An Analysis of US domestic airline delays and cancellation

Business Analytics with R project

Under - Prof. Subhra Rani Patra

By
Chou Tzu-Jui : TXC220003
Zulqarnain Sourathia : ZXS140630
Rakshit Mathur : RXM210132
Salman Shaik SXS210627
Mucharla Mohan Balachandra Nitish - MXM220054

Abstract

Flight delays have always been almost an inevitable part of flight experience. An analysis of the reason and the possible causes of flight delays can be crucial for airlines to evaluate an important part of their performance. Understanding what causes the delay can facilitate airlines and make improvement on the customer satisfaction and ultimately improve the overall profit. In this paper a linear regression model is proposed. We intend to find out the relationship between two variables: TaxiOut and CarrierDelay, more specifically, TaxiOut as a function of CarrierDelay. The reasoning behind is that oftentimes airlines claim they have nothing to do with the delay issues and put forth other external factors to evade the responsibility to do with the airline industry, however this is not the case. We have historical data that show that this carrier delay does have a correlation with Taxiout, the time for planes to leave once they are on the runway. Therefore, it is very possible that in fact these airlines should be held accountable and make efforts to improve their service. To be definitive about this statement, furthermore recent data will have to be included. Prior to this linear model, several exploratory data analysis visualizations are attempted to establish more understanding toward delays and cancellations.

Introduction:

This dataset we are using is from kaggle. It contains roughly 1.9 million records of US domestic flights from 1987 to 2008, and below is a detailed explanation of each of the columns from this dataset.

X - Serial Number

Year - Year on which the flight data was recorded

Month - Month on which the flight data was recorded

DayofMonth - Day on which the flight data was recorded

DepTime - Actual Departure time of the each flight

CRSDepTime - Scheduled Departure time

ArrTime - Actual arrival time

CRSArrTime - ArrTime scheduled arrival time

UniqueCarrier - Unique carrier code

FlightNum - Flight number

TailNum - Plane tail number

ActualElapsedTime - Actual time took to Elapse (in minutes)

CRSElapsedTime - Scheduled time to Elapse (in minutes)

AirTime - Time in Air (in minutes)

ArrDelay - Arrival delay(in minutes)

DepDelay - Departure delay(in minutes)

Origin - Origin IATA airport code

Dest - Des tination IATA airport code

Distance - Distance between Origin and Destination in miles

TaxiIn - Taxi in time, in minutes

TaxiOut - Taxi out time in minutes

Canceled - Was the flight canceled? (Boolean Value 0 or 1)

CancellationCode - Reason for cancellation (A = carrier, B = weather, C = NAS, D = security)

Diverted - Whether the flight was diverted or not (Boolean value 1 = yes, 0 = no)

CarrierDelay - Delay time took by the carrier, in minutes

WeatherDelay - Delay time due to Weather, in minutes

NASDelay - Delay due to National Aviation System ,in minutes

SecurityDelay - Time Delay due to Security check in minutes

LateAircraftDelay - Time Delay due to late arrival of same aircraft at previous airport, in minutes

Data preparation/Data summary

```
#Loading data
library(readxl)
setwd("C:/Users/raych/Downloads")
DF <- read.csv("DelayedFlights.csv")
str(DF)
```

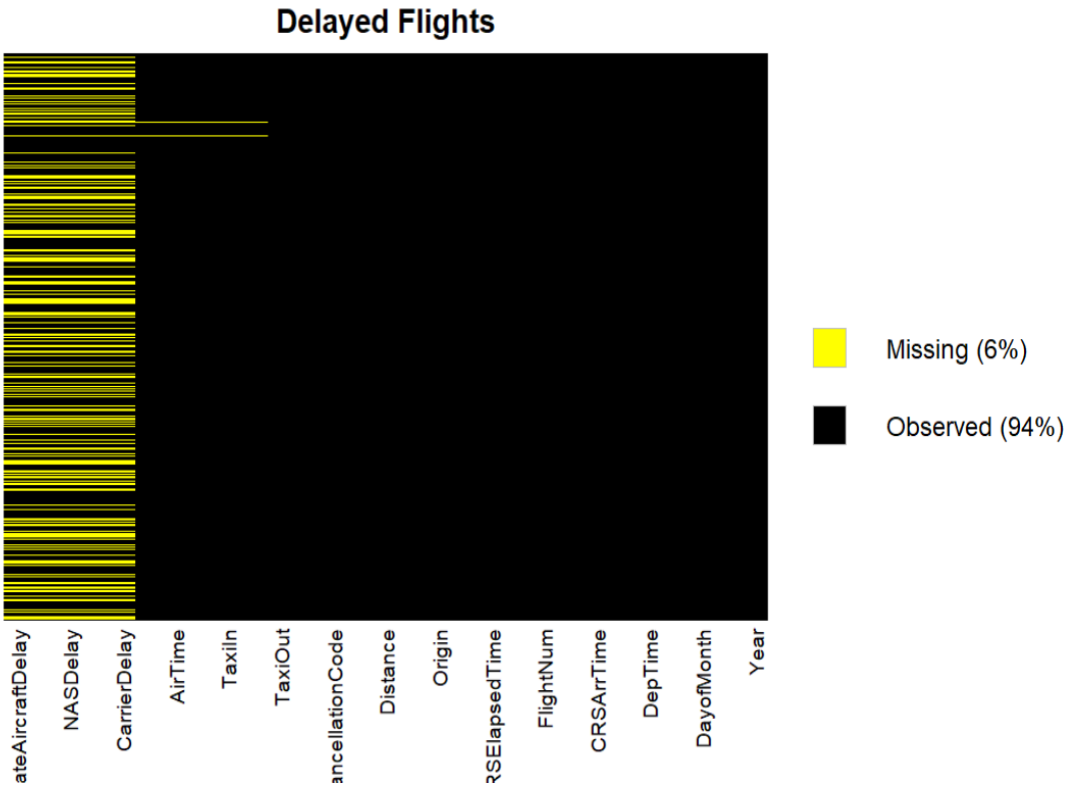
```
'data.frame':   1936758 obs. of  29 variables:
 $ Year             : int  2008 2008 2008 2008 2008 2008 2008 2008 2008 2008 ...
 $ Month            : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DayofMonth       : int  3 3 3 3 3 3 3 3 3 3 ...
 $ DayOfWeek        : int  4 4 4 4 4 4 4 4 4 4 ...
 $ DepTime          : num  2003 754 628 1829 1940 ...
 $ CRSDepTime       : int  1955 735 620 1755 1915 1830 700 1510 1020 1425 ...
 $ ArrTime          : num  2211 1002 804 1959 2121 ...
 $ CRSArrTime       : int  2225 1000 750 1925 2110 1940 915 1725 1010 1625 ...
 $ UniqueCarrier    : chr  "WN" "WN" "WN" "WN" ...
 $ FlightNum        : int  335 3231 448 3920 378 509 100 1333 2272 675 ...
 $ TailNum          : chr  "N712SW" "N772SW" "N428WN" "N464WN" ...
 $ ActualElapsedTime: num  128 128 96 90 101 240 130 121 52 228 ...
 $ CRSElapsedTime   : num  150 145 90 90 115 250 135 135 50 240 ...
 $ AirTime          : num  116 113 76 77 87 230 106 107 37 213 ...
 $ ArrDelay         : num  -14 2 14 34 11 57 1 80 11 15 ...
 $ DepDelay         : num  8 19 8 34 25 67 6 94 9 27 ...
 $ Origin           : chr  "IAD" "IAD" "IND" "IND" ...
 $ Dest             : chr  "TPA" "TPA" "BWI" "BWI" ...
 $ Distance         : int  810 810 515 515 688 1591 828 828 162 1489 ...
 $ TaxiIn           : num  4 5 3 3 4 3 5 6 6 7 ...
 $ TaxiOut          : num  8 10 17 10 10 7 19 8 9 8 ...
 $ Cancelled        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ CancellationCode : chr  "N" "N" "N" "N" ...
 $ Diverted         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ CarrierDelay     : num  NA NA NA 2 NA 10 NA 8 NA 3 ...
 $ WeatherDelay     : num  NA NA NA 0 NA 0 NA 0 NA 0 ...
 $ NASDelay         : num  NA NA NA 0 NA 0 NA 0 NA 0 ...
 $ SecurityDelay    : num  NA NA NA 0 NA 0 NA 0 NA 0 ...
 $ LateAircraftDelay: num  NA NA NA 32 NA 47 NA 72 NA 12 ...
```

