Airline Data Analysis

Mohd Javed

2023-07-25

library(readxl)  
library(ggplot2)  
library(dplyr)  
FlightDt<-read\_excel("2019\_airlinedataset.xlsx")  
#View(FlightDt)  
options(warn=-1)  
ClrDt <- FlightDt[,colSums(is.na(FlightDt))<nrow(FlightDt)]

##### Biggest airports

##---DATA Processing----  
  
BigArpt<-c("ORD","DFW","MCO","JFK","DEN","MIA","EWR","BWI","FLL","DTW")  
#  
FiltrDt<-subset(ClrDt,Dest %in% BigArpt)  
  
  
#summary(FiltrDt)

#we are selecting only based on the data visualization, these are the data visualization.   
finalOrgDt = select(FiltrDt, -c('Year','FlightDate','Reporting\_Airline','DOT\_ID\_Reporting\_Airline','IATA\_CODE\_Reporting\_Airline','Tail\_Number','Flight\_Number\_Reporting\_Airline','OriginCityMarketID','OriginCityName','OriginState','OriginStateFips','OriginStateName','OriginWac','DestCityMarketID','DestCityName','DestState','DestStateFips','DestStateName','DestWac','TaxiOut','WheelsOff','WheelsOn','TaxiIn','ArrTimeBlk','DivAirportLandings', 'DivReachedDest', 'DivActualElapsedTime', 'DivArrDelay','DivDistance', 'Div1Airport', 'Div1AirportID' ,'Div1AirportSeqID','Div1WheelsOn','Div1TotalGTime','Div1LongestGTime','Div1WheelsOff','Div1TailNum'))  
#numerical feilds- correlation  
#find out the codes for summary, descriptive statistics   
  
skimr::skim(finalOrgDt)

Data summary

|  |  |
| --- | --- |
| Name | finalOrgDt |
| Number of rows | 9494 |
| Number of columns | 40 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 4 |
| numeric | 36 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Origin | 0 | 1.00 | 3 | 3 | 0 | 275 | 0 |
| Dest | 0 | 1.00 | 3 | 3 | 0 | 10 | 0 |
| DepTimeBlk | 0 | 1.00 | 9 | 9 | 0 | 19 | 0 |
| CancellationCode | 9306 | 0.02 | 1 | 1 | 0 | 3 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Quarter | 0 | 1.00 | 2.54 | 1.11 | 1 | 2.00 | 3.0 | 4.00 | 4 | ▇▇▁▇▇ |
| Month | 0 | 1.00 | 6.62 | 3.42 | 1 | 4.00 | 7.0 | 10.00 | 12 | ▇▅▆▅▇ |
| DayofMonth | 0 | 1.00 | 15.71 | 8.71 | 1 | 8.00 | 16.0 | 23.00 | 31 | ▇▇▇▇▆ |
| DayOfWeek | 0 | 1.00 | 3.97 | 2.01 | 1 | 2.00 | 4.0 | 6.00 | 7 | ▇▃▃▅▇ |
| OriginAirportID | 0 | 1.00 | 12701.71 | 1570.40 | 10135 | 11274.00 | 12892.0 | 14100.00 | 16218 | ▇▅▅▇▂ |
| OriginAirportSeqID | 0 | 1.00 | 1270174.48 | 157039.48 | 1013505 | 1127402.00 | 1289208.0 | 1410005.00 | 1621802 | ▇▅▅▇▂ |
| DestAirportID | 0 | 1.00 | 12179.73 | 1095.84 | 10821 | 11298.00 | 11618.0 | 13204.00 | 13930 | ▇▂▁▂▃ |
| DestAirportSeqID | 0 | 1.00 | 1217977.71 | 109584.42 | 1082106 | 1129806.00 | 1161802.0 | 1320402.00 | 1393007 | ▇▂▁▂▃ |
| CRSDepTime | 0 | 1.00 | 1272.29 | 497.75 | 10 | 821.00 | 1245.0 | 1700.00 | 2359 | ▁▇▇▇▃ |
| DepTime | 182 | 0.98 | 1275.50 | 514.28 | 1 | 822.00 | 1251.0 | 1709.00 | 2400 | ▁▇▇▇▃ |
| DepDelay | 182 | 0.98 | 13.00 | 58.66 | -24 | -6.00 | -2.0 | 7.00 | 1855 | ▇▁▁▁▁ |
| DepDelayMinutes | 182 | 0.98 | 16.35 | 57.60 | 0 | 0.00 | 0.0 | 7.00 | 1855 | ▇▁▁▁▁ |
| DepDel15 | 182 | 0.98 | 0.20 | 0.40 | 0 | 0.00 | 0.0 | 0.00 | 1 | ▇▁▁▁▂ |
| DepartureDelayGroups | 182 | 0.98 | 0.17 | 2.47 | -2 | -1.00 | -1.0 | 0.00 | 12 | ▇▁▁▁▁ |
| CRSArrTime | 0 | 1.00 | 1437.75 | 524.31 | 1 | 1018.00 | 1451.0 | 1900.00 | 2400 | ▁▆▇▇▆ |
| ArrTime | 192 | 0.98 | 1418.53 | 545.24 | 1 | 1003.00 | 1440.5 | 1859.00 | 2400 | ▁▆▇▇▆ |
| ArrDelay | 227 | 0.98 | 7.33 | 60.91 | -62 | -16.00 | -7.0 | 8.00 | 1847 | ▇▁▁▁▁ |
| ArrDelayMinutes | 227 | 0.98 | 16.48 | 57.52 | 0 | 0.00 | 0.0 | 8.00 | 1847 | ▇▁▁▁▁ |
| ArrDel15 | 227 | 0.98 | 0.20 | 0.40 | 0 | 0.00 | 0.0 | 0.00 | 1 | ▇▁▁▁▂ |
| ArrivalDelayGroups | 227 | 0.98 | -0.11 | 2.63 | -2 | -2.00 | -1.0 | 0.00 | 12 | ▇▁▁▁▁ |
| Cancelled | 0 | 1.00 | 0.02 | 0.14 | 0 | 0.00 | 0.0 | 0.00 | 1 | ▇▁▁▁▁ |
| Diverted | 0 | 1.00 | 0.00 | 0.06 | 0 | 0.00 | 0.0 | 0.00 | 1 | ▇▁▁▁▁ |
| CRSElapsedTime | 0 | 1.00 | 153.63 | 63.22 | 45 | 107.00 | 145.0 | 183.00 | 585 | ▇▆▁▁▁ |
| ActualElapsedTime | 227 | 0.98 | 148.29 | 63.48 | 30 | 102.00 | 140.0 | 179.00 | 620 | ▇▆▁▁▁ |
| AirTime | 227 | 0.98 | 120.79 | 61.82 | 14 | 76.00 | 114.0 | 150.00 | 590 | ▇▅▁▁▁ |
| Flights | 0 | 1.00 | 1.00 | 0.00 | 1 | 1.00 | 1.0 | 1.00 | 1 | ▁▁▇▁▁ |
| Distance | 0 | 1.00 | 880.70 | 560.19 | 67 | 483.00 | 801.0 | 1095.00 | 4983 | ▇▃▁▁▁ |
| DistanceGroup | 0 | 1.00 | 3.98 | 2.19 | 1 | 2.00 | 4.0 | 5.00 | 11 | ▇▆▂▁▁ |
| CarrierDelay | 7641 | 0.20 | 21.94 | 79.66 | 0 | 0.00 | 0.0 | 14.00 | 1154 | ▇▁▁▁▁ |
| WeatherDelay | 7641 | 0.20 | 3.79 | 47.64 | 0 | 0.00 | 0.0 | 0.00 | 1847 | ▇▁▁▁▁ |
| NASDelay | 7641 | 0.20 | 24.66 | 55.44 | 0 | 0.00 | 7.0 | 27.00 | 1194 | ▇▁▁▁▁ |
| SecurityDelay | 7641 | 0.20 | 0.03 | 0.56 | 0 | 0.00 | 0.0 | 0.00 | 15 | ▇▁▁▁▁ |
| LateAircraftDelay | 7641 | 0.20 | 27.13 | 52.41 | 0 | 0.00 | 0.0 | 31.00 | 429 | ▇▁▁▁▁ |
| FirstDepTime | 9410 | 0.01 | 1223.43 | 527.08 | 153 | 752.75 | 1109.0 | 1703.00 | 2232 | ▂▇▇▅▅ |
| TotalAddGTime | 9410 | 0.01 | 40.35 | 38.14 | 1 | 17.00 | 29.0 | 45.25 | 196 | ▇▂▁▁▁ |
| LongestAddGTime | 9410 | 0.01 | 40.35 | 38.14 | 1 | 17.00 | 29.0 | 45.25 | 196 | ▇▂▁▁▁ |

