# Marketing Analytics Team Project, Final Report

*Author: Ran Dou, Maduduzi Langwenya, Siyan Lin, Jiaxin Zhang*

*Date: 12/05/2018*

# BUS256a: Marketing Analytics

## Team Project: Final Report

## TABLE OF CONTENTS

```
# Install required packages
pkg <- c("tidyverse", "broom", "pander", "knitr", "RColorBrewer", "grid", "sjPlot", "sca
les", "VIM", "usmap", "ggthemes", "gridExtra", "plm", "data.table", "lmtest", "sandwich"
, "stargazer", "pROC", "epiDisplay", "mclust")
pkg.uninstalled <- pkg[!(pkg %in% installed.packages())]
if (length(pkg.uninstalled)) {install.packages(pkg, repos ="http://cran.rstudio.com")}
# Library packages
library(tidyverse); library(broom); library(pander); library(knitr); library(RColorBrewe
r); library(grid);
library(sjPlot); library(scales); library(VIM); library(usmap); library(ggthemes); libra
ry(gridExtra); library(plm); library(data.table); library(lmtest); library(sandwich); li
brary(stargazer); library(pROC); library(epiDisplay); library(mclust)
rm(list=ls())  #clears the environment window pane in RStudio
setwd("~/Esther/Brandeis/2018 Fall/4 BUS256A Marketing Analytics  Bhoomija Ranjan/Team P
roject/data")
# Load csv files
FLG <- read_csv("flights.csv")
airports <- read_csv("airports.csv")
airlines <- read_csv("airlines.csv")
L_AIRPORT <- read.csv("L_AIRPORT.csv")
L_AIRPORT_ID <- read_csv("L_AIRPORT_ID.csv" )
```

# 1 INTRODUCTION

Delays are a major challenge of flight scheduling and airport management. Although some state and federal governments have invested significant efforts to improve airport infrastructure, most travelers still cannot avoid disruptions to their itineraries. In 2015, more than 1 in 3 of flights (Fig. 1) were delayed. A multitude of unpredictable factors (not exhaustive) such as weather and mechanical breakdowns make it harder for airport managers to predict which flights will likely be delayed. However, it may be possible to identify some top causes of delays (e.g. upstream flight delays) from mining a large 2015 flights dataset for insights. We plan to

investigate the flights' metadata such as airline, region, airport size, season, day, week and time of departure to build models to predict which flights are likely to be delayed. The conclusions will be useful for helping an FAA regulator devise an early warning system to help airlines proactively avoid delays.

**Specific Project Goals**

- How does Airport delay vary by Season, Month, Day of Week and Time of Day?
- How does Airport delay vary by Geography, Flight Distance, Airline and Airport Size?
- Are Winter and Summer delays worse than the year average?
- What are the best predictors for flights delay? What is the best model?
- Predict whether a flight will be delayed using a new flights dataset
- Segment flights delays, using their flight characteristics

---

# 2 DATA SECTION

The dataset is a 2015 USDOT flight data that contains around 5.8 million flights and 31 columns. It has 14 airlines and 322 airports from the United States. The columns consist of flight metadata such as airports, airlines, and delay times, flight timing, delay information, and causes of delay. The following tables list all the variables in the dataset and their corresponding data types.

**Check data types**

```
##   AIR_SYSTEM_DELAY AIR_TIME   AIRLINE AIRLINE_DELAY
## 1          integer  integer character       integer
```

```
##   ARRIVAL_DELAY ARRIVAL_TIME CANCELLATION_REASON CANCELLED
## 1       integer    character           character   integer
```

```
##       DAY DAY_OF_WEEK DEPARTURE_DELAY DEPARTURE_TIME
## 1 integer     integer         integer      character
```

```
##   DESTINATION_AIRPORT DISTANCE DIVERTED ELAPSED_TIME
## 1           character  integer  integer      integer
```

```
##   FLIGHT_NUMBER LATE_AIRCRAFT_DELAY   MONTH ORIGIN_AIRPORT
## 1       integer             integer integer      character
```

```
##   SCHEDULED_ARRIVAL SCHEDULED_DEPARTURE SCHEDULED_TIME SECURITY_DELAY
## 1         character           character        integer        integer
```

```
##   TAIL_NUMBER TAXI_IN TAXI_OUT WEATHER_DELAY
## 1   character integer  integer       integer
```

```
##   WHEELS_OFF WHEELS_ON    YEAR
## 1  character character integer
```

# 2.1 DESCRIPTION OF DATA

**Year, Month, Day, Day of Week:** all refer to the date and time of the flight.
**AIRLINE:** A code assigned by U.S. Department of Transportation to identify airlines
**FLIGHT_NUMBER and TAIL_NUMBER:** indicate the identification of a flight
**ORIGIN_AIRPORT and DESTINATION_AIRPORT:** 3-letter code assigned by IATA to uniquely identify the airports
**SCHEDULED_DEPARTURE and SCHEDULED_ARRIVAL:** Scheduled times of take-off and landing
**DEPARTURE_TIME and ARRIVAL_TIME:** Actual take-off and landing times
**DEPARTURE_DELAY and ARRIVAL_DELAY:** Difference (in minutes) between scheduled and actual departure or arrival times
**TAXI_IN, TAXI_OUT, WHEELS_ON, WHEELS_OFF:** taxi in time (minutes), taxi out time (minutes), and take-off time
**ELAPSED_TIME:** is the total time of taxi in, taxi out, and effective air time
**AIRTIME:** indicates the time between wheels off and wheels on
**DISTANCE:** Distance (in miles) between two airports
**DIVERTED:** indicates whether the flight was diverted
**CANCELLED:** indicates whether the flight was cancelled

# 2.2 DATA CLEANING

## 2.2 (i) DATA TYPES

```
FLG <- FLG %>% mutate_if(is.character, as.factor)
FLG$DAY <- as.factor(FLG$DAY)
FLG$DIVERTED <- as.factor(FLG$DIVERTED)
FLG$CANCELLED <- as.factor(FLG$CANCELLED)
airports$IATA_CODE <- as.factor(airports$IATA_CODE)
```

Categorical variables such as destination and origin airport, tail-number, airline, cancellation reason, day, cancelled, and flight number originally came with the wrong data type, so we converted them to factors throughout our analysis. Similarly, all time variables such as arrival time, scheduled arrival, wheels off, wheels on, departure time and month were converted to the date format throughout the analysis.

The categorical variables Month and Day of Week were originally stored as numerical values in the dataset. To ensure that they were handled properly throughout the analyses in the project, we changed the integer values to ordinal levels such as Monday to Friday and January to December. This made the visualizations easier to read.

## 2.2 (ii) NEWLY CREATED VARIABLES

We also created new variables such Airport Size, Region, Season, Time of Day, Distance Group, Delay Difference, Delay Category to aggregate the observations to a less granular level of analysis. We determined that 322 airports are too many to conduct meaningful analyses, so we segmented them (Apriori) into 6 categories based on the number of flights. Our intervals for these groups were based on how the FAA classifies

airport size but since we do not have passenger counts forairport (as done by the FAA), we used intervals based on the number of flights per airport in 2015 to classify the airports. We felt that the intervals were reasonable given that well know airports such as Chicago O'Hare International Airport were in categories we would expect.

We hypothesized that the region for airports affects the severity of delays. We expect snowy winters in the North-East, and increased spring and summer tourism in the South region to worsen delays. So, to investigate such regional effects we divided the 322 airports into 5 regions (West, Midwest, South, North East, and Overseas Territories) based on the state the airport is located.

We also looked at how flight distance affects the delays. We hypothesized that the increased frequency of short distance flights might lead to the congested airport runway and therefore worse departure and arrival delays. On the contrary, we expect flights with longer distances to have lower flight frequencies, less congested terminals, and therefore lower departure delays. The longer distances might also mean pilots can fly faster over long distances in order to reduce arrival delays.

We also created a season variable based on the month of the flight in order to explore seasonal effects on delay. We were interested in statistically testing whether adverse weather conditions in winter and increased travel during summer lead to worse than average delays. Another variable we created was Time of Day, which divided arrival and departure times into 6 categories (early morning, morning, midday, afternoon, evening and night) based on how Google Flights categorizes departure and arrival times. Exploring how delays vary as the day progresses could help airline and airport managers proactively shift operational resources to the time when they are most needed.

To measure the severity of the delays, we created Delay Category which divided delays into 30-minute intervals.

**(1) Clean Day of Week**

We converted the variable `DAY_OF_WEEK` from number to character, so that we can use the name in our graph to identify them more clearly.

```
FLG <- FLG %>% mutate(DAY_OF_WEEK = ifelse(DAY_OF_WEEK == 1, "Monday", ifelse(DAY_OF_WEE
K == 2, "Tuesday",
                                    ifelse(DAY_OF_WEEK == 3, "Wednesday", ifelse(DAY_OF_
WEEK == 4, "Thursday",
                                    ifelse(DAY_OF_WEEK == 5, "Friday", ifelse(DAY_OF_WEE
K == 6, "Saturday",
                                    ifelse(DAY_OF_WEEK == 7,"Sunday", "Missing" ))))))))
FLG$DAY_OF_WEEK <- factor(FLG$DAY_OF_WEEK, levels = c("Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday", "Sunday", "Missing"))
```

**(2) Clean up Distance**

We also created a variable called `DISTANCEGROUP` to divide the distance of each flight into five categories (<250, 250-749, 1250-1749, 1750-2249 and 2249+).

```
FLG <- FLG %>% mutate(DISTANCEGROUP = ifelse(DISTANCE < 250, "<250 Miles",
                                      ifelse(DISTANCE >= 250 & DISTANCE < 750 , "250-749
 Miles",
                                      ifelse(DISTANCE >= 750 & DISTANCE < 1250, "750-124
9 Miles",
                                      ifelse(DISTANCE >= 1250 & DISTANCE < 1750, "1250-1
749 Miles",
                                      ifelse(DISTANCE >= 1750 & DISTANCE < 2250, "1750-2
249 Miles",
                                      ifelse(DISTANCE >= 2250, "2249+ Miles", "Missing"
)))))))
```

### (3) Create Season

This `SEASON` variable was created based on months of the year. There are four levels for this factor variable: Spring, Summer, Fall, and Winter.

```
FLG <- FLG %>% mutate(SEASON = ifelse(MONTH == 12 | MONTH <= 2 , "Winter",
                               ifelse(MONTH >= 9 & MONTH <= 11 , "Fall",
                               ifelse(MONTH >= 6 & MONTH <= 8 ,"Summer",
                               ifelse(MONTH >= 3 & MONTH<=5, "Spring", "Missing")))))
FLG$SEASON <- factor(FLG$SEASON, levels = c("Spring", "Summer", "Fall", "Winter", "Missi
ng"))
```

### (4) Clean Month

Same as what we did to `DAY_OF_WEEK` , we converted the variable `MONTH` from number to character, so that we could use the name in our graph to identify days more clearly.

```
FLG <- FLG %>% mutate(MONTH = ifelse(MONTH == 1, "January", ifelse(MONTH == 2, "Februar
y",
                              ifelse(MONTH == 3, "March", ifelse(MONTH == 4, "April",
                              ifelse(MONTH == 5, "May", ifelse(MONTH == 6, "June",
                              ifelse(MONTH == 7, "July", ifelse(MONTH == 8, "August",
                              ifelse(MONTH == 9, "September", ifelse(MONTH == 10, "Octob
er",
                              ifelse(MONTH == 11, "November", ifelse(MONTH == 12,"Decemb
er",
                                                                     "Missing"
 ))))))))))))
FLG$MONTH <- factor(FLG$MONTH, levels = c("January", "February", "March", "April", "May"
, "June", "July", "August", "September", "October", "November", "December", "Missing"))
```

### (5) Break out Departure and Arrival Time

We broke out both the `SCHEDULED_DEPARTURE` and `SCHEDULED_ARRIVAL` into six categories: Morning, Midday, Afternoon, Evening, Night, and Early morning.

```
FLG$SCHEDULED_DEPARTURE <- as.character(FLG$SCHEDULED_DEPARTURE) %>% as.numeric(FLG$SCHE
DULED_DEPARTURE)
FLG <- FLG %>% mutate(DepartureTime_of_Day=ifelse(SCHEDULED_DEPARTURE>=0800&SCHEDULED_DE
PARTURE<1100,"Morning",
                                        ifelse(SCHEDULED_DEPARTURE>=1100&SCHEDULED_DEPART
URE<1400,"Midday",
                                        ifelse(SCHEDULED_DEPARTURE>=1400&SCHEDULED_DEPART
URE<1700,"Afternoon",
                                        ifelse(SCHEDULED_DEPARTURE>=1700&SCHEDULED_DEPART
URE<2100,"Evening",
                                        ifelse(SCHEDULED_DEPARTURE>=2100&SCHEDULED_DEPART
URE<=2400,"Night",
                                        ifelse(SCHEDULED_DEPARTURE>=0000&SCHEDULED_DEPART
URE<0800,"Early morning",

         "Missing")))))))
FLG$DepartureTime_of_Day<-factor(FLG$DepartureTime_of_Day,
                           levels=c("Early morning","Morning","Midday","Afternoon","Eveni
ng","Night", "Missing"))

FLG$SCHEDULED_ARRIVAL <- as.character(FLG$SCHEDULED_ARRIVAL) %>% as.numeric(FLG$SCHEDULE
D_ARRIVAL)
FLG <- FLG %>% mutate(ArrivalTime_of_Day = ifelse(SCHEDULED_ARRIVAL>=800&SCHEDULED_ARRIV
AL<1100,"Morning",
                                        ifelse(SCHEDULED_ARRIVAL>=1100&SCHEDULED_ARRIVAL<
1400,"Midday",
                                        ifelse(SCHEDULED_ARRIVAL>=1400&SCHEDULED_ARRIVAL<
1700,"Afternoon",
                                        ifelse(SCHEDULED_ARRIVAL>=1700&SCHEDULED_ARRIVAL<
2100,"Evening",
                                        ifelse(SCHEDULED_ARRIVAL>=2100&SCHEDULED_ARRIVAL<
=2400,"Night",
                                        ifelse(SCHEDULED_ARRIVAL>=0000&SCHEDULED_ARRIVAL<
800,"Early morning",

    "Missing")))))))
FLG$ArrivalTime_of_Day<-factor(FLG$ArrivalTime_of_Day,
                         levels=c("Early morning","Morning","Midday","Afternoon","Evenin
g","Night", "Missing"))
```

### (6) Break the delay time into several categories

We also created two new variables `Depdelay_category` and `Arrdelay_category` to categorize the delay time of each flight into five categories.

```
#departure delay
FLG$DEPARTURE_DELAY <- as.numeric(FLG$DEPARTURE_DELAY)
FLG <- FLG %>% mutate(Depdelay_category = ifelse(DEPARTURE_DELAY<=0, "ahead of schedule"
,
                                      ifelse(DEPARTURE_DELAY>0 & DEPARTURE_DELAY<30, "less
 than 30 minutes late",
                                      ifelse(DEPARTURE_DELAY>=30 & DEPARTURE_DELAY<60, "30
 to 60 minutes late",
                                      ifelse(DEPARTURE_DELAY>=60 & DEPARTURE_DELAY<90, "60
 to 90 minutes late",
                                      ifelse(DEPARTURE_DELAY>=90, "more than 90 minutes la
te", "Missing"))))))
#arrival delay
FLG$ARRIVAL_DELAY <- as.numeric(FLG$ARRIVAL_DELAY)
FLG <- FLG %>% mutate(Arrdelay_category = ifelse(ARRIVAL_DELAY<=0, "ahead of schedule",
                                      ifelse(ARRIVAL_DELAY>0 & ARRIVAL_DELAY<30, "less tha
n 30 minutes late",
                                      ifelse(ARRIVAL_DELAY>=30 & ARRIVAL_DELAY<60, "30 to
 60 minutes late",
                                      ifelse(ARRIVAL_DELAY>=60 & ARRIVAL_DELAY<90, "60 to
 90 minutes late",
                                      ifelse(ARRIVAL_DELAY>=90, "more than 90 minutes lat
e", "Missing"))))))
```

# 2.2 (iii) AIRLINES AND AIRPORTS

The Origin and Destination Airport contains 5-digit airport ID values in the month of October that range between 10125 and 16218 and do not have a match in the `airports` data set. The rest of the `flights` dataset uses 3-letter IATA codes to uniquely identify airports. To be able to conduct descriptive analyses such as the number of flights and create new variables such as `region` for the origin and destination airports, we converted the 5-digit airport ID into the widely used 3-letter IATA codes for a total of 322 distinct airports.

We downloaded two files that contain the 5-digit airport ID values and 3-letter IATA codes from the USDOT website, used them to convert the 5-digit airport IDs into their corresponding 3-letter IATA codes. There were two duplicate pairs of 3-letter IATA that referred to the same airport. For the first pair, NYL and YUM which referred to Yuma International airport, we deleted NYL after we found out from the USDOT website that the airport uses YUM. Similarly, for the BSM and AUS pair we deleted BSM after we realized that it was no longer used for Austin Bergstrom International Airport.

Once we had fixed the 5-digit airport ID issue, we converted the IATA codes to the colloquial names of the airports. Similarly, for airlines, we converted their 3-letter FAA airline designators into the colloquial names of the airport. Both changes were done to make visualizations easier to read.

## (1) FIX AIRPORT CODE

```
#filter to october data and clean it up a bit
october <- FLG %>% filter(MONTH == "October"); not_october <- FLG %>% filter(MONTH != "O
ctober")
#drop the levels for the factor variables
october$ORIGIN_AIRPORT <- droplevels(october$ORIGIN_AIRPORT)
october$DESTINATION_AIRPORT <- droplevels(october$DESTINATION_AIRPORT)
not_october$ORIGIN_AIRPORT <- droplevels(not_october$ORIGIN_AIRPORT)
not_october$DESTINATION_AIRPORT <- droplevels(not_october$DESTINATION_AIRPORT)
#get unique october airport codes
unique_origin_codes <- as.data.frame(unique(october$ORIGIN_AIRPORT)) %>%
                                     rename(ORIGIN_AIRPORT=`unique(october$ORIGIN_AIRPOR
T)`)
unique_dest_codes <- as.data.frame(unique(october$DESTINATION_AIRPORT)) %>%
                                    rename(DESTINATION_AIRPORT = `unique(october$DESTINAT
ION_AIRPORT)`)
unique_origin_codes$ORIGIN_AIRPORT <- droplevels(unique_origin_codes$ORIGIN_AIRPORT)
unique_dest_codes$DESTINATION_AIRPORT <- droplevels(unique_dest_codes$DESTINATION_AIRPOR
T)
#map 5 letter codes to 3 letter codes
#(1)origin
airport_origin_fix <- merge(L_AIRPORT, L_AIRPORT_ID, by.x = "Description", by.y = "Descr
iption") %>%
                      rename(ORIGIN_AIRPORT = Code.y ) %>% distinct(ORIGIN_AIRPORT, Cod
e.x) %>%
                      filter(ORIGIN_AIRPORT %in% unique_origin_codes$ORIGIN_AIRPORT) %>%
                      mutate_if(is.integer, as.factor);
airport_origin_fix$Code.x <- droplevels(airport_origin_fix$Code.x)
#(2)destination
airport_dest_fix <- merge(L_AIRPORT, L_AIRPORT_ID, by.x = "Description", by.y = "Descrip
tion") %>%
                    rename(DESTINATION_AIRPORT = Code.y ) %>% distinct(DESTINATION_AIRPO
RT, Code.x) %>%
                    filter(DESTINATION_AIRPORT %in% unique_dest_codes$DESTINATION_AIRPOR
T) %>%
                    mutate_if(is.integer, as.factor);
airport_dest_fix$Code.x <- droplevels(airport_dest_fix$Code.x)
#fix codes in october slide of the flights data
#(1)origin
october <- october %>% inner_join(airport_origin_fix) %>%
                       dplyr::select(-ORIGIN_AIRPORT) %>% rename(ORIGIN_AIRPORT = Code.
x)
#(2)destination
october <- october %>% inner_join(airport_dest_fix) %>%
                       dplyr::select(-DESTINATION_AIRPORT) %>% rename(DESTINATION_AIRPOR
T = Code.x)
#re-unite october dataframe with non-october dataframe
FLG <- rbind(not_october, october)
```

# (2) MERGE AIRLINES AND AIRPORTS

### Merge `flights` with `airlines`

```
FLG <- merge(FLG, airlines, by.x="AIRLINE", by.y="IATA_CODE") %>% dplyr::select(-AIRLIN
E) %>% rename(AIRLINE=AIRLINE.y)
FLG$AIRLINE <- as.factor(FLG$AIRLINE)
```

**Merge `flights` with `airport`**

```
#origin airport
FLG <- left_join(FLG, airports[,c("IATA_CODE","AIRPORT","STATE", "LATITUDE","LONGITUDE",
"REGION")],
                 by = c("ORIGIN_AIRPORT" = "IATA_CODE")) %>% dplyr::select(-ORIGIN_AIRPO
RT) %>%
                 rename(ORIGIN_AIRPORT=AIRPORT, ORIGIN_STATE=STATE, ORIGIN_LATITUDE=LATI
TUDE,
                        ORIGIN_LONGITUDE=LONGITUDE, ORIGIN_REGION=REGION)
FLG$ORIGIN_AIRPORT <- as.factor(FLG$ORIGIN_AIRPORT)
#destination airport
FLG <- left_join(FLG, airports[, c("IATA_CODE","AIRPORT","STATE", "LATITUDE","LONGITUDE"
,"REGION")],
                 by =c("DESTINATION_AIRPORT" = "IATA_CODE")) %>% dplyr::select(-DESTINAT
ION_AIRPORT) %>%
                 rename(DESTINATION_AIRPORT=AIRPORT, DESTINATION_STATE=STATE, DESTINATIO
N_LATITUDE=LATITUDE,
                        DESTINATION_LONGITUDE=LONGITUDE, DESTINATION_REGION=REGION)
FLG$DESTINATION_AIRPORT <- as.factor(FLG$DESTINATION_AIRPORT)
```

# 2.2 (iv) MISSING DATA

Before conducting exploratory analyses on departure and arrival delay, we investigated missing data within our target variables (departure delay and arrival delay). The departure and arrival delay variables contained 1.5% and 1.8% missing data, respectively.

