Hypothesis and Data

In this project, we will be working on finding factors that lead to flight delays. Among the general public delays are often associated with time of the year or day of the week. I believe that the perception that more delays happen in December or on weekends may not be grounded in data. It is just that people there are more flights in holidays and weekends and the impact of the delays are felt more in those times. In particular, we will look in to the factors affecting flight arrival delays.

Data

Here, we start with the data from Kaggle for airline delays. The data has 1,936,758 rows and 30 columns. The columns are

- Name: Description
  - This variable should not impact arrival delay of a flight.
- Year: 2008
  - Since all the data is for 2008, we can ignore this column.
- Month: 1-12 where 1 stands for January and 12 for December.
- DayofMonth: 1-31
  - Ignoring this column.
- DayOfWeek: 1 (Monday) - 7 (Sunday)
- DepTime: local actual departure time in hhmm format.
- CRSDepTime: local scheduled departure time in hhmm format.
  - Departure time according to computerized reservation system.
- ArrTime: local actual arrival time in hhmm format.
- CRSArrTime: local scheduled arrival time in hhmm format.
  - Arrival time according to Computerized reservation system.
- UniqueCarrier: unique carrier code
  - There are 20 unique carriers.
- FlightNum: flight number
  - We will not get much useful information from flight number.
- TailNum: plane tail number
  - We will not get much useful information from Tail number of an airplane.
- ActualElapsedTime: in minutes
- CRSElapsedTime: in minutes
  - Elapsed time according to Computerized reservation system.
- AirTime: in minutes
- ArrDelay: arrival delay, in minutes
- DepDelay: departure delay, in minutes
- Origin: origin IATA airport code

    - We will ignore this column
- Dest: destination IATA airport code
    - We will ignore this column
- Distance: in miles
- TaxiIn: taxi in time, in minutes
- TaxiOut: taxi out time in minutes
- Cancelled: was the flight cancelled?
    - Since we are going to study arrival delay, cancellation status may not be helpful.
- CancellationCode: reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
    - Since we are ignoring cancellation, we can ignore CancellationCode too.
- Diverted: 1 = yes, 0 = no
    - Diversion column will be ignored because it is rare. Also, a plane may still be on time according to the diversion schedule after departure.
- CarrierDelay: in minutes
    - Delay due to the carrier
- WeatherDelay: in minutes
    - Delay due to bad weather
- NASDelay: in minutes
    - Delay due to National Airspace System. This may happen due to non-extreme weather conditions, traffic volume at airport, etc.
- SecurityDelay: in minutes
    - May happen because of some security event, such as, airport evacuation
- LateAircraftDelay: in minutes

Data Preparation

We will ignore all the columns in Orange above. We have a lot of data here, about 2 million rows. Hence, we will also have the luxury of ignoring all the rows with NA.

Initial Data Analysis

We will analyze the impact of all the variables through some analysis and visualizations. This provides us with some idea of the data. We would also be able understand which of the 20 columns will have impact of arrival delay. Since R can be slow with large amounts of data, we will do some of the plots on a random sample of 300 size.

Summary

We have converted DepTime, CRSDepTime, ArrTime and CRSArrTime from ‘hhmm’ format to minutes. From the summary, we can observe a few things. The mean and median of DayOfWeek are close to 4, which means that the number of delayed flights are evenly distributed over the week. Similarly, flight delays are also evenly distributed over the year. We see DepTime, CRSDepTime, ArrTime and CRSArrTime all have means (and medians) greater than 720 (# of minutes at 12 noon). This maybe because there are no flights in the early part of the day as evidenced by high first quartile numbers.

Month	DayOfWeek	DepTime	CRSDepTime	ArrTime
Min. : 1.000	Min. :1.00	Min. : 1	Min. : 0.0	Min. : 1.0
1st Qu.: 3.000	1st Qu.:2.00	1st Qu.: 752	1st Qu.:710.0	1st Qu.: 806.0
Median : 6.000	Median :4.00	Median : 978	Median :929.0	Median :1057.0
Mean : 6.065	Mean :3.98	Mean : 947	Mean :903.8	Mean : 981.8
3rd Qu.: 9.000	3rd Qu.:6.00	3rd Qu.:1164	3rd Qu.:1110.0	3rd Qu.:1248.0
Max. :12.000	Max. :7.00	Max. :1440	Max. :1439.0	Max. :1440.0

