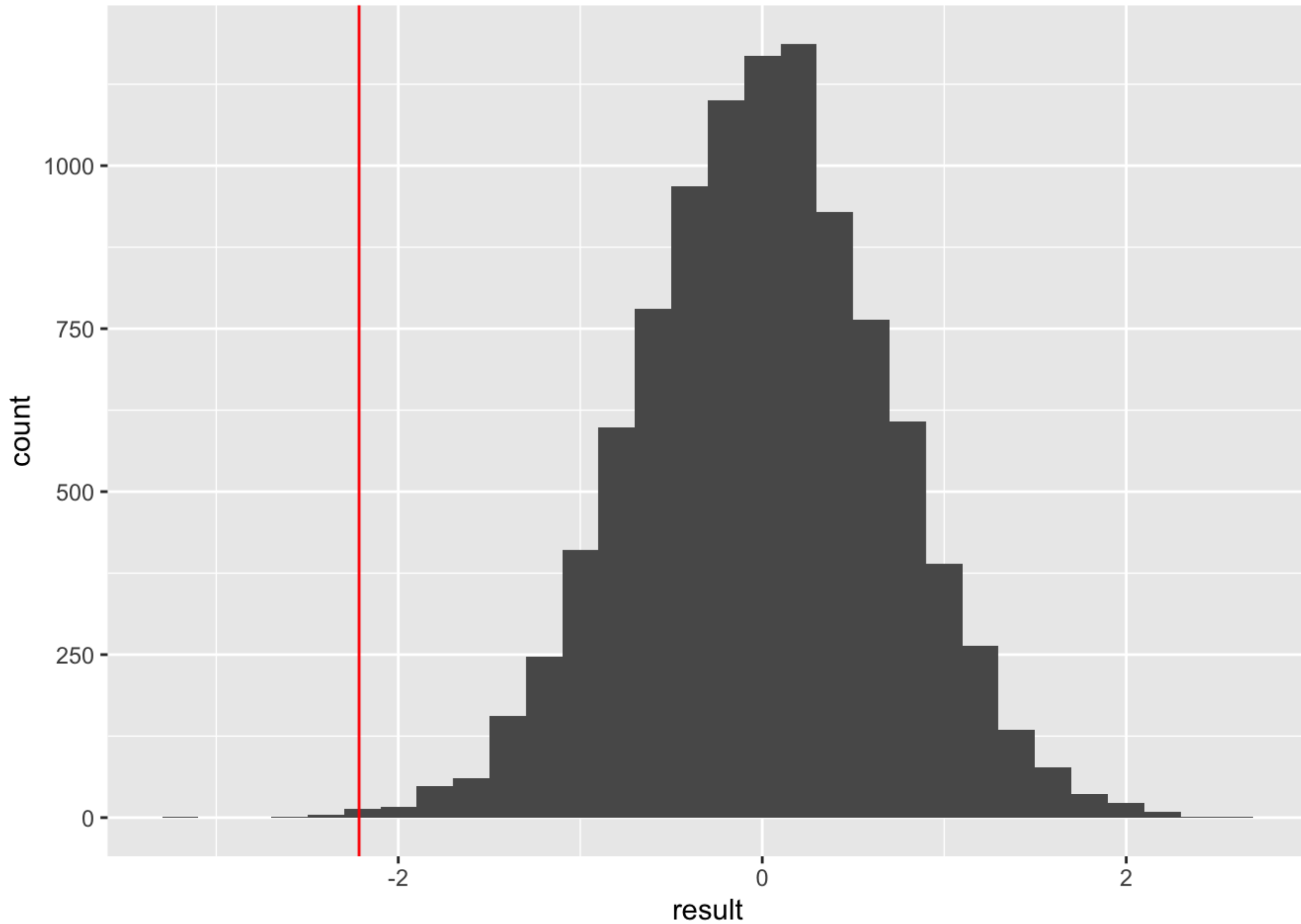

```
## The permutation for  morning  vs  night : [1] 2e-04
```




```
## The permutation for  afternoon  vs  evening : [1] 2e-04
```



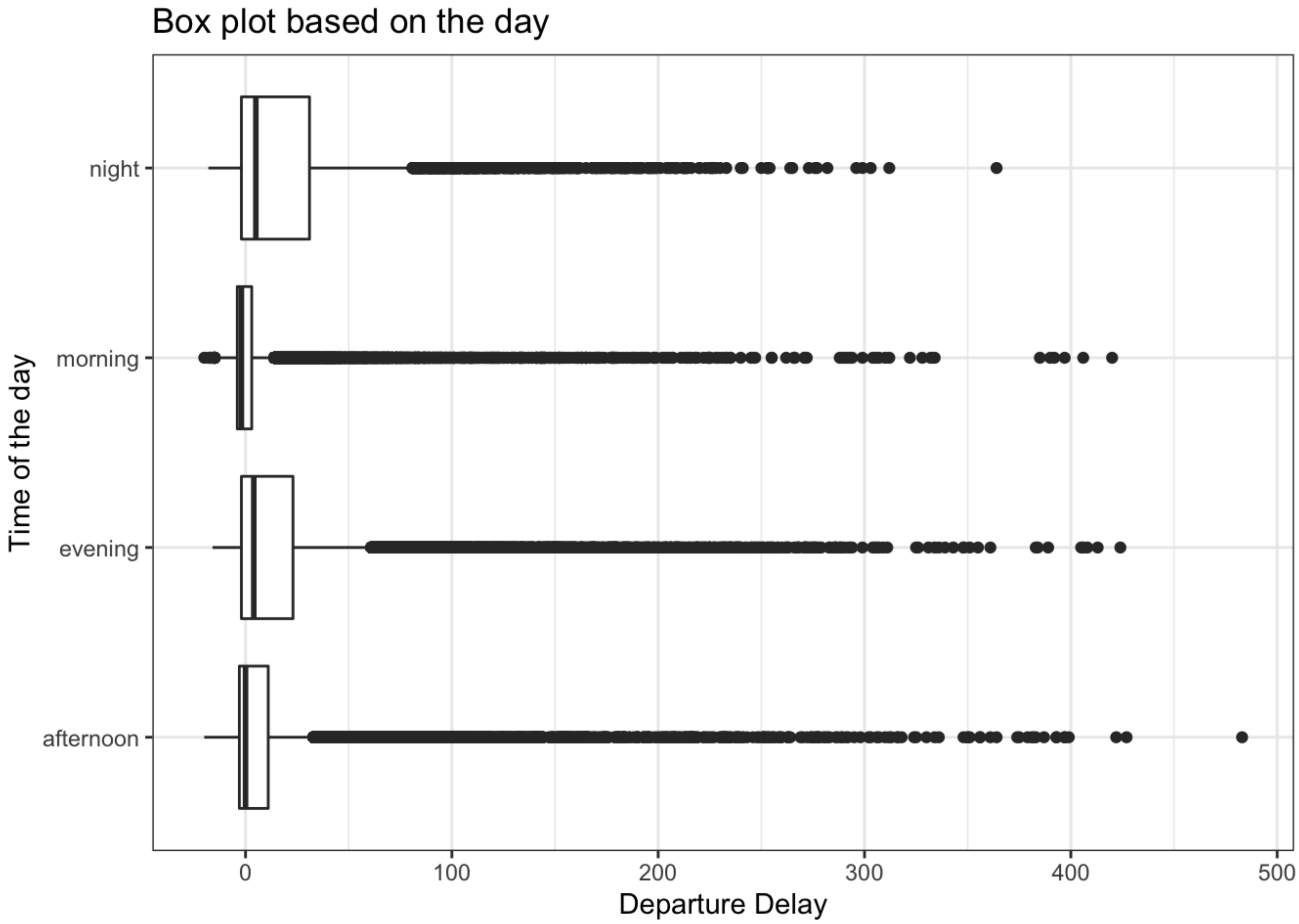

```
## The permutation for  afternoon  vs  night : [1] 2e-04
```




```
## The permutation for  evening  vs  night : [1] 0.003
```

```
ggplot(data= UA_flight_weather , aes(x = dep_delay, y = day_segment)) +
  geom_boxplot() +
  theme_bw() +
  labs(x = 'Departure Delay', title = 'Box plot based on the day',y='Time of the day')
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```




```
sum(is.na(UA_flight_weather$day))
```

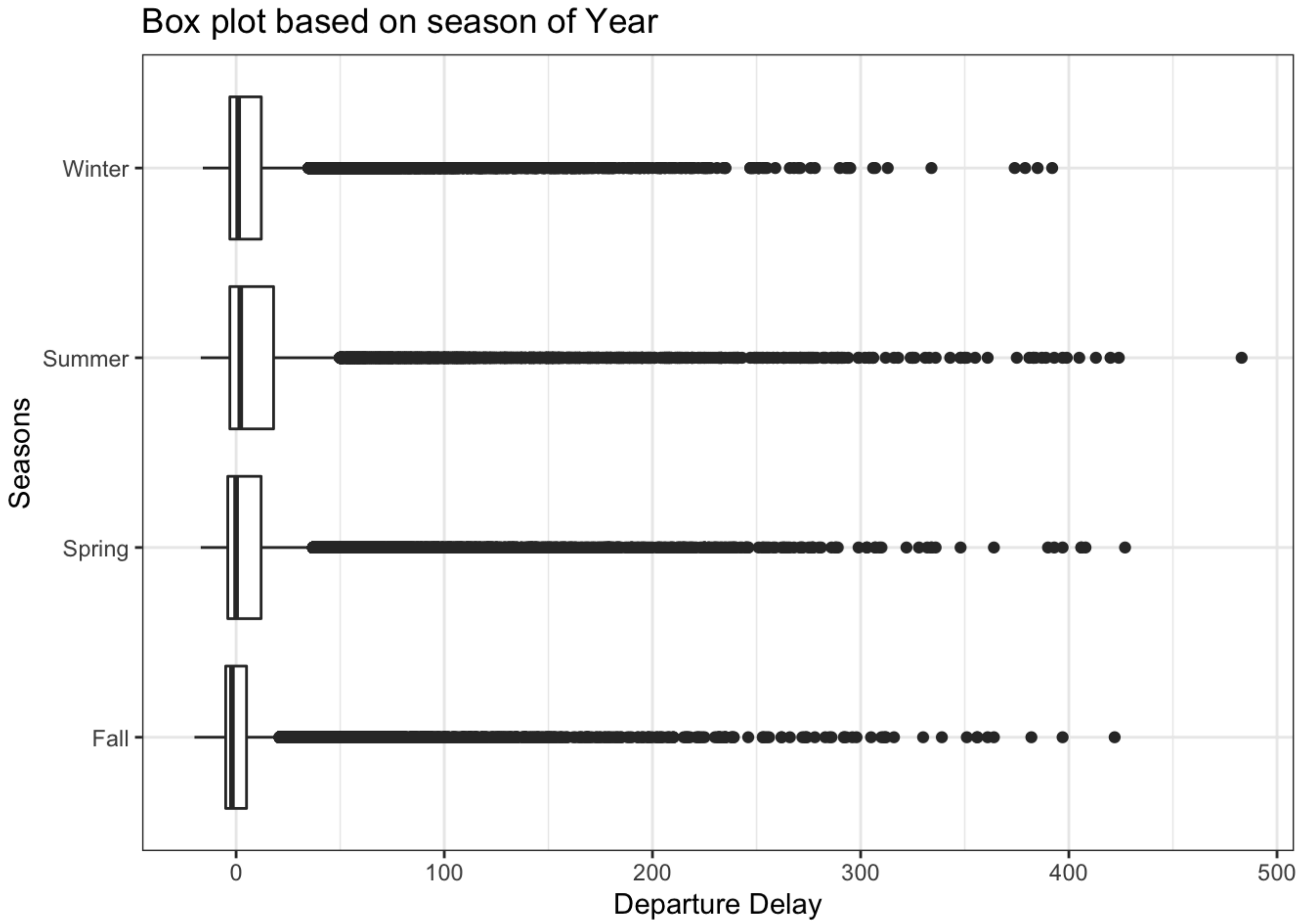
```
## [1] 0
```

```
#Add late and Very Late columns in the dataset
UA_flight_weather <- UA_flight_weather %>%
  mutate(month_segment = case_when(month >= 9 & month <= 11 ~ 'Fall',
                         month >= 3 & month <= 5 ~ 'Spring',
                         month >= 6 & month <= 8 ~ 'Summer',
                         month > 11 | month <3 ~ 'Winter')
         )
glimpse(UA_flight_weather)
```

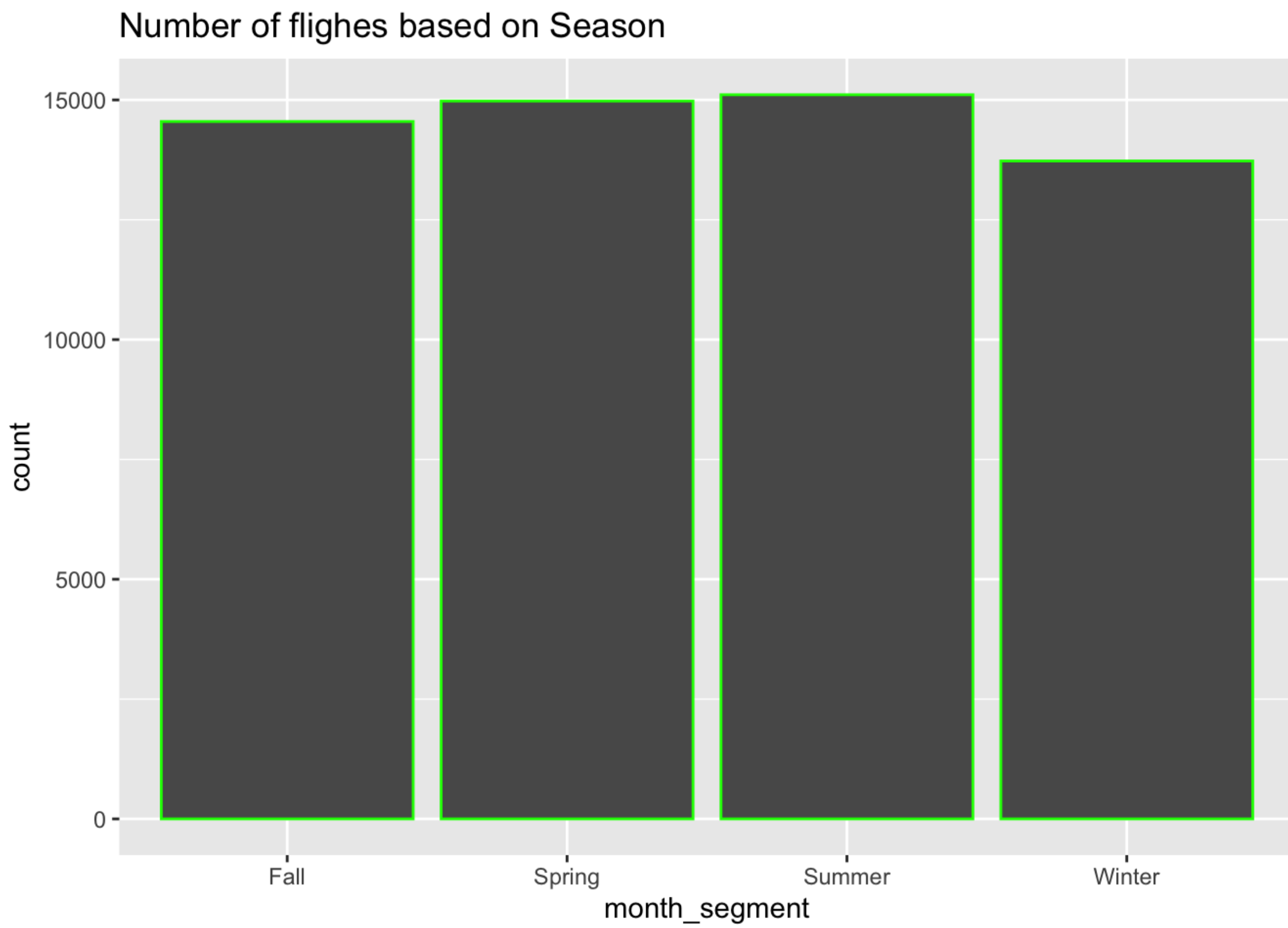
```
## Rows: 58,361
## Columns: 33
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ dep_time       <int> 517, 533, 554, 558, 558, 559, 607, 611, 623, 628, 643, …
## $ sched_dep_time <int> 515, 529, 558, 600, 600, 600, 607, 600, 627, 630, 646, …
## $ dep_delay      <dbl> 2, 4, -4, -2, -2, -1, 0, 11, -4, -2, -3, 8, 1, 1, -4, -…
## $ arr_time       <int> 830, 850, 740, 924, 923, 854, 858, 945, 933, 1016, 922,…
## $ sched_arr_time <int> 819, 830, 728, 917, 937, 902, 915, 931, 932, 947, 940, …
## $ arr_delay      <dbl> 11, 20, 12, 7, -14, -8, -17, 14, 1, 29, -18, -9, -6, -7…
## $ carrier        <chr> "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "UA", "…
## $ flight         <int> 1545, 1714, 1696, 194, 1124, 1187, 1077, 303, 496, 1665…
## $ tailnum        <chr> "N14228", "N24211", "N39463", "N29129", "N53441", "N765…
## $ origin         <chr> "EWR", "LGA", "EWR", "JFK", "EWR", "EWR", "EWR", "JFK",…
## $ dest           <chr> "IAH", "IAH", "ORD", "LAX", "SFO", "LAS", "MIA", "SFO",…
## $ air_time       <dbl> 227, 227, 150, 345, 361, 337, 157, 366, 229, 366, 146, …
## $ distance       <dbl> 1400, 1416, 719, 2475, 2565, 2227, 1085, 2586, 1416, 24…
## $ hour           <dbl> 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7…
## $ minute         <dbl> 15, 29, 58, 0, 0, 0, 7, 0, 27, 30, 46, 36, 45, 45, 0, 0…
## $ time_hour.x    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
## $ temp           <dbl> 39.02, 39.92, 39.02, 37.94, 37.94, 37.94, 37.94, 37.94,…
## $ dewp           <dbl> 28.04, 24.98, 28.04, 26.96, 28.04, 28.04, 28.04, 26.96,…
## $ humid          <dbl> 64.43, 54.81, 64.43, 64.29, 67.21, 67.21, 67.21, 64.29,…
## $ wind_dir       <dbl> 260, 250, 260, 260, 240, 240, 240, 260, 260, 240, 240, …
## $ wind_speed     <dbl> 12.65858, 14.96014, 12.65858, 13.80936, 11.50780, 11.50…
## $ wind_gust      <dbl> NA, 21.86482, NA, NA, NA, NA, NA, NA, 23.01560, NA, NA,…
## $ precip         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…
## $ pressure       <dbl> 1011.9, 1011.4, 1011.9, 1012.6, 1012.4, 1012.4, 1012.4,…
## $ visib          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…
## $ time_hour.y    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…
## $ late           <lgl> TRUE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, FA…
## $ very_late      <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,…
## $ day_segment    <chr> "morning", "morning", "morning", "morning", "morning", …
## $ month_segment  <chr> "Winter", "Winter", "Winter", "Winter", "Winter", "Wint…
```

