Authors:
Eric Shields (ESS190007)
Abigail Smith (ARS190011)

Regression

02/18/2023

Assignment: Linear Models

Reading and Cleansing Data

```
# Reading in DelayedFlights.csv, filling in all NA with "NA", and retaining
column names
df <- read.csv("C:/Users/83627/Downloads/DelayedFlights.csv",
na.strings="NA", header=TRUE)  # Please update the file path for
DelayedFlights.csv

head(df)          # Exploring top observations

##   X Year Month DayofMonth DayOfWeek DepTime CRSDepTime ArrTime CRSArrTime
## 1 0 2008     1          3         4    2003       1955    2211       2225
## 2 1 2008     1          3         4     754        735    1002       1000
## 3 2 2008     1          3         4     628        620     804        750
## 4 4 2008     1          3         4    1829       1755    1959       1925
## 5 5 2008     1          3         4    1940       1915    2121       2110
## 6 6 2008     1          3         4    1937       1830    2037       1940
##   UniqueCarrier FlightNum TailNum ActualElapsedTime CRSElapsedTime AirTime
## 1            WN       335  N712SW               128            150     116
## 2            WN      3231  N772SW               128            145     113
## 3            WN       448  N428WN                96             90      76
## 4            WN      3920  N464WN                90             90      77
## 5            WN       378  N726SW               101            115      87
## 6            WN       509  N763SW               240            250     230
##   ArrDelay DepDelay Origin Dest Distance TaxiIn TaxiOut Cancelled
## 1      -14        8    IAD  TPA      810      4       8         0
## 2        2       19    IAD  TPA      810      5      10         0
## 3       14        8    IND  BWI      515      3      17         0
## 4       34       34    IND  BWI      515      3      10         0
## 5       11       25    IND  JAX      688      4      10         0
## 6       57       67    IND  LAS     1591      3       7         0
##   CancellationCode Diverted CarrierDelay WeatherDelay NASDelay
SecurityDelay
## 1                N        0           NA           NA       NA
NA
## 2                N        0           NA           NA       NA
NA
## 3                N        0           NA           NA       NA
NA
## 4                N        0            2            0        0
0
```

```
## 5                N        0           NA           NA       NA
NA
## 6                N        0           10            0        0
0
##   LateAircraftDelay
## 1                NA
## 2                NA
## 3                NA
## 4                32
## 5                NA
## 6                47
```

```
str(df)            # Exploring the structure of the data frame
```

```
## 'data.frame':    1936758 obs. of  30 variables:
##  $ X                : int  0 1 2 4 5 6 10 11 15 16 ...
##  $ Year             : int  2008 2008 2008 2008 2008 2008 2008 2008 2008
2008 ...
##  $ Month            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DayofMonth       : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ DayOfWeek        : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ DepTime          : num  2003 754 628 1829 1940 ...
##  $ CRSDepTime       : int  1955 735 620 1755 1915 1830 700 1510 1020 1425
...
##  $ ArrTime          : num  2211 1002 804 1959 2121 ...
##  $ CRSArrTime       : int  2225 1000 750 1925 2110 1940 915 1725 1010 1625
...
##  $ UniqueCarrier    : chr  "WN" "WN" "WN" "WN" ...
##  $ FlightNum        : int  335 3231 448 3920 378 509 100 1333 2272 675 ...
##  $ TailNum          : chr  "N712SW" "N772SW" "N428WN" "N464WN" ...
##  $ ActualElapsedTime: num  128 128 96 90 101 240 130 121 52 228 ...
##  $ CRSElapsedTime   : num  150 145 90 90 115 250 135 135 50 240 ...
##  $ AirTime          : num  116 113 76 77 87 230 106 107 37 213 ...
##  $ ArrDelay         : num  -14 2 14 34 11 57 1 80 11 15 ...
##  $ DepDelay         : num  8 19 8 34 25 67 6 94 9 27 ...
##  $ Origin           : chr  "IAD" "IAD" "IND" "IND" ...
##  $ Dest             : chr  "TPA" "TPA" "BWI" "BWI" ...
##  $ Distance         : int  810 810 515 515 688 1591 828 828 162 1489 ...
##  $ TaxiIn           : num  4 5 3 3 4 3 5 6 6 7 ...
##  $ TaxiOut          : num  8 10 17 10 10 7 19 8 9 8 ...
##  $ Cancelled        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CancellationCode : chr  "N" "N" "N" "N" ...
##  $ Diverted         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CarrierDelay     : num  NA NA NA 2 NA 10 NA 8 NA 3 ...
##  $ WeatherDelay     : num  NA NA NA 0 NA 0 NA 0 NA 0 ...
##  $ NASDelay         : num  NA NA NA 0 NA 0 NA 0 NA 0 ...
##  $ SecurityDelay    : num  NA NA NA 0 NA 0 NA 0 NA 0 ...
##  $ LateAircraftDelay: num  NA NA NA 32 NA 47 NA 72 NA 12 ...
```

```
print("-----")     # Visual break between outputs
```

```
## [1] "-----"
```

```
summary(df)       # Exploring the summary of the data frame (min, max, mean,
etc.)
```

```
##        X                Year          Month          DayofMonth
##  Min.   :      0   Min.   :2008   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:1517452   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :3242558   Median :2008   Median : 6.000   Median :16.00
##  Mean   :3341651   Mean   :2008   Mean   : 6.111   Mean   :15.75
##  3rd Qu.:4972467   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00
##
##    DayOfWeek        DepTime       CRSDepTime       ArrTime       CRSArrTime
##  Min.   :1.000   Min.   :   1   Min.   :   0   Min.   :   1   Min.   :   0
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1316   1st Qu.:1325
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705
##  Mean   :3.985   Mean   :1519   Mean   :1467   Mean   :1610   Mean   :1634
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2400
##                                                NA's   :7110
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1936758     Min.   :   1   Length:1936758     Min.   :  14.0
##  Class :character   1st Qu.: 610   Class :character   1st Qu.:  80.0
##  Mode  :character   Median :1543   Mode  :character   Median : 116.0
##                     Mean   :2184                      Mean   : 133.3
##                     3rd Qu.:3422                      3rd Qu.: 165.0
##                     Max.   :9742                      Max.   :1114.0
##                                                       NA's   :8387
##  CRSElapsedTime     AirTime          ArrDelay          DepDelay
##  Min.   :-25.0   Min.   :   0.0   Min.   :-109.0   Min.   :   6.00
##  1st Qu.: 82.0   1st Qu.:  58.0   1st Qu.:   9.0   1st Qu.:  12.00
##  Median :116.0   Median :  90.0   Median :  24.0   Median :  24.00
##  Mean   :134.3   Mean   : 108.3   Mean   :  42.2   Mean   :  43.19
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:  56.0   3rd Qu.:  53.00
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.0   Max.   :2467.00
##  NA's   :198     NA's   :8387     NA's   :8387
##     Origin              Dest              Distance          TaxiIn
##  Length:1936758     Length:1936758     Min.   :  11.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 765.7   Mean   :  6.813
##                                        3rd Qu.: 998.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##                                                         NA's   :7110
##     TaxiOut         Cancelled          CancellationCode      Diverted
##  Min.   :  0.00   Min.   :0.0000000   Length:1936758     Min.   :0.000000
##  1st Qu.: 10.00   1st Qu.:0.0000000   Class :character   1st Qu.:0.000000
##  Median : 14.00   Median :0.0000000   Mode  :character   Median :0.000000
##  Mean   : 18.23   Mean   :0.0003268                      Mean   :0.004004
```

```
##  3rd Qu.: 21.00   3rd Qu.:0.0000000                            3rd Qu.:0.000000
##  Max.   :422.00   Max.   :1.0000000                            Max.   :1.000000
##  NA's   :455
##   CarrierDelay     WeatherDelay       NASDelay      SecurityDelay
##  Min.   :   0.0   Min.   :   0.0   Min.   :   0     Min.   :  0.0
##  1st Qu.:   0.0   1st Qu.:   0.0   1st Qu.:   0     1st Qu.:  0.0
##  Median :   2.0   Median :   0.0   Median :   2     Median :  0.0
##  Mean   :  19.2   Mean   :   3.7   Mean   :  15     Mean   :  0.1
##  3rd Qu.:  21.0   3rd Qu.:   0.0   3rd Qu.:  15     3rd Qu.:  0.0
##  Max.   :2436.0   Max.   :1352.0   Max.   :1357     Max.   :392.0
##  NA's   :689270   NA's   :689270   NA's   :689270   NA's   :689270
##  LateAircraftDelay
##  Min.   :   0.0
##  1st Qu.:   0.0
##  Median :   8.0
##  Mean   :  25.3
##  3rd Qu.:  33.0
##  Max.   :1316.0
##  NA's   :689270
```

The data frame holding the data from DelayedFlights.csv has a variety of characteristics that are of interest.

Majority of the columns from the data frame are quantitative with only a few columns being qualitative (UniqueCarrier, TailNum, Origin, Dest, and CancellationCode). For linear regression, we are interested in the quantitative columns.

Before starting linear regression, the NA entries in the data frame must be addressed. In the following section, the NA entries will be reviewed and accounted for.

Exploring NA Entries

From the summary above, we can see there are multiple columns with NA entries.

First, columns CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, and LateAircraftDelay will be reviewed since each of these columns have the same number of NA entries (689270).

```
print(paste("Count of all entries where flight was cancelled (i.e. column
Cancelled == 1): ",sum(df$Cancelled == 1)))

## [1] "Count of all entries where flight was cancelled (i.e. column
Cancelled == 1):  633"

print(paste("Count of all observations with NA in CarrierDelay and Cancelled
== 1: ", sum(is.na(df$CarrierDelay) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in CarrierDelay and Cancelled == 1:
633"
```

```r
print(paste("Count of all observations with NA in WeatherDelay and Cancelled
== 1: ",
sum(is.na(df$WeatherDelay) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in WeatherDelay and Cancelled == 1:
633"

print(paste("Count of all observations with NA in NASDelay and Cancelled ==
1: ",
sum(is.na(df$NASDelay) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in NASDelay and Cancelled == 1:
633"

print(paste("Count of all observations with NA in SecurityDelay and Cancelled
== 1: ",
sum(is.na(df$SecurityDelay ) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in SecurityDelay and Cancelled ==
1:  633"

print(paste("Count of all observations with NA in LateAircraftDelay and
Cancelled == 1: ", sum(is.na(df$LateAircraftDelay) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in LateAircraftDelay and Cancelled
== 1:  633"
```

As seen above, in all instances where a flight was cancelled, there is an NA in CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, and LateAircraftDelay.

To fix these NAs, we will replace these NAs with 0. This is because columns CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, and LateAircraftDelay are measurements of time, and when a flight is cancelled, it is reasonable to assert that there was not a delay on the flight.

```r
# Replacing all instances where the flight was cancelled and there is an NA
for each of the delay columns with 0.
df[df$Cancelled == 1 & is.na(df$CarrierDelay),]$CarrierDelay <- 0
df[df$Cancelled == 1 & is.na(df$WeatherDelay),]$WeatherDelay <- 0
df[df$Cancelled == 1 & is.na(df$NASDelay),]$NASDelay <- 0
df[df$Cancelled == 1 & is.na(df$SecurityDelay),]$SecurityDelay <- 0
df[df$Cancelled == 1 & is.na(df$LateAircraftDelay),]$LateAircraftDelay <- 0
```

We can see the replacement was successful:

```r
print(paste("Count of all entries where flight was cancelled (i.e. column
Cancelled == 1): ",sum(df$Cancelled == 1)))

## [1] "Count of all entries where flight was cancelled (i.e. column
Cancelled == 1):  633"

print(paste("Count of all observations with NA in CarrierDelay and Cancelled
== 1: ", sum(is.na(df$CarrierDelay) & df$Cancelled == 1)))
```

```
## [1] "Count of all observations with NA in CarrierDelay and Cancelled == 1:  
0"

print(paste("Count of all observations with NA in WeatherDelay and Cancelled 
== 1: ",
sum(is.na(df$WeatherDelay) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in WeatherDelay and Cancelled == 1:  
0"

print(paste("Count of all observations with NA in NASDelay and Cancelled == 
1: ",
sum(is.na(df$NASDelay) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in NASDelay and Cancelled == 1:  0"

print(paste("Count of all observations with NA in SecurityDelay and Cancelled 
== 1: ",
sum(is.na(df$SecurityDelay ) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in SecurityDelay and Cancelled == 
1:  0"

print(paste("Count of all observations with NA in LateAircraftDelay and 
Cancelled == 1: ", sum(is.na(df$LateAircraftDelay) & df$Cancelled == 1)))

## [1] "Count of all observations with NA in LateAircraftDelay and Cancelled 
== 1:  0"

summary(df)

##        X                Year          Month          DayofMonth   
##  Min.   :      0   Min.   :2008   Min.   : 1.000   Min.   : 1.00  
##  1st Qu.:1517452   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00  
##  Median :3242558   Median :2008   Median : 6.000   Median :16.00  
##  Mean   :3341651   Mean   :2008   Mean   : 6.111   Mean   :15.75  
##  3rd Qu.:4972467   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00  
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00  
##                                                                   
##    DayOfWeek        DepTime       CRSDepTime      ArrTime       CRSArrTime  
##  Min.   :1.000   Min.   :   1   Min.   :   0   Min.   :   1   Min.   :   0  
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1316   1st Qu.:1325  
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705  
##  Mean   :3.985   Mean   :1519   Mean   :1467   Mean   :1610   Mean   :1634  
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014  
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2400  
##                                                NA's   :7110                 
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1936758     Min.   :   1   Length:1936758     Min.   :  14.0   
##  Class :character   1st Qu.: 610   Class :character   1st Qu.:  80.0   
##  Mode  :character   Median :1543   Mode  :character   Median : 116.0   
##                     Mean   :2184                      Mean   : 133.3   
```

```
##                      3rd Qu.:3422                              3rd Qu.: 165.0
##                      Max.   :9742                              Max.   :1114.0
##                                                                NA's   :8387
##  CRSElapsedTime     AirTime          ArrDelay          DepDelay
##  Min.   :-25.0   Min.   :   0.0   Min.   :-109.0   Min.   :   6.00
##  1st Qu.: 82.0   1st Qu.:  58.0   1st Qu.:   9.0   1st Qu.:  12.00
##  Median :116.0   Median :  90.0   Median :  24.0   Median :  24.00
##  Mean   :134.3   Mean   : 108.3   Mean   :  42.2   Mean   :  43.19
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:  56.0   3rd Qu.:  53.00
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.0   Max.   :2467.00
##  NA's   :198     NA's   :8387     NA's   :8387
##     Origin              Dest              Distance          TaxiIn
##  Length:1936758     Length:1936758     Min.   :  11.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 765.7   Mean   :  6.813
##                                        3rd Qu.: 998.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##                                                         NA's   :7110
##     TaxiOut         Cancelled         CancellationCode      Diverted
##  Min.   :  0.00   Min.   :0.0000000   Length:1936758     Min.   :0.000000
##  1st Qu.: 10.00   1st Qu.:0.0000000   Class :character   1st Qu.:0.000000
##  Median : 14.00   Median :0.0000000   Mode  :character   Median :0.000000
##  Mean   : 18.23   Mean   :0.0003268                      Mean   :0.004004
##  3rd Qu.: 21.00   3rd Qu.:0.0000000                      3rd Qu.:0.000000
##  Max.   :422.00   Max.   :1.0000000                      Max.   :1.000000
##  NA's   :455
##   CarrierDelay     WeatherDelay       NASDelay      SecurityDelay
##  Min.   :   0.0   Min.   :   0.0   Min.   :   0     Min.   :  0.0
##  1st Qu.:   0.0   1st Qu.:   0.0   1st Qu.:   0     1st Qu.:  0.0
##  Median :   2.0   Median :   0.0   Median :   2     Median :  0.0
##  Mean   :  19.2   Mean   :   3.7   Mean   :  15     Mean   :  0.1
##  3rd Qu.:  21.0   3rd Qu.:   0.0   3rd Qu.:  15     3rd Qu.:  0.0
##  Max.   :2436.0   Max.   :1352.0   Max.   :1357     Max.   :392.0
##  NA's   :688637   NA's   :688637   NA's   :688637   NA's   :688637
##  LateAircraftDelay
##  Min.   :   0.0
##  1st Qu.:   0.0
##  Median :   8.0
##  Mean   :  25.3
##  3rd Qu.:  33.0
##  Max.   :1316.0
##  NA's   :688637
```

From the summary above, we can see there are still NA entries.

