Exploration of Flight Arrival Delays in 2019

Inna William

Contents

1. Introduction………………………………………………………………………...3
2. Description Of data………………………………………………………………...4
3. Discussion/Result…………………………………………………………………..6
4. Summary…………………………………………………………………………...10
5. Appendices………………………………………………………………………....11

**Introduction**

When considering airfare travel, some of the most important aspects are ticket prices, travel hours, and numbers of layovers. People rarely think about the possibility of delays and cancellations of diverted flights. Flight cancellations and delays can have serious negative effects on passengers, especially when costs are significant. According to Baumeister et al. (2001), events “that are negatively valenced...will have a greater impact on the individual than positively valenced events of the same type” (p. 323). Flight delay is considered to be a serious and widespread problem in the United States. Increasing flight delays also places a significant burden on the U.S. air travel system and society in general in terms of cost (Zou & Hansen, 2010, p. 1). For instance, in 2007 the U.S. Congress found that flight delays cost the U.S. economy about $40.1 billion. Delays burned additionally about 740 million gallons of fuel and released about 7.1 million metric tons of carbon dioxide into the atmosphere during flights (Blackwood (2001) p. 8). In 2017 the estimated annual cost of flight delays for passengers was $26.6 billion. This huge wastes and harms to the environment draws our attention to further analyze the flight delays problem.

In the United States, airlines are not always required to compensate passengers when flights are delayed or cancelled. Airlines are only required, by U.S. law, to compensate certain passengers if they are bumped from a flight that is at capacity (U.S. Department of Transportation, 2018). Studying delays and cancellation is important because of the emotional and monetary cost involved for passengers, airlines and society. Travellers need to be aware of the frequency of flight delays and cancellations when planning their travel. While airlines should realize that on-time arrival is one of the major factors of the customer experience, which satisfied customers will become loyal to the company, provide positive reviews and help increase profits (Dyer, 2016).

**Description of Data**

Dataset used in the project was obtained on Kaggle.com. This dataset contains 583,987 rows, 64 columns. The name of the dataset is On\_Time\_Reporting\_Carrier\_On\_Time\_Performance\_manipulated.csv. Here is the link to access the data <https://www.kaggle.com/aditideokar/2019-flight-delays-data> .

The purpose of the study is to explore this dataset, find the correlations between the variables,

and fit regression models to predict Arrival Delay in Minutes.

Only the following column numbers of the original dataset were used in exploration:

4, 16, 25, 31, 32, 34, 42, 43, 45, 54, 55, 57, 60 corresponding to

**"DayOfWeek" , "OriginState", "DestState", "DepDelay", "DepDelayMinutes", "DepartureDelayGroups", "ArrDelay", "ArrDelayMinutes", "ArrivalDelayGroups", "Distance", "DistanceGroup", “WeatherDelay”, “LateAircraftDelay”.**

This dataset is noted to have missing values in every column of the dataset.In the process of working with this data, it was found that ArrDelayMinutes and DepDelayMinutes is ArrDelay and DepDelay with negative values were set to be zero. It was also found that **"DepartureDelayGroups" and "ArrivalDelayGroups",** each had 14 levels and 2 negative levels. For convenience, the level were shifted from -2 to 12 to 0 to 14. NAs that was found in these 2 groups were set to be the maximum number. We regroup the data (Table 1 & 2) but end up finding that might not be helpful to use them in the multivariate regression model due to collinearity problem.

Detailed descriptions of other unused variables in this dataset can refer to <https://www.transtats.bts.gov/Fields.asp?Table_ID=236>.

**Methods:**

Generally, we use R for the whole project. SAS is used as a secondary language. We split the dataset into train (80%) and test dataset (20%) randomly. Then build up a correlation analysis to choose which variable(s) to use. We use lm and glm function in R to build the model. The built-in plots help us manually check the assumptions. Since we want to be more careful about the assumptions, we use gvlma function to do an overall test for linearity, independence, normality, and outliers etc.