```
ggplot(data= UA_flight_weather , aes(x = dep_delay, y = month_segment)) +
  geom_boxplot() +
  theme_bw() +
  labs(x = 'Departure Delay', title = 'Box plot based on season of Year',y='Seasons')
```

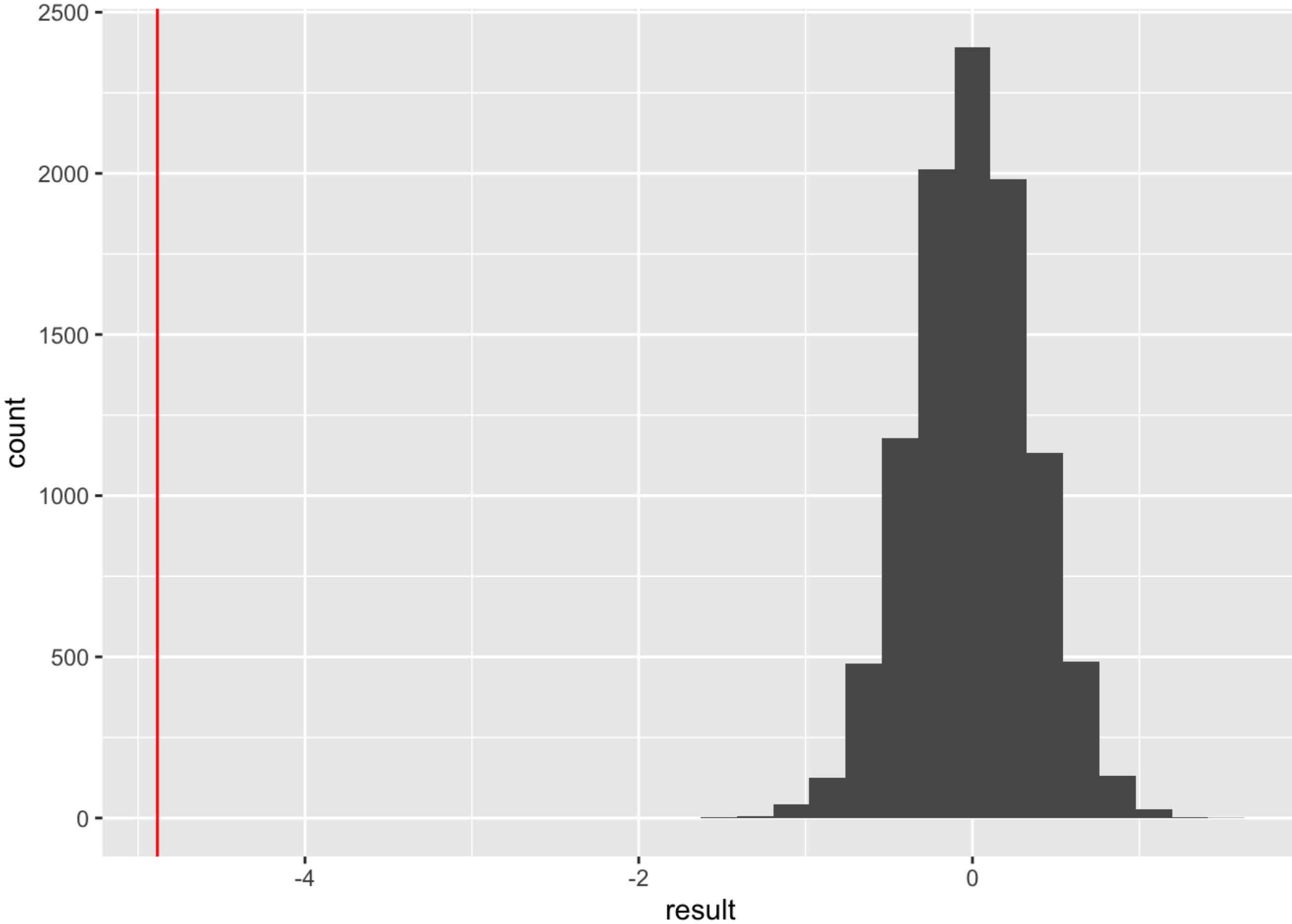
```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```



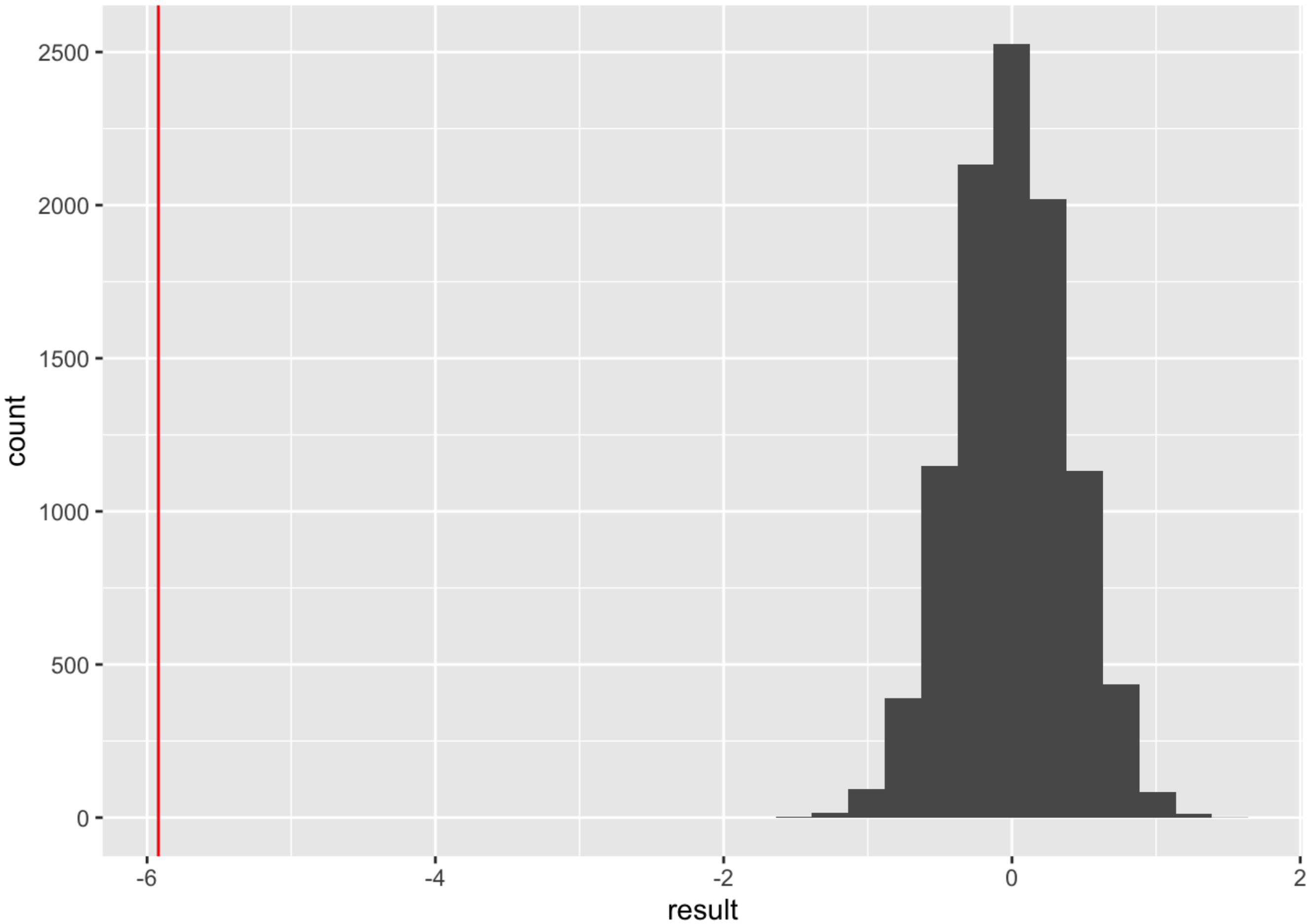

```r
# Create bar plot
ggplot(data = UA_flight_weather , aes(x= month_segment))+
  geom_bar(color = 'green') +
  ggtitle('Number of flighes based on Season')
```


Number of flighes based on Season
count
15000
10000
5000
0
Fall
Spring
Summer
Winter
month_segment

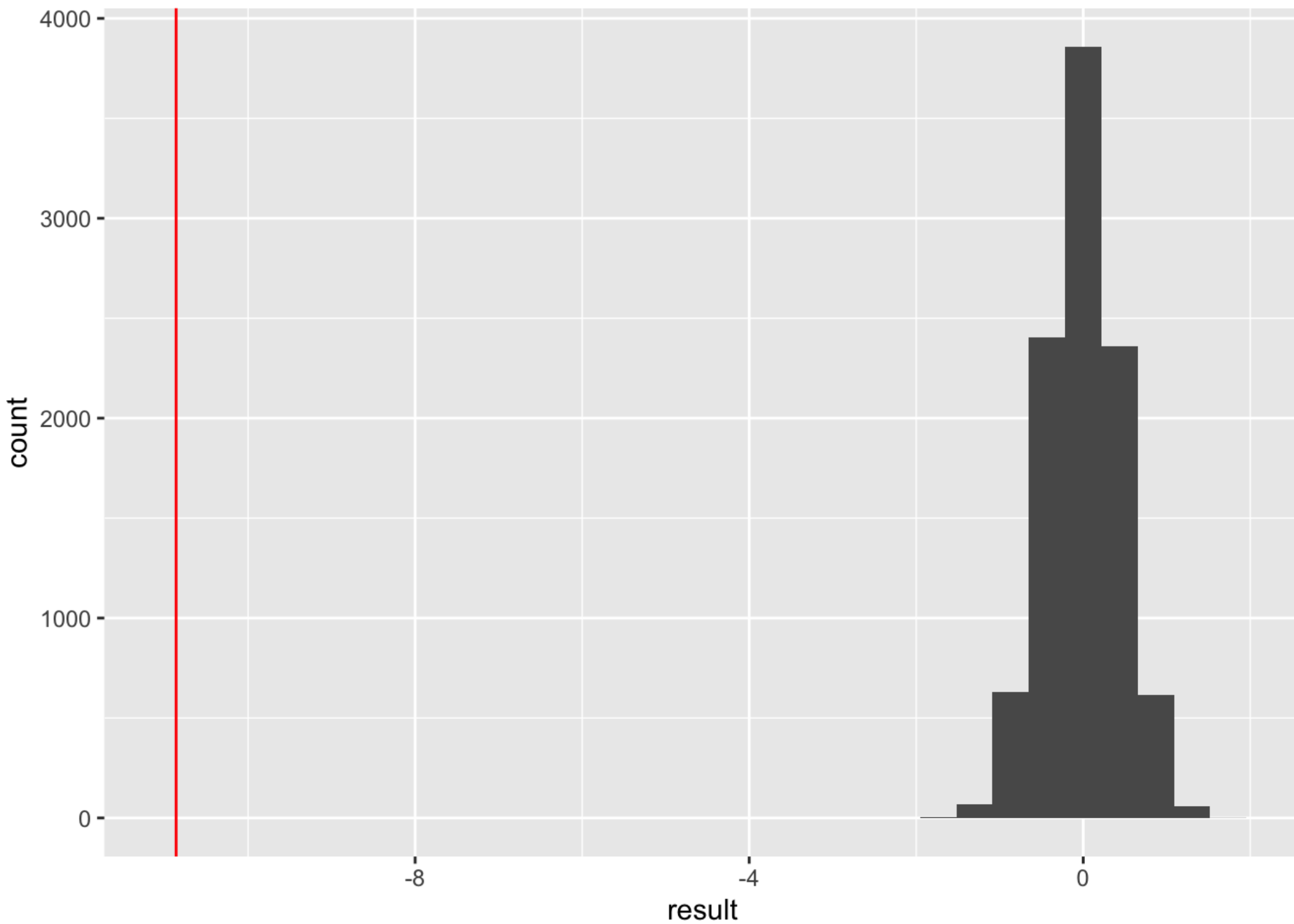
```
#N = number of simulations we will use
N <- 10^4-1
#create a blank vector to store the simulation results
result <- numeric(N)
#vector of the types of a day
vectorSeason = c("Fall", "Winter", "Spring", "Summer")
#loop through the types of a day and choose every time two of those
#calculate and store the observed difference in the sample
for(i in 1:length(vectorSeason))
{
  for(j in 1:length(vectorSeason)){
    if(j < 4 & i <= j){
      column1 = (vectorSeason[i])
    column2 = (vectorSeason[j+1])
    #reduce the data set to selected two seasons of a year
    reduced_flights <- UA_flight_weather %>%
    filter(month_segment==column1 | month_segment==column2)
    #sample.size = the number of observations in our sample
    sample.size = nrow(reduced_flights)
    #group.1.size = the number of observations in the first group
    group.1.size = nrow(reduced_flights[reduced_flights$month_segment==column1,])
    #calculate the observed value
    observed <- mean(reduced_flights$dep_delay[reduced_flights$month_segment ==column1])
-
    mean(reduced_flights$dep_delay[reduced_flights$month_segment == column2])

    for(k in 1:N)
    {
      index = sample(sample.size, size=group.1.size, replace = FALSE)
      result[k] = mean(reduced_flights$dep_delay[index])-
      mean(reduced_flights$dep_delay[-index])
    }

    print(ggplot(data=tibble(result), mapping = aes(x=result)) + geom_histogram(bins = 3
0) + geom_vline(xintercept = observed, color = "red"))

    if(observed > 0)
      {
        cat("The permutation for ", column1, " vs ", column2, ": ")
        print(p_value <- 2 * (sum(result >= observed) + 1) / (N + 1))
      }
    else{
      cat("The permutation for ", column1, " vs ", column2, ": ")
      print(p_value <- 2 * (sum(result <= observed) + 1) / (N + 1))
    }
    }
}}
```



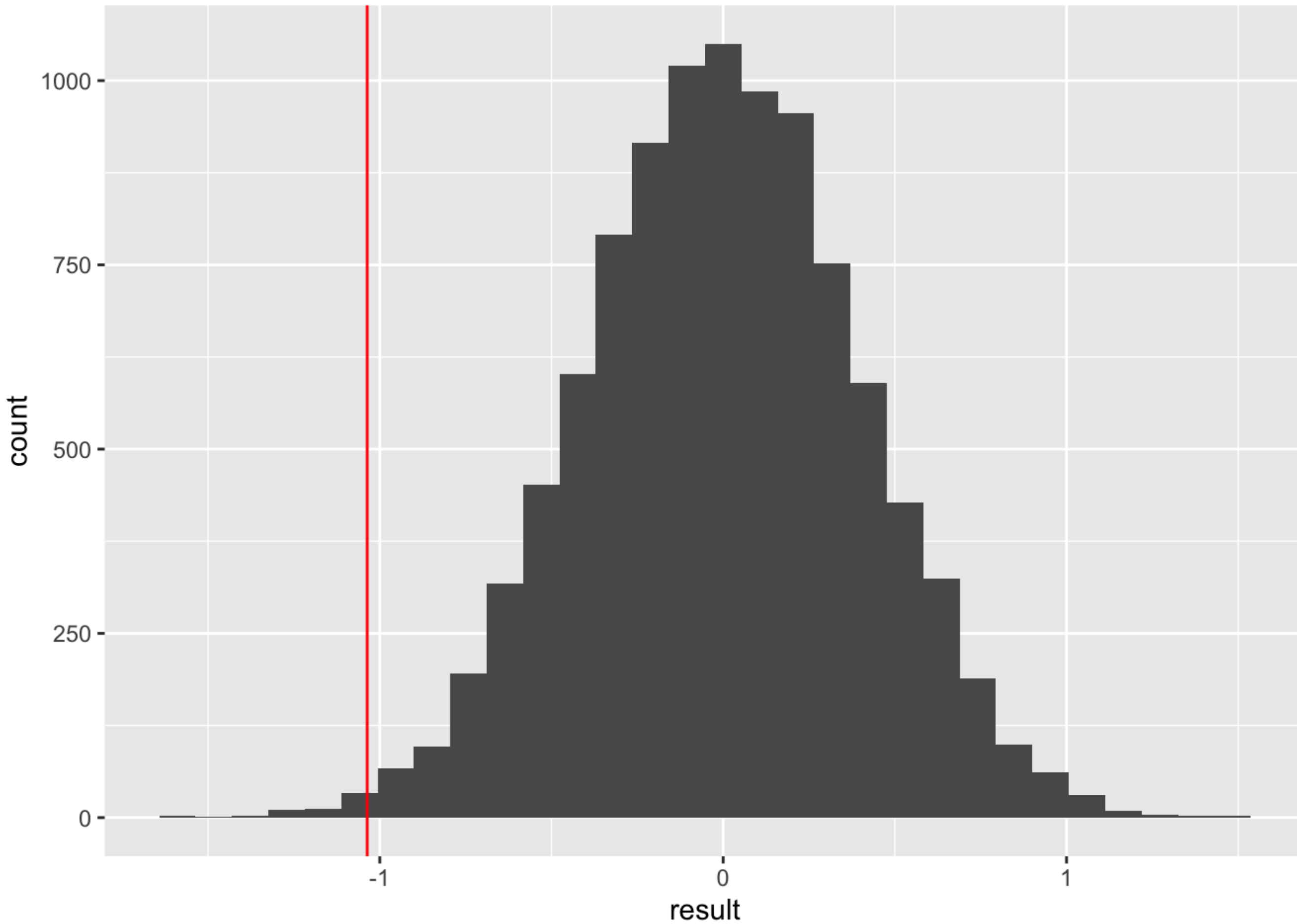

```
## The permutation for  Fall  vs  Winter : [1] 2e-04
```