To address this missing daata, we divided the data into departure (FLG_dep) and arrivals (FLG_arr) data. All missing data in the departure delays data frame was attributed to canceled flights. So, we removed all canceled flights from the departure delays dataset. Unfortunately, deleting all canceled flights meant losing about 3700 departure delays observations, which is a miniscule number given the 5.73 million flights that remained in the departure delays after this process. For the arrival delays dataset, we found that canceled and diverted flights had no arrival delay which was expected given that there can be no arrival delay if a flight never landed in its intended airport or was canceled. So, we excluded canceled and diverted flights in arrival delays data leaving a total of 5.71 million flights. The eliminated missing data in both departure and arrival data sets are not expected to bias our analysis in any way.

Once we eliminated NAs in both the departure and arrival variables, we combined ( `rbind` ) the departure and arrivals data sets into one data set with 11.4 million rows. To accomplish this, we renamed several variables so that both data sets had common column names.

**(1) Create departure dataframe**

```
#filter to only departure
FLG_dep <- FLG %>%
  filter(CANCELLED == 0) %>%
  dplyr::select(-contains("ARRIVAL"),-CANCELLED,-CANCELLATION_REASON,-Arrdelay_category,
-contains("DESTINATION"))%>%
  rename(SCHEDULED_TAKEOFF_LANDING = SCHEDULED_DEPARTURE,
         ACTUAL_TAKEOFF_LANDING = DEPARTURE_TIME,
         DELAY = DEPARTURE_DELAY,
         STATE = ORIGIN_STATE,
         REGION = ORIGIN_REGION,
         AIRPORT = ORIGIN_AIRPORT,
         LATITUDE = ORIGIN_LATITUDE,
         LONGITUDE = ORIGIN_LONGITUDE,
         TIME_OF_DAY = DepartureTime_of_Day,
         DELAY_TYPE = Depdelay_category) %>%
  mutate( ID = "Departure")
#check NAs
anyNA(FLG_dep$DELAY) # No NAs in DEPARTURE_DELAY column
```

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

### (2) Create arrival dataframe

```
#filter to only arrivals
FLG_arr <- FLG %>%
  filter(CANCELLED == 0 & DIVERTED == 0) %>%
  dplyr::select(-contains("DEPARTURE"),-CANCELLED,-CANCELLATION_REASON,-Depdelay_categor
y,-contains("ORIGIN"))%>%
  rename(SCHEDULED_TAKEOFF_LANDING = SCHEDULED_ARRIVAL,
         ACTUAL_TAKEOFF_LANDING = ARRIVAL_TIME,
         DELAY = ARRIVAL_DELAY,
         STATE = DESTINATION_STATE,
         REGION = DESTINATION_REGION,
         AIRPORT = DESTINATION_AIRPORT,
         LATITUDE = DESTINATION_LATITUDE,
         LONGITUDE = DESTINATION_LONGITUDE,
         TIME_OF_DAY = ArrivalTime_of_Day,
         DELAY_TYPE = Arrdelay_category) %>%
  mutate( ID = "Arrival")
anyNA(FLG_arr$DELAY) # No NAs in ARRIVAL_DELAY column
```

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

### (3) Merge arrivals and departure dataframes into one

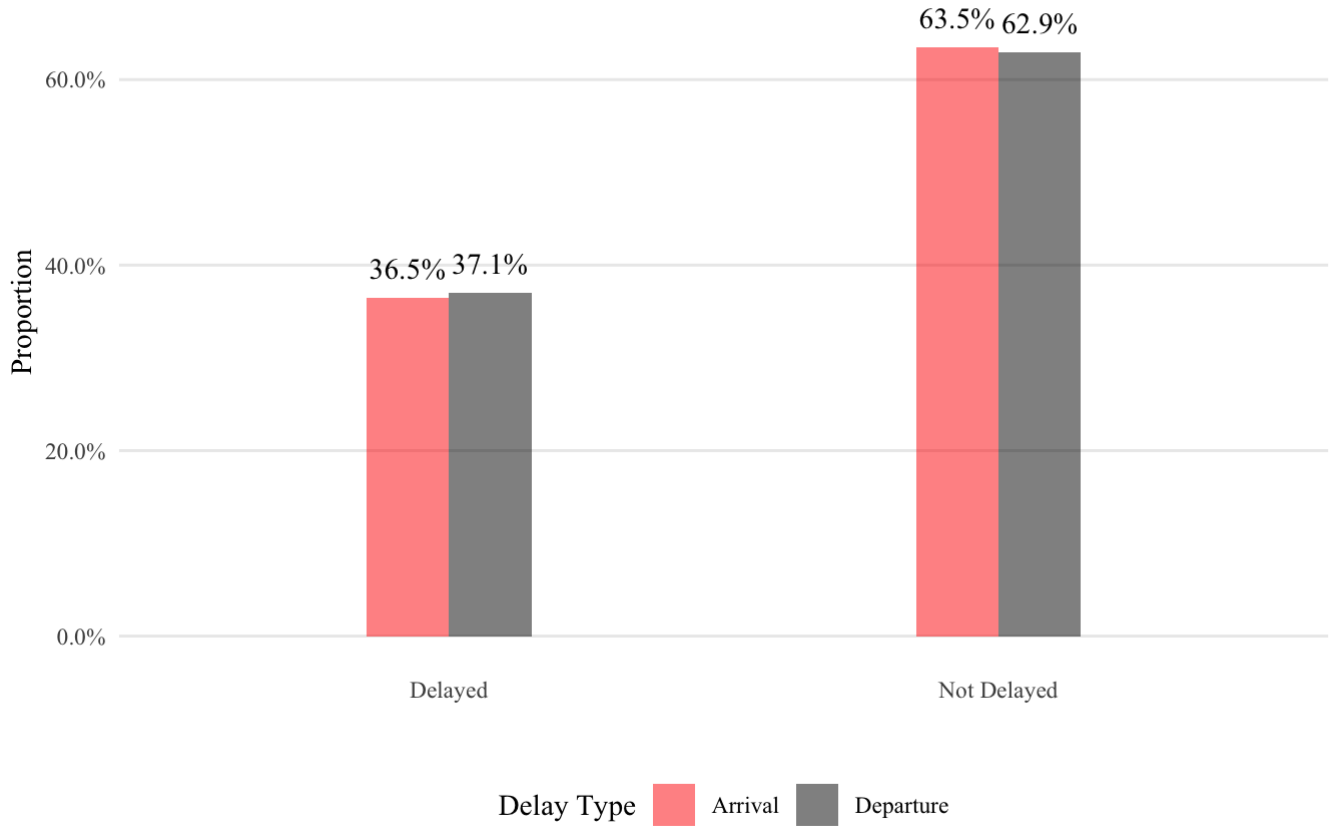
```
FLG <- rbind(FLG_dep, FLG_arr)
#create a dummy variable for delay
FLG <- FLG %>% mutate(DELAY_DUMMY = ifelse(DELAY > 0, "Delayed", "Not Delayed"))
#create airport_size to reflect the numbers of flights by airport
AIRPORT_ANALYSES <- FLG %>% group_by(AIRPORT, ID) %>% summarise(count = n()) %>%
                            mutate(AIRPORT_SIZE = ifelse(count<1000, "Basic",
                                                  ifelse(count>=1000 & count<10000, 'Loc
al',
                                                  ifelse(count>=10000 & count<40000, 'Re
gional',
                                                  ifelse(count>=40000 & count<60000, 'Sm
all National',
                                                  ifelse(count>=60000 & count<100000, 'M
edium National',
                                                  ifelse(count>=100000, 'Large National'
, 'Missing')))))) %>%
                            group_by(AIRPORT, AIRPORT_SIZE, ID) %>%
                            summarise(total = sum(count)) %>% dplyr::select(-total);
FLG <- FLG %>% left_join(AIRPORT_ANALYSES, by = c("AIRPORT", "ID"))
FLG$AIRPORT_SIZE <- factor(FLG$AIRPORT_SIZE,
                    levels = c('Basic','Local','Regional','Small National','Medium Natio
nal','Large National'))
```

# 3 VISUALIZATIONS

## 3.1 General View

```
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         legend.position = "bottom", panel.grid.minor = element_blank(), panel.grid.majo
r.x = element_blank())
p1 <- FLG %>% group_by(ID, DELAY_DUMMY) %>% tally() %>% mutate(Proportion = n/sum(n)) %
>%
      ggplot(aes(x = factor(DELAY_DUMMY), y = Proportion, fill = ID, group = ID)) +
      geom_bar(stat = "identity", position = "dodge", width = 0.3, alpha = 0.5) +
      geom_text(aes(label = scales::percent(Proportion)), position = position_dodge(widt
h = 0.3), vjust = -1,
      family="Times New Roman") + scale_y_continuous(labels = percent, limits = c(0, 0.7
)) +
      labs(title = "Fig. 1  Percentage of Delay for Each Type", x = "", fill = "Delay Ty
pe") + theme +
      scale_fill_manual(values = c("red", "black")); p1
```

Fig. 1 Percentage of Delay for Each Type




Generally, 36.5% of flights have arrival delay and 37.1% of flights have departure delay. As the proportion of the two types delay is similar, it's reasonable for us to make the conclusion that most departure delays led to an arrival delay.

# 3.2 Distribution of Delay by Season, Month, Day of Week, Time of Day

### (1) Overview of 2015

```
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
        panel.grid.minor = element_blank(), panel.grid.major.x = element_blank())
p2 <- FLG %>% mutate(DATE = paste(YEAR, MONTH, DAY, sep = "/"), DATE = as.Date(DATE, "%
Y/%B/%d")) %>%
      dplyr::select(DATE, DELAY, ID) %>% group_by(DATE, ID) %>% summarize(`Average Delay
` = mean(DELAY)) %>%
      ggplot(aes(x = DATE, y = `Average Delay`, fill = ID)) + geom_area(alpha = 0.5) + t
heme +
      labs(title = "Average Delay in Each Day (2015)", x = "", fill = "Delay Type") +
      scale_fill_manual(values = c("red", "black"))
```

### (2) Day

```
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         legend.position = " ", panel.grid.minor = element_blank(), panel.grid.major.x =
 element_blank(),
       axis.title.y = element_blank())
p3 <- FLG %>% group_by(ID, DAY, DELAY_DUMMY) %>% tally() %>% mutate(Proportion = n/sum
(n)) %>%
      filter(DELAY_DUMMY == "Delayed") %>% ggplot(aes(x = DAY, y = Proportion, group = I
D, color = ID)) +
      geom_line(alpha = 0.5) + theme + scale_color_manual(values=c("red", "black")) +
      labs(title = "Average Delay throughout Each Day of the Month", x = "", color = "De
lay Type") +
      scale_y_continuous(labels = percent) + scale_x_discrete(breaks = c(1,5,10,15,20,25
,31))
```

### (3) Day of Week

```
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         legend.position = " ", panel.grid.minor = element_blank(), panel.grid.major.x =
 element_blank(),
         axis.title.y = element_blank(), axis.text.x = element_text(angle = 45, hjust =
1))
p4 <- FLG %>% group_by(ID, DAY_OF_WEEK, DELAY_DUMMY) %>% tally() %>% mutate(Proportion =
 n/sum(n)) %>%
      filter(DELAY_DUMMY == "Delayed") %>%
      ggplot(aes(x = DAY_OF_WEEK, y = Proportion, group = ID, color = ID)) + geom_line(a
lpha = 0.5) + theme +
      labs(title = "Average Delay throughout the Week", x = "", color = "Delay Type") +
      scale_color_manual(values = c("red", "black")) + scale_y_continuous(labels = perce
nt)
```

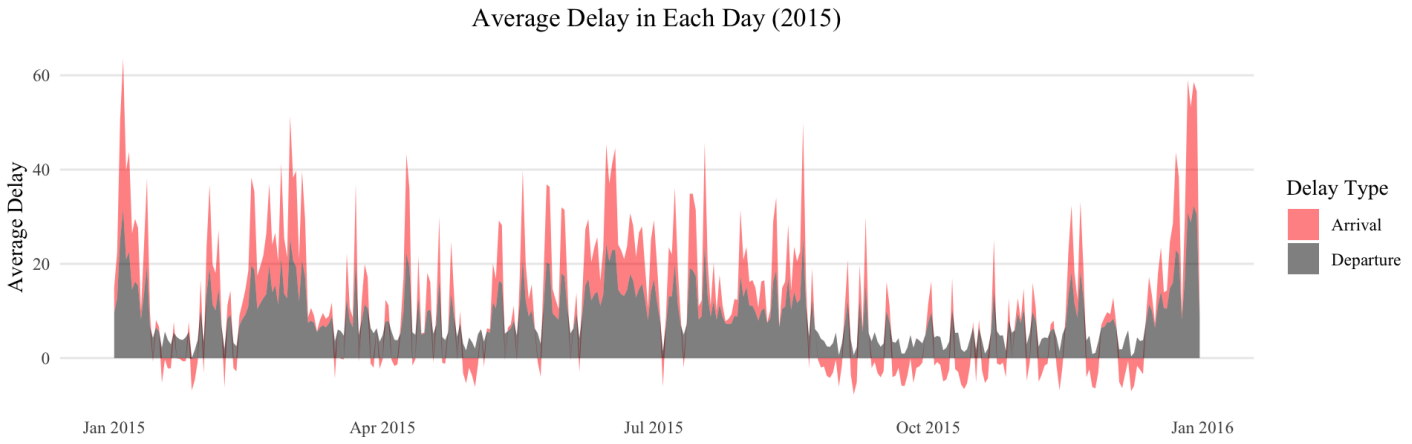
### (4) Month

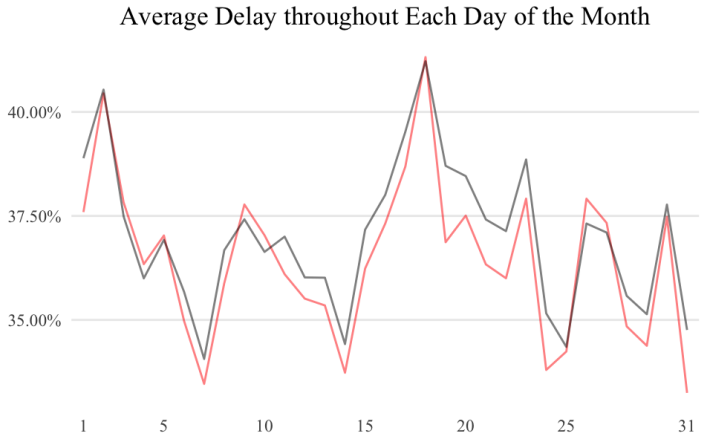
```
p5 <- FLG %>% group_by(ID, MONTH, DELAY_DUMMY) %>% tally() %>% mutate(Proportion = n/sum
(n)) %>%
      filter(DELAY_DUMMY == "Delayed") %>%
      ggplot(aes(x = MONTH, y = Proportion, group = ID, color=ID, fill=ID)) + geom_line
(alpha=0.5) + theme +
      labs(title = "Average Delay for Each Month", x = "", color = "Delay Type") +
      scale_color_manual(values = c("red", "black")) + scale_y_continuous(labels = perce
nt)
```

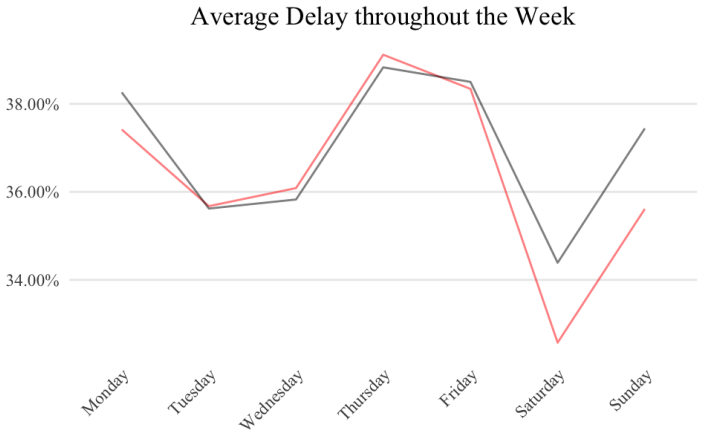
### (5) Season

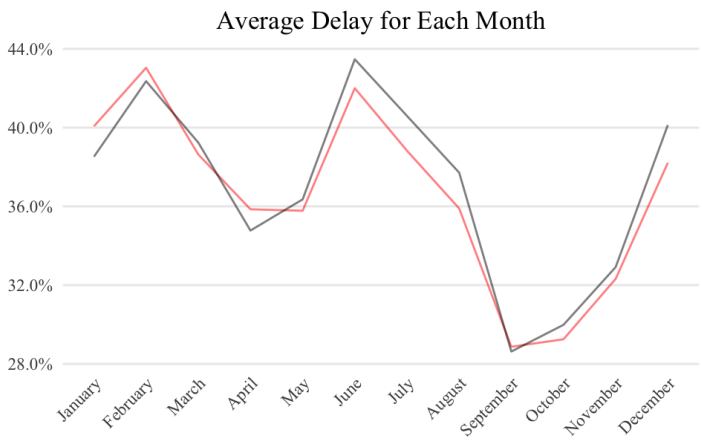
```
p6 <- FLG %>% group_by(ID, SEASON, DELAY_DUMMY) %>% tally() %>% mutate(Proportion = n/su
m(n)) %>%
      filter(DELAY_DUMMY == "Delayed") %>%
      ggplot(aes(x = SEASON, y = Proportion, group = ID, color=ID, fill=ID)) + geom_line
(alpha=0.5) + theme +
      labs(title = "Average Delay for Each Season", x = "", color = "Delay Type") +
      scale_color_manual(values = c("red", "black")) + scale_y_continuous(labels = perce
nt)
```

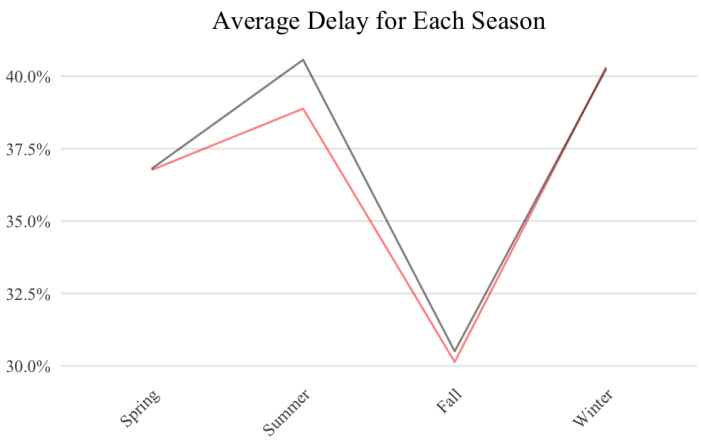
```
vplayout <- function(x,y){viewport(layout.pos.row = x, layout.pos.col = y)}
grid.newpage()  #create a new grid for the plots
pushViewport(viewport(layout = grid.layout(3,2))) #change the composition of the grid
print(p2, vp = vplayout(1,1:2))
print(p3, vp = vplayout(2,1))
print(p4, vp = vplayout(2,2))
print(p5, vp = vplayout(3,1))
print(p6, vp = vplayout(3,2))
```