**Discussion**

**Correlation Analysis**

In order to find the most important variable, we analyse the correlation between each variable and ArrDelayMinute. The correlation matrix is shown in Figure 1. From the findings of this figure, we see variables DepDelayMinutes have the strongest correlation with ArrDelayMinutes, which are greater than 0.6. So we select DepDelayMinutes as the predictor to start our simple linear regression model analysis. To further investigate the Pearson correlation coefficient, R indicates a very strong correlation 0.9776886 between these two variables. For our multivariate regression model, we use DepDelayMinutes, Distance, WeatherDelay, LateAircraftDelay to build our model. A similar correlation matrix (Figure 2 in appendix) as Figure 1 but with specified Pearson correlation coefficients is constructed for better understanding about the correlations. Also, we try to detect any multicollinearity problem among variables. None of the VIFs (Variance Inflation Factor) are close to 10, which means the four variables in the multivariate models would not cause the multicollinearity problem (Table 3).

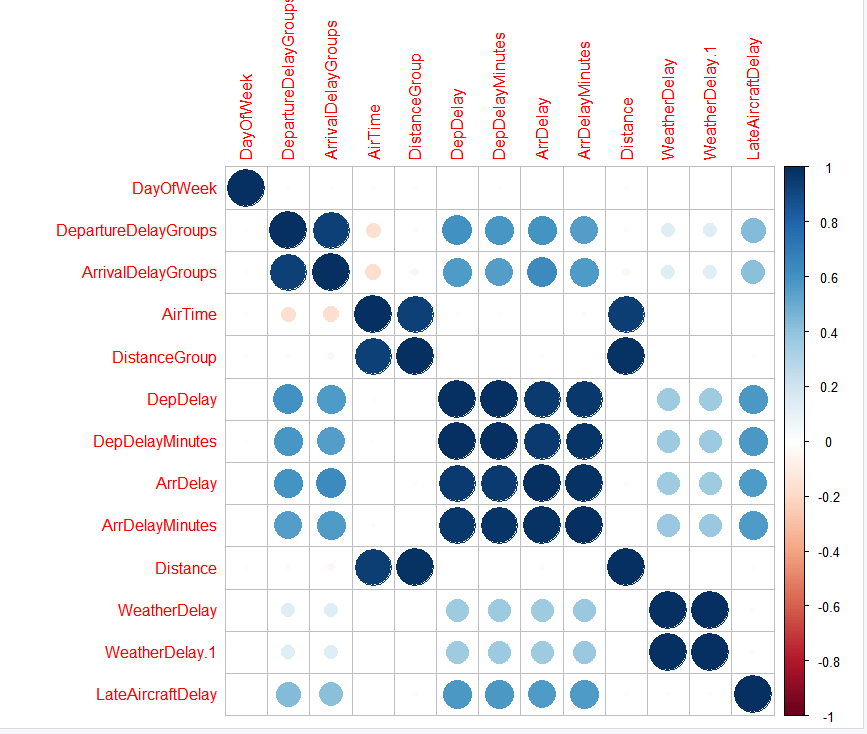


Figure 1: Correlation Matrix between different variables

train\_4vars$DepDelayMinutes train\_4vars$Distance

1.554985 1.003638

train\_4vars$WeatherDelay train\_4vars$LateAircraftDelay

1.204097 1.366676

Table 3: Output of Variance Inflation Factors for Multivariate Linear Regression Model

**Simple Linear Regression Model**

Fitted model:

**Estimated ArrDelayMinutes = 0.7179831 + 0.9815810 \* DepDelayMinutes**

This model means that arrival delay minutes is estimated to increase by 0.981581 for every minute increase in departure delay holding constant. Below is the detailed output (Table 4) and diagnostic plots (Figure 3) can be found in the appendix. Due to the huge sample size, we find that the diagnostic plots may not look perfect to the assumptions. However, we decide to ignore the imperfect diagnostic results after comparing to the model fitted with the test data, as b0 (0.7170613) and b1(0.9829974) for the test data barely change from the train one.

Call:

lm(formula = train$ArrDelayMinutes ~ train$DepDelayMinutes, data = train)

Residuals:

Min 1Q Median 3Q Max

-86.116 -0.718 -0.718 -0.718 312.411

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.7179831 0.0152803 46.99 <2e-16 \*\*\*

train$DepDelayMinutes 0.9815810 0.0003135 3131.47 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.901 on 452668 degrees of freedom

(1347 observations deleted due to missingness)

Multiple R-squared: 0.9559, Adjusted R-squared: 0.9559

F-statistic: 9.806e+06 on 1 and 452668 DF, p-value: < 2.2e-16

Table 4: Detailed Output of the Simple Linear Regression Model

The following plot (Figure 4) of DepDelayMinutes and ArrDelayMinutes shows a positive linear relationship with almost all dots being close to the regression line, which suggests constant variance. Constant variance assumption is consistent with the result we get from the gvlma function in R (Table 5 in appendix).