The columns CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, and LateAircraftDelay will continue to be reviewed since they still have the same number of NA entries remaining (688637).

```r
print(paste("Count of all entries where CarrierDelay is NA: ",
sum(is.na(df$CarrierDelay))))

## [1] "Count of all entries where CarrierDelay is NA:  688637"

print(paste("Count of all entries where WeatherDelay is NA: ",
sum(is.na(df$WeatherDelay))))

## [1] "Count of all entries where WeatherDelay is NA:  688637"

print(paste("Count of all entries where NASDelay is NA: ",
sum(is.na(df$NASDelay))))

## [1] "Count of all entries where NASDelay is NA:  688637"

print(paste("Count of all entries where SecurityDelay is NA: ",
sum(is.na(df$SecurityDelay))))

## [1] "Count of all entries where SecurityDelay is NA:  688637"

print(paste("Count of all entries where LateAircraftDelay  is NA: ",
sum(is.na(df$LateAircraftDelay))))

## [1] "Count of all entries where LateAircraftDelay  is NA:  688637"

## Showing observations where CarrierDelay is NA and another observation when
CarrierDelay is not NA
#df[is.na(df$CarrierDelay),]
#df[!is.na(df$CarrierDelay),]

## Showing the sum of all observations where each of the five columns are NA
print("Count of all observations where CarrierDelay, NASDelay, WeatherDelay,
SecurityDelay, and LateAircraftDelay are EACH NA: ")

## [1] "Count of all observations where CarrierDelay, NASDelay, WeatherDelay,
SecurityDelay, and LateAircraftDelay are EACH NA: "

print(sum(is.na(df$CarrierDelay & is.na(df$NASDelay) & is.na(df$WeatherDelay)
& is.na(df$SecurityDelay)) & is.na(df$LateAircraftDelay)))

## [1] 688637
```

From above, we can see that whenever there is an NA in any of the five columns – CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, or LateAircraftDelay – there is an NA in the remaining four columns. This is understandable since if there was not a delay on a flight, none of the five columns would need to be filled in.

To fix this, we will replace all NA entries for the five columns with 0, meaning there was not a delay on any of the flights.

```r
df[is.na(df$CarrierDelay) & is.na(df$NASDelay) & is.na(df$WeatherDelay) &
is.na(df$SecurityDelay) & is.na(df$LateAircraftDelay),]$CarrierDelay <- 0
df[is.na(df$NASDelay) & is.na(df$WeatherDelay) & is.na(df$SecurityDelay) &
```

```
is.na(df$LateAircraftDelay),]$NASDelay <- 0
df[is.na(df$WeatherDelay) & is.na(df$SecurityDelay) &
is.na(df$LateAircraftDelay),]$WeatherDelay <- 0
df[is.na(df$SecurityDelay) & is.na(df$LateAircraftDelay),]$SecurityDelay <- 0
df[is.na(df$LateAircraftDelay),]$LateAircraftDelay <- 0
```

We can see the replacement was successful:

```
print(paste("Count of all entries where CarrierDelay is NA: ",
sum(is.na(df$CarrierDelay))))

## [1] "Count of all entries where CarrierDelay is NA:  0"

print(paste("Count of all entries where WeatherDelay is NA: ",
sum(is.na(df$WeatherDelay))))

## [1] "Count of all entries where WeatherDelay is NA:  0"

print(paste("Count of all entries where NASDelay is NA: ",
sum(is.na(df$NASDelay))))

## [1] "Count of all entries where NASDelay is NA:  0"

print(paste("Count of all entries where SecurityDelay is NA: ",
sum(is.na(df$SecurityDelay))))

## [1] "Count of all entries where SecurityDelay is NA:  0"

print(paste("Count of all entries where LateAircraftDelay  is NA: ",
sum(is.na(df$LateAircraftDelay))))

## [1] "Count of all entries where LateAircraftDelay  is NA:  0"

print(paste("Count of all observations where CarrierDelay, NASDelay,
WeatherDelay, SecurityDelay, and LateAircraftDelay are EACH NA: ",
sum(is.na(df$CarrierDelay & is.na(df$NASDelay) & is.na(df$WeatherDelay) &
is.na(df$SecurityDelay)) & is.na(df$LateAircraftDelay))))

## [1] "Count of all observations where CarrierDelay, NASDelay, WeatherDelay,
SecurityDelay, and LateAircraftDelay are EACH NA:  0"

summary(df)

##        X                Year          Month          DayofMonth
##  Min.   :      0   Min.   :2008   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:1517452   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :3242558   Median :2008   Median : 6.000   Median :16.00
##  Mean   :3341651   Mean   :2008   Mean   : 6.111   Mean   :15.75
##  3rd Qu.:4972467   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00
##
##    DayOfWeek        DepTime       CRSDepTime       ArrTime        CRSArrTime
##  Min.   :1.000   Min.   :   1   Min.   :   0   Min.   :   1   Min.   :   0
```

```
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1316   1st Qu.:1325
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705
##  Mean   :3.985   Mean   :1519   Mean   :1467   Mean   :1610   Mean   :1634
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2400
##                                                NA's   :7110
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1936758     Min.   :   1   Length:1936758     Min.   :  14.0
##  Class :character   1st Qu.: 610   Class :character   1st Qu.:  80.0
##  Mode  :character   Median :1543   Mode  :character   Median : 116.0
##                     Mean   :2184                      Mean   : 133.3
##                     3rd Qu.:3422                      3rd Qu.: 165.0
##                     Max.   :9742                      Max.   :1114.0
##                                                       NA's   :8387
##  CRSElapsedTime     AirTime          ArrDelay         DepDelay
##  Min.   :-25.0   Min.   :   0.0   Min.   :-109.0   Min.   :   6.00
##  1st Qu.: 82.0   1st Qu.:  58.0   1st Qu.:   9.0   1st Qu.:  12.00
##  Median :116.0   Median :  90.0   Median :  24.0   Median :  24.00
##  Mean   :134.3   Mean   : 108.3   Mean   :  42.2   Mean   :  43.19
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:  56.0   3rd Qu.:  53.00
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.0   Max.   :2467.00
##  NA's   :198     NA's   :8387     NA's   :8387
##     Origin              Dest              Distance          TaxiIn
##  Length:1936758     Length:1936758     Min.   :  11.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 765.7   Mean   :  6.813
##                                        3rd Qu.: 998.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##                                                         NA's   :7110
##     TaxiOut         Cancelled          CancellationCode      Diverted
##  Min.   :  0.00   Min.   :0.0000000   Length:1936758     Min.   :0.000000
##  1st Qu.: 10.00   1st Qu.:0.0000000   Class :character   1st Qu.:0.000000
##  Median : 14.00   Median :0.0000000   Mode  :character   Median :0.000000
##  Mean   : 18.23   Mean   :0.0003268                      Mean   :0.004004
##  3rd Qu.: 21.00   3rd Qu.:0.0000000                      3rd Qu.:0.000000
##  Max.   :422.00   Max.   :1.0000000                      Max.   :1.000000
##  NA's   :455
##   CarrierDelay      WeatherDelay        NASDelay       SecurityDelay
##  Min.   :   0.00   Min.   :   0.000   Min.   :   0.000   Min.   :  0.0000
##  1st Qu.:   0.00   1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:  0.0000
##  Median :   0.00   Median :   0.000   Median :   0.000   Median :  0.0000
##  Mean   :  12.35   Mean   :   2.385   Mean   :   9.676   Mean   :  0.0581
##  3rd Qu.:  10.00   3rd Qu.:   0.000   3rd Qu.:   6.000   3rd Qu.:  0.0000
##  Max.   :2436.00   Max.   :1352.000   Max.   :1357.000   Max.   :392.0000
##
##  LateAircraftDelay
##  Min.   :   0.00
##  1st Qu.:   0.00
##  Median :   0.00
```

```
##  Mean   :  16.29
##  3rd Qu.:  18.00
##  Max.   :1316.00
##
```

We can see from the above summary that there are no longer NA entries in CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, and LateAircraftDelay.

The NA entries in columns AirTime and AirDelay will be reviewed next since they have the same number of NA entires (8387).

```
print(paste("Count of all observations where the flight was diverted
(Diverted == 1): ",
sum(df$Diverted == 1)))

## [1] "Count of all observations where the flight was diverted (Diverted ==
1):  7754"

print(paste("Count of all observations where AirTime is NA: ",
sum(is.na(df$AirTime))))

## [1] "Count of all observations where AirTime is NA:  8387"

print(paste("Count of all observations where ArrDelay is NA: ",
sum(is.na(df$ArrDelay))))

## [1] "Count of all observations where ArrDelay is NA:  8387"

print(paste("Count of all observations where the flight was diverted and
AirTime is NA: ",
sum(df$Diverted == 1 & is.na(df$AirTime))))

## [1] "Count of all observations where the flight was diverted and AirTime
is NA:  7754"

print(paste("Count of all observations where the flight was diverted and
ArrDelay is NA: ",
sum(df$Diverted == 1 & is.na(df$ArrDelay))))

## [1] "Count of all observations where the flight was diverted and ArrDelay
is NA:  7754"
```

From the above output, we can see that any time a flight was diverted, there are NA entries in AirTime and ArrDelay. This is reasonable, since if the flight was diverted, the flight time and arrival delay may not have been input.

To fix this, we will replace these NAs with 0 to mean there was not air time nor an arrival delay for delayed flights.

```
df[df$Diverted == 1 & is.na(df$AirTime) & is.na(df$ArrDelay),]$AirTime <- 0
df[df$Diverted == 1 & is.na(df$ArrDelay),]$ArrDelay <- 0
```

We can see the replacement was successful:

```
print(paste("Count of all observations where the flight was diverted
(Diverted == 1): ",
sum(df$Diverted == 1)))

## [1] "Count of all observations where the flight was diverted (Diverted ==
1):  7754"

print(paste("Count of all observations where AirTime is NA: ",
sum(is.na(df$AirTime))))

## [1] "Count of all observations where AirTime is NA:  633"

print(paste("Count of all observations where ArrDelay is NA: ",
sum(is.na(df$ArrDelay))))

## [1] "Count of all observations where ArrDelay is NA:  633"

print(paste("Count of all observations where the flight was diverted and
AirTime is NA: ",
sum(df$Diverted == 1 & is.na(df$AirTime))))

## [1] "Count of all observations where the flight was diverted and AirTime
is NA:  0"

print(paste("Count of all observations where the flight was diverted and
ArrDelay is NA: ",
sum(df$Diverted == 1 & is.na(df$ArrDelay))))

## [1] "Count of all observations where the flight was diverted and ArrDelay
is NA:  0"

summary(df)

##        X                 Year          Month          DayofMonth
##  Min.   :      0   Min.   :2008   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:1517452   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :3242558   Median :2008   Median : 6.000   Median :16.00
##  Mean   :3341651   Mean   :2008   Mean   : 6.111   Mean   :15.75
##  3rd Qu.:4972467   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00
##
##    DayOfWeek        DepTime        CRSDepTime      ArrTime       CRSArrTime
##  Min.   :1.000   Min.   :   1   Min.   :   0   Min.   :   1   Min.   :   0
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1316   1st Qu.:1325
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705
##  Mean   :3.985   Mean   :1519   Mean   :1467   Mean   :1610   Mean   :1634
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2400
##                                                NA's   :7110
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1936758     Min.   :   1   Length:1936758     Min.   :  14.0
##  Class :character   1st Qu.: 610   Class :character   1st Qu.:  80.0
```

```
##  Mode  :character   Median :1543   Mode  :character   Median : 116.0
##                     Mean   :2184                     Mean   : 133.3
##                     3rd Qu.:3422                     3rd Qu.: 165.0
##                     Max.   :9742                     Max.   :1114.0
##                                                      NA's   :8387
##  CRSElapsedTime     AirTime          ArrDelay          DepDelay
##  Min.   :-25.0   Min.   :   0.0   Min.   :-109.00   Min.   :   6.00
##  1st Qu.: 82.0   1st Qu.:  58.0   1st Qu.:   9.00   1st Qu.:  12.00
##  Median :116.0   Median :  90.0   Median :  24.00   Median :  24.00
##  Mean   :134.3   Mean   : 107.8   Mean   :  42.03   Mean   :  43.19
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:  56.00   3rd Qu.:  53.00
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.00   Max.   :2467.00
##  NA's   :198     NA's   :633      NA's   :633
##     Origin              Dest              Distance          TaxiIn
##  Length:1936758     Length:1936758     Min.   :  11.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 765.7   Mean   :  6.813
##                                        3rd Qu.: 998.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##                                                         NA's   :7110
##     TaxiOut         Cancelled         CancellationCode      Diverted
##  Min.   :  0.00   Min.   :0.0000000   Length:1936758     Min.   :0.000000
##  1st Qu.: 10.00   1st Qu.:0.0000000   Class :character   1st Qu.:0.000000
##  Median : 14.00   Median :0.0000000   Mode  :character   Median :0.000000
##  Mean   : 18.23   Mean   :0.0003268                      Mean   :0.004004
##  3rd Qu.: 21.00   3rd Qu.:0.0000000                      3rd Qu.:0.000000
##  Max.   :422.00   Max.   :1.0000000                      Max.   :1.000000
##  NA's   :455
##   CarrierDelay      WeatherDelay        NASDelay        SecurityDelay
##  Min.   :   0.00   Min.   :   0.000   Min.   :   0.000   Min.   :  0.0000
##  1st Qu.:   0.00   1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:  0.0000
##  Median :   0.00   Median :   0.000   Median :   0.000   Median :  0.0000
##  Mean   :  12.35   Mean   :   2.385   Mean   :   9.676   Mean   :  0.0581
##  3rd Qu.:  10.00   3rd Qu.:   0.000   3rd Qu.:   6.000   3rd Qu.:  0.0000
##  Max.   :2436.00   Max.   :1352.000   Max.   :1357.000   Max.   :392.0000
##
##  LateAircraftDelay
##  Min.   :   0.00
##  1st Qu.:   0.00
##  Median :   0.00
##  Mean   :  16.29
##  3rd Qu.:  18.00
##  Max.   :1316.00
##
```

We can see there are still NA entries in AirTime and ArrDelay.

```
print(paste("Count of NA entries in AirTime: ",
sum(is.na(df$AirTime))))
```

```
## [1] "Count of NA entries in AirTime:  633"

print(paste("Count of NA entries in ArrDelay: ",
sum(is.na(df$ArrDelay))))

## [1] "Count of NA entries in ArrDelay:  633"

print(paste("Count of flights Cancelled: ",
sum(df$Cancelled == 1)))

## [1] "Count of flights Cancelled:  633"

print(paste("Count of NA entries in AirTime and ArrDelay when the flight was
cancelled: ",
sum(is.na(df$AirTime) & is.na(df$ArrDelay) & df$Cancelled == 1)))

## [1] "Count of NA entries in AirTime and ArrDelay when the flight was
cancelled:  633"
```

From the above, we can see that any time a flight was cancelled, there is an NA entry for AirTime and ArrDelay.

