Data Analyst Airline Challenge

Andres Izquierdo

2022-08-01

Data Set Up

```
# Data cleaning and set up.
library(dplyr)
```

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

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

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

```
library(ggplot2)
```

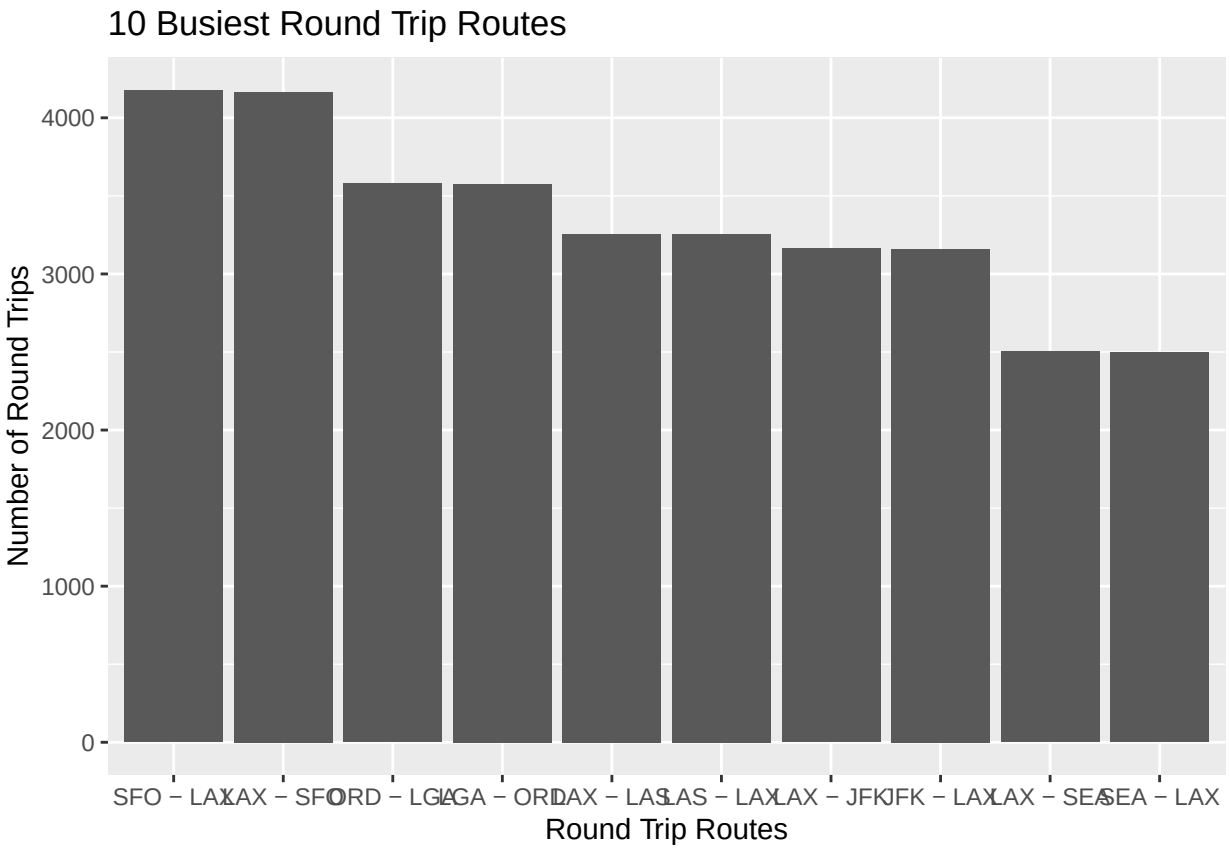
Data setup and clean up.

```
Airport_codes <- read.csv("C:/Users/andre/OneDrive/Documents/UVA SYS ME/Job Interview Resources/Capital
Flights <- read.csv("C:/Users/andre/OneDrive/Documents/UVA SYS ME/Job Interview Resources/Capital One/Fl
Tickets <- read.csv("C:/Users/andre/OneDrive/Documents/UVA SYS ME/Job Interview Resources/Capital One/Ti

# Filtering the Airports in the Dataframe to only show Medium and Large Airports in the US.
Airport_codes <- Airport_codes %>%
  filter(ISO_COUNTRY == "US") %>%
  filter(TYPE == "medium_airport"| TYPE == "large_airport")

# Renaming unique domestic airport columns and removing blank name
Dom_ac <- data.frame(unique(Airport_codes$IATA_CODE))
names(Dom_ac)[1] <- 'IATA_CODE'
Dom_ac <- Dom_ac[!(is.na(Dom_ac$IATA_CODE) | Dom_ac$IATA_CODE==""), ]

# Filtering out Canceled flights
```

```r
Flights <- Flights %>%
  filter(CANCELLED == "0")

# Creating Variable that combines the two variable ORIGIN and DESTINATION in Flights dataframe
Flights$ORIG_DEST <- paste(Flights$ORIGIN, "-", Flights$DESTINATION)
Flights_orde <- (unique(Flights$ORIG_DEST))

# Tickets
Tickets <- Tickets %>%
  filter(ROUNDTRIP == "1")

# Creating Variable that combines the two variable ORIGIN and DESTINATION in Tickets dataframe
Tickets$ORIG_DEST <- paste(Tickets$ORIGIN, "-", Tickets$DESTINATION)
Tickets_orde <- (unique(Tickets$ORIG_DEST))
Tickets_carr <- (unique(Tickets$REPORTING_CARRIER))
```