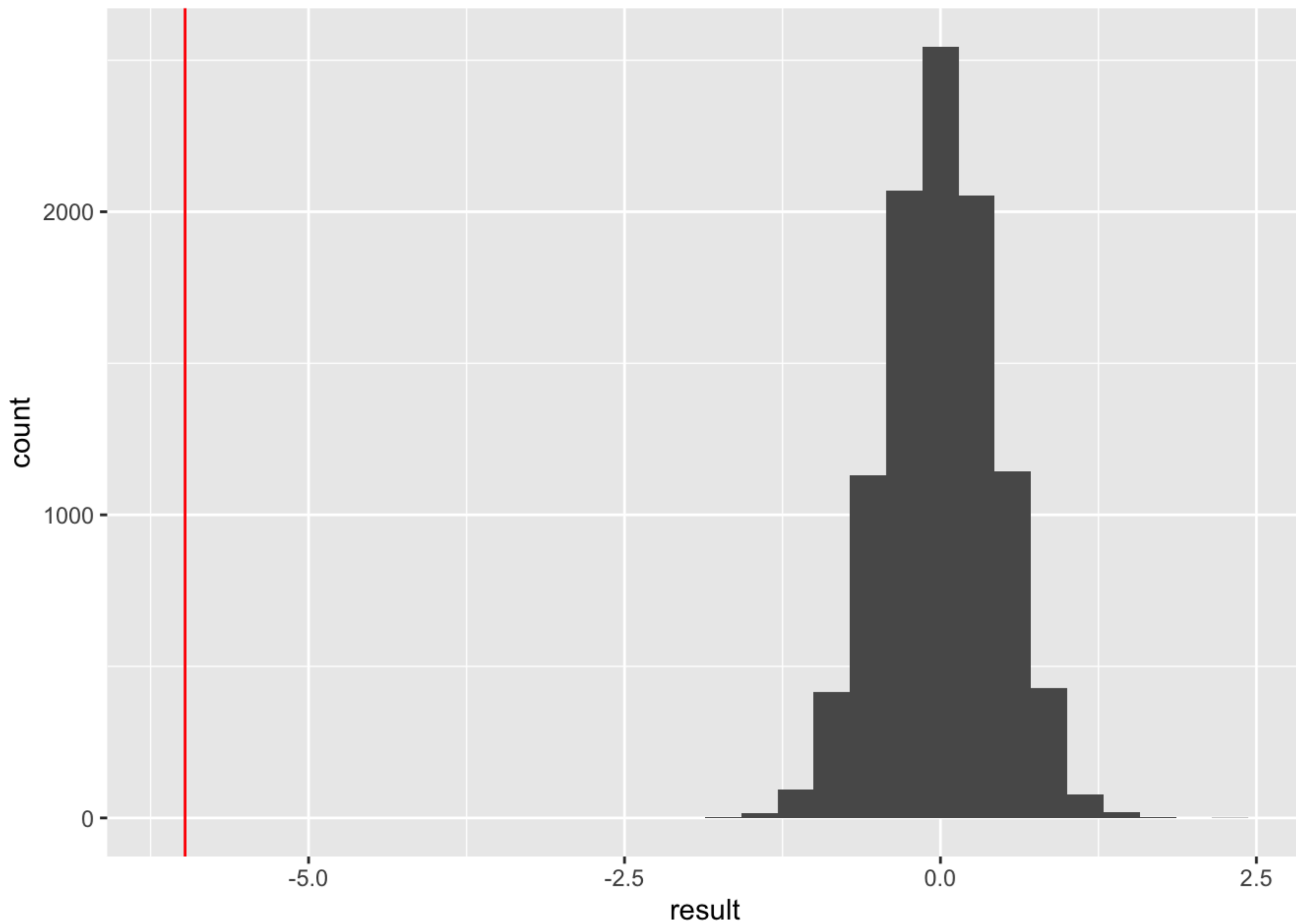

```
## The permutation for  Fall  vs  Spring : [1] 2e-04
```



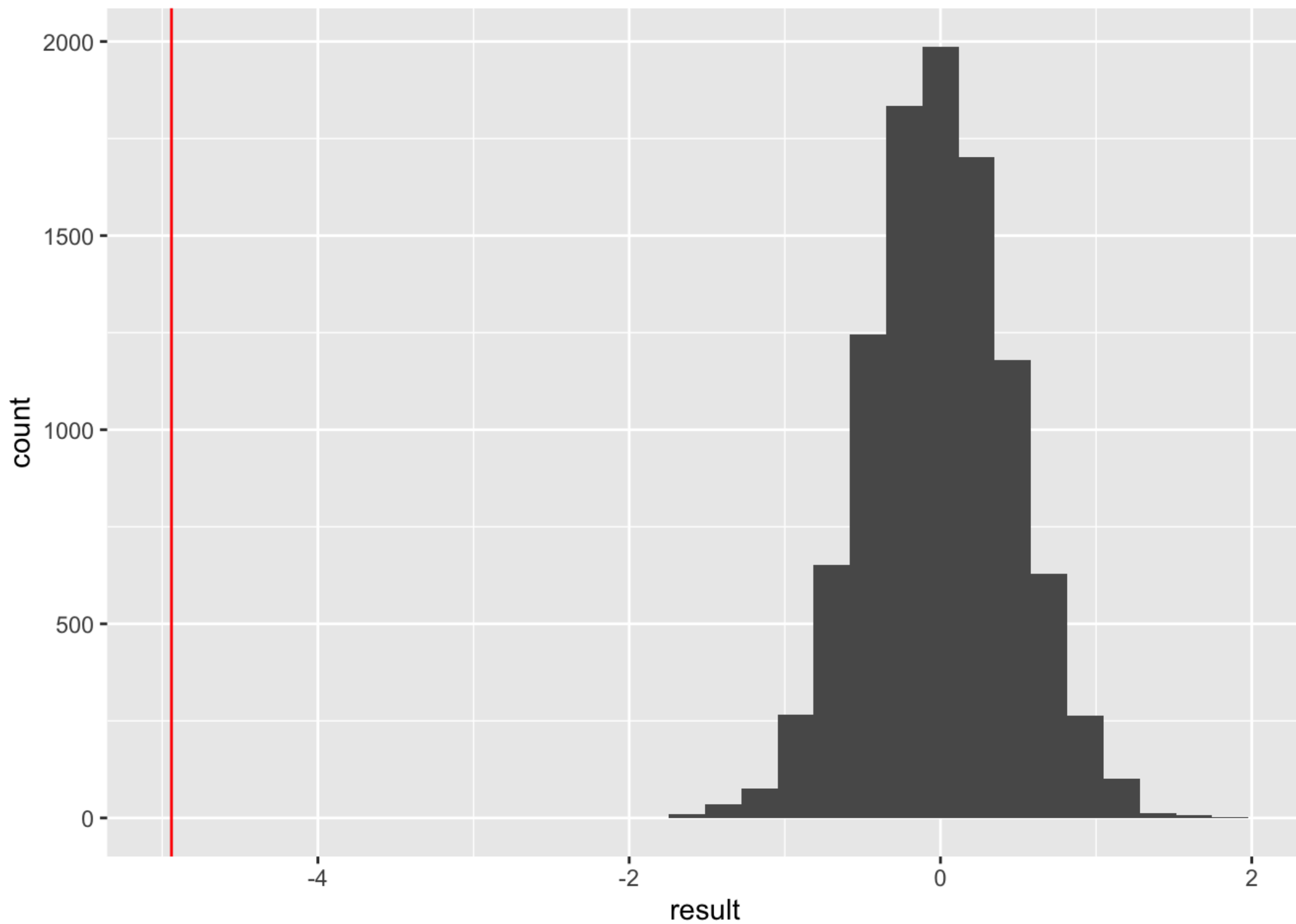

```
## The permutation for  Fall  vs  Summer : [1] 2e-04
```




```
## The permutation for  Winter  vs  Spring : [1] 0.0104
```




```
## The permutation for  Winter  vs  Summer : [1] 2e-04
```




```
## The permutation for  Spring  vs  Summer : [1] 2e-04
```

#Let’s analyze the temperature variable

```
summary(UA_flight_weather$temp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   10.94   42.08   57.92   57.30   73.04  100.04       7
```

Note : Temperature is in Fahrenheit Minimum temperature : 10.94 Maximum temperature : 100.04

```
cat('Number of records where temperature value is missing' , sum(is.na(UA_flight_weather
$temp)),'\n')
```

```
## Number of records where temperature value is missing 7
```

```
cat('Percentage of missing data for temperature for the UA carrier' ,sum((is.na(UA_fligh
t_weather$temp))/nrow(UA_flight_weather))*100,'\n')
```

```
## Percentage of missing data for temperature for the UA carrier 0.01199431
```

```
tab <- matrix(c(sum(is.na(UA_flight_weather$temp)),sum((is.na(UA_flight_weather$temp))/n
row(UA_flight_weather))*100), ncol=2, byrow=TRUE)
colnames(tab) <- c('Null values in dataset','Percentage of null values')

kable(tab) %>%
  kable_styling()
```

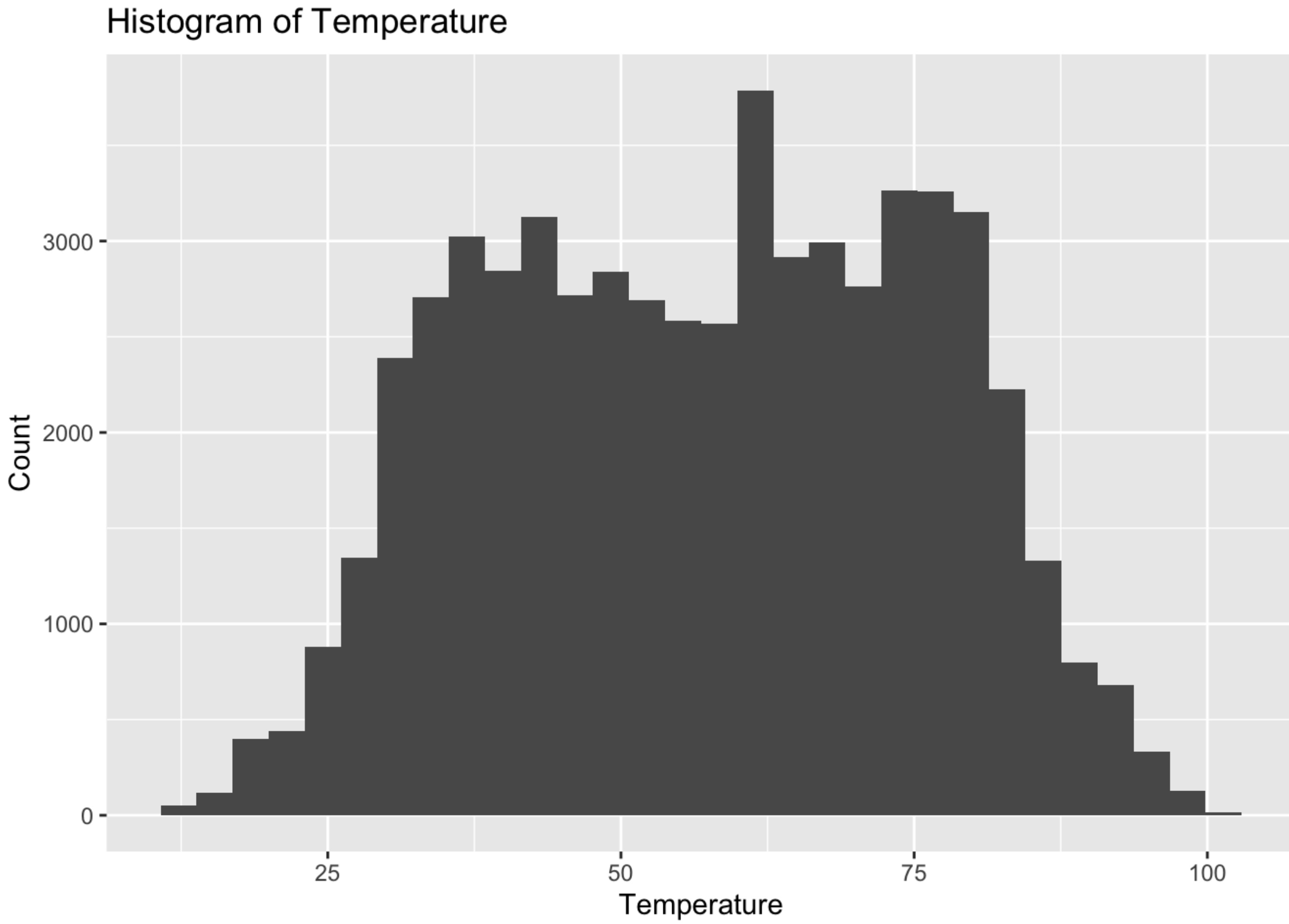
Null values in dataset	Percentage of null values
7	0.0119943

```
# Impute missing values with mean in temperature  column
UA_flight_weather$temp <- with(UA_flight_weather, impute(temp, mean))
```

```
ggplot(data = UA_flight_weather , mapping = aes(x = temp)) +
  geom_histogram()+
  labs(title = 'Histogram of Temperature',x = 'Temperature' , y = 'Count')
```

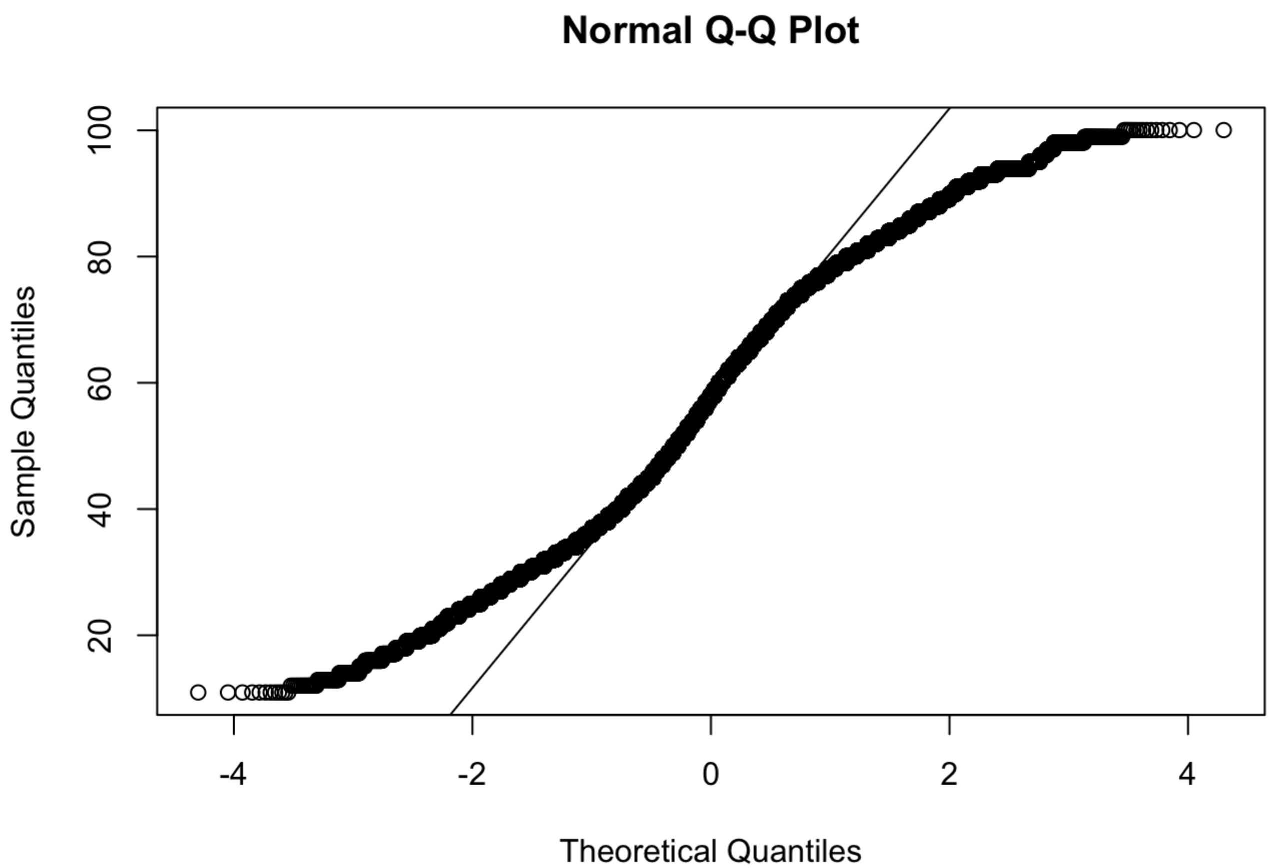
```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```

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



Let’s check if the temperature in the dataset follows a Normal distribution