To fix this, we will replace these NAs with 0 to represent that there was not air time nor an arrival delay for cancelled flights.

```
df[df$Cancelled == 1 & is.na(df$AirTime) & is.na(df$ArrDelay),]$AirTime <- 0
df[df$Cancelled == 1 & is.na(df$ArrDelay),]$ArrDelay <- 0
```

We can see the replacement was successful:

```
print(paste("Count of NA entries in AirTime: ",
sum(is.na(df$AirTime))))

## [1] "Count of NA entries in AirTime:  0"

print(paste("Count of NA entries in ArrDelay: ",
sum(is.na(df$ArrDelay))))

## [1] "Count of NA entries in ArrDelay:  0"

print(paste("Count of flights Cancelled: ",
sum(df$Cancelled == 1)))

## [1] "Count of flights Cancelled:  633"

print(paste("Count of NA entries in AirTime and ArrDelay when the flight was
cancelled: ",
sum(is.na(df$AirTime) & is.na(df$ArrDelay) & df$Cancelled == 1)))

## [1] "Count of NA entries in AirTime and ArrDelay when the flight was
cancelled:  0"

summary(df)
```

```
##        X                 Year          Month          DayofMonth
##  Min.   :      0   Min.   :2008   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:1517452   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :3242558   Median :2008   Median : 6.000   Median :16.00
##  Mean   :3341651   Mean   :2008   Mean   : 6.111   Mean   :15.75
##  3rd Qu.:4972467   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00
##
##    DayOfWeek        DepTime       CRSDepTime      ArrTime       CRSArrTime
##  Min.   :1.000   Min.   :   1   Min.   :   0   Min.   :   1   Min.   :   0
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1316   1st Qu.:1325
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705
##  Mean   :3.985   Mean   :1519   Mean   :1467   Mean   :1610   Mean   :1634
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2400
##                                                NA's   :7110
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1936758     Min.   :   1   Length:1936758     Min.   :  14.0
##  Class :character   1st Qu.: 610   Class :character   1st Qu.:  80.0
##  Mode  :character   Median :1543   Mode  :character   Median : 116.0
##                     Mean   :2184                      Mean   : 133.3
##                     3rd Qu.:3422                      3rd Qu.: 165.0
##                     Max.   :9742                      Max.   :1114.0
##                                                       NA's   :8387
##  CRSElapsedTime     AirTime          ArrDelay          DepDelay
##  Min.   :-25.0   Min.   :   0.0   Min.   :-109.00   Min.   :   6.00
##  1st Qu.: 82.0   1st Qu.:  58.0   1st Qu.:   9.00   1st Qu.:  12.00
##  Median :116.0   Median :  90.0   Median :  24.00   Median :  24.00
##  Mean   :134.3   Mean   : 107.8   Mean   :  42.02   Mean   :  43.19
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:  55.00   3rd Qu.:  53.00
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.00   Max.   :2467.00
##  NA's   :198
##     Origin             Dest              Distance          TaxiIn
##  Length:1936758     Length:1936758     Min.   :  11.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 765.7   Mean   :  6.813
##                                        3rd Qu.: 998.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##                                                         NA's   :7110
##     TaxiOut         Cancelled         CancellationCode      Diverted
##  Min.   :  0.00   Min.   :0.0000000   Length:1936758     Min.   :0.000000
##  1st Qu.: 10.00   1st Qu.:0.0000000   Class :character   1st Qu.:0.000000
##  Median : 14.00   Median :0.0000000   Mode  :character   Median :0.000000
##  Mean   : 18.23   Mean   :0.0003268                      Mean   :0.004004
##  3rd Qu.: 21.00   3rd Qu.:0.0000000                      3rd Qu.:0.000000
##  Max.   :422.00   Max.   :1.0000000                      Max.   :1.000000
##  NA's   :455
##   CarrierDelay      WeatherDelay         NASDelay        SecurityDelay
##  Min.   :   0.00   Min.   :   0.000   Min.   :   0.000   Min.   :  0.0000
```

```
##  1st Qu.:   0.00   1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:  0.0000
##  Median :   0.00   Median :   0.000   Median :   0.000   Median :  0.0000
##  Mean   :  12.35   Mean   :   2.385   Mean   :   9.676   Mean   :  0.0581
##  3rd Qu.:  10.00   3rd Qu.:   0.000   3rd Qu.:   6.000   3rd Qu.:  0.0000
##  Max.   :2436.00   Max.   :1352.000   Max.   :1357.000   Max.   :392.0000
##
##  LateAircraftDelay
##  Min.   :   0.00
##  1st Qu.:   0.00
##  Median :   0.00
##  Mean   :  16.29
##  3rd Qu.:  18.00
##  Max.   :1316.00
##
```

We can see there are no longer NA entries in AirTime and ArrDelay.

The NA entries in columns ArrTime and TaxiIn will now be reviewed since they have the same number of NAs (7110)

```
print(paste("Count of NA entries in ArrTime: ",
sum(is.na(df$ArrTime))))

## [1] "Count of NA entries in ArrTime:  7110"

print(paste("Count of NA entries in TaxiIn: ",
sum(is.na(df$TaxiIn))))

## [1] "Count of NA entries in TaxiIn:  7110"

print(paste("Count of flights diverted: ",
sum(df$Diverted == 1)))

## [1] "Count of flights diverted:  7754"

print(paste("Count of flights cancelled: ",
sum(df$Cancelled == 1)))

## [1] "Count of flights cancelled:  633"

print(paste("Count of observations where both ArrTime and TaxiIn are NA: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn))))

## [1] "Count of observations where both ArrTime and TaxiIn are NA:  7110"

print(paste("Count of observations where both ArrTime and TaxiIn are NA and
the flight was diverted: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn) & df$Diverted == 1)))

## [1] "Count of observations where both ArrTime and TaxiIn are NA and the
flight was diverted:  6477"
```

```r
print(paste("Count of observations where both ArrTime and TaxiIn are NA and the flight was cancelled: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn) & df$Cancelled == 1)))

## [1] "Count of observations where both ArrTime and TaxiIn are NA and the flight was cancelled:  633"
```

We can see that in every instance where the flight was cancelled, ArrTime and TaxiIn are NA.

Since we cannot replace entries in ArrTime since ArrTime represents military time (and replacing with 0 would represent midnight), we will remove these rows.

```r
## Removing all rows where cancelled == 1 and ArrTime is NA
df <- df[!(df$Cancelled == 1 & is.na(df$ArrTime)),]
```

We can see the removal was successful:

```r
print(paste("Count of NA entries in ArrTime: ",
sum(is.na(df$ArrTime))))

## [1] "Count of NA entries in ArrTime:  6477"

print(paste("Count of NA entries in TaxiIn: ",
sum(is.na(df$TaxiIn))))

## [1] "Count of NA entries in TaxiIn:  6477"

print(paste("Count of flights diverted: ",
sum(df$Diverted == 1)))

## [1] "Count of flights diverted:  7754"

print(paste("Count of flights cancelled: ",
sum(df$Cancelled == 1)))

## [1] "Count of flights cancelled:  0"

print(paste("Count of observations where both ArrTime and TaxiIn are NA: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn))))

## [1] "Count of observations where both ArrTime and TaxiIn are NA:  6477"

print(paste("Count of observations where both ArrTime and TaxiIn are NA and the flight was diverted: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn) & df$Diverted == 1)))

## [1] "Count of observations where both ArrTime and TaxiIn are NA and the flight was diverted:  6477"

print(paste("Count of observations where both ArrTime and TaxiIn are NA and the flight was cancelled: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn) & df$Cancelled == 1)))
```

```
## [1] "Count of observations where both ArrTime and TaxiIn are NA and the
flight was cancelled:  0"
```

```
summary(df)
```

```
##        X                Year          Month          DayofMonth
##  Min.   :      0   Min.   :2008   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:1517049   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :3241778   Median :2008   Median : 6.000   Median :16.00
##  Mean   :3340592   Mean   :2008   Mean   : 6.109   Mean   :15.75
##  3rd Qu.:4969673   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00
##
##    DayOfWeek        DepTime       CRSDepTime      ArrTime       CRSArrTime
##  Min.   :1.000   Min.   :   1   Min.   :   0   Min.   :   1   Min.   :   0
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1316   1st Qu.:1325
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705
##  Mean   :3.985   Mean   :1519   Mean   :1467   Mean   :1610   Mean   :1634
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2400
##                                                NA's   :6477
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1936125     Min.   :   1   Length:1936125     Min.   :  14.0
##  Class :character   1st Qu.: 610   Class :character   1st Qu.:  80.0
##  Mode  :character   Median :1543   Mode  :character   Median : 116.0
##                     Mean   :2184                      Mean   : 133.3
##                     3rd Qu.:3422                      3rd Qu.: 165.0
##                     Max.   :9742                      Max.   :1114.0
##                                                       NA's   :7754
##  CRSElapsedTime     AirTime          ArrDelay          DepDelay
##  Min.   :-25.0   Min.   :   0.0   Min.   :-109.00   Min.   :   6.00
##  1st Qu.: 82.0   1st Qu.:  58.0   1st Qu.:   9.00   1st Qu.:  12.00
##  Median :116.0   Median :  90.0   Median :  24.00   Median :  24.00
##  Mean   :134.3   Mean   : 107.8   Mean   :  42.03   Mean   :  43.17
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:  56.00   3rd Qu.:  53.00
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.00   Max.   :2467.00
##  NA's   :198
##     Origin              Dest              Distance          TaxiIn
##  Length:1936125     Length:1936125     Min.   :  11.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 765.7   Mean   :  6.813
##                                        3rd Qu.: 998.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##                                                         NA's   :6477
##     TaxiOut         Cancelled CancellationCode     Diverted
##  Min.   :  0.00   Min.   :0   Length:1936125     Min.   :0.000000
##  1st Qu.: 10.00   1st Qu.:0   Class :character   1st Qu.:0.000000
##  Median : 14.00   Median :0   Mode  :character   Median :0.000000
##  Mean   : 18.23   Mean   :0                      Mean   :0.004005
```

```
##   3rd Qu.: 21.00   3rd Qu.:0                      3rd Qu.:0.000000
##   Max.   :422.00   Max.   :0                      Max.   :1.000000
##
##    CarrierDelay      WeatherDelay        NASDelay        SecurityDelay
##   Min.   :   0.00   Min.   :   0.000   Min.   :   0.000   Min.   :  0.0000
##   1st Qu.:   0.00   1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:  0.0000
##   Median :   0.00   Median :   0.000   Median :   0.000   Median :  0.0000
##   Mean   :  12.36   Mean   :   2.386   Mean   :   9.679   Mean   :  0.0581
##   3rd Qu.:  10.00   3rd Qu.:   0.000   3rd Qu.:   6.000   3rd Qu.:  0.0000
##   Max.   :2436.00   Max.   :1352.000   Max.   :1357.000   Max.   :392.0000
##
##   LateAircraftDelay
##   Min.   :   0.0
##   1st Qu.:   0.0
##   Median :   0.0
##   Mean   :  16.3
##   3rd Qu.:  18.0
##   Max.   :1316.0
##
```

We can see that whenever the flight was diverted, there is an NA in ArrTime and TaxiIn.