CRSArrTime	UniqueCarrier	ActualElapsedTime	CRSElapsedTime	ArrDelay
Min. : 0	WN :203559	Min. : 14.0	Min. :-21.0	Min. : 15.00
1st Qu.:820	AA :132257	1st Qu.: 83.0	1st Qu.: 80.0	1st Qu.: 26.00
Median :1042	MQ : 97555	Median : 118.0	Median :115.0	Median : 43.00
Mean :1003	UA : 95465	Mean : 135.4	Mean :131.8	Mean : 63.29
3rd Qu.:1222	OO : 88991	3rd Qu.: 167.0	3rd Qu.:161.0	3rd Qu.: 79.00
Max. :1439	DL : 72252	Max. :1114.0	Max. :660.0	Max. :2461.00
	(Other):557407			

DepDelay	TaxiIn	TaxiOut	AirTime
Min. : 6.00	Min. : 0.000	Min. : 0.00	Min. : 0.0

1st Qu.: 24.00	1st Qu.: 4.000	1st Qu.: 11.00	1st Qu.: 58.0
Median : 41.00	Median : 6.000	Median : 16.00	Median : 90.0
Mean : 59.68	Mean : 7.297	Mean : 20.66	Mean : 107.4
3rd Qu.: 75.00	3rd Qu.: 8.000	3rd Qu.: 24.00	3rd Qu.: 136.0
Max. :2467.00	Max. :240.000	Max. :422.00	Max. :1091.0

Distance	CarrierDelay	WeatherDelay	NASDelay
Min. : 11.0	Min. : 0.00	Min. : 0.000	Min. : 0.00
1st Qu.: 334.0	1st Qu.: 0.00	1st Qu.: 0.000	1st Qu.: 0.00
Median : 595.0	Median : 2.00	Median : 0.000	Median : 2.00
Mean : 741.6	Mean : 19.18	Mean : 3.703	Mean : 15.02
3rd Qu.: 972.0	3rd Qu.: 21.00	3rd Qu.: 0.000	3rd Qu.: 15.00
Max. :4962.0	Max. :2436.00	Max. :1352.000	Max. :1357.00

SecurityDelay	LateAircraftDelay
Min. : 0.0000	Min. : 0.0
1st Qu.: 0.0000	1st Qu.: 0.0
Median : 0.0000	Median : 8.0
Mean : 0.0901	Mean : 25.3
3rd Qu.: 0.0000	3rd Qu.: 33.0
Max. :392.0000	Max. :1316.0

Day of the week

We expect more delays to happen on weekends and less so on weekdays.

DayOfWeek	1.0	2.0	3.0	4.0	5.0	6.0	7.0
Mean DepDelay (in mins)	59.9	60.4	57.3	57.4	59.6	59.2	63.6

We see that the highest mean departure delay happens on Sunday and then Tuesday. If we ignore the outliers, we see in the figure below that mean departure delay is still high on weekends but also on Wednesday.

Month

On grouping data departure delays on months, we observe that the mean departure delays in December is actually less than the mean departure delays in November. We also see that the lowest mean departure delays are in April and May. We can consider the month of the year as a variable in modeling departure delay.

Unique Carrier

We see that there is also a lot of variation in departure delay depending on carrier. We can consider the carrier also as a variable in modeling departure delay.

Visualizing columns as pairs

We will visualize some of the columns as pairs. Since observing all the pairs in graph does not provide any visual insights, we will plot only a few variables as pairs.
We can observe from the pairs plot below that there is positive linear relationship between Arrival delay and departure delay, which is expected. We also see some sort of positive linear relationship between arrival delay and Carrier Delay, Weather Delay, NAS Delay and Late Aircraft delay.
Observing linear positive relationship between Arrival Delay and Other variables

We see the linear relationship between ArrDelay and DepDelay, in the figure to the left.

We want to see the relationship between ArrDelay and Distance. We see the relationship here.

After calculating the cook distance, we see that there are some outliers. I removed the outliers but the regression line does not move much. The red line is with outliers and the green line is without outliers. Hence, outliers do not affect much the actual regression. I will henceforth, overlook any affect from outliers.