10 busiest round trip routes in terms of number of round trip flights in the quarter.

```r
# Keeping only the flights in dataframe Flights that are round trips in dataframe Tickets and operated
Flights <- filter(Flights, OP_CARRIER %in% Tickets_carr)
Flights <- filter(Flights, ORIG_DEST %in% Tickets_orde)

# Counting the number of round trip flights in the quarter
ORIG_DEST_COUNT <- Flights %>%
  group_by(ORIG_DEST) %>%
  count

# Renaming column
names(ORIG_DEST_COUNT)[2] <- 'ORIG_DEST_TOTAL'

# Sorting from greatest to least
ORIG_DEST_COUNT <- ORIG_DEST_COUNT[with(ORIG_DEST_COUNT, order(-ORIG_DEST_TOTAL)),]
ORIG_DEST_10 <- ORIG_DEST_COUNT[1:10,]

# 10 busiest round trip routes plot
ggplot(ORIG_DEST_10, aes(x = reorder(ORIG_DEST, -ORIG_DEST_TOTAL), y = ORIG_DEST_TOTAL)) + geom_bar(sta
```




```
# 10 busiest round trip routes table
print.data.frame(ORIG_DEST_10)
```

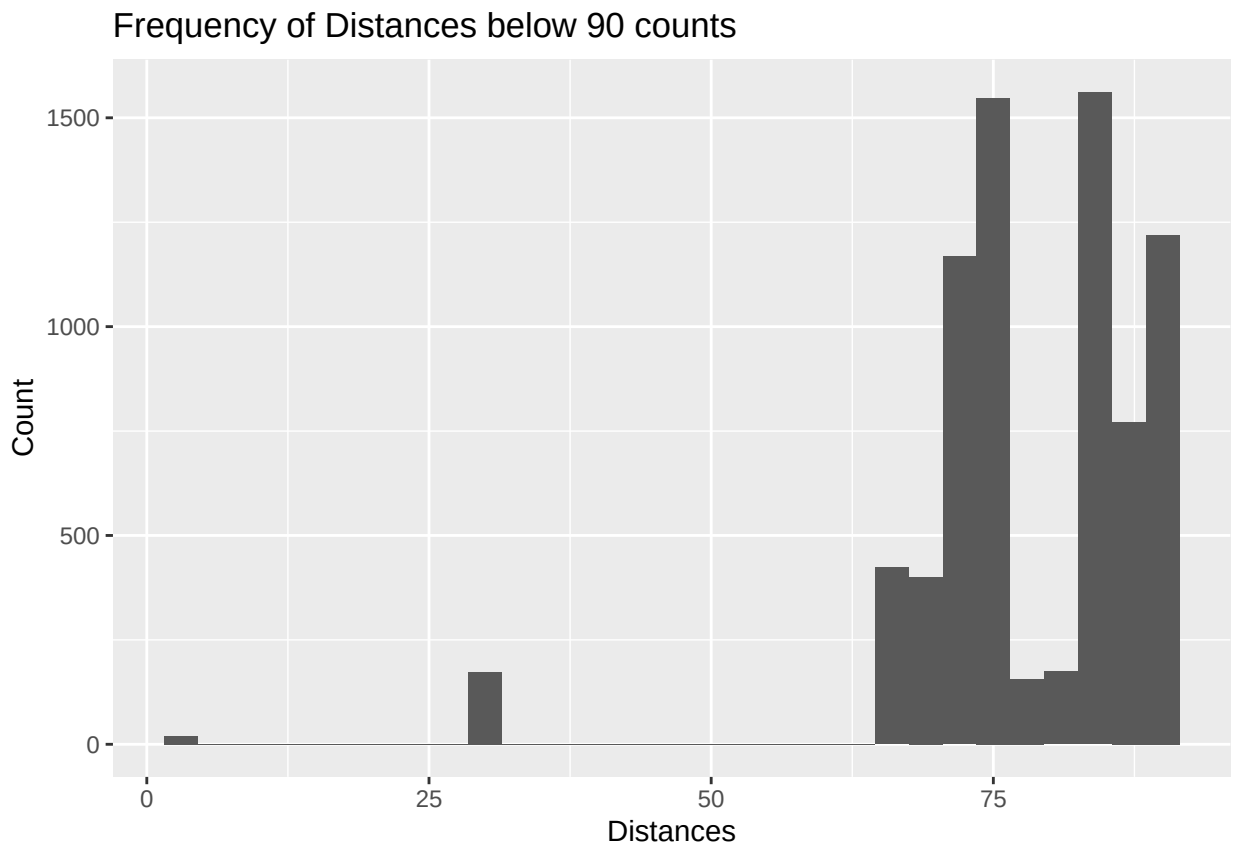
```
##    ORIG_DEST ORIG_DEST_TOTAL
## 1  SFO - LAX            4176
## 2  LAX - SFO            4164
## 3  ORD - LGA            3580
## 4  LGA - ORD            3576
## 5  LAX - LAS            3257
## 6  LAS - LAX            3254
## 7  LAX - JFK            3162
## 8  JFK - LAX            3158
## 9  LAX - SEA            2502
## 10 SEA - LAX            2497
```

10 most profitable round trip routes

```
# Creating a Table to show Profits
Profit <- data.table::copy(Flights)

# Dropping all unnecessary columns
drops <- c("FL_DATE",
           "TAIL_NUM",
```