To fix this, we will remove these rows for the same reason as before (to replace ArrTime with a numeric value would create an inaccuracy).

```
df <- df[!(df$Diverted == 1 & is.na(df$ArrTime)),]
```

We can see the removal was successful:

```
print(paste("Count of NA entries in ArrTime: ",
sum(is.na(df$ArrTime))))

## [1] "Count of NA entries in ArrTime:  0"

print(paste("Count of NA entries in TaxiIn: ",
sum(is.na(df$TaxiIn))))

## [1] "Count of NA entries in TaxiIn:  0"

print(paste("Count of flights diverted: ",
sum(df$Diverted == 1)))

## [1] "Count of flights diverted:  1277"

print(paste("Count of flights cancelled: ",
sum(df$Cancelled == 1)))

## [1] "Count of flights cancelled:  0"

print(paste("Count of observations where both ArrTime and TaxiIn are NA: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn))))

## [1] "Count of observations where both ArrTime and TaxiIn are NA:  0"
```

```r
print(paste("Count of observations where both ArrTime and TaxiIn are NA and
the flight was diverted: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn) & df$Diverted == 1)))

## [1] "Count of observations where both ArrTime and TaxiIn are NA and the
flight was diverted:  0"

print(paste("Count of observations where both ArrTime and TaxiIn are NA and
the flight was cancelled: ",
sum(is.na(df$ArrTime) & is.na(df$TaxiIn) & df$Cancelled == 1)))

## [1] "Count of observations where both ArrTime and TaxiIn are NA and the
flight was cancelled:  0"

summary(df)

##        X                Year          Month          DayofMonth
##  Min.   :      0   Min.   :2008   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:1517713   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :3242312   Median :2008   Median : 6.000   Median :16.00
##  Mean   :3341794   Mean   :2008   Mean   : 6.111   Mean   :15.75
##  3rd Qu.:4973529   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00
##
##    DayOfWeek        DepTime       CRSDepTime      ArrTime       CRSArrTime
##  Min.   :1.000   Min.   :   1   Min.   :   0   Min.   :   1   Min.   :   0
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1316   1st Qu.:1325
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705
##  Mean   :3.985   Mean   :1519   Mean   :1468   Mean   :1610   Mean   :1634
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2359
##
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1929648     Min.   :   1   Length:1929648     Min.   :  14.0
##  Class :character   1st Qu.: 611   Class :character   1st Qu.:  80.0
##  Mode  :character   Median :1543   Mode  :character   Median : 116.0
##                     Mean   :2184                      Mean   : 133.3
##                     3rd Qu.:3423                      3rd Qu.: 165.0
##                     Max.   :9741                      Max.   :1114.0
##                                                       NA's   :1277
##  CRSElapsedTime     AirTime          ArrDelay          DepDelay
##  Min.   :-21.0   Min.   :   0.0   Min.   :-109.00   Min.   :   6.0
##  1st Qu.: 82.0   1st Qu.:  58.0   1st Qu.:   9.00   1st Qu.:  12.0
##  Median :116.0   Median :  90.0   Median :  24.00   Median :  24.0
##  Mean   :134.2   Mean   : 108.2   Mean   :  42.17   Mean   :  43.1
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:  56.00   3rd Qu.:  53.0
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.00   Max.   :2467.0
##
##     Origin              Dest              Distance          TaxiIn
##  Length:1929648     Length:1929648     Min.   :  11.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
```

```
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 765.2   Mean   :  6.813
##                                        3rd Qu.: 998.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##
##     TaxiOut         Cancelled CancellationCode     Diverted
##  Min.   :  0.00   Min.   :0   Length:1929648     Min.   :0.0000000
##  1st Qu.: 10.00   1st Qu.:0   Class :character   1st Qu.:0.0000000
##  Median : 14.00   Median :0   Mode  :character   Median :0.0000000
##  Mean   : 18.22   Mean   :0                      Mean   :0.0006618
##  3rd Qu.: 21.00   3rd Qu.:0                      3rd Qu.:0.0000000
##  Max.   :422.00   Max.   :0                      Max.   :1.0000000
##
##   CarrierDelay     WeatherDelay        NASDelay        SecurityDelay
##  Min.   :   0.0   Min.   :   0.000   Min.   :   0.000   Min.   :  0.0000
##  1st Qu.:   0.0   1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:  0.0000
##  Median :   0.0   Median :   0.000   Median :   0.000   Median :  0.0000
##  Mean   :  12.4   Mean   :   2.394   Mean   :   9.711   Mean   :  0.0583
##  3rd Qu.:  10.0   3rd Qu.:   0.000   3rd Qu.:   6.000   3rd Qu.:  0.0000
##  Max.   :2436.0   Max.   :1352.000   Max.   :1357.000   Max.   :392.0000
##
##  LateAircraftDelay
##  Min.   :   0.00
##  1st Qu.:   0.00
##  Median :   0.00
##  Mean   :  16.35
##  3rd Qu.:  18.00
##  Max.   :1316.00
##
```

We can see there are no longer NA entries in ArrTime and TaxiIn.

The NA entries in ActualElapsedTime will be reviewed since it is the only remaining column with NA.

```
print(paste("Count of NA entries in ActualElapsedTime: ",
sum(is.na(df$ActualElapsedTime))))

## [1] "Count of NA entries in ActualElapsedTime:  1277"

print(paste("Count of flights diverted: ",
sum(df$Diverted == 1)))

## [1] "Count of flights diverted:  1277"

print(paste("Count of flights cancelled: ",
sum(df$Cancelled == 1)))

## [1] "Count of flights cancelled:  0"
```

```r
print(paste("Count of observations where ActualElapsedTime NA and the flight was diverted: ",
sum(is.na(df$ActualElapsedTime) & df$Diverted == 1)))
```

```
## [1] "Count of observations where ActualElapsedTime NA and the flight was diverted:  1277"
```

```r
print(paste("Count of observations where ActualElapsedTime NA and the flight was cancelled: ",
sum(is.na(df$ActualElapsedTime) & df$Cancelled == 1)))
```

```
## [1] "Count of observations where ActualElapsedTime NA and the flight was cancelled:  0"
```

```r
#df[is.na(df$ActualElapsedTime),]
```

We can see that whenever a flight was diverted, there are NA entries in ActualElapsedTime. This makes sense since if the flight was diverted, data may not have needed to be recorded for ActualElapsedTime.

To fix, we will replace these NA with 0 to represent there was no elapsed time when the flight was diverted.

```r
df[is.na(df$ActualElapsedTime) & df$Diverted == 1,] <- 0
```

We can see the replacement was successful:

```r
print(paste("Count of NA entries in ActualElapsedTime: ",
sum(is.na(df$ActualElapsedTime))))
```

```
## [1] "Count of NA entries in ActualElapsedTime:  0"
```

```r
print(paste("Count of flights diverted: ",
sum(df$Diverted == 1)))
```

```
## [1] "Count of flights diverted:  0"
```

```r
print(paste("Count of flights cancelled: ",
sum(df$Cancelled == 1)))
```

```
## [1] "Count of flights cancelled:  0"
```

```r
print(paste("Count of observations where ActualElapsedTime NA and the flight was diverted: ",
sum(is.na(df$ActualElapsedTime) & df$Diverted == 1)))
```

```
## [1] "Count of observations where ActualElapsedTime NA and the flight was diverted:  0"
```

```r
print(paste("Count of observations where ActualElapsedTime NA and the flight was cancelled: ",
sum(is.na(df$ActualElapsedTime) & df$Cancelled == 1)))
```

```
## [1] "Count of observations where ActualElapsedTime NA and the flight was
cancelled:  0"
```

```
summary(df)
```

```
##        X                Year          Month          DayofMonth
##  Min.   :      0   Min.   :   0   Min.   : 0.000   Min.   : 0.00
##  1st Qu.:1513978   1st Qu.:2008   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :3239305   Median :2008   Median : 6.000   Median :16.00
##  Mean   :3337505   Mean   :2007   Mean   : 6.104   Mean   :15.74
##  3rd Qu.:4967922   3rd Qu.:2008   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :7009727   Max.   :2008   Max.   :12.000   Max.   :31.00
##    DayOfWeek        DepTime       CRSDepTime      ArrTime       CRSArrTime
##  Min.   :0.000   Min.   :   0   Min.   :   0   Min.   :   0   Min.   :   0
##  1st Qu.:2.000   1st Qu.:1203   1st Qu.:1135   1st Qu.:1315   1st Qu.:1325
##  Median :4.000   Median :1545   Median :1510   Median :1715   Median :1705
##  Mean   :3.982   Mean   :1518   Mean   :1467   Mean   :1609   Mean   :1633
##  3rd Qu.:6.000   3rd Qu.:1900   3rd Qu.:1815   3rd Qu.:2030   3rd Qu.:2014
##  Max.   :7.000   Max.   :2400   Max.   :2359   Max.   :2400   Max.   :2359
##  UniqueCarrier        FlightNum      TailNum          ActualElapsedTime
##  Length:1929648     Min.   :   0   Length:1929648     Min.   :   0.0
##  Class :character   1st Qu.: 609   Class :character   1st Qu.:  80.0
##  Mode  :character   Median :1542   Mode  :character   Median : 116.0
##                     Mean   :2183                      Mean   : 133.2
##                     3rd Qu.:3422                      3rd Qu.: 165.0
##                     Max.   :9741                      Max.   :1114.0
##  CRSElapsedTime     AirTime         ArrDelay          DepDelay
##  Min.   :-21.0   Min.   :   0.0   Min.   :-109.00   Min.   :   0.00
##  1st Qu.: 81.0   1st Qu.:  58.0   1st Qu.:    9.00   1st Qu.:  12.00
##  Median :116.0   Median :  90.0   Median :   24.00   Median :  24.00
##  Mean   :134.1   Mean   : 108.2   Mean   :   42.17   Mean   :  43.06
##  3rd Qu.:165.0   3rd Qu.: 137.0   3rd Qu.:   56.00   3rd Qu.:  53.00
##  Max.   :660.0   Max.   :1091.0   Max.   :2461.00   Max.   :2467.00
##     Origin              Dest              Distance          TaxiIn
##  Length:1929648     Length:1929648     Min.   :   0.0   Min.   :  0.000
##  Class :character   Class :character   1st Qu.: 338.0   1st Qu.:  4.000
##  Mode  :character   Mode  :character   Median : 606.0   Median :  6.000
##                                        Mean   : 764.4   Mean   :  6.807
##                                        3rd Qu.: 997.0   3rd Qu.:  8.000
##                                        Max.   :4962.0   Max.   :240.000
##     TaxiOut         Cancelled CancellationCode     Diverted  CarrierDelay
##  Min.   :  0.00   Min.   :0   Length:1929648     Min.   :0   Min.   :
0.0
##  1st Qu.: 10.00   1st Qu.:0   Class :character   1st Qu.:0   1st Qu.:
0.0
##  Median : 14.00   Median :0   Mode  :character   Median :0   Median :
0.0
##  Mean   : 18.21   Mean   :0                      Mean   :0   Mean   :
12.4
##  3rd Qu.: 21.00   3rd Qu.:0                      3rd Qu.:0   3rd Qu.:
```

```
10.0
##  Max.   :422.00   Max.   :0                      Max.   :0   Max.   
:2436.0  
##   WeatherDelay         NASDelay        SecurityDelay      
LateAircraftDelay
##  Min.   :   0.000   Min.   :   0.000   Min.   :  0.0000   Min.   :   0.00  
##  1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:  0.0000   1st Qu.:   0.00  
##  Median :   0.000   Median :   0.000   Median :  0.0000   Median :   0.00  
##  Mean   :   2.394   Mean   :   9.711   Mean   :  0.0583   Mean   :  16.35  
##  3rd Qu.:   0.000   3rd Qu.:   6.000   3rd Qu.:  0.0000   3rd Qu.:  18.00  
##  Max.   :1352.000   Max.   :1357.000   Max.   :392.0000   Max.   :1316.00
```

We can see there are no longer NA entries!

We can now begin linear regression.

Creating Linear Model

Linear regression depends on quantitative data rather than qualitative data. As seen in the output below, there are various qualitative columns (UniqueCarrier, TailNum, Origin, Dest, and CancellationCode) in our data frame.

```
str(df)

## 'data.frame':    1929648 obs. of  30 variables:
##  $ X                : num  0 1 2 4 5 6 10 11 15 16 ...
##  $ Year             : num  2008 2008 2008 2008 2008 ...
##  $ Month            : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DayofMonth       : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ DayOfWeek        : num  4 4 4 4 4 4 4 4 4 4 ...
##  $ DepTime          : num  2003 754 628 1829 1940 ...
##  $ CRSDepTime       : num  1955 735 620 1755 1915 ...
##  $ ArrTime          : num  2211 1002 804 1959 2121 ...
##  $ CRSArrTime       : num  2225 1000 750 1925 2110 ...
##  $ UniqueCarrier    : chr  "WN" "WN" "WN" "WN" ...
##  $ FlightNum        : num  335 3231 448 3920 378 ...
##  $ TailNum          : chr  "N712SW" "N772SW" "N428WN" "N464WN" ...
##  $ ActualElapsedTime: num  128 128 96 90 101 240 130 121 52 228 ...
##  $ CRSElapsedTime   : num  150 145 90 90 115 250 135 135 50 240 ...
##  $ AirTime          : num  116 113 76 77 87 230 106 107 37 213 ...
##  $ ArrDelay         : num  -14 2 14 34 11 57 1 80 11 15 ...
##  $ DepDelay         : num  8 19 8 34 25 67 6 94 9 27 ...
##  $ Origin           : chr  "IAD" "IAD" "IND" "IND" ...
##  $ Dest             : chr  "TPA" "TPA" "BWI" "BWI" ...
##  $ Distance         : num  810 810 515 515 688 ...
##  $ TaxiIn           : num  4 5 3 3 4 3 5 6 6 7 ...
##  $ TaxiOut          : num  8 10 17 10 10 7 19 8 9 8 ...
##  $ Cancelled        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CancellationCode : chr  "N" "N" "N" "N" ...
##  $ Diverted         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CarrierDelay     : num  0 0 0 2 0 10 0 8 0 3 ...
```

```
##  $ WeatherDelay     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NASDelay         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SecurityDelay    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LateAircraftDelay: num  0 0 0 32 0 47 0 72 0 12 ...
```

We will remove these qualitative columns since they will not be of use for linear regression.

```
df <- df[,!names(df) %in% c("UniqueCarrier", "TailNum", "Origin", "Dest",
"CancellationCode")]
```

a. Dividing into 80/20 Train/Test

```
dt = sort(sample(nrow(df), nrow(df)*.8, replace=FALSE))
train <- df[dt,]
test <- df[-dt,]
```

b. 1/5 Data Exploration on Training Data - Correlation

Correlation will be the first data exploration. Correlation shows how well two columns correlate to one another and provide a basis for identifying potential relationships. Correlation ranges [-1, 1] where the closer to -1, more negative the relationship and the closer to 1, the more positive the relationship. The closer to 0, the more there is not a relationship.

```
print(paste("The correlation between Distance and ActualElapsedTime: ",
cor(train$Distance, train$ActualElapsedTime)))

## [1] "The correlation between Distance and ActualElapsedTime:
0.952902746050109"

print(paste("The correlation between Distance and DeptTime: ",
cor(train$Distance, train$DepTime)))

## [1] "The correlation between Distance and DeptTime:  -0.0525949760217026"

print(paste("The correlation between Distance and AirTime: ",
cor(train$Distance, train$AirTime)))

## [1] "The correlation between Distance and AirTime:  0.980274181004953"

print(paste("The correlation between Distance and TaxiIn: ",
cor(train$Distance, train$TaxiIn)))

## [1] "The correlation between Distance and TaxiIn:  0.0730640609399879"

print(paste("The correlation between Distance and DayOfWeek: ",
cor(train$Distance, train$DayOfWeek)))

## [1] "The correlation between Distance and DayOfWeek:  0.0101466647493154"

print(paste("The correlation between Actual Elapsed Time and AirTime: ",
cor(train$ActualElapsedTime, train$AirTime)))
```

```
## [1] "The correlation between Actual Elapsed Time and AirTime:
0.976657964481313"
```

As seen above… 1. Distance and ActualElapsedTime have a near perfect positive relationship 2. Distance and DeptTime have a barely negative relationship 3. Distance and AirTime have a near perfect positive relationship 4. Distance and DayOfWeek have a barely positive relationship 5. ActualElapsedTime and AirTime have a near perfect positive relationship.

