Data Mining Final Project

8/17/2018

Can load all data frames/model objects built in this analysis by running the below code-chunk

```
#load('DM_Project_vF')
```

-Setup and Data Prep

```
library(lubridate)
library(tidyverse)
library(poLCA)
library(MASS)
library(dplyr)
library(rgdal)
#library(ggmap)
library(rpart)
```

```
# set your working directory here once
dmprojectpath <- "/Users/anupriyathirumurthy/Documents/AnuBackUp/University/MScA_UoC/Courses/DataMining
df_TSM <- read.csv(paste(dmprojectpath, 'Transportation_Sites_Modified.csv', sep = '/'), stringsAsFacto

# Change the Schools_Serviced Column header name in Bus_Breakdown_and_Delays.csv into OPT_Code before r

df_BBD <- read.csv(paste(dmprojectpath, 'Bus_Breakdown_and_Delays_cleaned.csv', sep = '/'), stringsAsFa

df_Routes <- read.csv(paste(dmprojectpath, 'Routes.csv', sep = '/'), stringsAsFactors = F)
table(as.numeric(df_BBD$How_Long_Delayed))
```

```
## Warning in table(as.numeric(df_BBD$How_Long_Delayed)): NAs introduced by
## coercion

##
##     0     1     2     3     4     5     6     7     8     9    10    11
##    96    17    25    98    34  1199    75    86    78    29  8432    38
##    12    13    14    15    16    17    18    19    20    21    22    23
##    73    28    26 29582    24    26     9     4 29473    11    16    19
##    24    25    26    27    28    29    30    31    32    34    35    37
##    24 10800     4     4     6     2 60190     4    13     4  3437     2
##    39    40    41    42    43    45    49    50    51    54    55    56
##     1  3510     1     3     1 33272     1   343     3     1   116    12
##    57    58    59    60    65    70    75    80    90   100   104   105
##     1     1     1 16737     6     3    39    63  3627    12     1     8
##   112   115   117   120   126   130   135   140   145   146   150   152
##     1     2     1   349     1     4     1     1     3     1     1     1
##   154   156   160   180   182   187   190   197   200   202   215   218
##     1     3    18     3     1     1     1     1     2     3     1     1
##   220   225   230   250   260   300   302   306   320   330   401   414
##     3     2     1     3     3     2     1     1     1     1     1     1
##   445   450   454   579   602   612   622   627   634   658   715   800
```

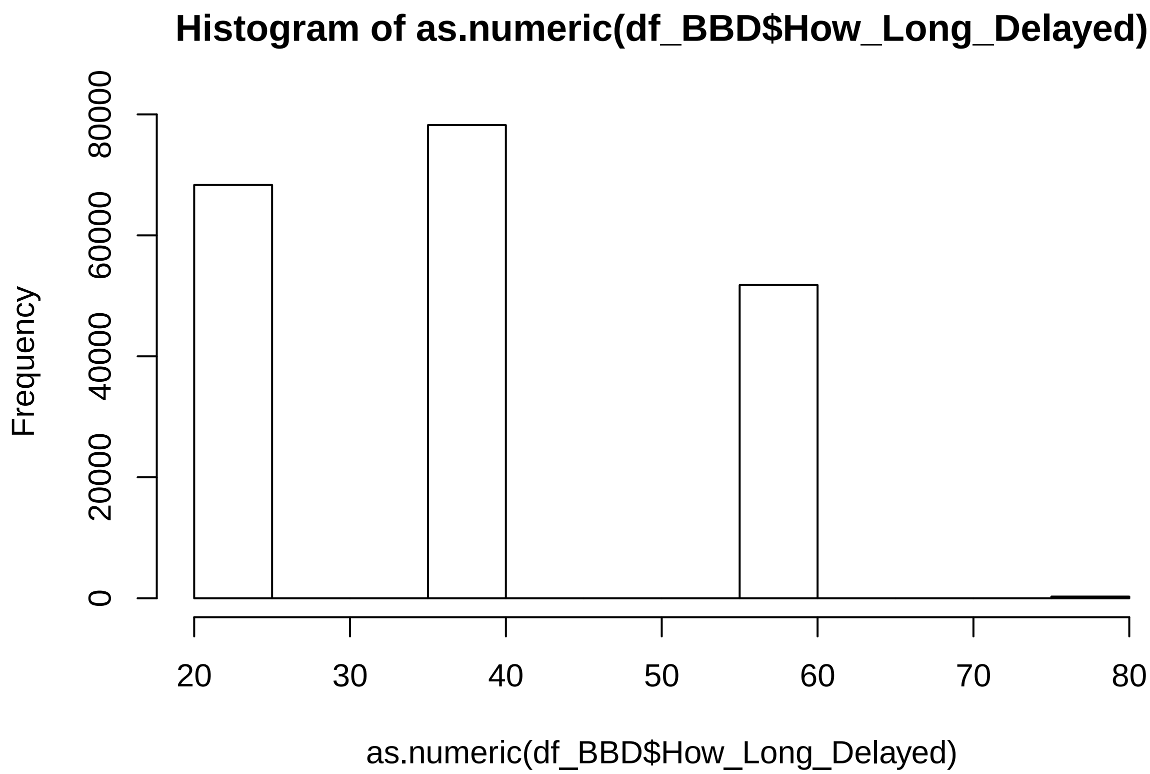
```
##    1    2    3    1    1    1    1    2    1    1    1    1
##  843  882  900  915  932  933 1003 1133 1195 1254 1256 1295
##    1    1    1    1    1    1    1    1    1    1    1    1
## 1300 1304 1310 1311 1337 1344 1346 1349 1354 1366 1369 1375
##    1    1    1    1    1    1    1    1    2    1    1    1
## 1376 1388 1390 1403 1405 1412 1419 1420 1424 1425 1427 1435
##    1    1    1    3    1    2    1    1    1    1    1    2
## 1448 1449 1456 1466 1467 1469 1478 1481 1486 1489 1492 1494
##    1    1    1    1    2    1    1    1    1    2    1    1
## 1495 1512 1603 1641 1657 1665 1924 2085 2134 2228 2254 2305
##    1    1    1    1    1    1    1    1    1    1    1    1
## 2308 2345 2359 2557 2587 2601 2640 2653 2663 2759 3021 3254
##    1    1    1    1    1    1    1    1    1    1    1    3
## 3256 3258 3596 4032 4104 4487 5594 5618 5698 5819 6541 8012
##    1    1    1    1    1    1    1    1    1    1    1    1
## 8521 8916 9025 9540 9541 9568 9584 9651
##    1    1    1    1    1    1    1    1
```

```
for(i in 1:nrow(df_BBD)) {
  if(is.na(df_BBD$How_Long_Delayed[i])) {
    df_BBD$How_Long_Delayed[i] <- "NULL"
  } else if(df_BBD$How_Long_Delayed[i] <= 20) {
      df_BBD$How_Long_Delayed[i] <- 20
  } else if (df_BBD$How_Long_Delayed[i] <= 40) {
      df_BBD$How_Long_Delayed[i] <- 40
  } else if (df_BBD$How_Long_Delayed[i] <= 60) {
      df_BBD$How_Long_Delayed[i] <- 60
  } else if (df_BBD$How_Long_Delayed[i] <= 80) {
    df_BBD$How_Long_Delayed[i] <- 80
  } else if (df_BBD$How_Long_Delayed[i] <= 100) {
    df_BBD$How_Long_Delayed[i]  <- 100
  } else if (df_BBD$How_Long_Delayed[i] <= 120) {
    df_BBD$How_Long_Delayed[i] <- 120
  } else if (df_BBD$How_Long_Delayed[i] <= 140) {
    df_BBD$How_Long_Delayed[i] <- 140
  } else if (df_BBD$How_Long_Delayed[i] <= 160) {
    df_BBD$How_Long_Delayed[i] <- 160
  } else {
    df_BBD$How_Long_Delayed[i] <- "160orMore"
  }
}

unique(df_BBD$How_Long_Delayed)
```

```
## [1] "160orMore" "20"        "40"        "60"        "80"
```

```
hist(as.numeric(df_BBD$How_Long_Delayed))
```

```
## Warning in hist(as.numeric(df_BBD$How_Long_Delayed)): NAs introduced by
## coercion
```




```
as.data.frame(table(df_BBD$How_Long_Delayed))
```

```
##        Var1  Freq
## 1 160orMore 37141
## 2        20 68325
## 3        40 78226
## 4        60 51783
## 5        80   284
```

```
df_BBD$Occurred_On <- as.POSIXct(df_BBD$Occurred_On, format="%m/%d/%Y %H:%M")

df_BBD$Created_On <- as.POSIXct(df_BBD$Created_On, format="%m/%d/%Y %H:%M")

df_BBD$Informed_On <- as.POSIXct(df_BBD$Informed_On, format="%m/%d/%Y %H:%M")

df_BBD$Last_Updated_On <- as.POSIXct(df_BBD$Last_Updated_On, format="%m/%d/%Y %H:%M")
```

```
df_TSMBBD <- merge(x = df_TSM, y=df_BBD)
# 174218 obs. of 37 variables

df_clean <- subset(df_TSMBBD, df_TSMBBD$Run_Type != '' & df_TSMBBD$School_Age_or_PreK == 'School-Age')
# 143917 obs. of 37 variables

df_merge <- as.data.frame(merge(x = df_clean, y = df_Routes))
# 142603 obs. of 49 variables

df_final <- subset(df_merge, df_merge$Boro != '' & df_merge$Boro != 'All Boroughs')
# 135237 obs. of 49 variables
```

LCA

First LCA

City, Run_Type, Vehicle_TypeDescription, Reason

```
fml1 <- data.frame(df_final[, c("Run_Type", "Reason", "City", "Vehicle_TypeDescription")])
unique(fml1$Run_Type)
```

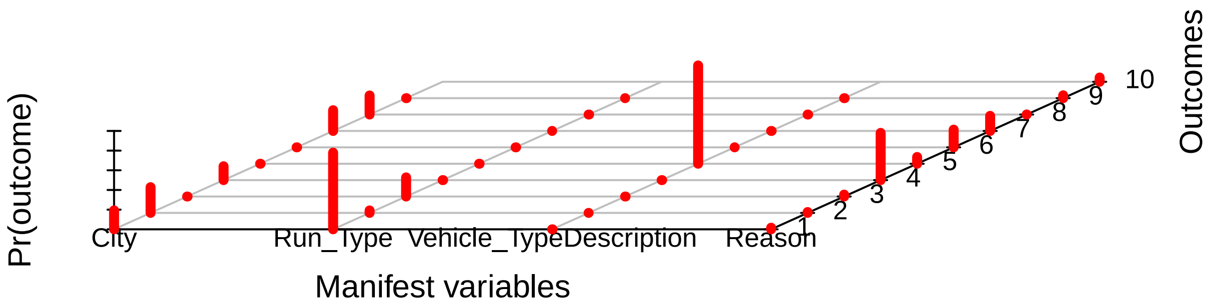
```
## [1] "Special Ed AM Run"       "Special Ed PM Run"
## [3] "Special Ed Field Trip"   "General Ed AM Run"
## [5] "General Ed Field Trip"   "General Ed PM Run"
## [7] "Project Read PM Run"     "Project Read AM Run"
## [9] "Project Read Field Trip"
```

```
df1 <- as.data.frame(apply(fml1, 2, factor))
```

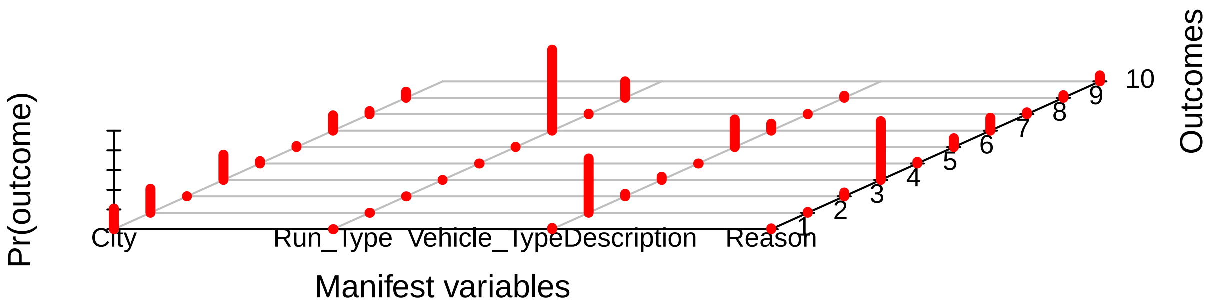
```
nSample <- 135237
train_size <- round(nSample * (70 / 100))

set.seed(7881)
train_data <- sample(1:nrow(df1), train_size)
df.train <- df1[train_data, ]
df.holdout <- df1[-train_data,]

i <-  2:4
lca.results.1 <- lapply(i, function(i) {
  f1 <- cbind(City,
              Run_Type,
              Vehicle_TypeDescription,
              Reason) ~ 1
  results <- list()
  results$noOfClasses <- i
  cat("No.ofclasses", i)
  cat("\n\n\n")
  LCA1 <- poLCA(f1, df.train, nclass = i, nrep = 10, tol = .001, verbose = FALSE, graphs = TRUE)
  results$aic <- LCA1$aic
  results$bic <- LCA1$bic
  return(list(results = results, LCA = LCA1))
})
```

```
## No.ofclasses 2
```