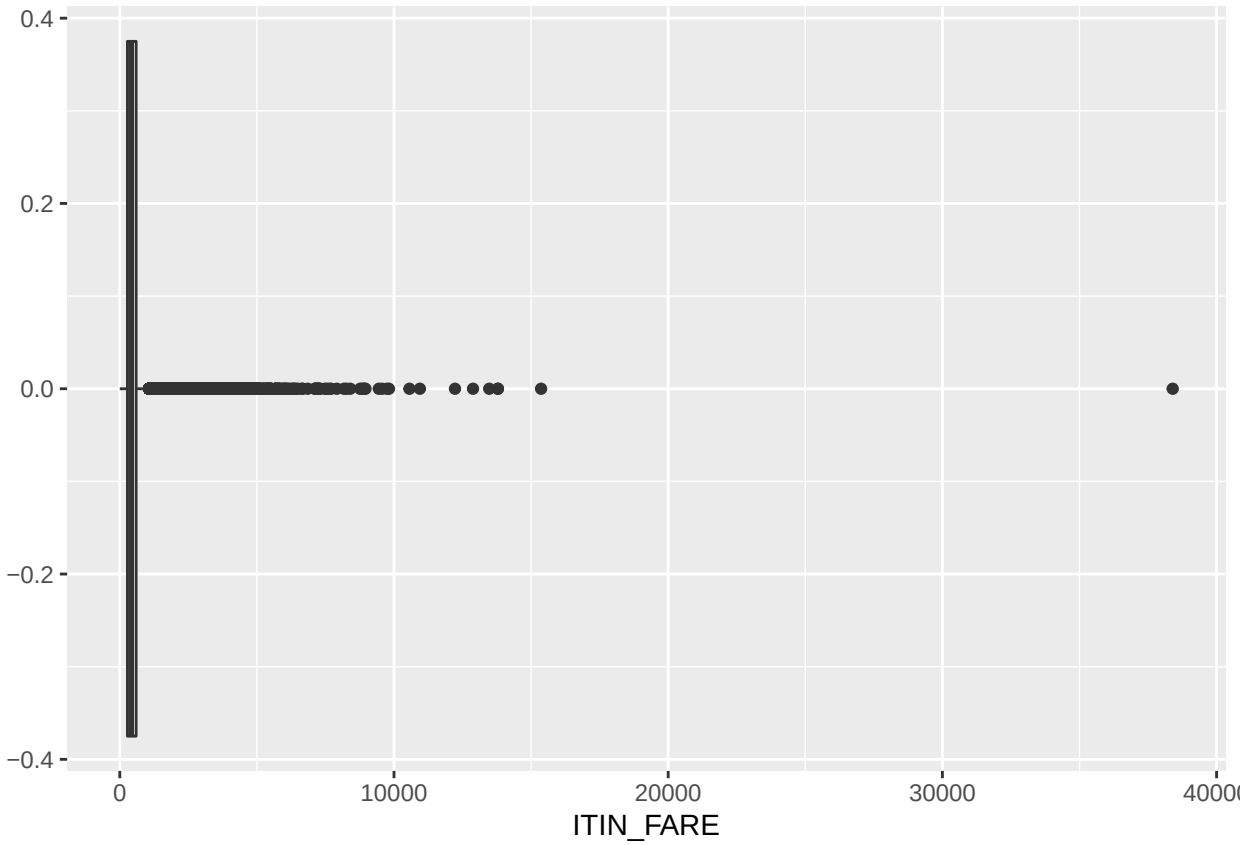
```
            "OP_CARRIER_FL_NUM",
            "ORIGIN_AIRPORT_ID",
            "ORIGIN_CITY_NAME",
            "DEST_AIRPORT_ID",
            "DEST_CITY_NAME")
Profit <- Profit[ , !(names(Profit) %in% drops)]

# Moving columns
Profit <- Profit %>% relocate(ORIG_DEST, .after = DESTINATION)

# Subtracting all delay and arrival time by 15 minutes since the first 15 min of a delay are free
Profit$DEP_DELAY <- (Profit$DEP_DELAY - 15)
Profit$ARR_DELAY <- (Profit$ARR_DELAY - 15)

# All flights that arrive early and have negative values incur no cost.
Profit$DEP_DELAY[Profit$DEP_DELAY < 0] <- 0
Profit$ARR_DELAY[Profit$ARR_DELAY < 0] <- 0

# Checking distance data
DISTANCE_COUNT <- Profit %>%
  group_by(DISTANCE) %>%
  count

# Data is imperfect here there are NAN Values, negative values, Missing values, and Values that have be
Profit <- Profit[!(Profit$DISTANCE=="NAN" | Profit$DISTANCE=="Hundred" | Profit$DISTANCE=="Twenty"| Pro

# Fixing negative value
Profit$DISTANCE[Profit$DISTANCE == -1947] <- 1947.0

# Passing the Distance column as numeric
Profit$DISTANCE <- as.numeric(Profit$DISTANCE)

# Distance Histogram
H <- ggplot(data = subset(Profit, DISTANCE <= 90), aes(x = DISTANCE))+
   geom_histogram()
H + ggtitle("Frequency of Distances below 90 counts") +
  xlab("Distances") + ylab("Count")
```

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




```r
# Outliers located below 10 miles in distance
# 20 observations have distances equal to 2 miles with Airports that are further than 2 miles
# Removing 20 observations
Profit <- Profit[!(Profit$DISTANCE=="2"),]

# Setting up Cost per distance column (Fuel, Oil, Maintenance, Crew - $8) + (Depreciation, Insurance, O
Profit$DISTANCE_COST <- (Profit$DISTANCE * 9.18)

# Setting up Cost of delay for every minute over 15 minutes.
Profit$DEP_DELAY_COST <- (Profit$DEP_DELAY * 75)
Profit$ARR_DELAY_COST <- (Profit$ARR_DELAY * 75)

# Creating dataframe with Airport type and Airport code
Airport_type <- data.frame(Airport_codes$IATA_CODE, Airport_codes$TYPE)
names(Airport_type)[1] <- 'ORIGIN'
names(Airport_type)[2] <- 'TYPE'
Airport_type <- Airport_type[!(Airport_type$ORIGIN==""),]

# Adding Origin Airport Type to Profit dataframe
Profit <- Profit %>% left_join(Airport_type, by = "ORIGIN")
Profit <- Profit %>% relocate(TYPE, .after = ORIGIN)
names(Profit)[3] <- 'ORIGIN_TYPE'

# Adding Destination Airport Type to Profit dataframe
names(Airport_type)[1] <- 'DESTINATION'
Profit <- Profit %>% left_join(Airport_type, by = "DESTINATION")
```

```r
Profit <- Profit %>% relocate(TYPE, .after = DESTINATION)
names(Profit)[5] <- 'DESTINATION_TYPE'

# Setting up Airport Operational Costs
Profit <- cbind(Profit, Profit[,c(3,5)])
names(Profit)[16] <- 'ORIG_TYPE_COST'
names(Profit)[17] <- 'DEST_TYPE_COST'

Profit["ORIG_TYPE_COST"][Profit["ORIG_TYPE_COST"] == "large_airport"] <- 10000
Profit["ORIG_TYPE_COST"][Profit["ORIG_TYPE_COST"] == "medium_airport"] <- 5000
Profit["DEST_TYPE_COST"][Profit["DEST_TYPE_COST"] == "large_airport"] <- 10000
Profit["DEST_TYPE_COST"][Profit["DEST_TYPE_COST"] == "medium_airport"] <- 5000

# Getting total Airport Operational Cost for each Round Trip
Profit$AIRPORT_COST <- (as.numeric(Profit$ORIG_TYPE_COST) + as.numeric(Profit$DEST_TYPE_COST))

# Looking at the different Fare counts
FARE_COUNT <- Tickets %>%
  group_by(ITIN_FARE) %>%
  count

# Noticed Missing values, values that have 200 $ and 820 $$$, and Values that have $ in front, removing
Tickets <- Tickets[!(Tickets$ITIN_FARE == "" | Tickets$ITIN_FARE == "200 $" | Tickets$ITIN_FARE == "820
Tickets$ITIN_FARE[Tickets$ITIN_FARE == "$ 100.00"] <- 100
Tickets$ITIN_FARE <- as.numeric(Tickets$ITIN_FARE)

# Looking at the distribution of ticket prices (ITIN_FARE) with box-plots
ggplot(Tickets, aes(x=ITIN_FARE)) + geom_boxplot()
```




```
summary(Tickets$ITIN_FARE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   279.0   416.0   473.5   596.0 38400.0
```