```
#Create a q-q plot between the difference
qqnorm(UA_flight_weather$temp)
qqline(UA_flight_weather$temp)
```



The

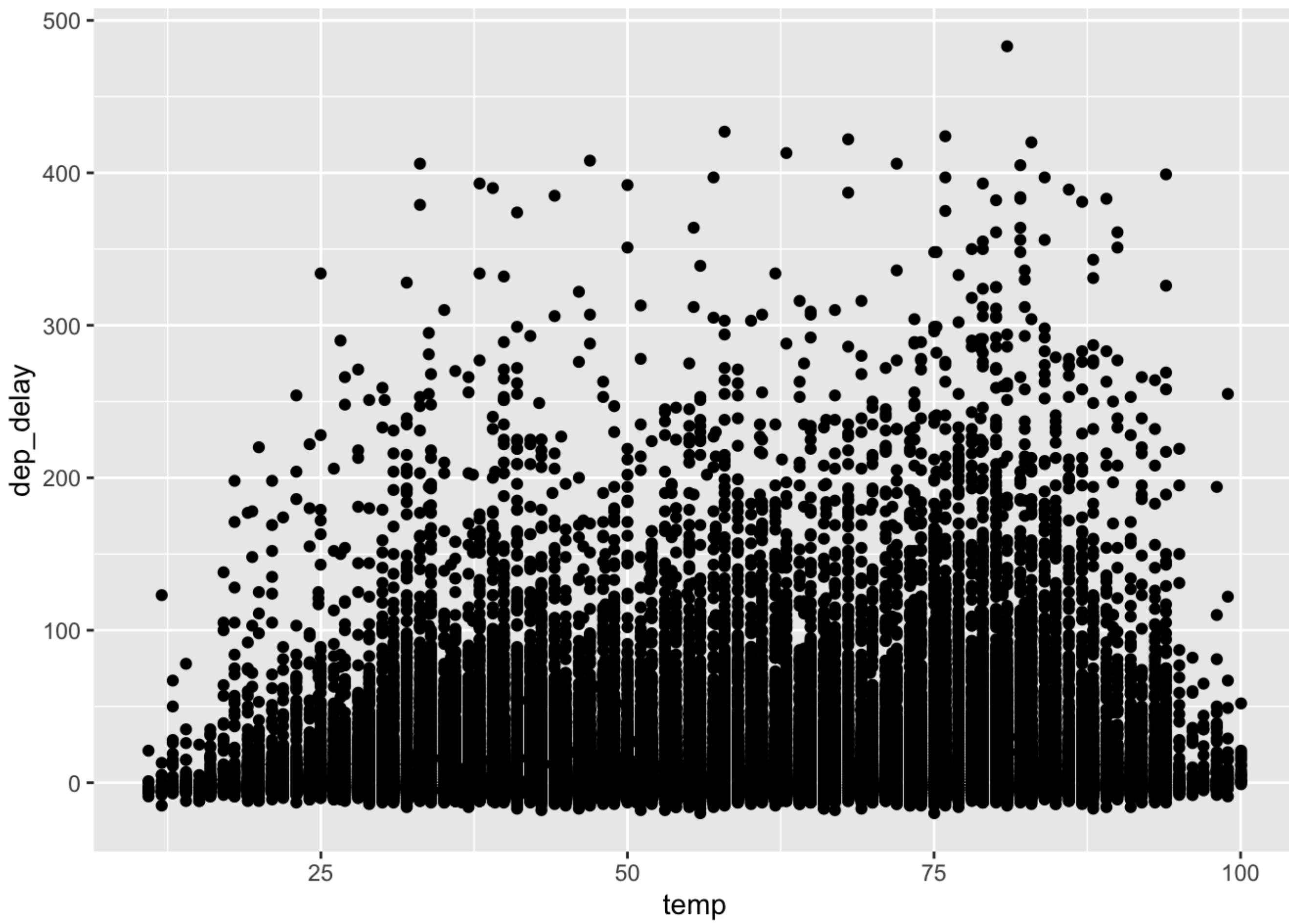
temperature follows a Normal Distribution.

Let’s analyse the departure delay based on the temperature:

Data type of temperature variable: double Data type of departure delay : double

```
ggplot(data = UA_flight_weather , aes( x = temp , y = dep_delay))+
  geom_point()
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```



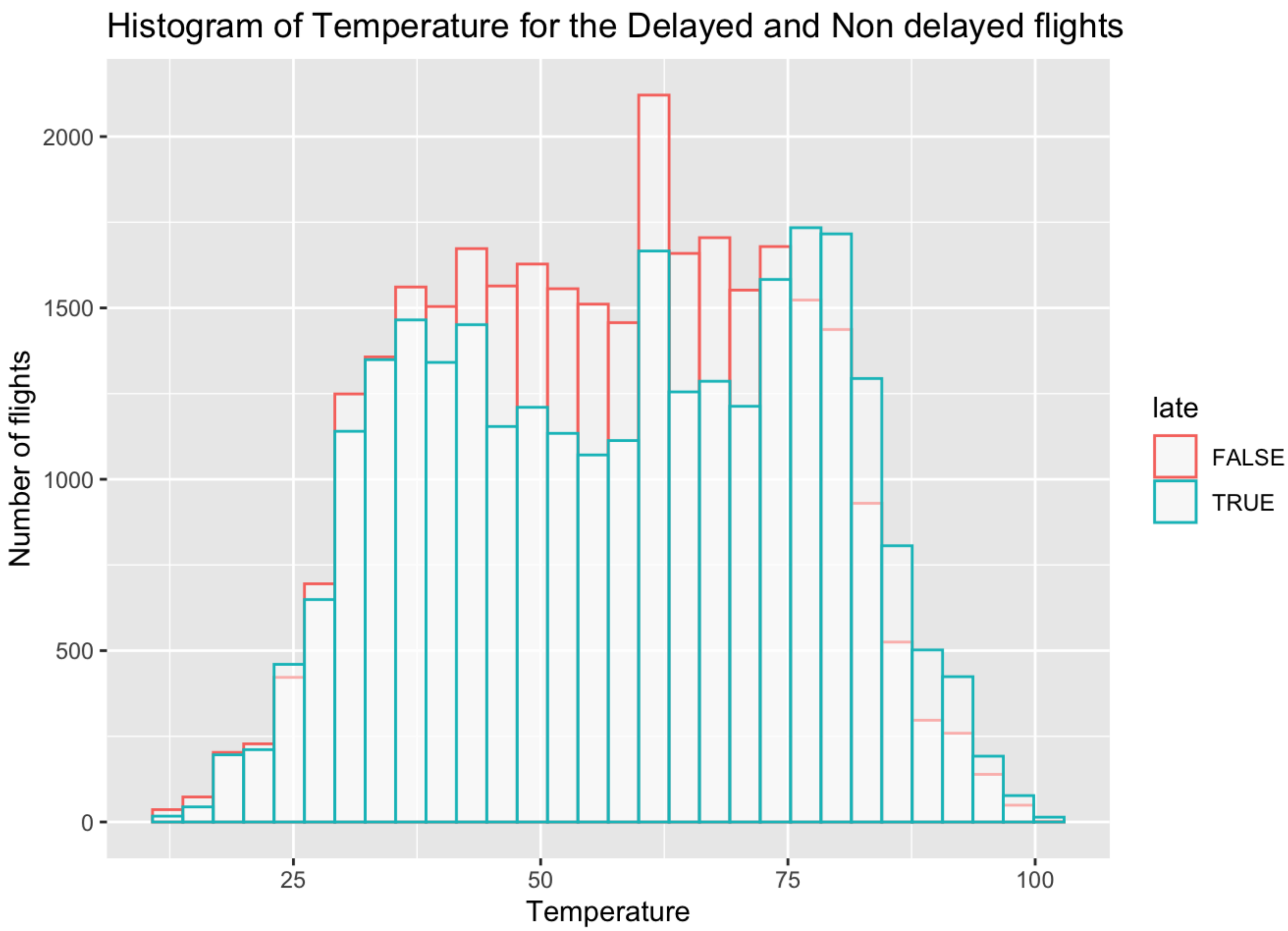
We can’t conclude much from this graph.

It’s interesting to know which to compare the temperature for the flights which have dep_delay > 0 Note: We have already filtered the data based on the dep_delay and that variable in our dataset is called as Late.

```
ggplot(data = UA_flight_weather , mapping = aes(x = temp,color = late)) +
  geom_histogram(fill="white", alpha=0.5, position="identity")+
  labs(title = 'Histogram of Temperature for the Delayed and Non delayed flights',x = 'T
emperature' , y = 'Number of flights')
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```

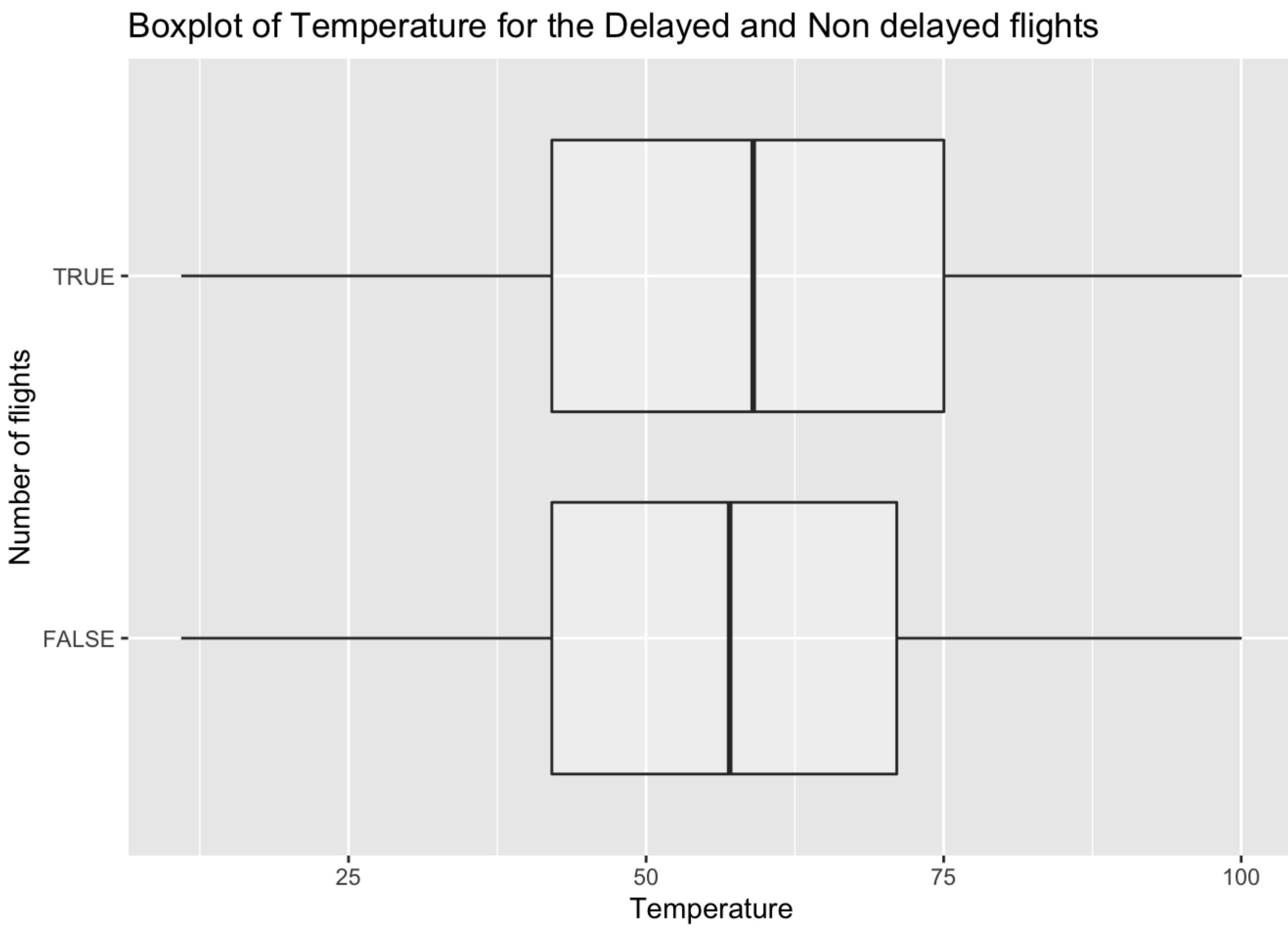
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Temperature for both the late and non delay flight is following the same distribution. Both the graphs are following overlapping.

```
ggplot(data= subset(UA_flight_weather, !is.na(late)) , aes(x = temp, y = late)) +
  geom_boxplot( alpha=0.3) +
  labs(title = 'Boxplot of Temperature for the Delayed and Non delayed flights',x = 'Tem
perature' , y = 'Number of flights')
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```



By

seeing the box plot of two graphs we can see that there’s not much difference between the Flights which were delayed and which were on time based on the temperature variable.

Both the box plots are overlapping. We can conduct a permutation test and see if there’s any relationship.

Question : Is the mean of temperature of flights for Delayed and Non - Delayed is equal or not?

H0 : Mean(Temp of flights which were delayed) = Mean(Temp of flights which were on time) H1 : Mean(Temp of flights which were delayed) != Mean(Temp of flights which were on time)

Let’s do a permutation test and compare the mean values between both the values.

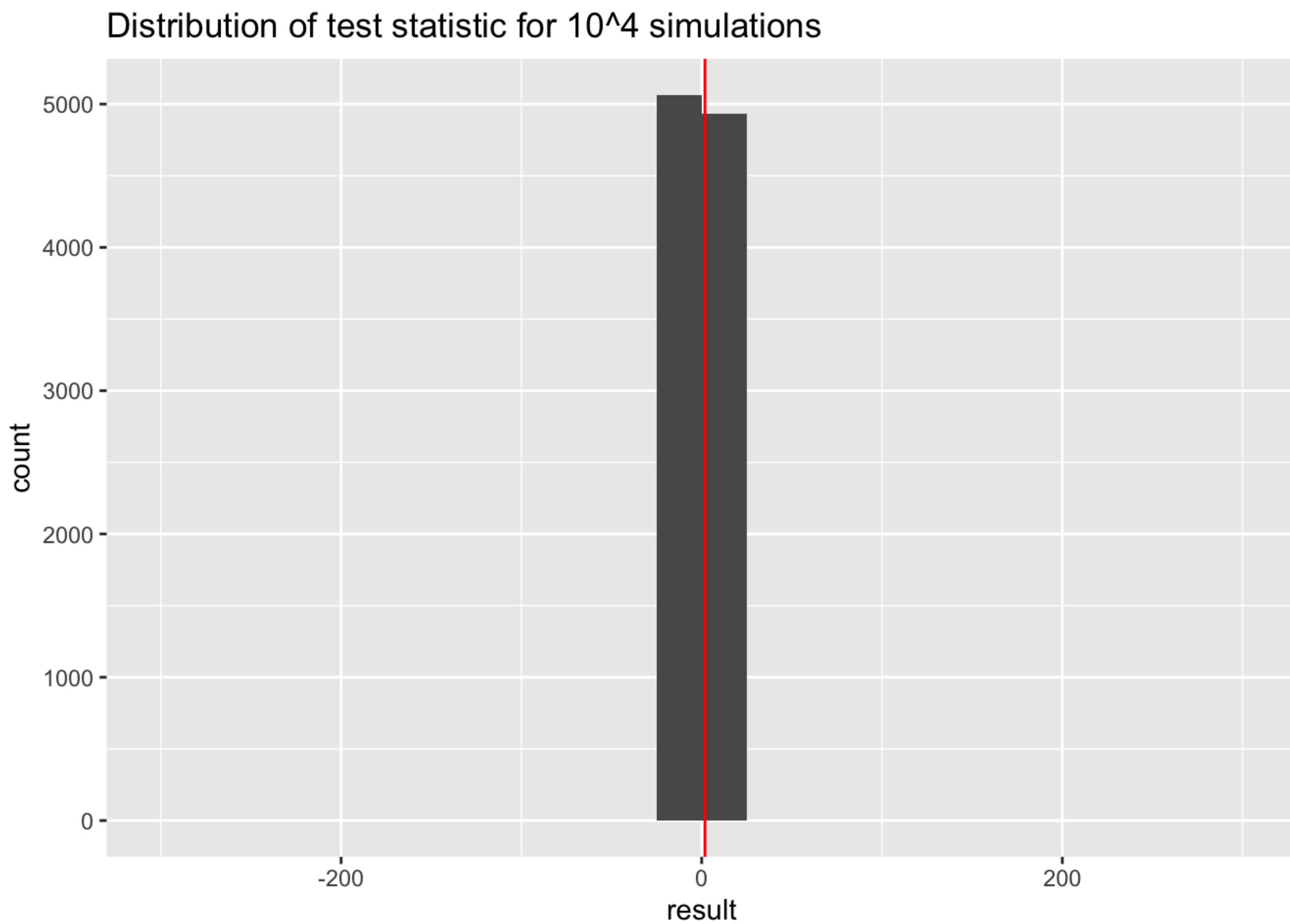
```
#Find the observed difference between flight delays
observed_diff = mean(UA_flight_weather$temp[UA_flight_weather$late == TRUE]) -      mean
(UA_flight_weather$temp[UA_flight_weather$late == FALSE])
print(observed_diff)
```

```
## [1] 1.83431
```

```
# Number of simulation we will use
N <- 10^4-1
#sample.size = the number of observations in our sample
sample.size = nrow(UA_flight_weather)
#group.1.size = the number of observations in the first group : Flights were delayed
group.1.size = length(UA_flight_weather$late[UA_flight_weather$late == TRUE])
print(group.1.size)
```

```
## [1] 27767
```

```
#create a blank vector to store the simulation results
result <- numeric(N)
#use a for loop to cycle through values of i ranging from 1 to N
for(i in 1:N)
{
  #each iteration, randomly sample index values
  #sample.size gives the total number of index values to sample from
  #group.1.size gives the number of index values to sample
  #sample without replacement
  #indexes sampled will be treated as the "TRUE" group, indexes not sample as "FALSE"
  index = sample(sample.size, size=group.1.size, replace = FALSE)

  #calculate and store the difference in
  #median rainfall between the index and non-index groups
  result[i] = mean(UA_flight_weather$temp[index]) - mean(UA_flight_weather$temp[-index])
}

#plot a histogram of the simulated differences
#add a vertical line at the observed difference
ggplot(data=tibble(result), mapping = aes(x=result)) +
  geom_histogram(breaks=seq(-300,300,by=25)) +
  geom_vline(xintercept = observed_diff, color = "red") +
  ggtitle('Distribution of test statistic for 10^4 simulations')
```




```
#Calculate the p-value
p_value <- 2*(sum(result >= observed_diff) + 1) / (N + 1)
p_value
```

```
## [1] 2e-04
```

Observations from the permutation test: 1. The p-value is very small. It means that we can reject our null hypothesis. That is the mean of both the flights which were delayed and on-time is not equal. There's a evidence that the alternate hypothesis can be true. We meed to investigate more about it.

It means that there's a possibility that the mean temperature will be different for the flight which were delayed and which were on time.

Let's try to compare the variance of both the variables.

Question : Is the variance of temperature of flights for delayed and non delayed flights is equal or not?

H0 : var(Temp of flights which were delayed) = var(Temp of flights which were on time) H1 : var(Temp of flights which were delayed) != var(Temp of flights which were on time)

Let's do a permutation test and compare the variance values between both the values.