Class 1: population share = 0.189

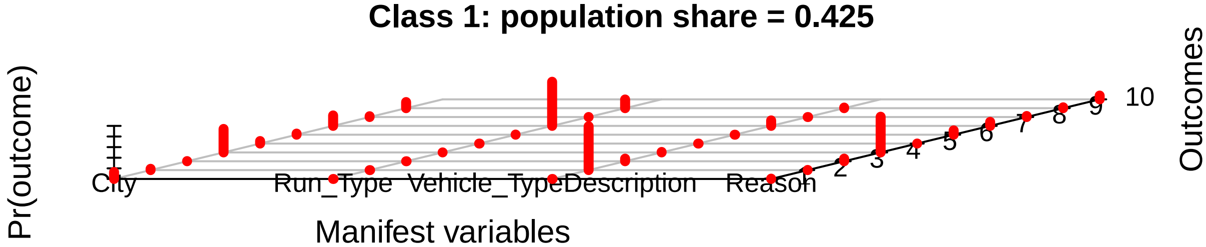


Class 2: population share = 0.811

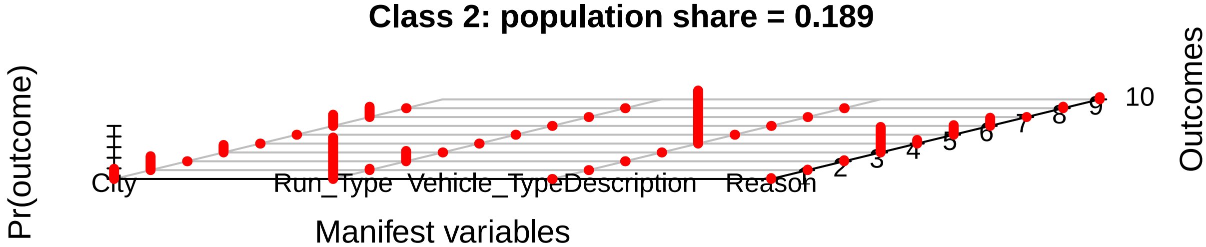



```
## No.ofclasses 3
```

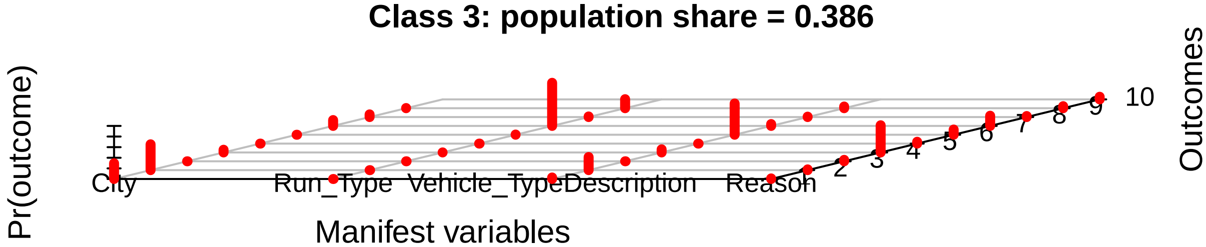
Class 1: population share = 0.425



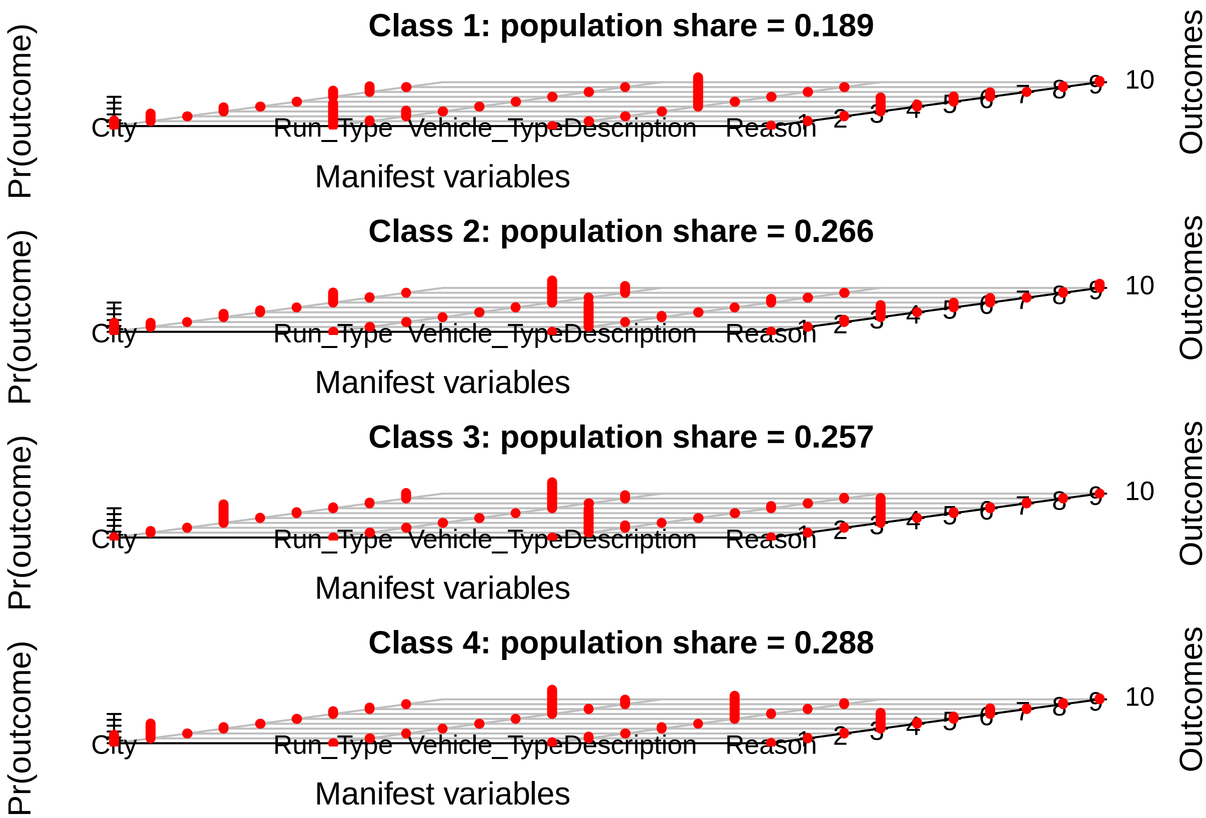
Class 2: population share = 0.189



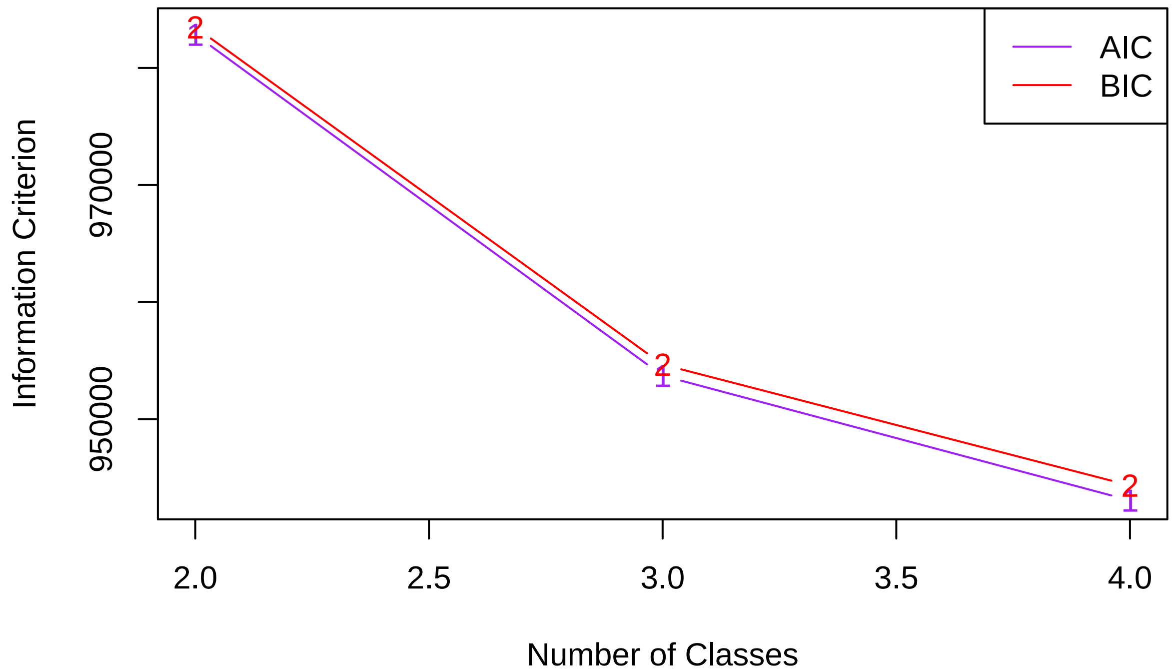
Class 3: population share = 0.386




```
## No.ofclasses 4
```




```r
lca.results.report.1 <- data.frame( do.call(rbind, lapply(lca.results.1, function(x){x$res})) )

matplot(lca.results.report.1$noOfClasses,
        cbind(lca.results.report.1$aic, lca.results.report.1$bic),
        type = "b",
        col = c("purple","red"),
        lty = c(1, 1),
        xlab = 'Number of Classes',
        ylab = 'Information Criterion')
legend(x = 'topright', legend = c('AIC','BIC'), lty = c(1, 1), col = c('purple','red'))
```



Second LCA

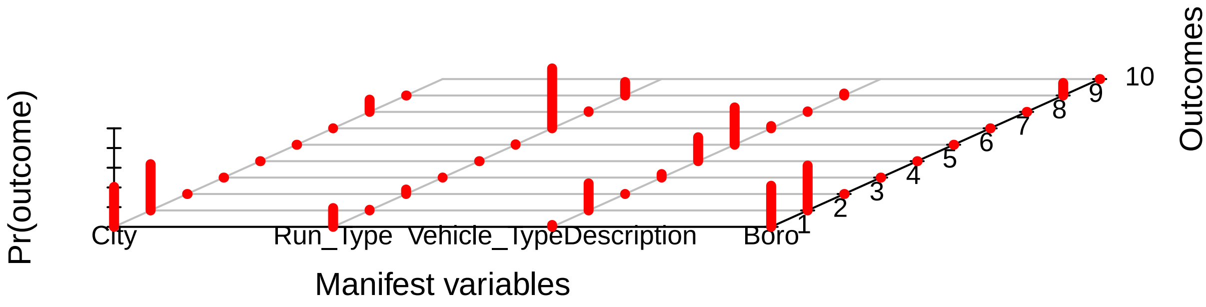
City, Run_Type, Vehicle_TypeDescription, Boro

```
fml2 <- data.frame(df_final[, c("Run_Type", "Boro", "City", "Vehicle_TypeDescription")])

df2 <- as.data.frame(apply(fml2, 2, factor))
```

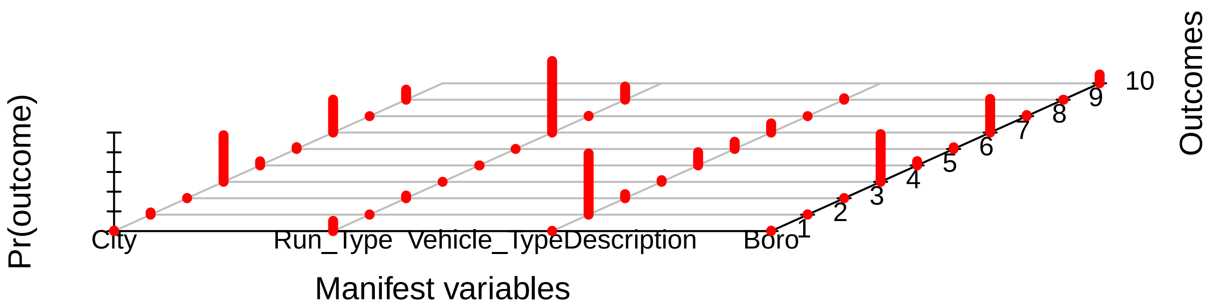
```
nSample <- 135237
train_size <- nSample * (70 / 100)

set.seed(7881)
train_data <- sample(1:nrow(df2), train_size)
df.train <- df2[train_data, ]
df.holdout <- df2[-train_data,]

i <-  2:4
lca.results.2 <- lapply( i, function(i) {
  f2 <- cbind(City,
              Run_Type,
              Vehicle_TypeDescription,
              Boro) ~ 1
  results <- list()
  results$noOfClasses <- i
  cat("No.ofclasses", i)
  cat("\n\n\n")
  LCA2 <- poLCA(f2, df.train, nclass = i, nrep = 10, tol = .001, verbose = FALSE, graphs = TRUE)
  results$aic <- LCA2$aic
  results$bic <- LCA2$bic
  results$probs <- LCA2$probs
  return(list(results = results, LCA = LCA2))
})
```

```
## No.ofclasses 2
```

Class 1: population share = 0.504

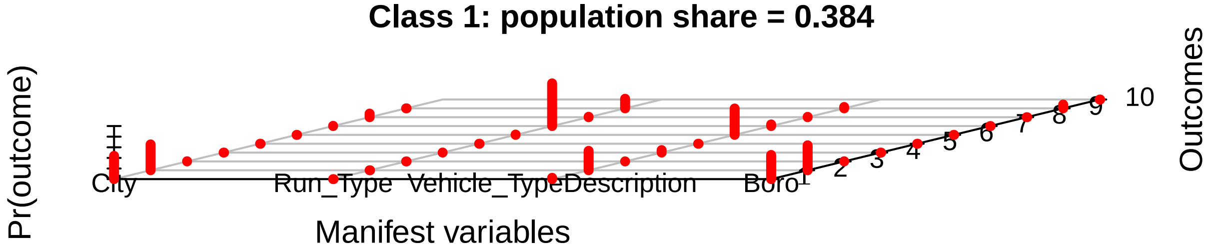


Class 2: population share = 0.496

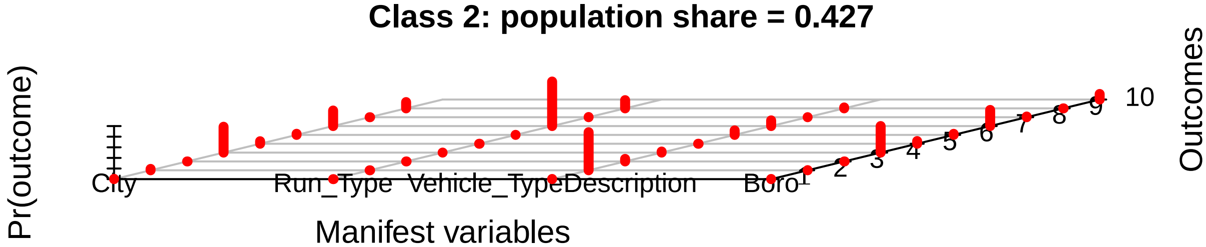



```
## No.ofclasses 3
```

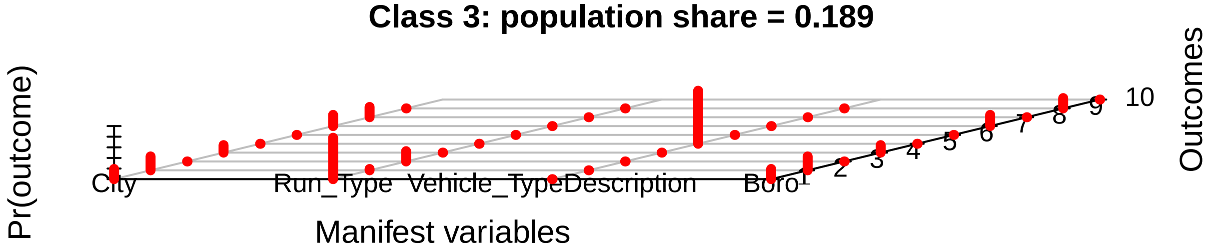
Class 1: population share = 0.384



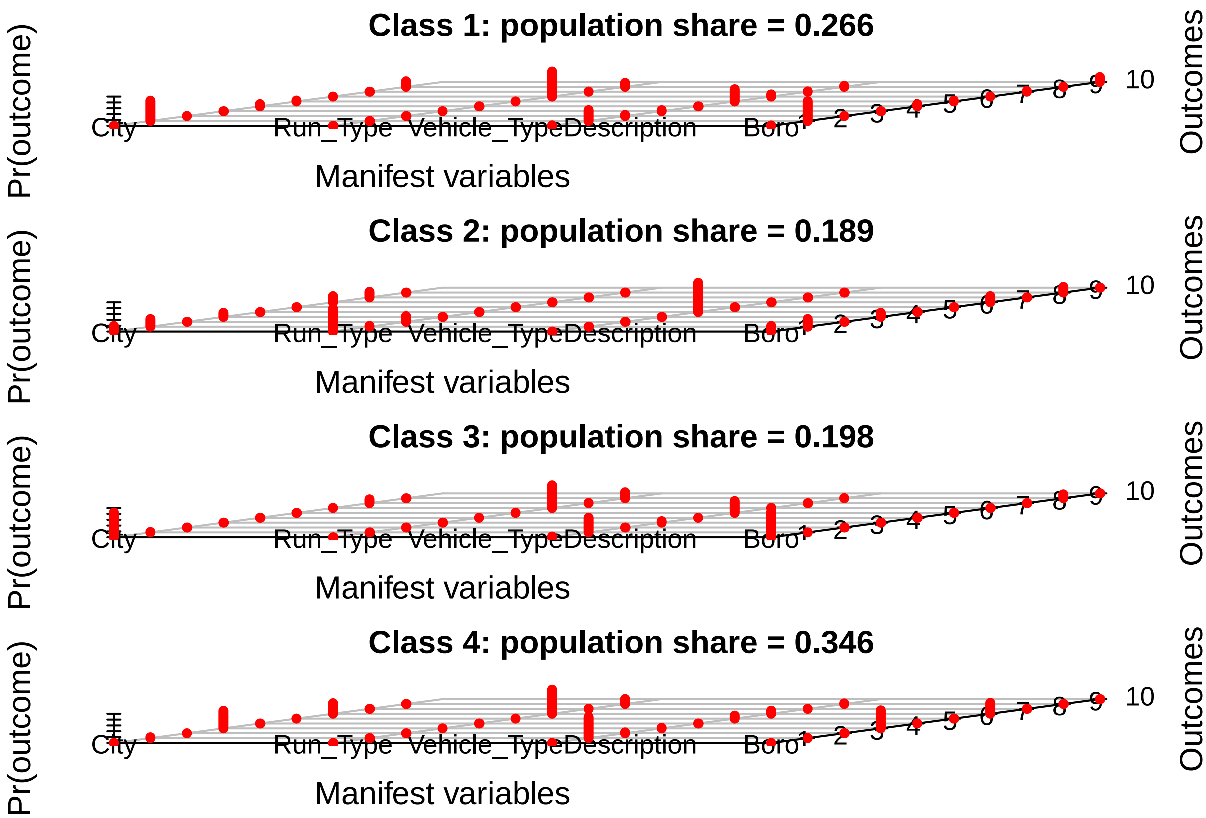
Class 2: population share = 0.427



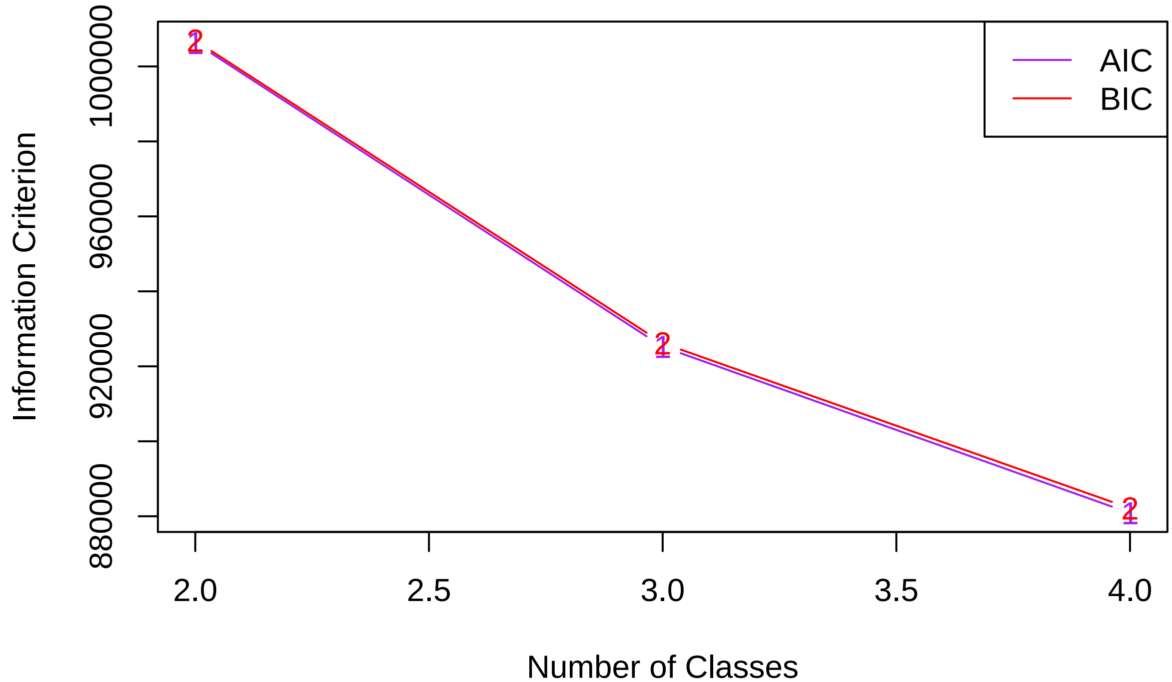
Class 3: population share = 0.189




```
## No.ofclasses 4
```