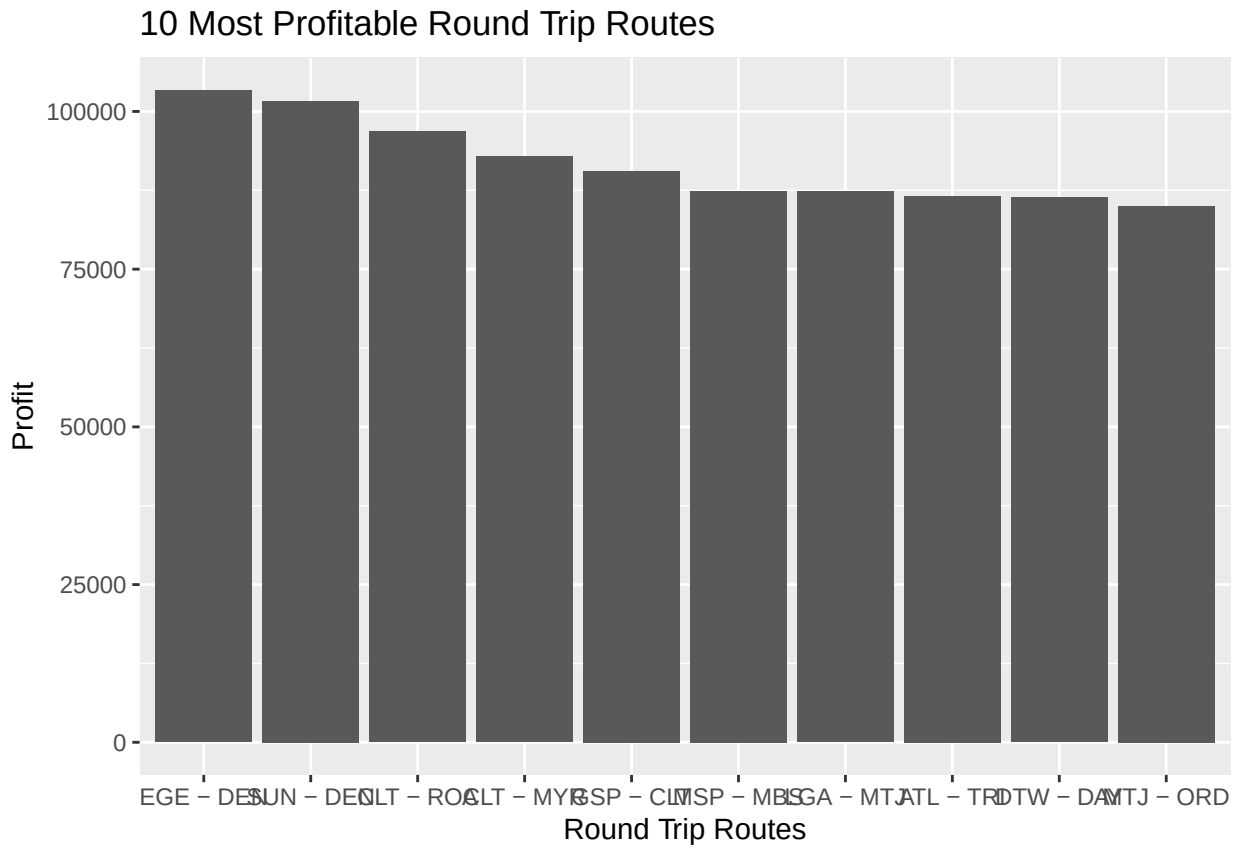
```
before <- dim(Tickets)

quartiles <- quantile(Tickets$ITIN_FARE, probs=c(.25, .75), na.rm = FALSE)
IQR <- IQR(Tickets$ITIN_FARE)

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

Tickets <- subset(Tickets, Tickets$ITIN_FARE > Lower & Tickets$ITIN_FARE < Upper)

after <- dim(Tickets)
difference <- before - after
difference
```

```
## [1] 30688     0
```

```
# removed 30688 outliers from ticket prices (ITIN_FARE)

# Getting Average Ticket Price for Each roundtrip
```

```r
Ticket_Fare_ave <- Tickets %>%
  group_by(ORIG_DEST) %>%
  summarise_at(vars(ITIN_FARE),
               list(name = mean))
names(Ticket_Fare_ave)[2] <- 'ITIN_FARE'

# Joining average ticket fare to the profit table and calculating the passengers, fare revenue, and bag
Profit <- Profit %>% left_join(Ticket_Fare_ave, by = "ORIG_DEST")
Profit$PASSENGERS <- round(Profit$OCCUPANCY_RATE * 200)
Profit$FARE_REVE <- (Profit$PASSENGERS * Profit$ITIN_FARE)
Profit$BAGG_REVE <- (Profit$PASSENGERS * .5 * 70)

# calculating the total cost of the routes
Profit$COST <- (as.numeric(Profit$DISTANCE_COST) +
                  as.numeric(Profit$DEP_DELAY_COST) +
                  as.numeric(Profit$ARR_DELAY_COST) +
                  as.numeric(Profit$AIRPORT_COST))

# calculating the revenue of each route
Profit$REVENUE <- (as.numeric(Profit$FARE_REVE) +
                     as.numeric(Profit$BAGG_REVE))

# calculating the the profit of each route
Profit$PROFIT <- (Profit$REVENUE - Profit$COST)

# Removing all observations that have na in any fields to be able to do analysis.
Profit <- na.omit(Profit)

# Creating Table for Route Profits
Route_Profit <- data.frame(Profit$ORIG_DEST,
                                  as.numeric(Profit$PROFIT),
                                  as.numeric(Profit$REVENUE),
                                  as.numeric(Profit$COST),
                                  as.numeric(Profit$DISTANCE_COST),
                                  as.numeric(Profit$DEP_DELAY_COST),
                                  as.numeric(Profit$ARR_DELAY_COST),
                                  as.numeric(Profit$ORIG_TYPE_COST),
                                  as.numeric(Profit$DEST_TYPE_COST),
                                  as.numeric(Profit$AIRPORT_COST),
                                  as.numeric(Profit$ITIN_FARE),
                                  as.numeric(Profit$FARE_REVE),
                                  as.numeric(Profit$BAGG_REVE))

# Renaming Columns
names(Route_Profit)[1] <- 'ORIG_DEST'
names(Route_Profit)[2] <- 'PROFIT'
names(Route_Profit)[3] <- 'REVENUE'
names(Route_Profit)[4] <- 'COST'
names(Route_Profit)[5] <- 'DISTANCE_COST'
names(Route_Profit)[6] <- 'DEP_DELAY_COST'
names(Route_Profit)[7] <- 'ARR_DELAY_COST'
names(Route_Profit)[8] <- 'ORIG_TYPE_COST'
names(Route_Profit)[9] <- 'DEST_TYPE_COST'
```

```r
names(Route_Profit)[10] <- 'AIRPORT_COST'
names(Route_Profit)[11] <- 'ITIN_FARE'
names(Route_Profit)[12] <- 'FARE_REVE'
names(Route_Profit)[13] <- 'BAGG_REVE'

# Grouping by Origin and Destination
Route_Profit <- group_by(Route_Profit, ORIG_DEST)

# Taking the average of all profits, costs, and revenues
Route_Profit <- Route_Profit %>% mutate(PROFIT = mean(PROFIT),