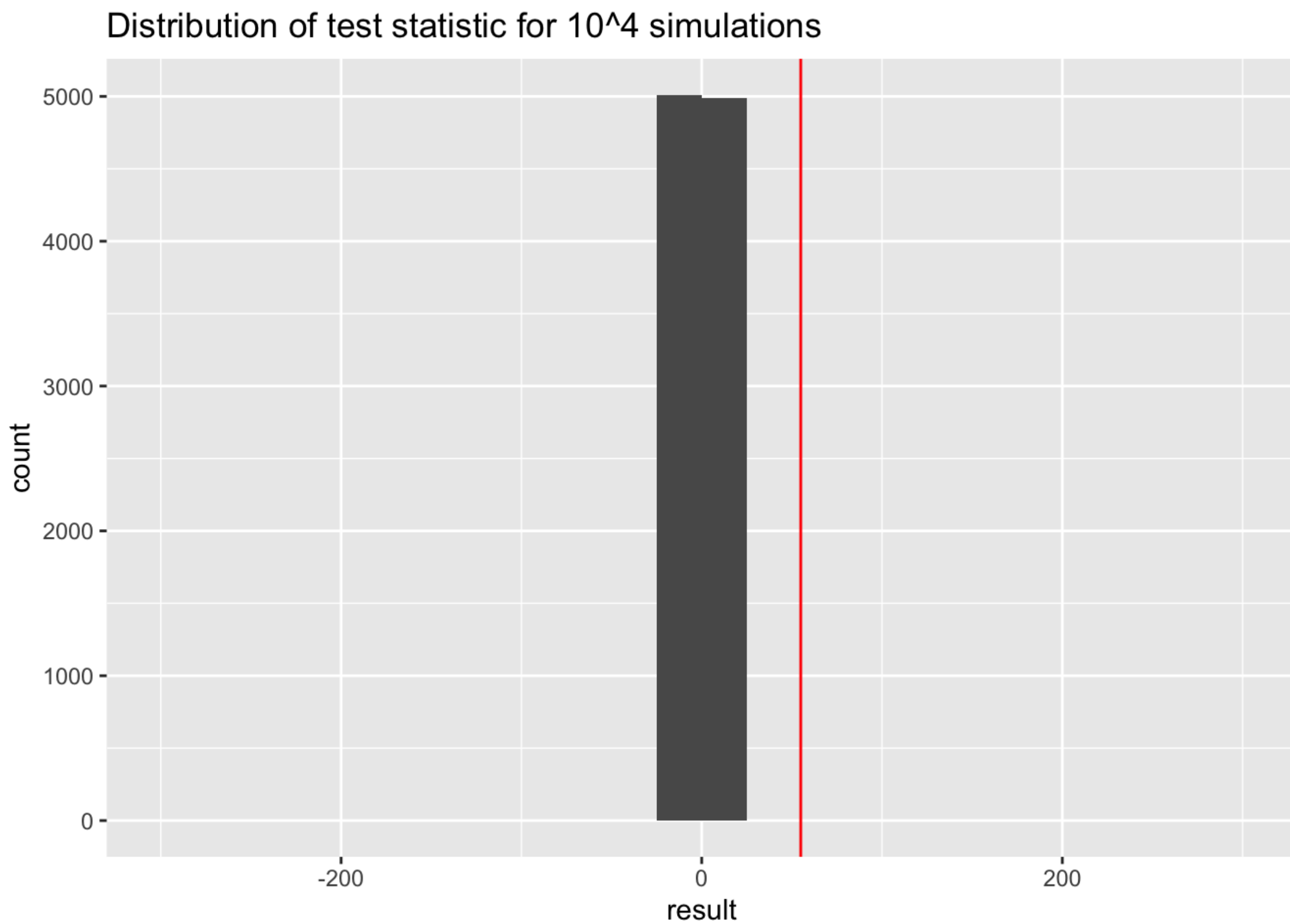
```r
#Find the observed difference between flight delays
observed_diff = var(UA_flight_weather$temp[UA_flight_weather$late == TRUE]) -      var(U
A_flight_weather$temp[UA_flight_weather$late == FALSE])
print(observed_diff)
```

```
## [1] 54.92397
```

```r
# Number of simulation we will use
N <- 10^4-1
#sample.size = the number of observations in our sample
sample.size = nrow(UA_flight_weather)
#group.1.size = the number of observations in the first group : Flights were delayed
group.1.size = length(UA_flight_weather$late[UA_flight_weather$late == TRUE])
print(group.1.size)
```

```
## [1] 27767
```

```r
#create a blank vector to store the simulation results
result <- numeric(N)
#use a for loop to cycle through values of i ranging from 1 to N
for(i in 1:N)
{
  #each iteration, randomly sample index values
  #sample.size gives the total number of index values to sample from
  #group.1.size gives the number of index values to sample
  #sample without replacement
  #indexes sampled will be treated as the "TRUE" group, indexes not sample as "FALSE"
  index = sample(sample.size, size=group.1.size, replace = FALSE)

  #calculate and store the difference in
  #median rainfall between the index and non-index groups
  result[i] = var(UA_flight_weather$temp[index]) - var(UA_flight_weather$temp[-index])
}

#plot a histogram of the simulated differences
#add a vertical line at the observed difference
ggplot(data=tibble(result), mapping = aes(x=result)) +
  geom_histogram(breaks=seq(-300,300,by=25)) +
  geom_vline(xintercept = observed_diff, color = "red") +
  ggtitle('Distribution of test statistic for 10^4 simulations')
```




```
#Calculate the p-value
p_value <- 2*(sum(result >= observed_diff) + 1) / (N + 1)
p_value
```

```
## [1] 2e-04
```

Observations from the permutation test: The p-value for the two sided permutation is very small. This indicates that the value of observed variance difference, under the null hypothesis is more likely a chance. We can reject our null hypothesis and hence there's a evidence that variance delay for both the late and flights on time might be different.

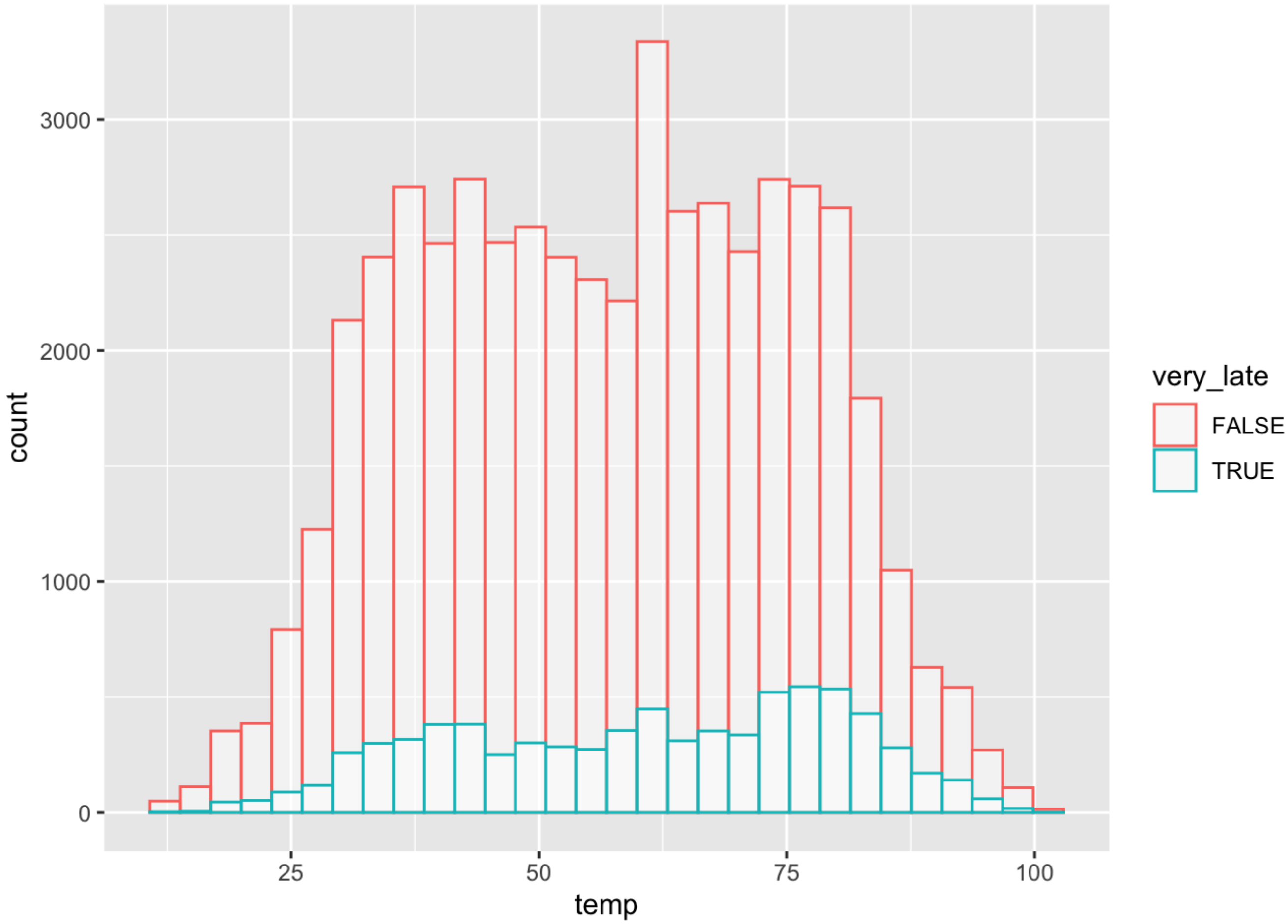
Very Late and temperature

Let's try to visualize the very_late with the temperature and see if there's any trend.

```
ggplot(data = UA_flight_weather , mapping = aes(x = temp,color = very_late)) +
  geom_histogram(fill="white", alpha=0.5, position="identity")
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```

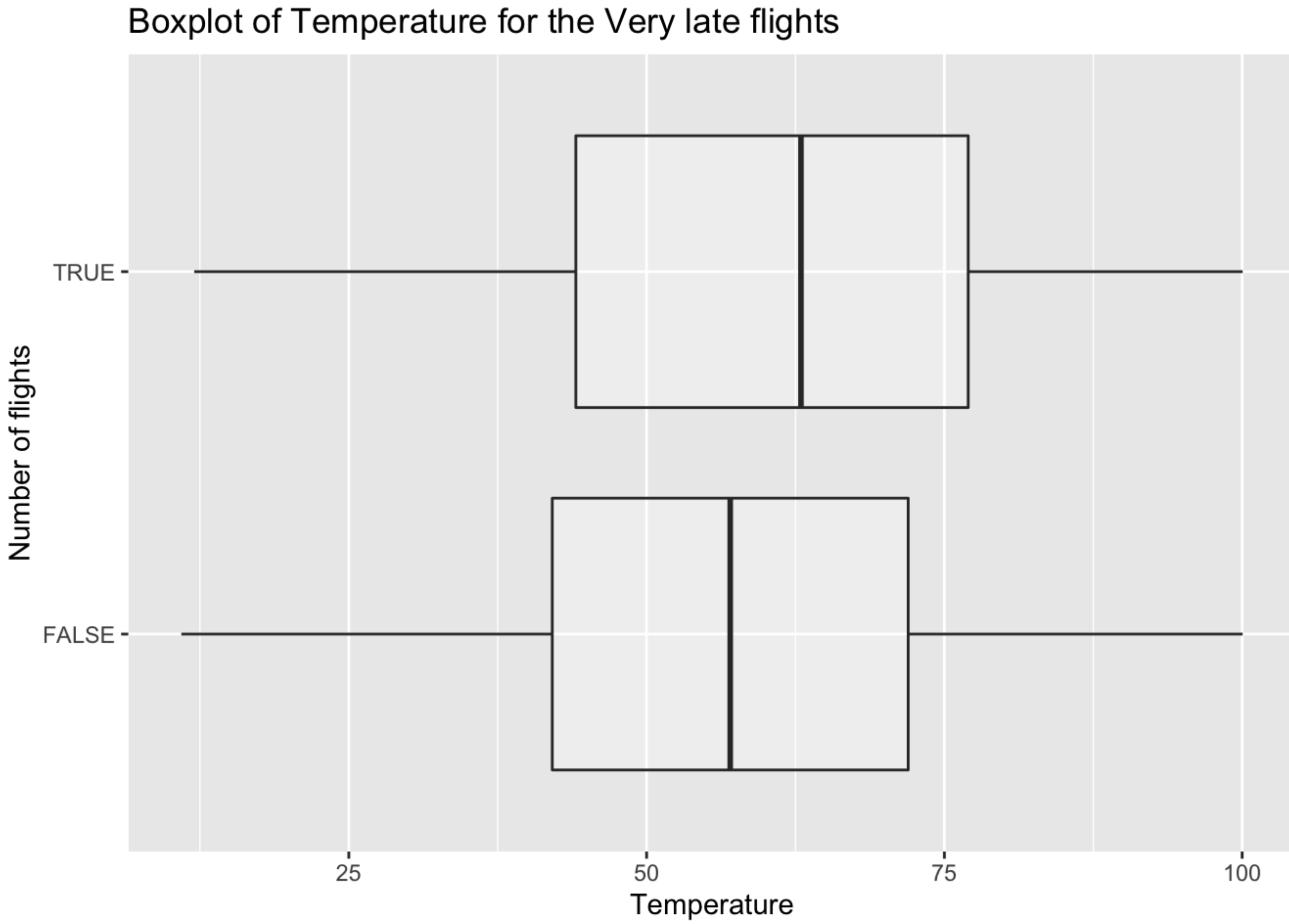
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The temperature follows the same trend for both the flights which were on-time or very late between the flights.

```
ggplot(data= UA_flight_weather , aes(x = temp, y = very_late)) +

  geom_boxplot( alpha=0.3) +
  labs(title = 'Boxplot of Temperature for the Very late flights',x = 'Temperature' , y
 = 'Number of flights')
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```



We can see that there's difference between the temperature mean values for the flights which were very late and almost on time.

Question : Is the mean of temperature of flights for True/False for very_late is equal or not?

H0 : Mean(Temp of flights which were delayed (very late)) = Mean(Temp of flights which were on time and not delayed by 30 mins (very late)) H1 : Mean(Temp of flights which were delayed (very late)) != Mean(Temp of flights which were on time and not delayed by 30 mins (very late))

Let's do a permutation test and compare the mean values between both the values.

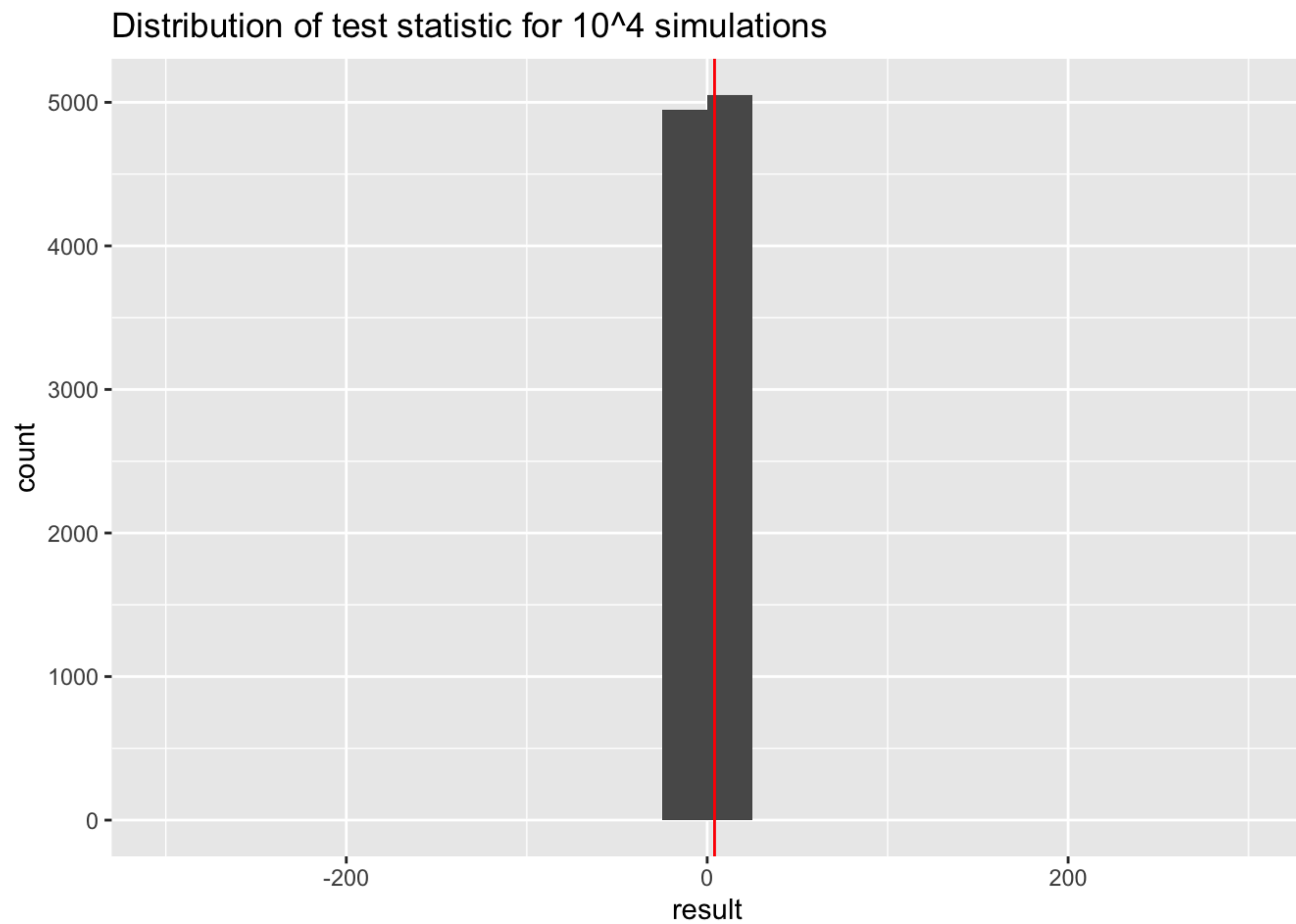
```
#Find the observed difference between flight delays
observed_diff = mean(UA_flight_weather$temp[UA_flight_weather$very_late == TRUE]) -
mean(UA_flight_weather$temp[UA_flight_weather$very_late == FALSE])
print(observed_diff)
```

```
## [1] 4.078207
```

```
# Number of simulation we will use
N <- 10^4-1
#sample.size = the number of observations in our sample
sample.size = nrow(UA_flight_weather)
#group.1.size = the number of observations in the first group : Flights were delayed
group.1.size = length(UA_flight_weather$very_late[UA_flight_weather$very_late == TRUE])
print(group.1.size)
```

```
## [1] 7569
```

```r
#create a blank vector to store the simulation results
result <- numeric(N)
#use a for loop to cycle through values of i ranging from 1 to N
for(i in 1:N)
{
  #each iteration, randomly sample index values
  #sample.size gives the total number of index values to sample from
  #group.1.size gives the number of index values to sample
  #sample without replacement
  #indexes sampled will be treated as the "TRUE" group, indexes not sample as "FALSE"
  index = sample(sample.size, size=group.1.size, replace = FALSE)

  #calculate and store the difference in
  #median rainfall between the index and non-index groups
  result[i] = mean(UA_flight_weather$temp[index]) - mean(UA_flight_weather$temp[-index])
}

#plot a histogram of the simulated differences
#add a vertical line at the observed difference
ggplot(data=tibble(result), mapping = aes(x=result)) +
  geom_histogram(breaks=seq(-300,300,by=25)) +
  geom_vline(xintercept = observed_diff, color = "red") +
  ggtitle('Distribution of test statistic for 10^4 simulations')
```




```
#Calculate the p-value
p_value <- 2*(sum(result >= observed_diff) + 1) / (N + 1)
p_value
```

```
## [1] 2e-04
```

Observations from the permutation test: 1. The p-value is very small. It means that we can reject our null hypothesis. There's a evidence that the alternate hypothesis can be true. We meed to investigate more about it.