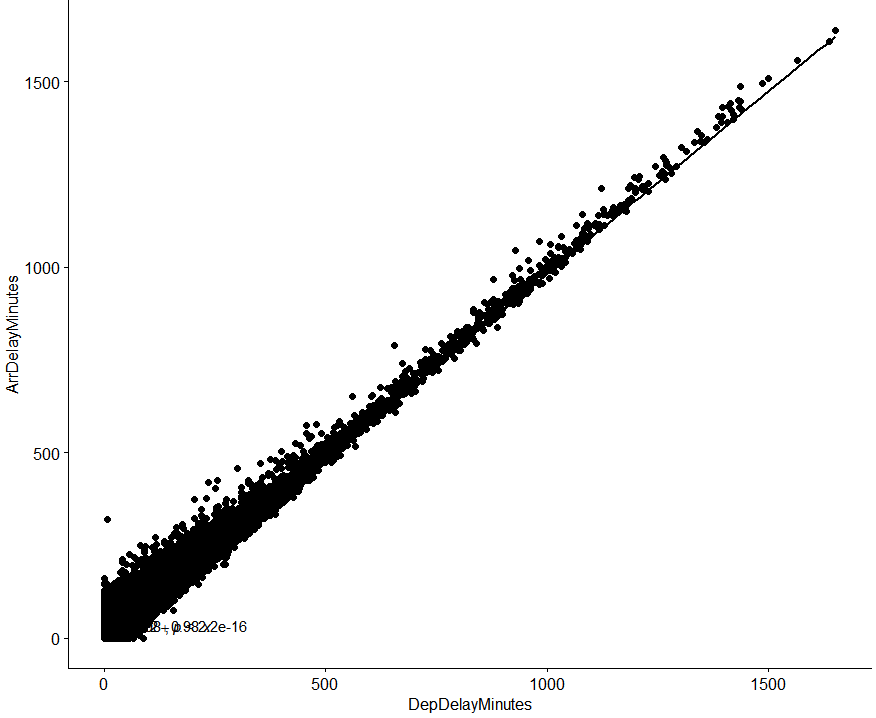


Figure 4: regression of DepDelayMinuntes and ArrDelayMinutes

**Multiple Regression Model**

In order to explore if other variables also contribute greatly to predict arrival delay minutes, we try to fit a model with four variables DepDelayMinutes, WeatherDelay, LateAircraftDelay, and Distance. By using glm function in R, the multivariate fitted model for the train data is

**Estimated ArrDelayMinutes = 9.5168783 + 0.9528274 \* DepDelayMinutes** - **0.0007236 \* Distance + 0.0436415 \* WeatherDelay – 0.0309165 \* LateAircraftDelay.** Output (Table 6) can be found below.

Call:

glm(formula = train$ArrDelayMinutes ~ train$DepDelayMinutes +

train$Distance + train$WeatherDelay + train$LateAircraftDelay)

Deviance Residuals:

Min 1Q Median 3Q Max

-66.304 -13.253 -2.491 10.008 304.209

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 9.5168783 0.1239350 76.789 < 2e-16 \*\*\*

train$DepDelayMinutes 0.9528274 0.0008905 1069.978 < 2e-16 \*\*\*

train$Distance -0.0007236 0.0001141 -6.343 2.27e-10 \*\*\*

train$WeatherDelay 0.0436415 0.0019938 21.888 < 2e-16 \*\*\*

train$LateAircraftDelay -0.0309165 0.0015539 -19.896 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 375.9753)

Null deviance: 698875957 on 84147 degrees of freedom

Residual deviance: 31635690 on 84143 degrees of freedom

(369869 observations deleted due to missingness)

AIC: 737766

Table 6: Output of the Multivariate Regression Model

We use gvlma function in R to check the assumptions for the regression model (Table 7 & Figure 5). Unfortunately, we get the unsatisfied results of all assumptions. This suggests us that we should do some transformation of the raw data. On the other hand, we also think that the large sample size is another factor for assumptions unsatisfied.