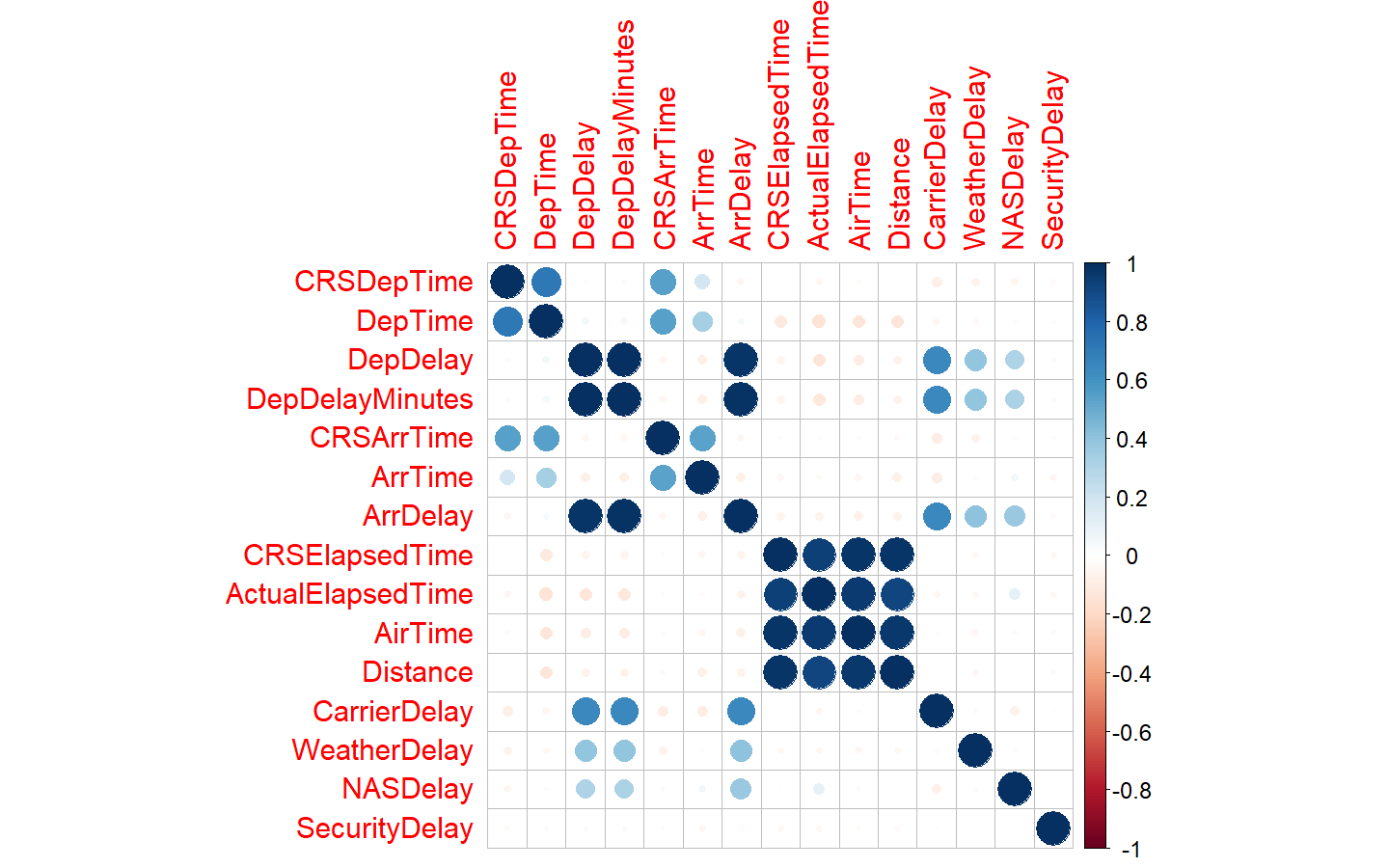
#we are dropping these variables because its identical  
finalOrgDt$OriginAirportSeqID<-NULL  
finalOrgDt$DestAirportSeqID<-NULL  
  
#Here we are taking our output variables   
FltonTime\_Dt<-finalOrgDt[!is.na(finalOrgDt$ArrDel15)&finalOrgDt$ArrDel15!=""&!is.na(finalOrgDt$DepDel15)&finalOrgDt$DepDel15!="",]  
#kitne the aur kitne ho gaye?  
nrow(finalOrgDt)

## [1] 9494

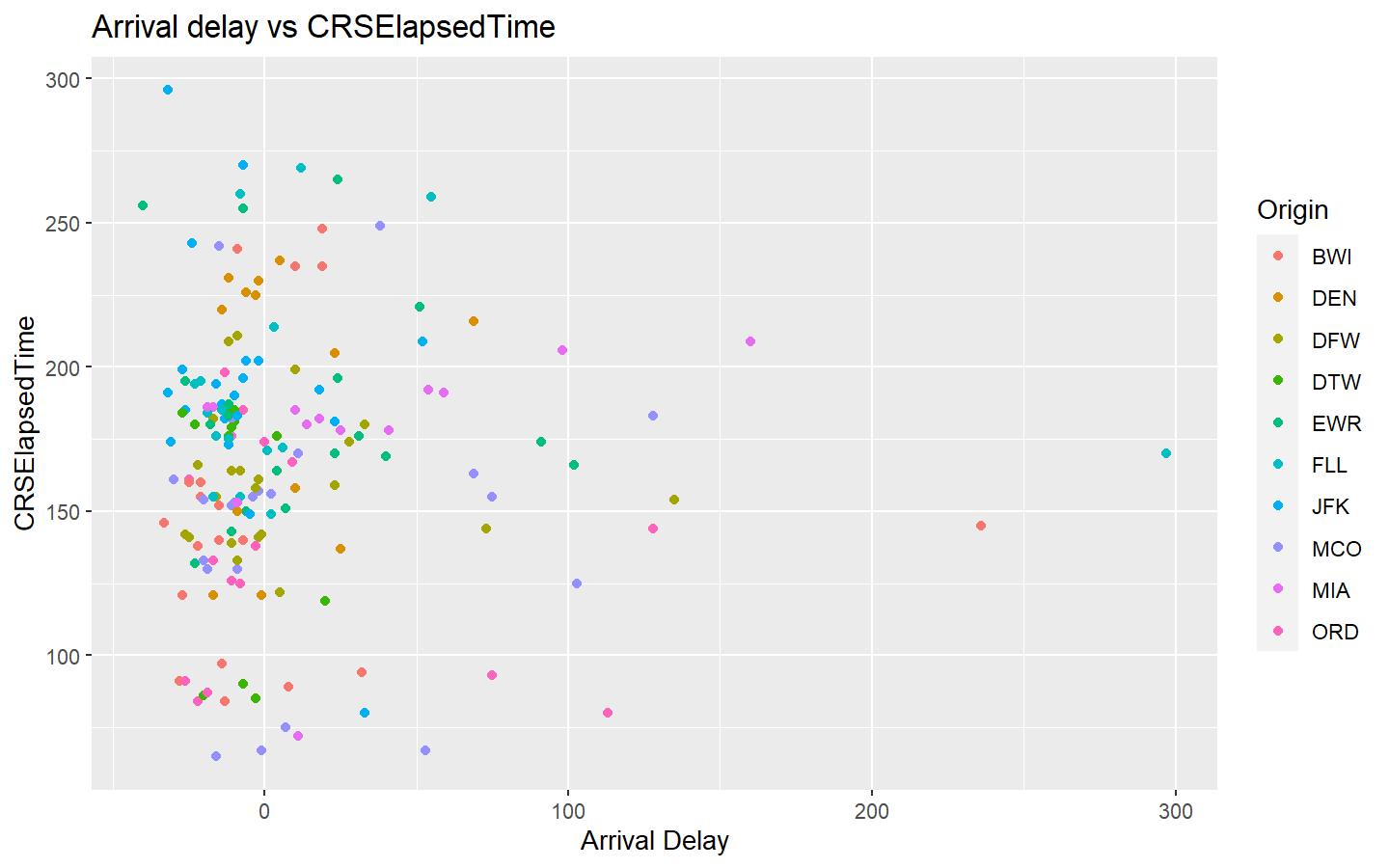
nrow(FltonTime\_Dt)