First we remove the redundant column

```
DF <- DF[,-1]
```

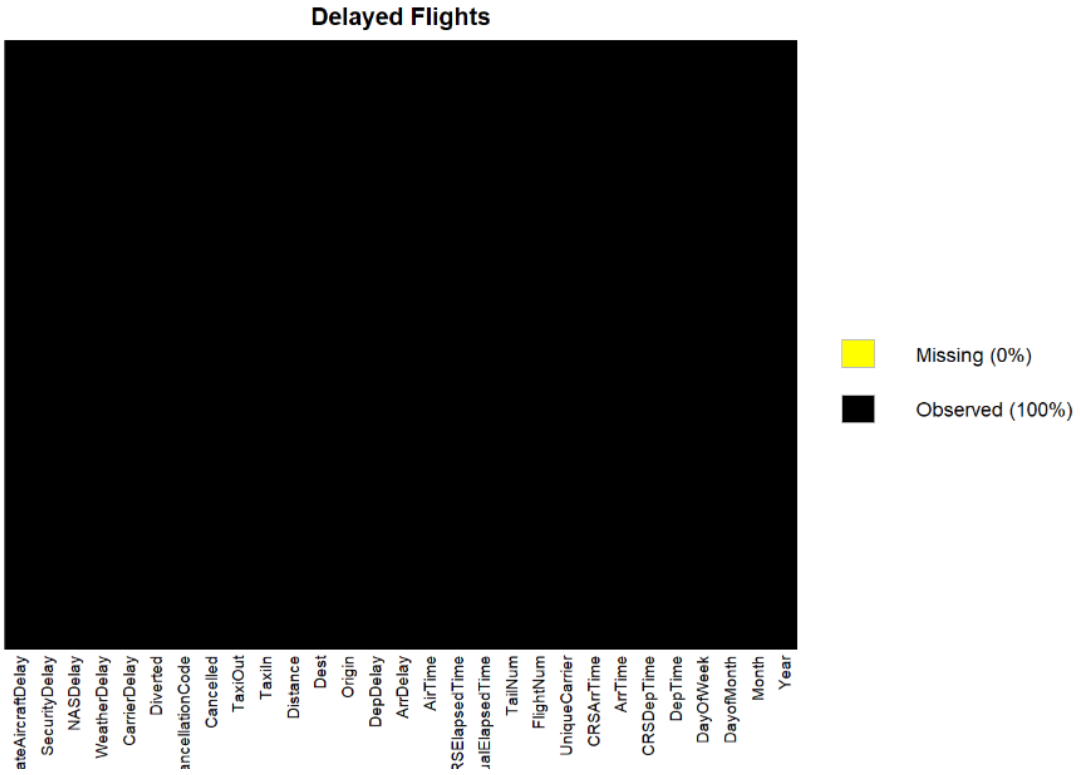
Factorize the columns for them to be used in the missmap

```
DF$UniqueCarrier = factor(DF$UniqueCarrier)
DF$TailNum = factor(DF$TailNum)
DF$Origin = factor(DF$Origin)
DF$Dest = factor(DF$Dest)
DF$CancellationCode = factor(DF$CancellationCode)
DF$DayOfWeek = factor(DF$DayOfWeek)
library(dplyr)
#create a missmap the visualize the missing value
library(Amelia)
missmap(DF,
        main="Delayed Flights",
        y.labels = NULL,
        y.at = NULL,
        col=c("yellow", "black"),
        legend = TRUE)
```



As can be seen in the missmap there are plenty of NA values, now we will clean out the NA values and run a missmap again.

```
DF1
DF_NA_cleared <- na.omit(DF)
any(is.na(DF_NA_cleared))
#Visualize the missing values again after removing the NA values
library(Amelia)
missmap(DF_NA_cleared,
        main="Delayed Flights",
        y.labels = NULL,
        y.at = NULL,                    #here, the standard ylab should be y.label in
Amelia. Also, y.label should always appear with y.at
        col=c("yellow", "black"),
        legend = TRUE)
```



The missmap now displays no NA Values. The dataset has been cleaned.

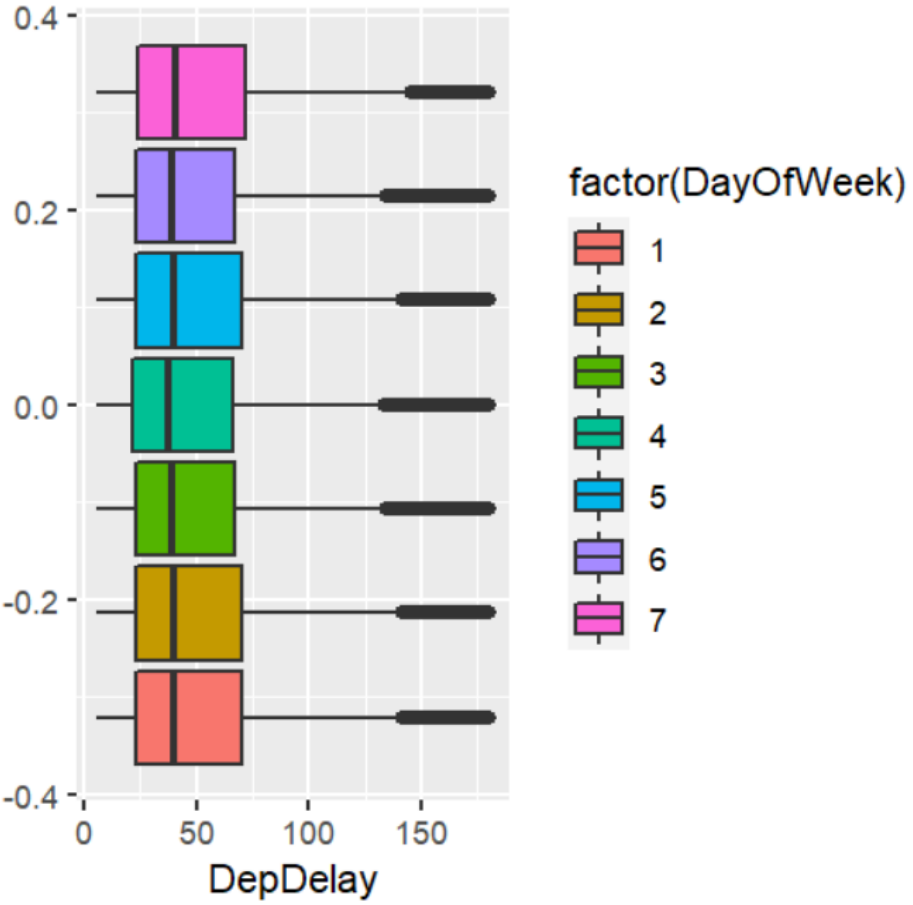
EDA1 -What day of the week tends to have the most departure delay?