Call:

lm(formula = train\_4vars$ArrDelayMinutes ~ train\_4vars$DepDelayMinutes +

train\_4vars$Distance + train\_4vars$WeatherDelay + train\_4vars$LateAircraftDelay)

Coefficients:

(Intercept) train\_4vars$DepDelayMinutes

9.5168783 0.9528274

train\_4vars$Distance train\_4vars$WeatherDelay

-0.0007236 0.0436415

train\_4vars$LateAircraftDelay

-0.0309165

ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS

USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:

Level of Significance = 0.05

Call:

gvlma(x = mmultimodel)

Value p-value Decision

Global Stat 96541.14 0.0000000 Assumptions NOT satisfied!

Skewness 21895.86 0.0000000 Assumptions NOT satisfied!

Kurtosis 70123.70 0.0000000 Assumptions NOT satisfied!

Link Function 4510.23 0.0000000 Assumptions NOT satisfied!

Heteroscedasticity 11.36 0.0007521 Assumptions NOT satisfied!

Table 7: Diagnostic Outputs for Multivariate Regression Model

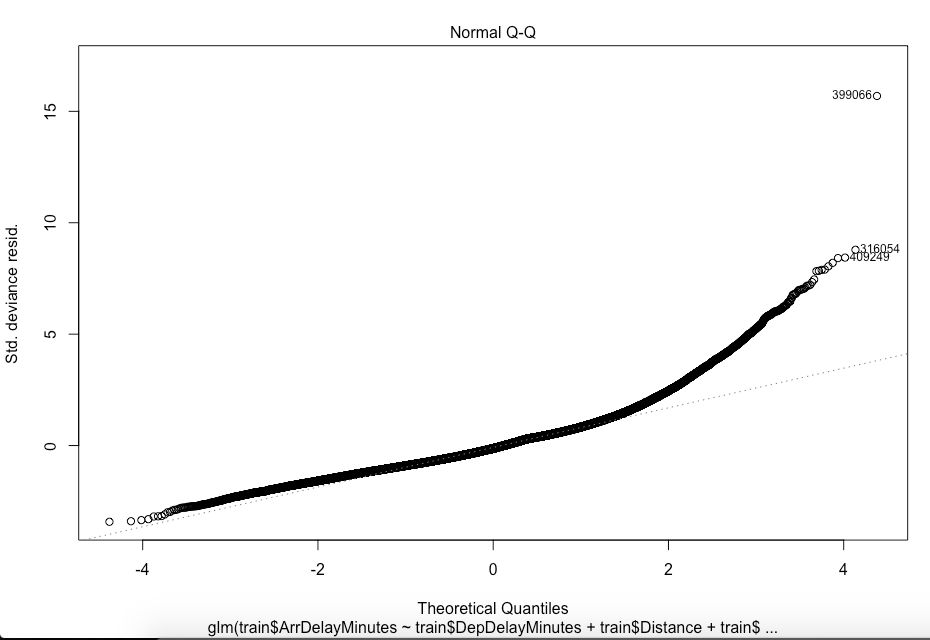


Figure 5: Normal QQ-plot of Multivariate Model

**Summary**

Based on our discussion results, we found ArrivalDelayMinutes has a strong correlation with DepartureDelayMinutes. A simple linear relationship has been built up:

**Estimated ArrDelayMinutes = 0.7179831 + 0.9815810 \* DepDelayMinutes.** With an increasing departure delay minute holding constant, a corresponding 0.981581 minute of arrival delay increases. Since the multivariate model does not satisfy any assumptions for regression, we prefer the simple regression model than the multivariate one. As what is mentioned in discussion, conducting a transformation of the data might give us a better model. Probably, filtering the sample size in a proper way might also do good to our purpose.