## [1] 9267

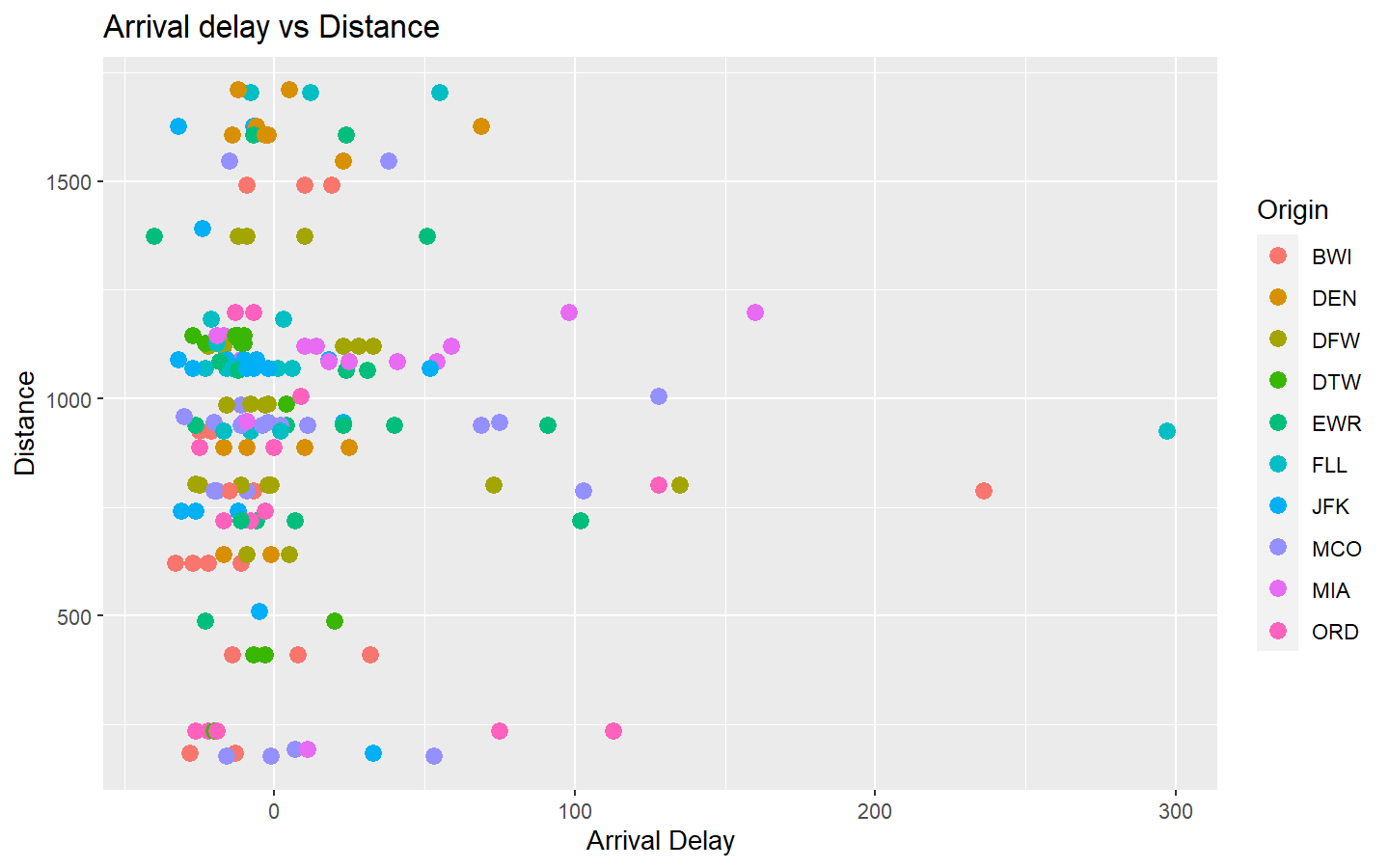
final\_corr = select(FltonTime\_Dt, c("CRSDepTime","DepTime","DepDelay","DepDelayMinutes","CRSArrTime","ArrTime","ArrDelay","CRSElapsedTime","ActualElapsedTime","AirTime","Distance","CarrierDelay","WeatherDelay","NASDelay","SecurityDelay"))  
final\_corr<-na.omit(final\_corr)  
library(corrplot)  
# correlation matrix  
corrplot(cor(final\_corr))



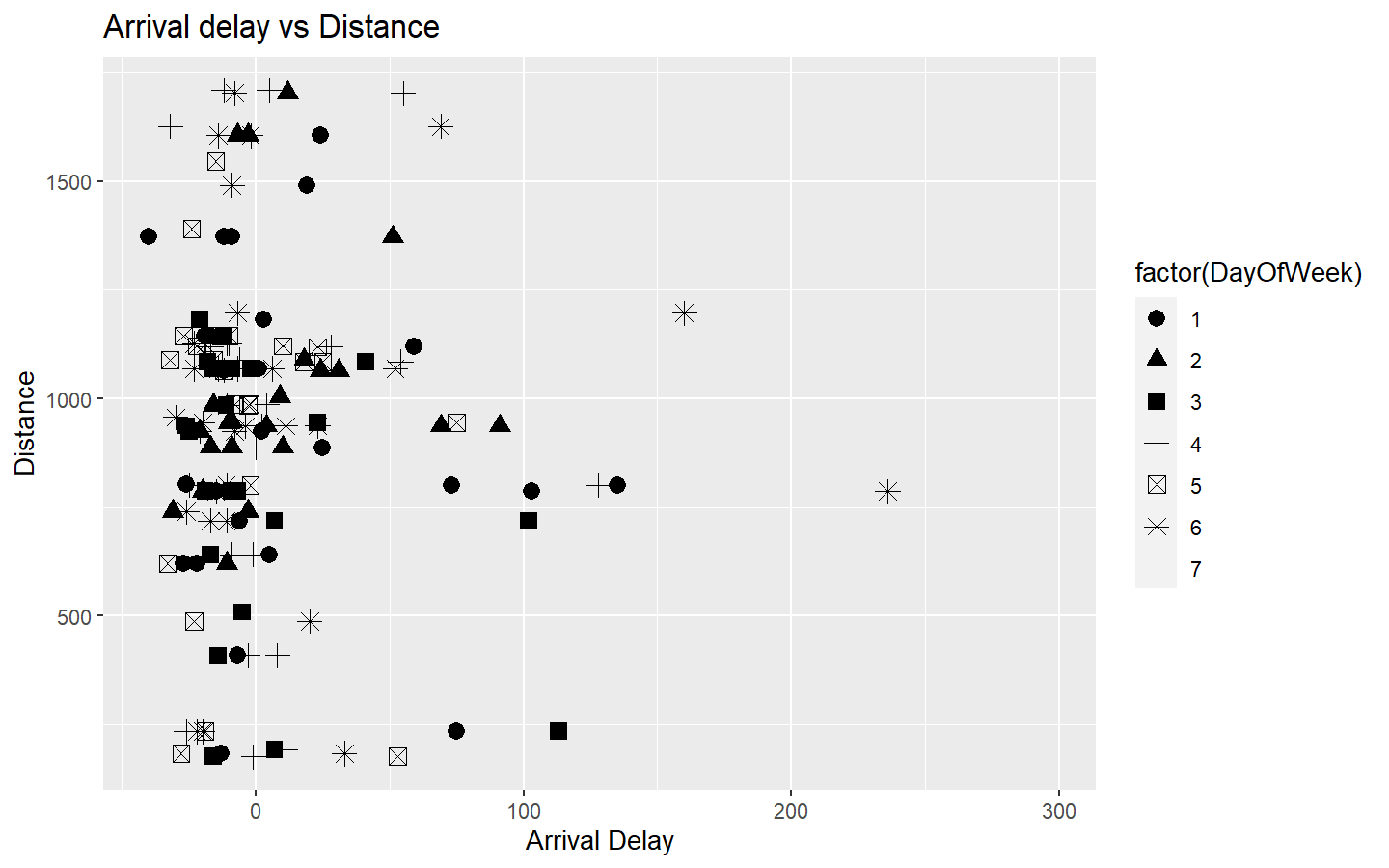
FltonTime\_Dt1<-FltonTime\_Dt%>%  
filter(Origin==c("ORD","DFW","MCO","JFK","DEN","MIA","EWR","BWI","FLL","DTW"))  
ggplot(FltonTime\_Dt1, aes(x=ArrDelay,y=CRSElapsedTime,color=Origin)) +  
 geom\_point()+  
 labs (title = "Arrival delay vs CRSElapsedTime", x = "Arrival Delay", y = "CRSElapsedTime")+  
 theme\_grey()