R-Script:

```
delay_under_180 <- subset(DF_NA_cleared, DepDelay <= 180) #get rid of the ones that are over 180

#(We get this value(180 minutes) by US. department of transportation: "For flights landing at U.S. airports, airlines are required to

#provide passengers with an opportunity to safely get off of the airplane before 3 hours for domestic flights.")

install.packages("lattice")

library("lattice")

library(ggplot2)

attach(delay_under_180)

EDA1 <- ggplot(delay_under_180, aes(x = DepDelay)) + geom_boxplot(aes(fill = factor(DayOfWeek)))

EDA1
```

```
LogDepDelay <- log(DepDelay) # logarize the DepDeplay column to remove the outliers

EDA1_log <- ggplot(delay_under_180, aes(x = LogDepDelay)) +
geom_boxplot(aes(fill = factor(DayOfWeek)))

EDA1_log
```

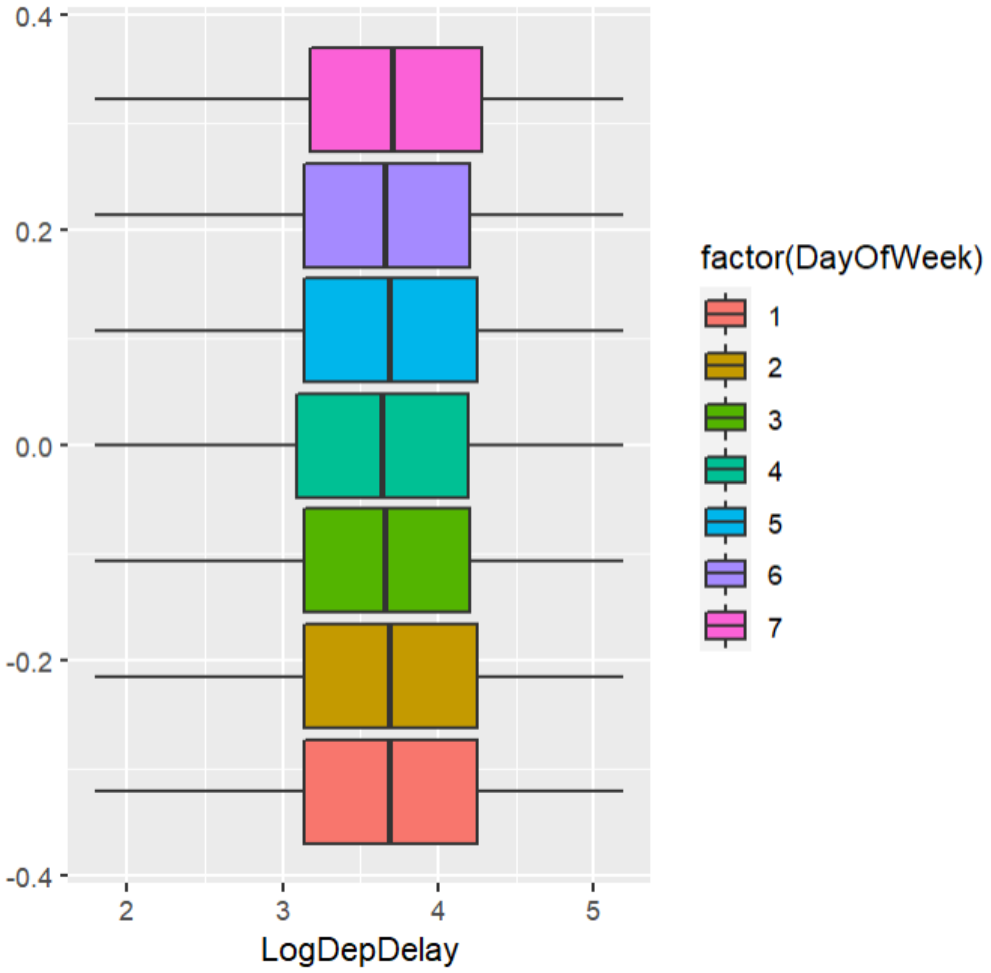
Plot:



Comparison: Departure Delay on Each Day of the Week

Explanantion:

One thing to note from this model is that so far it seems as if there is not much difference between the departure delays across the different days of the week. However, we have observed that there are many outliers for each day of the week. So in order to make better sense of the visualization we will log departure delay in the following model.



Departure Delay on Each Day of the Week: Logarized

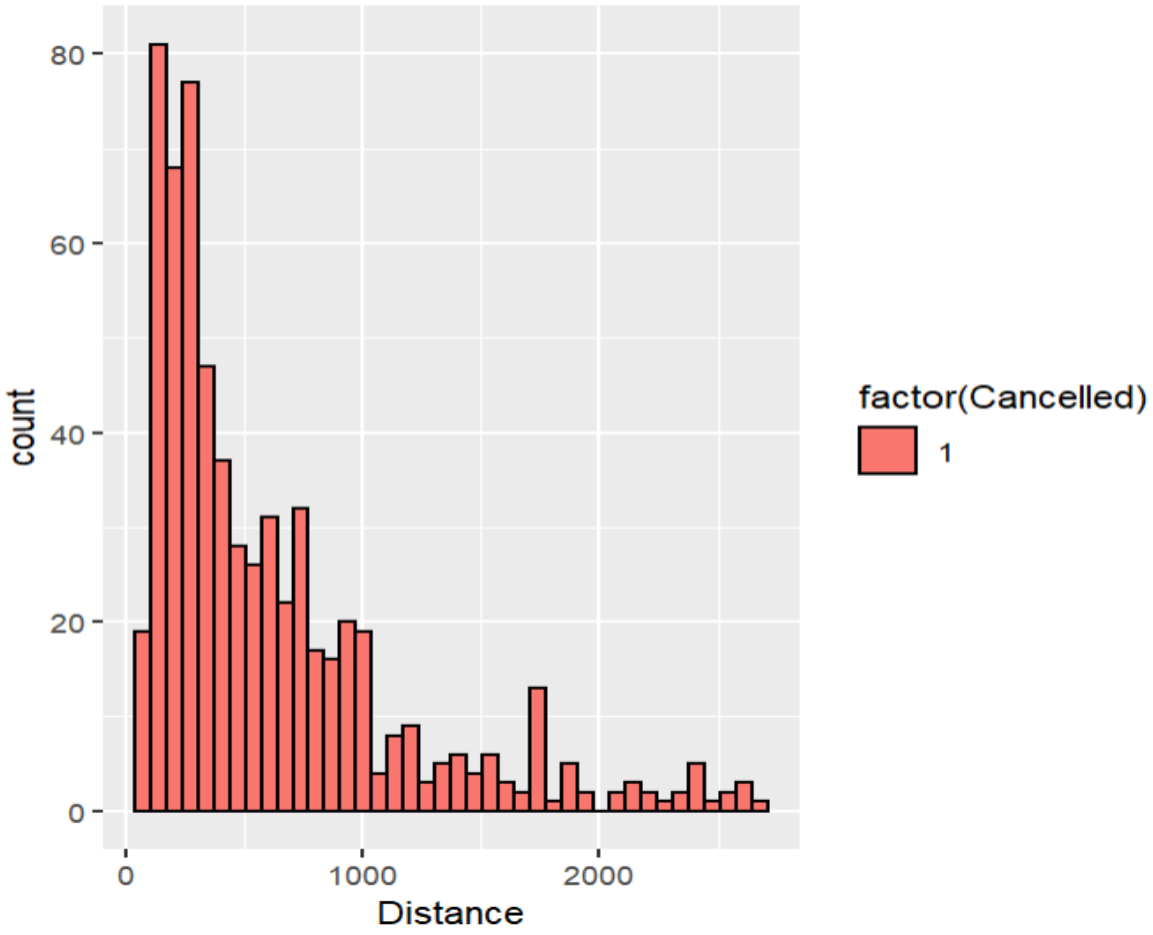
EDA2-Do longer or shorter flights tend to have more cancellations?