**Appendices**

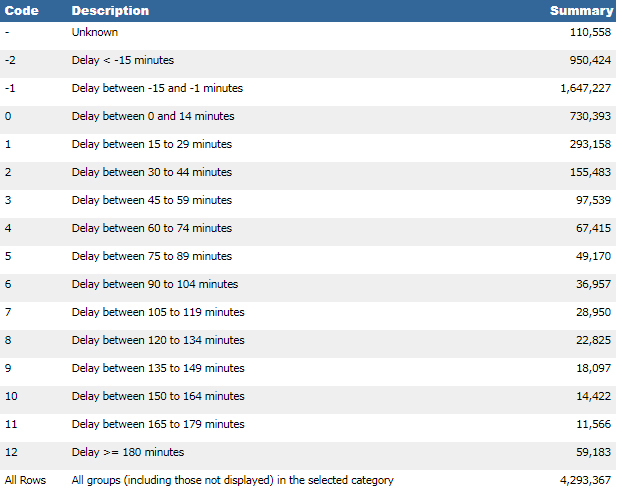


Table 1: dividing Delay Minutes into 12 groups

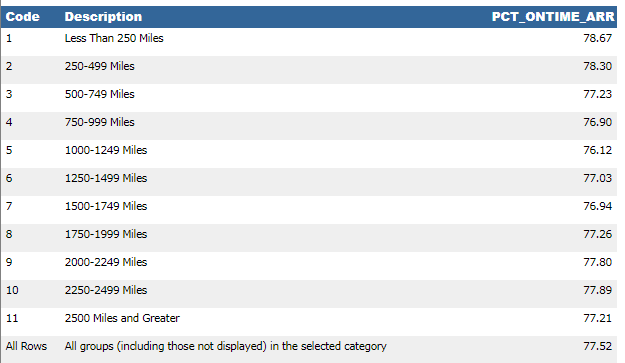


Table 2: dividing Distance to 11 groups

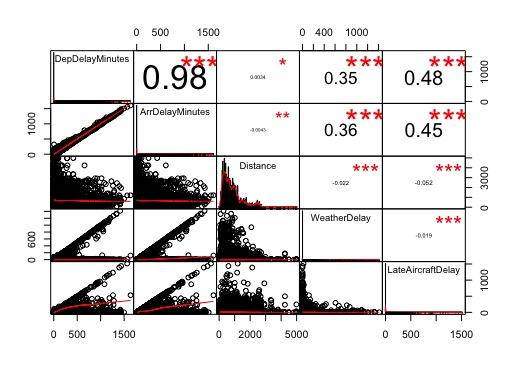


Figure 2: Correlation Matrix (Multivariate) with Specified Coefficients

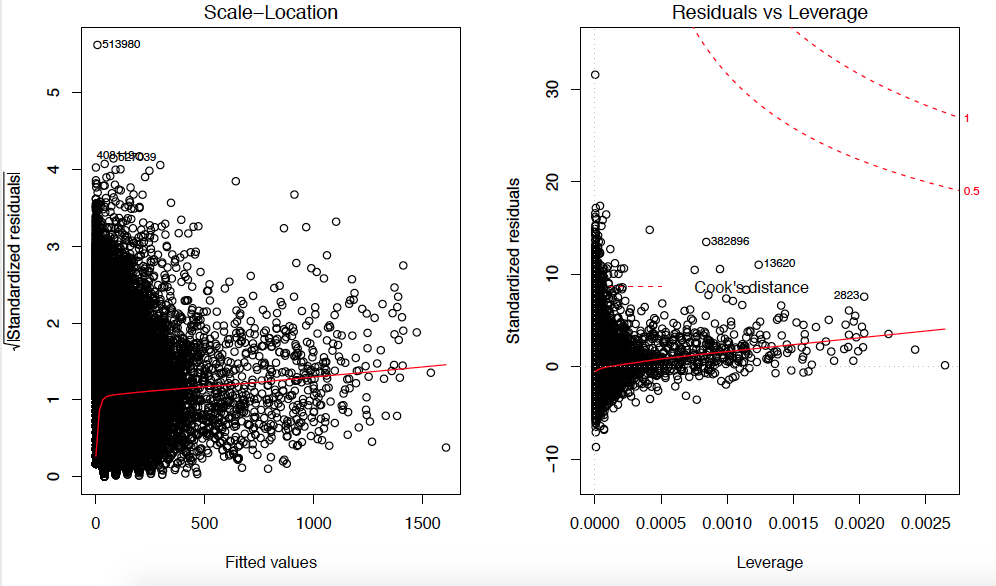
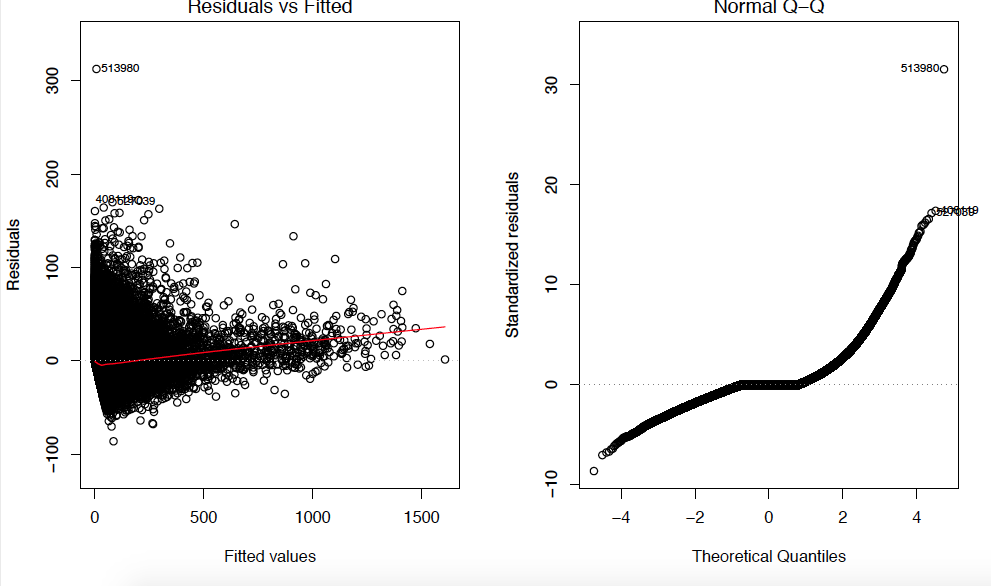