                                                    REVENUE = mean(REVENUE),
                                                    COST = mean(COST),
                                                    DISTANCE_COST = mean(DISTANCE_COST),
                                                    DEP_DELAY_COST = mean(DEP_DELAY_COST),
                                                    ARR_DELAY_COST = mean(ARR_DELAY_COST),
                                                    ORIG_TYPE_COST = mean(ORIG_TYPE_COST),
                                                    DEST_TYPE_COST = mean(DEST_TYPE_COST),
                                                    AIRPORT_COST = mean(AIRPORT_COST),
                                                    ITIN_FARE = mean(ITIN_FARE),
                                                    BAGG_REVE = mean(BAGG_REVE))
# Keeping all Distinct Origin and Destinations
Route_Profit <- Route_Profit %>% distinct(ORIG_DEST, .keep_all = TRUE)

# Ungrouping
Route_Profit <- ungroup(Route_Profit)

# Adding total round trip flights to the table
Route_Profit <- Route_Profit %>% left_join(ORIG_DEST_COUNT, by = "ORIG_DEST")

# Sorting route by greatest to least
Route_Profit <- Route_Profit[with(Route_Profit, order(-PROFIT)),]
Route_Profit_10 <- Route_Profit[1:10,]

# Graphing Results
ggplot(Route_Profit_10, aes(x = reorder(ORIG_DEST, -PROFIT), y = PROFIT)) + geom_bar(stat = "identity")
```




```
# 10 most profitable route table
print.data.frame(Route_Profit_10)
```

```
##    ORIG_DEST    PROFIT  REVENUE     COST DISTANCE_COST DEP_DELAY_COST
## 1  EGE - DEN 103346.66 122031.1 18684.49       1101.60      1158.5041
## 2  SUN - DEN 101619.11 122896.8 21277.67       5113.26       505.9322
## 3  CLT - ROA  96814.72 120251.1 23436.36       1422.90      1032.6923
## 4  CLT - MYR  92829.16 109852.6 17023.40       1441.26       291.0714
## 5  GSP - CLT  90536.15 112312.4 21776.29        688.50       588.1477
## 6  MSP - MBS  87404.69 113401.8 25997.08       4250.34       852.1739
## 7  LGA - MTJ  87296.38 128062.2 40765.80      16615.80      4650.0000
## 8  ATL - TRI  86583.89 109519.9 22936.02       2083.86       443.8272
## 9  DTW - DAY  86328.55 110521.3 24192.77       1523.88      1398.9550
## 10 MTJ - ORD  85035.81 116507.9 31472.06       9923.58      3245.4545
##    ARR_DELAY_COST ORIG_TYPE_COST DEST_TYPE_COST AIRPORT_COST ITIN_FARE
## 1       1424.3852           5000          10000        15000  884.0000
## 2        658.4746           5000          10000        15000  884.0000
## 3        980.7692          10000          10000        20000  793.0000
## 4        291.0714          10000           5000        15000  818.5000
## 5        499.6438          10000          10000        20000  832.0000
## 6        894.5652          10000          10000        20000  839.8571
## 7       4500.0000          10000           5000        15000  646.1818
## 8        408.3333          10000          10000        20000  819.3333
## 9       1269.9357          10000          10000        20000  820.6667
## 10      3303.0303           5000          10000        15000  832.5000
##    FARE_REVE BAGG_REVE ORIG_DEST_TOTAL
```