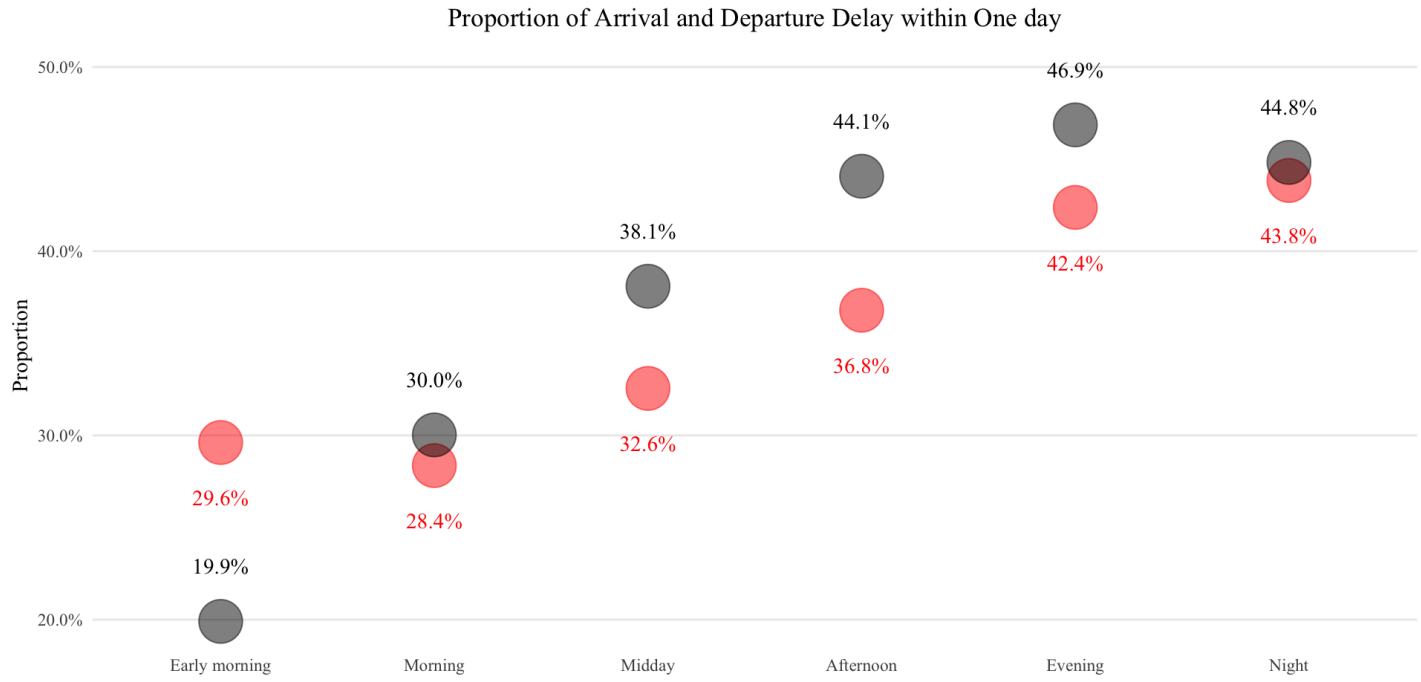

## (6) Proportion of delay in different time of day

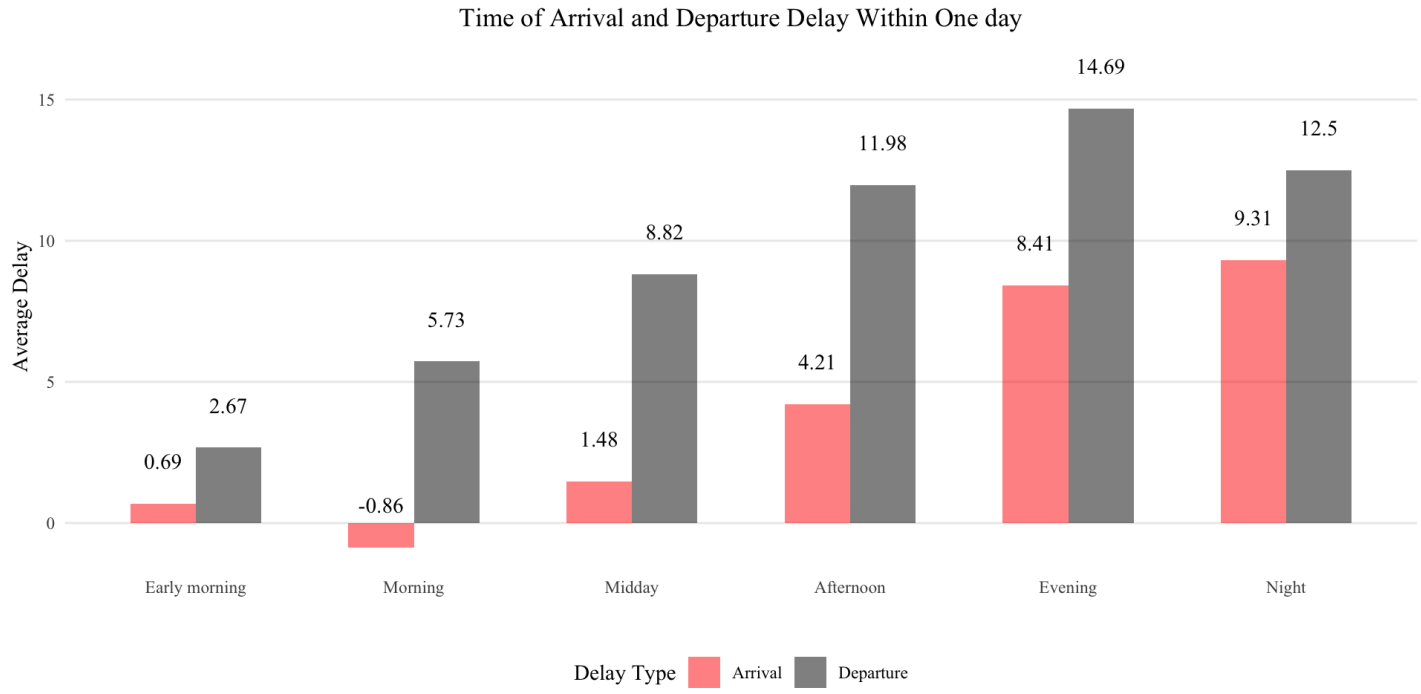
```
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         legend.position = " ", panel.grid.minor = element_blank(), panel.grid.major.x =
 element_blank())
p7 <- FLG %>% group_by(ID, TIME_OF_DAY, DELAY_DUMMY) %>% tally() %>% mutate(Proportion =
 n/sum(n)) %>%
      filter(DELAY_DUMMY == "Delayed") %>% ggplot(aes(x = TIME_OF_DAY, y = Proportion, g
roup=ID, color=ID)) +
      geom_point(size = 10, alpha = 0.5) + geom_text(aes(label=scales::percent(Proportio
n,digits=3),
      y=ifelse(ID=="Arrival",Proportion-0.03,Proportion+0.03)),family ="Times New Roman"
) + theme +
      labs(title = "Proportion of Arrival and Departure Delay within One day", x = "", c
olor = "Delay Type") +
      scale_color_manual(values = c("red", "black")) + scale_y_continuous(labels = perce
nt)
```

### (7) Average delay in different time of day

```
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         legend.position = "bottom",panel.grid.minor = element_blank(), panel.grid.majo
r.x = element_blank())
p8 <- FLG %>% group_by(ID, TIME_OF_DAY) %>% summarize(`Average Delay` = mean(DELAY)) %>%
      ggplot(aes(x = TIME_OF_DAY, y = `Average Delay`, group = ID, fill = ID)) +
      geom_bar(stat = "identity", position = "dodge", width=.6, alpha = 0.5) + theme +
      geom_text(aes(label = round(`Average Delay`, 2), y = `Average Delay`+ 1.5),
      family = "Times New Roman", position = position_dodge(width = 0.6)) +
      labs(title = "Time of Arrival and Departure Delay Within One day", x = "", fill =
"Delay Type") +
      scale_fill_manual(values = c("red", "black")) + scale_color_manual(values = c("re
d", "black"))
```

```
grid.newpage()  #create a new grid for the plots
pushViewport(viewport(layout = grid.layout(2,1))) #change the composition of the grid
print(p7, vp = vplayout(1,1))
print(p8, vp = vplayout(2,1))
```




To get a sense of how delays vary across time, we calculated the proportion of delayed flights and plotted it against time of day, day of week, month and season. We found that January, March, July, August, and December have the highest delays in 2015. This is consistent with our hypothesis that adverse conditions during winter maybe causing flight delays. Similarly, more air traffic due to vacations in the summer maybe causing congestions at airports. The peak around March may be related to Spring break holidays when many students either travel home or to vacations destinations. Also, worth highlighting are the early months of the Fall season, which had significantly shorter delays.

On average, dates around the beginning and middle of the month have the highest proportion of delayed flights. We believe this could be due to increased air congestion around these dates. From a week perspective, Thursday and Friday tend to have the highest proportion of delayed flights probably due to increased air traffic due to travel ahead of the weekend. Similarly, Mondays have the highest percentage of delayed flights

probably due to people returning from weekend getaways. Within each day, the afternoon, evening and night have on average the highest proportion of delayed flights. Departure delays tend to be longer than Arrival delays maybe because pilots fly faster so that travelers can catch the next connection, although the data clearly shows that it's typically not enough to eliminate arrival delays entirely.

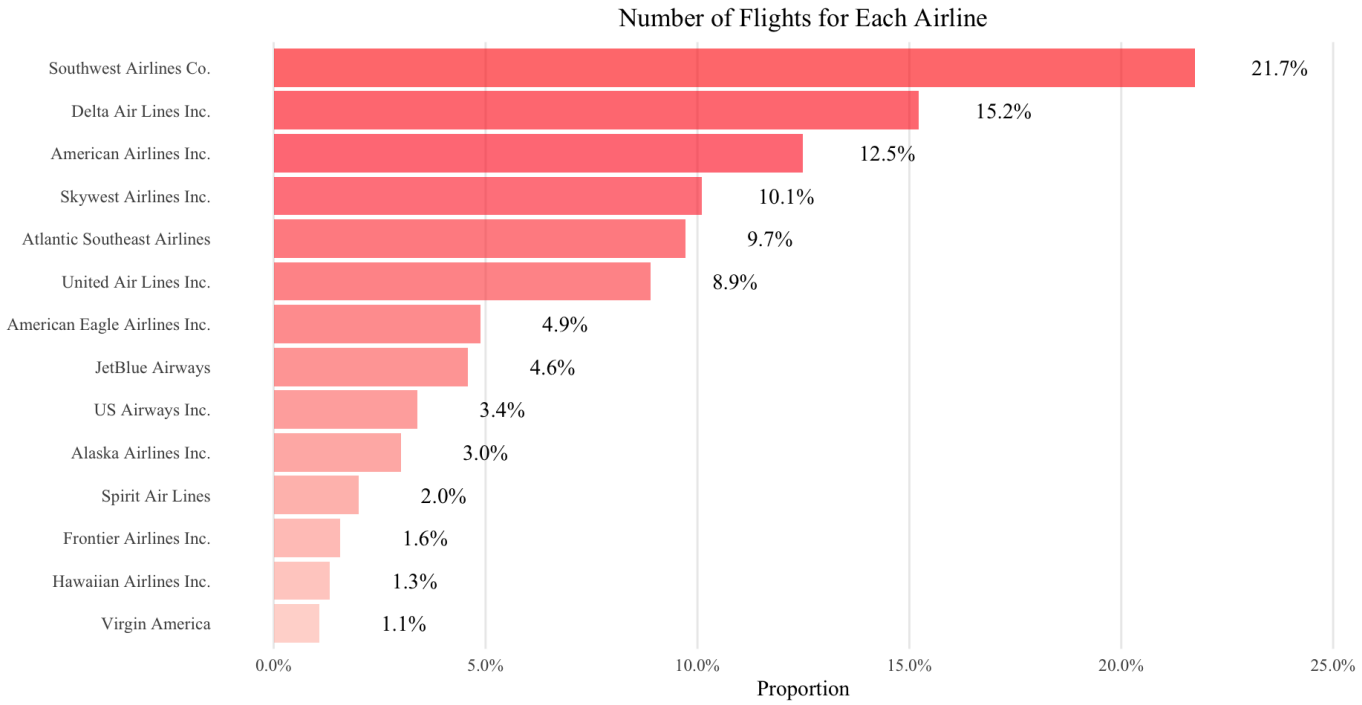
# 3.3 Distribution of Delay by Airline and Airport Size

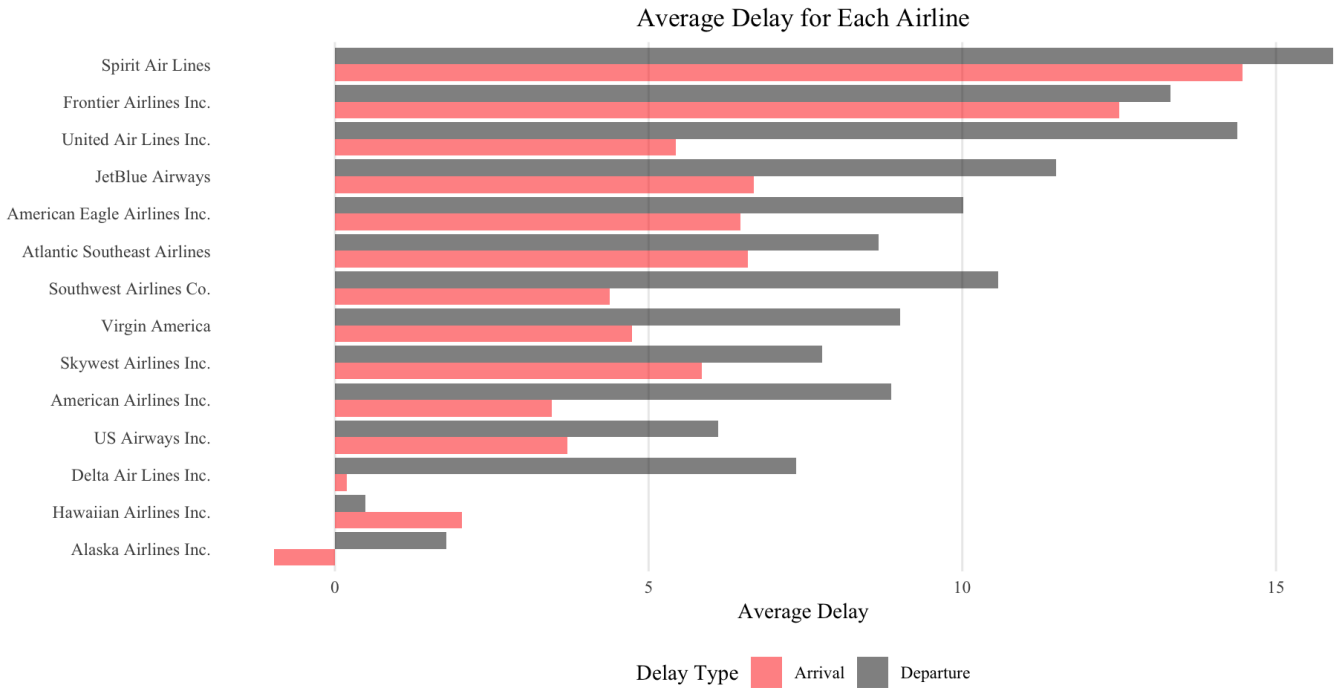
## (1) AIRLINE

```
red <- colorRampPalette(c("#FFBEB2", "red"))
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         legend.position = " ", panel.grid.minor = element_blank(), panel.grid.major.y =
 element_blank())
p9 <- FLG %>% group_by(AIRLINE) %>% summarize(number = n()) %>% mutate(Proportion = numb
er/sum(number)) %>%
      transform(AIRLINE = reorder(AIRLINE, Proportion)) %>%
      ggplot(aes(x=AIRLINE, y=Proportion, fill=factor(Proportion))) + geom_bar(stat="ide
ntity", alpha=0.6) +
      coord_flip() + geom_text(aes(label = scales::percent(Proportion, digits = 2), y =
 Proportion+0.02),
      family = "Times New Roman") + labs(title = "Number of Flights for Each Airline", x
 = " ") +
      scale_fill_manual(values = red(14)) + theme +scale_y_continuous(labels = percent,
 limits = c(0, 0.25))

p10 <- FLG %>% group_by(ID, AIRLINE) %>% summarize(`Average Delay` = mean(DELAY)) %>%
       transform(AIRLINE = reorder(AIRLINE, `Average Delay`)) %>%
       ggplot(aes(x=AIRLINE,y=`Average.Delay`,fill=ID)) + geom_bar(stat="identity",posit
ion="dodge",alpha=0.5) +
       coord_flip() + theme + theme(legend.position="bottom") + scale_fill_manual(values
=c("red", "black")) +
       labs(title = "Average Delay for Each Airline", x = " ", fill = "Delay Type", y =
"Average Delay")
```

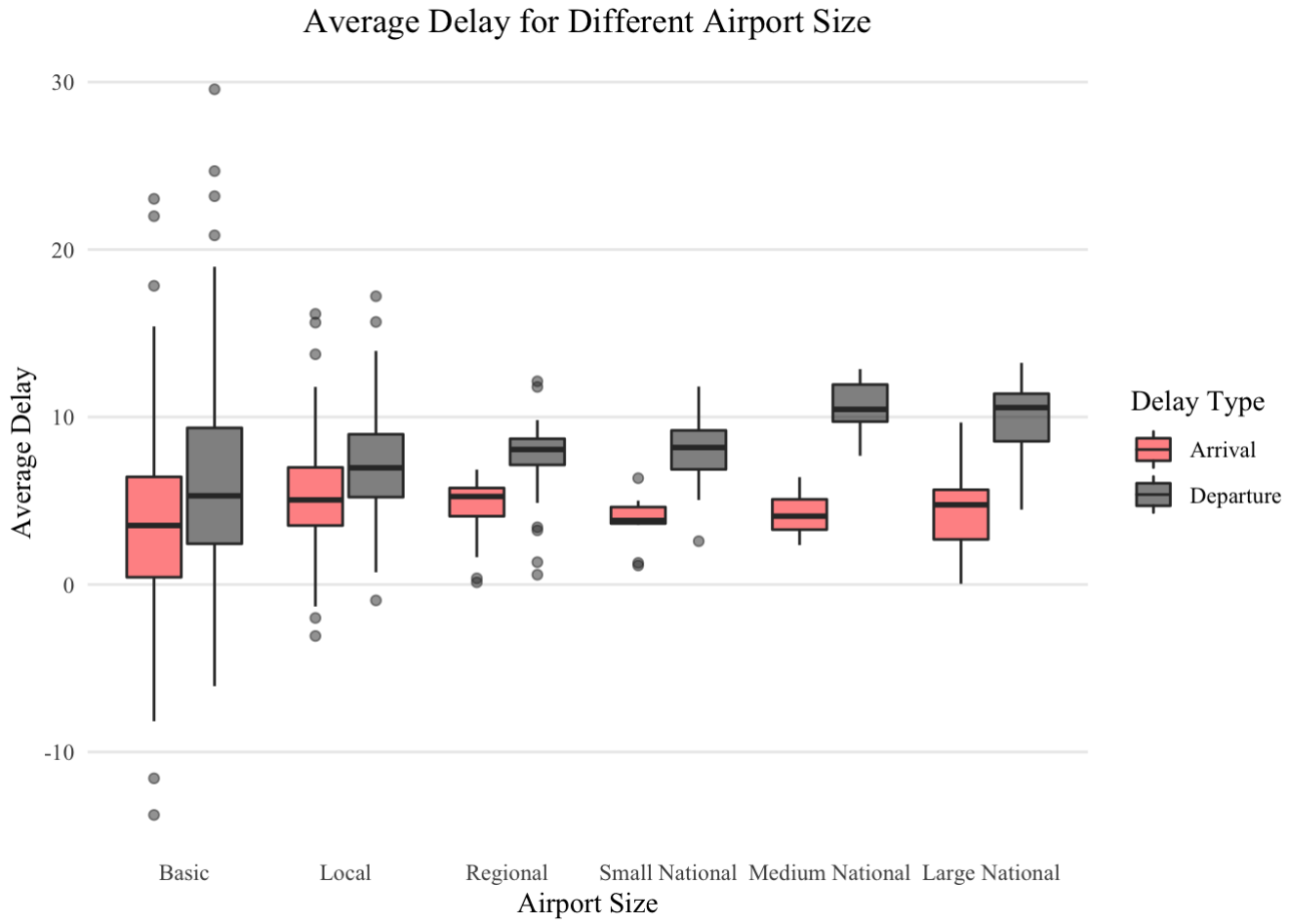
```
grid.newpage()  #create a new grid for the plots
pushViewport(viewport(layout = grid.layout(2,1))) #change the composition of the grid
print(p9, vp = vplayout(1,1))
print(p10, vp = vplayout(2,1))
```


Number of Flights for Each Airline
Southwest Airlines Co.
Delta Air Lines Inc.
American Airlines Inc.
Skywest Airlines Inc.
Atlantic Southeast Airlines
United Air Lines Inc.
American Eagle Airlines Inc.
JetBlue Airways
US Airways Inc.
Alaska Airlines Inc.
Spirit Air Lines
Frontier Airlines Inc.
Hawaiian Airlines Inc.
Virgin America
21.7%
15.2%
12.5%
10.1%
9.7%
8.9%
4.9%
4.6%
3.4%
3.0%
2.0%
1.6%
1.3%
1.1%
0.0%
5.0%
10.0%
15.0%
20.0%
25.0%
Proportion


Average Delay for Each Airline
Spirit Air Lines
Frontier Airlines Inc.
United Air Lines Inc.
JetBlue Airways
American Eagle Airlines Inc.
Atlantic Southeast Airlines
Southwest Airlines Co.
Virgin America
Skywest Airlines Inc.
American Airlines Inc.
US Airways Inc.
Delta Air Lines Inc.
Hawaiian Airlines Inc.
Alaska Airlines Inc.
0
5
10
15
Average Delay
Delay Type
Arrival
Departure

# (2) AIRPORT

```
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         panel.grid.minor = element_blank(), panel.grid.major.x = element_blank())
p11 <- FLG %>% group_by(AIRPORT, ID) %>% summarize(FLIGHT_NUMBER = n(), AIRPORT_SIZE = u
nique(AIRPORT_SIZE),
       AVERAGE_DELAY = mean(DELAY)) %>% transform(AIRPORT = reorder(AIRPORT, AIRPORT_SIZ
E)) %>%
       ggplot(aes(x = AIRPORT_SIZE, y = AVERAGE_DELAY, fill = ID)) + geom_boxplot(alpha
 = 0.5) + theme +
       labs(x="Airport Size",y="Average Delay",fill="Delay Type") + scale_fill_manual(va
lues=c("red", "black")) +
       ggtitle("Average Delay for Different Airport Size"); p11
```




We also analyzed the average delay of each airline and found that Spirit Airlines tops the list of airlines with the worst delays, with an average delay of 16.62 minutes. Frontier Airlines and United Airlines come in close at second and third places with average delays of about 15 and 14 minutes respectively. We expected to find mostly low-cost airlines in the top 5 of the delay ranking because of their lack of limited customer support in addressing flight disruptions. We were, however, surprised to find the largest airlines such as United, JetBlue, Southwest, and American with above average delays. We expected them to have streamlined operations to limit disruptions.