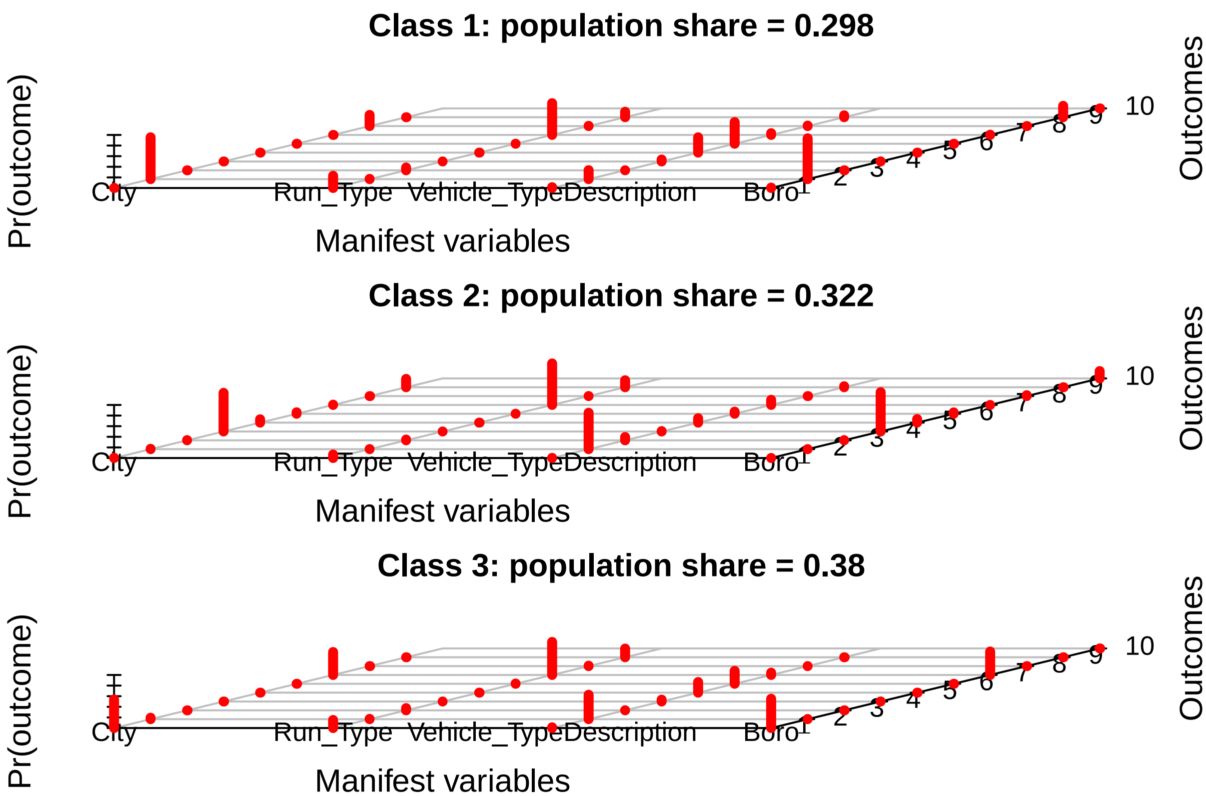



```
lca.results.report.2 <- data.frame( do.call(rbind, lapply(lca.results.2, function(x){x$res})) )

matplot(lca.results.report.2$noOfClasses,
        cbind(lca.results.report.2$aic, lca.results.report.2$bic),
        type = "b",
        col = c("purple","red"),
        lty = c(1, 1),
        xlab = 'Number of Classes',
        ylab = 'Information Criterion')
legend(x = 'topright', legend = c('AIC','BIC'), lty = c(1, 1), col = c('purple','red'))
```



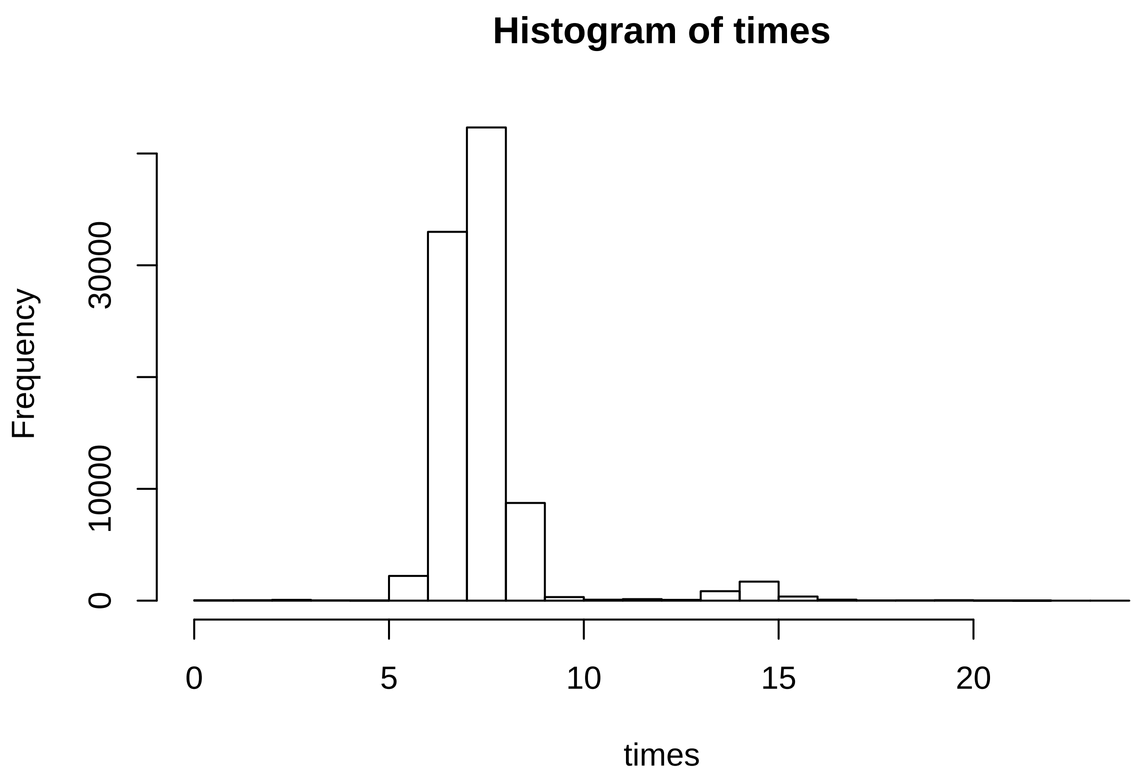
Holdout validation of chosen solution (3-Class)

```
f2 <- cbind(City,
              Run_Type,
              Vehicle_TypeDescription,
              Boro) ~ 1
LCA2.3.holdout <- poLCA(f2, df.train, nclass = 3, nrep = 10, tol = .001, verbose = FALSE, graphs = TRUE
                        probs.start=lca.results.report.2$probs[2])
```

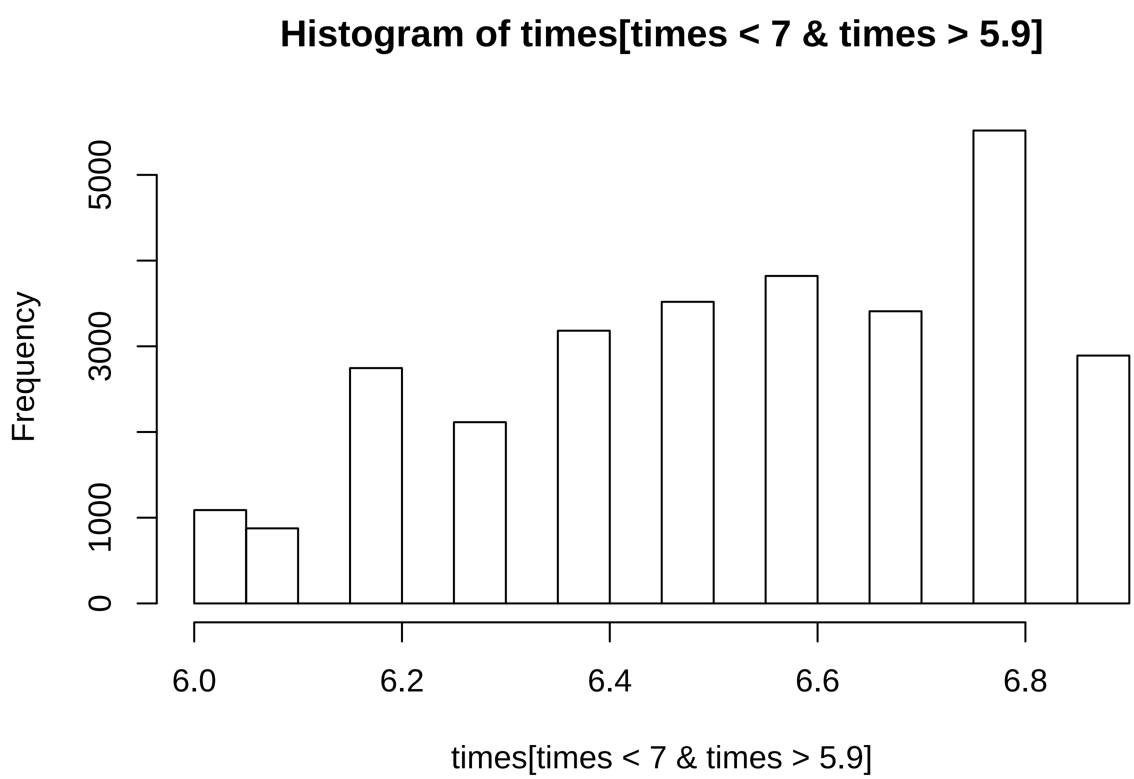


When do delays occur for Special Ed AM Runs?

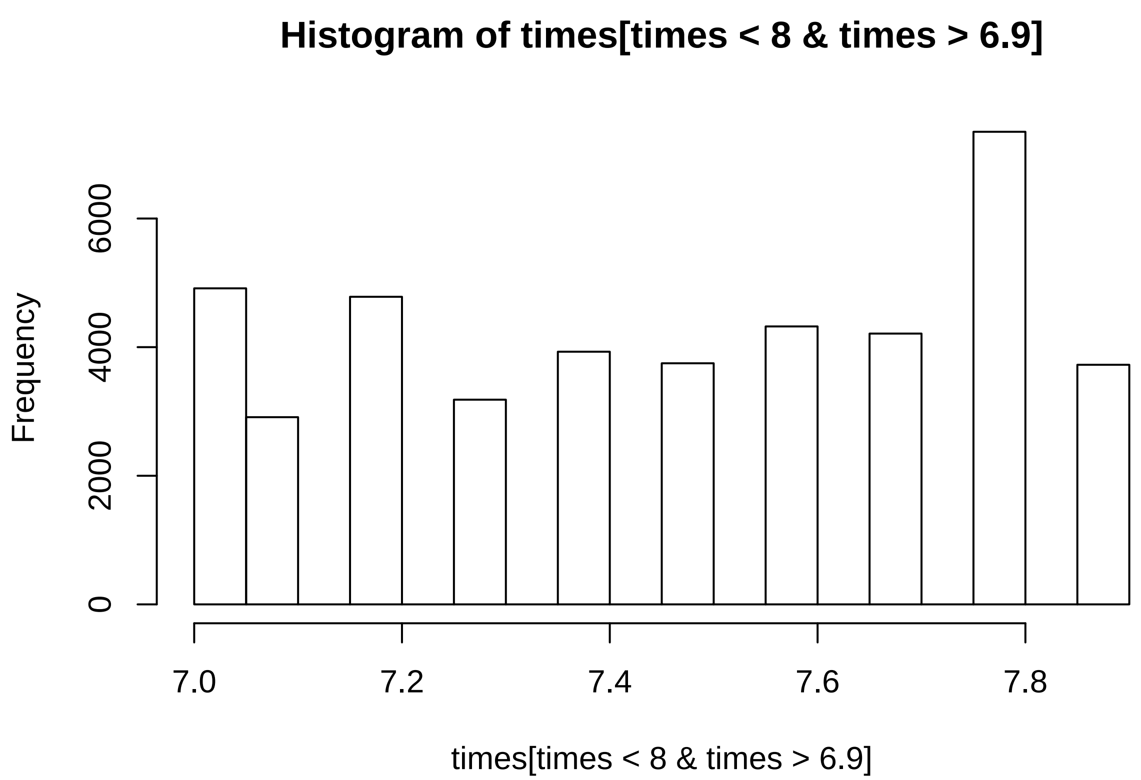
```
times <- apply((subset(cbind(substr(df_final$Occurred_On, 12, 13),substr(df_final$Occurred_On, 15, 16))
                 df_final$Run_Type=='Special Ed AM Run')), 2, as.numeric)
times <- times[,1] + round((times[,2]/60),1)
hist(times)
```




```r
hist(times[times<7 & times>5.9])
```




```r
hist(times[times<8 & times>6.9])
```



How many delays/breakdowns correspond to routes with multiple schools listed?

```
length(which(grepl("@",unique(df_final$Name))))/length(unique(df_final$Name))
```