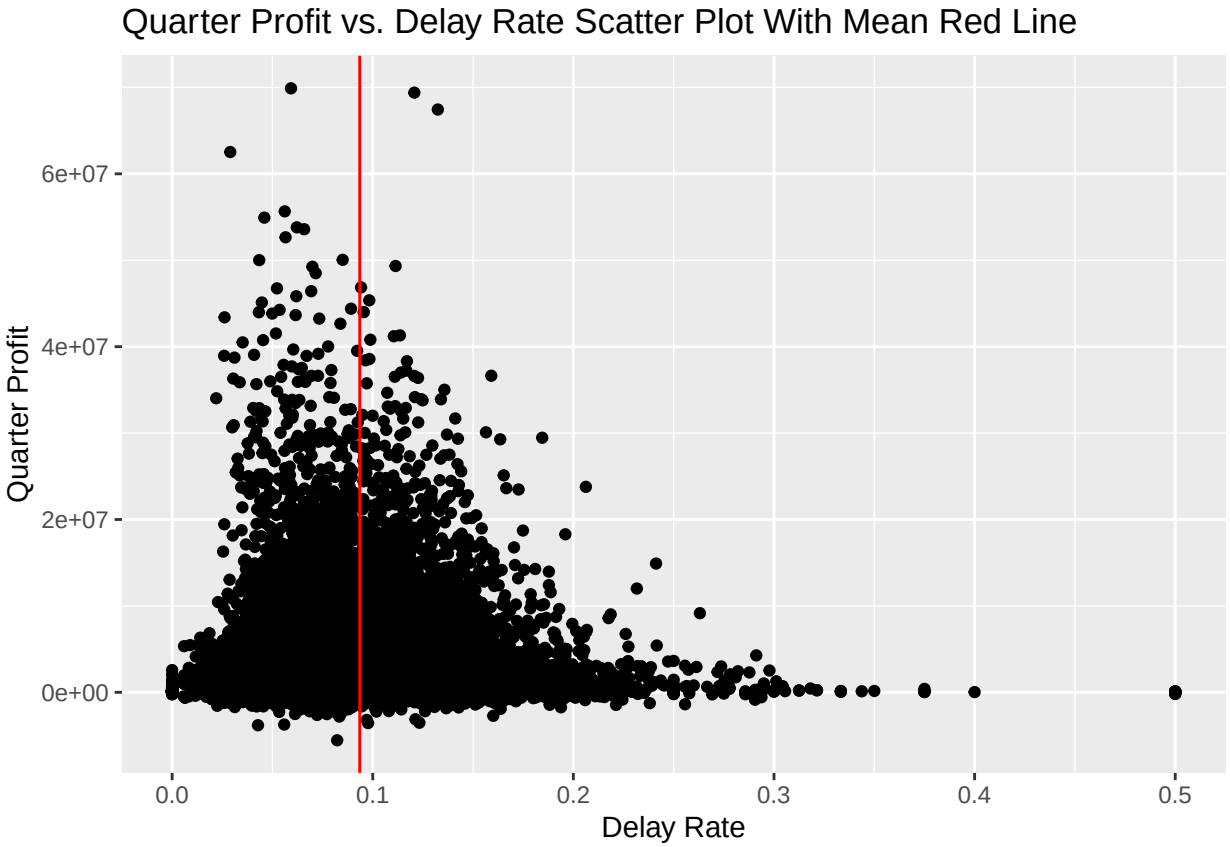
```
## 1  118456.00  4647.541            244
## 2  104312.00  4680.508             59
## 3  155428.00  5083.077             39
## 4  127686.00  4504.792            672
## 5  138112.00  4533.951            772
## 6  162932.29  4536.812             69
## 7  121482.18  6580.000              1
## 8  127816.00  4486.770            486
## 9   50881.33  4520.740            311
## 10  96570.00  4700.606             99
```