Figure 3: Diagnostic Plots of the Simple Linear Model

Call:

lm(formula = train$ArrDelayMinutes ~ train$DepDelayMinutes, data = train)

Coefficients:

(Intercept) train$DepDelayMinutes

0.7180 0.9816

ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS

USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:

Level of Significance = 0.05

Call:

gvlma(x = simplemodel)

Value p-value Decision

Global Stat 1.119e+07 0.0000 Assumptions NOT satisfied!

Skewness 5.826e+05 0.0000 Assumptions NOT satisfied!

Kurtosis 1.060e+07 0.0000 Assumptions NOT satisfied!

Link Function 4.077e+03 0.0000 Assumptions NOT satisfied!

Heteroscedasticity 2.001e+00 0.1572 Assumptions acceptable.

Table 5: Output of Assumptions Check for the Simple Linear Regression Model

To run the code some packages require  **jdk-10.0.1\_windows-x64\_bin.exe** needs to be installed (64 bits) on your computer

**Code**

# Import the data.

> delays<-read.csv(file = "/Users/christinechen/Desktop/MAT 449/On\_Time\_Reporting\_Carrier\_On\_Time\_Performance\_manipulated.csv", header = T)

# Filter the data.

> library(dplyr)

> delays\_filter<-select(delays,DayOfWeek, OriginState, DestState, DepDelay, DepDelayMinutes, DepartureDelayGroups, ArrDelay, ArrDelayMinutes, ArrivalDelayGroups, Distance, DistanceGroup, WeatherDelay, LateAircraftDelay)

> dim(delays\_filter)

[1] 583985 13

> newdelaysfilter<-subset(delays\_filter,delays\_filter$DepDelayMinutes!="" | delays\_filter$ArrDelayMinutes!="")

> dim(newdelaysfilter)

[1] 567630 13

# Use a runif to randomly split the data into 2 sets.

> blank<-matrix(runif(567630),ncol=1,nrow=567630)

> new\_blank<-cbind(blank,newdelaysfilter)

> b<-subset(new\_blank,new\_blank[ ,1]<=0.8)

> length(b$DayOfWeek)

[1] 454017

> a<-subset(new\_blank,new\_blank[ ,1]>0.8)

> length(a$DayOfWeek)

[1] 113613

> train<-select(b,-blank)

> dim(train)

[1] 454017 13

> test<-select(a,-blank)

> dim(test)

[1] 113613 13

# Fit simple linear regression model for the train dataset.

> library(lmtest)

> simplemodel<-lm(train$ArrDelayMinutes~train$DepDelayMinutes,data = train)

> summary(simplemodel)