b. 2/5 Data Exploration on Training Data - Covariance

Covariance is correlation, but its range is [-inf., inf.]. Covariance measures how changes in one column are associated with changes in a another column.

```
print(paste("Covariance of distance and actual elapsed time: ",
cov(train$Distance, train$ActualElapsedTime, method="pearson")))

## [1] "Covariance of distance and actual elapsed time:  39462.3793046268"

print(paste("Covariance of ditance and departure time: ",
cov(train$Distance, train$DepTime, method="pearson")))

## [1] "Covariance of ditance and departure time:  -13654.0020420205"

print(paste("Covariance of distance and air time: ",
cov(train$Distance, train$AirTime, method="pearson")))

## [1] "Covariance of distance and air time:  38657.3007667116"

print(paste("Covariance of distance and taxi in: ",
cov(train$Distance, train$TaxiIn, method="pearson")))

## [1] "Covariance of distance and taxi in:  221.399599116644"

print(paste("Covariance of distance and day of the week: ",
cov(train$Distance, train$DayOfWeek, method="pearson")))

## [1] "Covariance of distance and day of the week:  11.641329679695"

print(paste("Covariance of actual elapsed time and air time: ",
cov(train$ActualElapsedTime, train$AirTime, method="pearson")))

## [1] "Covariance of actual elapsed time and air time:  4838.46020801567"
```

Compared to correlation, covariance is much more difficult to read and quickly understand how different columns impact one another.

b. 3/5 Data Exploration on Training Data - Dimension

```
dim(train)

## [1] 1543718      25
```

From calling dimension, we can see there are 1543718 rows and 25 columns. This is more than sufficient for linear regression.

b. 4/5 Data Exploration on Training Data - Structure

```
str(train)

## 'data.frame':    1543718 obs. of  25 variables:
##  $ X                : num  0 1 2 4 5 10 11 15 17 18 ...
##  $ Year             : num  2008 2008 2008 2008 2008 ...
##  $ Month            : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DayofMonth       : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ DayOfWeek        : num  4 4 4 4 4 4 4 4 4 4 ...
##  $ DepTime          : num  2003 754 628 1829 1940 ...
##  $ CRSDepTime       : num  1955 735 620 1755 1915 ...
##  $ ArrTime          : num  2211 1002 804 1959 2121 ...
##  $ CRSArrTime       : num  2225 1000 750 1925 2110 ...
##  $ FlightNum        : num  335 3231 448 3920 378 ...
##  $ ActualElapsedTime: num  128 128 96 90 101 130 121 52 226 123 ...
##  $ CRSElapsedTime   : num  150 145 90 90 115 135 135 50 250 135 ...
##  $ AirTime          : num  116 113 76 77 87 106 107 37 205 110 ...
##  $ ArrDelay         : num  -14 2 14 34 11 1 80 11 -15 16 ...
##  $ DepDelay         : num  8 19 8 34 25 6 94 9 9 28 ...
##  $ Distance         : num  810 810 515 515 688 ...
##  $ TaxiIn           : num  4 5 3 3 4 5 6 6 5 4 ...
##  $ TaxiOut          : num  8 10 17 10 10 19 8 9 16 9 ...
##  $ Cancelled        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Diverted         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CarrierDelay     : num  0 0 0 2 0 0 8 0 0 0 ...
##  $ WeatherDelay     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NASDelay         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SecurityDelay    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LateAircraftDelay: num  0 0 0 32 0 0 72 0 0 16 ...
```

As seen before and above, our training set consists of only quantitative data. This is ideal for linear regression.

b. 5/5 Data Exploration on Training Data - Head

```
head(train)

##    X Year Month DayofMonth DayOfWeek DepTime CRSDepTime ArrTime CRSArrTime
## 1  0 2008     1          3         4    2003       1955    2211       2225
## 2  1 2008     1          3         4     754        735    1002       1000
## 3  2 2008     1          3         4     628        620     804        750
## 4  4 2008     1          3         4    1829       1755    1959       1925
## 5  5 2008     1          3         4    1940       1915    2121       2110
## 7 10 2008     1          3         4     706        700     916        915
##   FlightNum ActualElapsedTime CRSElapsedTime AirTime ArrDelay DepDelay
Distance
## 1       335               128            150     116      -14        8
810
```

```
## 2      3231               128            145     113        2        19
810
## 3       448                96             90      76       14         8
515
## 4      3920                90             90      77       34        34
515
## 5       378               101            115      87       11        25
688
## 7       100               130            135     106        1         6
828
##   TaxiIn TaxiOut Cancelled Diverted CarrierDelay WeatherDelay NASDelay
## 1      4       8         0        0            0            0        0
## 2      5      10         0        0            0            0        0
## 3      3      17         0        0            0            0        0
## 4      3      10         0        0            2            0        0
## 5      4      10         0        0            0            0        0
## 7      5      19         0        0            0            0        0
##   SecurityDelay LateAircraftDelay
## 1             0                 0
## 2             0                 0
## 3             0                 0
## 4             0                32
## 5             0                 0
## 7             0                 0
```

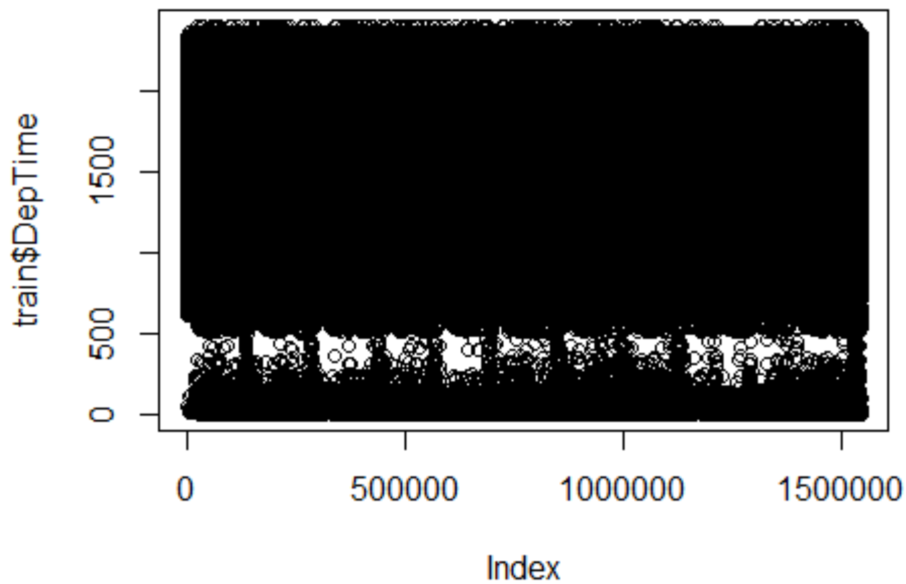
From visual inspection of the output above, we can see that all columns have reasonable inputs.

b. Additional Data Exploration on Training Data - Mean, Median, Range,

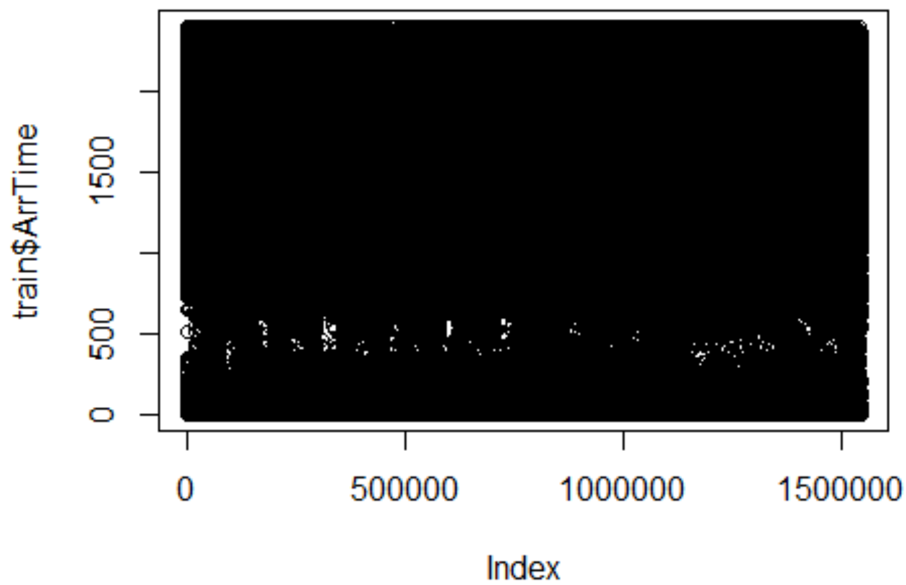
```
print(paste("Mean of distance: ", mean(train$Distance)))

## [1] "Mean of distance:  764.220110797438"

print(paste("Median of distance: ", median(train$Distance)))

## [1] "Median of distance:  606"

print(paste("Range of distance: ", max(train$Distance) -
min(train$Distance)))

## [1] "Range of distance:  4962"

print("Unique elements in column Year: ")

## [1] "Unique elements in column Year: "

unique(train$Year)

## [1] 2008    0

lm_train <- lm(train$Distance~train$ActualElapsedTime, data=train)
```

c. 1/2 Informative Graphs on Training Data

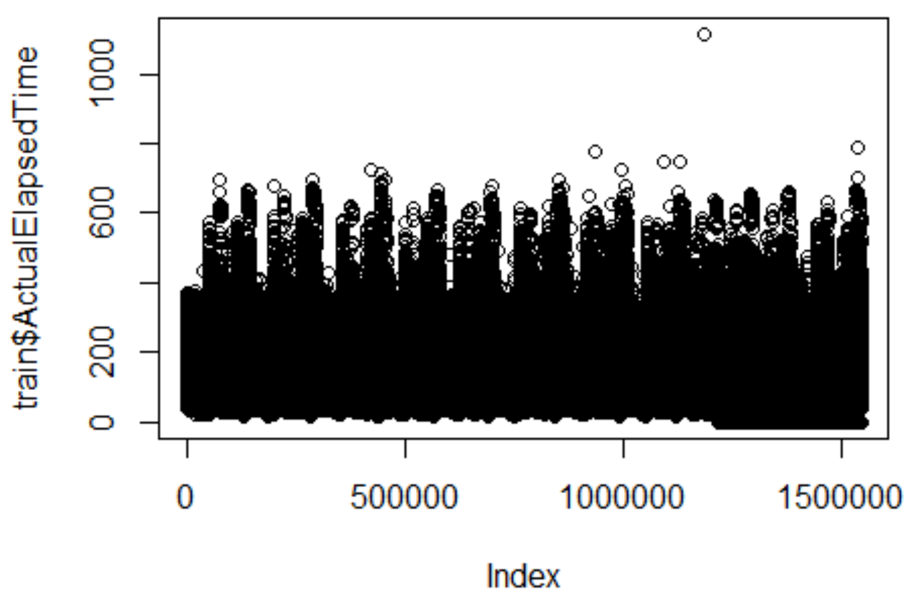
```
plot(train$DepTime)
```




```
plot(train$ArrTime)
```




```
plot(train$ActualElapsedTime)
```



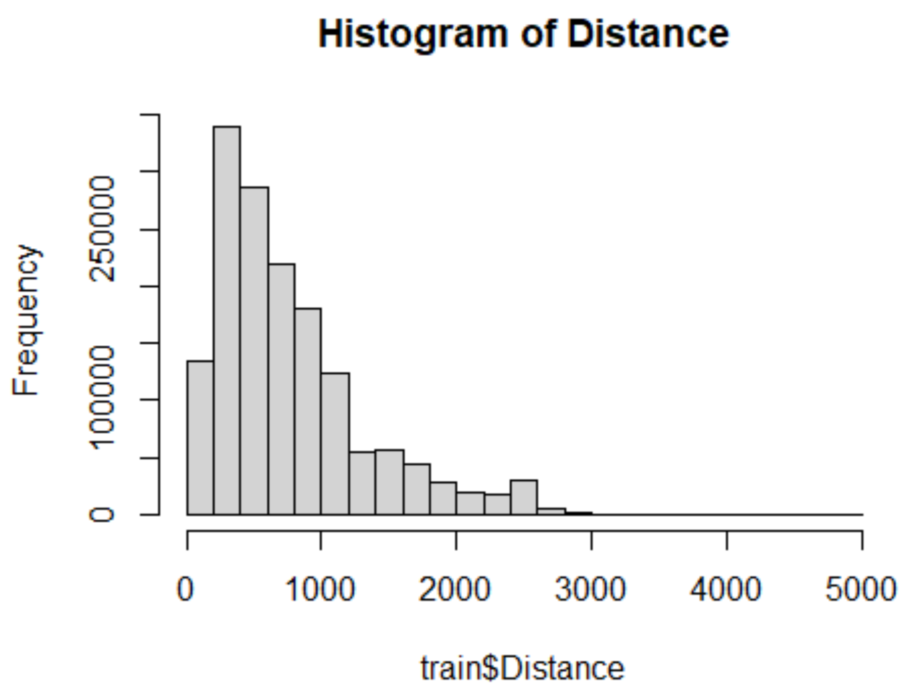
Each of the plots output above shows a visual representation of how each column is distrubuted by index.

These graphs can be used to gain a quick understanding of how our data is distributed and identify any outliers.

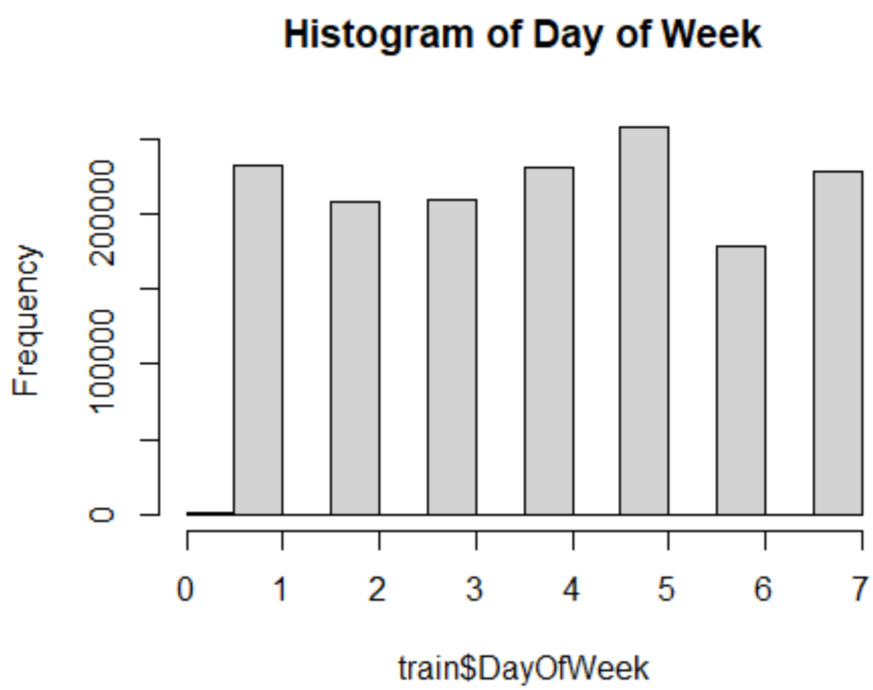
c. 2/2 Informative Graphs on Training Data

Here we will create a histogram to see the distribution for different distances traveled.