5 round trip routes that you recommend to invest in

```
# Calculating the average fare price for each route and carrier
Ticket_Fare_ave <- Tickets %>%
  group_by(REPORTING_CARRIER, ORIG_DEST) %>%
  summarise_at(vars(ITIN_FARE),
               list(name = mean))
names(Ticket_Fare_ave)[1] <- 'OP_CARRIER'
names(Ticket_Fare_ave)[3] <- 'ITIN_FARE'

# Creating Date Frame that shows profit and information of each route by the Carrier who was operating
Carrier_Route_Profit <- data.table::copy(Profit)

# Adding new average fare price
Carrier_Route_Profit <- Carrier_Route_Profit %>% left_join(Ticket_Fare_ave, by = c("ORIG_DEST", "OP_CARR

# Dropping old average fare price
Carrier_Route_Profit <- Carrier_Route_Profit[ -c(19) ]

# Moving new average fare price and renaming
Carrier_Route_Profit <- Carrier_Route_Profit %>% relocate(ITIN_FARE.y, .after = AIRPORT_COST)
names(Carrier_Route_Profit)[19] <- 'ITIN_FARE'

# Calculating Fare Revenue with new average fare for each route and carrier
Carrier_Route_Profit$FARE_REVE <- (Carrier_Route_Profit$PASSENGERS * Carrier_Route_Profit$ITIN_FARE)

# calculating the revenue of each route with new average fare for each route and carrier
Carrier_Route_Profit$REVENUE <- (as.numeric(Carrier_Route_Profit$FARE_REVE) +
                    as.numeric(Carrier_Route_Profit$BAGG_REVE))

# calculating the the profit of each route with new average fare for each route and carrier
Carrier_Route_Profit$PROFIT <- (Carrier_Route_Profit$REVENUE - Carrier_Route_Profit$COST)

# Removing all observations that have na in any fields to be able to do analysis.
Carrier_Route_Profit <- na.omit(Carrier_Route_Profit)

# Replacing Delay Costs with 1 or zero to be able to calculate the rate of delay for each route.
Carrier_Route_Profit$DEP_DELAY_COST <- ifelse(Carrier_Route_Profit$DEP_DELAY_COST == "0", 0, 1)
Carrier_Route_Profit$ARR_DELAY_COST <- ifelse(Carrier_Route_Profit$ARR_DELAY_COST == "0", 0, 1)
```

```r
# Creating Table for Route Profits
Carrier_Route_Profit <- data.frame(Carrier_Route_Profit$OP_CARRIER,
                                   Carrier_Route_Profit$ORIG_DEST,
                                   as.numeric(Carrier_Route_Profit$PROFIT),
                                   as.numeric(Carrier_Route_Profit$REVENUE),
                                   as.numeric(Carrier_Route_Profit$COST),
                                   as.numeric(Carrier_Route_Profit$DISTANCE_COST),
                                   as.numeric(Carrier_Route_Profit$DEP_DELAY_COST),
                                   as.numeric(Carrier_Route_Profit$ARR_DELAY_COST),
                                   as.numeric(Carrier_Route_Profit$ORIG_TYPE_COST),
                                   as.numeric(Carrier_Route_Profit$DEST_TYPE_COST),
                                   as.numeric(Carrier_Route_Profit$AIRPORT_COST),
                                   as.numeric(Carrier_Route_Profit$ITIN_FARE),
                                   as.numeric(Carrier_Route_Profit$FARE_REVE),
                                   as.numeric(Carrier_Route_Profit$BAGG_REVE))

# Renaming Columns
names(Carrier_Route_Profit)[1] <- 'OP_CARRIER'
names(Carrier_Route_Profit)[2] <- 'ORIG_DEST'
names(Carrier_Route_Profit)[3] <- 'PROFIT'
names(Carrier_Route_Profit)[4] <- 'REVENUE'
names(Carrier_Route_Profit)[5] <- 'COST'
names(Carrier_Route_Profit)[6] <- 'DISTANCE_COST'
names(Carrier_Route_Profit)[7] <- 'DEP_DELAY_COUNT'
names(Carrier_Route_Profit)[8] <- 'ARR_DELAY_COUNT'
names(Carrier_Route_Profit)[9] <- 'ORIG_TYPE_COST'
names(Carrier_Route_Profit)[10] <- 'DEST_TYPE_COST'
names(Carrier_Route_Profit)[11] <- 'AIRPORT_COST'
names(Carrier_Route_Profit)[12] <- 'ITIN_FARE'
names(Carrier_Route_Profit)[13] <- 'FARE_REVE'
names(Carrier_Route_Profit)[14] <- 'BAGG_REVE'

# Getting averages or sum for each value grouped by the Carrier and Route.
Carrier_Route_Profit <- group_by(Carrier_Route_Profit, OP_CARRIER, ORIG_DEST)
Carrier_Route_Profit <- Carrier_Route_Profit %>% mutate(PROFIT = mean(PROFIT),
                                                        REVENUE = mean(REVENUE),
                                                        COST = mean(COST),
                                                        DISTANCE_COST = mean(DISTANCE_COST),
                                                        DEP_DELAY_COUNT = sum(DEP_DELAY_COUNT),
                                                        ARR_DELAY_COUNT = sum(ARR_DELAY_COUNT),
                                                        ORIG_TYPE_COST = mean(ORIG_TYPE_COST),
                                                        DEST_TYPE_COST = mean(DEST_TYPE_COST),
                                                        AIRPORT_COST = mean(AIRPORT_COST),
                                                        ITIN_FARE = mean(ITIN_FARE),
                                                        FARE_REVE = mean(FARE_REVE),
                                                        BAGG_REVE = mean(BAGG_REVE))

# Counting the number of Roundtrips for each route by Carrier
ORIG_DEST_COUNT <- Carrier_Route_Profit %>%
  group_by(OP_CARRIER, ORIG_DEST) %>%
  count

# Joining the total number of roundtrips by Carrier to the table
```

```r
Carrier_Route_Profit <- Carrier_Route_Profit %>% left_join(ORIG_DEST_COUNT, by = c("ORIG_DEST", "OP_CARR

# Keeping all distinct routes and carriers
Carrier_Route_Profit <- Carrier_Route_Profit %>% distinct(OP_CARRIER, ORIG_DEST, .keep_all = TRUE)

# Renaming columns
names(Carrier_Route_Profit)[15] <- 'ORIG_DEST_TOTAL'

# Ungrouping
Carrier_Route_Profit <- ungroup(Carrier_Route_Profit)

# Calculating the Delay Rate for each route and carrier
Carrier_Route_Profit$DEP_DELAY_COUNT <- as.numeric(Carrier_Route_Profit$DEP_DELAY_COUNT)
Carrier_Route_Profit$ARR_DELAY_COUNT <- as.numeric(Carrier_Route_Profit$ARR_DELAY_COUNT)
Carrier_Route_Profit$DELAY_RATE <- ((Carrier_Route_Profit$DEP_DELAY_COUNT + Carrier_Route_Profit$ARR_DEL

# Calculating the profit made that quarter by individual airlines for each route
Carrier_Route_Profit$QUARTER_PROFIT <- (Carrier_Route_Profit$PROFIT * Carrier_Route_Profit$ORIG_DEST_TOT

# Filtering the routes by number of flights and delay rate by the mean
Carrier_Route_Profit_Rec <- filter(Carrier_Route_Profit, ORIG_DEST_TOTAL > 100 & DELAY_RATE > mean(Carri

# Sorting by decreasing Profit
Carrier_Route_Profit_Rec <- Carrier_Route_Profit_Rec[with(Carrier_Route_Profit_Rec, order(-QUARTER_PROFI

# Creating Scatterplot
ggplot(Carrier_Route_Profit, aes(x=DELAY_RATE, y=QUARTER_PROFIT)) + geom_point() + geom_vline(xintercep
```




```
# Keeping my five recommendations only
Carrier_Route_Profit_Rec <- Carrier_Route_Profit_Rec[1:6,]
Carrier_Route_Profit_Rec <- Carrier_Route_Profit_Rec[-2,]
print.data.frame(Carrier_Route_Profit_Rec)
```

```
##   OP_CARRIER ORIG_DEST   PROFIT  REVENUE     COST DISTANCE_COST DEP_DELAY_COUNT
## 1         YX LGA - DCA 41578.90 65359.38 23780.47       1964.52             356
## 2         DL ATL - LGA 37891.86 66566.57 28674.71       6995.16             285
## 3         WN DAL - HOU 30645.52 53694.71 23049.19       2194.02             316
## 4         AA CLT - MCO 52136.46 77303.05 25166.58       4296.24             168
## 5         AA CLT - DFW 49159.79 78608.95 29449.16       8592.48             164
##   ARR_DELAY_COUNT ORIG_TYPE_COST DEST_TYPE_COST AIRPORT_COST ITIN_FARE
## 1             450          10000          10000        20000  467.3662
## 2             295          10000          10000        20000  476.8147
## 3             260          10000          10000        20000  374.7912
## 4             174          10000          10000        20000  562.4173
## 5             178          10000          10000        20000  569.7778
##   FARE_REVE BAGG_REVE ORIG_DEST_TOTAL DELAY_RATE QUARTER_PROFIT
## 1  60805.77  4553.607            1669 0.12073098       69395192
## 2  62014.48  4552.097            1302 0.11136713       49335207
## 3  49108.68  4586.030            1529 0.09417920       46857003
## 4  72774.21  4528.839             870 0.09827586       45358723
## 5  74059.65  4549.296             895 0.09553073       43998014
```