Linear Regression

We will start with a lot of the variables in the model and then prune the model based on p-value and R square value.

Multicollinearity

	Month	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	ActualElapsedTime
Month	1.00	0.01	-0.01	-0.01	0.00	0.00	0.00
DayOfWeek	0.01	1.00	0.02	0.03	0.01	0.02	0.00
DepTime	-0.01	0.02	1.00	0.84	0.37	0.71	-0.06
CRSDepTime	-0.01	0.03	0.84	1.00	0.29	0.73	-0.04
ArrTime	0.00	0.01	0.37	0.29	1.00	0.48	-0.03
CRSArrTime	0.00	0.02	0.71	0.73	0.48	1.00	0.03
ActualElapsedTime	0.00	0.00	-0.06	-0.04	-0.03	0.03	1.00
CRSElapsedTime	0.01	0.01	-0.04	-0.02	-0.03	0.05	0.96
DepDelay	0.02	0.01	0.12	0.04	-0.08	0.04	0.00
TaxiIn	0.02	0.01	-0.03	-0.04	0.04	0.00	0.16
TaxiOut	0.01	-0.01	0.00	-0.01	-0.01	0.03	0.32
Distance	0.01	0.01	-0.05	-0.03	-0.04	0.03	0.94
CarrierDelay	0.00	0.01	-0.05	-0.11	-0.08	-0.10	0.01
WeatherDelay	0.01	0.01	0.01	-0.01	-0.03	-0.01	0.00
NASDelay	0.01	-0.01	0.02	-0.03	0.03	0.01	0.18
SecurityDelay	0.00	0.00	-0.02	-0.02	-0.01	-0.01	0.01

	CRSElapsedTime	DepDelay	TaxiIn	TaxiOut	Distance	CarrierDelay	WeatherDelay

Month	0.01	0.02	0.02	0.01	0.01	0.00	0.01
DayOfWeek	0.01	0.01	0.01	-0.01	0.01	0.01	0.01
DepTime	-0.04	0.12	-0.03	0.00	-0.05	-0.05	0.01
CRSDepTime	-0.02	0.04	-0.04	-0.01	-0.03	-0.11	-0.01
ArrTime	-0.03	-0.08	0.04	-0.01	-0.04	-0.08	-0.03
CRSArrTime	0.05	0.04	0.00	0.03	0.03	-0.10	-0.01
ActualElapsedTime	0.96	0.00	0.16	0.32	0.94	0.01	0.00
CRSElapsedTime	1.00	0.03	0.09	0.13	0.98	0.03	-0.02
DepDelay	0.03	1.00	0.03	-0.02	0.01	0.53	0.24
TaxiIn	0.09	0.03	1.00	0.04	0.06	-0.02	0.03
TaxiOut	0.13	-0.02	0.04	1.00	0.07	-0.03	0.08
Distance	0.98	0.01	0.06	0.07	1.00	0.03	-0.02
CarrierDelay	0.03	0.53	-0.02	-0.03	0.03	1.00	-0.07
WeatherDelay	-0.02	0.24	0.03	0.08	-0.02	-0.07	1.00
NASDelay	0.05	0.23	0.23	0.43	0.02	-0.12	0.00
SecurityDelay	0.01	0.00	0.00	0.00	0.01	-0.02	-0.01

	NASDelay	SecurityDelay
Month	0.01	0.00
DayOfWeek	-0.01	0.00
DepTime	0.02	-0.02
CRSDepTime	-0.03	-0.02
ArrTime	0.03	-0.01
CRSArrTime	0.01	-0.01
ActualElapsedTime	0.18	0.01
CRSElapsedTime	0.05	0.01
DepDelay	0.23	0.00
TaxiIn	0.23	0.00

TaxiOut	0.43	0.00
Distance	0.02	0.01
CarrierDelay	-0.12	-0.02
WeatherDelay	0.00	-0.01
NASDelay	1.00	-0.01
SecurityDelay	-0.01	1.00

As we can see here, multicollinearity will be a problem when doing regression. We see many columns (highlighted with pink) with correlations greater than zero. We should choose only one of the column among the highly correlated column.

Hence, we start with Month, DayOfWeek, DepTime, ArrTime, ActualElapsedTime, DepDelay, TaxiIn, TaxiOut, WeatherDelay, NASDelay and SecurityDelay columns when doing regression.