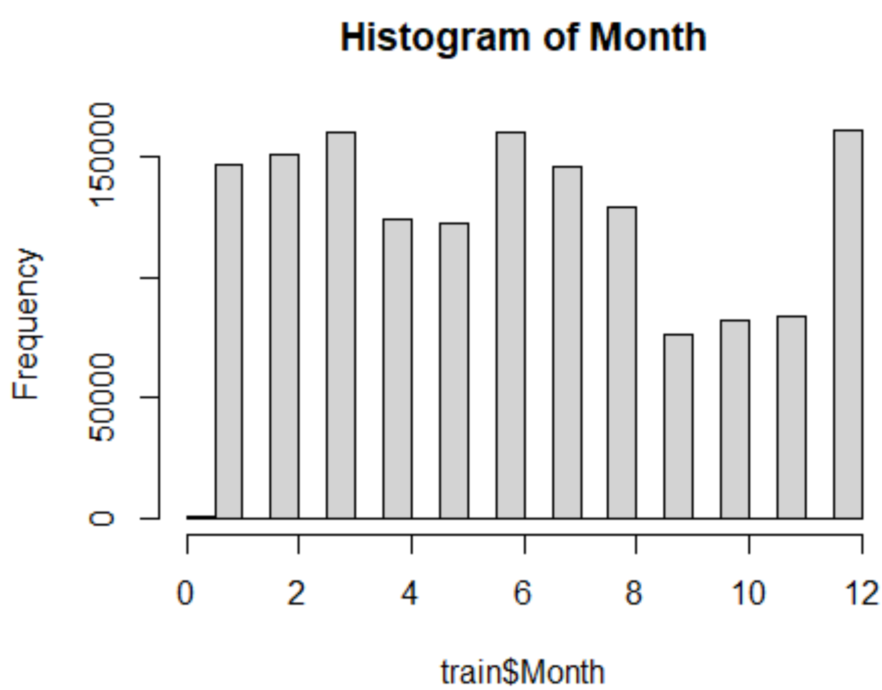
```
hist(train$Distance, main="Histogram of Distance")
```



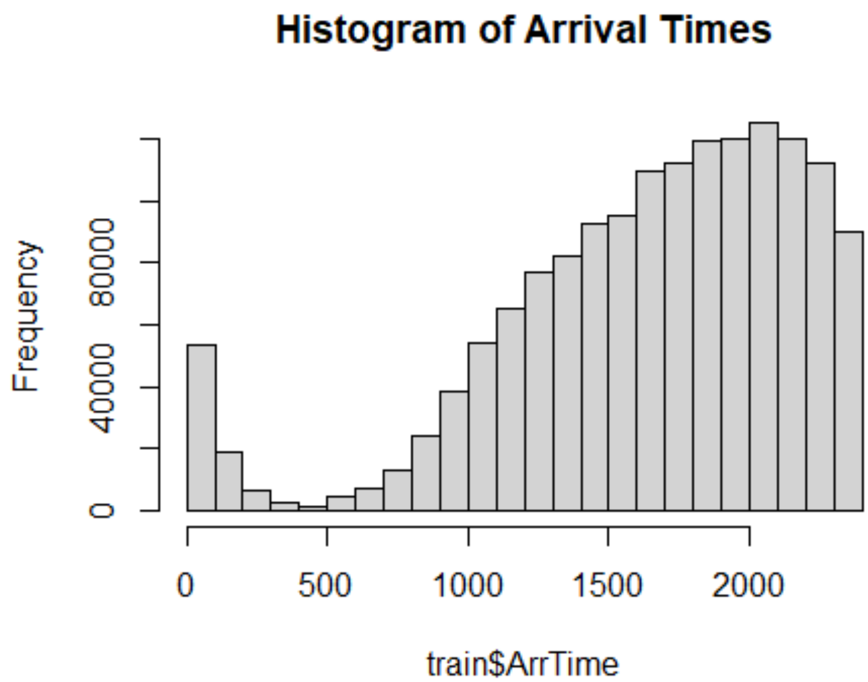

```
hist(train$DayOfWeek, main="Histogram of Day of Week")
```



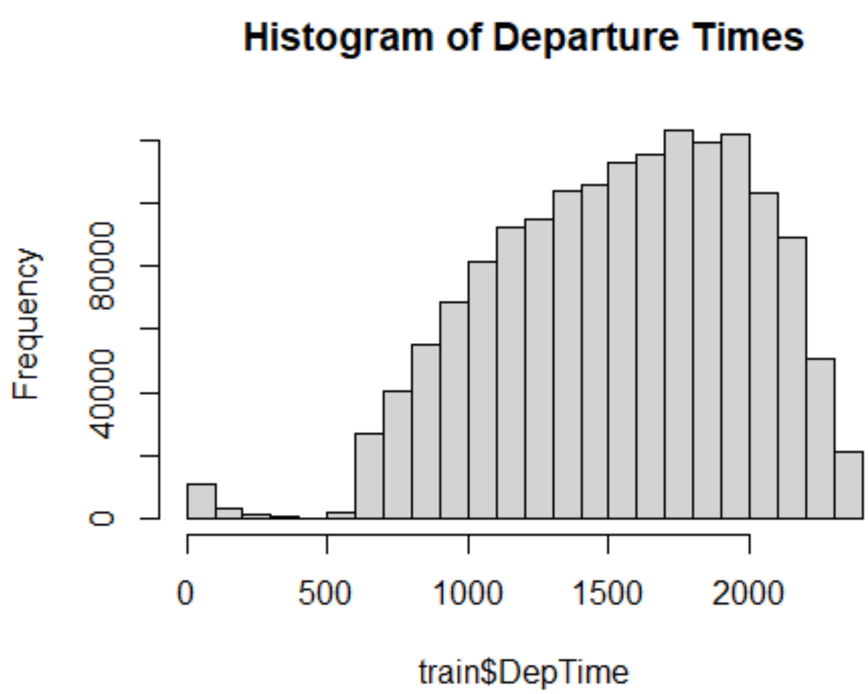

```
hist(train$Month, main="Histogram of Month")
```




```
hist(train$ArrTime, main="Histogram of Arrival Times")
```




```
hist(train$DepTime, main="Histogram of Departure Times")
```

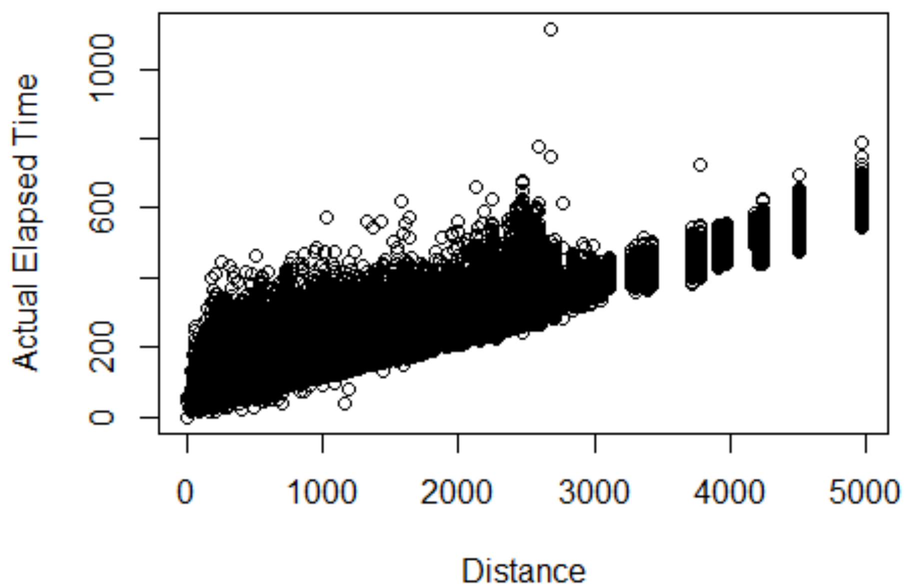


From these histograms, we can see how often the data fits into certain categories. From the histograms, we can see: 1. Histogram of Distance - We can see that majority of flights are below 3000 2.

Histogram of day of week - we can see that the frequency for each day of the week is relatively even 3. Histogram of Month - We can see that the flight frequency for each month is relatively even 4. Histogram of Arrival Times - We can see that majority of the flights occurred after 5:00 AM 5. Histogram of Departure Times - We can see that the majority of flights arrived after 5:00 AM These histograms allow us to better understand how the data is distributed.

c. Additional Informational Graphs

```
plot(train$Distance, train$ActualElapsedTime, xlab="Distance", ylab="Actual
Elapsed Time")
```



d. Building Simple Linear Regression Model (One Predictor) & Summary

Now we will build a simple linear regression model (with one predictor) and output its summary.

Here our one predictor is ActualElapsedTime.

```
lm_train_simple <- lm(train$ActualElapsedTime~train$Distance, data=train)

summary(lm_train_simple)

##
## Call:
## lm(formula = train$ActualElapsedTime ~ train$Distance, data = train)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -139.04  -13.22   -4.02    8.84  751.84
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.170e+01  2.931e-02    1423   <2e-16 ***
## train$Distance 1.197e-01  3.066e-05    3904   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.87 on 1543716 degrees of freedom
## Multiple R-squared:  0.908,  Adjusted R-squared:  0.908
## F-statistic: 1.524e+07 on 1 and 1543716 DF,  p-value: < 2.2e-16
```

As seen above, the summary call provides a rich overview of our linear regression model and its fit.

The first section we will look at is the coefficient section which indicates how well each coefficient modeled the true data.

The estimated coefficient for actual elapsed time and the intercept are provided, along with standard error, t-value, and the p-value.

The standard error provides an estimate of variation in the coefficient estimate and can be used to predict a confidence interval for the coefficient. The standard error is used for the hypothesis test on the coefficient, where the null hypothesis is that there is no relationship between the predictor variable and the target variable.

We can see from our output that the standard error is very small, meaning there is little variation in the coefficient estimate.

The standard error is used to calculate the t-value. The t-value measures the number of standard deviations the estimate coefficient is from 0. The distribution of the t-value has a bell shape which makes it easy to compute the probability of observing a t-value larger in absolute value than what was computed, if the null hypothesis were true.

The p-value is used to determine if the null hypothesis can be rejected. The larger the data set, the more confidence can be taken from the p-value. From our data set, we can definitely have confidence in our p-value.

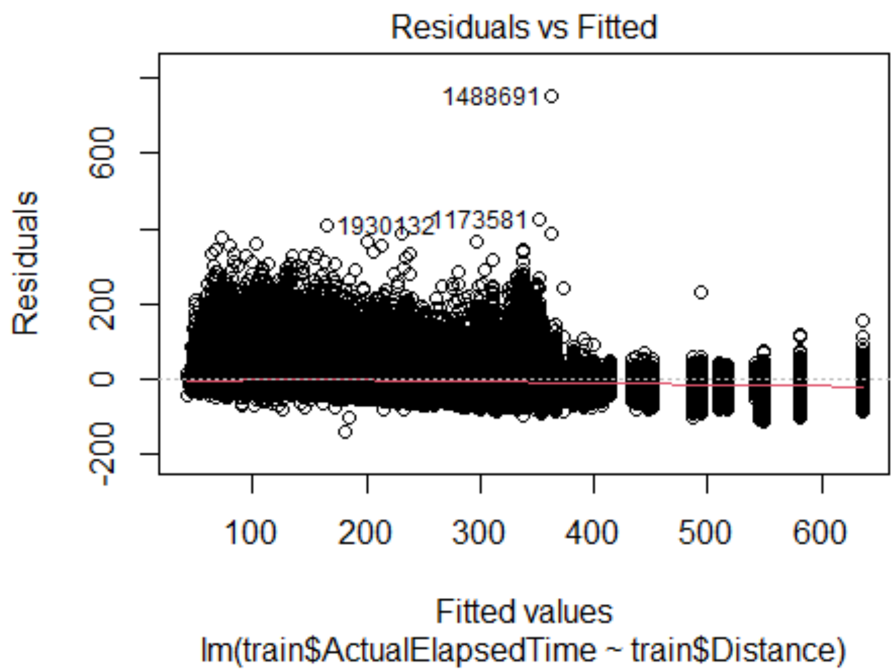
The last section of the summary provides information on the residual standard error, the multiple r-squared, the adjusted r-squared, the f-statistics, and the p-value. Unlike the coefficient section, this section tells us how well the model as a whole fit the training data.

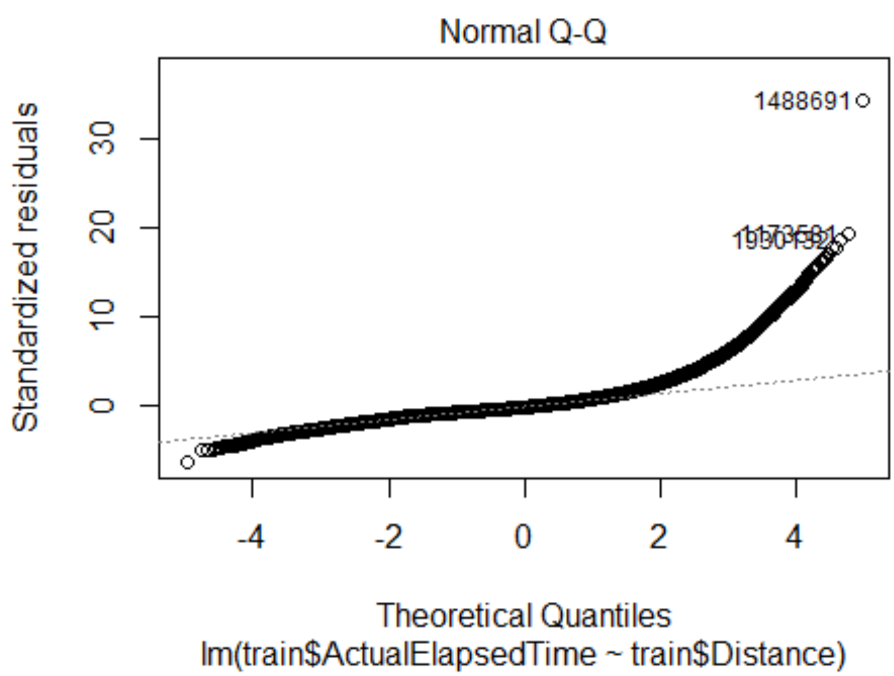
The residual standard error is found from the residual sum of squares (we square them to correct for negative directions) and measures how off our model was from the data, the lack of fit of the model.