R.Script:

```
#1: more short distance flight so more cancellation, or short distance flight are going to generate more cancellation
#since DF_NA_cleared contains no cancellation(because there are so few of them), we will use the original dataset:DF
library(dplyr)
library(ggplot2)
Distance_Cancelled <- DF %>% select(Distance, Cancelled)
EDA2_only_cancellations <- subset(Distance_Cancelled, Cancelled > 0) # create a subset with all of cancellation = 1
summary(EDA2_only_cancellations$Cancelled) # check if all of them are Cancelled = 1
EDA2 <- ggplot(EDA2_only_cancellations, aes(Distance)) + geom_histogram(aes(fill = factor(Cancelled)), bins = 40, color = "black")
EDA2
```

#shorter flights have more cancellations or there are just too much short flight

PLOT:



The count of cancellations based upon distance

Explanation:

Although Cancellations can be seen throughout all the flights, we find shorter distance flights have more cancellations, while there are longer flights which are canceled but the number is very less.

Fuel Charges, Crew and maintenance, PerHead charges, Carrying capacities might be few reasons for the shorter flights to get canceled more is what we determine from our observation.

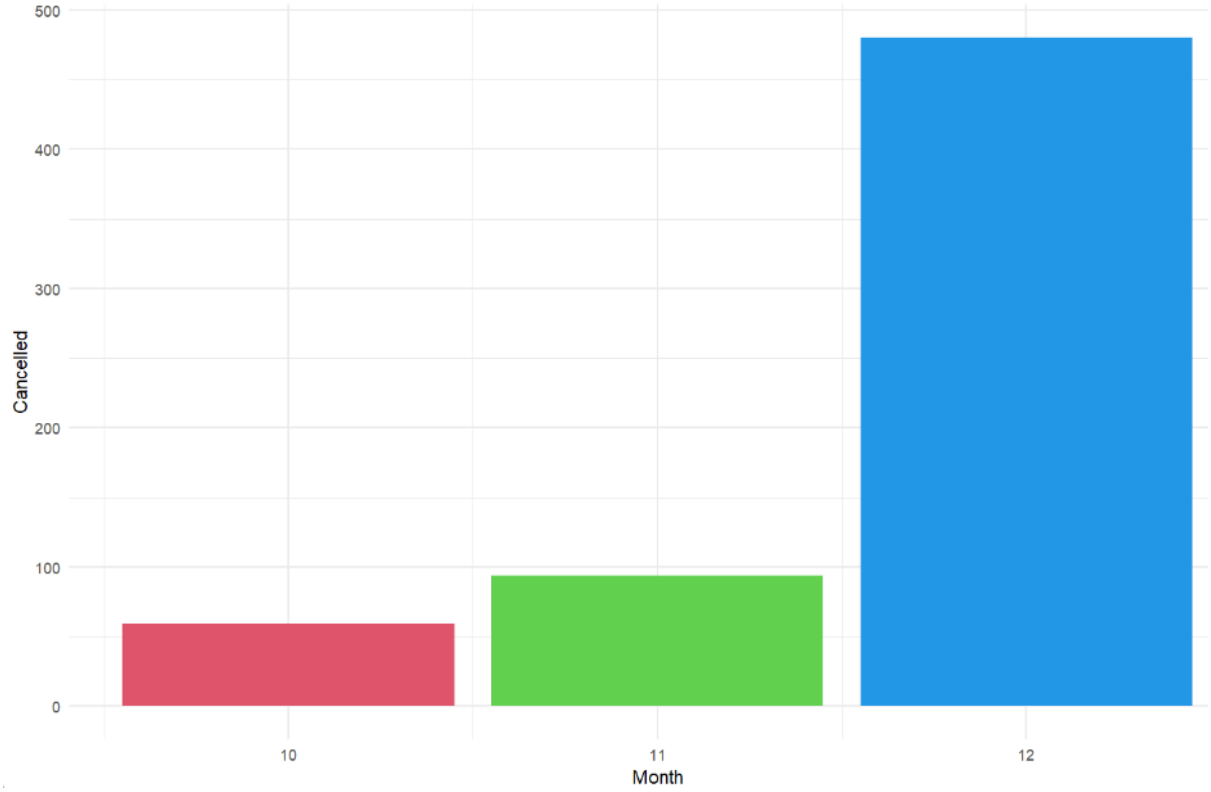
EDA3 -Which month tends to have the most cancellations?

R-Script:

```
# data_subset <- DF %>% group_by(Month) %>% summarize(Cancelled =
sum(Cancelled))
# ggplot(data_subset,aes(Cancelled)) + geom_histogram(aes(fill = factor(Month)))
```

```
library(ggplot2)
Month_Cancelled <- DF %>% select(Month, Cancelled)
EDA3_only_cancellations <- subset(Month_Cancelled, Cancelled > 0)
# getting the count of cancellation per month
EDA3 <- ggplot(data=EDA3_only_cancellations, aes(x=Month, y=Cancelled)) +
geom_bar(stat="identity", fill = factor(EDA3_only_cancellations$Month)) +
theme_minimal()
EDA3
table(Month_Cancelled$only_cancellations) # get the exact number of the
cancellations each month
#majority of the cancellation occur in the last three months of a year
# EDA3_A <- ggplot(data=Month_Cancelled, aes(x=Month, y=Cancelled)) +
#   geom_bar(stat="identity", fill = factor(Month_Cancelled$Month)) +
#   theme_minimal()
# EDA3_A
```

PLOT:



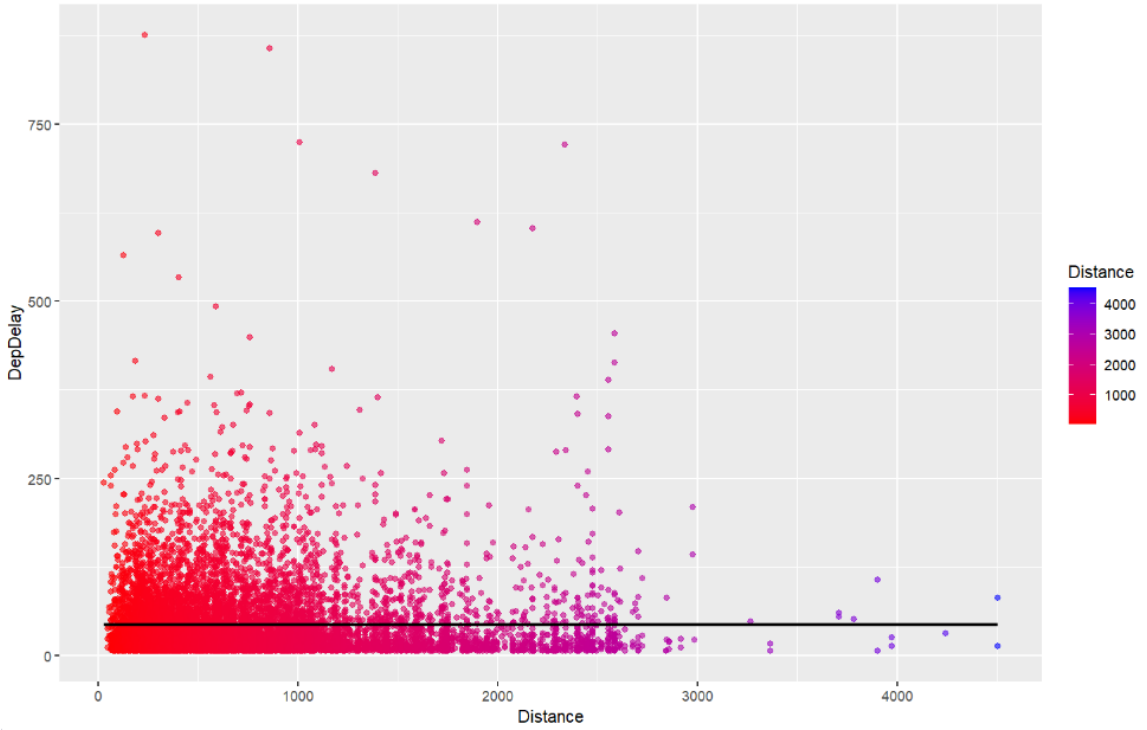
Explanation:

An interest thing we can tell from this plots is that all the cancellations happened in the last three month of a year, with the majority of flights are canceled in the month of December, since it is the end of the year. This could be due to poor weather conditions thus grounding flights, carriers having employees off for the holidays, and possible wear and tear on the aircraft due to hostile weather conditions.

EDA4-Do longer or shorter flights have more Departure Delay ?

R.Script:

```
length(unique(DF_NA_cleared$Dest))
table(DF_NA_cleared$Dest)#get the number of unique values in UniqueCarrier
EDA4_DepDelay <-  ggplot(DF_RD, mapping = aes(x = Distance, y = DepDelay, color = Distance) )+
geom_point(alpha = 0.6) +
scale_color_gradient(low = "red", high = "blue") + geom_smooth(col = "black")
EDA4_DepDelay
```

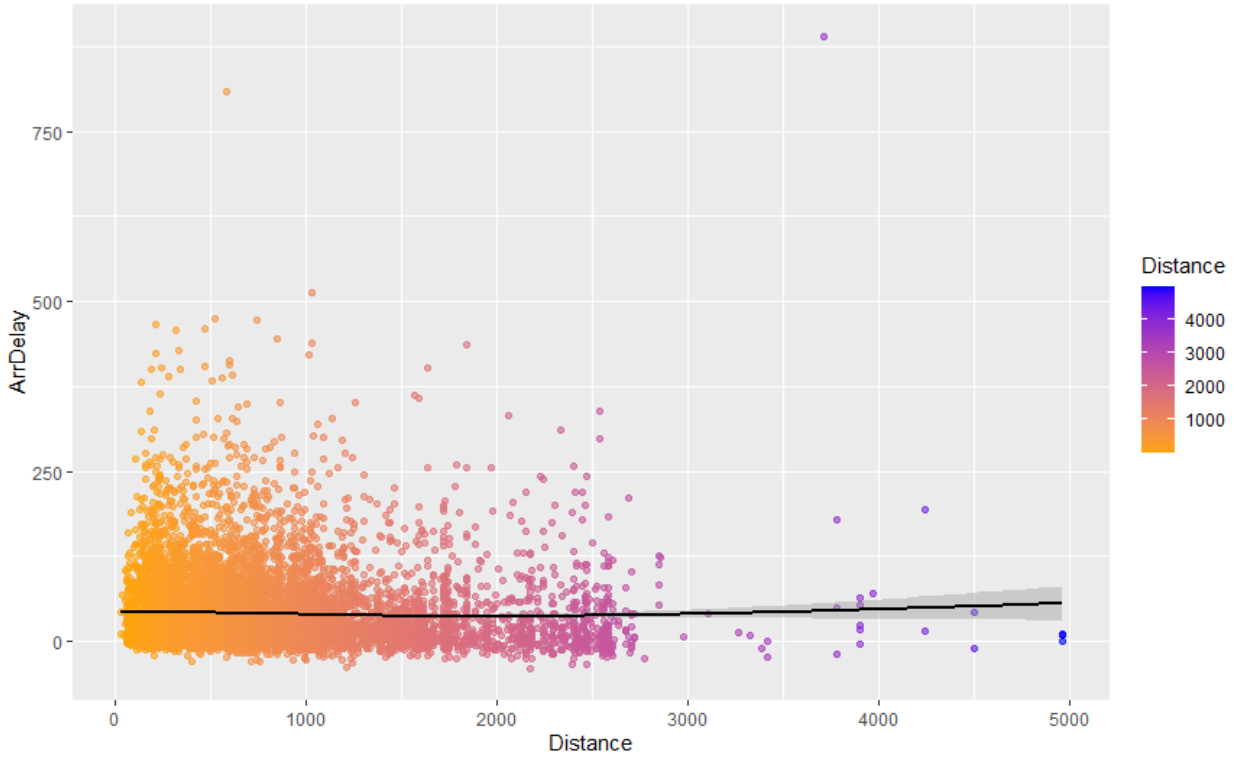


PLOT:

Units: Mins/Miles

EDA4.1 :Do longer or shorter flights have more Arrival Delay ?