# Output can be found above.

# Fit with the test data.

> simplemodel\_test<-lm(test$ArrDelayMinutes~test$DepDelayMinutes,data = test)

> summary(simplemodel\_test)

Call:

lm(formula = test$ArrDelayMinutes ~ test$DepDelayMinutes, data = test)

Residuals:

Min 1Q Median 3Q Max

-78.065 -0.717 -0.717 -0.717 189.279

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.7170613 0.0306543 23.39 <2e-16 \*\*\*

test$DepDelayMinutes 0.9829974 0.0006072 1618.99 <2e-16 \*\*\*

# Correlation coefficient test for the simple regression train data.

> cor.test(train$DepDelayMinutes,train$ArrDelayMinutes,method="pearson")

Pearson's product-moment correlation

data: train$DepDelayMinutes and train$ArrDelayMinutes

t = 3131.5, df = 452670, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.9775597 0.9778168

sample estimates:

cor

0.9776886

# Use gvlma function to check assumptions for simple regression.

> install.packages("gvlma")

> library(gvlma)

> gvsimtrain<-gvlma(simplemodel)

> gvsimtrain

Call:

gvlma(x = simplemodel)

Value p-value Decision

Global Stat 1.119e+07 0.0000 Assumptions NOT satisfied!

Skewness 5.826e+05 0.0000 Assumptions NOT satisfied!

Kurtosis 1.060e+07 0.0000 Assumptions NOT satisfied!

Link Function 4.077e+03 0.0000 Assumptions NOT satisfied!

Heteroscedasticity 2.001e+00 0.1572 Assumptions acceptable.

# Diagnostic plots for simple regression.

> par(mfrow=c(2,2))

> plot(simplemodel)

# Fit multivariate linear regression model. Output can be found above.

>multimodel<-glm(train$ArrDelayMinutes~train$DepDelayMinutes+train$Distance+train$WeatherDelay+train$LateAircraftDelay)

> summary(multimodel)

# Construct a correlation matrix for multiple variables.

> install.packages("PerformanceAnalytics")

> library(PerformanceAnalytics)

> chart.Correlation(train\_4vars, histogram = TRUE, method="pearson")

# Use vif function to check multicollinearity.

> library(car)

> car::vif(multimodel)

# Check assumptions by using gvlma function. Multimodel gotten from lm function is the same as the multimodel gotten from glm.

>mmultimodel<-lm(train\_4vars$ArrDelayMinutes~train\_4vars$DepDelayMinutes+train\_4vars$Distance+train\_4vars$WeatherDelay+train\_4vars$LateAircraftDelay)

> gvmultrain<-gvlma(mmultimodel)

> gvmultrain

# The following codes are used for constructing the correlation matrix and ggplots that help us visualize the data in the early beginning.

install.packages(aod)

install.packages(aod3)

install.packages("Hmisc")

install.packages("corrplot")

install.packages("corrgram")

library(aod)

library(corrplot)

library("Hmisc")

library("ggpubr")

library(ggplot2)

library(data.table)

library(corrplot)

par(mar=c(1,1,1,1))

dff <- read.csv("F:/project/delays/On\_Time\_Reporting\_Carrier\_On\_Time\_Performance\_manipulated.csv")[ ,c(16,25,4,34,45,52,55,31,32,42,43,54,57,57,60)]

dff = as.data.frame(dff)

names(dff)

dff$ArrivalDelayGroups[is.na(dff$ArrivalDelayGroups)] <- 13

dff$DepartureDelayGroups[is.na(dff$DepartureDelayGroups)] <- 13

dff<-dff[(dff$ArrivalDelayGroups < 13 ),]

groupadd=min(dff$ArrivalDelayGroups)

dff[,5]=dff[,5]-groupadd

dff[,4]=dff[,4]-groupadd

dff[is.na(dff)] <- 0

head(dff)

summary(dff)

str(dff)

unique(dff$DepartureDelayGroups)

unique(dff$DistanceGroup)

unique((dff$ArrivalDelayGroups))

unique(dff$DayOfWeek)

unique(dff$OriginState)

unique(dff$DestState)

x <- dff[3:15]

y <- dff[3:15]

dff.cor=cor(x, y)

palette = colorRampPalette(c("green", "white", "red","brown","blue","darkred",

"yellow","gray","black","pink","orange","cyan")) (12)