We also found that the largest airports tended to have the worst delays, which suggests that congestion (number of flights) and operational efficiency of the air control can play a huge role in solving flight delays. We also believe that delays in bigger airports tend to affect a larger number of flights, further increasing the delays.

# 3.4 Delay by Region and Season

## REGION

```
FLG_v_state <- as.data.frame(FLG %>% group_by(ID, STATE) %>%
                                    rename(state = STATE) %>% summarize(`Average Delay`
= mean(DELAY)))
FLG_v_state_dep <- FLG_v_state %>% filter(ID == "Departure") %>% mutate(`Category(min)`
 =cut(`Average Delay`,
                   breaks=c(-Inf,0,3,6,9,12,15,Inf),labels=c("quicker","0-3","3-6","6-9"
,"9-12","12-15","15+")))
FLG_v_state_arr <- FLG_v_state %>% filter(ID == "Arrival") %>% mutate(`Category(min)` =c
ut(`Average Delay`,
                   breaks=c(-Inf,0,3,6,9,12,15,Inf),labels=c("quicker","0-3","3-6","6-9"
,"9-12","12-15","15+")))

t1 <- FLG_v_state_dep %>% arrange(desc(`Average Delay`)) %>% dplyr::select(-ID) %>% head
(n = 5L)
t2 <- FLG_v_state_dep %>% arrange(`Average Delay`) %>% dplyr::select(-ID) %>% head(n = 5
L)
t3 <- FLG_v_state_arr %>% arrange(desc(`Average Delay`)) %>% dplyr::select(-ID) %>% head
(n = 5L)
t4 <- FLG_v_state_arr %>% arrange(`Average Delay`) %>% dplyr::select(-ID) %>% head(n = 5
L)

theme <- theme_minimal(base_family = "Times New Roman") + theme(line = element_blank(),
         axis.title = element_blank(), axis.text = element_blank(), plot.title = element
_text(hjust = 0.5))
g1 <- qplot(1:10, 1:10, geom = "blank", main = "Top 5 States with the Highest Departure
 Delay") +
      theme + annotation_custom(grob = tableGrob(t1))
g2 <- qplot(1:10, 1:10, geom = "blank", main = "Top 5 States with the Lowest Departure D
elay") +
      theme + annotation_custom(grob = tableGrob(t2))
g3 <- qplot(1:10, 1:10, geom = "blank", main = "Top 5 States with the Highest Arrival De
lay") +
      theme + annotation_custom(grob = tableGrob(t3))
g4 <- qplot(1:10, 1:10, geom = "blank", main = "Top 5 States with the Lowest Arrival Del
ay") +
      theme + annotation_custom(grob = tableGrob(t4))

p12 <- plot_usmap(data = FLG_v_state_dep, values = "Category(min)", lines = "white") +
       theme(legend.position = "right") + ggtitle("Average Departure Delay throughout th
e States") +
       scale_fill_manual(name = "Average Delay (minutes)",
       values = c("#B9DDF1", "#9CC2DB", "#7FA7C4", "#DC747E", "#C5435C", "#AE123A"), na.
value = "grey66")
p13 <- plot_usmap(data = FLG_v_state_arr, values = "Category(min)", lines = "white") +
       theme(legend.position = "right") + ggtitle("Average Arrival Delay throughout the
 States") +
       scale_fill_manual(name = "Average Delay (minutes)",
       values = c("#B9DDF1", "#9CC2DB", "#7FA7C4", "#AE123A"), na.value = "grey66")
```

```
grid.newpage()  #create a new grid for the plots
pushViewport(viewport(layout = grid.layout(4,3))) #change the composition of the grid
print(g1, vp = vplayout(1,1))
print(g2, vp = vplayout(2,1))
print(g3, vp = vplayout(3,1))
print(g4, vp = vplayout(4,1))
print(p12, vp = vplayout(1:2,2:3))
print(p13, vp = vplayout(3:4,2:3))
```

Top 5 States with the Highest Departure Delay

| | state | Average Delay | Category(min) |
|---|---|---|---|
| 1 | DE | 29.56842 | 15+ |
| 2 | AS | 20.84483 | 15+ |
| 3 | GU | 15.08333 | 15+ |
| 4 | NJ | 13.00955 | 12-15 |
| 5 | MD | 12.86168 | 12-15 |

Top 5 States with the Lowest Departure Delay

| | state | Average Delay | Category(min) |
|---|---|---|---|
| 1 | MT | 2.044940 | 0-3 |
| 2 | HI | 2.166358 | 0-3 |
| 3 | AK | 2.856291 | 0-3 |
| 4 | UT | 4.404211 | 3-6 |
| 5 | WY | 4.721718 | 3-6 |

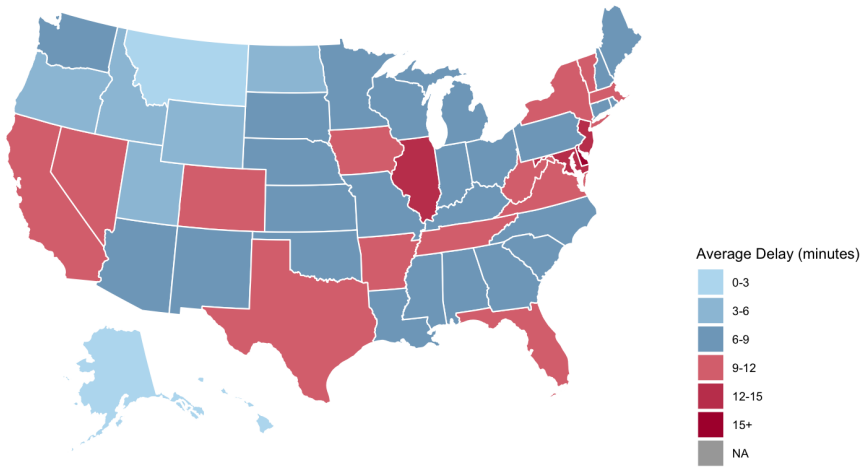
Top 5 States with the Highest Arrival Delay

| | state | Average Delay | Category(min) |
|---|---|---|---|
| 1 | DE | 21.989474 | 15+ |
| 2 | GU | 14.566667 | 12-15 |
| 3 | VT | 8.385727 | 6-9 |
| 4 | AS | 7.844828 | 6-9 |
| 5 | NY | 7.617003 | 6-9 |

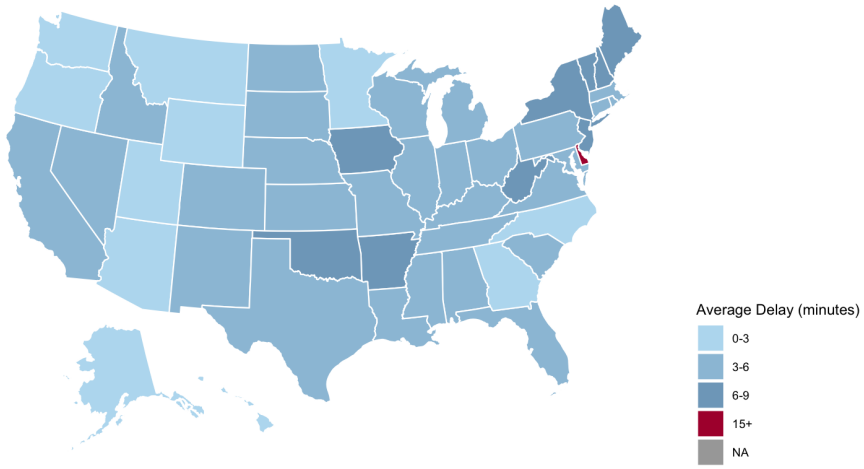
Top 5 States with the Lowest Arrival Delay

| | state | Average Delay | Category(min) |
|---|---|---|---|
| 1 | UT | 0.1271539 | 0-3 |
| 2 | MT | 0.4389679 | 0-3 |
| 3 | OR | 1.7601758 | 0-3 |
| 4 | WA | 1.8593922 | 0-3 |
| 5 | GA | 1.9840744 | 0-3 |

Average Departure Delay throughout the States




Average Arrival Delay throughout the States

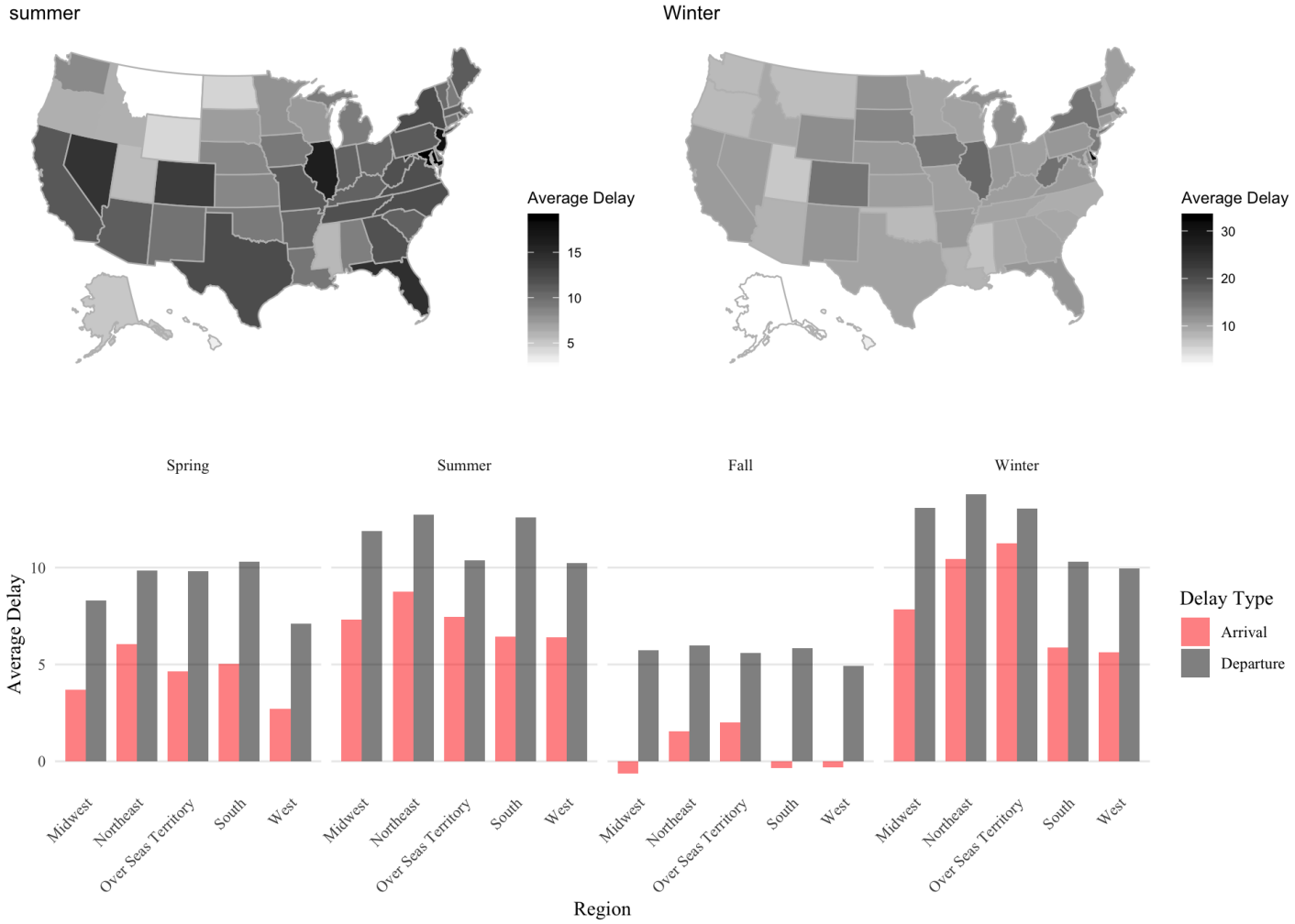



# SEASON

```
FLG_v_state_season <- as.data.frame(FLG %>% group_by(ID, SEASON, STATE) %>% rename(state
= STATE) %>%
                                                summarize(`Average Delay` = mean(DELAY)))
# Summer
summert <- FLG_v_state_season %>% filter(SEASON == "Summer") %>% filter(ID == "Departur
e")
summer <- plot_usmap(data = summert, values = "Average Delay", lines = "grey") +
          scale_fill_gradient(name = "Average Delay", low = "white", high = "black") +
          theme(legend.position = "right") + ggtitle("summer")
# Winter
wintert <- FLG_v_state_season %>% filter(SEASON == "Winter") %>% filter(ID == "Departur
e")
winter <- plot_usmap(data = wintert, values = "Average Delay", lines = "grey") +
          scale_fill_gradient(name = "Average Delay", low = "white", high = "black") +
          theme(legend.position = "right") + ggtitle("Winter")

theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(),
         panel.grid.minor = element_blank(), panel.grid.major.x = element_blank(),
         axis.text.x = element_text(angle = 45, hjust = 1))
season <- FLG %>% group_by(ID, REGION, SEASON) %>% summarize(`Average Delay` = mean(DELA
Y)) %>%
          ggplot(aes(x = REGION, y = `Average Delay`, fill = ID, group = ID)) +
          geom_bar(stat = "identity", position = "dodge", width = 0.8, alpha = 0.5) +
          facet_wrap(.~SEASON, nrow = 1) + theme + labs(fill = "Delay Type", x = "Regio
n") +
          scale_fill_manual(values = c("red", "black"))
```

```
grid.newpage()  #create a new grid for the plots
pushViewport(viewport(layout = grid.layout(2,2))) #change the composition of the grid
print(summer, vp = vplayout(1, 1))
print(winter, vp = vplayout(1, 2))
print(season, vp = vplayout(2, 1:2))
```




We also looked at delays by region for summer and winter, which have the highest proportion of delays. In winter, we found that the Midwest, South and North-East States have the highest departure delays which are somewhat consistent with the hypothesis that adverse weather conditions in these states may be causing delays. During summer months, departure delays are highest in western and southern states, such as Florida, Texas, Nevada and California have the highest delays. We also have Illinois, Maryland, and New Jersey which all serve major economic hubs such as Chicago, Washington D.C., the New York City and Newark.

---

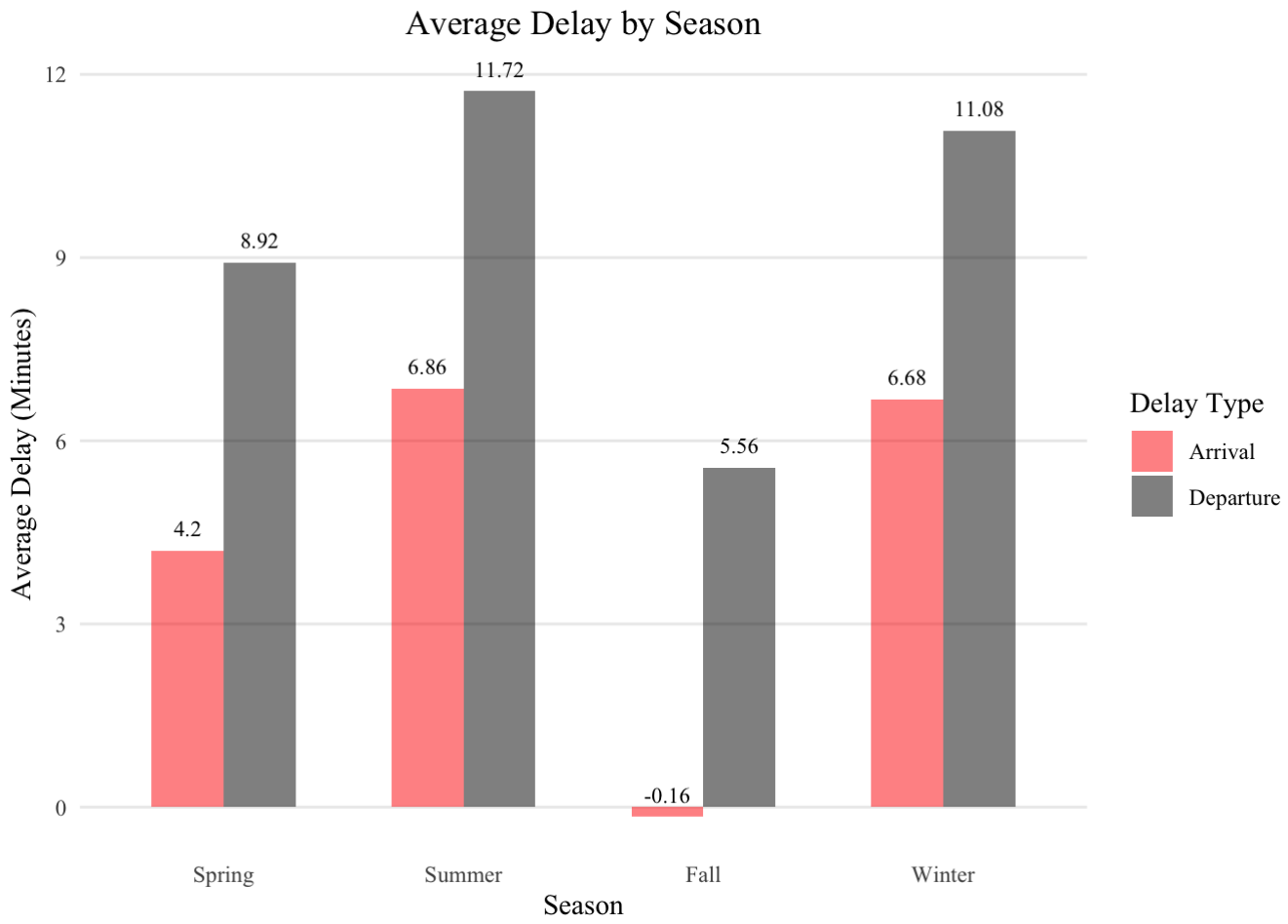
# 4 STATISTICAL ANALYSES

In this section, we conducted tests to test that whether departure delays for each season, region, distance group, and airport size are statistically different from the average departure delay of the whole year.