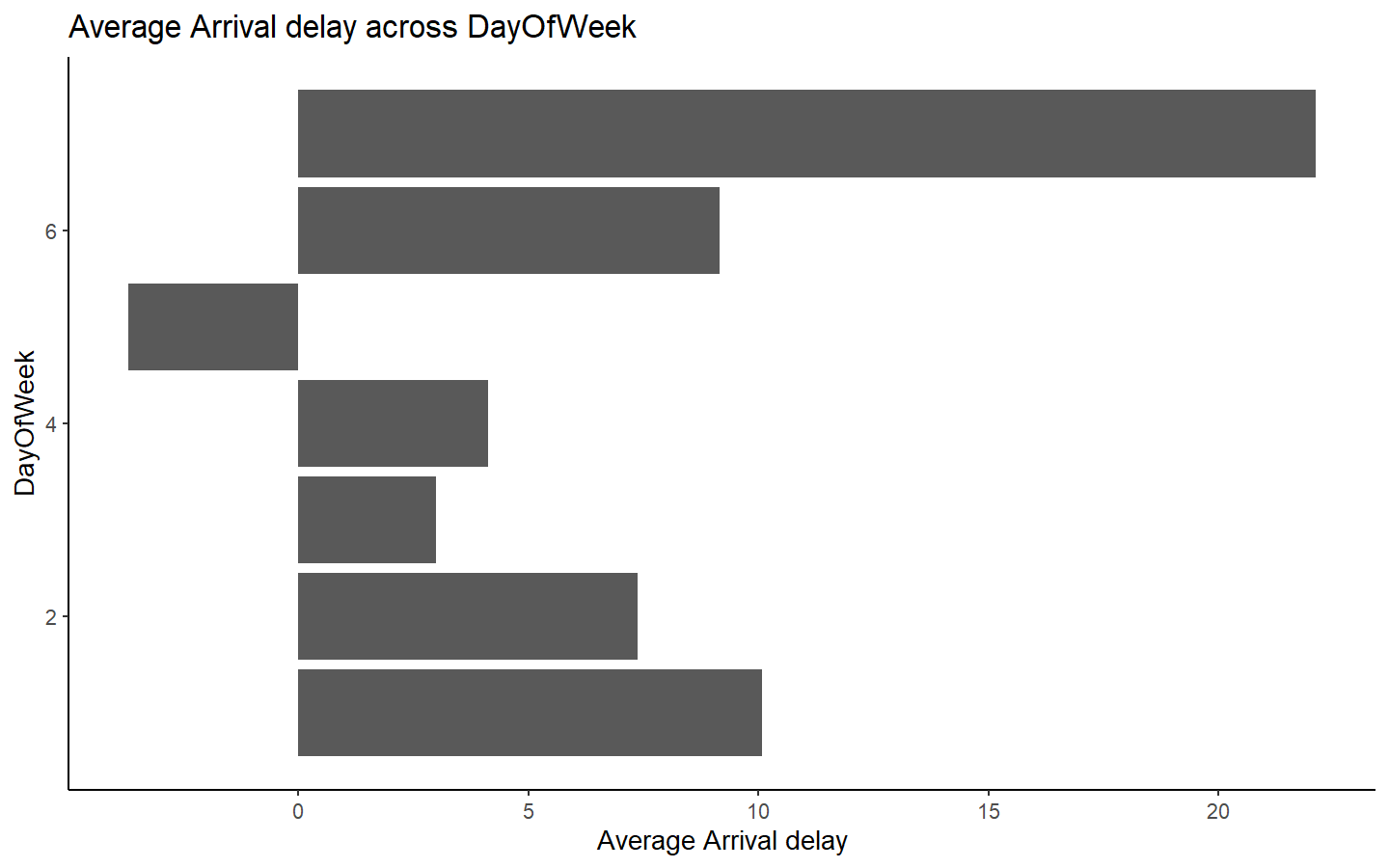
ggplot(FltonTime\_Dt1, aes(x=ArrDelay,y=Distance,color=Origin)) +  
 geom\_point(size=3)+  
 labs (title = "Arrival delay vs Distance", x = "Arrival Delay", y = "Distance")+  
 theme\_grey()



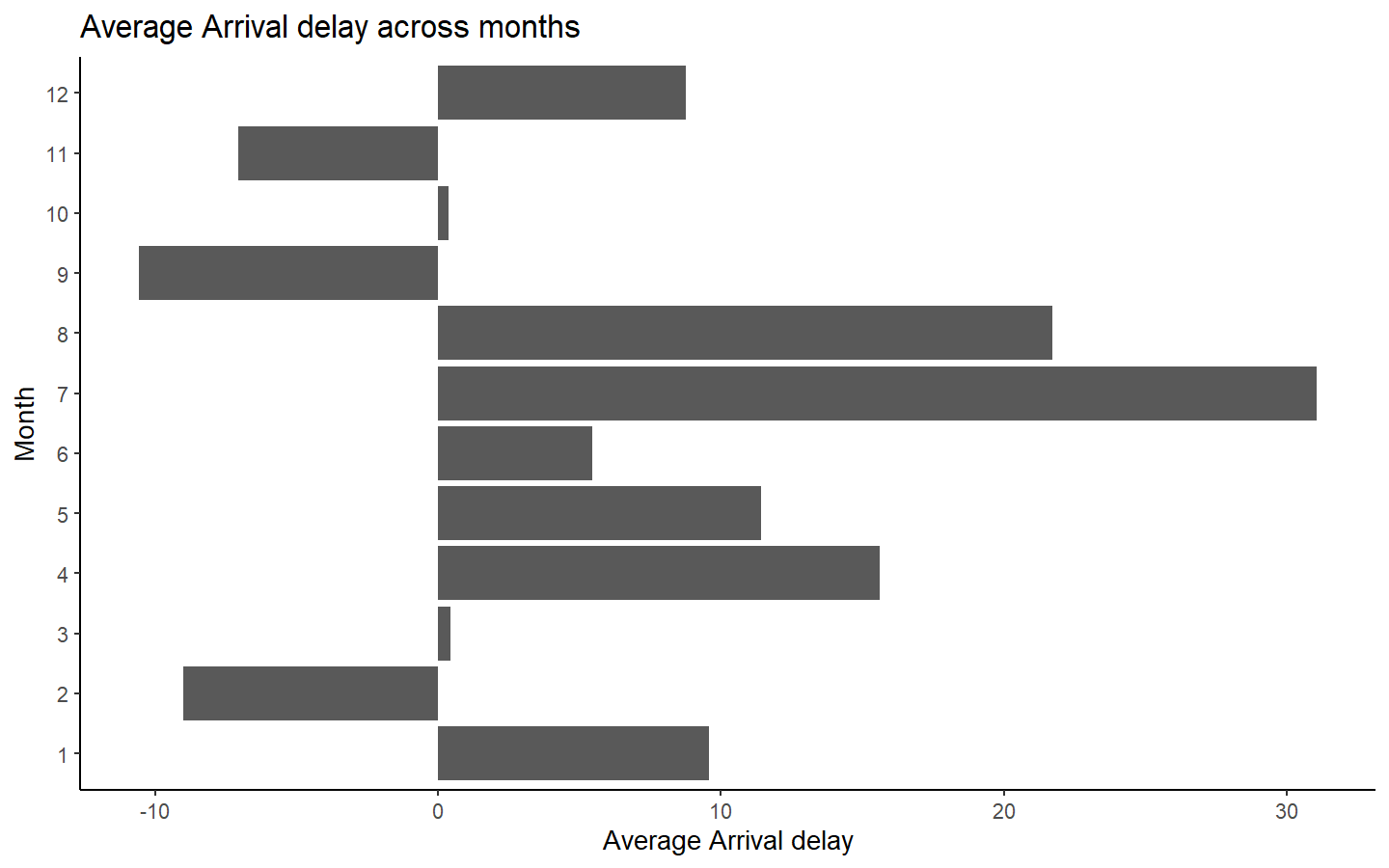
ggplot(FltonTime\_Dt1, aes(x=ArrDelay,y=Distance)) +  
 geom\_point(size=3,aes(shape = factor(DayOfWeek)))+  
 labs (title = "Arrival delay vs Distance", x = "Arrival Delay", y = "Distance")+  
 theme\_grey()



ggplot(FltonTime\_Dt1, aes(DayOfWeek, ArrDelay)) + geom\_bar(position = "dodge", stat = "summary", fun = "mean")+  
 coord\_flip()+  
 labs (title = "Average Arrival delay across DayOfWeek", x = "DayOfWeek", y = "Average Arrival delay")+  
 theme\_classic()



ggplot(FltonTime\_Dt1, aes(factor(Month), ArrDelay)) + geom\_bar(position = "dodge", stat = "summary", fun = "mean")+  
 coord\_flip()+  
 labs (title = "Average Arrival delay across months", x = "Month", y = "Average Arrival delay")+  
 theme\_classic()