heatmap(x = dff.cor, col = palette, symm = TRUE)

corrplot(dff.cor)

pl1=ggplot(dff, aes(x = dff$ArrDelayMinutes)) +

geom\_histogram(colour="darkblue", size=1, fill="red")

pl2=ggplot(dff, aes(x = dff$DepDelayMinutes)) +

geom\_histogram(colour="darkblue", size=1, fill="red")

pl3=ggplot(dff, aes(x = dff$DistanceGroup)) +

geom\_histogram(colour="darkblue", size=1, fill="blue")

pl4=ggplot(dff, aes(x = dff$Distance)) +

geom\_histogram(colour="darkblue", size=1, fill="green")

pl5=ggplot(dff, aes(x = dff$ArrivalDelayGroups)) +

geom\_histogram(colour="darkblue", size=1, fill="blue") + scale\_x\_log10()

pl6=ggplot(dff, aes(x = dff$DepartureDelayGroups)) +

geom\_histogram(colour="darkblue", size=1, fill="blue") + scale\_x\_log10()

ggarrange(pl1, pl2, pl3,pl4,pl5,pl6 + rremove("x.text"),

labels = c("A", "B", "C","D", "E", "F","G","H"),

ncol = 2, nrow = 3)

bp1=ggplot(dff, aes(x=factor(dff$ArrivalDelayGroups),

y=dff$ArrDelayMinutes,color=dff$ArrivalDelayGroups)) +

geom\_boxplot(outlier.colour="red", outlier.shape=2,outlier.size=1)

bp2=ggplot(dff, aes(x=factor(dff$DepartureDelayGroups),

y=dff$ArrDelayMinutes,color=dff$DepartureDelayGroups)) +

geom\_boxplot(outlier.shape=NA)

bp3=ggplot(dff, aes(x=factor(dff$DayOfWeek),

y=dff$ArrDelayMinutes,color=dff$DayOfWeek)) +

geom\_boxplot(outlier.shape=NA)

bp4=ggplot(dff, aes(x=factor(dff$DistanceGroup),

y=dff$ArrDelayMinutes,color=dff$DistanceGroup)) +

geom\_boxplot(outlier.shape=NA)

bp5=ggplot(dff, aes(x=dff$OriginState,

y=dff$ArrDelayMinutes,color=dff$OriginState)) +

geom\_boxplot(outlier.shape=NA) + ylim(0, 50)

bp6=ggplot(dff, aes(x=factor(dff$DestState),

y=dff$ArrDelayMinutes,color=dff$DestState)) +

geom\_boxplot(outlier.shape=NA) + ylim(0, 200)

# Remove outliers

#dff$Distance[which(dff$Distance > 1400)]=1400

#dff$ArrDelayMinutes[which(dff$ArrDelayMinutes > 180)]=180

#dff$DepDelayMinutes[which(dff$DepDelayMinutes > 180)]=180

#ggarrange(bp1,bp2,bp3,bp4 + rremove("x.text"),

# labels = c("A", "B","C","D","E","F"),

# ncol = 1, nrow = 6)

ps1=ggscatter(dff, x = "DepDelayMinutes",

y = "ArrDelayMinutes",add = "reg.line") +

stat\_cor(method = "pearson",label.x = 3, label.y = 32) +

stat\_regline\_equation(label.x = 3, label.y = 32)

ps2=ggscatter(dff, x = "ArrivalDelayGroups",

y = "ArrDelayMinutes",add = "reg.line") +

stat\_cor(method = "pearson",label.x = 3, label.y = 32) +

stat\_regline\_equation(label.x = 3, label.y = 32)

ps3=ggscatter(dff, x = "DepartureDelayGroups",

y = "ArrDelayMinutes",add = "reg.line") +

stat\_cor(method = "pearson",label.x = 3, label.y = 32) +

stat\_regline\_equation(label.x = 3, label.y = 32)

ps4=ggscatter(dff, x = "DistanceGroup",

y = "ArrDelayMinutes",add = "reg.line") +

stat\_cor(method = "pearson",label.x = 3, label.y = 32) +

stat\_regline\_equation(label.x = 3, label.y = 32)

#ggarrange(ps1, ps2,ps3 + rremove("x.text"),

#labels = c("A", "B","C","D"),

#ncol = 1, nrow = 3)