## 4.1 SEASON

```
#create average delay by season and ID
Season <- FLG %>% group_by(SEASON, ID) %>% summarise(DELAY = mean(DELAY))

#plot season by average delay
theme <- theme_minimal(base_family = "Times New Roman") + theme(plot.title = element_tex
t(hjust = 0.5),
         panel.grid.minor = element_blank(), panel.grid.major.x = element_blank())
Season %>% ggplot(aes( x = SEASON, y = DELAY, fill = ID)) +
           geom_bar(stat="identity", position="Dodge", alpha=0.5, width=0.6) +
           geom_text(aes(label=round(DELAY,2)), position=position_dodge(width =0.6),
           family="Times New Roman",size=3,vjust=-1) + ggtitle("Average Delay by Season"
) +
           labs(fill="Delay Type", x="Season", y="Average Delay (Minutes)") +
           theme + scale_fill_manual(values = c("red", "black"))
```



```
# conduct t-tests for each season
all_year_dep_delay <- FLG %>% filter(ID == "Departure") %>%  dplyr::select(DELAY)
summer_dep_delay <- FLG %>% filter(SEASON == "Summer" & ID == "Departure")%>% dplyr::sel
ect(DELAY)
winter_dep_delay <- FLG %>% filter(SEASON == "Winter" & ID == "Departure")%>% dplyr::sel
ect(DELAY)
fall_dep_delay <- FLG %>% filter(SEASON == "Fall" & ID == "Departure")%>% dplyr::select
(DELAY)
spring_dep_delay <- FLG %>% filter(SEASON == "Spring" & ID == "Departure")%>% dplyr::sel
ect(DELAY)

t1 <- tidy(t.test(all_year_dep_delay$DELAY,summer_dep_delay$DELAY))[2:5]  #summer vs all
 year
t2 <- tidy(t.test(all_year_dep_delay$DELAY, winter_dep_delay$DELAY))[2:5]  #winter vs al
l year
t3 <- tidy(t.test(all_year_dep_delay$DELAY, fall_dep_delay$DELAY))[2:5]  #fall vs all ye
ar
t4 <- tidy(t.test(all_year_dep_delay$DELAY, spring_dep_delay$DELAY))[2:5]  #spring vs al
l year

t_season <- data.frame(rbind(t1,t2,t3,t4))%>%rename(`Delay for 12months`=estimate1,`Dela
y by Season`=estimate2)
rownames(t_season) <- c("Summer","Winter","Fall","Spring"); pandoc.table(t_season, style
 = "grid")
```

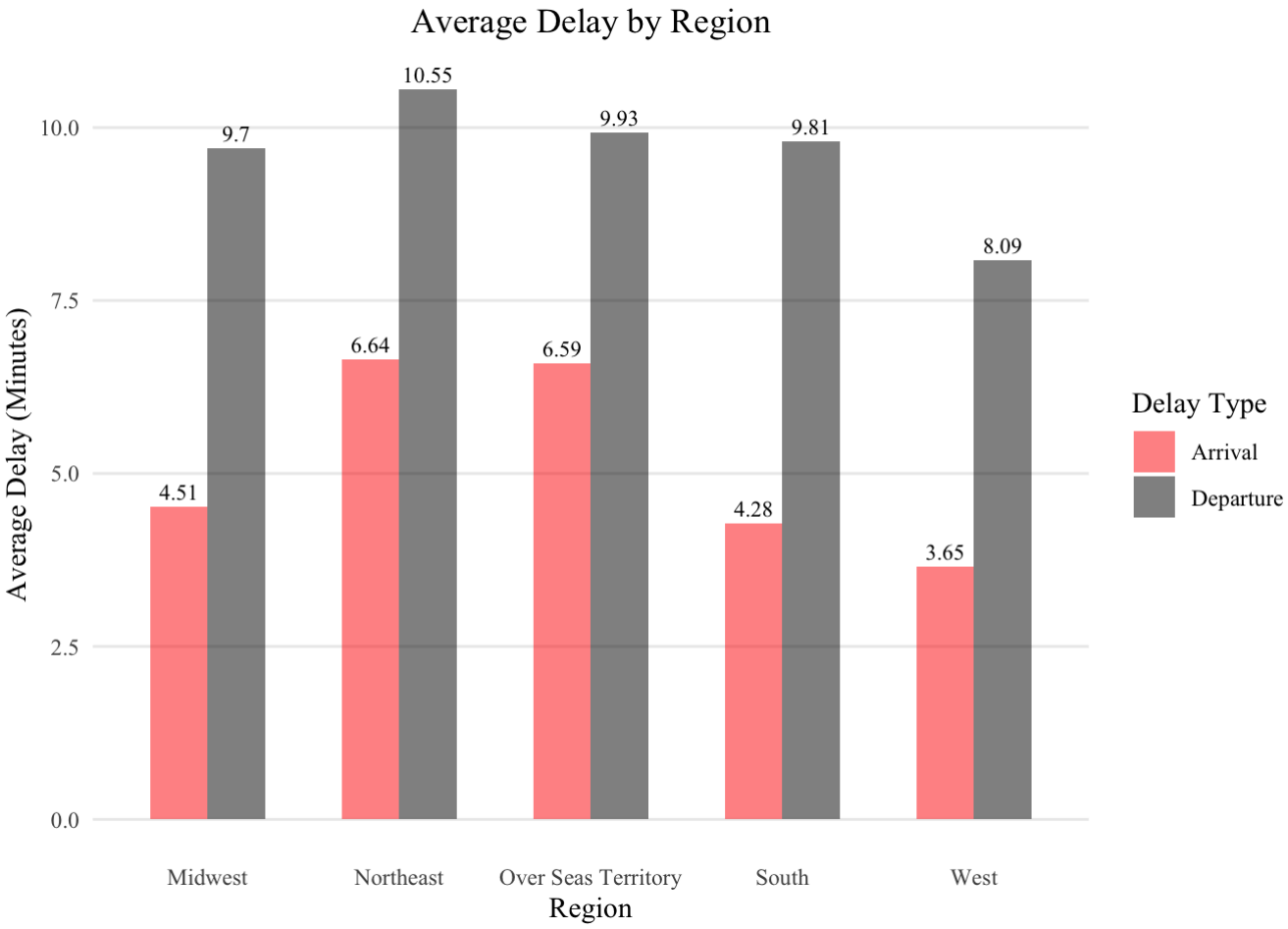
```
##
##
## +------------+--------------------+-----------------+-----------+-----------+
## |       | Delay for 12months | Delay by Season | statistic |  p.value  |
## +============+====================+=================+===========+===========+
## | **Summer** |       9.339        |      11.72      |  -67.28   |     0     |
## +------------+--------------------+-----------------+-----------+-----------+
## | **Winter** |       9.339        |      11.08      |  -46.14   |     0     |
## +------------+--------------------+-----------------+-----------+-----------+
## |  **Fall**  |       9.339        |      5.561      |   122.3   |     0     |
## +------------+--------------------+-----------------+-----------+-----------+
## | **Spring** |       9.339        |      8.92       |   12.4    | 2.611e-35 |
## +------------+--------------------+-----------------+-----------+-----------+
```

Our analyses show that average departure delays by season are statistically different from the average delay of the whole year, we therefore conclude that Summer and Winter have above average departure delays while spring and fall have below average departure delays.

# 4.2 REGION

```
#REGION t-tests
REGION <- FLG %>% group_by(REGION, ID) %>% summarise(DELAY = mean(DELAY))

#plot season by average delay
REGION %>% ggplot(aes( x = REGION, y = DELAY, fill = ID)) +
          geom_bar(stat="identity", position="Dodge", alpha=.5, width=.6) + geom_text(ae
s(label=round(DELAY,2)),
          position=position_dodge(width = 0.6),family="Times New Roman", size=3, vjust=-
.5) +
          labs(fill="Delay Type", x="Region", title="Average Delay by Region", y="Averag
e Delay (Minutes)") +
          theme + scale_fill_manual(values = c("red", "black"))
```



```
all_US_dep_delay <- FLG %>% filter(ID == "Departure") %>%  dplyr::select(DELAY)
NE_dep_delay <- FLG %>% filter(REGION == "Northeast" & ID == "Departure")%>% dplyr::sele
ct(DELAY)
SOUTH_dep_delay <- FLG %>% filter(REGION == "South"& ID == "Departure")%>% dplyr::select
(DELAY)
MIDWEST_dep_delay <- FLG %>% filter(REGION == "Midwest" & ID == "Departure")%>% dplyr::s
elect(DELAY)
WEST_dep_delay <- FLG %>% filter(REGION == "West" & ID == "Departure")%>% dplyr::select
(DELAY)
overseas_dep_delay <- FLG %>% filter(REGION == "Over Seas Territory" & ID == "Departure"
)%>% dplyr::select(DELAY)

ne <- tidy(t.test(all_US_dep_delay$DELAY, NE_dep_delay$DELAY))[2:5]  #ne vs US
sou <- tidy(t.test(all_US_dep_delay$DELAY, SOUTH_dep_delay$DELAY))[2:5]  #sou vs US
mw <- tidy(t.test(all_US_dep_delay$DELAY, MIDWEST_dep_delay$DELAY))[2:5]  #mw vs US
we <- tidy(t.test(all_US_dep_delay$DELAY, WEST_dep_delay$DELAY))[2:5]  #we vs US
overseas <- tidy(t.test(all_US_dep_delay$DELAY, overseas_dep_delay$DELAY))[2:5]  #overse
a states vs US

t_region <- data.frame(rbind(ne,sou,mw,we,overseas)) %>%
            rename(`Average for US` = estimate1, `Average by Region` = estimate2)
rownames(t_region) <- c("NE","South","Midwest","West", "Overseas"); pandoc.table(t_regio
n, style = "grid")
```

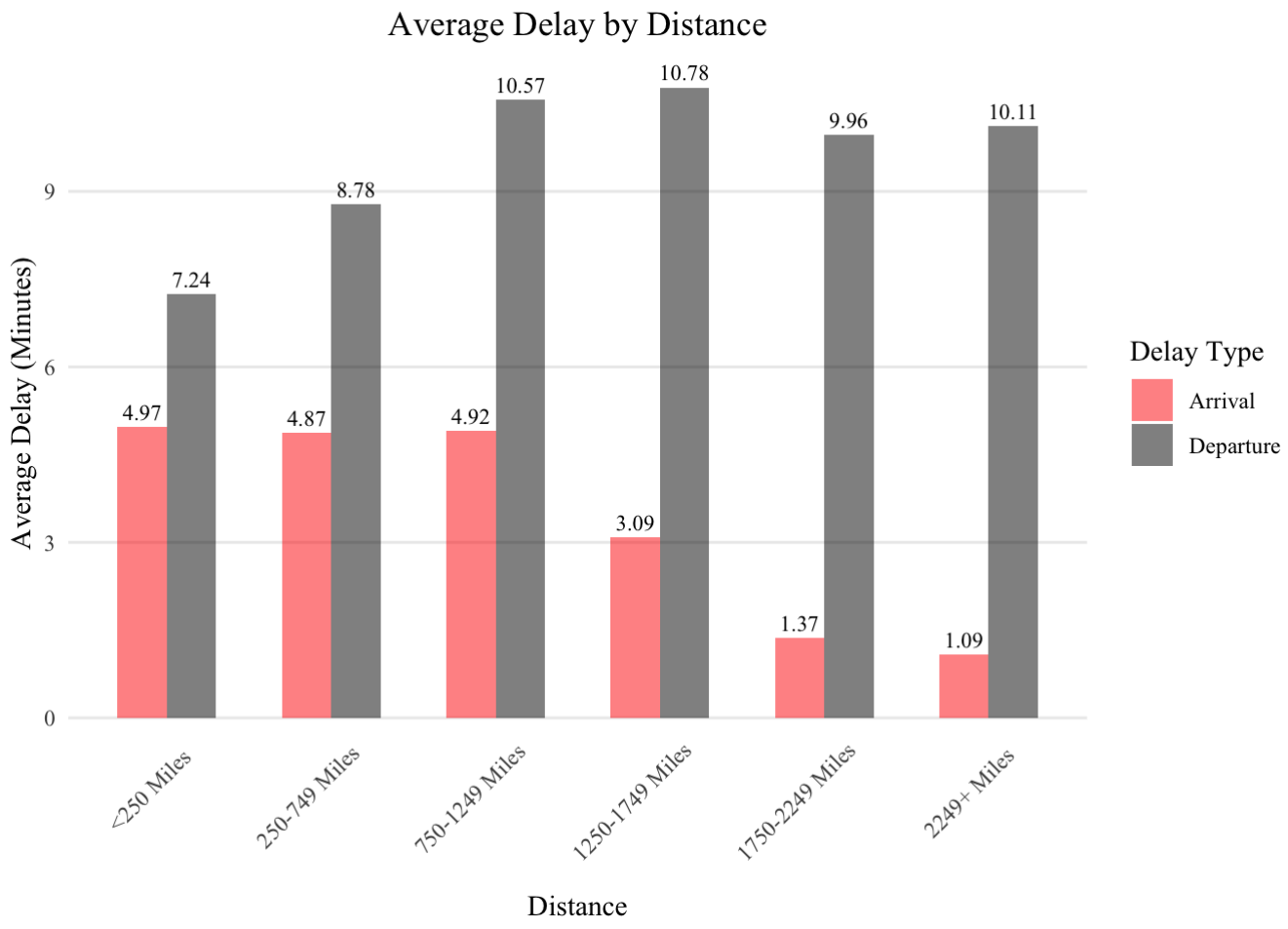
```
##
##
## +--------------+----------------+-------------------+-----------+------------+
## |         | Average for US | Average by Region | statistic |  p.value   |
## +==============+================+===================+===========+============+
## |    **NE**    |     9.339      |       10.55       |  -23.45   | 1.351e-121 |
## +--------------+----------------+-------------------+-----------+------------+
## |  **South**   |     9.339      |       9.806       |  -15.93   | 3.991e-57  |
## +--------------+----------------+-------------------+-----------+------------+
## | **Midwest**  |     9.339      |       9.701       |  -8.889   |  6.15e-19  |
## +--------------+----------------+-------------------+-----------+------------+
## |   **West**   |     9.339      |       8.086       |   41.59   |     0      |
## +--------------+----------------+-------------------+-----------+------------+
## | **Overseas** |     9.339      |       9.925       |  -2.262   |  0.02371   |
## +--------------+----------------+-------------------+-----------+------------+
```

We also looked at average delays by region and found that all regions are statistically different from the average delay of the entire U.S. We, therefore, conclude that only the West has below average delays.

# 4.3 DISTANCE

```
#DISTANCEGROUP t-tests
order <- c("<250 Miles","250-749 Miles","750-1249 Miles","1250-1749 Miles","1750-2249 Mi
les","2249+ Miles")
DISTANCEGROUP <- FLG %>% group_by(DISTANCEGROUP, ID) %>% summarise(DELAY = mean(DELAY))%
>% ungroup %>%
                  mutate(DISTANCEGROUP = factor(DISTANCEGROUP, levels =order))%>% arrange
(DISTANCEGROUP)

#plot season by average delay
DISTANCEGROUP %>% ggplot(aes( x = DISTANCEGROUP, y = DELAY, fill = ID)) +
                  geom_bar(stat="identity", position="Dodge", alpha=.5, width=.6) +
                  scale_fill_manual(values=c("red","black")) + ggtitle("Average Delay by
 Distance") +
                  geom_text(aes(label=round(DELAY,2)), position=position_dodge(width =
0.6),
                  family="Times New Roman", size=3,  vjust=-.5) + theme +
                  theme(axis.text.x = element_text(angle = 45, hjust = 0.8)) +
                  labs(fill="Delay Type", x="Distance", y="Average Delay (Minutes)")
```



```
all_dist <- FLG %>% filter(ID == "Departure") %>%  dplyr::select(DELAY)
less_250 <- FLG %>% filter(DISTANCEGROUP == "<250 Miles" & ID == "Departure")%>% dplyr::
select(DELAY)
d250_750 <- FLG %>% filter(DISTANCEGROUP == "250-749 Miles" & ID == "Departure")%>% dply
r::select(DELAY)
d750_1250 <- FLG %>% filter(DISTANCEGROUP == "750-1249 Miles" & ID == "Departure")%>% dp
lyr::select(DELAY)
d1250_1750 <- FLG %>% filter(DISTANCEGROUP == "1250-1749 Miles" & ID == "Departure")%>%
 dplyr::select(DELAY)
d1750_2250 <- FLG %>% filter(DISTANCEGROUP == "1750-2249 Miles" & ID == "Departure")%>%
 dplyr::select(DELAY)
d2250plus <- FLG %>% filter(DISTANCEGROUP == "2249+ Miles"& ID == "Departure")%>% dply
r::select(DELAY)

tless_250 <- tidy(t.test(all_dist$DELAY, less_250$DELAY))[2:5]
t250_750 <- tidy(t.test(all_dist$DELAY, d250_750$DELAY))[2:5]
t750_1250 <- tidy(t.test(all_dist$DELAY, d750_1250$DELAY))[2:5]
t1250_1750 <- tidy(t.test(all_dist$DELAY, d1250_1750$DELAY))[2:5]
t1750_2250 <- tidy(t.test(all_dist$DELAY, d1750_2250$DELAY))[2:5]
t2250plus <- tidy(t.test(all_dist$DELAY, d2250plus$DELAY))[2:5]

t_distance <- data.frame(rbind(tless_250, t250_750, t750_1250 ,t1250_1750, t1750_2250, t
2250plus)) %>%
              rename(`all Distances` = estimate1, `Delay by Distance` = estimate2)
rownames(t_distance) <- c("<250 Miles","250-749 Miles","750-1249 Miles","1250-1749 Mile
s","1750-2249 Miles", "2249+ Miles"); pandoc.table(t_distance, style = "grid")
```

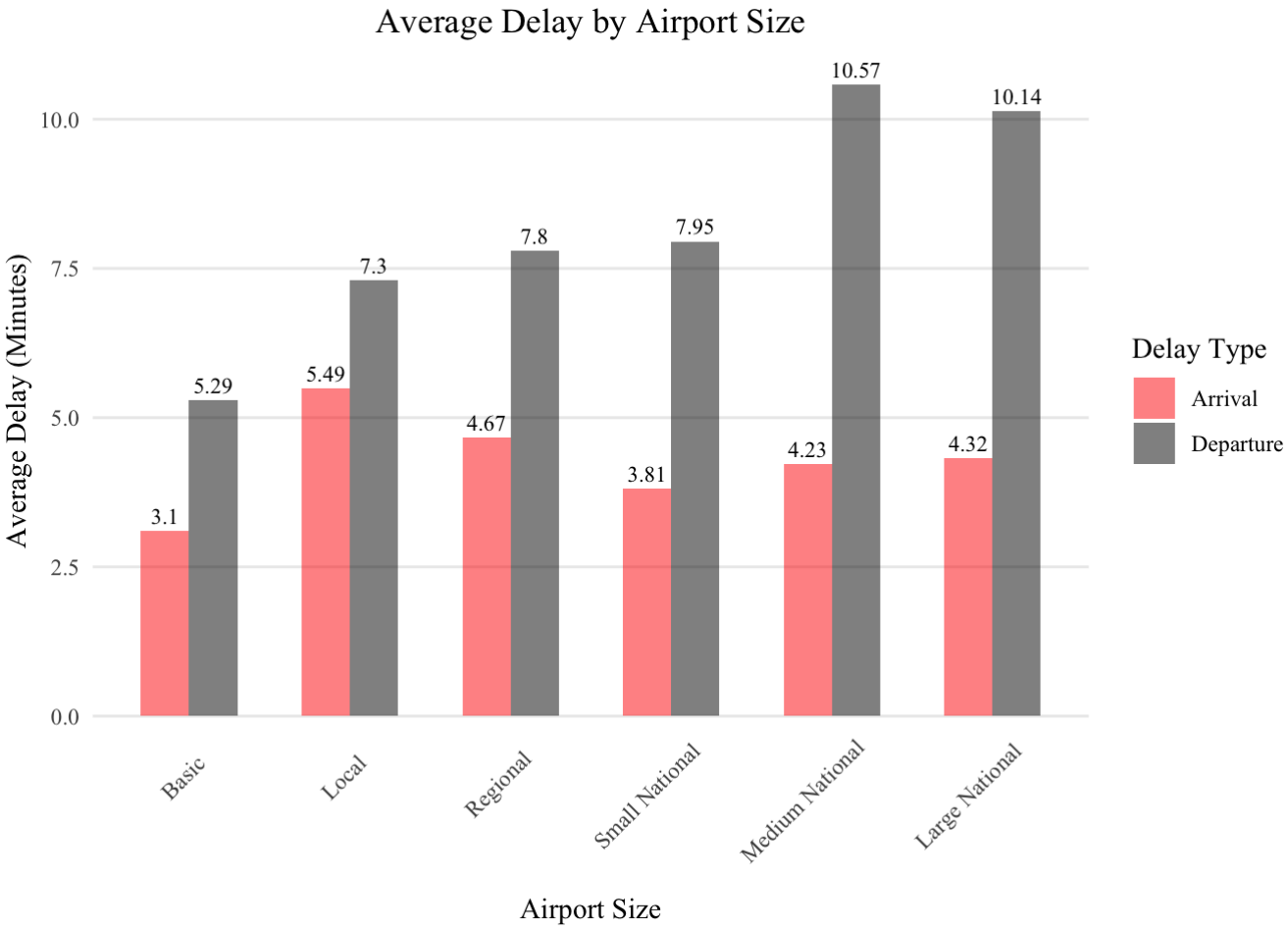
```
##
##
## +--------------------+---------------+--------------------+-----------+
## |               | all Distances | Delay by Distance | statistic |
## +====================+===============+====================+===========+
## |   **<250 Miles**   |     9.339     |       7.237        |   47.77   |
## +--------------------+---------------+--------------------+-----------+
## |  **250-749 Miles** |     9.339     |       8.78         |   20.4    |
## +--------------------+---------------+--------------------+-----------+
## | **750-1249 Miles** |     9.339     |       10.57        |  -34.56   |
## +--------------------+---------------+--------------------+-----------+
## | **1250-1749 Miles** |     9.339     |       10.78        |  -26.07   |
## +--------------------+---------------+--------------------+-----------+
## | **1750-2249 Miles** |     9.339     |       9.963        |  -8.391   |
## +--------------------+---------------+--------------------+-----------+
## |   **2249+ Miles**  |     9.339     |       10.11        |  -9.676   |
## +--------------------+---------------+--------------------+-----------+
##
## Table: Table continues below
##
##
##
## +--------------------+------------+
## |               |  p.value   |
## +====================+============+
## |   **<250 Miles**   |     0      |
## +--------------------+------------+
## |  **250-749 Miles** | 1.525e-92  |
## +--------------------+------------+
## | **750-1249 Miles** | 1.147e-261 |
## +--------------------+------------+
## | **1250-1749 Miles** | 1.053e-149 |
## +--------------------+------------+
## | **1750-2249 Miles** | 4.827e-17  |
## +--------------------+------------+
## |   **2249+ Miles**  | 3.829e-22  |
## +--------------------+------------+
```

We further investigated average delays by the distance of flight and found that shorter (less than 250 miles) have delays that are statistically different from the average and have lower than average delays, which contrasts with our hypothesis that shorter flights would have higher departure delays. It turns out flights with distances that range from 750 to 1750 have above average delays.