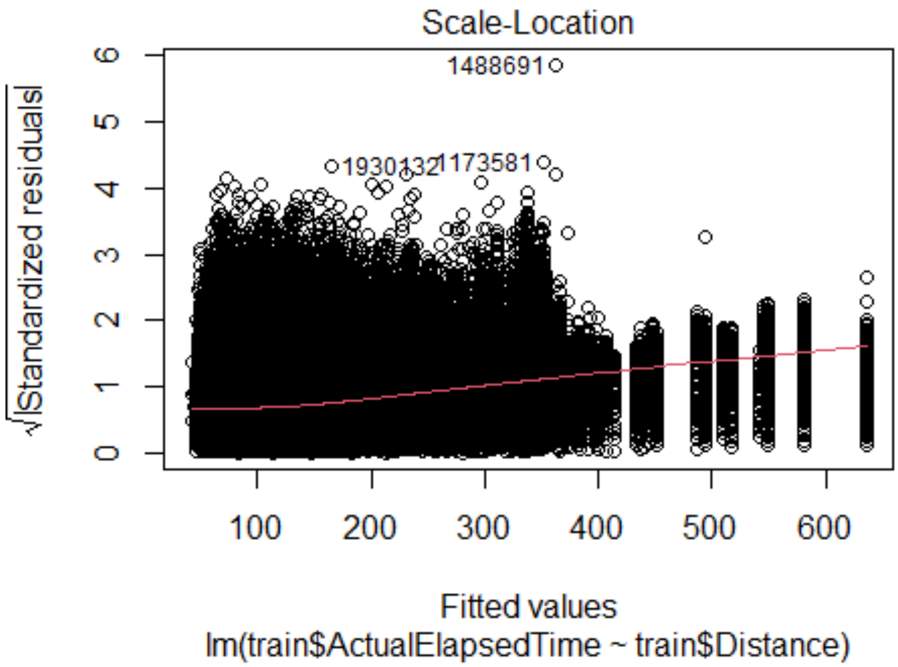
The f-statistic takes into account all of the predictors to determine if they are significant predictors of Y. It provides evidence against the null hypothesis that the predictors are not really predictors.

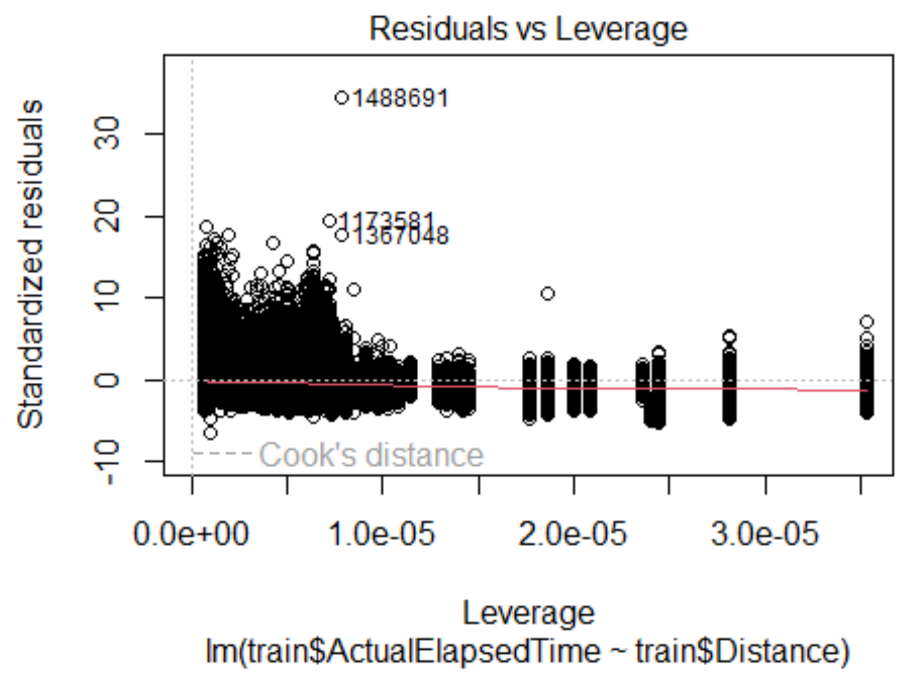
e. Plotting the Residuals

```
plot(lm_train_simple)
```


Residuals vs Fitted
1488691
1930132
1173581
Residuals
600
200
0
-200
100
200
300
400
500
600
Fitted values
lm(train$ActualElapsedTime ~ train$Distance)


Normal Q-Q
1488691
1173581
1930132
Standardized residuals
30
20
10
0
-4
-2
0
2
4
Theoretical Quantiles
lm(train$ActualElapsedTime ~ train$Distance)



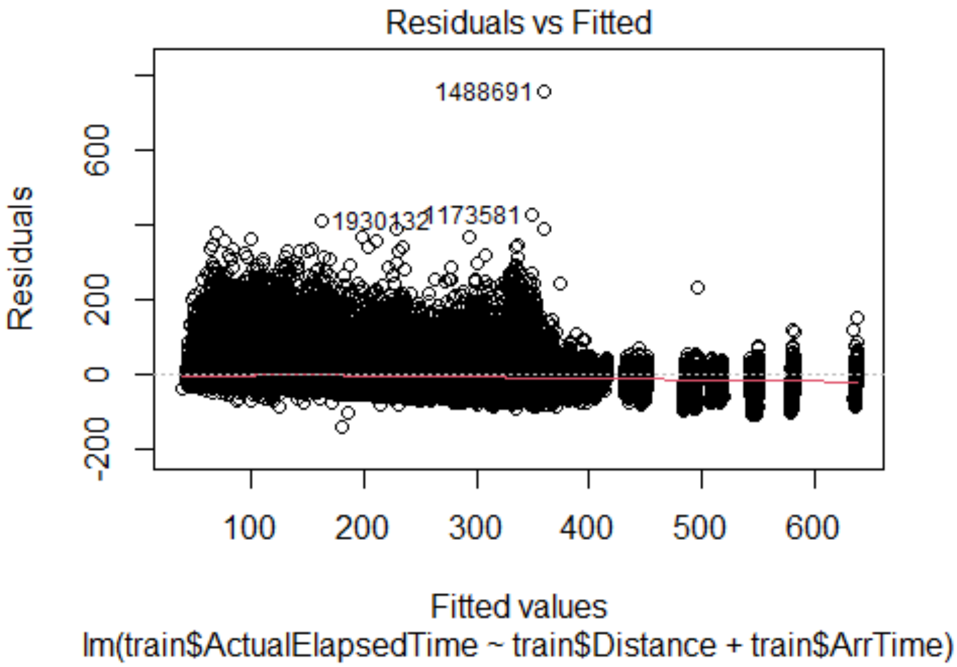


The output above shows the four residual plots. Each residual plot is meant to be used to aid in understanding and improving the regression model.

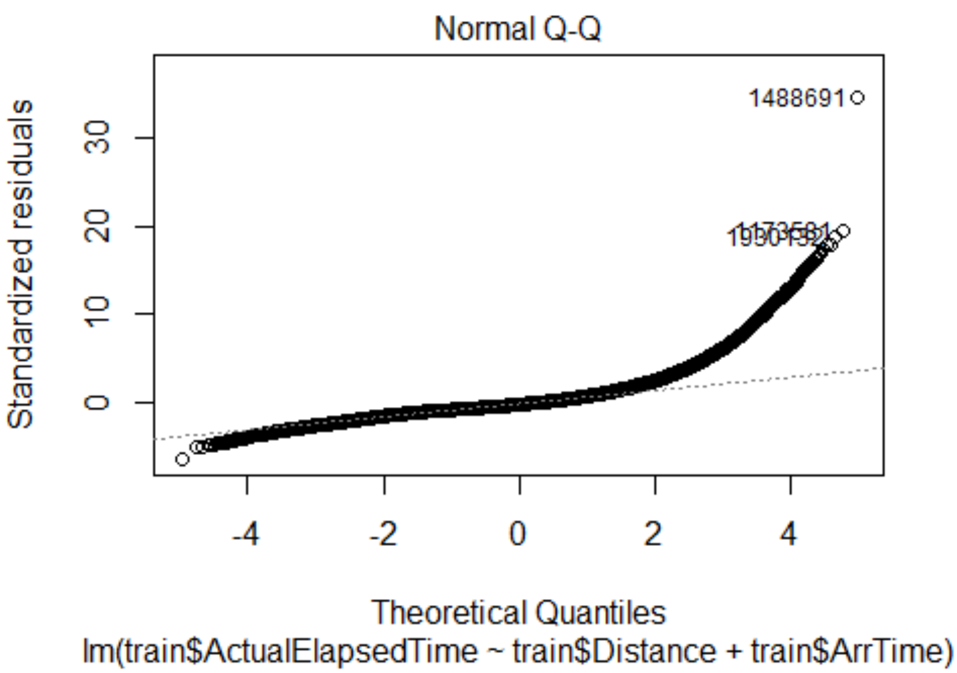
1. Residual vs. Fitted - This plot shows the residual (errors) with a red trend line. The more horizontal the red line, the less variation in the data that the model did not capture. Since the plot has a relatively horizontal line, we can confirm there is less variation.
2. Normal Q-Q - This plot shows if the residuals are somewhat normally distributed (since there is a fairly straight diagonal line). The closer the data is to the line, the more normally distributed the data is. When the points are further away from the line, the model may need to be reviewed.
3. Scale-Location - This plot shows if the data is homoscedastic (meaning "same variance). Since there is not a fairly straight line with points distributed equally around it, we can say the data is not homoscedastic. We can see that the red lined is curved since there is a cluster of data favoring the lower x-axis.
4. Residuals vs. Leverage - This plot indicates leverage points which are influencing the regression line (they may or may not be outliers). Cook's distance (the grey dashed line) shows the impact of removing points as the points outside of the dotted line have high influence.

f. Building a Multiple Linear Regression Model (Multiple Predictors), Summary, and Residuals Plot

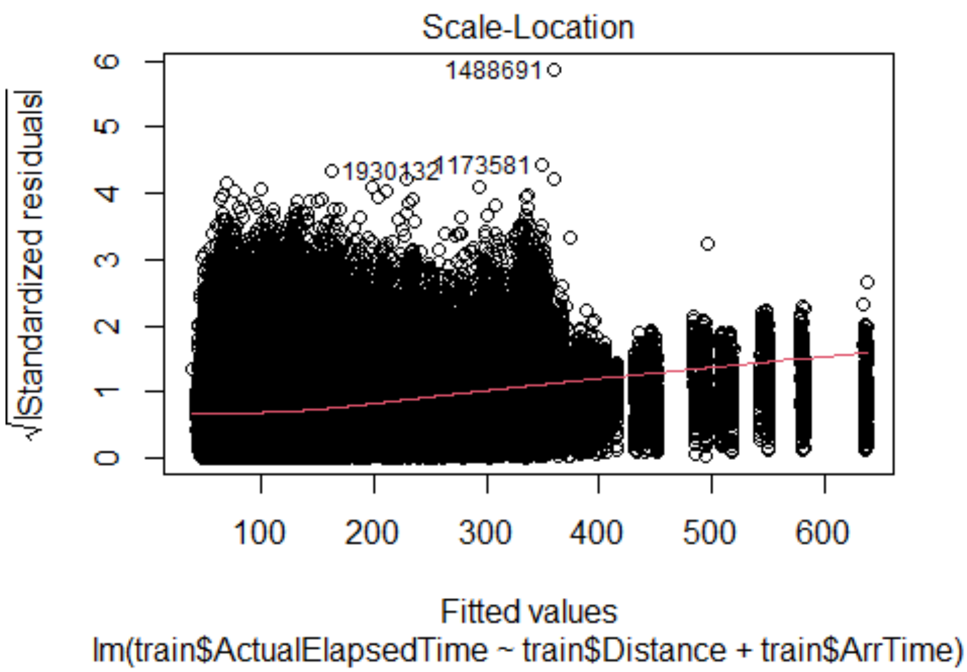
```
lm_train_multiple <- lm(train$ActualElapsedTime~train$Distance+train$ArrTime,
data=train)

summary(lm_train_multiple)

##
## Call:
## lm(formula = train$ActualElapsedTime ~ train$Distance + train$ArrTime,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -138.56  -13.24   -4.00    8.84  753.67
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.877e+01  5.975e-02  648.93   <2e-16 ***
## train$Distance 1.198e-01  3.064e-05 3908.02   <2e-16 ***
## train$ArrTime  1.798e-03  3.200e-05   56.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.85 on 1543715 degrees of freedom
## Multiple R-squared:  0.9082, Adjusted R-squared:  0.9082
## F-statistic: 7.637e+06 on 2 and 1543715 DF,  p-value: < 2.2e-16

plot(lm_train_multiple)
```

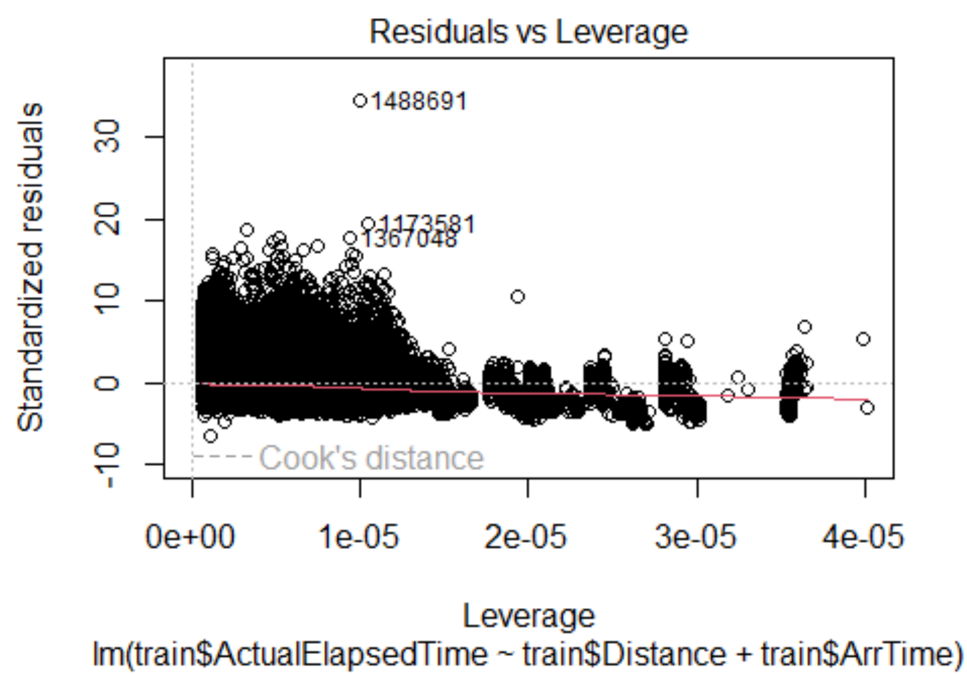


Residuals vs Fitted
1488691
1930132
1173581
Residuals
600
200
0
-200
100
200
300
400
500
600
Fitted values
lm(train$ActualElapsedTime ~ train$Distance + train$ArrTime)



Normal Q-Q
1488691
1173581
1930132
Standardized residuals
30
20
10
0
-4
-2
0
2
4
Theoretical Quantiles
lm(train$ActualElapsedTime ~ train$Distance + train$ArrTime)