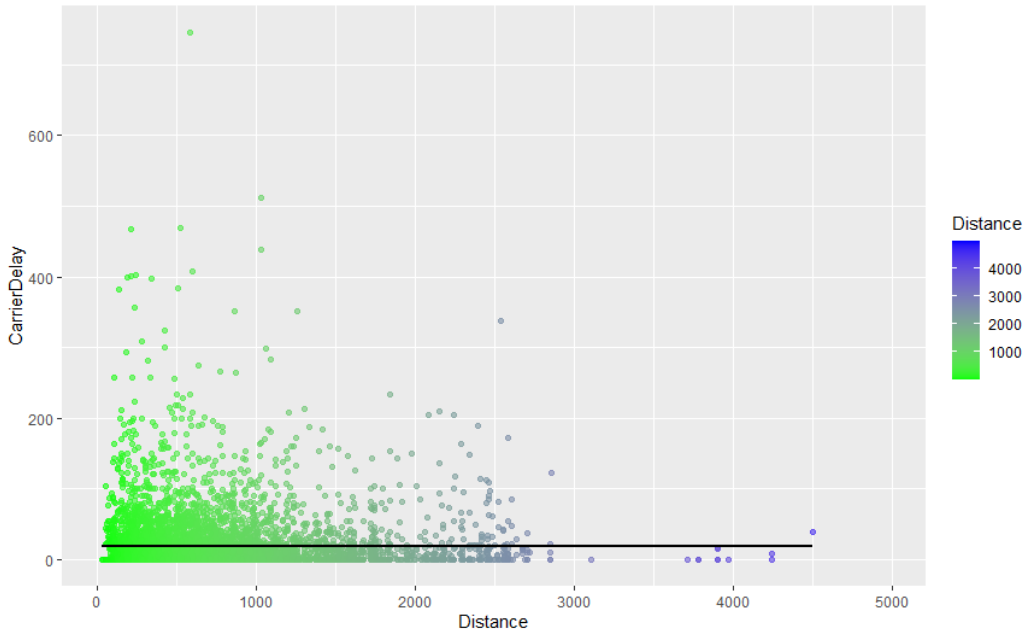
```
#----
attach(DF_RD)
EDA4_ArrDelay <-  ggplot(DF_RD, mapping = aes(x = Distance, y = ArrDelay, color = Distance) )+
geom_point(alpha = 0.6) +
scale_color_gradient(low = "orange", high = "blue") + geom_smooth(col = "black")
EDA4_ArrDelay
```



EDA4.2:Do longer or shorter flights have more Carrier Delay ?

#----

```
EDA4_CarrierDelay <-  ggplot(DF_RD, mapping = aes(x = Distance, y =
CarrierDelay, color = Distance) )+
geom_point(alpha = 0.6) +
scale_color_gradient(low = "green", high = "blue") + geom_smooth(col = "black")
EDA4_CarrierDelay
```



Explanation:

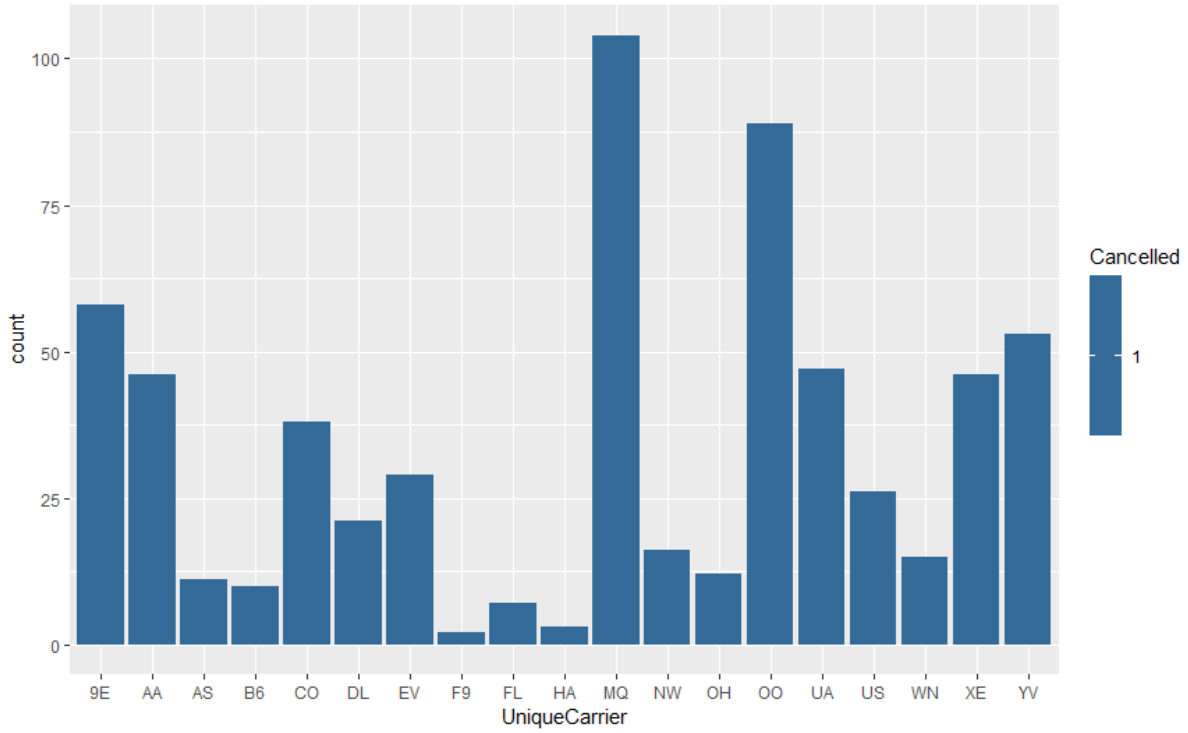
In this section of continued analysis, we try to get to the bottom of how different types of delays get impacted based on the distance traveled. Delays are interpreted in minutes and distance in miles. It does not seem like there is much of a correlation between the distance and the delays, but it was worth examining if there was a consistent issue with longer/more laborious flights.

EDA5-Which Carriers have more Cancellations?

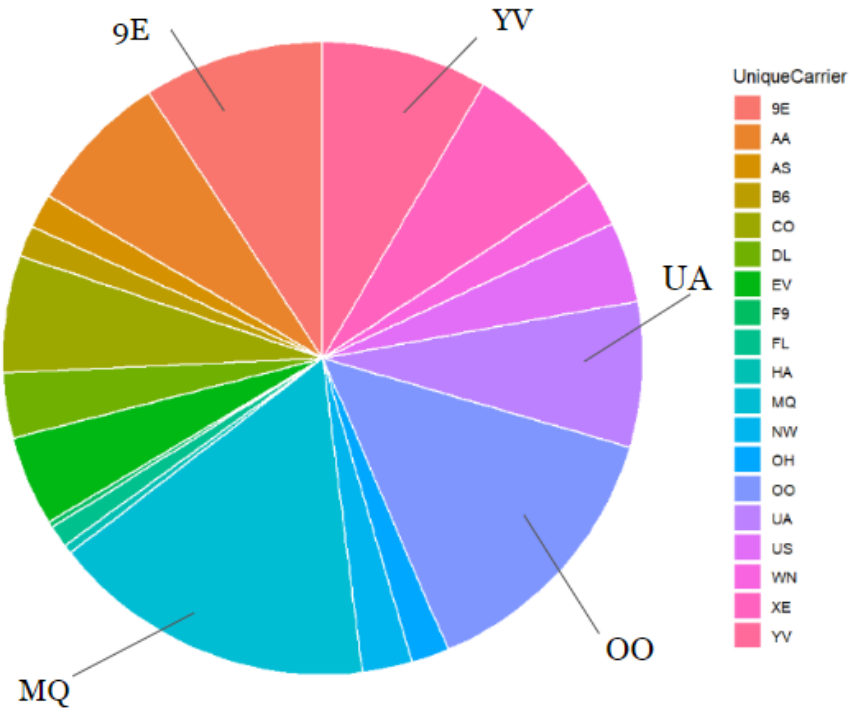
R-script:

```
library(dplyr)
attach(DF_NA_cleared)
UniqueCarrier_Cancelled <- DF %>% select(UniqueCarrier, Cancelled)
EDA5_only_cancelled <- subset(UniqueCarrier_Cancelled, Cancelled > 0)
attach(EDA5_only_cancelled)
cancelled_count <- EDA5_only_cancelled %>% count(UniqueCarrier) #count
Cancelled by group
attach(cancelled_count)
cancelled_count$cancel_percen = round(100*(n/sum(n)),2) #calculate percentage by
group and round it

# Basic piechart
library(ggplot2)
library(dplyr)
install.packages(scale)
attach(cancelled_count)
EDA5 <- ggplot(cancelled_count, aes(x="", y=cancel_percen, fill=UniqueCarrier)) +
  geom_bar(stat="identity", width=1, color = 'white') +
  coord_polar("y", start=0)+
  theme_void()
```

PLOT:



PIECHART:



Explanation:
In the mentioned exploratory analysis, we try to figure out the flight carriers where most flights are canceled. This is done by creating a subset of the data set with just two parameters namely, UniqueCarrier and Canceled.