#converting string to integer (why?)  
FltonTime\_Dt$Distance<-as.integer(FltonTime\_Dt$Distance)  
FltonTime\_Dt$Cancelled<-as.integer(FltonTime\_Dt$Cancelled)  
FltonTime\_Dt$Diverted<-as.integer(FltonTime\_Dt$Diverted)

#converting string to factors for modelling   
FltonTime\_Dt$ArrDel15<-as.factor(FltonTime\_Dt$ArrDel15)  
FltonTime\_Dt$DepDel15<-as.factor(FltonTime\_Dt$DepDel15)  
FltonTime\_Dt$DestAirportID<-as.factor(FltonTime\_Dt$DestAirportID)  
FltonTime\_Dt$OriginAirportID<-as.factor(FltonTime\_Dt$OriginAirportID)  
FltonTime\_Dt$DayOfWeek<-as.factor(FltonTime\_Dt$DayOfWeek)  
FltonTime\_Dt$Dest<-as.factor(FltonTime\_Dt$Dest)  
FltonTime\_Dt$Origin<-as.factor(FltonTime\_Dt$Origin)  
FltonTime\_Dt$DepTimeBlk<-as.factor(FltonTime\_Dt$DepTimeBlk)

skimr::skim(FltonTime\_Dt)

Data summary

|  |  |
| --- | --- |
| Name | FltonTime\_Dt |
| Number of rows | 9267 |
| Number of columns | 38 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 1 |
| factor | 8 |
| numeric | 29 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CancellationCode | 9267 | 0 | NA | NA | 0 | 0 | 0 |

**Variable type: factor**

| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| --- | --- | --- | --- | --- | --- |
| DayOfWeek | 0 | 1 | FALSE | 7 | 5: 1377, 1: 1372, 3: 1350, 7: 1345 |
| OriginAirportID | 0 | 1 | FALSE | 275 | 103: 358, 128: 265, 129: 261, 107: 246 |
| Origin | 0 | 1 | FALSE | 275 | ATL: 358, LAX: 265, LGA: 261, BOS: 246 |
| DestAirportID | 0 | 1 | FALSE | 10 | 139: 1788, 112: 1613, 112: 1353, 114: 885 |
| Dest | 0 | 1 | FALSE | 10 | ORD: 1788, DFW: 1613, DEN: 1353, DTW: 885 |
| DepDel15 | 0 | 1 | FALSE | 2 | 0: 7452, 1: 1815 |
| DepTimeBlk | 0 | 1 | FALSE | 19 | 060: 933, 070: 700, 120: 670, 170: 577 |
| ArrDel15 | 0 | 1 | FALSE | 2 | 0: 7414, 1: 1853 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Quarter | 0 | 1.00 | 2.55 | 1.11 | 1 | 2.0 | 3 | 4 | 4 | ▇▇▁▇▇ |
| Month | 0 | 1.00 | 6.65 | 3.42 | 1 | 4.0 | 7 | 10 | 12 | ▇▅▆▅▇ |
| DayofMonth | 0 | 1.00 | 15.67 | 8.72 | 1 | 8.0 | 16 | 23 | 31 | ▇▇▇▇▆ |
| CRSDepTime | 0 | 1.00 | 1270.85 | 498.54 | 10 | 820.0 | 1245 | 1700 | 2359 | ▁▇▇▇▃ |
| DepTime | 0 | 1.00 | 1274.59 | 514.49 | 1 | 821.0 | 1250 | 1709 | 2400 | ▁▇▇▇▃ |
| DepDelay | 0 | 1.00 | 12.85 | 58.49 | -24 | -6.0 | -2 | 7 | 1855 | ▇▁▁▁▁ |
| DepDelayMinutes | 0 | 1.00 | 16.20 | 57.42 | 0 | 0.0 | 0 | 7 | 1855 | ▇▁▁▁▁ |
| DepartureDelayGroups | 0 | 1.00 | 0.16 | 2.46 | -2 | -1.0 | -1 | 0 | 12 | ▇▁▁▁▁ |
| CRSArrTime | 0 | 1.00 | 1436.07 | 524.72 | 1 | 1015.0 | 1450 | 1858 | 2400 | ▁▆▇▇▆ |
| ArrTime | 0 | 1.00 | 1417.78 | 544.33 | 1 | 1003.0 | 1440 | 1858 | 2400 | ▁▆▇▇▆ |
| ArrDelay | 0 | 1.00 | 7.33 | 60.91 | -62 | -16.0 | -7 | 8 | 1847 | ▇▁▁▁▁ |
| ArrDelayMinutes | 0 | 1.00 | 16.48 | 57.52 | 0 | 0.0 | 0 | 8 | 1847 | ▇▁▁▁▁ |
| ArrivalDelayGroups | 0 | 1.00 | -0.11 | 2.63 | -2 | -2.0 | -1 | 0 | 12 | ▇▁▁▁▁ |
| Cancelled | 0 | 1.00 | 0.00 | 0.00 | 0 | 0.0 | 0 | 0 | 0 | ▁▁▇▁▁ |
| Diverted | 0 | 1.00 | 0.00 | 0.00 | 0 | 0.0 | 0 | 0 | 0 | ▁▁▇▁▁ |
| CRSElapsedTime | 0 | 1.00 | 153.80 | 63.34 | 45 | 107.0 | 146 | 183 | 585 | ▇▆▁▁▁ |
| ActualElapsedTime | 0 | 1.00 | 148.29 | 63.48 | 30 | 102.0 | 140 | 179 | 620 | ▇▆▁▁▁ |
| AirTime | 0 | 1.00 | 120.79 | 61.82 | 14 | 76.0 | 114 | 150 | 590 | ▇▅▁▁▁ |
| Flights | 0 | 1.00 | 1.00 | 0.00 | 1 | 1.0 | 1 | 1 | 1 | ▁▁▇▁▁ |
| Distance | 0 | 1.00 | 882.83 | 561.66 | 67 | 484.5 | 802 | 1096 | 4983 | ▇▃▁▁▁ |
| DistanceGroup | 0 | 1.00 | 3.99 | 2.19 | 1 | 2.0 | 4 | 5 | 11 | ▇▆▂▁▁ |
| CarrierDelay | 7414 | 0.20 | 21.94 | 79.66 | 0 | 0.0 | 0 | 14 | 1154 | ▇▁▁▁▁ |
| WeatherDelay | 7414 | 0.20 | 3.79 | 47.64 | 0 | 0.0 | 0 | 0 | 1847 | ▇▁▁▁▁ |
| NASDelay | 7414 | 0.20 | 24.66 | 55.44 | 0 | 0.0 | 7 | 27 | 1194 | ▇▁▁▁▁ |
| SecurityDelay | 7414 | 0.20 | 0.03 | 0.56 | 0 | 0.0 | 0 | 0 | 15 | ▇▁▁▁▁ |
| LateAircraftDelay | 7414 | 0.20 | 27.13 | 52.41 | 0 | 0.0 | 0 | 31 | 429 | ▇▁▁▁▁ |
| FirstDepTime | 9190 | 0.01 | 1212.00 | 533.40 | 153 | 749.0 | 1039 | 1701 | 2232 | ▂▇▇▅▅ |
| TotalAddGTime | 9190 | 0.01 | 38.29 | 34.98 | 1 | 17.0 | 29 | 41 | 195 | ▇▂▁▁▁ |
| LongestAddGTime | 9190 | 0.01 | 38.29 | 34.98 | 1 | 17.0 | 29 | 41 | 195 | ▇▂▁▁▁ |

FltonTime\_Dt<-FltonTime\_Dt[,c(1:21)]

nrow(finalOrgDt)

## [1] 9494

nrow(FltonTime\_Dt)