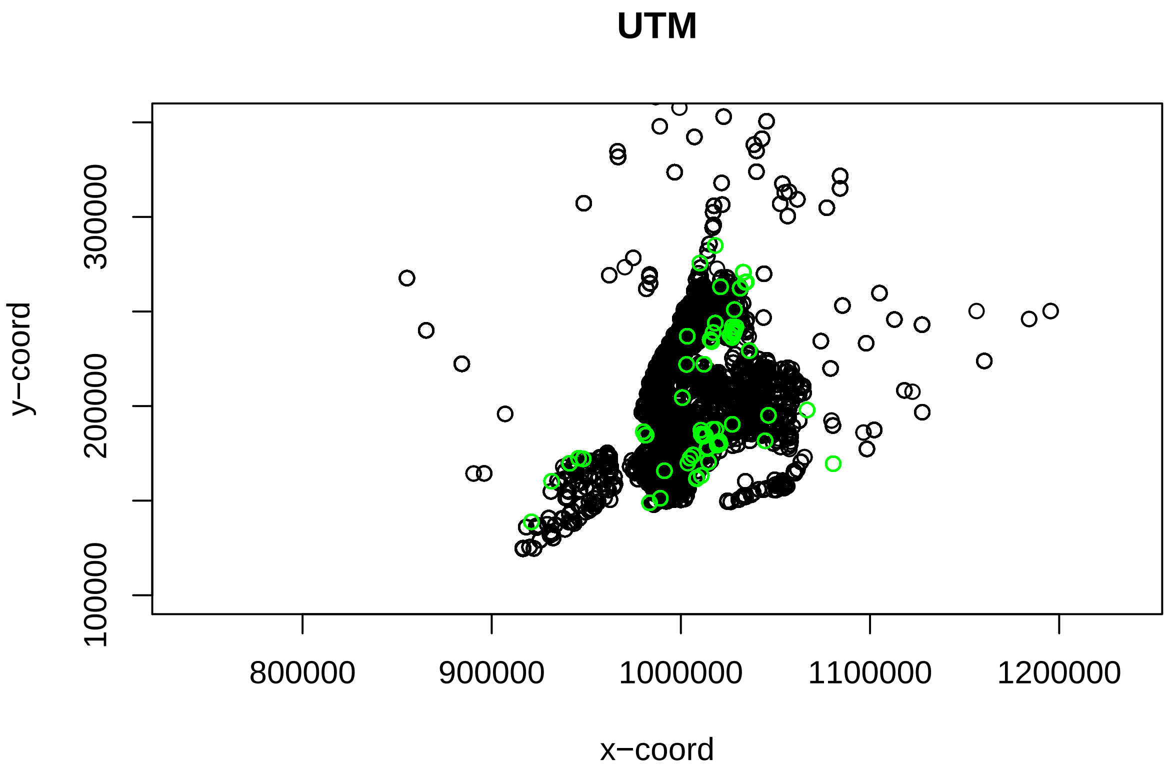
```
## [1] 0.1665914
```

———————————————————————————**-CART**——————————————————————————————

———

Convert lat and long of school to UTM before building tree model

```
Trans_Sites <- data.frame(Latitude=df_final$Latitude, Longitude=df_final$Longitude, School.Year=df_final
#map <- get_map('New York City', maptype='terrain', source='google')
#ggmap(map, base_layer = ggplot(aes(x=Longitude, y=Latitude), data=Trans_Sites)) + geom_point(color="re
cord.dec <- SpatialPoints(cbind(Trans_Sites$Longitude, Trans_Sites$Latitude), proj4string=CRS("+proj=lon
cord.UTM <- spTransform(cord.dec, CRS("+init=epsg:2263"))
plot(cord.UTM, axes=TRUE, xlab="x-coord", ylab="y-coord", main="UTM",pch=1,xlim=c(975000,1000000),ylim=c
points(df_final$XCoordinates, df_final$YCoordinates, col='green')
```



Generate Tree 1

```r
unique(df_final$City)
```

```
## [1] "Connecticut"   "New Jersey"    "Brooklyn"      "Nassau"
## [5] "Manhattan"     "Queens"        "Bronx"         "Staten Island"
## [9] "Westchester"
```

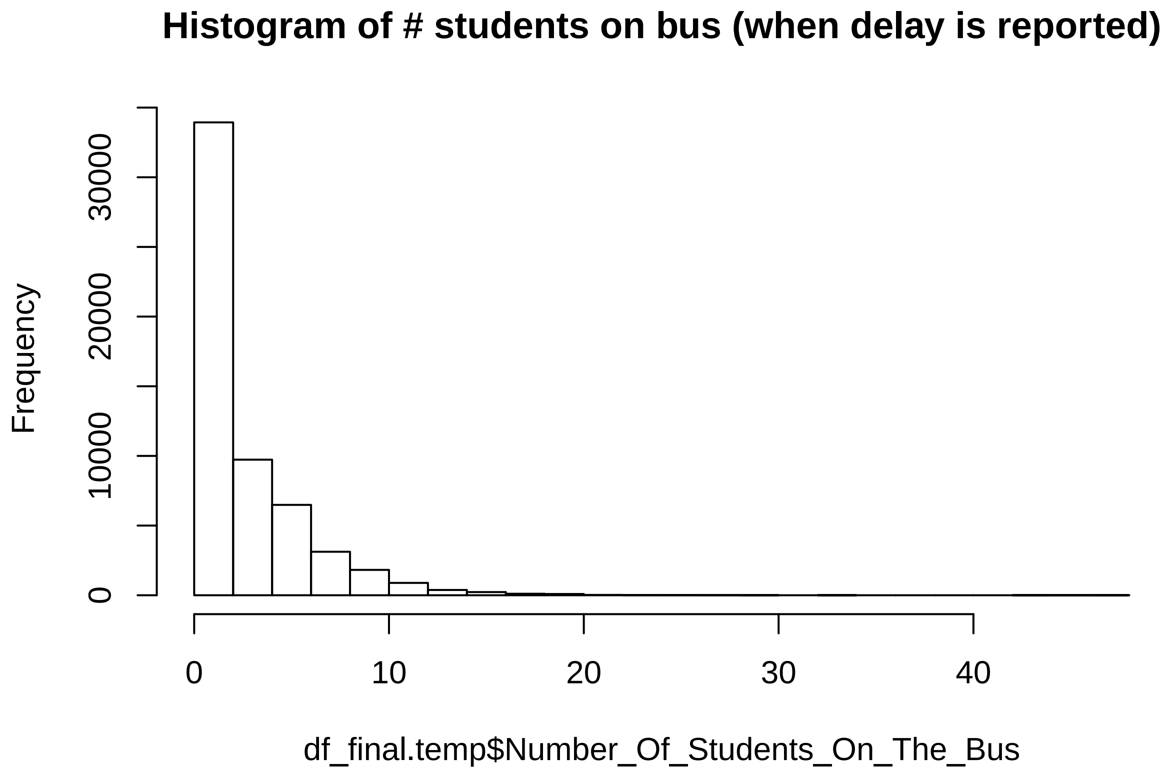
```r
unique(df_final$Garage_City)
```

```
##  [1] "Pelham Manor"  "Brooklyn"      "BROOKLYN"      "BRONX"
##  [5] "bronx"         "SI"            "QUEENS"        "BKLYN"
##  [9] "brooklyn"      "Jamaica"       "BROOKLYN NY"   "Astoria"
## [13] "Ozone Park"    "Bronx"         "Staten Island" "Elmont"
## [17] "Whitestone"    "Oceanside"     "B'klyn"        "Yonkers"
## [21] "Mount Vernon"  ""              "BRON"          "Pelham"
```

```r
df_final.temp <- data.frame(df_final, data.frame(cord.UTM))

df_final.temp$Number_Of_Students_On_The_Bus <- as.numeric(df_final.temp$Number_Of_Students_On_The_Bus)
```

```
## Warning: NAs introduced by coercion
```

```r
df_final.temp <- subset(df_final.temp, df_final.temp$Number_Of_Students_On_The_Bus!='NA' &
                        df_final.temp$Number_Of_Students_On_The_Bus<=50 & df_final.temp$Run_Type=='Spec
                        & df_final.temp$Garage_City!='' & df_final.temp$Reason=='Heavy Traffic')
hist(df_final.temp$Number_Of_Students_On_The_Bus,main='Histogram of # students on bus (when delay is re
```




```r
df_final.temp$Number_Of_Students_On_The_Bus <- (df_final.temp$Number_Of_Students_On_The_Bus==0)*1

df_final.temp$Garage_City[which(df_final.temp$Garage_City=='BROOKLYN')] <- 'Brooklyn'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=='brooklyn')] <- 'Brooklyn'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=='BKLYN')] <- 'Brooklyn'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=='BROOKLYN NY')] <- 'Brooklyn'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="B'klyn")] <- 'Brooklyn'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="BRONX")] <- 'Bronx'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="bronx")] <- 'Bronx'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="BRON")] <- 'Bronx'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="SI")] <- 'Staten Island'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Pelham Manor")] <- 'Westchester'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Pelham")] <- 'Westchester'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="QUEENS")] <- 'Queens'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Jamaica")] <- 'Queens'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Yonkers")] <- 'Westchester'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Astoria")] <- 'Queens'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Whitestone")] <- 'Queens'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Elmont")] <- 'Nassau'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Ozone Park")] <- 'Queens'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Mount Vernon")] <- 'Westchester'
df_final.temp$Garage_City[which(df_final.temp$Garage_City=="Oceanside")] <- 'Nassau'

unique(df_final.temp$City)
```

```
## [1] "Connecticut"   "New Jersey"    "Brooklyn"      "Nassau"
## [5] "Manhattan"     "Queens"        "Staten Island" "Westchester"
## [9] "Bronx"
```

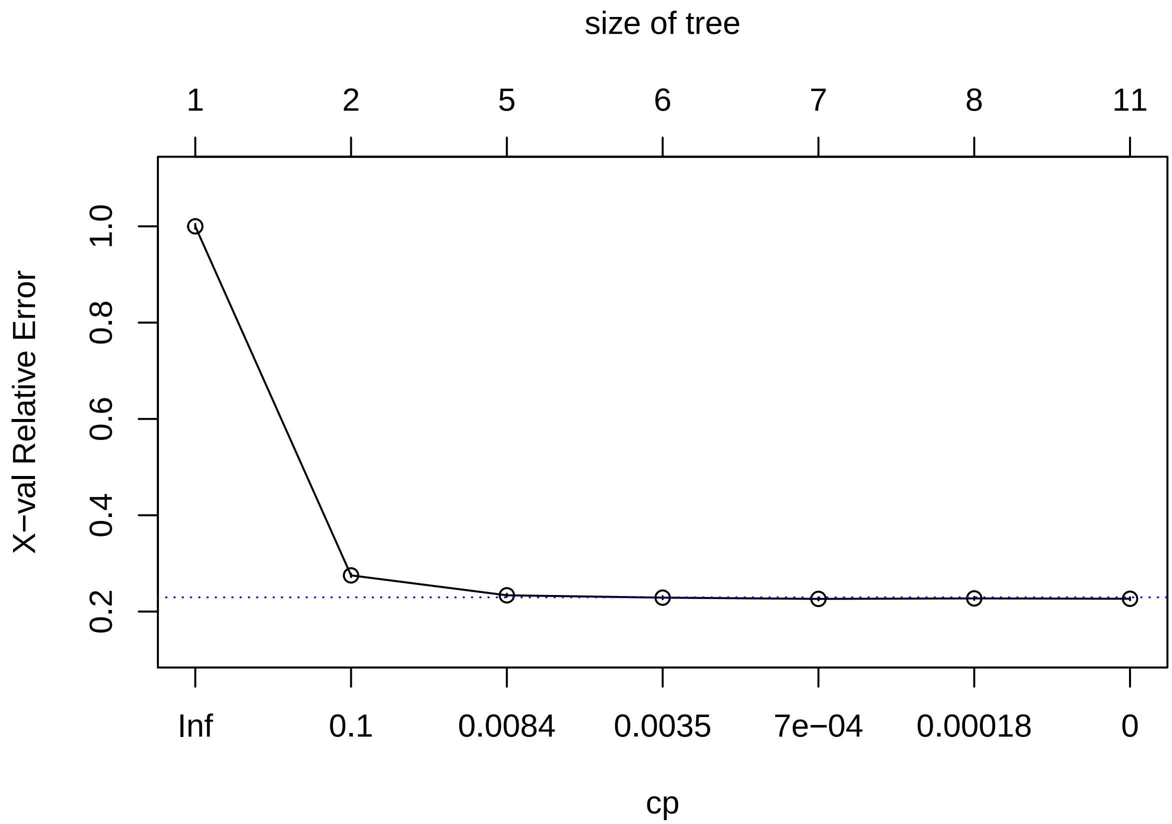
```r
unique(df_final.temp$Garage_City)
```

```
## [1] "Westchester"   "Brooklyn"      "Bronx"         "Staten Island"
## [5] "Queens"        "Nassau"
```

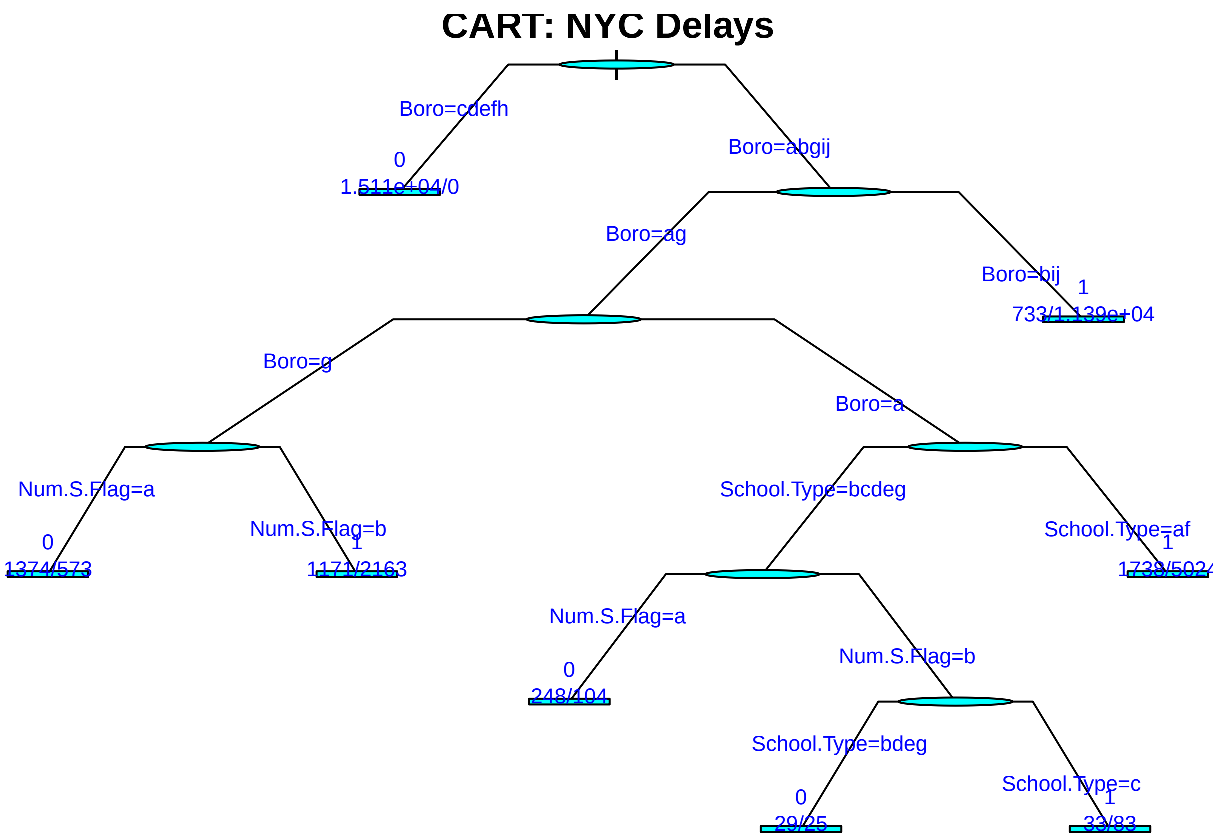
```r
City.Flag <- (df_final.temp$Garage_City==df_final.temp$Boro)*1

tree.df <- data.frame(City.Flag=factor(City.Flag), Num.S.Flag=factor(df_final.temp$Number_Of_Students_Or
                      Boro=factor(df_final.temp$Boro), School.Type=factor(df_final.temp$Affiliation))

nSample <- 56855
train_size_tree <- round(nSample * (70 / 100))

set.seed(7881)
train_data_tree <- sample(1:nrow(tree.df), train_size_tree)

df.train.tree <- tree.df[train_data_tree, ]
df.holdout.tree <- tree.df[-train_data_tree, ]

tree.train <- rpart(df.train.tree,control=rpart.control(cp=0,minsplit=30,xval=10, maxsurrogate=0))
printcp(tree.train)
```

```
##
## Classification tree:
## rpart(formula = df.train.tree, control = rpart.control(cp = 0,
##     minsplit = 30, xval = 10, maxsurrogate = 0))
##
## Variables actually used in tree construction:
## [1] Boro        Num.S.Flag  School.Type
##
## Root node error: 19359/39798 = 0.48643
##
## n= 39798
##
##           CP nsplit rel error  xerror      xstd
## 1 0.72488248      0   1.00000 1.00000 0.0051506
## 2 0.01379203      1   0.27512 0.27512 0.0035085
## 3 0.00506224      4   0.23374 0.23374 0.0032713
## 4 0.00237616      5   0.22868 0.22889 0.0032414
## 5 0.00020662      6   0.22630 0.22630 0.0032254
## 6 0.00015497      7   0.22610 0.22734 0.0032318
## 7 0.00000000     10   0.22563 0.22656 0.0032270
```

```r
plotcp(tree.train,minline=TRUE,col=4)
```




```
cptable <- data.frame(tree.train$cptable)
cp <- cptable[which.min(cptable$xerror),1]

tree.train.pruned <- rpart(df.train.tree,control=rpart.control(cp=cp, minsplit=30, xval=10, maxsurrogat

par(mai=c(0.1,0.1,0.1,0.1))
plot(tree.train.pruned,main="CART: NYC Delays",col=3, compress=TRUE,
     branch=0.5,uniform=TRUE)
text(tree.train.pruned,cex=0.7,col=4,use.n=TRUE,fancy=TRUE,fwidth=0.2,fheight=0.05,bg=c(5))
```




```
table(df.train.tree[,1], predict(tree.train.pruned,type="class"))
```

```
##
##         0     1
##   0 16764  3675
##   1   702 18657
```

```
round(prop.table(table(df.train.tree[,1], predict(tree.train.pruned,type="class")),1),2)
```

```
##
##        0    1
##   0 0.82 0.18
##   1 0.04 0.96
```