The count (number of canceled flights) is displayed on y-axis in line with the carrier names on y-axis, in the bar chart. The similar analysis is further explained via a piechart. For the piechart, the color variation explains the list of carriers in the data set.

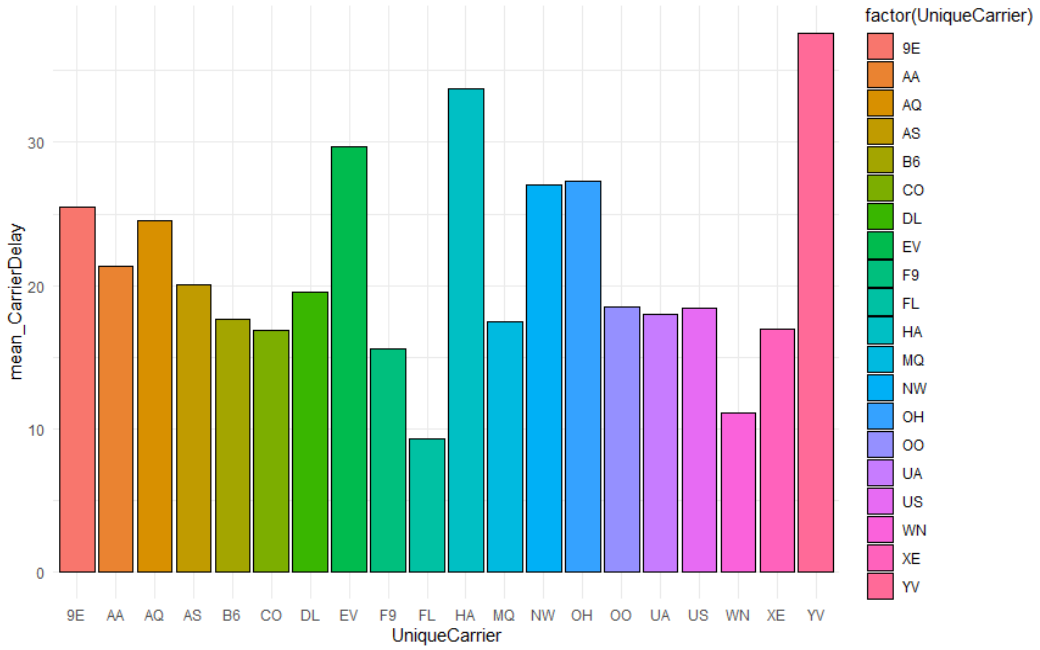
The top five carriers which have the most canceled flights in descending order are, MQ, OO, 9E, YV and UA. Top two carriers which have the least number of cancellations are F9 and HA respectively.

EDA6 Which carriers have More Delay ?

R-script:

```
#What carriers have more delay
UniqueCarrier_CarrierDelay <- DF_NA_cleared %>% select(UniqueCarrier,
CarrierDelay)
# Group by mean using dplyr
library(dplyr)
average_of_UniqueCarrier <- DF_NA_cleared %>% group_by(UniqueCarrier) %>%
  summarise(mean_CarrierDelay=mean(CarrierDelay),
            .groups = 'drop')
attach(average_of_UniqueCarrier)
EDA6 <- ggplot(data=average_of_UniqueCarrier, aes(x=UniqueCarrier,
y=mean_CarrierDelay, fill = factor(UniqueCarrier))) +
  geom_bar(stat = 'identity', col = 'black') +
  theme_minimal()
EDA6
```

PLOT:



Explanation

Here, we are trying to understand which carriers have the most delay. Taking the CarrierDelay parameter into account, we take the average values (mean) and determine a barplot. To get this done, we create a subset with two parameters, namely UniqueCarrier and applying Mean on CarrierDelay values.

Plotting carrier details on x-axis and delay minutes on y-axis, we interpret carrier-induced delay and also, it can be determined the carrier which has the most delay and the carrier which has the least. Carrier information in the plot is differentiated using a color scale.

After factoring the initial data set values of the CarrierDelay and plotting the same, we get the following, YV and HA are having the most delay whereas FL and WN are having the least delay, in that order.

Modeling:

Dataset is split into two parts to create the model and run it:
Training Data: This Dataset is a subset randomly drawn from the original cleaned dataset.
Testing Data: This Dataset is a subset remaining after taking out the training data from the original cleaned dataset.

We divide the data into training data and testing data with a split ratio of 0.7 on CarrierDelay.

R-code:

```
set.seed(101)
split <- sample.split(DF_RD$CarrierDelay, SplitRatio = 0.7)
train = subset(DF_RD, split == TRUE)
test = subset(DF_RD, split == FALSE)
```

Apply algorithm:

Our aim is to create a model and predict the outcome values. We train the model to calculate TaxiOut based upon the CarrierDelay. We train the data with our splitted training data and summarize the model to find the estimates and the standard error. we find the coefficients of the data (Y-intercept and the slope of the linear model).

R-code:

```
attach(DF_RD)
model <- lm(TaxiOut~CarrierDelay, data = train)
summary(model)
model$coefficients
```

```
Call:
lm(formula = TaxiOut ~ CarrierDelay, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-19.655  -9.450  -4.655   3.345 204.345

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.654930   0.214089  96.478  < 2e-16 ***
CarrierDelay -0.013665   0.004588  -2.979  0.00291 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.29 on 7014 degrees of freedom
Multiple R-squared:  0.001263,  Adjusted R-squared:  0.001121
F-statistic: 8.872 on 1 and 7014 DF,  p-value: 0.002906
```

```
> model$coefficients
 (Intercept) CarrierDelay
 20.65493049  -0.01366547
```

Predict Outcome:

Now we predict the outcome based upon the model from training data and check predictions on the testing data. We store the outcomes as a result and convert into a dataframe to compare the testing data and the predicted outcome.