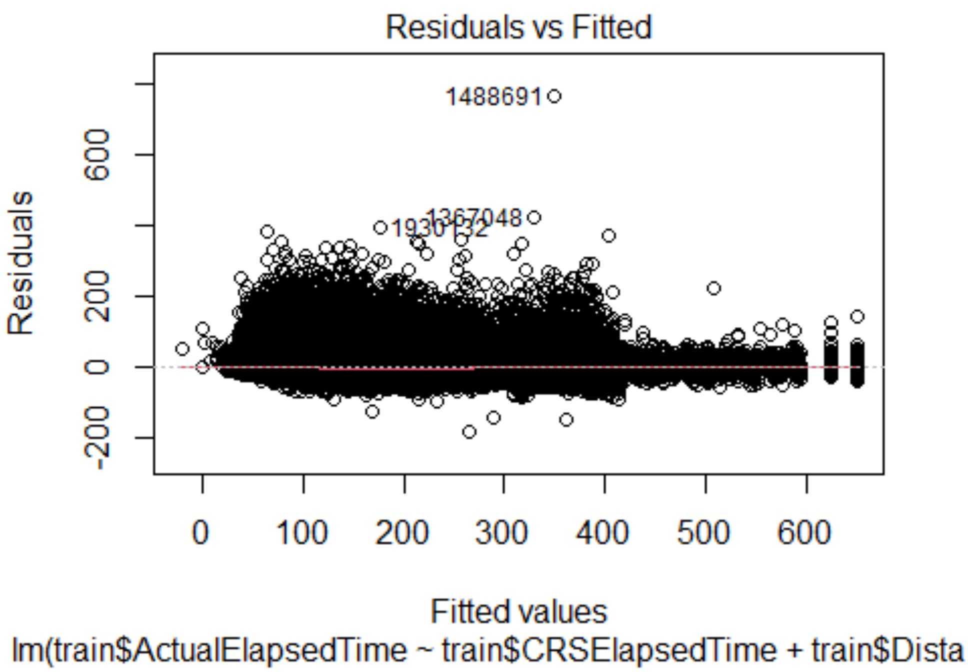
g. Building a Third Linear Model (Different Combination of Predictors), Summary, and Residual Plot

```
lm_train_third <-
lm(train$ActualElapsedTime~train$CRSElapsedTime+train$Distance, data=train)
```

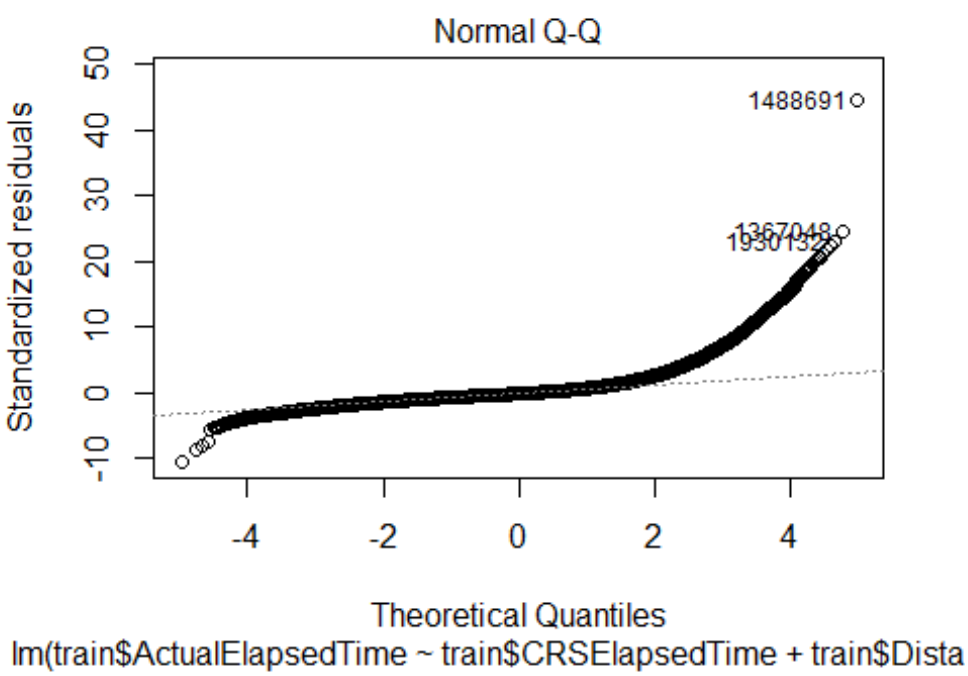
```
summary(lm_train_third)
```

```
##
## Call:
## lm(formula = train$ActualElapsedTime ~ train$CRSElapsedTime +
##     train$Distance, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -181.02   -9.32   -2.87    5.54  764.16
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.7934144  0.0478381   16.59   <2e-16 ***
## train$CRSElapsedTime  1.0004407  0.0010252  975.82   <2e-16 ***
## train$Distance       -0.0022836  0.0001273  -17.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.2 on 1543715 degrees of freedom
## Multiple R-squared:  0.9431, Adjusted R-squared:  0.9431
## F-statistic: 1.28e+07 on 2 and 1543715 DF,  p-value: < 2.2e-16
```

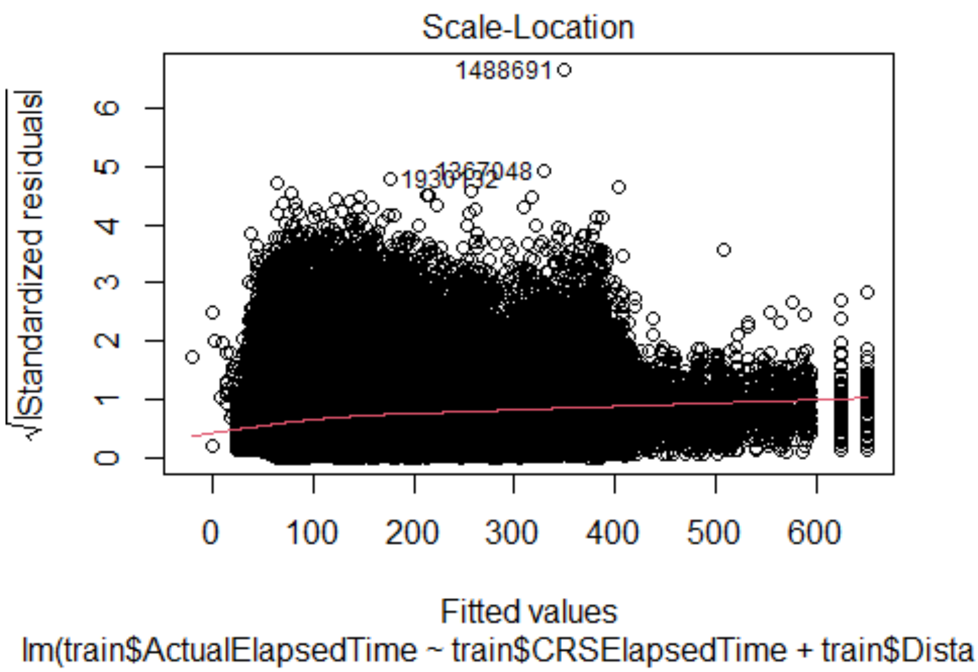
```
plot(lm_train_third)
```

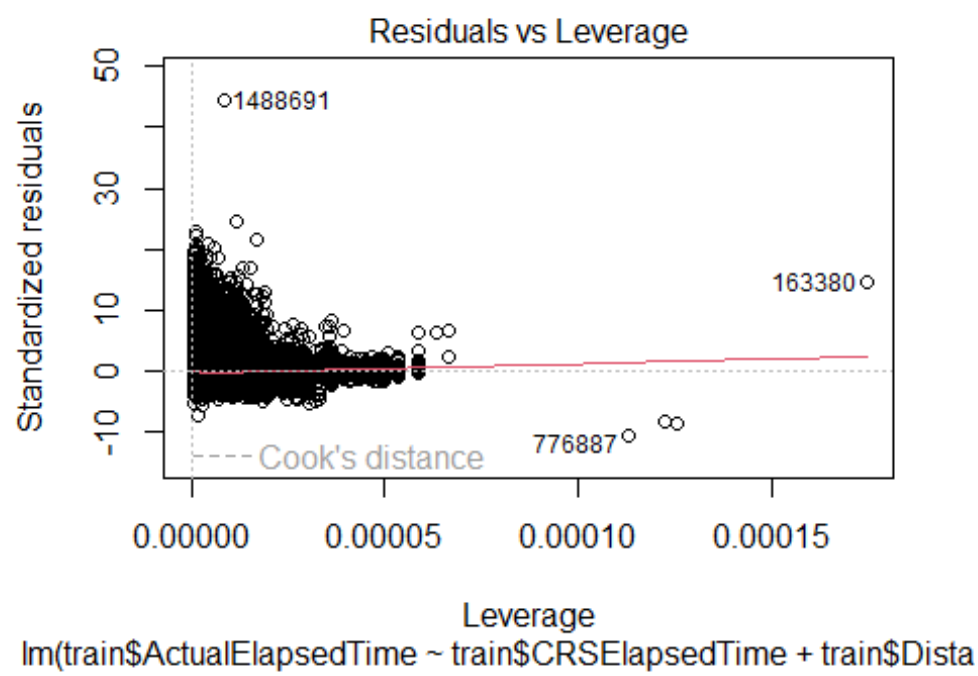


Residuals vs Fitted
Residuals
1488691
1367048
1930132
Fitted values
lm(train$ActualElapsedTime ~ train$CRSElapsedTime + train$Dista



Normal Q-Q
Standardized residuals
1488691
1367048
1930132
Theoretical Quantiles
lm(train$ActualElapsedTime ~ train$CRSElapsedTime + train$Dista





h. Comparing the Results

From the three models, the third model is the best. This can be seen from both the summary and the residual plots. Within the summary for each model, the residual standard error, multiple r-squared, and adjusted r-squared improve as we work toward the third model. The residual standard error measures the standard deviation of the residuals in a regression model, and the smaller the residual standard error is, the better. As we can see in the third model, the residual standard error is much smaller than the previous two models. Additionally, having a multiple-r squared and adjusted r-squared closer to 1 means that the third model can better explain any variance by its predictors. Since these three factors help in indicating how well a model works, we can see that the third model has the best evidence for being the best model. Beyond the summary, the third model is best from its residual plots. Compared to the previous models, the residual plots of the third model are more evenly distributed. As seen in the Residuals vs. Fitted plot, the data is much more evenly distrubuted for the first model and the red line is more horizontal, meaning the third model captures more variation than the previous models. Additionally, in the Normal Q-Q plot, the data points are placed more on the straight line, meaning the data points are fairly evenly distributed. Third, the Scale-Location plot for the third model shows a slightly more even distribution around the red line. Lastly, the Residual vs. Leverage plot for the third models shows a more clustered distribution. Each of these factors combined indicate the third model works best.

i. Predict and Evaluate by Correlation and MSE for the Three Models

```
lm_third <- lm(test$ActualElapsedTime~test$Distance, data=test)
pred <- predict(lm_third, newdata=test)
correlation <- cor(pred, test$ActualElapsedTime)
print(paste("Correlation: ", correlation))

## [1] "Correlation:  0.953252174862133"

mse <- mean((pred - test$ActualElapsedTime)^2)
print(paste("mse: ", mse))

## [1] "mse:  474.326260486443"

rmse <- sqrt(mse)
print(paste("rmse: ", rmse))

## [1] "rmse:  21.7790325883966"
```

As seen above, the correlation is 0.953202682068884, which is very good since it is close to 1. Correlation is used to evaluate how well different columns impact one another, and as discussed before, correlation is scaled on a [-1, 1] range where the close to -1, the more negative the relationship, and the closer to 1, the more positive the relationship (with closer to 0 meanng there does not exist a relationship). Since the correlation shown above is so close to 1, we can say there is a near perfect positive relationship.

mse and rmse are used to quanitfy the amount of error. In isolation, the mse is difficult to interpret; however, as seen above, an rmse of 21.7881277992924 represents how off the test data was on average. This is relatively good rmse given the size of the data, but this, mse, and correlation will be improved slightly in the next section.

```r
lm_third <- lm(test$ActualElapsedTime~test$Distance+test$ArrTime, data=test)
pred <- predict(lm_third, newdata=test)
correlation <- cor(pred, test$ActualElapsedTime)
print(paste("Correlation: ", correlation))
```

```
## [1] "Correlation:  0.953354174325697"
```

```r
mse <- mean((pred - test$ActualElapsedTime)^2)
print(paste("mse: ", mse))
```

```
## [1] "mse:  473.316039550303"
```

```r
rmse <- sqrt(mse)
print(paste("rmse: ", rmse))
```

```
## [1] "rmse:  21.7558277146677"
```

As seen above, the correlation is 0.953302230313854, which is very good since it is close to 1. Correlation is used to evaluate how well different columns impact one another, and as discussed before, correlation is scaled on a [-1, 1] range where the close to -1, the more negative the relationship, and the closer to 1, the more positive the relationship (with closer to 0 meanng there does not exist a relationship). Since the correlation shown above is so close to 1, we can say there is a near perfect positive relationship.

mse and rmse are used to quanitfy the amount of error. In isolation, the mse is difficult to interpret; however, as seen above, an rmse of 21.7654960150852 represents how off the test data was on average. This is relatively good rmse given the size of the data, but this, mse, and correlation will be improved in the next section.

The first two models have very similar correlation, mse, and rmse. This is understandable given how similar the summary and residual plots for these two models were. Both the first and second model had a variety of similarities that placed them into similar categories for how well they could be used to represent a well-rounded linear regression. In contrast, we can see below that the third model, which had a different summary and set of residual plots to the first two models, had very different correlation, mse, and rmse. This is because the third model showed various signs of better representing the data.

```r
lm_third <- lm(test$ActualElapsedTime~test$CRSElapsedTime+test$Distance,
data=test)
pred <- predict(lm_third, newdata=test)
correlation <- cor(pred, test$ActualElapsedTime)
print(paste("Correlation: ", correlation))
```

```
## [1] "Correlation:  0.971409956041144"
```

```r
mse <- mean((pred - test$ActualElapsedTime)^2)
print(paste("mse: ", mse))
```

```
## [1] "mse:  292.78526127892"
```

```r
rmse <- sqrt(mse)
print(paste("rmse: ", rmse))

## [1] "rmse:  17.1109690338952"
```

As seen above, the correlation is 0.971424643867911, which is very good since it is close to 1. Correlation is used to evaluate how well different columns impact one another, and as discussed before, correlation is scaled on a [-1, 1] range where the closer to -1, the more negative the relationship, and the closer to 1, the more positive the relationship (with closer to 0 meaning there does not exist a relationship). Since the correlation shown above is so close to 1, we can say there is a near perfect positive relationship.

mse and rmse are used to quantify the amount of error. In isolaion, the mse is difficult to interpret; however, as seen above, an rmse of 17.1049454210921 represents how off the test data was on average. This is a relatively good sized rmse given the size of the data.

We can see that the correlation, mse, and rmse improved with the third model. As we can see in the three summaries, as we created new linear regression models, the residual standard error, multiple r-squared, and intercept improved overall.