# 4.4 AIRPORT

```
# AIRPORT t-tests
order <- c('Basic', 'Local',  'Regional', 'Small National','Medium National', 'Large Nat
ional' )
AIRPORT <- FLG %>% group_by(AIRPORT_SIZE, ID) %>% summarise(DELAY = mean(DELAY)) %>%
           ungroup %>% arrange(AIRPORT_SIZE)

#plot season by average delay
AIRPORT %>% ggplot(aes( x = AIRPORT_SIZE, y = DELAY, fill = ID)) +
            geom_bar(stat="identity", position="Dodge", alpha=.5, width=.6) +
            geom_text(aes(label=round(DELAY,2)), position=position_dodge(width=0.6),fami
ly="Times New Roman",
            size=3, vjust=-.5) + labs(fill = "Delay Type", x="Airport Size", y = "Averag
e Delay (Minutes)") +
            theme + theme(axis.text.x = element_text(angle = 45, hjust = 0.8)) +
            scale_fill_manual(values = c("red", "black")) + ggtitle("Average Delay by Ai
rport Size")
```



```
# t-tests
all_airport <- FLG %>% filter(ID == "Departure") %>%  dplyr::select(DELAY)
basic <- FLG %>% filter(AIRPORT_SIZE == "Basic" & ID == "Departure")%>% dplyr::select(DE
LAY)
local <- FLG %>% filter(AIRPORT_SIZE == "Local" & ID == "Departure")%>% dplyr::select(DE
LAY)
reg <- FLG %>% filter(AIRPORT_SIZE == "Regional" & ID == "Departure")%>% dplyr::select(D
ELAY)
small_nat <- FLG %>% filter(AIRPORT_SIZE == "Small National" & ID == "Departure")%>% dpl
yr::select(DELAY)
medium_nat <- FLG %>% filter(AIRPORT_SIZE == "Medium National" & ID == "Departure")%>% d
plyr::select(DELAY)
large_nat <- FLG %>% filter(AIRPORT_SIZE == "Large National" & ID == "Departure")%>% dpl
yr::select(DELAY)

tbasic <- tidy(t.test(all_airport$DELAY, basic$DELAY))[2:5]
tlocal <- tidy(t.test(all_airport$DELAY, local$DELAY))[2:5]
treg <- tidy(t.test(all_airport$DELAY, reg$DELAY))[2:5];
tsmall_nat <- tidy(t.test(all_airport$DELAY, small_nat$DELAY))[2:5]
tmedium_nat <- tidy(t.test(all_airport$DELAY, medium_nat$DELAY))[2:5]
tlarge_nat <- tidy(t.test(all_airport$DELAY, large_nat$DELAY))[2:5]

t_airport <- data.frame(rbind(tbasic, tlocal, treg ,tsmall_nat, tmedium_nat, tlarge_na
t)) %>%
             rename(`all Aiports` = estimate1, `by Size` = estimate2)
rownames(t_airport) <- c('Basic', 'Local',  'Regional', 'Small National','Medium Nationa
l', 'Large National')
pandoc.table(t_airport, style = "grid")
```

```
##
##
## +---------------------+-------------+---------+-----------+------------+
## |                | all Aiports | by Size | statistic |  p.value   |
## +=====================+=============+=========+===========+============+
## |      **Basic**      |    9.339    |  5.289  |   21.53   | 2.314e-102 |
## +---------------------+-------------+---------+-----------+------------+
## |      **Local**      |    9.339    |  7.299  |   34.73   | 5.188e-264 |
## +---------------------+-------------+---------+-----------+------------+
## |    **Regional**     |    9.339    |  7.796  |   32.42   | 2.02e-230  |
## +---------------------+-------------+---------+-----------+------------+
## | **Small National**  |    9.339    |  7.954  |   28.43   | 9.683e-178 |
## +---------------------+-------------+---------+-----------+------------+
## | **Medium National** |    9.339    |  10.57  |  -26.81   | 2.745e-158 |
## +---------------------+-------------+---------+-----------+------------+
## | **Large National**  |    9.339    |  10.14  |  -31.23   | 4.748e-214 |
## +---------------------+-------------+---------+-----------+------------+
```

Lastly, we tested if the average departure delay by airport size differs from the average and found that medium and larger national airports have above average delays while basic, local, regional and small national airports have below average departure delays.

# 5 REGRESSIONS

In part 5, we ran two regressions: (1) a logistic regression with delay binary variable (1 if flight is delayed, and zero otherwise) as a dependent variable and (2) a linear regression with delay time (in minutes) as a dependent variable. We wanted to find a model from the 2015 flights data, which we can use to predict the probability of delay and predict the severity of delay in minutes on new data set (e.g. new 2016 flights data). Because departure delay and arrival delay are highly related, and that arrival delay is caused by upstream departure delays, we focused our regression analysis only on departure delay.

## 5.1 PREPARATION FOR REGRESSION ANALYSES

```
# filter to departure and aggregate the data set
agg <- FLG %>% filter( ID == "Departure") %>%
       dplyr::select(-TAIL_NUMBER, -DAY, -SCHEDULED_TIME, -SCHEDULED_TAKEOFF_LANDING, -A
CTUAL_TAKEOFF_LANDING,
       -WHEELS_OFF, -WHEELS_ON,-ELAPSED_TIME, -YEAR, -DIVERTED, -TAXI_IN, -DELAY_TYPE)

agg <- agg %>% group_by(REGION, AIRLINE, AIRPORT_SIZE, DAY_OF_WEEK, SEASON, DISTANCEGROU
P, TIME_OF_DAY) %>%
       summarize(DELAY=mean(DELAY), TAXI_OUT=mean(TAXI_OUT)) %>% ungroup %>% mutate_if(i
s.character,as.factor)
# Create delay dummy. 1 for delay flights, 0 for non-delay flights.
agg <- agg %>% mutate(DELAY_DUMMY = if_else(agg$DELAY > 0, 1 ,0))

# Set seed
set.seed(100000000)

randOrder = order(runif(nrow(agg)))
training.data = subset(agg,randOrder < .9 * nrow(agg))
validation.data =  subset(agg,randOrder >= .9*nrow(agg)&(randOrder <= .95*nrow(agg)))
test.data = subset(agg,randOrder > .95 * nrow(agg))


prediction.error = function(lm_model, test.data){
  predicted.DELAY = predict(lm_model, test.data)
  error = sqrt(mean((predicted.DELAY-test.data$DELAY)^2))
}
```

## 5.2 LOGISTIC REGRESSION

### (1) METHODOLOGY

In this section, we built a logistic regression model to predict the probability of delay. First, we aggregate the data set by region, airline, airport size, day of the week, season, distance group and time of day. This reduced our data from 11.4 million rows to about 89,000 rows, which we further divided into training, validation and

test sets using a 90-5-5 split. The training dataset was used to build the model and validation dataset was used to select the model with best predictive ability. In the end, we used test dataset to make prediction and recommendations for FAA.

We started with a baseline regression that had the `delay` dummy (1 = delayed, 0 = No delay) as the dependent variable, and season dummies as explanatory variables and continually added variables (including interaction terms of region and season) to check for omitted variable bias. If there was omitted variable bias, we expected the coefficients to vary significantly and the adjusted R-squared to improve after adding the omitted variable. We picked the model with the lowest BIC as the best model. We found that day of week, airport size, airline, region, distance group and time of day significantly improved the explanation power of the model according to the BIC. We use a threshold of 0.5, above which we interpret the probability to be delayed. Our best model gave us a prediction accuracy of about 80%.

**(2) LOGISTIC REGRESSION FOR PROBABILITY OF DELAYS**

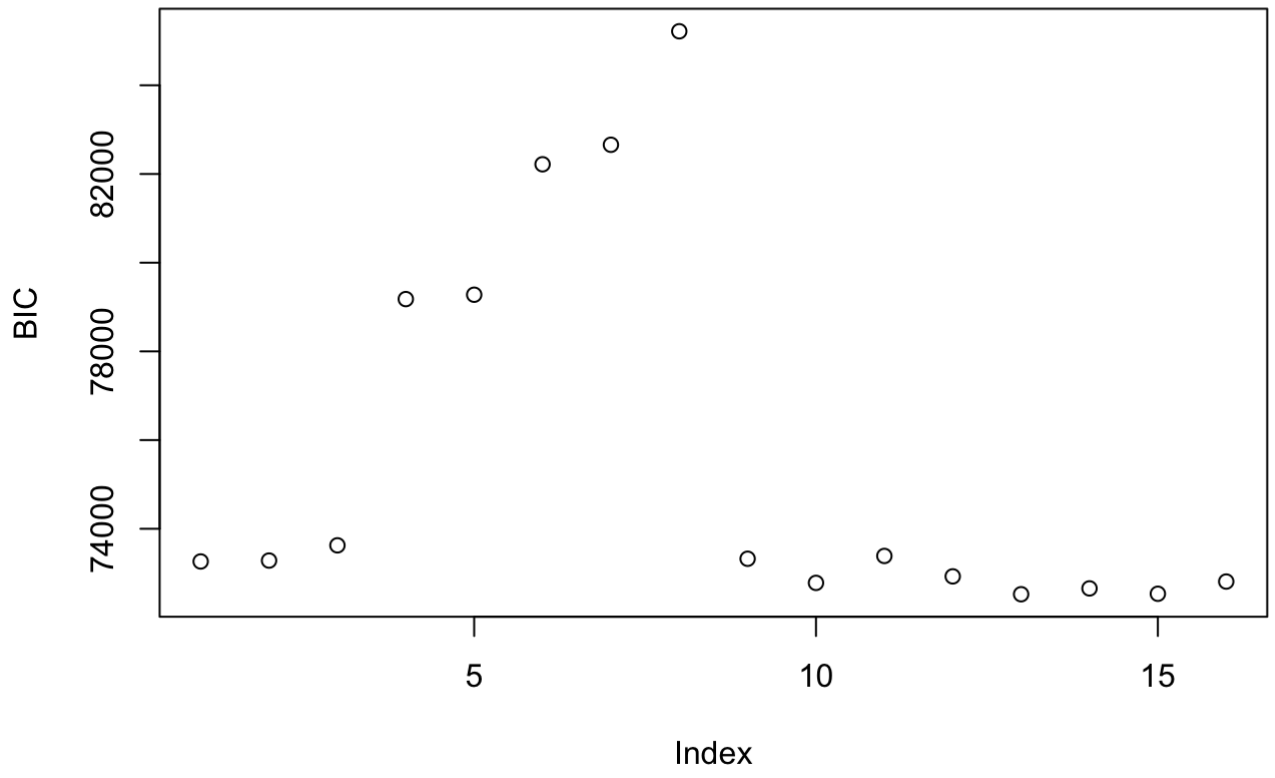
```
# omitted variable test and model choice based on BIC
logres_0 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + T
IME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION, data = training.data, family = binomial)
logres_1 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + T
IME_OF_DAY +
                DISTANCEGROUP, data = training.data, family = binomial)
logres_2 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + T
IME_OF_DAY,
                data = training.data, family = binomial)
logres_3 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK, da
ta = training.data,
                family = binomial)
logres_4 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE, data = training.
data, family=binomial)
logres_5 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION, data = training.data, family =
 binomial)
logres_6 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE, data = training.data, family = binomial)

logres_7 <- glm(DELAY_DUMMY ~ SEASON, data = training.data, family = binomial)
logres_8 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + T
IME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON, data = training.data, fa
mily = binomial)
logres_9 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + T
IME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE, d
ata = training.data, family = binomial)
logres_10 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK +
 TIME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK, data = training.data, family = binomial)
logres_11 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK +
 TIME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY, data = training.data, family = binomial)
logres_12 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK +
 TIME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY + AIRLINE*DISTANCEGROUP, data = training.dat
a, family = binomial)
logres_13 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK +
 TIME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY + AIRLINE*DISTANCEGROUP + AIRPORT_SIZE*DISTAN
CEGROUP, data = training.data, family = binomial)
logres_14 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK +
 TIME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY + AIRLINE*DISTANCEGROUP + AIRPORT_SIZE*REGION
 + AIRPORT_SIZE*DISTANCEGROUP, data = training.data, family = binomial)
logres_15 <- glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK +
 TIME_OF_DAY +
```

```
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY + AIRLINE*DISTANCEGROUP  + AIRPORT_SIZE*REGIO
N + AIRPORT_SIZE*DISTANCEGROUP + AIRPORT_SIZE*DAY_OF_WEEK, data = training.data, family
 = binomial)

BIC_1 <- BIC(logres_1); BIC_2 <- BIC(logres_2); BIC_3 <- BIC(logres_3); BIC_4 <- BIC(log
res_4)
BIC_5 <- BIC(logres_5); BIC_6 <- BIC(logres_6); BIC_7 <- BIC(logres_7); BIC_0 <- BIC(log
res_0)
BIC_8 <- BIC(logres_8); BIC_9 <- BIC(logres_9); BIC_10 <- BIC(logres_10); BIC_11 <- BIC
(logres_11);BIC_12 <- BIC(logres_12)
BIC_13 <- BIC(logres_13); BIC_14 <- BIC(logres_14); BIC_15 <- BIC(logres_15)

# choose logre_1 with min BIC
BIC <- rbind(BIC_0,BIC_1 ,BIC_2,BIC_3 ,BIC_4,BIC_5,BIC_6,BIC_7,BIC_8,BIC_9 ,BIC_10,BIC_1
1 ,BIC_12,BIC_13,BIC_14,BIC_15)
plot(BIC) # logre_12 is the lowest BIC
```




```
## Model discrimination ability (AUC = 0.797)
tst_pred <- ifelse(predict(logres_12, newdata = validation.data, type = "response") > 0.
5, "Yes", "No")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

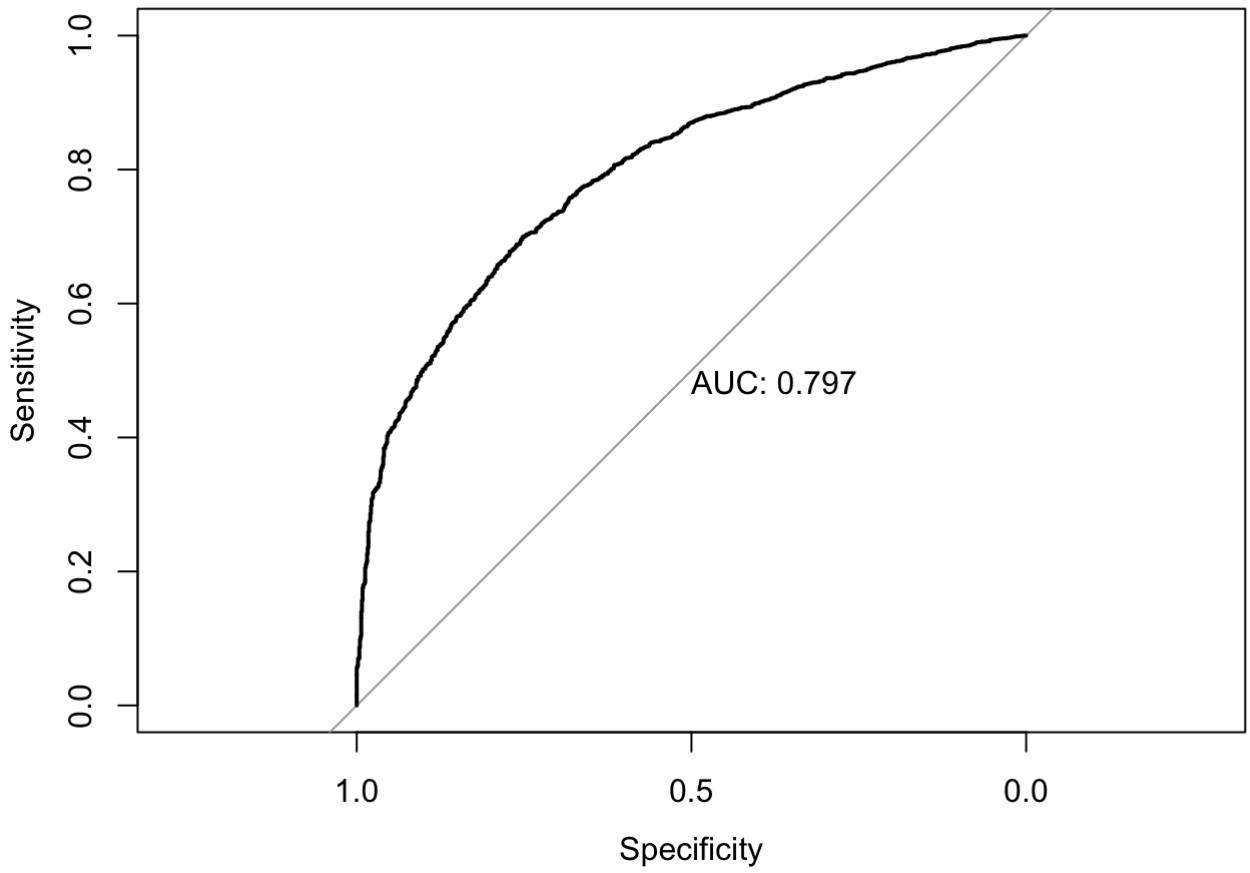
```
tst_tab <- table(predicted = tst_pred, actual = validation.data$DELAY_DUMMY)
tst_tab
```

```
##          actual
## predicted    0    1
##       No   312  226
##       Yes  722 3194
```

```
test_prob <- predict(logres_12, newdata = validation.data, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
test_roc <- roc(validation.data$DELAY_DUMMY ~ test_prob, plot = TRUE, print.auc = TRUE)
```




```
## Predicted probability with thresh at 0.5;
Yfac <- factor(training.data$DELAY_DUMMY, labels=c("lo", "hi"))
Yhat <- fitted(logres_12);head(Yhat)
```

```
##         1         2         3         4         5         6
## 0.5268618 0.5252054 0.2419790 0.6096264 0.6080442 0.4758051
```

```
thresh <- 0.5  # threshold for dichotomizing according to predicted probability
YhatFac <- cut(Yhat, breaks=c(-Inf, thresh, Inf), labels=c("lo", "hi"))
cTab <- table(Yfac, YhatFac)    # contingency table
addmargins(cTab) # marginal sums
```

```
##      YhatFac
## Yfac     lo    hi   Sum
##   lo   5798 12564 18362
##   hi   3685 58125 61810
##   Sum  9483 70689 80172
```

```
sum(diag(cTab)) / sum(cTab)  # percentage correct for training data   # 0.7973233
```

```
## [1] 0.7973233
```

```
## Confint of odds for each coefficient
anova(logres_12)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DELAY_DUMMY
##
## Terms added sequentially (first to last)
##
##
##                       Df Deviance Resid. Df Resid. Dev
## NULL                                  80171      86282
## SEASON                 3   1109.6     80168      85172
## AIRLINE               13   2705.7     80155      82467
## REGION                 4    485.0     80151      81982
## AIRPORT_SIZE           5   2997.5     80146      78984
## DAY_OF_WEEK            6    166.2     80140      78818
## TIME_OF_DAY            5   5609.2     80135      73209
## DISTANCEGROUP          5    399.5     80130      72809
## SEASON:REGION         12    156.0     80118      72653
## SEASON:AIRLINE        38    367.9     80080      72285
## AIRLINE:AIRPORT_SIZE  62   1244.1     80018      71041
## AIRLINE:DAY_OF_WEEK   78    273.4     79940      70768
## AIRLINE:TIME_OF_DAY   65   1194.6     79875      69573
## AIRLINE:DISTANCEGROUP 54   1012.4     79821      68561
```

## (4) RESULTS AND INTEPRETATION

After some experimentation, we picked model `logres_12` due to its low BIC value. The model is as follows:

DELAY_DUMMY = SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY + DISTANCEGROUP + SEASON x REGION + AIRLINE x SEASON + AIRLINE x AIRPORT_SIZE + AIRLINE x DAY_OF_WEEK + AIRLINE x TIME_OF_DAY + AIRLINE x DISTANCEGROUP

According to the seasonal effect estimates, we found that delay increases during summer and winter. Specifically, the odds rate of delay in winter and summer are 2.04 and 1.70 higher when compared to the fall. In summer, airports in the West had significantly much more delays. The odds rate increased by 9.62 compared with that of midwest region. In winter, northeast region was more likely to have delays. The odds rate increased to 1.23 compared with the Midwest(as the base line) while other regions was less likely to have delays than Midwest.

The model also showed weekly patterns and timely patterns. On Thursday, Friday, the average delay showed higher value. The odds rate was 1.02 for Thursday and 1.0 for Friday. Wednesday and Tuesday had relatively lower odds rate. In the evening and afternoon, the proportion of delay were higher than usual while the morning had significantly lower delays. We took odds rate in afternoon as the base line. The odds rate in the evening was 0.99 and odds rate in the morning was 0.53.

Delay was positively related to airport size. Large national airport had significantly higher delay probability than others. Comparing with basic airport, the odds rate was 4.13, bigger than 1.17 for medium national and 1.21 for small national airport. As for the distance factor, the delay probability showed a “U” shape distribution. Flights with distance less than 750 miles and more than 1250 seems to have more delays.

# 5.3 LINEAR REGRESSION

### (1) METHODOLOGY

After predicting the probability of flights delay, we were curious about how severe the delay would be because it closely related to what recommendations we would like to give FAA. Therefore, we built a linear regression model of average delay time for flights with same region, airline, season, day of week, time of day and distance group. Through those models, we could find the top factors affecting delay time and judge if the delays are economically significant.

To run our linear regression, we started with the same dataset as the logistic regression above but we filtered to where delay time is positive. This left about 62, 000 observations. We started out with a baseline regression of delay time (dependent variable), airline and season as explanatory variables. We added one variable at a time until we had all the variables that we thought would affect delay. We then tested all the models against out validation set.

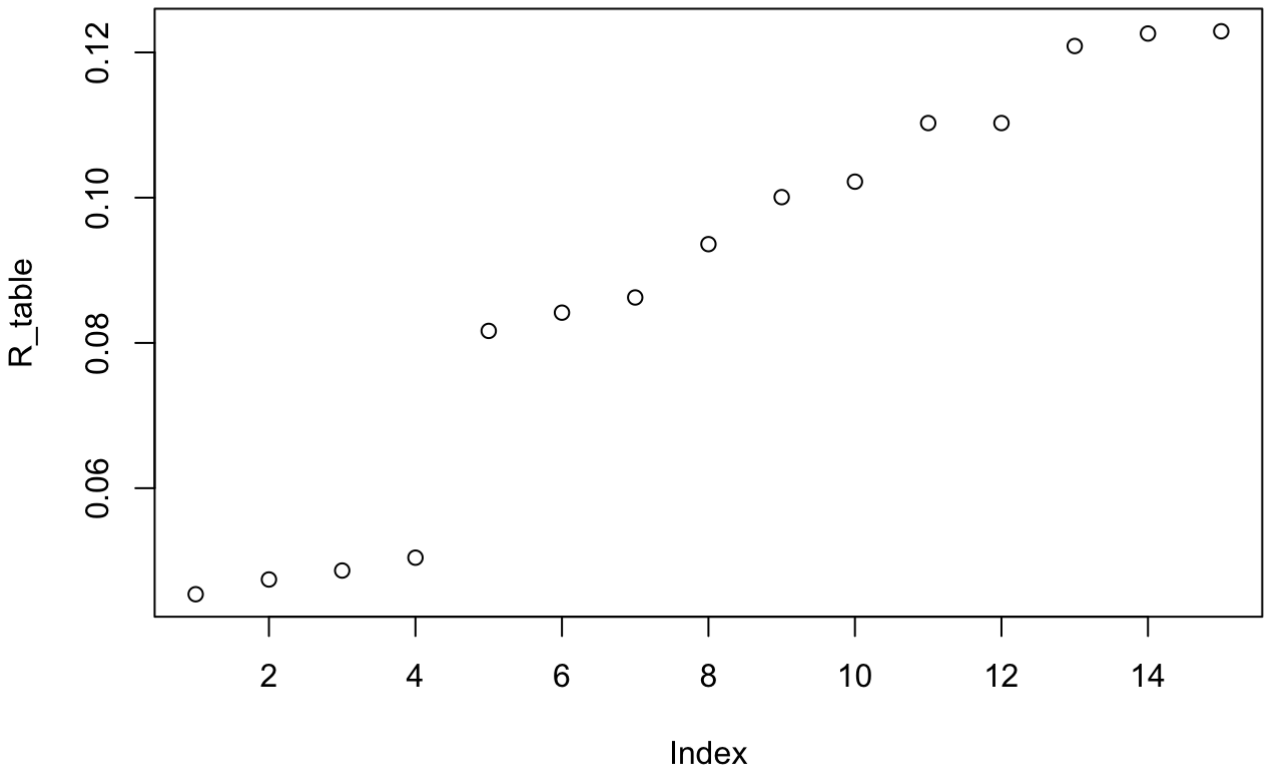
The model(LM_14) with delay time as dependent variable and season, airline, region, airport size, day of week, distance group, time of day, along with the interaction of season and region, airline and time of day, airline and airport size, airline and day of week, airline and time of day, airline and distance group, airport size and distance group had the a relatively low prediction error and high adjusted R-squared. The prediction error was 16.97 and R-squared is 12.26%. Note that the interaction term of airport size and region was not significant. Airline and time of day were relatively the most important factors.

### (2) LINEAR REGRESSION