Generate Tree 2

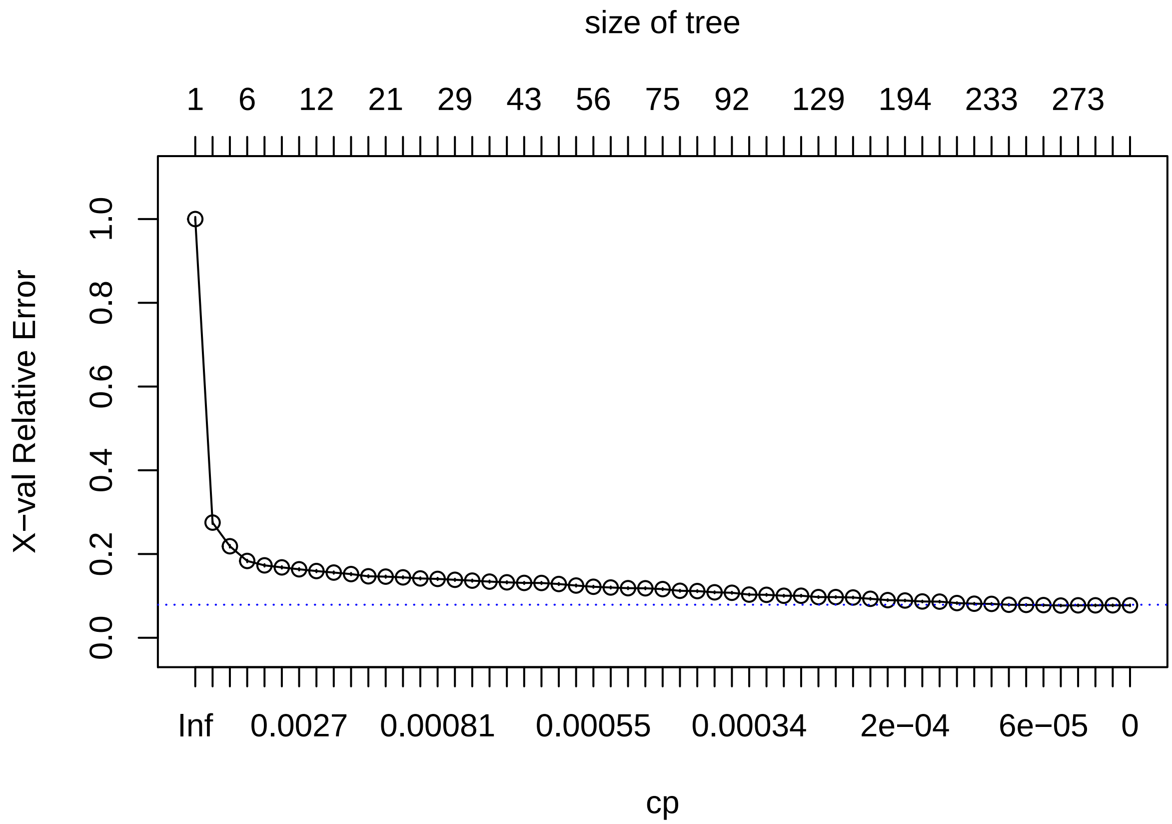
```
dist <- sqrt((df_final.temp$XCoordinates-df_final.temp$coords.x1)^2 + (df_final.temp$YCoordinates-df_fi

times <- apply((cbind(substr(df_final.temp$Occurred_On, 12, 13),substr(df_final.temp$Occurred_On, 15, 1
times <- times[,1] + round((times[,2]/60),1)

tree.df <- data.frame(City.Flag=factor(City.Flag),
                      Num.S.Flag=factor(df_final.temp$Number_Of_Students_On_The_Bus),
                      School.Type=factor(df_final.temp$Affiliation), dist=dist, Boro=factor(df_final.te

nSample <- 56855
train_size_tree <- round(nSample * (70 / 100))

set.seed(7881)
```

```r
train_data_tree <- sample(1:nrow(tree.df), train_size_tree)

df.train.tree <- tree.df[train_data_tree, ]
df.holdout.tree <- tree.df[-train_data_tree, ]

tree.train <- rpart(df.train.tree,control=rpart.control(cp=0,minsplit=30,xval=10, maxsurrogate=0))
printcp(tree.train)
```

```
##
## Classification tree:
## rpart(formula = df.train.tree, control = rpart.control(cp = 0,
##     minsplit = 30, xval = 10, maxsurrogate = 0))
##
## Variables actually used in tree construction:
## [1] Boro        dist        Num.S.Flag  School.Type
##
## Root node error: 19359/39798 = 0.48643
##
## n= 39798
##
##            CP nsplit rel error   xerror      xstd
## 1  7.2488e-01      0  1.000000 1.000000 0.0051506
## 2  2.8281e-02      1  0.275118 0.275118 0.0035085
## 3  1.7537e-02      3  0.218555 0.218555 0.0031764
## 4  5.2689e-03      5  0.183481 0.183481 0.0029380
## 5  3.7967e-03      7  0.172943 0.172943 0.0028604
## 6  2.9444e-03      9  0.165349 0.168087 0.0028236
## 7  2.4795e-03     10  0.162405 0.163593 0.0027889
## 8  1.7821e-03     11  0.159926 0.159357 0.0027556
## 9  1.6013e-03     13  0.156361 0.155742 0.0027268
## 10 1.4722e-03     15  0.153159 0.152074 0.0026971
## 11 1.2656e-03     17  0.150214 0.146805 0.0026536
## 12 1.1881e-03     20  0.145307 0.146030 0.0026472
## 13 9.0397e-04     23  0.141743 0.144274 0.0026324
## 14 8.5232e-04     25  0.139935 0.141691 0.0026105
## 15 7.7483e-04     27  0.138230 0.140555 0.0026008
## 16 7.5761e-04     28  0.137455 0.138437 0.0025825
## 17 7.1285e-04     32  0.134304 0.136474 0.0025655
## 18 6.9735e-04     37  0.130740 0.134149 0.0025451
## 19 6.5430e-04     39  0.129346 0.132393 0.0025295
## 20 6.4569e-04     42  0.127383 0.130999 0.0025171
## 21 6.3709e-04     44  0.126091 0.130844 0.0025157
## 22 6.1987e-04     53  0.119324 0.128364 0.0024933
## 23 5.6821e-04     54  0.118704 0.124748 0.0024603
## 24 5.3377e-04     55  0.118136 0.121804 0.0024329
## 25 5.1656e-04     58  0.116535 0.120048 0.0024164
## 26 4.9934e-04     62  0.114469 0.118446 0.0024012
## 27 4.9589e-04     66  0.112299 0.118188 0.0023988
## 28 4.3046e-04     74  0.108270 0.116225 0.0023800
## 29 4.2616e-04     77  0.106979 0.112196 0.0023408
## 30 4.1324e-04     81  0.105274 0.111473 0.0023337
## 31 3.8742e-04     89  0.101710 0.108580 0.0023049
## 32 3.6159e-04     91  0.100935 0.107495 0.0022940
## 33 3.2715e-04     94  0.099850 0.103156 0.0022497
```

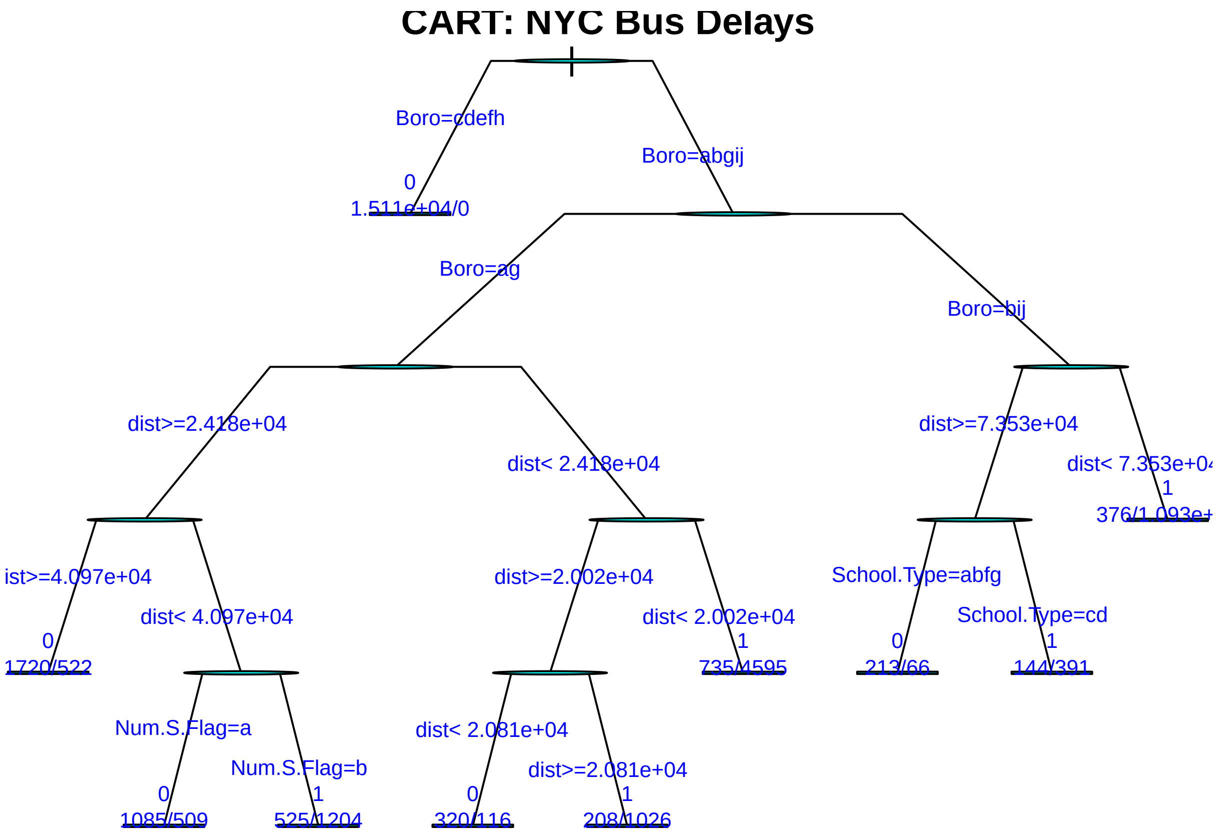
```
## 34 3.0993e-04    102  0.097061 0.102588 0.0022438
## 35 2.9271e-04    117  0.091895 0.100367 0.0022207
## 36 2.8411e-04    120  0.091017 0.100418 0.0022212
## 37 2.7550e-04    128  0.088744 0.097164 0.0021867
## 38 2.6473e-04    131  0.087918 0.097164 0.0021867
## 39 2.5828e-04    143  0.084560 0.096183 0.0021762
## 40 2.3245e-04    150  0.082649 0.093083 0.0021426
## 41 2.0662e-04    182  0.072731 0.089881 0.0021071
## 42 1.9801e-04    193  0.070458 0.089003 0.0020972
## 43 1.8079e-04    200  0.069012 0.086678 0.0020709
## 44 1.7219e-04    207  0.067669 0.086316 0.0020668
## 45 1.5497e-04    220  0.065034 0.082856 0.0020267
## 46 1.2914e-04    230  0.063226 0.081358 0.0020090
## 47 1.0331e-04    232  0.062968 0.080893 0.0020035
## 48 8.6093e-05    251  0.061005 0.078878 0.0019794
## 49 6.8874e-05    254  0.060747 0.078465 0.0019744
## 50 5.1656e-05    257  0.060540 0.077948 0.0019682
## 51 3.0993e-05    267  0.060024 0.076967 0.0019562
## 52 2.5828e-05    272  0.059869 0.077483 0.0019625
## 53 2.0662e-05    274  0.059817 0.077483 0.0019625
## 54 1.0331e-05    279  0.059714 0.077535 0.0019632
## 55 0.0000e+00    284  0.059662 0.077587 0.0019638
```

```r
plotcp(tree.train,minline=TRUE,col=4)
```




```r
cptable <- data.frame(tree.train$cptable)
cp <- cptable[which.min(cptable$xerror),1]

tree.train.pruned <- rpart(df.train.tree,control=rpart.control(cp=.003, minsplit=30, xval=10, maxsurroga
```

```r
par(mai=c(0.1,0.1,0.1,0.1))
plot(tree.train.pruned,main="CART: NYC Bus Delays",col=3, compress=TRUE,
     branch=0.5,uniform=TRUE)
text(tree.train.pruned,cex=0.7,col=4,use.n=TRUE,fancy=TRUE,fwidth=0.2,fheight=0.02,bg=c(5))
```




```r
table(df.train.tree[,1], predict(tree.train.pruned,type="class"))
```

```
##
##          0     1
##   0 18451  1988
##   1  1213 18146
```

```r
round(prop.table(table(df.train.tree[,1], predict(tree.train.pruned,type="class")),1),2)
```

```
##
##        0    1
##   0 0.90 0.10
##   1 0.06 0.94
```

Generate Tree 3 - Selected Tree Solution

```r
tree.df <- data.frame(Boro=factor(df_final.temp$Boro), City.Flag=factor(City.Flag),
                      Num.S.Flag=factor(df_final.temp$Number_Of_Students_On_The_Bus),
                      School.Type=factor(df_final.temp$Affiliation), dist=dist)

nSample <- 56855
train_size_tree <- round(nSample * (70 / 100))

set.seed(7881)
```