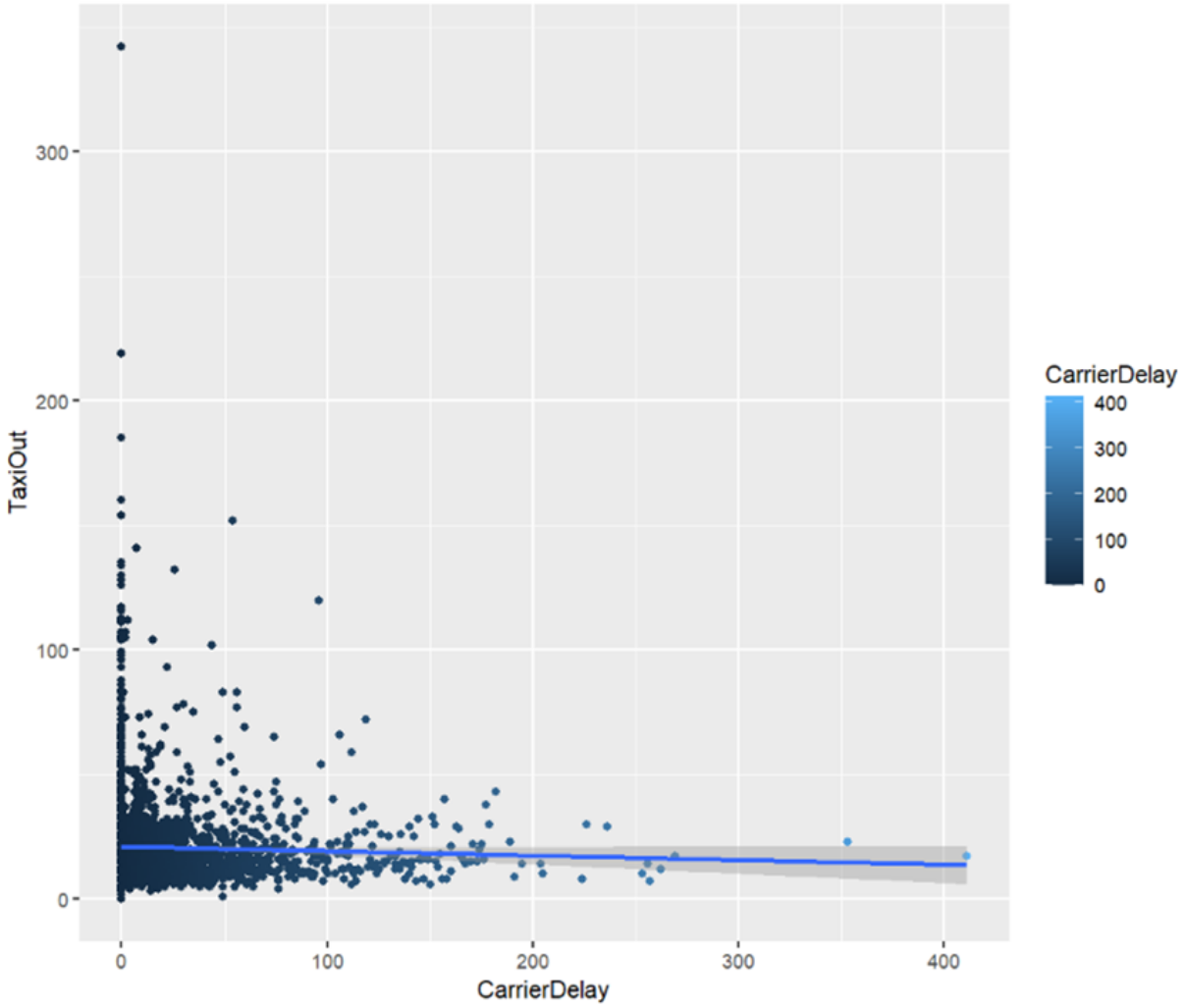
```
> TaxiOut.predictions <- predict(model,test)
> results <- cbind(TaxiOut.predictions,test$TaxiOut)
> colnames(results) <- c('pred','real')
> results <- as.data.frame(results)
> print(head(results))
       pred real
5  20.65493   27
9  20.43628    9
12 20.65493   25
14 20.65493   11
15 20.65493   18
18 20.65493   10
```

R-code:

```
# prediction
TaxiOut.predictions <- predict(model,test)
results <- cbind(TaxiOut.predictions,test$TaxiOut)
colnames(results) <- c('pred','real')
results <- as.data.frame(results)
print(head(results))
```

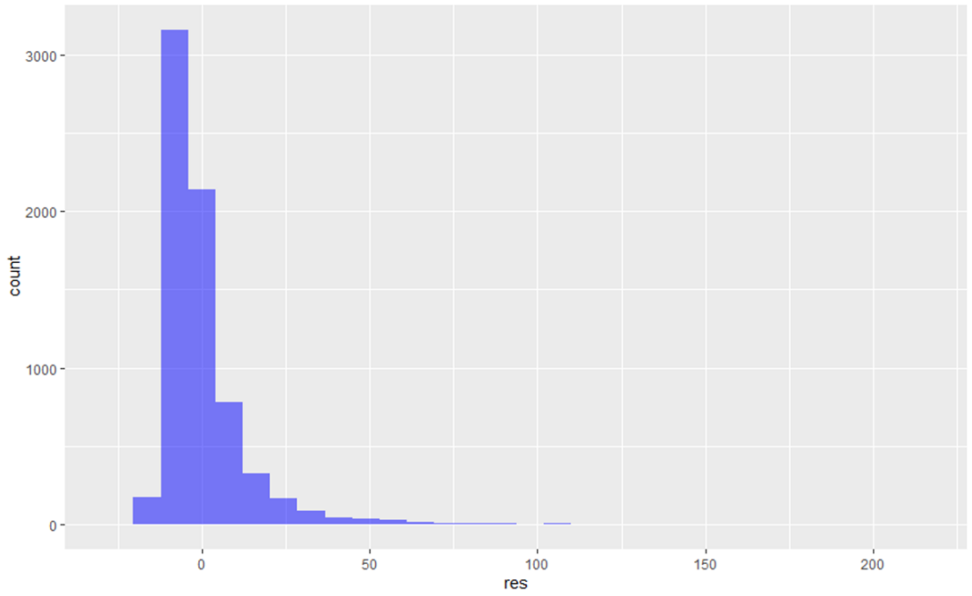
Visualize Model and residuals:

We scatterplot the values of TaxiOut on the CarrierDelay and plot the linear model we trained. We successfully plotted the model and observed a downward trend in the regression line. We also plotted the residuals histogram. Residuals are the difference between actual datapoint and predicted linear regression model results. We always try to minimize the residual value during the training of our model. There is a low R2 value and a low P value which points to a certain level of significance that carrier delay (x) has upon taxi out (y). With that being stated, it is important to note that low p value and low R2 value models generally mean we can still draw a relationship between the variables, but there will be a higher level of variability.



R-code:

```
scatter <- ggplot(test, aes(x = CarrierDelay, y = TaxiOut, color = CarrierDelay)) +
geom_point() + stat_smooth(method = "lm")
scatter
res <- residuals(model)
class(res)
res <- as.data.frame(res)
head(res)
ggplot(res,aes(res)) + geom_histogram(fill='blue',alpha=0.5)
```



Conclusion

In conclusion, the dataset that we utilized had a lot of difficulty that comes with it: the dataset contains 1.9 million records, making it a burden to run, so we decided to sample it. One of the interesting observations is that since it contains a lot of NA values, some of the important variables we need for our models are clear because the same row contains NA values, and this really contributes to the difficulty in analyzing this model. The description of where the dataset is provided does not match what it actually looks like. Therefore an analysis on this dataset is challenge but we manage to generate some descriptive analytic and ultimately get a model that can consistently predict a downward trend or inverse relationship between the carrier delay and TaxiOut, our R square value and p-value are both low and in this case a trend can still be established. However, predictions will have a high degree of variability which could be due to other unexplained variables influencing our y variable. The real world implications of this model mean that there is a relationship between the carrier delays (malfunctioning machinery, possible overbooking, employee attendance issues, etc) and the amount of time it takes for the aircraft to leave the runway once it is ready for departure (taxi out).