## [1] 9267

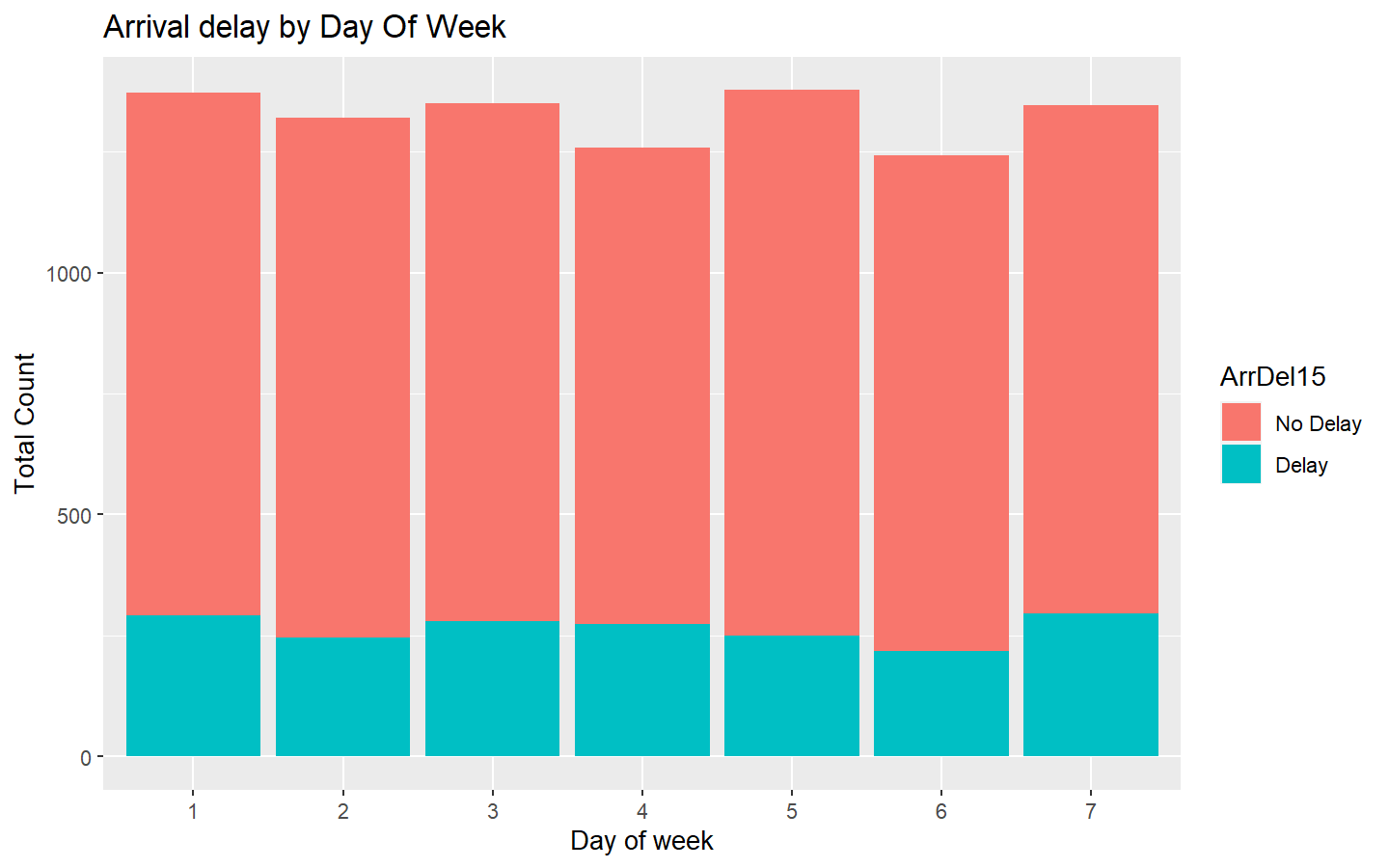
#See how many delays vs. non-delays occur in the data - this will help us understand if we have enough data to build prediction.   
tapply(FltonTime\_Dt$ArrDel15,FltonTime\_Dt$ArrDel15,length)

## 0 1   
## 7414 1853

library(caret)  
set.seed(123)  
#Feature Selection  
featureSelectionCol<-c("ArrDel15","DayOfWeek","Dest","Origin")  
FltonTime\_DtFiltered<-FltonTime\_Dt[,featureSelectionCol]  
levels(FltonTime\_DtFiltered$ArrDel15)<-c("No Delay","Delay")

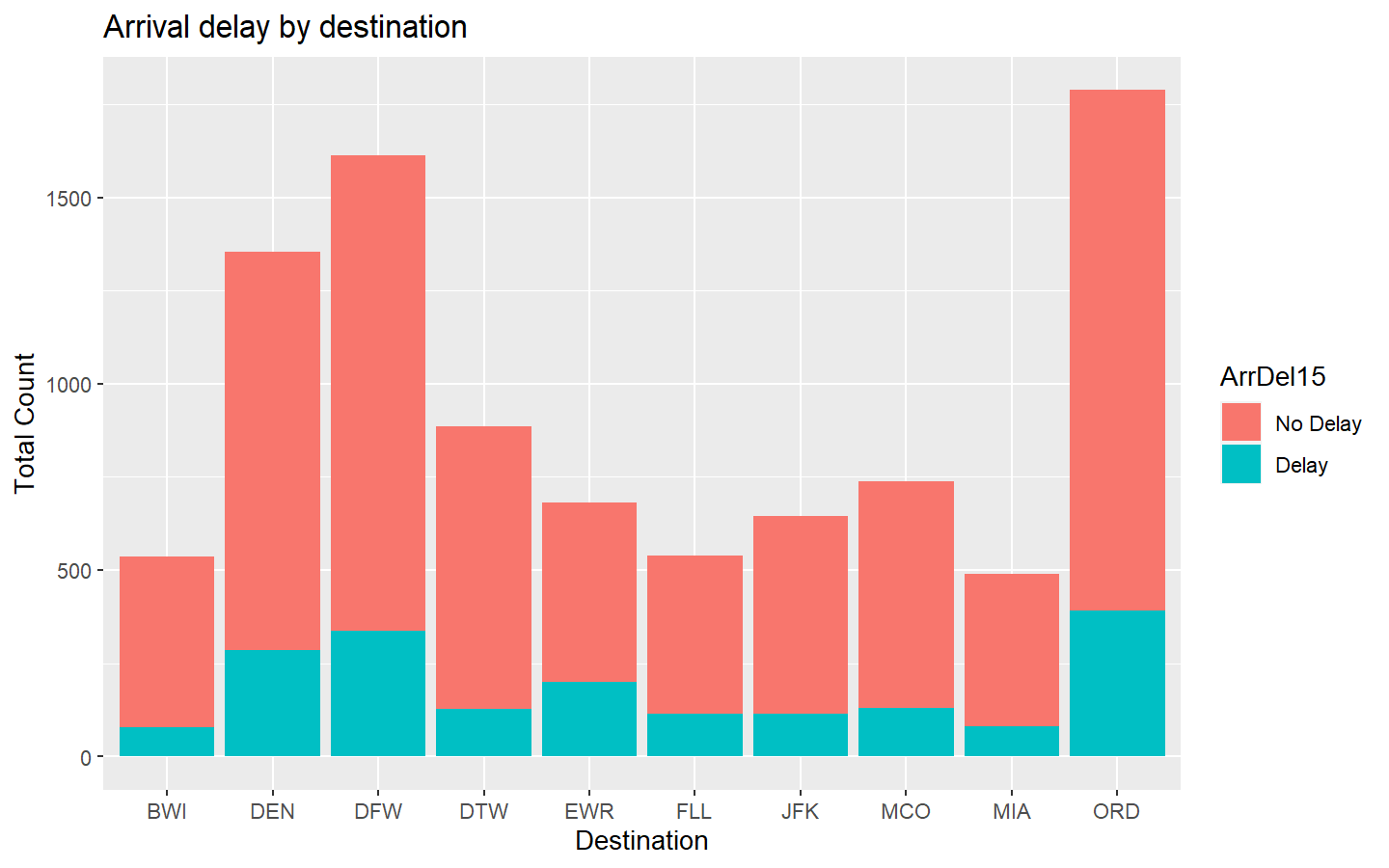
### Arrival delay by dat of week

ggplot(data=FltonTime\_DtFiltered, aes(x=DayOfWeek, fill=ArrDel15)) +  
geom\_bar() +  
 labs (title = "Arrival delay by Day Of Week", x = "Day of week", y = "Total Count")