Let's try to compare the variance of both the variables.

Question : Is the variance of temperature of flights of True/False is equal or not?

H0 : Var(Temp of flights which were delayed (very late)) = Var(Temp of flights which were on time and not delayed by 30 mins (very late)) H1 : Var(Temp of flights which were delayed (very late)) = Var(Temp of flights which were on time and not delayed by 30 mins (very late))

Let's do a permutation test and compare the variance values between both the values.

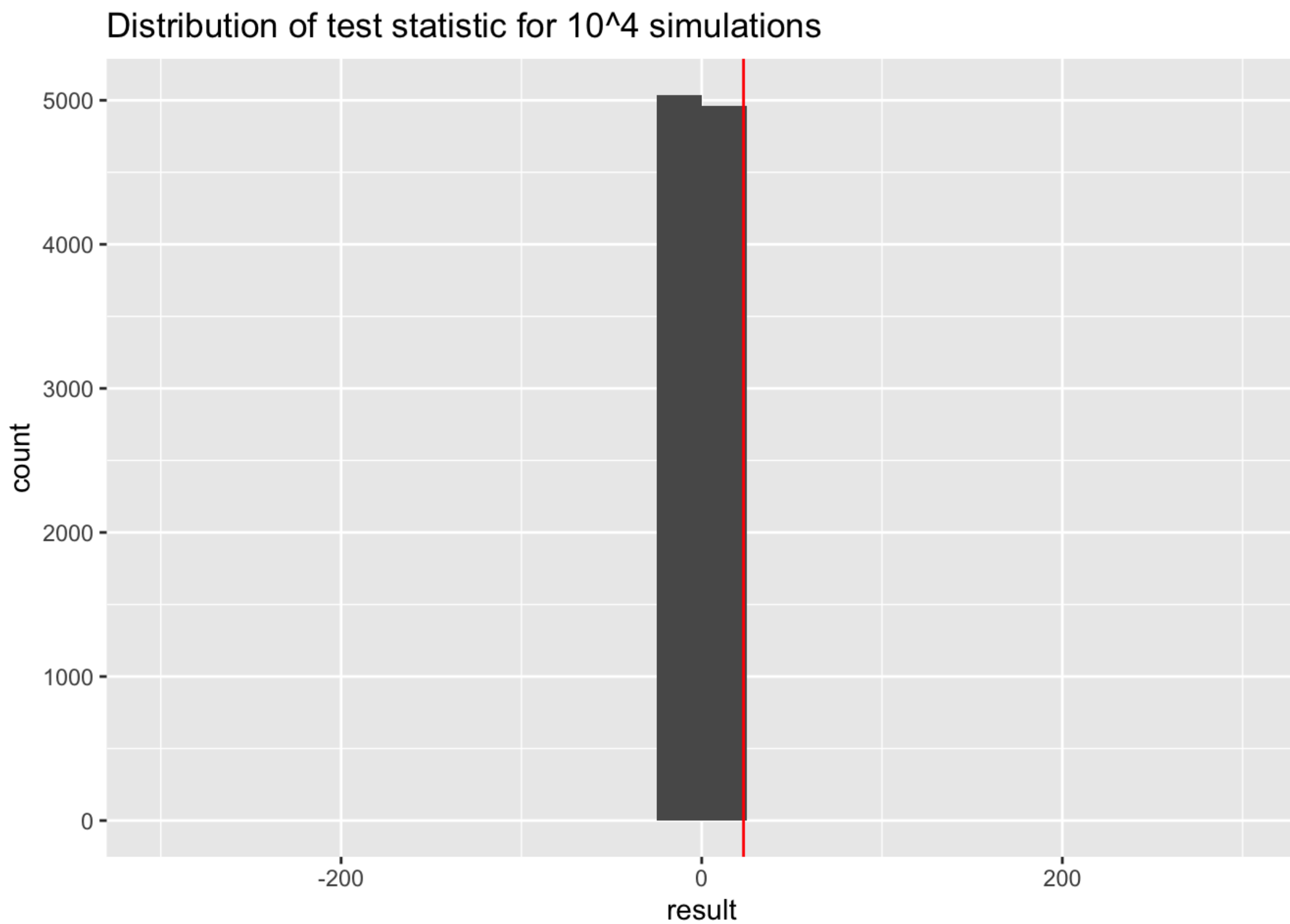
```
#Find the observed difference between flight delays
observed_diff = var(UA_flight_weather$temp[UA_flight_weather$very_late == TRUE]) -
 var(UA_flight_weather$temp[UA_flight_weather$very_late == FALSE])
print(observed_diff)
```

```
## [1] 23.19752
```

```
# Number of simulation we will use
N <- 10^4-1
#sample.size = the number of observations in our sample
sample.size = nrow(UA_flight_weather)
#group.1.size = the number of observations in the first group : Flights were delayed
group.1.size = length(UA_flight_weather$very_late[UA_flight_weather$very_late == TRUE])
print(group.1.size)
```

```
## [1] 7569
```

```
#create a blank vector to store the simulation results
result <- numeric(N)
#use a for loop to cycle through values of i ranging from 1 to N
for(i in 1:N)
{
  #each iteration, randomly sample index values
  #sample.size gives the total number of index values to sample from
  #group.1.size gives the number of index values to sample
  #sample without replacement
  #indexes sampled will be treated as the "TRUE" group, indexes not sample as "FALSE"
  index = sample(sample.size, size=group.1.size, replace = FALSE)

  #calculate and store the difference in
  #median rainfall between the index and non-index groups
  result[i] = var(UA_flight_weather$temp[index]) - var(UA_flight_weather$temp[-index])
}

#plot a histogram of the simulated differences
#add a vertical line at the observed difference
ggplot(data=tibble(result), mapping = aes(x=result)) +
  geom_histogram(breaks=seq(-300,300,by=25)) +
  geom_vline(xintercept = observed_diff, color = "red") +
  ggtitle('Distribution of test statistic for 10^4 simulations')
```




```
#Calculate the p-value
p_value <- 2*(sum(result >= observed_diff) + 1) / (N + 1)
p_value
```

```
## [1] 2e-04
```

Observations from the permutation test: The p-value for the two sided permutation is very small. This indicates that the value of observed variance difference, under the null hypothesis is more likely a chance. We can reject our null hypothesis and hence there's a evidence that mean delay of both the carriers might be different. It means that there's a chance that both the variance are different.

```
quantile(UA_flight_weather$temp,probs=c(.025,.975))
```

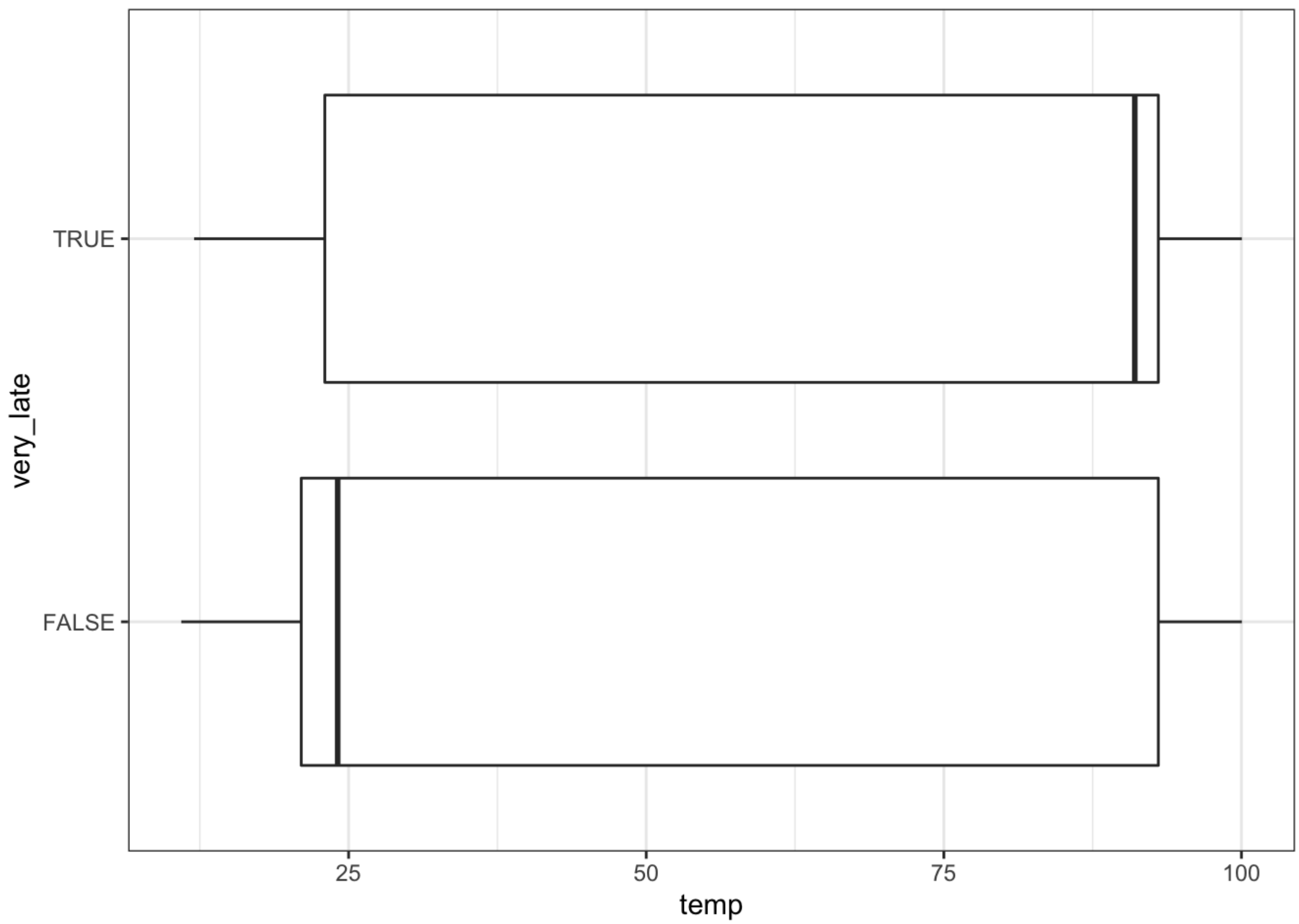
```
##  2.5% 97.5%
## 24.98 89.06
```

24.98 and 89.06 are the 95% confidence value for the dataset. Based on these values we can find the extreme temperatures and see if there are any flights which are delayed or non delayed

```
extreme_temp <- UA_flight_weather %>%
  filter(UA_flight_weather$temp > 89.06 | UA_flight_weather$temp < 24.98)
```

Extreme temperature data

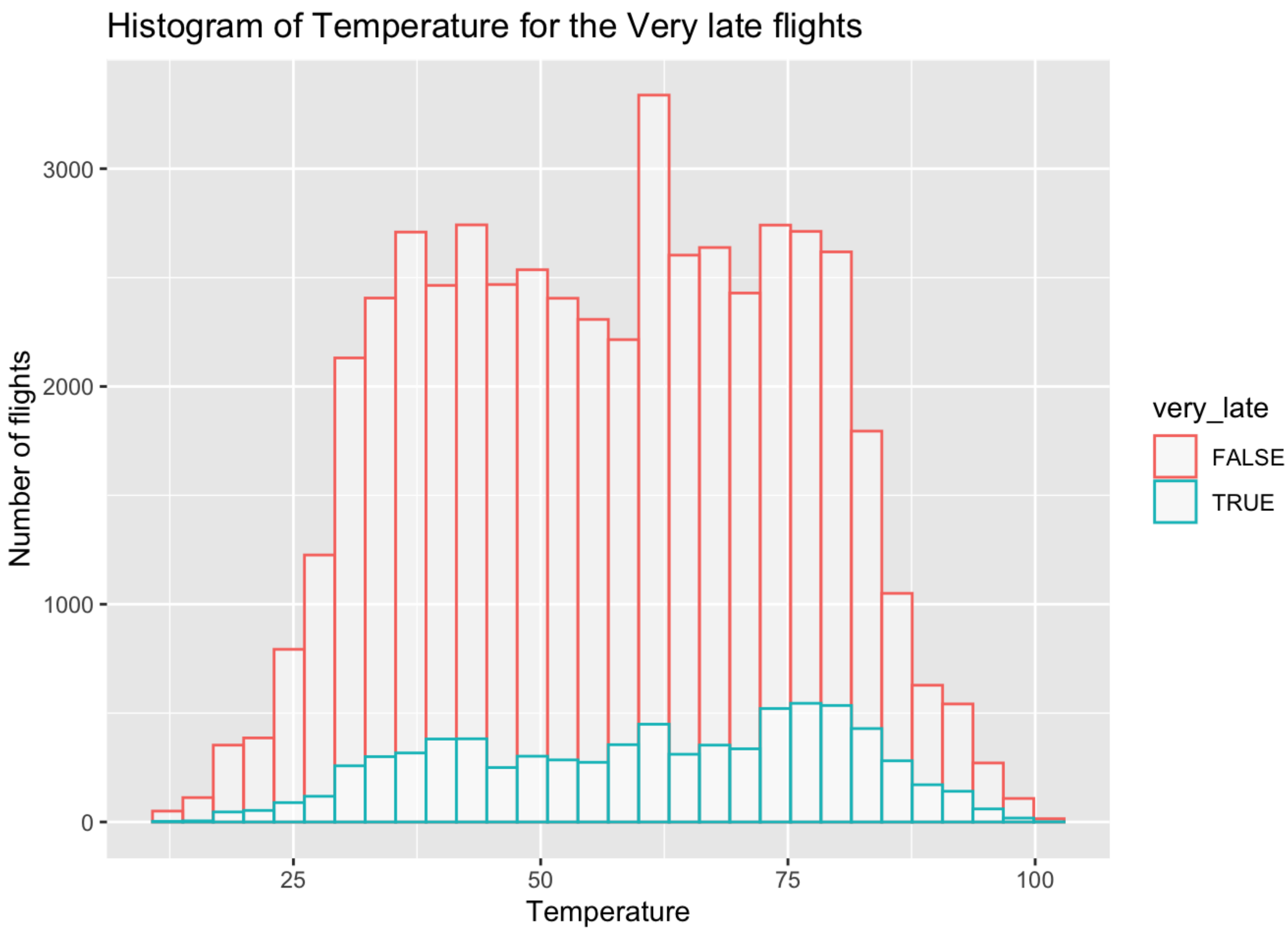
```
ggplot(data= extreme_temp , aes(x = temp, y = very_late)) +
  geom_boxplot() +
  theme_bw()
```




```
ggplot(data = UA_flight_weather , mapping = aes(x = temp,color = very_late)) +
  geom_histogram(fill="white", alpha=0.5, position="identity")+
  labs(title = 'Histogram of Temperature for the Very late flights',x = 'Temperature' ,
 y = 'Number of flights')
```

```
## Don't know how to automatically pick scale for object of type impute. Defaulting to c
ontinuous.
```

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



No,

extreme temperatures does not impact the flight delays.

```
#print out the mean temperature of UA flights for the very late group
mean(UA_flight_weather$temp[UA_flight_weather$very_late==TRUE])
```

```
## [1] 60.84745
```

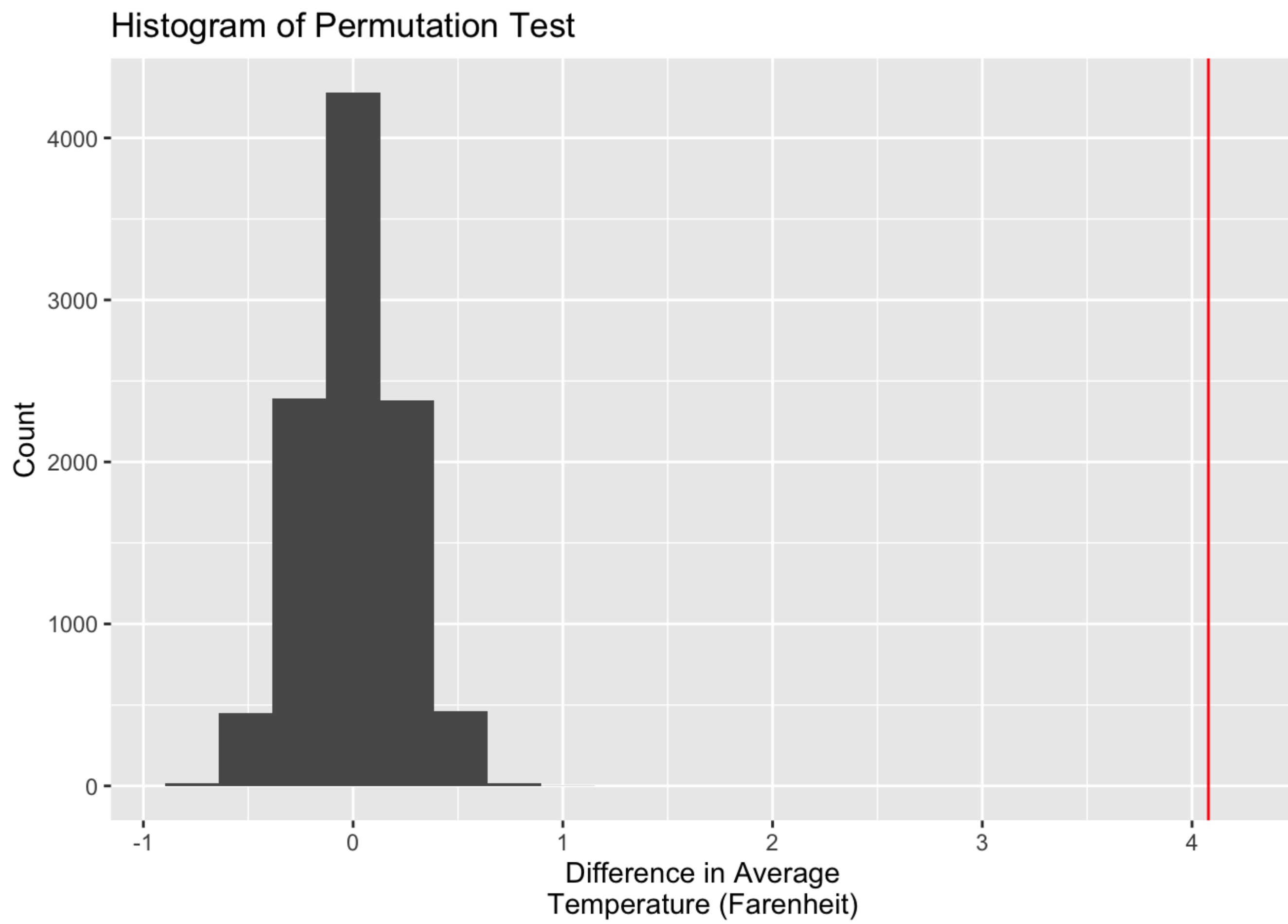
```
#print out the mean teperature of UA flights for the not very late group
mean(UA_flight_weather$temp[UA_flight_weather$very_late==FALSE])
```

```
## [1] 56.76924
```

```
#calculate and store the observed difference between the mean of temperature in the very
late group and that in the not very late group
observed.temp <- mean(UA_flight_weather$temp[UA_flight_weather$very_late==TRUE]) - mean
(UA_flight_weather$temp[UA_flight_weather$very_late==FALSE])
observed.temp
```

```
## [1] 4.078207
```