We find all the variables to have very low p-values, and hence statistically significant. P-value is 0.01 for ActualElapsedTime and 0.007 for DayOfWeek. For columns Month, DepTime, ArrTime, DepDelay, TaxiIn, TaxiOut, WeatherDelay, NASDelay and SecurityDelay, p-value is almost zero.

R-squared is 0.97, which also very close to 1. Thus, the model explains 97% of the ArrDelay.

Coefficient:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-1.09E+01	6.00E-02	-182.177	<2.00E-16	***
Month	-1.09E-01	3.20E-03	-34	<2.00E-16	***
DayOfWeek	1.51E-02	5.64E-03	2.68	0.00737	**
DepTime	-1.84E-03	2.71E-05	-67.862	<2.00E-16	***
ArrTime	-2.53E-04	2.10E-05	-12.049	<2.00E-16	***
ActualElapsedTime	4.25E-04	1.66E-04	2.559	0.0105	*
DepDelay	9.46E-01	2.06E-04	4601.815	<2.00E-16	***

TaxiIn	6.68E-01	1.94E-03	343.746	<2.00E-16	***
TaxiOut	7.05E-01	7.93E-04	889.225	<2.00E-16	***
WeatherDelay	5.38E-02	5.44E-04	98.775	<2.00E-16	***
NASDelay	1.29E-01	3.95E-04	325.368	<2.00E-16	***
SecurityDelay	3.77E-02	5.45E-03	6.923	<4.43E-12	***

Prediction versus actual on test data

I had reserved 20% of the data for testing and when I plot the predicted ArrDelay against actual ArrDelay, we do see that predictions follow the 45% degree line. The root mean squared error on test data is 0.22.