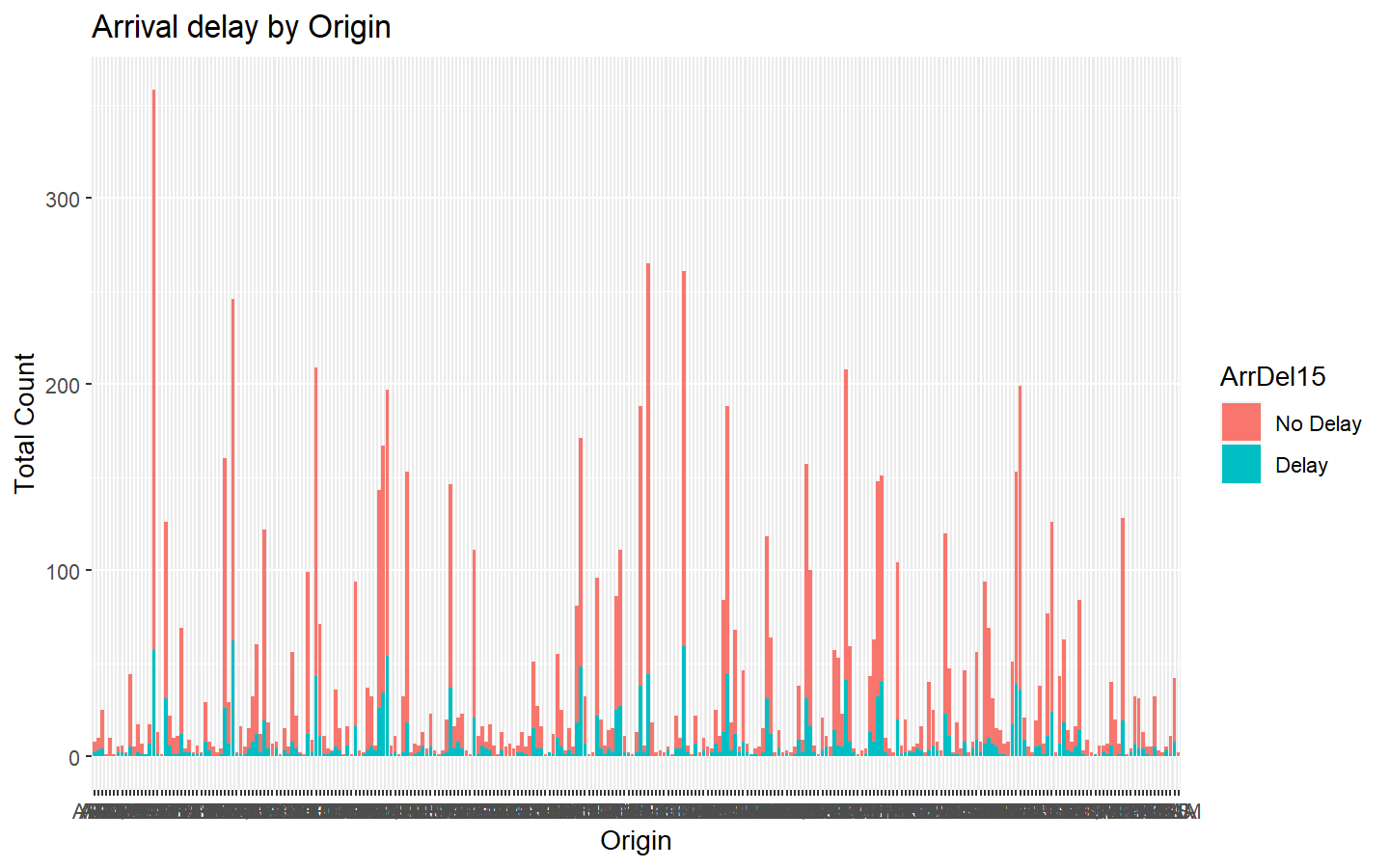


### Arrival delay by destination

ggplot(data=FltonTime\_DtFiltered, aes(x=Dest, fill=ArrDel15)) +  
geom\_bar() +  
 labs (title = "Arrival delay by destination", x = "Destination", y = "Total Count")



ggplot(data=FltonTime\_DtFiltered, aes(x=Origin, fill=ArrDel15)) +  
geom\_bar() +  
 labs (title = "Arrival delay by Origin", x = "Origin", y = "Total Count")



FltonTime\_DtFiltered1<-FltonTime\_DtFiltered%>%  
filter(Origin==c("ORD","DFW","MCO","JFK","DEN","MIA","EWR","BWI","FLL","DTW"))  
set.seed(7)  
#75 training 25 testing   
Dt\_Partisioning<-createDataPartition(y=FltonTime\_DtFiltered1$ArrDel15, times = 1,p=0.75, list= FALSE)

#after spliting   
TrainDt1<-FltonTime\_DtFiltered1[Dt\_Partisioning,]  
TestDt1<-FltonTime\_DtFiltered1[-Dt\_Partisioning,]

#checcking the spliting of data  
nrow(TrainDt1)/(nrow(TestDt1)+nrow(TrainDt1))

## [1] 0.7556818

nrow(TestDt1)/(nrow(TestDt1)+nrow(TrainDt1))

## [1] 0.2443182

#TestDt1$ArrDel15<-ifelse(TestDt1$ArrDel15==1,0,1)

#logistic regression model- put in last   
logisticRegModel<-glm(ArrDel15~.,data=TrainDt1,family="binomial")  
logRegPrediction<-predict(logisticRegModel,TestDt1)  
summary(logisticRegModel)

##   
## Call:  
## glm(formula = ArrDel15 ~ ., family = "binomial", data = TrainDt1)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.51911 -0.72226 -0.44236 -0.00007 2.36020   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.15319 1.25028 -0.123 0.9025   
## DayOfWeek2 -0.86037 0.89171 -0.965 0.3346   
## DayOfWeek3 -1.48269 1.03110 -1.438 0.1504   
## DayOfWeek4 -0.58187 0.92552 -0.629 0.5296   
## DayOfWeek5 -0.12154 0.82565 -0.147 0.8830   
## DayOfWeek6 -1.06214 0.86195 -1.232 0.2179   
## DayOfWeek7 0.19199 0.78676 0.244 0.8072   
## DestDEN -0.90282 1.24444 -0.725 0.4682   
## DestDFW -1.62262 1.37948 -1.176 0.2395   
## DestDTW -1.54972 1.35819 -1.141 0.2539   
## DestEWR -1.76295 1.24482 -1.416 0.1567   
## DestFLL -0.94103 1.20000 -0.784 0.4329   
## DestJFK -2.65915 1.47698 -1.800 0.0718 .  
## DestMCO 0.29242 1.24526 0.235 0.8143   
## DestMIA -2.13525 1.51352 -1.411 0.1583   
## DestORD -0.64456 1.11966 -0.576 0.5648   
## OriginDEN 0.40043 1.14177 0.351 0.7258   
## OriginDFW 0.05727 0.88489 0.065 0.9484   
## OriginDTW -15.95301 1169.49782 -0.014 0.9891   
## OriginEWR 1.49605 0.92780 1.612 0.1069   
## OriginFLL -0.40723 1.22567 -0.332 0.7397   
## OriginJFK 0.42724 0.96296 0.444 0.6573   
## OriginMCO 0.70575 0.97792 0.722 0.4705   
## OriginMIA 2.30047 1.04450 2.202 0.0276 \*  
## OriginORD 0.76122 1.12433 0.677 0.4984   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 149.03 on 132 degrees of freedom  
## Residual deviance: 119.35 on 108 degrees of freedom  
## AIC: 169.35  
##   
## Number of Fisher Scoring iterations: 16

##############convert numeric to binary  
predicted\_delay<-ifelse(logRegPrediction>0.5,1,0)  
table(predicted\_delay,TestDt1$ArrDel15)

##   
## predicted\_delay No Delay Delay  
## 0 29 10  
## 1 4 0

#Accuracy  
mean(predicted\_delay==TestDt1$ArrDel15)

## [1] 0

#random forest- 1st model   
library(randomForest)  
  
RandomForModel <- train(ArrDel15 ~ .,  
 data = TrainDt1,  
 method = 'rf',  
 trControl = trainControl(method = "cv", number = 10),  
 preproc = c("center", "scale"))  
  
  
RandomForPredict <- predict(RandomForModel, newdata = TestDt1)  
table(RandomForPredict,TestDt1$ArrDel15)

##   
## RandomForPredict No Delay Delay  
## No Delay 33 10  
## Delay 0 0

#Accuracy  
mean(RandomForPredict==TestDt1$ArrDel15)

## [1] 0.7674419

library(e1071)  
  
svmforModel <- svm(ArrDel15 ~ .,  
 data = TrainDt1,  
 type = 'C-classification',  
 kernel = 'sigmoid',gamma=0.25,cost=1,coef.0=0)

svmforPredict <- predict(svmforModel,  
 newdata = TestDt1)  
  
#ConfMtx3<-confusionMatrix(svmforPredict,reference = TestDt$ArrDel15)  
#ConfMtx3  
table(svmforPredict,TestDt1$ArrDel15)