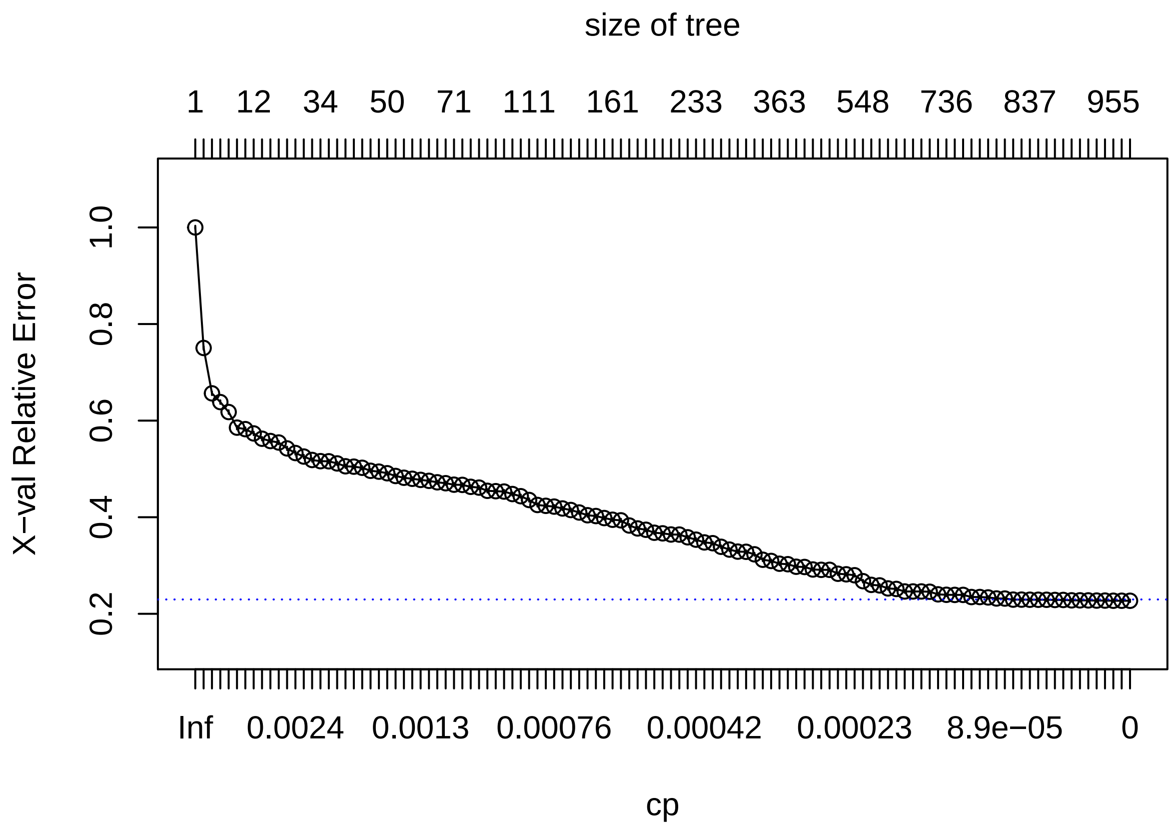
```r
train_data_tree <- sample(1:nrow(tree.df), train_size_tree)

df.train.tree <- tree.df[train_data_tree, ]
df.holdout.tree <- tree.df[-train_data_tree, ]

tree.train <- rpart(df.train.tree,control=rpart.control(cp=0,minsplit=30,xval=10, maxsurrogate=0))
printcp(tree.train)
```

```
##
## Classification tree:
## rpart(formula = df.train.tree, control = rpart.control(cp = 0,
##     minsplit = 30, xval = 10, maxsurrogate = 0))
##
## Variables actually used in tree construction:
## [1] City.Flag   dist        Num.S.Flag  School.Type
##
## Root node error: 26596/39798 = 0.66827
##
## n= 39798
##
##              CP nsplit rel error  xerror      xstd
## 1   2.4951e-01      0   1.00000 1.00000 0.0035317
## 2   9.3999e-02      1   0.75049 0.75049 0.0037504
## 3   3.0117e-02      2   0.65649 0.65649 0.0037222
## 4   9.0803e-03      3   0.62637 0.63840 0.0037099
## 5   5.1324e-03      5   0.60821 0.61780 0.0036931
## 6   4.4368e-03      8   0.59020 0.58550 0.0036607
## 7   4.3240e-03      9   0.58576 0.58238 0.0036572
## 8   3.5814e-03     11   0.57712 0.57351 0.0036468
## 9   3.3464e-03     15   0.56279 0.56253 0.0036331
## 10  3.0832e-03     16   0.55945 0.55775 0.0036269
## 11  3.0205e-03     20   0.54711 0.55467 0.0036228
## 12  2.4064e-03     23   0.53805 0.54245 0.0036059
## 13  2.3688e-03     26   0.53083 0.53290 0.0035918
## 14  2.2936e-03     31   0.51899 0.52561 0.0035806
## 15  2.0304e-03     32   0.51669 0.51850 0.0035693
## 16  1.9928e-03     33   0.51466 0.51609 0.0035654
## 17  1.9552e-03     34   0.51267 0.51579 0.0035649
## 18  1.8424e-03     35   0.51072 0.51139 0.0035577
## 19  1.7484e-03     39   0.50293 0.50545 0.0035476
## 20  1.7296e-03     43   0.49594 0.50466 0.0035462
## 21  1.6168e-03     45   0.49248 0.50233 0.0035422
## 22  1.5416e-03     46   0.49086 0.49613 0.0035312
## 23  1.5040e-03     47   0.48932 0.49436 0.0035280
## 24  1.4664e-03     49   0.48631 0.49105 0.0035220
## 25  1.3912e-03     55   0.47721 0.48537 0.0035115
## 26  1.3724e-03     57   0.47443 0.48173 0.0035045
## 27  1.3348e-03     59   0.47169 0.47955 0.0035004
## 28  1.3160e-03     63   0.46578 0.47733 0.0034961
## 29  1.1280e-03     64   0.46447 0.47533 0.0034922
## 30  1.0904e-03     65   0.46334 0.47225 0.0034861
## 31  1.0528e-03     69   0.45898 0.47037 0.0034823
## 32  1.0277e-03     70   0.45793 0.46736 0.0034762
## 33  1.0152e-03     73   0.45484 0.46691 0.0034753
```

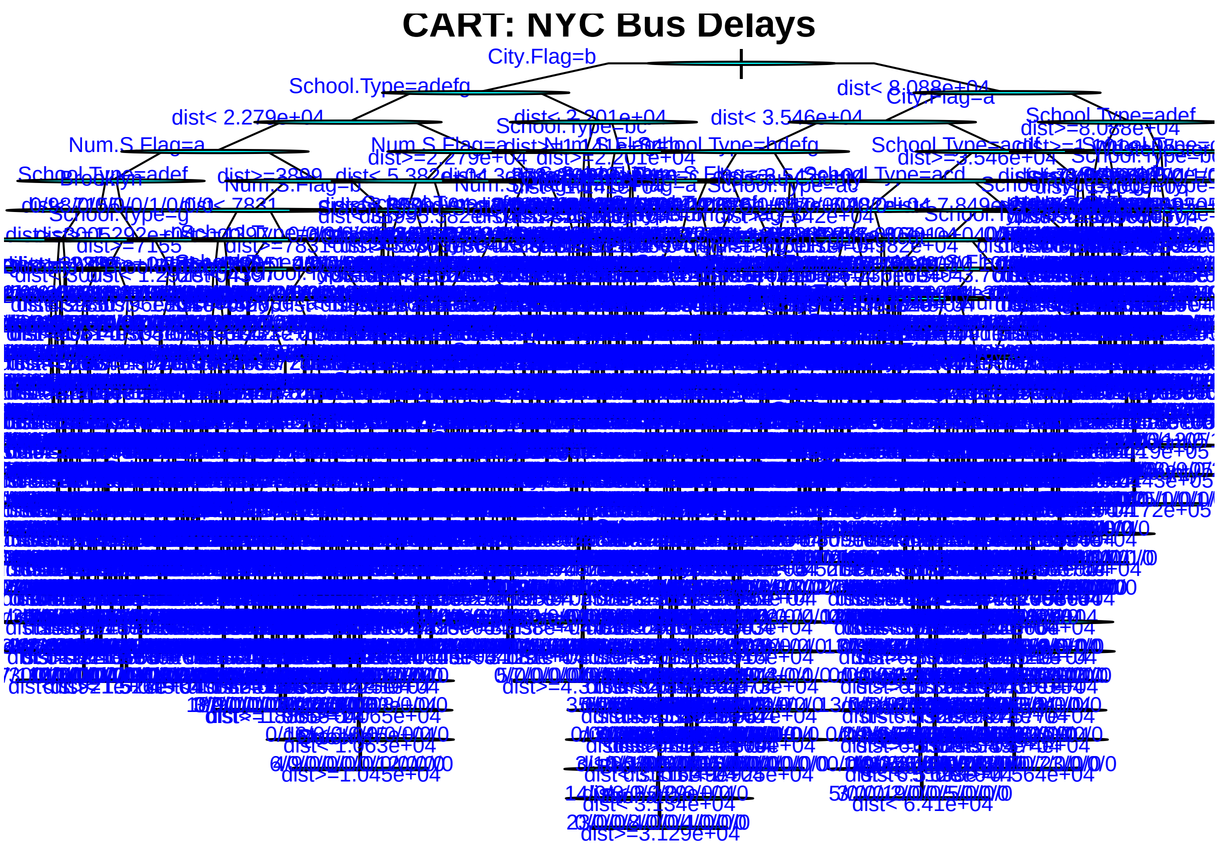
```
## 34  9.7759e-04     78   0.44977 0.46293 0.0034672
## 35  9.5879e-04     79   0.44879 0.46131 0.0034638
## 36  9.3999e-04     83   0.44495 0.45465 0.0034498
## 37  9.2119e-04     84   0.44401 0.45390 0.0034482
## 38  9.0239e-04    104   0.42330 0.45311 0.0034465
## 39  8.6479e-04    106   0.42149 0.44800 0.0034353
## 40  8.4599e-04    108   0.41976 0.44353 0.0034254
## 41  7.8959e-04    110   0.41807 0.43567 0.0034076
## 42  7.7079e-04    122   0.40792 0.42533 0.0033833
## 43  7.6453e-04    128   0.40269 0.42356 0.0033790
## 44  7.5199e-04    135   0.39668 0.42198 0.0033752
## 45  7.3946e-04    141   0.39216 0.41811 0.0033657
## 46  7.1439e-04    144   0.38995 0.41491 0.0033578
## 47  6.7679e-04    148   0.38709 0.40976 0.0033448
## 48  6.5799e-04    149   0.38641 0.40408 0.0033303
## 49  6.3919e-04    151   0.38510 0.40243 0.0033259
## 50  6.2039e-04    154   0.38318 0.39826 0.0033150
## 51  6.1413e-04    160   0.37927 0.39472 0.0033055
## 52  6.0159e-04    165   0.37607 0.39352 0.0033023
## 53  5.4519e-04    171   0.37246 0.38280 0.0032728
## 54  5.2639e-04    173   0.37137 0.37675 0.0032556
## 55  5.0760e-04    182   0.36663 0.37397 0.0032476
## 56  4.8880e-04    184   0.36562 0.36787 0.0032298
## 57  4.7626e-04    191   0.36220 0.36660 0.0032260
## 58  4.7000e-04    194   0.36077 0.36419 0.0032188
## 59  4.5120e-04    199   0.35840 0.36404 0.0032184
## 60  4.3240e-04    216   0.34990 0.35844 0.0032014
## 61  4.1830e-04    232   0.34167 0.35351 0.0031862
## 62  4.1360e-04    244   0.33588 0.34783 0.0031683
## 63  3.9480e-04    252   0.33257 0.34622 0.0031632
## 64  3.7600e-04    261   0.32877 0.33874 0.0031390
## 65  3.6660e-04    294   0.31539 0.33298 0.0031199
## 66  3.5720e-04    300   0.31309 0.32873 0.0031056
## 67  3.5344e-04    306   0.31095 0.32840 0.0031045
## 68  3.3840e-04    311   0.30918 0.32321 0.0030867
## 69  3.1960e-04    336   0.30031 0.31193 0.0030469
## 70  3.1020e-04    356   0.29354 0.30945 0.0030379
## 71  3.0080e-04    362   0.29083 0.30369 0.0030168
## 72  2.8576e-04    380   0.28485 0.30271 0.0030132
## 73  2.8200e-04    409   0.27418 0.29745 0.0029935
## 74  2.7573e-04    423   0.27004 0.29696 0.0029916
## 75  2.6320e-04    430   0.26733 0.29174 0.0029716
## 76  2.5380e-04    446   0.26312 0.29102 0.0029689
## 77  2.5066e-04    453   0.26128 0.29102 0.0029689
## 78  2.4440e-04    459   0.25978 0.28294 0.0029371
## 79  2.3500e-04    501   0.24816 0.28177 0.0029325
## 80  2.2560e-04    505   0.24722 0.27982 0.0029247
## 81  2.0680e-04    547   0.23699 0.26760 0.0028744
## 82  2.0053e-04    563   0.23368 0.25993 0.0028417
## 83  1.8800e-04    566   0.23308 0.25880 0.0028369
## 84  1.7860e-04    603   0.22597 0.25256 0.0028095
## 85  1.6920e-04    622   0.22203 0.25143 0.0028045
## 86  1.6407e-04    653   0.21541 0.24624 0.0027812
## 87  1.6293e-04    666   0.21304 0.24624 0.0027812
```

```
## 88  1.5792e-04    669   0.21255 0.24624 0.0027812
## 89  1.5040e-04    674   0.21176 0.24564 0.0027785
## 90  1.3787e-04    729   0.20270 0.24030 0.0027540
## 91  1.3536e-04    735   0.20187 0.23936 0.0027496
## 92  1.3160e-04    740   0.20120 0.23913 0.0027485
## 93  1.2533e-04    759   0.19853 0.23913 0.0027485
## 94  1.2220e-04    762   0.19815 0.23473 0.0027279
## 95  1.1280e-04    768   0.19713 0.23473 0.0027279
## 96  1.0027e-04    784   0.19533 0.23387 0.0027238
## 97  9.3999e-05    792   0.19450 0.23173 0.0027136
## 98  8.4599e-05    800   0.19375 0.23158 0.0027129
## 99  7.5199e-05    804   0.19341 0.22936 0.0027022
## 100 6.7679e-05    831   0.19131 0.22936 0.0027022
## 101 6.2666e-05    836   0.19097 0.22917 0.0027013
## 102 5.6399e-05    842   0.19059 0.22909 0.0027009
## 103 5.0133e-05    873   0.18867 0.22906 0.0027008
## 104 4.7000e-05    879   0.18837 0.22864 0.0026988
## 105 4.5120e-05    883   0.18819 0.22864 0.0026988
## 106 3.7600e-05    888   0.18796 0.22785 0.0026949
## 107 3.0080e-05    923   0.18657 0.22785 0.0026949
## 108 2.8200e-05    933   0.18627 0.22778 0.0026946
## 109 2.5066e-05    941   0.18604 0.22737 0.0026926
## 110 2.2560e-05    944   0.18597 0.22740 0.0026927
## 111 1.8800e-05    954   0.18574 0.22706 0.0026911
## 112 7.5199e-06    962   0.18559 0.22706 0.0026911
## 113 0.0000e+00    967   0.18555 0.22699 0.0026907
```

```
plotcp(tree.train,minline=TRUE,col=4)
```



```
cptable <- data.frame(tree.train$cptable)
cp <- cptable[which.min(cptable$xerror),1]

tree.train.pruned <- rpart(df.train.tree,control=rpart.control(cp=cp, minsplit=30, xval=10, maxsurrogat

par(mai=c(0.1,0.1,0.1,0.1))
plot(tree.train.pruned,main="CART: NYC Bus Delays",col=3, compress=TRUE,
     branch=0.5,uniform=TRUE)
text(tree.train.pruned,cex=0.7,col=4,use.n=TRUE,fancy=TRUE,fwidth=0.2,fheight=0.02,bg=c(5))
```




```
round(prop.table(table(df.train.tree[,1], predict(tree.train.pruned,type="class")),1),2)
```

```
##
##                  Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx           0.85     0.06        0.00      0.03          0.00
##   Brooklyn        0.07     0.85        0.00      0.02          0.00
##   Connecticut     0.00     0.00        0.94      0.01          0.04
##   Manhattan       0.01     0.00        0.00      0.96          0.00
##   Nassau County   0.00     0.00        0.00      0.07          0.90
##   New Jersey      0.00     0.00        0.00      0.02          0.04
##   Queens          0.09     0.07        0.00      0.12          0.00
##   Rockland County 0.01     0.00        0.00      0.00          0.02
##   Staten Island   0.05     0.05        0.00      0.03          0.00
##   Westchester     0.00     0.01        0.00      0.01          0.00
##
##                  New Jersey Queens Rockland County Staten Island
##   Bronx                0.00   0.05            0.00          0.01
```

```
##   Brooklyn              0.00   0.05          0.00           0.01
##   Connecticut           0.00   0.00          0.00           0.00
##   Manhattan             0.00   0.02          0.00           0.00
##   Nassau County         0.00   0.01          0.01           0.00
##   New Jersey            0.90   0.00          0.03           0.00
##   Queens                0.00   0.70          0.00           0.01
##   Rockland County       0.00   0.03          0.94           0.00
##   Staten Island         0.00   0.04          0.00           0.82
##   Westchester           0.00   0.01          0.01           0.00
##
##                   Westchester
##   Bronx                  0.00
##   Brooklyn               0.00
##   Connecticut            0.01
##   Manhattan              0.00
##   Nassau County          0.00
##   New Jersey             0.00
##   Queens                 0.01
##   Rockland County        0.00
##   Staten Island          0.00
##   Westchester            0.96
```

```r
round(prop.table(table(df.holdout.tree[,1], predict(tree.train.pruned,newdata = df.holdout.tree[,-1],typ
```

```
##
##                   Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx            0.81     0.08        0.00      0.03          0.00
##   Brooklyn         0.08     0.84        0.00      0.02          0.00
##   Connecticut      0.00     0.00        0.92      0.00          0.08
##   Manhattan        0.02     0.00        0.00      0.94          0.00
##   Nassau County    0.01     0.00        0.00      0.09          0.86
##   New Jersey       0.00     0.00        0.00      0.02          0.02
##   Queens           0.09     0.08        0.00      0.14          0.00
##   Rockland County  0.01     0.00        0.00      0.00          0.04
##   Staten Island    0.05     0.07        0.00      0.03          0.00
##   Westchester      0.01     0.01        0.00      0.01          0.00
##
##                   New Jersey Queens Rockland County Staten Island
##   Bronx                 0.00   0.06            0.00          0.01
##   Brooklyn              0.00   0.05            0.00          0.01
##   Connecticut           0.00   0.00            0.00          0.00
##   Manhattan             0.00   0.03            0.00          0.00
##   Nassau County         0.01   0.03            0.00          0.00
##   New Jersey            0.92   0.00            0.04          0.00
##   Queens                0.00   0.66            0.00          0.01
##   Rockland County       0.00   0.03            0.92          0.00
##   Staten Island         0.00   0.05            0.00          0.78
##   Westchester           0.00   0.01            0.01          0.00
##
##                   Westchester
##   Bronx                  0.00
##   Brooklyn               0.00
##   Connecticut            0.00
##   Manhattan              0.00
##   Nassau County          0.00
```