```
## Divide dataset into training and validation data (80-20)
training.data2 <- training.data %>% filter(DELAY > 0)
validation.data2 <- validation.data %>% filter(DELAY > 0)

# Run regression step by step to find the best model based on adjusted R-squared
LM_15 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON+ AIRLINE*AIRPORT_SIZE + A
IRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY + AIRLINE*DISTANCEGROUP  + AIRPORT_SIZE*REGION
 + AIRPORT_SIZE*DISTANCEGROUP + AIRPORT_SIZE*DAY_OF_WEEK, data = training.data2)
LM_14 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON+ AIRLINE*AIRPORT_SIZE + A
IRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY + AIRLINE*DISTANCEGROUP  + AIRPORT_SIZE*REGION
 + AIRPORT_SIZE*DISTANCEGROUP , data = training.data2)
LM_13 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON+ AIRLINE*AIRPORT_SIZE + A
IRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY + AIRLINE*DISTANCEGROUP  + AIRPORT_SIZE*REGION,
 data = training.data2)
LM_12 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY, data = training.data2)
LM_11 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY, data = training.data2)
LM_10 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK, data = training.data2)
LM_9 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON+ AIRLINE*AIRPORT_SIZE  ,
 data = training.data2)
LM_8 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 + DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON,
         data = training.data2)
LM_7 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 + DISTANCEGROUP + SEASON*REGION,
        data = training.data2) # 0.08121
LM_6 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY
 + DISTANCEGROUP, data = training.data2)
# time_of_day is an important factor (LM below)
LM_5 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DAY,
 data = training.data2)
LM_4 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK, data = traini
ng.data2)
LM_3 <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE, data = training.data2)
LM_2 <- lm(DELAY ~ SEASON + AIRLINE + REGION, data = training.data2)
LM_1 <- lm(DELAY ~ SEASON + AIRLINE, data = training.data2)
```

```
# Compare the R-squared of models
R1 <- summary(LM_1)$adj.r.squared
R2 <- summary(LM_2)$adj.r.squared
R3 <- summary(LM_3)$adj.r.squared
R4 <- summary(LM_4)$adj.r.squared
R5 <- summary(LM_5)$adj.r.squared
R6 <- summary(LM_6)$adj.r.squared
R7 <- summary(LM_7)$adj.r.squared
R8 <- summary(LM_8)$adj.r.squared
R9 <- summary(LM_9)$adj.r.squared
R10 <- summary(LM_10)$adj.r.squared
R11 <- summary(LM_11)$adj.r.squared
R12<- summary(LM_12)$adj.r.squared
R13<- summary(LM_13)$adj.r.squared
R14 <- summary(LM_14)$adj.r.squared
R15 <- summary(LM_15)$adj.r.squared
R_table <- rbind(R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15)
plot(R_table)
```



```
########### Prediction error comparition and choose regression with REGION-SEASON intera
ction term
error_model1 = prediction.error(LM_1, validation.data2)
error_model2 = prediction.error(LM_2, validation.data2)
error_model3 = prediction.error(LM_3, validation.data2)
error_model4 = prediction.error(LM_4, validation.data2)
error_model5 = prediction.error(LM_5, validation.data2)
error_model6 = prediction.error(LM_6, validation.data2)
error_model7 = prediction.error(LM_7, validation.data2)
error_model8 = prediction.error(LM_8, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
error_model9 = prediction.error(LM_9, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
error_model10 = prediction.error(LM_10, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
error_model11 = prediction.error(LM_11, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
error_model12 = prediction.error(LM_12, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
error_model13 = prediction.error(LM_13, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

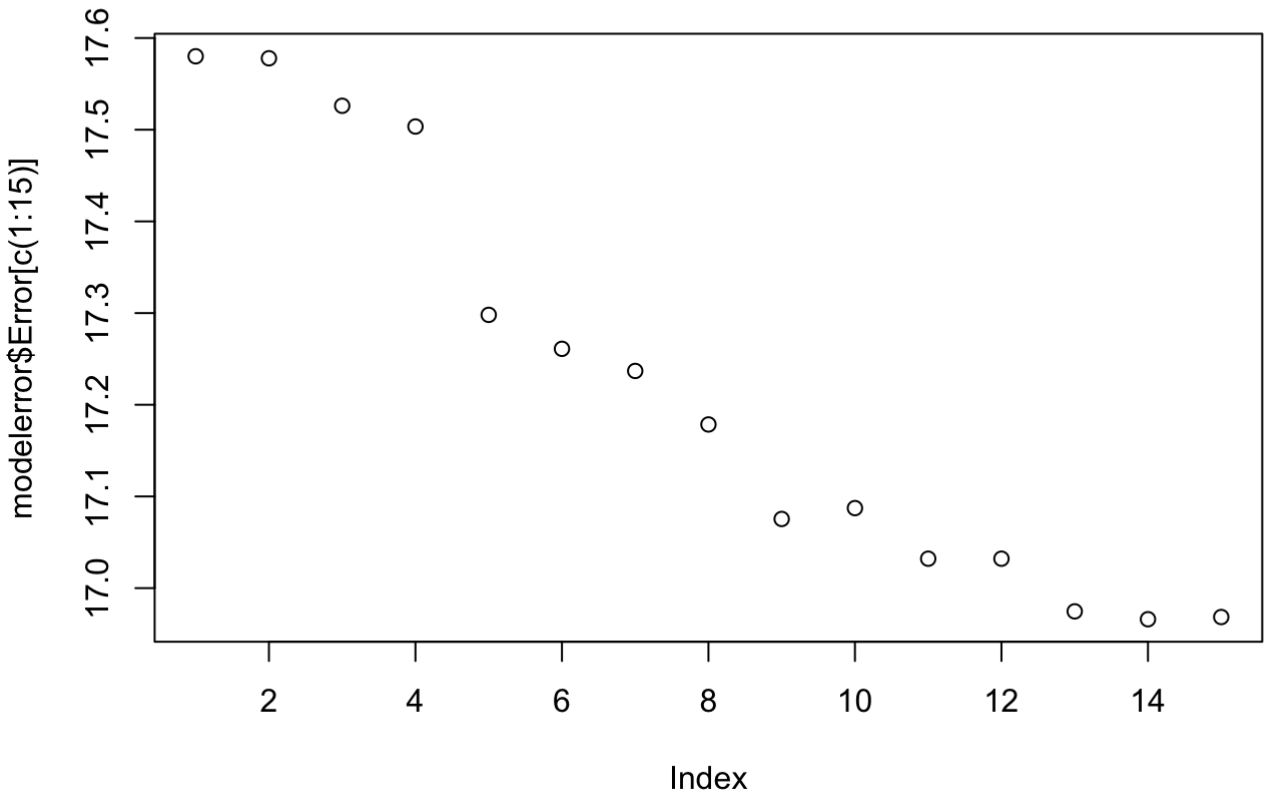
```
error_model14 = prediction.error(LM_14, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
error_model15 = prediction.error(LM_15, validation.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
modelerror <- data.frame(Names = c("SEASON_AIRLINE","RESION", "DAT_OF_WEEK", "AIRPORT_SI
ZE", "TIME_OF_DAY",
                                   "DISTANCEGROUP","LOGDELAY","POLYDELAY","REGION-SEASO
N","RE-SE_LOG","11","12","13","14","15"),
                           Error = c(error_model1,error_model2, error_model3,
                                     error_model4, error_model5, error_model6,
                                     error_model7, error_model8, error_model9,
                                     error_model10,error_model11,error_model12,
                                     error_model13,error_model14,error_model15))
# Plot model error table (delete model 8 because the error is too large)
plot(modelerror$Error[c(1:15)])
```




```
# model we choose
anova(LM_14)
```

```
## Analysis of Variance Table
##
## Response: DELAY
##                              Df   Sum Sq Mean Sq  F value    Pr(>F)
## SEASON                        3   266743   88914 325.3272 < 2.2e-16 ***
## AIRLINE                      13   612070   47082 172.2686 < 2.2e-16 ***
## REGION                        4    40256   10064  36.8230 < 2.2e-16 ***
## AIRPORT_SIZE                  5    25201    5040  18.4416 < 2.2e-16 ***
## DAY_OF_WEEK                   6    35847    5975  21.8602 < 2.2e-16 ***
## TIME_OF_DAY                   5   602310  120462 440.7561 < 2.2e-16 ***
## DISTANCEGROUP                 5    49773    9955  36.4229 < 2.2e-16 ***
## SEASON:REGION                12    43627    3636  13.3022 < 2.2e-16 ***
## SEASON:AIRLINE               38   151967    3999  14.6324 < 2.2e-16 ***
## AIRLINE:AIRPORT_SIZE         62   141831    2288   8.3701 < 2.2e-16 ***
## AIRLINE:DAY_OF_WEEK          78    62730     804   2.9426 < 2.2e-16 ***
## AIRLINE:TIME_OF_DAY          65   172819    2659   9.7281 < 2.2e-16 ***
## AIRLINE:DISTANCEGROUP        54   192263    3560  13.0272 < 2.2e-16 ***
## REGION:AIRPORT_SIZE          16    30098    1881   6.8828  4.41e-16 ***
## AIRPORT_SIZE:DISTANCEGROUP   25    39777    1591   5.8216 < 2.2e-16 ***
## Residuals                 61418 16785999     273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**(3) RESULTS AND INTEPRETATION**

Delay time was significantly higher in winter, with an average of 4min longer than that in fall. What surprised us was that delay time in summer was shorter than that in fall, implying that although more delays in summer, the delay was not so bad. Considering that, flights in summer may have more small delays, we'll change the delay standard to 10 or 15 minutes in the future to robust our regression.

As for airlines, Frontier Airlines and Hawaiian Airlines had significantly longer delay time while AA, Eagle Airlines, Jetwat Airways, United Air Lines also had relatively longer delays. This was interesting as we found that Frontier Airlines and Hawaiian Airlines were less likely to having delays, however, they're more likely to having bad delays. Frontier had 330 minutes longer delays compared with Alaska Airlines while Hawaiian had 26 minutes longer. Delta Airlines, Spirit Airlines had shorter delay time in average, which were 2minutes and 2.7 minutes less than that of Alaska Airlines. Note that they also were less likely to having delays.

Airlines performed differently in operations. Frontier had significantly longer delays in winter than other airlines and Spirit had longer delays in summer. This might be caused by weather and also management ability of different airlines to the change of weather.

South, Northeast and west had longer delays than midwest. The gap was around 5 minutes. Northeast had 8 minutes longer delays in winter and spring than in fall, implying the snowy days might cause longer delays. For south, delays were longer in spring and summer.

Friday had the longest delays in average. Monday, Sunday and Thursday were followed. Flights in night had the longest delay time compared with that in afternoon while morning and early morning had less delays.

Besides, Longer flights tended to have longer delays. Large national airports tended to have flights with longer distance, affecting the delay time.

# 6 SEGMENTATION

## 6.1 SEGMENTATION ANALYSIS

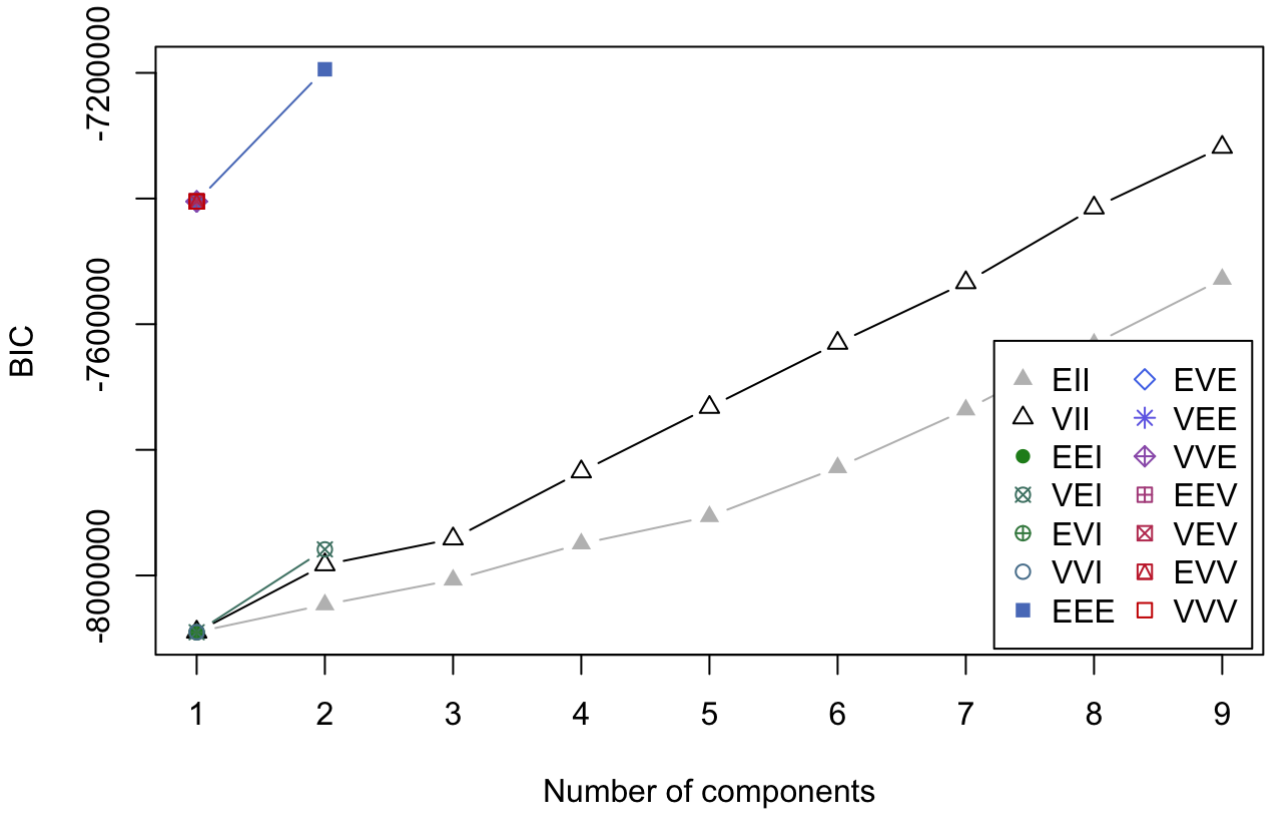
### METHODOLOGY

Our goal in this section was to conduct post-hoc segmentation to determine if there were underlying characteristics we had not considered in our regression analyses. To segment our data, we used the same aggregated data from the regression section above but we scaled it to ensure that our clustering method performed well. We then ran a simulation to determine the number of clusters with the best BIC and used the mclust method to cluster our data into groups.

```
# make dummies
seg.flg.num <- model.matrix(~DELAY + TAXI_OUT + SEASON + AIRPORT_SIZE + REGION + AIRLINE
 + TIME_OF_DAY, data = agg)
seg.flg.num <- seg.flg.num [,-1]

# scaling the data
scaled_data <- scale(seg.flg.num)
scaled_data <- as.data.frame(scaled_data) %>% dplyr::select(-SEASONMissing, -TIME_OF_DAY
Missing)

# plot BIC
BIC <- mclustBIC(scaled_data)
plot(BIC)
```




```
# clustering
set.seed(123456789); cl = Mclust(scaled_data); summary(cl, parameters = F)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 2
## components:
##
##  log.likelihood     n  df      BIC      ICL
##       -3593779 89081 593 -7194316 -7194316
##
## Clustering table:
##     1     2
## 36433 52648
```

```
#Creat cluster variable
agg$cluster = cl$classification

propREG <- as.array(table(agg$cluster, agg$REGION)); prop.table(propREG, margin = 1)
```

```
##
##        Midwest  Northeast Over Seas Territory      South       West
##   1 0.18944364 0.20088930          0.06266297 0.32687399 0.22013010
##   2 0.20847895 0.16139265          0.00000000 0.29171099 0.33841741
```

```
propSEA <- as.array(table(agg$cluster, agg$SEASON)); prop.table(propSEA, margin = 1)
```

```
##
##        Spring    Summer      Fall    Winter   Missing
##   1 0.2635248 0.2486757 0.2166168 0.2711827 0.0000000
##   2 0.2366282 0.2539698 0.2526782 0.2567239 0.0000000
```

```
propSIZE <- as.array(table(agg$cluster, agg$AIRPORT_SIZE)); prop.table(propSIZE, margin
= 1)
```

```
##
##          Basic      Local   Regional Small National Medium National
##   1 0.03883842 0.15236187 0.20602201     0.10926907      0.13712843
##   2 0.02155827 0.11141924 0.20878286     0.16312111      0.17079471
##
##     Large National
##   1     0.35638020
##   2     0.32432381
```

```
propTIME <- as.array(table(agg$cluster, agg$TIME_OF_DAY)); prop.table(propTIME, margin =
 1)
```

```
##
##     Early morning    Morning     Midday  Afternoon    Evening      Night
##   1    0.17797052 0.17983696 0.17788818 0.18181319 0.19372547 0.08876568
##   2    0.19147926 0.17905713 0.18097554 0.17953199 0.17985489 0.08910120
##
##        Missing
##   1 0.00000000
##   2 0.00000000
```

```
propDAY <- as.array(table(agg$cluster, agg$DAY_OF_WEEK)); prop.table(propDAY, margin = 1
)
```

```
##
##        Monday   Tuesday Wednesday  Thursday    Friday  Saturday    Sunday
##   1 0.1441001 0.1421513 0.1420965 0.1432218 0.1431120 0.1407790 0.1445393
##   2 0.1427974 0.1420757 0.1421327 0.1428734 0.1428354 0.1412399 0.1460454
##
##       Missing
##   1 0.0000000
##   2 0.0000000
```

```
propAIRLINE <- as.array(table(agg$cluster, agg$AIRLINE)); prop.table(propAIRLINE, margin
= 1)
```

```
##
##     Alaska Airlines Inc. American Airlines Inc.
##   1          0.000000000            0.013367002
##   2          0.076698070            0.192144051
##
##     American Eagle Airlines Inc. Atlantic Southeast Airlines
##   1                  0.137402904                 0.182965992
##   2                  0.000000000                 0.000000000
##
##     Delta Air Lines Inc. Frontier Airlines Inc. Hawaiian Airlines Inc.
##   1          0.006285510            0.110614004            0.024483298
##   2          0.186084941            0.000000000            0.000000000
##
##     JetBlue Airways Skywest Airlines Inc. Southwest Airlines Co.
##   1     0.178931189           0.000000000            0.006944254
##   2     0.000000000           0.145570582            0.222078711
##
##     Spirit Air Lines United Air Lines Inc. US Airways Inc. Virgin America
##   1      0.127658990           0.008453874     0.141163231    0.061729751
##   2      0.000000000           0.177423644     0.000000000    0.000000000
```

** (2) RESULTS AND INTEPRETATION**

We managed to find 2 clusters of our aggregated flights set. The most differentiating aspect between the groups was airline size, with the biggest airlines by the number of flights falling into one group while the rest of airlines are in the other one. Alaska Airlines was the exception because it is not among the biggest airlines, but it was grouped together with them. We then added this the new segment groups into our regressions and found that it does not improve the performance of our models significantly.

# 6.2 REGRESSION WITH SEGMENTATION

### (1) METHODOLOGY

In this section, we included the segmentation group variable (“cluster”) into our regression to see if it improved the predictive ability of the model. We selected the best model of logistic and linear regression in sections 5.2 and 5.3 and added the group variable to the regression.