My 5 recommendations for round trips would be LGA - DCA, ATL - LGA, DAL - HOU, CLT - MCO, and CLT - DFW. The factors I chose to evaluate the routes were the profit of each trip, delay rate and quarter

profit of each route. For this analysis I looked at the profit made on each route by each airline instead of merging all the airlines into one route, this allowed me to show how each airline operated on the route and which ones operated better. Since the cause of delays were not given in the databases I could not tell which delays were caused by weather, airport error, or airline error so I chose to look at all the routes with the airline that was operating it to see how well or poorly the airline was operating based on the rate of the arrival or departure being delayed for the round trip. The other factor I took into consideration was the profit made on the quarter to show how much demand and money there is to be made a quarter flying these routes, though these routes will not be flown that often by our planes there is a big market that we can come in and operate in. Since punctuality is a big part of this company I looked at routes that had a sample size greater than a hundred and that had delay rates that were above the average of the other airlines and routes. These routes that I am singling out show that there is opportunity to come in and operate on these routes better than the competitors that are operating this route poorly and due to being punctual consumers will know to rely on our airline versus others.

4. Number of round trip flights it will take to breakeven

```
# Creating a new column to calculate the breakeven for each route.
Carrier_Route_Profit_Rec$BREAKEVEN_TRIPS <- (90000000 / Carrier_Route_Profit_Rec$PROFIT)
Carrier_Route_Profit_Rec$BREAKEVEN_TRIPS <- ceiling(Carrier_Route_Profit_Rec$BREAKEVEN_TRIPS)
# removing unnecessary columns
Carrier_Route_Profit_Rec_Breakeven <- Carrier_Route_Profit_Rec[ -c(6:15) ]
print.data.frame(Carrier_Route_Profit_Rec_Breakeven)
```

```
##   OP_CARRIER ORIG_DEST   PROFIT  REVENUE     COST DELAY_RATE QUARTER_PROFIT
## 1         YX LGA - DCA 41578.90 65359.38 23780.47 0.12073098       69395192
## 2         DL ATL - LGA 37891.86 66566.57 28674.71 0.11136713       49335207
## 3         WN DAL - HOU 30645.52 53694.71 23049.19 0.09417920       46857003
## 4         AA CLT - MCO 52136.46 77303.05 25166.58 0.09827586       45358723
## 5         AA CLT - DFW 49159.79 78608.95 29449.16 0.09553073       43998014
##   BREAKEVEN_TRIPS
## 1            2165
## 2            2376
## 3            2937
## 4            1727
## 5            1831
```

Key Performance Indicators (KPI's)

KPI's I recommend tracking are profit, delay rate, and quarter profit. Profit and quarter profit is of course the main one it is important to observe the change in profits to see if you are doing better than the average or worse and determine the reason of the change, the more time that flight is flown per quarter the better understanding the company will have of how well we are doing on the individual flight and how well we are doing in the quarter. Quarter profit is also another indicator that will help us determine how our flight route is performing if it grows we should consider adding another plane on that route to meet demands if it shrinks we should explore other routes. Delay rate is important to the company as we are using this metric to cut into existing routes where other airlines are performing above the average rate of delay, we need to ensure that our operations are performing at a delay rate lower than our competitors on that route so consumers know we are reliable.

Further Work

Future work I would do would be to try to see which airports have the highest number of delays and see if the airport could be attributed to the cause of the delay so as to avoid that airport when making future recommendations. I would also see if there is any correlation between the airline carrier and the number of delays so as to see how many of the delays could be attributed to the operations of the airline and observe which airlines have less delay rates so as to see what is working for them to prevent delays and which airlines have more delays due to operations that we can compete with. I also would look at the delays in minutes and counts to explore if there are any major outliers that we can get rid of that won't skew our data one way. I also noticed while looking through the ticket fare there are a lot of tickets that were sold for nothing or very cheap, I did not take them out as I thought that they were due to exchanges, refunds, or other reasons. I would take all the cheap tickets below a certain count frequency out and run this analysis again to see what my new results are all of those free/cheap tickets are skewing my data so that ticket prices are cheaper and that profits are less due to that.