```
##   New Jersey             0.00
##   Queens                 0.01
##   Rockland County        0.00
##   Staten Island          0.01
##   Westchester            0.95
```

```
tree.train.pruned$variable.importance
```

```
##        dist   City.Flag School.Type   Num.S.Flag
##  14579.1691   5808.6016   3472.5750    591.5293
```

```
City.Flag2 <- (df_final.temp$City==df_final.temp$Boro)*1
(table(df_final.temp$Garage_City, df_final.temp$Boro, City.Flag==0, df_final.temp$Number_Of_Students_On
       City.Flag2==1, df_final.temp$Run_Type=='Special Ed AM Run'))
```

```
## , ,  = FALSE,  = FALSE,  = FALSE,  = TRUE
##
##
##                  Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx            227        0           0         0             0
##   Brooklyn           0       23           0         0             0
##   Nassau             0        0           0         0             0
##   Queens             0        0           0         0             0
##   Staten Island      0        0           0         0             0
##   Westchester        0        0           0         0             0
##
##                  New Jersey Queens Rockland County Staten Island
##   Bronx                   0      0               0             0
##   Brooklyn                0      0               0             0
##   Nassau                  0      0               0             0
##   Queens                  0    105               0             0
##   Staten Island           0      0               0             7
##   Westchester             0      0               0             0
##
##                  Westchester
##   Bronx                    0
##   Brooklyn                 0
##   Nassau                   0
##   Queens                   0
##   Staten Island            0
##   Westchester              1
##
## , ,  = TRUE,  = FALSE,  = FALSE,  = TRUE
##
##
##                  Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx              0        7           0       119           110
##   Brooklyn         256        0           0       166           310
##   Nassau             0        1           0        46            33
##   Queens            42        6           0        15            24
##   Staten Island     10        0           0        11             3
##   Westchester        4       11           0        18            76
##
##                  New Jersey Queens Rockland County Staten Island
##   Bronx                   0     68              56             8
```

```
##   Brooklyn              0    158              78            72
##   Nassau                0      1               1             0
##   Queens                0      0              15             2
##   Staten Island         0      8               0             0
##   Westchester           0      3             208             0
##
##                Westchester
##   Bronx                  0
##   Brooklyn               1
##   Nassau                 0
##   Queens                 0
##   Staten Island          0
##   Westchester            0
##
## , ,  = FALSE,  = TRUE,  = FALSE,  = TRUE
##
##
##                Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx           156        0           0         0             0
##   Brooklyn          0       20           0         0             0
##   Nassau            0        0           0         0             0
##   Queens            0        0           0         0             0
##   Staten Island     0        0           0         0             0
##   Westchester       0        0           0         0             0
##
##                New Jersey Queens Rockland County Staten Island
##   Bronx                 0      0               0             0
##   Brooklyn              0      0               0             0
##   Nassau                0      0               0             0
##   Queens                0    222               0             0
##   Staten Island         0      0               0             9
##   Westchester           0      0               0             0
##
##                Westchester
##   Bronx                  0
##   Brooklyn               0
##   Nassau                 0
##   Queens                 0
##   Staten Island          0
##   Westchester            0
##
## , ,  = TRUE,  = TRUE,  = FALSE,  = TRUE
##
##
##                Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx             0        7           0        95           193
##   Brooklyn         33        0           0        58           212
##   Nassau            2        0           0        32            18
##   Queens           21       32           0        18           206
##   Staten Island     7        1           0        10             1
##   Westchester       4        1           1        11            55
##
##                New Jersey Queens Rockland County Staten Island
##   Bronx                 0     42              76             1
```

```
##   Brooklyn              0     48             15             9
##   Nassau                0      0              1             0
##   Queens                1      0             60             3
##   Staten Island         1      9              0             0
##   Westchester           0      6             33             0
##
##                 Westchester
##   Bronx                   0
##   Brooklyn                0
##   Nassau                  0
##   Queens                  0
##   Staten Island           0
##   Westchester             0
##
## , ,  = FALSE,  = FALSE,  = TRUE,  = TRUE
##
##
##                 Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx          3858        0           0         0             0
##   Brooklyn          0     5613           0         0             0
##   Nassau            0        0           0         0             0
##   Queens            0        0           0         0             0
##   Staten Island     0        0           0         0             0
##   Westchester       0        0           0         0             0
##
##                 New Jersey Queens Rockland County Staten Island
##   Bronx                 0      0               0             0
##   Brooklyn              0      0               0             0
##   Nassau                0      0               0             0
##   Queens                0    695               0             0
##   Staten Island         0      0               0           392
##   Westchester           0      0               0             0
##
##                 Westchester
##   Bronx                   0
##   Brooklyn                0
##   Nassau                  0
##   Queens                  0
##   Staten Island           0
##   Westchester          3688
##
## , ,  = TRUE,  = FALSE,  = TRUE,  = TRUE
##
##
##                 Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx             0       96           0      6781             0
##   Brooklyn        267        0           0      3969             0
##   Nassau            0        0           0         0             0
##   Queens            0       37           0       305             0
##   Staten Island    11       14           0       116             0
##   Westchester    1624        0          75         3             0
##
##                 New Jersey Queens Rockland County Staten Island
##   Bronx                97    536               0            17
```

```
##   Brooklyn            262   1003               0             89
##   Nassau                0     10               0              0
##   Queens               32      0               0              0
##   Staten Island       135    214               0              0
##   Westchester          28      0               0              0
##
##                 Westchester
##   Bronx                 146
##   Brooklyn              135
##   Nassau                  0
##   Queens                  0
##   Staten Island           0
##   Westchester             0
##
## , ,  = FALSE,  = TRUE,  = TRUE,  = TRUE
##
##
##                 Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx          3106        0           0         0             0
##   Brooklyn          0     3890           0         0             0
##   Nassau            0        0           0         0             0
##   Queens            0        0           0         0             0
##   Staten Island     0        0           0         0             0
##   Westchester       0        0           0         0             0
##
##                 New Jersey Queens Rockland County Staten Island
##   Bronx                  0      0               0             0
##   Brooklyn               0      0               0             0
##   Nassau                 0      0               0             0
##   Queens                 0   2861               0             0
##   Staten Island          0      0               0          2155
##   Westchester            0      0               0             0
##
##                 Westchester
##   Bronx                   0
##   Brooklyn                0
##   Nassau                  0
##   Queens                  0
##   Staten Island           0
##   Westchester           519
##
## , ,  = TRUE,  = TRUE,  = TRUE,  = TRUE
##
##
##                 Bronx Brooklyn Connecticut Manhattan Nassau County
##   Bronx             0       77           0      5111             0
##   Brooklyn        212        0           0      1749             0
##   Nassau            0        0           0         0             0
##   Queens            0       52           0       184             0
##   Staten Island     1       18           0        80             0
##   Westchester     436        0          46         7             0
##
##                 New Jersey Queens Rockland County Staten Island
##   Bronx                 84    639               0            19
```

```
##    Brooklyn            134    816                0              41
##    Nassau                0     16                0               0
##    Queens               36      0                0               3
##    Staten Island        40    103                0               0
##    Westchester          10      0                0               0
##
##                  Westchester
##    Bronx                  74
##    Brooklyn               48
##    Nassau                  0
##    Queens                  0
##    Staten Island           0
##    Westchester             0
```