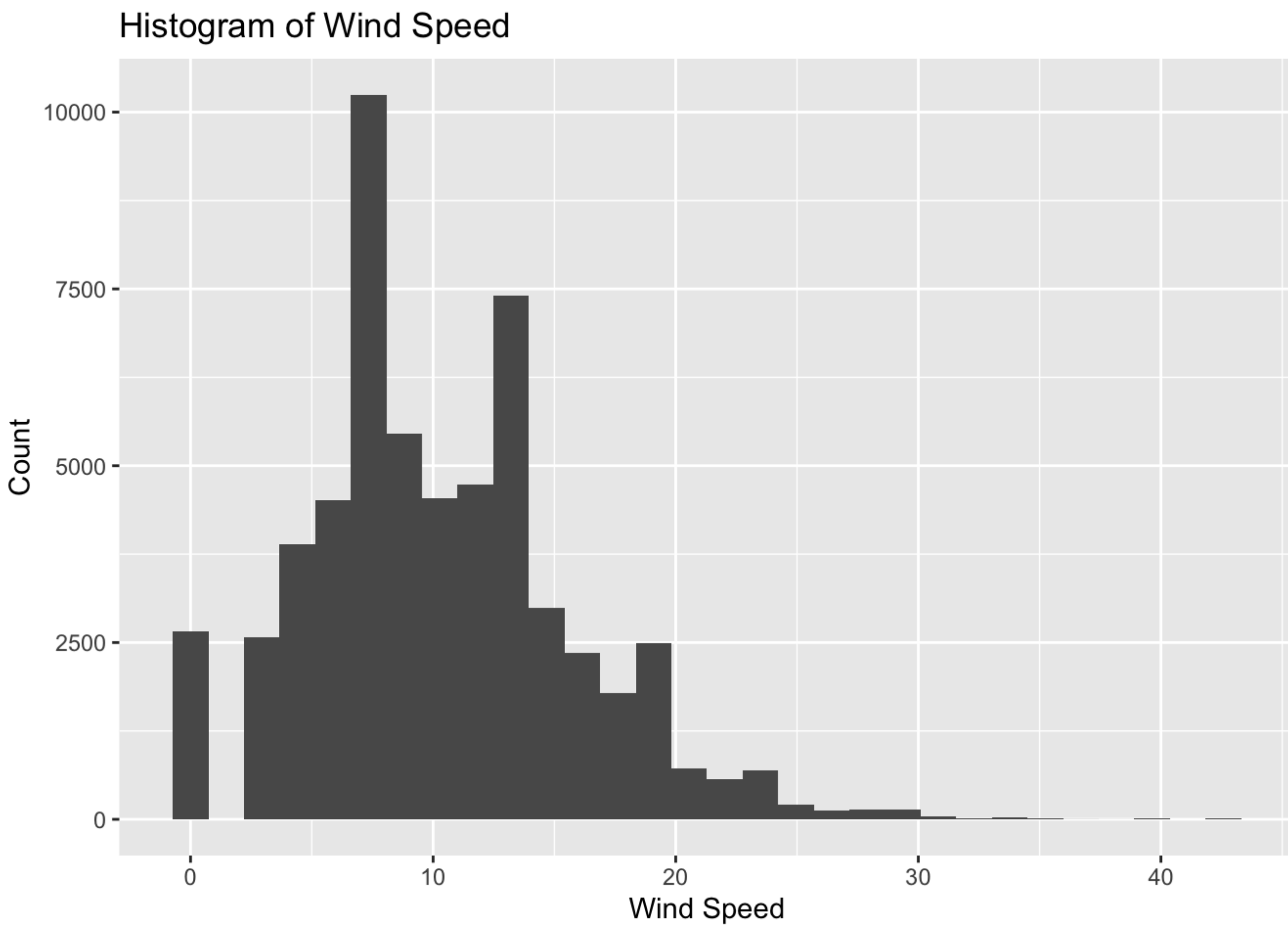
```r
#set N to be 10^4-1, this is large enough to keep results stable from run to run
N <- 10^4-1
#calculate and store the sample size, which is the number of observations in the very la
te group and that in the not very late group
sample.size.temp = nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,]) + nrow(UA
_flight_weather[UA_flight_weather$very_late==FALSE,])
#find and store the sample size for the very late group
group.1.size.temp <- nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,])
#initialize the vector that stores the N many results
result.temp <- numeric(N)
#create the for loop
for(i in 1:N)
{
#sample group.1.size many numbers from sample.size.distance numbers without replacement
index.temp = sample(sample.size.temp,size=group.1.size.temp, replace = FALSE)
#sampled indexes are taken as the indexes for very late group, and the rest are for not
 very late group
#calculate and store the difference between the mean of new groups
result.temp[i] = mean(UA_flight_weather$temp[index.temp]) -
mean(UA_flight_weather$temp[-index.temp])
}
#create the histogram of the means as well as a verticle line that respresent the observ
ed mean
ggplot(data=tibble(result.temp), mapping = aes(x=result.temp)) +
geom_histogram(bins = 20) +
geom_vline(xintercept = observed.temp, color = "red") +
labs(title = "Histogram of Permutation Test", x = "Difference in Average
Temperature (Farenheit)", y = "Count")
```




```
ggplot(data = UA_flight_weather , mapping = aes(x = wind_speed)) +
  geom_histogram()+
  labs(title = 'Histogram of Wind Speed',x = 'Wind Speed' , y = 'Count')
```

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

```
## Warning: Removed 17 rows containing non-finite values (stat_bin).
```




```
#print out the mean wind speed of UA flights for the very late group
mean(UA_flight_weather$wind_speed[UA_flight_weather$very_late==TRUE],na.rm=TRUE)
```

```
## [1] 10.77716
```

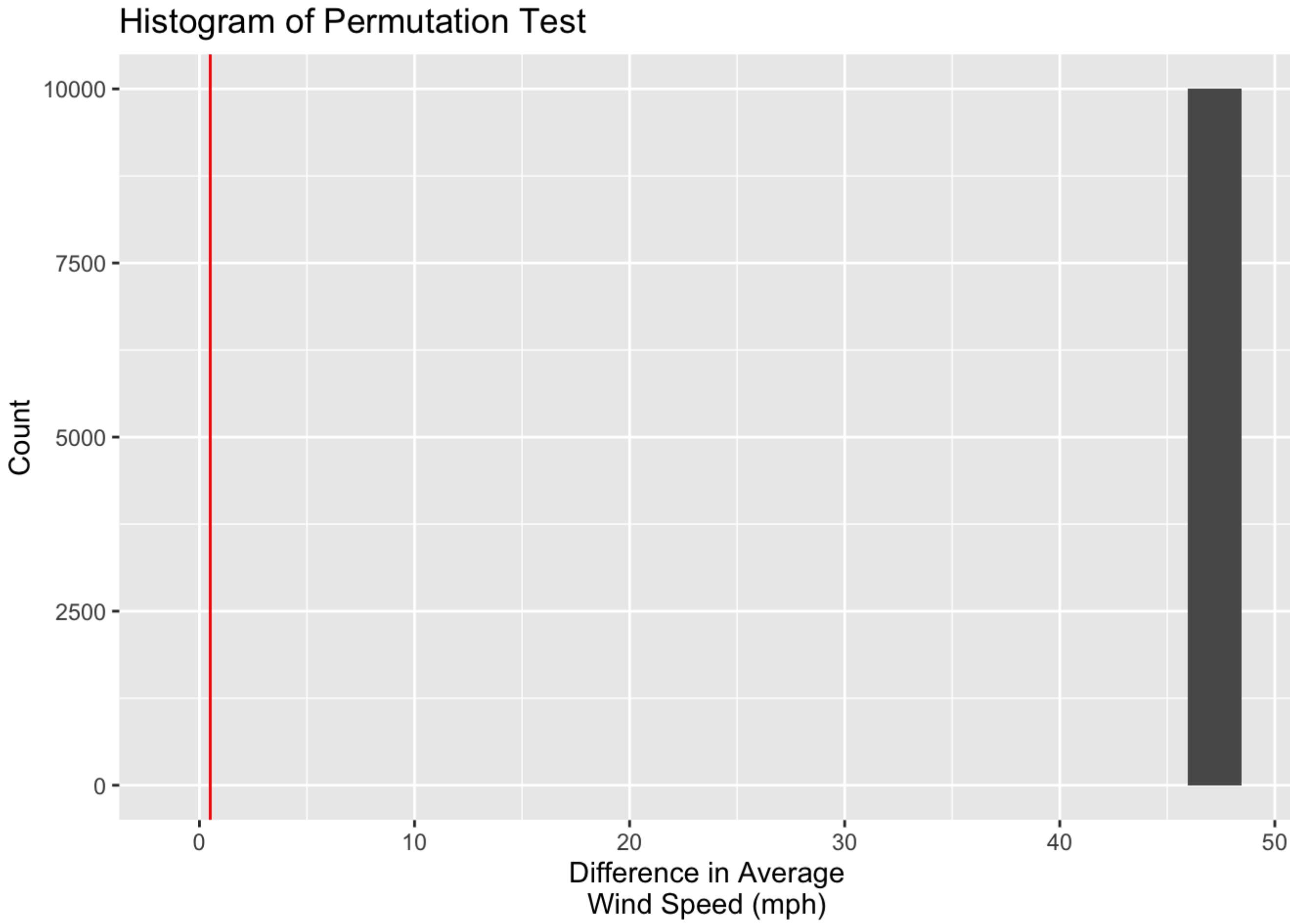
```
#print out the mean wind speed of UA flights for the not very late group
mean(UA_flight_weather$wind_speed[UA_flight_weather$very_late==FALSE],na.rm=TRUE)
```

```
## [1] 10.27529
```

```
#calculate and store the observed difference between the mean of wind speed in the very
 late group and that in the not very late group
observed.wind_speed <- mean(UA_flight_weather$wind_speed[UA_flight_weather$very_late==TR
UE],na.rm=TRUE) -
mean(UA_flight_weather$wind_speed[UA_flight_weather$very_late==FALSE],na.rm=TRUE)
observed.wind_speed
```

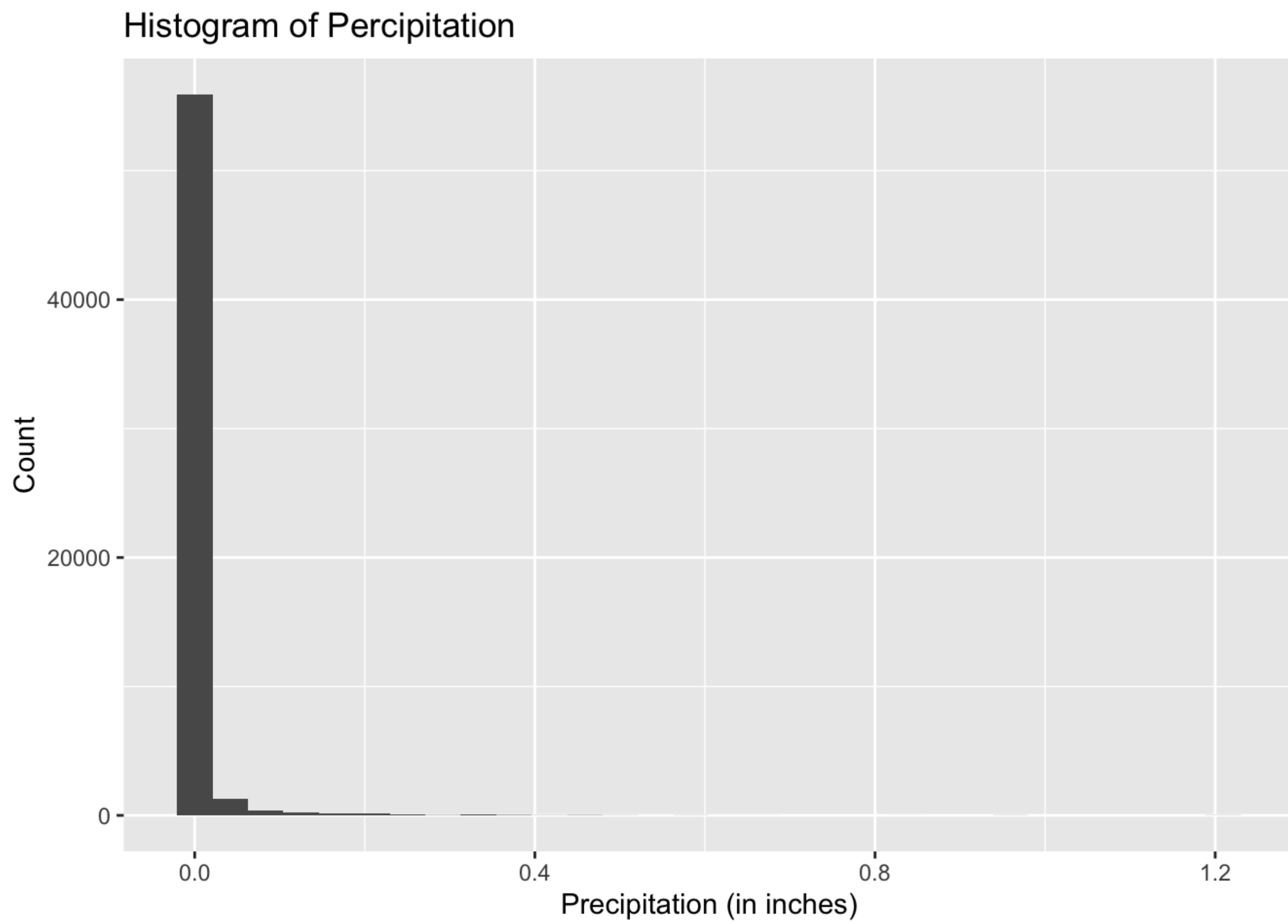
```
## [1] 0.501864
```

```r
N <- 10^4-1
#calculate and store the sample size, which is the number of observations in the very la
te group and that in the not very late group
sample.size.wind_speed = nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,]) +
nrow(UA_flight_weather[UA_flight_weather$very_late==FALSE,])
#find and store the sample size for the very late group
group.1.size.wind_speed <- nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,])
#initialize the vector that stores the N many results
result.wind_speed <- numeric(N)
#create the for loop
for(i in 1:N)
{
#sample group.1.size many numbers from sample.size.distance numbers without replacement
index.temp = sample(sample.size.temp,size=group.1.size.temp, replace = FALSE)
#sampled indexes are taken as the indexes for very late group, and the rest are for not
 very late group
#calculate and store the difference between the mean of new groups
result.temp[i] = mean(UA_flight_weather$temp[index.temp],na.rm=TRUE) -
mean(UA_flight_weather$wind_speed[-index.temp],na.rm=TRUE)
}
#create the histogram of the means as well as a verticle line that respresent the observ
ed mean
ggplot(data=tibble(result.wind_speed), mapping = aes(x=result.temp)) +
geom_histogram(bins = 20) +
geom_vline(xintercept = observed.wind_speed, color = "red") +
labs(title = "Histogram of Permutation Test", x = "Difference in Average
Wind Speed (mph)", y = "Count")
```