Appendix

```
library(DAAG)
library(lattice)

#Loading the csv file
dfAll <- read.csv("DelayedFlights.csv", na.strings=c("","  ","NA"))
dfAll <- na.omit(dfAll)
#Sampling for plotting purposes
df1000 <- dfAll[sample(nrow(dfAll), 1000), ]
df <- dfAll[sample(nrow(dfAll), 300), ]
summary(df)

#changing time in hhmm format to minutes
df$DepTime <- floor(df$DepTime/100)*60 + (df$DepTime%%100)
df$CRSDepTime <- floor(df$CRSDepTime/100)*60+ (df$CRSDepTime%%100)
df$ArrTime = floor(df$ArrTime/100)*60+ (df$ArrTime%%100)
df$CRSArrTime = floor(df$CRSArrTime/100)*60+ (df$CRSArrTime%%100)
df$Month <- as.factor(df$Month)
df$DayOfWeek <- as.factor(df$DayOfWeek)
df$UniqueCarrier <- as.factor(df$UniqueCarrier)
df$Origin <- as.factor(df$Origin)
df$Dest <- as.factor(df$Dest)
dim(df)

boxplot(df$ArrDelay ~ df$DayOfWeek, data=df, horizontal = TRUE, col = "turquoise1",
```

```
          main= "Comparing Arrival Delay and Day of the week",
          xlab="Arrival Delay (in minutes)",
          ylab="DayOfWeek (1=Monday, ..., 7=Sunday")

aggregate(dfAll[, "DepDelay"], list(dfAll$DayOfWeek), mean)

boxplot(df$ArrDelay ~ df$Month, data=df, horizontal = TRUE, col = "palevioletred2",
          main= "Comparing Arrival Delay and Month",
          xlab="Arrival Delay (in minutes)",
          ylab="Month")

boxplot(df$ArrDelay ~ df$UniqueCarrier, data=df, horizontal = TRUE, col = "maroon4",
           main= "Comparing Arrival Delay and Unique Carrier",
          xlab="Arrival Delay (in minutes)",
          ylab="UniqueCarrier")

plot(df$DepDelay ~ df$TaxiOut)

aggregate(df[, "DepDelay"], list(df$Month), mean)

df$Month <- as.factor(df$Month)
summary(df$Month)

pairs(df[,c("ArrDelay", "DepDelay", "TaxiIn", "TaxiOut",
          "AirTime", "Distance","CarrierDelay","WeatherDelay",
          "NASDelay", "LateAircraftDelay")])

df$Month <- as.numeric(df$Month)
df$DayOfWeek <- as.numeric(df$DayOfWeek)
##Create a scatterplot
plot(DepDelay ~ Month, data = df, pch = 15, col = "palevioletred2")
panel.smooth(df$DepDelay, df$Month)
plot(df, main = " Dep Delay as per Month")
abline(lm(DepDelay ~ Month, data = df))
panel.smooth(df$DepDelay, df$Month)

dfAll$DepTime <- floor(dfAll$DepTime/100)*60 + (dfAll$DepTime%%100)
dfAll$CRSDepTime <- floor(dfAll$CRSDepTime/100)*60+ (dfAll$CRSDepTime%%100)
dfAll$ArrTime = floor(dfAll$ArrTime/100)*60+ (dfAll$ArrTime%%100)
dfAll$CRSArrTime = floor(dfAll$CRSArrTime/100)*60+ (dfAll$CRSArrTime%%100)
summary(dfAll[,c("Month", "DayOfWeek", "DepTime", "CRSDepTime",
          "ArrTime", "CRSArrTime", "UniqueCarrier",
          "ActualElapsedTime", "CRSElapsedTime",
          "ArrDelay", "DepDelay", "TaxiIn", "TaxiOut",
          "AirTime", "Distance","CarrierDelay","WeatherDelay",
          "NASDelay", "SecurityDelay", "LateAircraftDelay")])

model1 <- lm(ArrDelay ~ DepDelay, data = df1000)
plot(ArrDelay ~ DepDelay, data = df1000, pch = 17,col = "maroon")
abline(model1)
```

```
model2 <- lm(ArrDelay ~ Distance, data = df1000)
plot(ArrDelay ~ Distance, data = df1000, pch = 17,col = "maroon")
abline(model2)

##linear regression model
model3 <- lm(ArrDelay ~ Distance, data = df1000)
model3
##
cook <- cooks.distance(model3)
outliers <- cook[cook > 0.02]
plot(cook, ylab = "Cook's Distance")

##
plot(ArrDelay ~ Distance, data = df1000)
abline(model3, col="green")

model4 <- lm(ArrDelay ~ Distance, data = df1000[-outliers, ])
abline(model4, lty = 2, col="red")
#
plot(ArrDelay ~ Distance, data = df1000)
abline(model3)
abline(model4, lty = 2, col="red")

##Testing and Training data set
df1000$DayOfWeek <-  as.factor(df1000$DayOfWeek)
df1000$Month <-  as.factor(df1000$Month)
training.rows <-sample(1:nrow(df1000), size = 800)

df1000.train <- df1000[training.rows, ]
df1000.test <- df1000[-training.rows, ]
#Fit a linear regression model to predict  using all of the other variables on the training data.


dfAll$DayOfWeek <-  as.factor(dfAll$DayOfWeek)
dfAll$Month <-  as.factor(dfAll$Month)
training.rows <-sample(1:nrow(dfAll), size = floor(nrow(dfAll)*0.8))

dfAll.train <- dfAll[training.rows, ]
dfAll.test <- dfAll[-training.rows, ]
#Fit a linear regression model to predict  using all of the other variables on the training data.
dfAll.model1 <- lm(ArrDelay ~
                        Month+ DayOfWeek+ DepTime+
                        ArrTime+
                        ActualElapsedTime+
                        DepDelay+ TaxiIn+ TaxiOut+
                        WeatherDelay+
                NASDelay+ SecurityDelay,
                        data = dfAll.train)
summary(dfAll.model1)
```

```
dfAll.predict.train <- predict(dfAll.model1, newdata = dfAll.train)
sqrt(mean(dfAll.predict.train - dfAll.train$ArrDelay))

dfAll.predict.test <- predict(dfAll.model1, newdata = dfAll.test)
sqrt(mean(dfAll.predict.test - dfAll.test$ArrDelay))

keeps <- c("Month", "DayOfWeek", "DepTime", "CRSDepTime",
                        "ArrTime", "CRSArrTime",
           "ActualElapsedTime", "CRSElapsedTime",
                "DepDelay", "TaxiIn", "TaxiOut",
           "Distance", "CarrierDelay", "WeatherDelay",
           "NASDelay", "SecurityDelay")
```