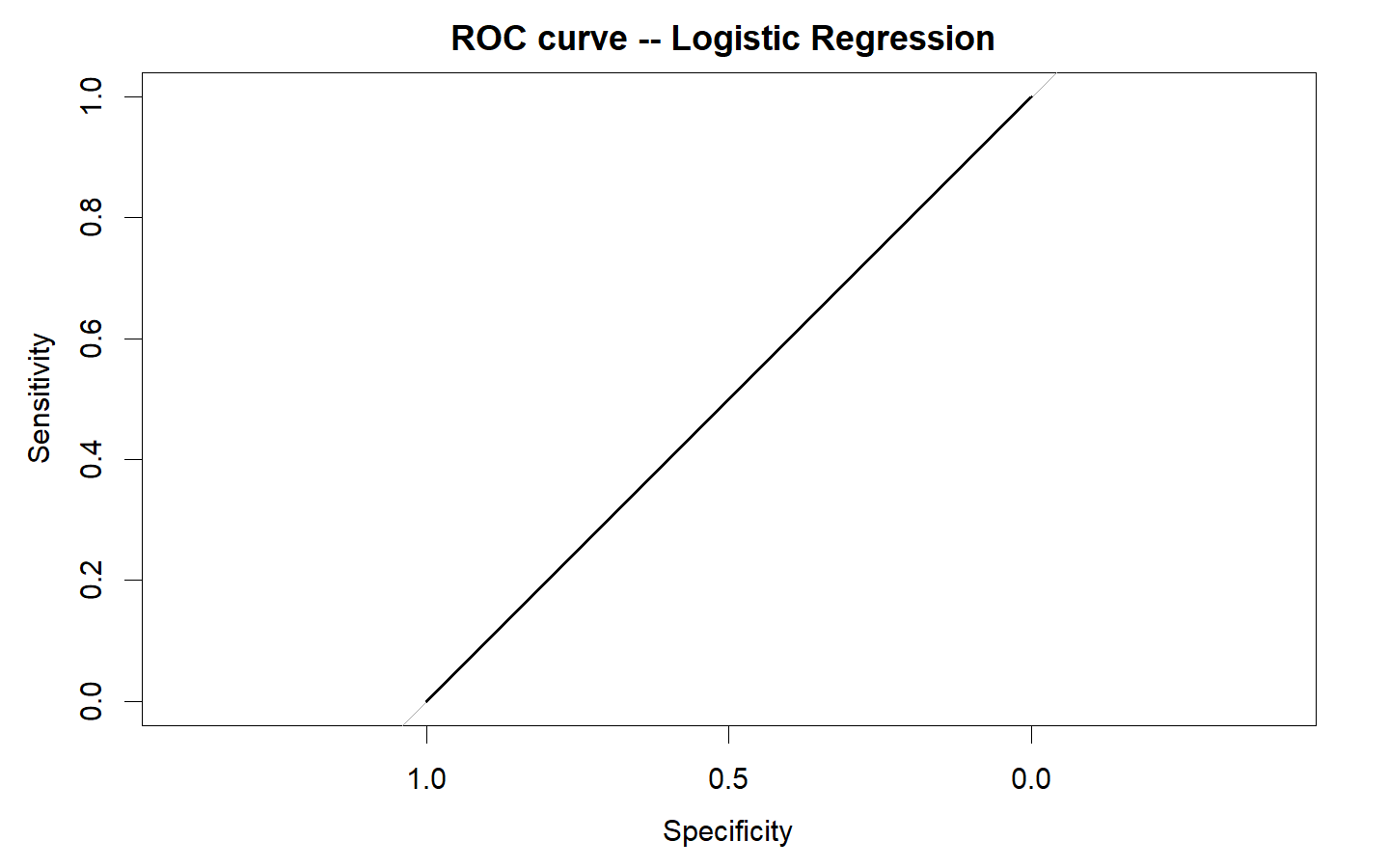
##   
## svmforPredict No Delay Delay  
## No Delay 33 10  
## Delay 0 0

#Accuracy  
mean(svmforPredict==TestDt1$ArrDel15)

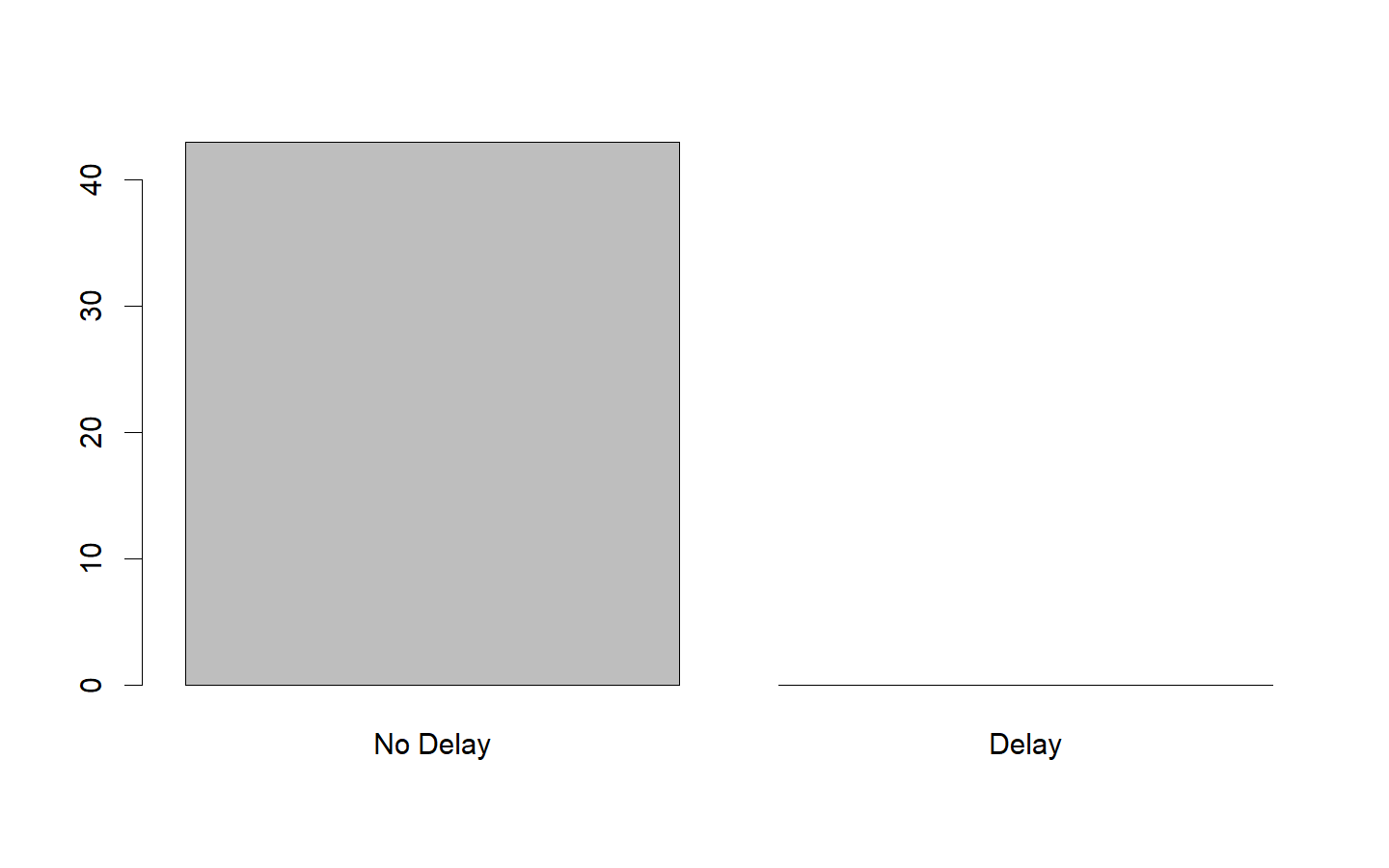
## [1] 0.7674419

#install.packages("pROC")  
library(pROC)  
library(ggplot2)  
library(InformationValue)

TestDt1$ArrDel15<-as.numeric(TestDt1$ArrDel15)  
pred\_svm<-as.numeric(svmforPredict)  
#plotROC(pred\_svm,TestDt$ArrDel15)  
roc\_score=roc(TestDt1$ArrDel15, pred\_svm) #AUC score  
plot(roc\_score ,main ="ROC curve -- Logistic Regression ")



plot(svmforPredict)



KNNforModel<- train(ArrDel15~., data = TrainDt1, method = "knn", metric="Accuracy",na.action = na.exclude)  
print(KNNforModel)

## k-Nearest Neighbors   
##   
## 133 samples  
## 3 predictor  
## 2 classes: 'No Delay', 'Delay'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 133, 133, 133, 133, 133, 133, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 5 0.7140676 0.06949619  
## 7 0.7176109 0.02158997  
## 9 0.7325360 0.04353921  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 9.

KNNPredict<-predict(KNNforModel, newdata = TestDt1)  
confusionMatrix(KNNPredict, TestDt1$ArrDel15)

## No Delay Delay  
## 1 41 2

table(KNNPredict, TestDt1$ArrDel15)

##   
## KNNPredict 1 2  
## No Delay 31 10  
## Delay 2 0

#Accuracy  
mean(KNNPredict==TestDt1$ArrDel15)

## [1] 0

#-----CHECK HOW CAN YOU INCREASE ACCURACY OF MODELS-------  
# adding more data records  
# adding more relevant features  
# trying more Classification algorithm  
#