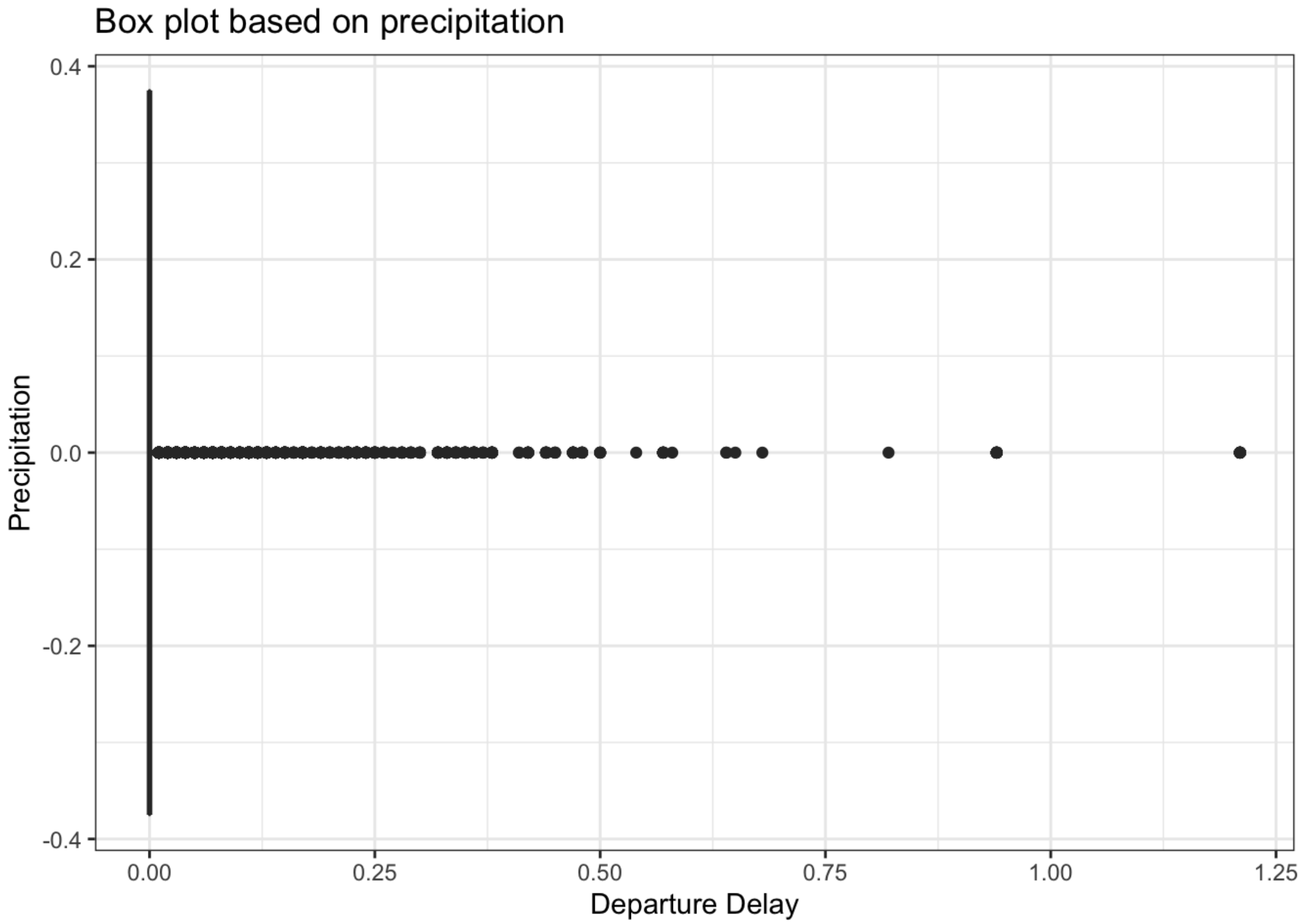



```
ggplot(data = UA_flight_weather , mapping = aes(x = precip)) +
  geom_histogram()+
  labs(title = 'Histogram of Percipitation',x = 'Precipitation (in inches)' , y = 'Coun
t')
```

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




```
ggplot(data= UA_flight_weather , aes(x = precip)) +
  geom_boxplot() +
  theme_bw() +
  labs(x = 'Departure Delay', title = 'Box plot based on precipitation',y='Precipitatio
n')
```




```
#mean precip of UA flights for the very late group
mean(UA_flight_weather$precip[UA_flight_weather$very_late==TRUE],na.rm=TRUE)
```

```
## [1] 0.01166072
```

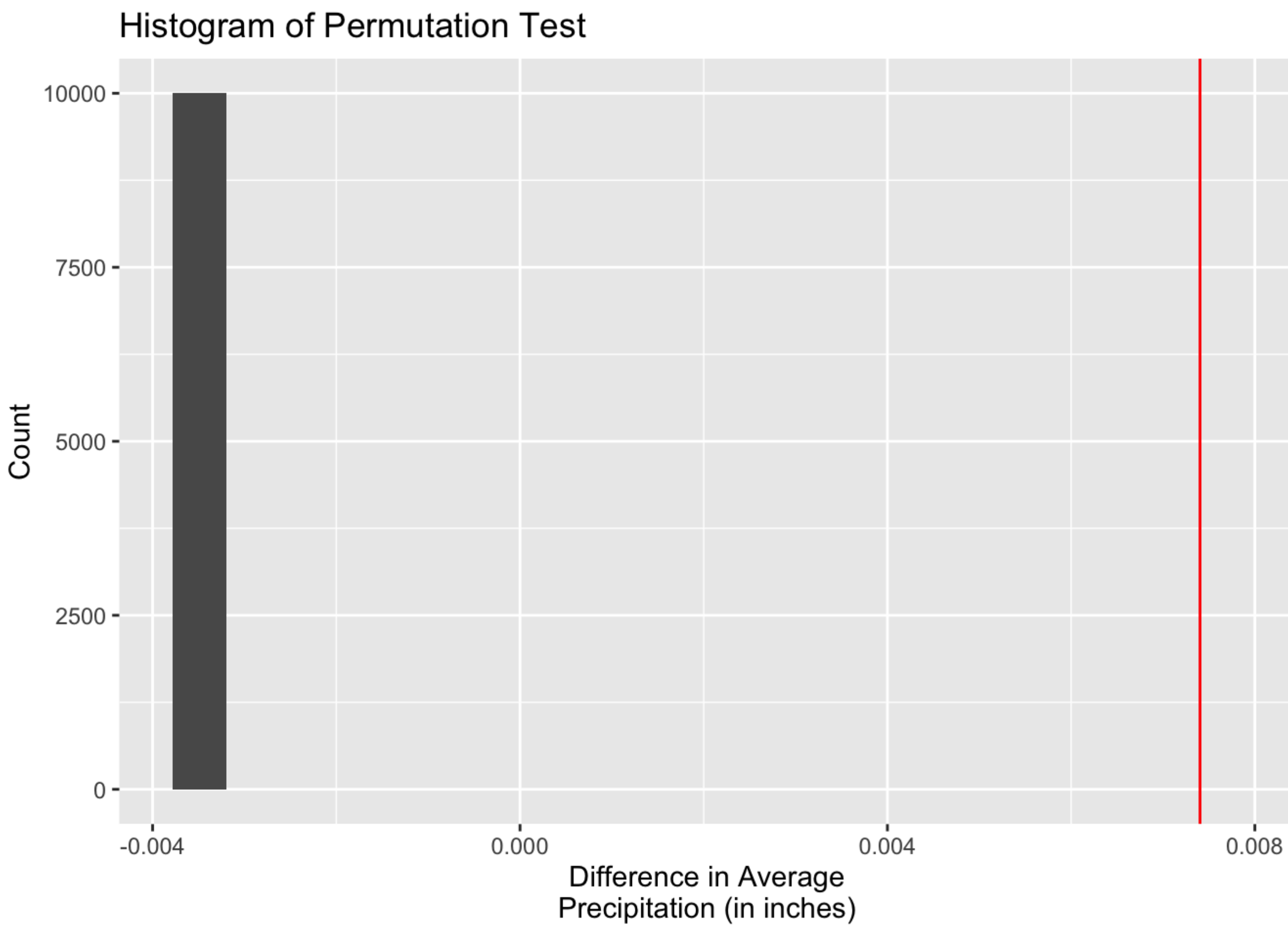
```
#mean precip of UA flights for the not very late group
mean(UA_flight_weather$precip[UA_flight_weather$very_late==FALSE],na.rm=TRUE)
```

```
## [1] 0.00425756
```

```
#calculate and store the observed difference between the mean of precip in the very late
group and that in the not very late group
observed.precip <- mean(UA_flight_weather$precip[UA_flight_weather$very_late==TRUE],na.r
m=TRUE) -
mean(UA_flight_weather$precip[UA_flight_weather$very_late==FALSE],na.rm=TRUE)
observed.precip
```

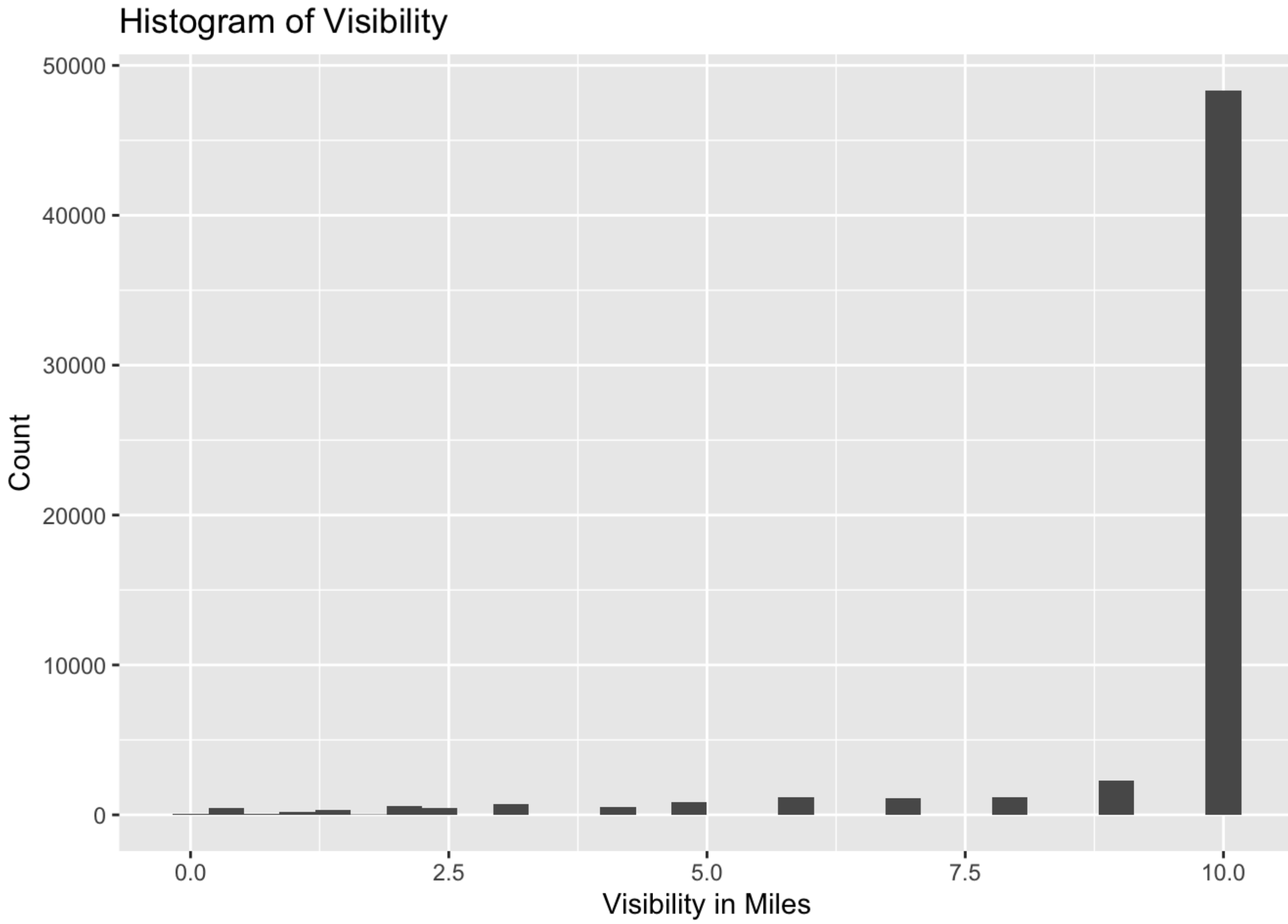
```
## [1] 0.007403161
```

```
N <- 10^4-1
#calculate and store the sample size, which is the number of observations in the very la
te group and that in the not very late group
sample.size.precip = nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,])
#find and store the sample size for the very late group
group.1.size.precip <- nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,])
#initialize the vector that stores the N many results
result.temp <- numeric(N)
#create the for loop
for(i in 1:N)
{
#sample group.1.size many numbers from sample.size.distance numbers without replacement
index.precip = sample(sample.size.precip,size=group.1.size.precip, replace = FALSE)
#sampled indexes are taken as the indexes for very late group, and the rest are for not
 very late group
#calculate and store the difference between the mean of new groups
result.temp[i] = mean(UA_flight_weather$precip[index.precip],na.rm=TRUE) -
mean(UA_flight_weather$precip[-index.precip],na.rm=TRUE)
}
#create the histogram of the means as well as a verticle line that respresent the observ
ed mean
ggplot(data=tibble(result.temp), mapping = aes(x=result.temp)) +
geom_histogram(bins = 20) +
geom_vline(xintercept = observed.precip, color = "red") +
labs(title = "Histogram of Permutation Test", x = "Difference in Average
Precipitation (in inches)", y = "Count")
```




```
ggplot(data = UA_flight_weather , mapping = aes(x = visib)) +
  geom_histogram()+
  labs(title = 'Histogram of Visibility',x = 'Visibility in Miles' , y = 'Count')
```

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




```
#print out the mean visibility of UA flights for the very late group
mean(UA_flight_weather$visib[UA_flight_weather$very_late==TRUE],na.rm=TRUE)
```

```
## [1] 8.976314
```

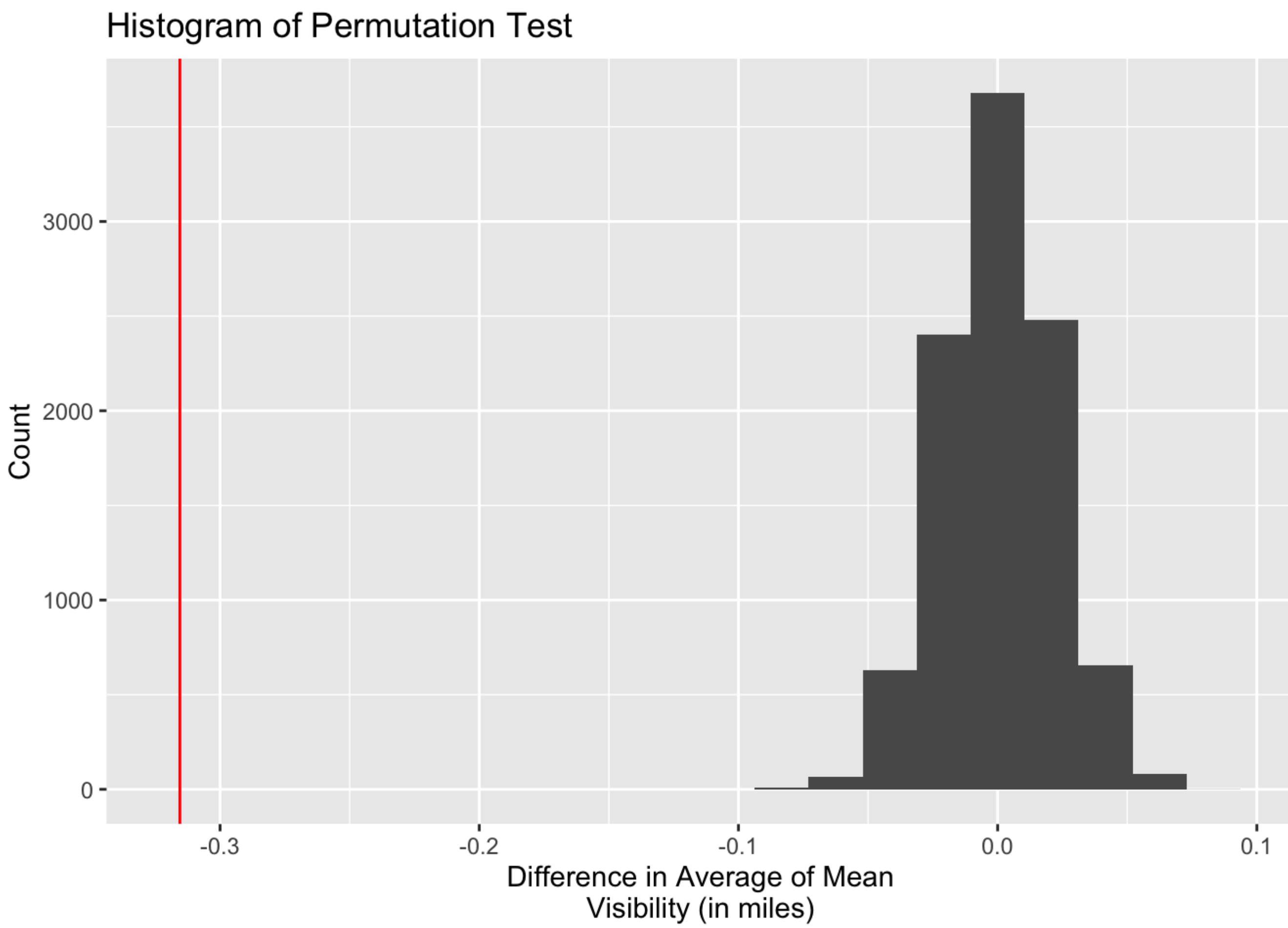
```
#print out the mean visibility of UA flights for the not very late group
mean(UA_flight_weather$visib[UA_flight_weather$very_late==FALSE],na.rm=TRUE)
```

```
## [1] 9.291829
```

```
#calculate and store the observed difference between the mean of visibility in the very
 late group and that in the not very late group
observed.visib <- mean(UA_flight_weather$visib[UA_flight_weather$very_late==TRUE],na.rm=
TRUE) -
mean(UA_flight_weather$visib[UA_flight_weather$very_late==FALSE],na.rm=TRUE)
observed.visib
```

```
## [1] -0.3155153
```

```
N <- 10^4-1
#calculate and store the sample size, which is the number of observations in the very la
te group and that in the not very late group
sample.size.temp = nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,]) +
nrow(UA_flight_weather[UA_flight_weather$very_late==FALSE,])
#find and store the sample size for the very late group
group.1.size.temp <- nrow(UA_flight_weather[UA_flight_weather$very_late==TRUE,])
#initialize the vector that stores the N many results
result.temp <- numeric(N)
#create the for loop
for(i in 1:N)
{
#sample group.1.size many numbers from sample.size.distance numbers without replacement
index.temp = sample(sample.size.temp,size=group.1.size.temp, replace = FALSE)
#sampled indexes are taken as the indexes for very late group, and the rest are for not
 very late group
#calculate and store the difference between the mean of new groups
result.temp[i] = mean(UA_flight_weather$visib[index.temp],na.rm=TRUE) -
mean(UA_flight_weather$visib[-index],na.rm=TRUE)
}
#create the histogram of the means as well as a verticle line that respresent the observ
ed mean
ggplot(data=tibble(result.temp), mapping = aes(x=result.temp)) +
geom_histogram(bins = 20) +
geom_vline(xintercept = observed.visib, color = "red") +
labs(title = "Histogram of Permutation Test", x = "Difference in Average of Mean
Visibility (in miles)", y = "Count")
```


Histogram of Permutation Test
Count
3000
2000
1000
0
-0.3
-0.2
-0.1
0.0
0.1
Difference in Average of Mean
Visibility (in miles)