Negative Binomial Regression

```r
dfNBR <- df_BBD[,c("Run_Type","Boro","Reason","Occurred_On")]
dfNBR <- subset(dfNBR, dfNBR$Run_Type=="Special Ed AM Run")
dfNBR <- subset(dfNBR, dfNBR$Reason=="Heavy Traffic")
dfNBR <- subset(dfNBR, dfNBR$Boro!='')
dfNBR <- subset(dfNBR, dfNBR$Boro!='All Boroughs')
dfNBR$Occurred_On <- as.POSIXct(dfNBR$Occurred_On, format="%m/%d/%Y")
dfNBR$Date <- format(dfNBR$Occurred_On, "%m/%d/%Y")
dfNBR$weekdays <- weekdays(dfNBR$Occurred_On)

cleaned <- dfNBR[c("Boro", "Date", "weekdays")]
counts <- cleaned %>% group_by(Boro,Date,weekdays) %>% summarise(count_delays=n())
counts$weekdays <- factor(counts$weekdays)
counts$Date <- as.Date(counts$Date, '%m/%d/%Y')

# Verify date/borough combinations with missing counts are holidays
date.range <-seq.Date(as.Date("2015-09-01"), as.Date("2018-06-28"), by="day")
wday <- weekdays(date.range)
date.range.df <- data.frame(date.range, wday)
colnames(date.range.df) <- c("Date", "weekdays")
date.range.df <- subset(date.range.df, date.range.df$weekdays != 'Sunday')
date.range.df <- subset(date.range.df, date.range.df$weekdays != 'Saturday')
Boroughs <- unique(counts$Boro)
date.range.boro.df <- merge(Boroughs, date.range.df, by=NULL)
all.dates <- left_join(date.range.boro.df, counts)
```

```
## Joining, by = c("Date", "weekdays")

## Warning: Column `weekdays` joining factors with different levels, coercing
## to character vector
```

```r
missing.dates <- subset(all.dates, is.na(all.dates$count_delays))
table(missing.dates$Date)
```

```
##
## 2015-09-01 2015-09-02 2015-09-03 2015-09-04 2015-09-07 2015-09-08
##         10         10         10         10         10         10
## 2015-09-09 2015-09-10 2015-09-11 2015-09-14 2015-09-15 2015-09-16
##         10         10         10         10         10         10
## 2015-09-17 2015-09-18 2015-09-21 2015-09-22 2015-09-23 2015-09-24
```

```
##         10         10         10         10         10         10
## 2015-09-25 2015-09-28 2015-09-29 2015-09-30 2015-10-01 2015-10-02
##         10         10         10         10         10         10
## 2015-10-05 2015-10-06 2015-10-07 2015-10-08 2015-10-09 2015-10-12
##         10         10         10         10         10         10
## 2015-10-13 2015-10-14 2015-10-15 2015-10-16 2015-10-19 2015-10-20
##         10         10         10         10         10         10
## 2015-10-21 2015-10-22 2015-10-23 2015-10-26 2015-10-27 2015-10-28
##         10         10         10         10         10         10
## 2015-10-29 2015-10-30 2015-11-02 2015-11-03 2015-11-04 2015-11-05
##         10         10         10         10         10         10
## 2015-11-06 2015-11-09 2015-11-10 2015-11-11 2015-11-12 2015-11-13
##         10         10         10         10         10         10
## 2015-11-16 2015-11-17 2015-11-18 2015-11-19 2015-11-20 2015-11-23
##         10         10         10         10         10         10
## 2015-11-24 2015-11-25 2015-11-26 2015-11-27 2015-11-30 2015-12-01
##         10         10         10         10         10         10
## 2015-12-02 2015-12-03 2015-12-04 2015-12-07 2015-12-08 2015-12-09
##         10         10         10         10         10         10
## 2015-12-10 2015-12-11 2015-12-14 2015-12-15 2015-12-16 2015-12-17
##         10         10         10         10         10         10
## 2015-12-18 2015-12-21 2015-12-22 2015-12-23 2015-12-24 2015-12-25
##         10         10         10         10         10         10
## 2015-12-28 2015-12-29 2015-12-30 2015-12-31 2016-01-01 2016-01-04
##         10         10         10         10         10         10
## 2016-01-05 2016-01-06 2016-01-07 2016-01-08 2016-01-11 2016-01-12
##         10         10         10         10         10         10
## 2016-01-13 2016-01-14 2016-01-15 2016-01-18 2016-01-19 2016-01-20
##         10         10         10         10         10         10
## 2016-01-21 2016-01-22 2016-01-25 2016-01-26 2016-01-27 2016-01-28
##         10         10         10         10         10         10
## 2016-01-29 2016-02-01 2016-02-02 2016-02-03 2016-02-04 2016-02-05
##         10         10         10         10         10         10
## 2016-02-08 2016-02-09 2016-02-10 2016-02-11 2016-02-12 2016-02-15
##         10         10         10         10         10         10
## 2016-02-16 2016-02-17 2016-02-18 2016-02-19 2016-02-22 2016-02-23
##         10         10         10         10         10         10
## 2016-02-24 2016-02-25 2016-02-26 2016-02-29 2016-03-01 2016-03-02
##         10         10         10         10         10         10
## 2016-03-03 2016-03-04 2016-03-07 2016-03-08 2016-03-09 2016-03-10
##         10         10         10         10         10         10
## 2016-03-11 2016-03-14 2016-03-15 2016-03-16 2016-03-17 2016-03-18
##         10         10         10         10         10         10
## 2016-03-21 2016-03-22 2016-03-23 2016-03-24 2016-03-25 2016-03-28
##         10         10         10         10         10         10
## 2016-03-29 2016-03-30 2016-03-31 2016-04-01 2016-04-04 2016-04-05
##         10         10         10         10         10         10
## 2016-04-06 2016-04-07 2016-04-08 2016-04-11 2016-04-12 2016-04-13
##         10         10         10         10         10         10
## 2016-04-14 2016-04-15 2016-04-18 2016-04-19 2016-04-20 2016-04-21
##         10         10         10         10         10         10
## 2016-04-22 2016-04-25 2016-04-26 2016-04-27 2016-04-28 2016-04-29
##         10         10         10         10         10         10
## 2016-05-02 2016-05-03 2016-05-04 2016-05-05 2016-05-06 2016-05-09
```

```
##         10         10         10         10         10         10
## 2016-05-10 2016-05-11 2016-05-12 2016-05-13 2016-05-16 2016-05-17
##         10         10         10         10         10         10
## 2016-05-18 2016-05-19 2016-05-20 2016-05-23 2016-05-24 2016-05-25
##         10         10         10         10         10         10
## 2016-05-26 2016-05-27 2016-05-30 2016-05-31 2016-06-01 2016-06-02
##         10         10         10         10         10         10
## 2016-06-03 2016-06-06 2016-06-07 2016-06-08 2016-06-09 2016-06-10
##         10         10         10         10         10         10
## 2016-06-13 2016-06-14 2016-06-15 2016-06-16 2016-06-17 2016-06-20
##         10         10         10         10         10         10
## 2016-06-21 2016-06-22 2016-06-23 2016-06-24 2016-06-27 2016-06-28
##         10         10         10         10         10         10
## 2016-06-29 2016-06-30 2016-07-01 2016-07-04 2016-07-05 2016-07-06
##         10         10         10         10         10         10
## 2016-07-07 2016-07-08 2016-07-11 2016-07-12 2016-07-13 2016-07-14
##         10         10         10         10         10         10
## 2016-07-15 2016-07-18 2016-07-19 2016-07-20 2016-07-21 2016-07-22
##         10         10         10         10         10         10
## 2016-07-25 2016-07-26 2016-07-27 2016-07-28 2016-07-29 2016-08-01
##         10         10         10         10         10         10
## 2016-08-02 2016-08-03 2016-08-04 2016-08-05 2016-08-08 2016-08-09
##         10         10         10         10         10         10
## 2016-08-10 2016-08-11 2016-08-12 2016-08-15 2016-08-16 2016-08-17
##         10         10         10         10         10         10
## 2016-08-18 2016-08-19 2016-08-22 2016-08-23 2016-08-24 2016-08-25
##         10         10         10         10         10         10
## 2016-08-26 2016-08-29 2016-08-30 2016-08-31 2016-09-01 2016-09-02
##         10         10         10         10         10         10
## 2016-09-05 2016-09-06 2016-09-07 2016-09-08 2016-09-09 2016-09-12
##         10         10         10         10         10         10
## 2016-09-13 2016-09-14 2016-09-15 2016-09-16 2016-09-19 2016-09-20
##         10         10         10         10         10         10
## 2016-09-21 2016-09-22 2016-09-23 2016-09-26 2016-09-27 2016-09-28
##         10         10         10         10         10         10
## 2016-09-29 2016-09-30 2016-10-03 2016-10-04 2016-10-05 2016-10-06
##         10         10         10         10         10         10
## 2016-10-07 2016-10-10 2016-10-11 2016-10-12 2016-10-13 2016-10-14
##         10         10         10         10         10         10
## 2016-10-17 2016-10-18 2016-10-19 2016-10-20 2016-10-21 2016-10-24
##         10         10         10         10         10         10
## 2016-10-25 2016-10-26 2016-10-27 2016-10-28 2016-10-31 2016-11-01
##         10         10         10         10         10         10
## 2016-11-02 2016-11-03 2016-11-04 2016-11-07 2016-11-08 2016-11-09
##         10         10         10         10         10         10
## 2016-11-10 2016-11-11 2016-11-14 2016-11-15 2016-11-16 2016-11-17
##         10         10         10         10         10         10
## 2016-11-18 2016-11-21 2016-11-22 2016-11-23 2016-11-24 2016-11-25
##         10         10         10         10         10         10
## 2016-11-28 2016-11-29 2016-11-30 2016-12-01 2016-12-02 2016-12-05
##         10         10         10         10         10         10
## 2016-12-06 2016-12-07 2016-12-08 2016-12-09 2016-12-12 2016-12-13
##         10         10         10         10         10         10
## 2016-12-14 2016-12-15 2016-12-16 2016-12-19 2016-12-20 2016-12-21
```

```
##         10         10         10         10         10         10
## 2016-12-22 2016-12-23 2016-12-26 2016-12-27 2016-12-28 2016-12-29
##         10         10         10         10         10         10
## 2016-12-30 2017-01-02 2017-01-03 2017-01-04 2017-01-05 2017-01-06
##         10         10         10         10         10         10
## 2017-01-09 2017-01-10 2017-01-11 2017-01-12 2017-01-13 2017-01-16
##         10         10         10         10         10         10
## 2017-01-17 2017-01-18 2017-01-19 2017-01-20 2017-01-23 2017-01-24
##         10         10         10         10         10         10
## 2017-01-25 2017-01-26 2017-01-27 2017-01-30 2017-01-31 2017-02-01
##         10         10         10         10         10         10
## 2017-02-02 2017-02-03 2017-02-06 2017-02-07 2017-02-08 2017-02-09
##         10         10         10         10         10         10
## 2017-02-10 2017-02-13 2017-02-14 2017-02-15 2017-02-16 2017-02-17
##         10         10         10         10         10         10
## 2017-02-20 2017-02-21 2017-02-22 2017-02-23 2017-02-24 2017-02-27
##         10         10         10         10         10         10
## 2017-02-28 2017-03-01 2017-03-02 2017-03-03 2017-03-06 2017-03-07
##         10         10         10         10         10         10
## 2017-03-08 2017-03-09 2017-03-10 2017-03-13 2017-03-14 2017-03-15
##         10         10         10         10         10         10
## 2017-03-16 2017-03-17 2017-03-20 2017-03-21 2017-03-22 2017-03-23
##         10         10         10         10         10         10
## 2017-03-24 2017-03-27 2017-03-28 2017-03-29 2017-03-30 2017-03-31
##         10         10         10         10         10         10
## 2017-04-03 2017-04-04 2017-04-05 2017-04-06 2017-04-07 2017-04-10
##         10         10         10         10         10         10
## 2017-04-11 2017-04-12 2017-04-13 2017-04-14 2017-04-17 2017-04-18
##         10         10         10         10         10         10
## 2017-04-19 2017-04-20 2017-04-21 2017-04-24 2017-04-25 2017-04-26
##         10         10         10         10         10         10
## 2017-04-27 2017-04-28 2017-05-01 2017-05-02 2017-05-03 2017-05-04
##         10         10         10         10         10         10
## 2017-05-05 2017-05-08 2017-05-09 2017-05-10 2017-05-11 2017-05-12
##         10         10         10         10         10         10
## 2017-05-15 2017-05-16 2017-05-17 2017-05-18 2017-05-19 2017-05-22
##         10         10         10         10         10         10
## 2017-05-23 2017-05-24 2017-05-25 2017-05-26 2017-05-29 2017-05-30
##         10         10         10         10         10         10
## 2017-05-31 2017-06-01 2017-06-02 2017-06-05 2017-06-06 2017-06-07
##         10         10         10         10         10         10
## 2017-06-08 2017-06-09 2017-06-12 2017-06-13 2017-06-14 2017-06-15
##         10         10         10         10         10         10
## 2017-06-16 2017-06-19 2017-06-20 2017-06-21 2017-06-22 2017-06-23
##         10         10         10         10         10         10
## 2017-06-26 2017-06-27 2017-06-28 2017-06-29 2017-06-30 2017-07-03
##         10         10         10         10         10         10
## 2017-07-04 2017-07-05 2017-07-06 2017-07-07 2017-07-10 2017-07-11
##         10         10         10         10         10         10
## 2017-07-12 2017-07-13 2017-07-14 2017-07-17 2017-07-18 2017-07-19
##         10         10         10         10         10         10
## 2017-07-20 2017-07-21 2017-07-24 2017-07-25 2017-07-26 2017-07-27
##         10         10         10         10         10         10
## 2017-07-28 2017-07-31 2017-08-01 2017-08-02 2017-08-03 2017-08-04
```

```
##         10         10         10         10         10         10
## 2017-08-07 2017-08-08 2017-08-09 2017-08-10 2017-08-11 2017-08-14
##         10         10         10         10         10         10
## 2017-08-15 2017-08-16 2017-08-17 2017-08-18 2017-08-21 2017-08-22
##         10         10         10         10         10         10
## 2017-08-23 2017-08-24 2017-08-25 2017-08-28 2017-08-29 2017-08-30
##         10         10         10         10         10         10
## 2017-08-31 2017-09-01 2017-09-04 2017-09-05 2017-09-06 2017-09-07
##         10         10         10         10         10         10
## 2017-09-08 2017-09-11 2017-09-12 2017-09-13 2017-09-14 2017-09-15
##         10         10         10         10         10         10
## 2017-09-18 2017-09-19 2017-09-20 2017-09-21 2017-09-22 2017-09-25
##         10         10         10         10         10         10
## 2017-09-26 2017-09-27 2017-09-28 2017-09-29 2017-10-02 2017-10-03
##         10         10         10         10         10         10
## 2017-10-04 2017-10-05 2017-10-06 2017-10-09 2017-10-10 2017-10-11
##         10         10         10         10         10         10
## 2017-10-12 2017-10-13 2017-10-16 2017-10-17 2017-10-18 2017-10-19
##         10         10         10         10         10         10
## 2017-10-20 2017-10-23 2017-10-24 2017-10-25 2017-10-26 2017-10-27
##         10         10         10         10         10         10
## 2017-10-30 2017-10-31 2017-11-01 2017-11-02 2017-11-03 2017-11-06
##         10         10         10         10         10         10
## 2017-11-07 2017-11-08 2017-11-09 2017-11-10 2017-11-13 2017-11-14
##         10         10         10         10         10         10
## 2017-11-15 2017-11-16 2017-11-17 2017-11-20 2017-11-21 2017-11-22
##         10         10         10         10         10         10
## 2017-11-23 2017-11-24 2017-11-27 2017-11-28 2017-11-29 2017-11-30
##         10         10         10         10         10         10
## 2017-12-01 2017-12-04 2017-12-05 2017-12-06 2017-12-07 2017-12-08
##         10         10         10         10         10         10
## 2017-12-11 2017-12-12 2017-12-13 2017-12-14 2017-12-15 2017-12-18
##         10         10         10         10         10         10
## 2017-12-19 2017-12-20 2017-12-21 2017-12-22 2017-12-25 2017-12-26
##         10         10         10         10         10         10
## 2017-12-27 2017-12-28 2017-12-29 2018-01-01 2018-01-02 2018-01-03
##         10         10         10         10         10         10
## 2018-01-04 2018-01-05 2018-01-08 2018-01-09 2018-01-10 2018-01-11
##         10         10         10         10         10         10
## 2018-01-12 2018-01-15 2018-01-16 2018-01-17 2018-01-18 2018-01-19
##         10         10         10         10         10         10
## 2018-01-22 2018-01-23 2018-01-24 2018-01-25 2018-01-26 2018-01-29
##         10         10         10         10         10         10
## 2018-01-30 2018-01-31 2018-02-01 2018-02-02 2018-02-05 2018-02-06
##         10         10         10         10         10         10
## 2018-02-07 2018-02-08 2018-02-09 2018-02-12 2018-02-13 2018-02-14
##         10         10         10         10         10         10
## 2018-02-15 2018-02-16 2018-02-19 2018-02-20 2018-02-21 2018-02-22
##         10         10         10         10         10         10
## 2018-02-23 2018-02-26 2018-02-27 2018-02-28 2018-03-01 2018-03-02
##         10         10         10         10         10         10
## 2018-03-05 2018-03-06 2018-03-07 2018-03-08 2018-03-09 2018-03-12
##         10         10         10         10         10         10
## 2018-03-13 2018-03-14 2018-03-15 2018-03-16 2018-03-19 2018-03-20
```

```
##         10         10         10         10         10         10
## 2018-03-21 2018-03-22 2018-03-23 2018-03-26 2018-03-27 2018-03-28
##         10         10         10         10         10         10
## 2018-03-29 2018-03-30 2018-04-02 2018-04-03 2018-04-04 2018-04-05
##         10         10         10         10         10         10
## 2018-04-06 2018-04-09 2018-04-10 2018-04-11 2018-04-12 2018-04-13
##         10         10         10         10         10         10
## 2018-04-16 2018-04-17 2018-04-18 2018-04-19 2018-04-20 2018-04-23
##         10         10         10         10         10         10
## 2018-04-24 2018-04-25 2018-04-26 2018-04-27 2018-04-30 2018-05-01
##         10         10         10         10         10         10
## 2018-05-02 2018-05-03 2018-05-04 2018-05-07 2018-05-08 2018-05-09
##         10         10         10         10         10         10
## 2018-05-10 2018-05-11 2018-05-14 2018-05-15 2018-05-16 2018-05-17
##         10         10         10         10         10         10
## 2018-05-18 2018-05-21 2018-05-22 2018-05-23 2018-05-24 2018-05-25
##         10         10         10         10         10         10
## 2018-05-28 2018-05-29 2018-05-30 2018-05-31 2018-06-01 2018-06-04
##         10         10         10         10         10         10
## 2018-06-05 2018-06-06 2018-06-07 2018-06-08 2018-06-11 2018-06-12
##         10         10         10         10         10         10
## 2018-06-13 2018-06-14 2018-06-15 2018-06-18 2018-06-19 2018-06-20
##         10         10         10         10         10         10
## 2018-06-21 2018-06-22 2018-06-25 2018-06-26 2018-06-27 2018-06-28
##         10         10         10         10         10         10
```

```r
nSample <- 5211
train_size_nbr <- round(nSample * (70 / 100))
set.seed(7881)
train_data_nbr <- sample(1:nrow(counts), train_size_nbr)
df.train.nbr <- counts[train_data_nbr, ]
df.holdout.nbr <- counts[-train_data_nbr, ]

# Train and Holdout
nb.fit.train <- glm.nb(count_delays ~ Boro + relevel(weekdays,"Monday"), data=df.train.nbr)
summary(nb.fit.train)
```

```
##
## Call:
## glm.nb(formula = count_delays ~ Boro + relevel(weekdays, "Monday"),
##     data = df.train.nbr, init.theta = 2.969837641, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6540  -0.7625  -0.1780   0.4045   3.2757
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                            3.20053    0.03765  85.017  < 2e-16
## BoroBrooklyn                           0.09498    0.04143   2.292  0.02188
## BoroConnecticut                       -2.90200    0.13461 -21.559  < 2e-16
## BoroManhattan                          0.82098    0.04054  20.251  < 2e-16
## BoroNassau County                     -2.11968    0.05219 -40.612  < 2e-16
## BoroNew Jersey                        -2.14318    0.05840 -36.697  < 2e-16
## BoroQueens                            -0.21582    0.04176  -5.168 2.37e-07
```

```
## BoroRockland County                   -2.57204    0.07019 -36.646  < 2e-16
## BoroStaten Island                     -1.19506    0.04451 -26.852  < 2e-16
## BoroWestchester                       -0.95144    0.04363 -21.807  < 2e-16
## relevel(weekdays, "Monday")Friday    -0.22569    0.03639  -6.202 5.59e-10
## relevel(weekdays, "Monday")Thursday  -0.10921    0.03596  -3.037  0.00239
## relevel(weekdays, "Monday")Tuesday   -0.15762    0.03602  -4.375 1.21e-05
## relevel(weekdays, "Monday")Wednesday -0.13671    0.03590  -3.809  0.00014
##
## (Intercept)                          ***
## BoroBrooklyn                         *
## BoroConnecticut                      ***
## BoroManhattan                        ***
## BoroNassau County                    ***
## BoroNew Jersey                       ***
## BoroQueens                           ***
## BoroRockland County                  ***
## BoroStaten Island                    ***
## BoroWestchester                      ***
## relevel(weekdays, "Monday")Friday    ***
## relevel(weekdays, "Monday")Thursday  **
## relevel(weekdays, "Monday")Tuesday   ***
## relevel(weekdays, "Monday")Wednesday ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(2.9698) family taken to be 1)
##
##     Null deviance: 11678.6  on 3647  degrees of freedom
## Residual deviance:  3522.2  on 3634  degrees of freedom
## AIC: 23361
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  2.9698
##           Std. Err.:  0.0881
##
##  2 x log-likelihood:  -23331.1510
```

```
RMSE.train <- sqrt(mean((nb.fit.train$y-nb.fit.train$fitted.values)^2))
RMSE.holdout <- sqrt(mean((df.holdout.nbr$count_delays-predict(nb.fit.train, newdata=df.holdout.nbr[,-4
round(cbind(RMSE.train, RMSE.holdout),2)
```

```
##      RMSE.train RMSE.holdout
## [1,]      13.76        14.23
```

```
# Predict delay count by day of week, borough
new.data <- merge(unique(counts$Boro), unique(counts$weekdays), by=NULL)
colnames(new.data) <- c("Boro", "weekdays")
predictions <- predict(nb.fit.train, newdata = new.data, type='response')
pred.df <- data.frame(new.data, round(predictions))

Manhattan <- subset(pred.df, pred.df$Boro=='Manhattan')
Queens <- subset(pred.df, pred.df$Boro=='Queens')
Bronx <- subset(pred.df, pred.df$Boro=='Bronx')
```

```r
Staten.Island <- subset(pred.df, pred.df$Boro=='Staten Island')
Brooklyn <- subset(pred.df, pred.df$Boro=='Brooklyn')

Manhattan <- Manhattan[c(3,1,2,4,5),]
Queens <- Queens[c(3,1,2,4,5),]
Bronx <- Bronx[c(3,1,2,4,5),]
Staten.Island <- Staten.Island[c(3,1,2,4,5),]
Brooklyn <- Brooklyn[c(3,1,2,4,5),]

plot(Manhattan$round.predictions.,ylim=c(0,65), type='l', ylab="Count of Delays", xlab = 'Weekday', xax
     main = "Predicted Count of Delays, By Weekday")
axis(1, at=1:5, labels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))
lines(Queens$round.predictions.,type='l',col='red')
lines(Bronx$round.predictions.,type='l',col='blue')
lines(Staten.Island$round.predictions.,type='l',col='green')
lines(Brooklyn$round.predictions.,type='l',col='purple')
legend('topright', legend=c("Manhattan", "Brooklyn", "Bronx", "Queens", "Staten Island"),
       col=c("black", "purple", "blue", "red", "green"), lty=1, cex=0.6)
```