```
# Divide dataset into training and validation(80-20)
training_clu.data = subset(agg,randOrder < .9 * nrow(agg))
validation_clu.data =  subset(agg,randOrder >= .9*nrow(agg)&(randOrder <= .95*nrow(ag
g)))
test_clu.data = subset(agg,randOrder > .95 * nrow(agg))

# Take the best model that we used before
logres_clu1 = glm(DELAY_DUMMY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK +
 TIME_OF_DAY +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON + AIRLINE*AIRPORT_SIZE +
 AIRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY
                + AIRLINE*DISTANCEGROUP  + AIRPORT_SIZE*REGION + AIRPORT_SIZE*DISTANCEGR
OUP + AIRPORT_SIZE*DAY_OF_WEEK,
                data = training_clu.data, family = binomial)
anova(logres_clu1)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DELAY_DUMMY
##
## Terms added sequentially (first to last)
##
##
##                            Df Deviance Resid. Df Resid. Dev
## NULL                                       80171      86282
## SEASON                      3   1109.6     80168      85172
## AIRLINE                    13   2705.7     80155      82467
## REGION                      4    485.0     80151      81982
## AIRPORT_SIZE                5   2997.5     80146      78984
## DAY_OF_WEEK                 6    166.2     80140      78818
## TIME_OF_DAY                 5   5609.2     80135      73209
## DISTANCEGROUP               5    399.5     80130      72809
## SEASON:REGION              12    156.0     80118      72653
## SEASON:AIRLINE             38    367.9     80080      72285
## AIRLINE:AIRPORT_SIZE       62   1244.1     80018      71041
## AIRLINE:DAY_OF_WEEK        78    273.4     79940      70768
## AIRLINE:TIME_OF_DAY        65   1194.6     79875      69573
## AIRLINE:DISTANCEGROUP      54   1012.4     79821      68561
## REGION:AIRPORT_SIZE        16    313.4     79805      68247
## AIRPORT_SIZE:DISTANCEGROUP 25    138.8     79780      68108
## AIRPORT_SIZE:DAY_OF_WEEK   30     63.6     79750      68045
```

```
BIC(logres_clu1) #73296.52> 72524 model before
```

```
## [1] 72810.08
```

```
# Discrimination ability test
tst_pred2 <- ifelse(predict(logres_clu1, newdata = validation_clu.data, type = "respons
e") > 0.5, "Yes", "No")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

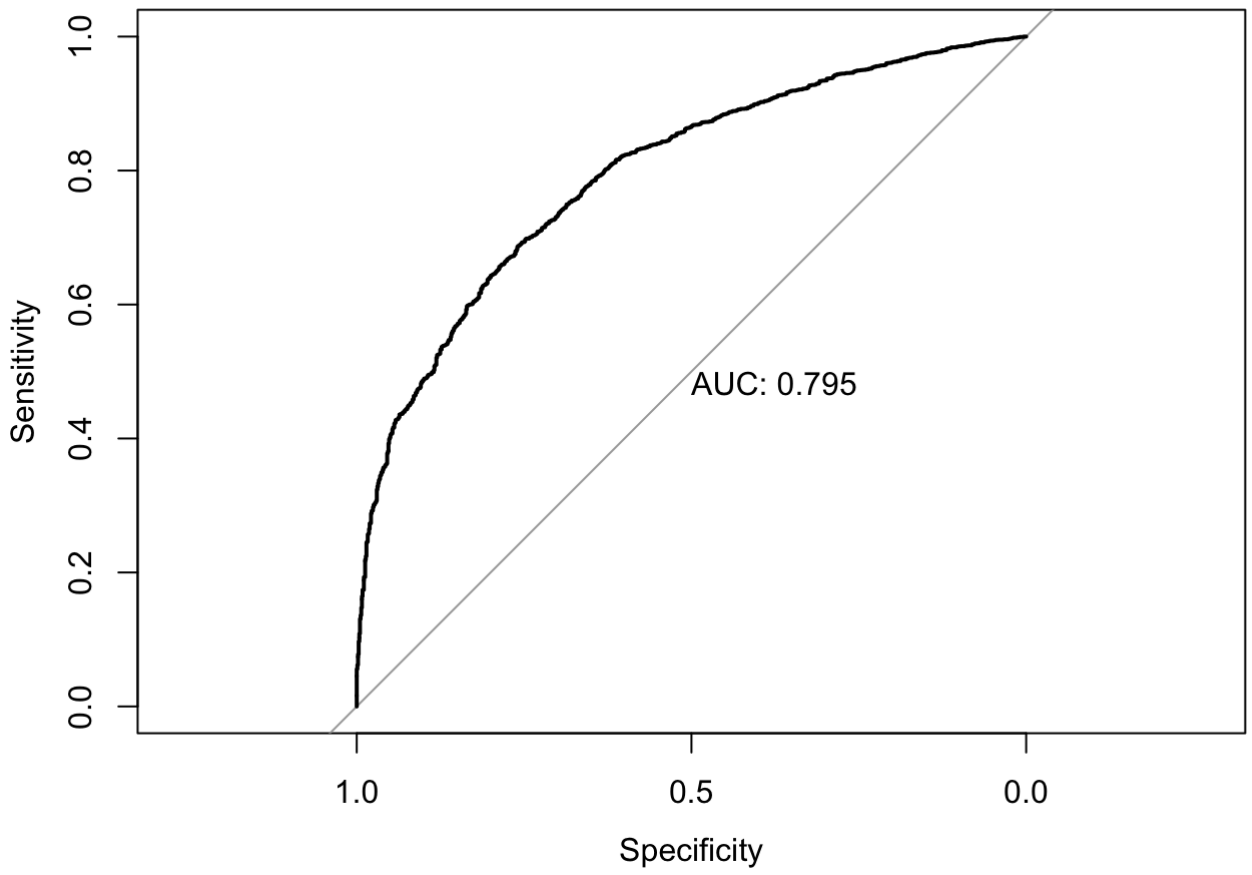
```
tst_tab2 <- table(predicted = tst_pred2, actual = validation_clu.data$DELAY_DUMMY); tst_
tab2
```

```
##          actual
## predicted    0    1
##       No   315  226
##       Yes  719 3194
```

```
test_prob2 <- predict(logres_clu1, newdata = validation_clu.data, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
test_roc2 <- roc(validation_clu.data$DELAY_DUMMY ~ test_prob2, plot = TRUE, print.auc =
TRUE) # 0.772<0.797
```




```
# predicted probabilities test
Yfac2 <- factor(training_clu.data$DELAY_DUMMY, labels=c("lo", "hi"))
Yhat2 <- fitted(logres_clu1)
thresh <- 0.5  # threshold for dichotomizing according to predicted probability
YhatFac2 <- cut(Yhat2, breaks=c(-Inf, thresh, Inf), labels=c("lo", "hi"))
cTab <- table(Yfac2, YhatFac2)    # contingency table
addmargins(cTab)  # marginal sums
```

```
##      YhatFac2
## Yfac2    lo    hi   Sum
##   lo   5941 12421 18362
##   hi   3758 58052 61810
##   Sum  9699 70473 80172
```

```
sum(diag(cTab)) / sum(cTab)  # percentage correct for training data 0.7836028<0.797
```

```
## [1] 0.7981964
```

```
# DELAY TIME
training_clu.data2 <- training_clu.data %>% filter(DELAY > 0)
training_clu.data2 <- training_clu.data2 %>% mutate(log_DELAY = log(DELAY),poly_DELAY =
 DELAY*DELAY)
validation_clu.data2 <- validation_clu.data %>% filter(DELAY > 0)

LM_clu <- lm(DELAY ~ SEASON + AIRLINE + REGION + AIRPORT_SIZE + DAY_OF_WEEK + TIME_OF_DA
Y +
                DISTANCEGROUP + SEASON*REGION + AIRLINE*SEASON+ AIRLINE*AIRPORT_SIZE + A
IRLINE*DAY_OF_WEEK + AIRLINE*TIME_OF_DAY
             + AIRLINE*DISTANCEGROUP  + AIRPORT_SIZE*REGION + AIRPORT_SIZE*DISTANCEGROU
P,data = training_clu.data2) # 0.2802  Interaction
anova(LM_clu)
```

```
## Analysis of Variance Table
##
## Response: DELAY
##                               Df   Sum Sq Mean Sq  F value    Pr(>F)
## SEASON                         3   266743   88914 325.3272 < 2.2e-16 ***
## AIRLINE                       13   612070   47082 172.2686 < 2.2e-16 ***
## REGION                         4    40256   10064  36.8230 < 2.2e-16 ***
## AIRPORT_SIZE                   5    25201    5040  18.4416 < 2.2e-16 ***
## DAY_OF_WEEK                    6    35847    5975  21.8602 < 2.2e-16 ***
## TIME_OF_DAY                    5   602310  120462 440.7561 < 2.2e-16 ***
## DISTANCEGROUP                  5    49773    9955  36.4229 < 2.2e-16 ***
## SEASON:REGION                 12    43627    3636  13.3022 < 2.2e-16 ***
## SEASON:AIRLINE                38   151967    3999  14.6324 < 2.2e-16 ***
## AIRLINE:AIRPORT_SIZE          62   141831    2288   8.3701 < 2.2e-16 ***
## AIRLINE:DAY_OF_WEEK           78    62730     804   2.9426 < 2.2e-16 ***
## AIRLINE:TIME_OF_DAY           65   172819    2659   9.7281 < 2.2e-16 ***
## AIRLINE:DISTANCEGROUP         54   192263    3560  13.0272 < 2.2e-16 ***
## REGION:AIRPORT_SIZE           16    30098    1881   6.8828  4.41e-16 ***
## AIRPORT_SIZE:DISTANCEGROUP    25    39777    1591   5.8216 < 2.2e-16 ***
## Residuals                  61418 16785999     273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error_model_clu2 <- prediction.error(LM_clu, validation_clu.data2)
```

```
## Warning in predict.lm(lm_model, test.data): prediction from a rank-
## deficient fit may be misleading
```

```
error_model_clu2  # 13.45118 < 16.96
```

```
## [1] 16.96607
```

## (2) RESULTS AND INTEPRETATION

The predictive ability of the both models did not improve significantly after adding the new segment variable. For the logistic regression, the BIC increased while the hit rate decreased. For the linear regression, however, the prediction error decreased from 16.96 to 13.45. It might be because the segmentation was affected mainly by airline and we have already included airline and its interaction terms with other variables in the original regressions. We therefore conclude that the group variable did not add new information to the model.

# 7 PREDICTION

### (1) METHODOLOGY

As we mentioned above, we split the dataset into training, validation and test sets. We used test set to do the prediction analysis, which had 5% observations of the whole dataset after our aggregation. There're nearly 4500 observations in this data set.

The reason why we used this dataset as test data is that, we attempted to bring out some business insights through the prediction. For example, which airport or airline probably has more delays and longer delay time. As we did not get the data for flights in the future, we can use the test data to make a summary and give hints to future flights prediction. Basically, we predicted the delay probability and delay time with the best model in part 5. And then we made predictions based on airlines so that FAA can regulate according to airline operational ability.

### (2) PREDICTION WITH BEST MODEL FOR DELAY PROBABILITY

**Part 1** PREDICTION ON DELAY PROBABILITY

According to our theory, exogenous shocks affected the capacity of the airport, which causes delays. Therefore, FAA can make specific regulations on the airport management companies to rearrange the capacity of the airport at different times for different regions, etc. We considered season, region, day of week, time of day, season and region interaction terms as measures of those exogenous shocks. Through prediction, we found that generally more delays in summer and winter. Specifically, Northeast in winter, Midwest in winter, Midwest in summer have the highest probability of delays, which are 86.3, 83.5 and 82.3%. Also, south will have more delays throughout the whole year. Large national airport will have the highest delay probability according to the airport size. Monday, Sunday and Friday will have more delays in the future. Evening contributes more to delays, 86.8% delay probability. Flights with 1250-1749 miles are more likely to having delays.

**Part 2** PREDICTION ON DELAY TIME

After adjustment on the capacity of the airport, another important key is schedule of the flights of different airlines, which is related with the demand of the airport capacity. For example, more scheduled arrival flights at the same time at one airport will cause more delays. Therefore, FAA can regulate on the airlines or give them recommendations on how to schedule their flights according to the shocks we mentioned above. Here, we made predictions based on the interaction terms to see how different airlines will react in terms of the shocks.

Through the prediction, we found that in summer, Spirit Airlines, UA, Southwest Airlines will have more delays while Hawaiian, Virgin America will have less delays. In winter, Sprit Airlines, SA, UA still have more delays. However, Virgin America has delay probability higher than the average. For large national airport, UA, Skywest Airline, Spirit Airlines, Delta Airlines and AA will have more delays while for small airport, SA, UA, American Eagle Airlines, Sprit Airlines and AA will have the higher delay probability.

On Friday, UA, Spirit, SA will have more delay. In the evening, SA, JetBlue, UA, Virgin, Spirit will have higher delay probability while in the morning, Skywest, UA, SA tend to have more delays. For flights with 1250 to 1749 miles, which has the highest probability of delays, Spirit, UA, SA, Frontier have more delays. For the longest flights, JetBlue Airways, UA, Spirit Air Lines, SA have more delays. For the shortest flights with less than 250 miles, SA, UA, Skywest have more delays.

```
predicted.y = predict(logres_12, test.data, se.fit = TRUE,type='response')

test.data$predicted.y = predicted.y$fit
test.data$se.fit = predicted.y$se.fit
test.data$upper.limit = test.data$predicted.y + qnorm(0.975)*test.data$se.fit
## Upper limit of confidence interval for each predicted y
test.data$lower.limit = test.data$predicted.y - qnorm(0.975)*test.data$se.fit
## Lower limit of confidence interval for each predicted y
test.data$conf.int <- paste(test.data$lower.limit,",",test.data$upper.limit)

# PREDICTION
test1 <- test.data %>% group_by(AIRLINE) %>% summarize(avg_delay = mean(predicted.y))

test2 <- test.data %>% group_by(AIRPORT_SIZE) %>% summarize(avg_delay = mean(predicted.
y))

test3 <- test.data %>% group_by(TIME_OF_DAY) %>% summarize(avg_delay = mean(predicted.
y))

test4 <- test.data %>% group_by(SEASON) %>% summarize(avg_delay = mean(predicted.y))

test5 <- test.data %>% group_by(DAY_OF_WEEK) %>% summarize(avg_delay = mean(predicted.
y))

test6 <- test.data %>% group_by(REGION) %>% summarize(avg_delay = mean(predicted.y))

test7 <- test.data %>% group_by(DISTANCEGROUP) %>% summarize(avg_delay = mean(predicted.
y))

test8 <- test.data %>% group_by(REGION, SEASON) %>% summarize(avg_delay = mean(predicte
d.y))

test9 <- test.data %>% group_by(AIRLINE, SEASON) %>% summarize(avg_delay = mean(predicte
d.y))
test9D <- test9 %>% filter(SEASON == 'Summer')
test9D <- test9 %>% filter(SEASON == 'Winter')

test10 <- test.data %>% group_by(AIRLINE, AIRPORT_SIZE) %>% summarize(avg_delay = mean(p
redicted.y))
test10D <- test10 %>% filter(AIRPORT_SIZE == 'Large National')

test11 <- test.data %>% group_by(AIRLINE, DAY_OF_WEEK) %>% summarize(avg_delay = mean(pr
edicted.y))
test11D <- test11 %>% filter(DAY_OF_WEEK == 'Friday')

test12 <- test.data %>% group_by(AIRLINE, TIME_OF_DAY) %>% summarize(avg_delay = mean(pr
edicted.y))
test12D <- test12 %>% filter(TIME_OF_DAY == 'Evening')

test13 <- test.data %>% group_by(AIRLINE, DISTANCEGROUP) %>% summarize(avg_delay = mean
(predicted.y))
test13D <- test13 %>% filter(DISTANCEGROUP == '1250-1749 Miles')
```

## (3) PREDICTION WITH BEST MODEL FOR DELAY TIME

Considering that if a late departure aircraft has no empty space in its down line schedule, it will continue to be late. If that aircraft enters a connecting airport, it can pass its lateness on to other aircraft. Therefore, flights more likely to delay in certain conditions deserves us to do more research on so that we can reduce the effects of delay in total. As for this part, we only predict the delay time of airlines in situations with higher probability of delays.

Here's the conclusions that we found based on the interaction terms in the regression:

1. In winter, northeast, oversea, midwest have longer delays. In summer, northeast and south have longer delays.
2. In summer, spirit, UA, Frontier, Atlantic have longer delays while in winter, Frontier, JetBlue, Spirit will have longer delays.
3. For large national airport, Spirit, Frontier, UA, Hawaiian contributes to longer delays.
4. For Friday, Spirit, Frontier, American eagle airlines tend to have longer delays, which are around 16 minutes.
5. In the evening, Frontier, Spirit, UA, JetBlues will have longer average delays, which are 23, 22.4, 22.3, 20.3 minutes.
6. For short flights, JetBlue, UA, Spirit, Virgin have longer delays from 19 to 14 minutes. For 1250-1749 miles flights, American Eagle, Spirit, Frontier tend to have more delays. For long flights, Spirit, AA, UA, SA will have more delay.

```
test.data2 <- test.data %>% filter(predicted.y>0.5)
# LM
predicted.y2 = predict(LM_14, test.data2, se.fit = TRUE)

test.data2$predicted.y2 = predicted.y2$fit
test.data2$se.fit = predicted.y2$se.fit
test.data2$upper.limit = test.data2$predicted.y2 + qnorm(0.975)*test.data2$se.fit
## Upper limit of confidence interval for each predicted y
test.data2$lower.limit = test.data2$predicted.y2 - qnorm(0.975)*test.data2$se.fit
## Lower limit of confidence interval for each predicted y

test11 <- test.data2 %>% group_by(REGION, SEASON) %>% summarize(avg_delay = mean(predict
ed.y2))
test1S <- test11 %>% filter(SEASON == 'Summer')
test1S <- test11 %>% filter(SEASON == 'Winter')

## AIRLINE
test22 <- test.data2 %>% group_by(AIRLINE, AIRPORT_SIZE) %>% summarize(avg_delay = mean
(predicted.y2))
test2S <- test22 %>% filter(AIRPORT_SIZE == 'Large National')

test33 <- test.data2 %>% group_by(AIRLINE, DAY_OF_WEEK) %>% summarize(avg_delay = mean(p
redicted.y2))
test3S <- test33 %>% filter(DAY_OF_WEEK == 'Friday')

test44 <- test.data2 %>% group_by(AIRLINE, TIME_OF_DAY) %>% summarize(avg_delay = mean(p
redicted.y2))
test4S <- test44 %>% filter(TIME_OF_DAY == 'Evening')

test55 <- test.data2 %>% group_by(AIRLINE, DISTANCEGROUP) %>% summarize(avg_delay = mean
(predicted.y2))
test5S <- test55 %>% filter(DISTANCEGROUP == '1250-1749 Miles')

test66 <- test.data2 %>% group_by(AIRLINE, SEASON) %>% summarize(avg_delay = mean(predic
ted.y2))
test6S <- test66 %>% filter(SEASON == 'Winter')
test6S <- test66 %>% filter(SEASON == 'summer')
```

# CONCLUSION

It is apparent that the best choice of delay prediction depends on the specific regression model we chose. All time Variables and Airline seem to affect delay factor very significantly. Prediction of flights delay based on those factors can help FAA regulators anticipate delays and therefore reduce the economic losses of flights delays. The flight delay is directedly related with the time of the flights including season, the day of week and time of day. Time of day significantly affects both the probability of delay and time of delay. Specifically, flights departure in the evening usually has more delays. Also, flights in winter and summer, Monday and

Friday also have more delays. People who work in the services industries usually fly to the client sites at the beginning of each week and come back to its own workplace to report to the mangers. In addition, more delays in winter and summer due to the weather and vacations. FAA regulators should pay attention to keep the balance between the number of flights within a specific time period to meet the demands and the decreasing customer satisfaction due to the long delay caused by the larger number of flights. Also, management of flights can vary from different seasons. In winter, airlines need better operational effieciencies to limit the number of delayed flights. For example, airlines can proactively inspect planes before issues crop up. They can also assign more flight crews during winter times or whenever a storm is predicted.

The airline is another important factor to explain the flight delay. Surprisingly, large airlines tend to have more flights delays though we expected them to have better operations to limit disruptions. We highly recommended the FAA airline companies to monitor Airlines and ensure that protocals are regulations are complied with. For example, the FAA Could control the addition of flights in an airport to make sure that busy airports dont end up congested, especially in regions such as the North East during winter months.