```
#######################################
## Street Sweeping Citations Cleanup ##
#######################################

## Read in the data set and set as 'p'
p <- read.csv("streetsweeping-citations-2018.csv", header=T)

# Add columns for citation issue year, month, calendar day, and week day
p$Issue.Year <- as.numeric(format(as.Date(p$Issue.Date),format="%Y"))
p$Issue.Month <- as.numeric(format(as.Date(p$Issue.Date),format="%m"))
p$Issue.Day <- as.numeric(format(as.Date(p$Issue.Date),format="%d"))
p$Issue.Weekday <- format(as.Date(p$Issue.Date),format="%a")

# Review week day values and then numerically represent citation issue week day comlumn
library(dplyr)

unique(p$Issue.Weekday)

p <-
  p %>%
  mutate(Issue.Weekday = recode(Issue.Weekday, "Fri"=6, "Wed"=4, "Thu"=5, "Tue"=3, "Sat"=7, "Mon"=2, "Sun"=1))

## Convert issue time numeric value into an actual time format, first by finding hour:00 values and then by inserting 0 for NA
p[which(nchar(p$Issue.time)==1),4] <- p[which(nchar(p$Issue.time)==1),4]*100
p[which(nchar(p$Issue.time)==2 & p$Issue.time <= 24),4] <- p[which(nchar(p$Issue.time)==2 & p$Issue.time <= 24),4]*100
p[which(is.na(p$Issue.time)),4] <- 0

# convert numeric to time
library(chron)

p$Issue.time <- times(sub("(.{2})","\\1:",sprintf("%04d:00",p$Issue.time)))

# bins representing four hour increments, starting at 12:00 AM, converting 0:00 to 0
p$Issue.time.bin <- cut(p$Issue.time, breaks=6, labels=F)
p[which(p$Issue.time == 0),25] <- 0

## Break up plate expiry date into month and year, adding a flag for expired plates
p[which(nchar(p$Plate.Expiry.Date)==1),8] <- 0
p[which(nchar(p$Plate.Expiry.Date)==2),8] <- 0
p[which(nchar(p$Plate.Expiry.Date)==8),8] <- substr(p[which(nchar(p$Plate.Expiry.Date)==8),8],1,6)
p[which(is.na(p$Plate.Expiry.Date)),8] <- 0

p$Plate.Expiry.Year <- substr(p$Plate.Expiry.Date,1,4)
p$Plate.Expiry.Month <- substr(p$Plate.Expiry.Date,5,6)

p[which(p$Plate.Expiry.Month==""),27] <- 0
p[which(p$Plate.Expiry.Month > 12),27] <- 0

tapply(p$X, p$Plate.Expiry.Year, NROW) # check for drop off in registration expiry count (assuming > 2020 since CA renews annually)
p[which(p$Plate.Expiry.Year > 2020),27] <- 0

# Plates expired if the expiry month/year <= the citation month/year
p$Plate.Expired.Flag <- ifelse(p$Issue.Year >= p$Plate.Expiry.Year & p$Issue.Month >= p$Plate.Expiry.Month, 0, 1)

## Clean up the car attributes -- Make, Body Style, Color
tapply(p$X, p$Make, NROW)

# Convert the well known car make values to the most prevalant representation, and then set all others as "OTHR"
p <-
  p %>%
  mutate(Make = recode(Make,
                       "ACRU"="ACUR", "ACU"="ACUR","AUD"="AUDI","BMV"="BMW","CAD"="CADI","GM"="GMC","HNDA"="HOND","HONDA"="HOND",
                       "HYAU"="HYUN","HYN"="HYUN","HYND"="HYUN","INF"="INFI","ISUZ"="ISU","JAG"="JAGU","JAGR"="JAGU","KAWA"="KAWK",
                       "LEX"="LEXS","LEXU"="LEXS","LNCI"="LINC","LND"="LNDR","LROV"="LNDR","LRVR"="LNDR","MADZ"="MAZD","MAZ"="MAZD",
                       "MAZA"="MAZD","MBEN"="BENT","MBNZ"="BENZ","MERK"="MERC","MINI"="MNNI","MIT"="MITS","OLD"="OLDS","PORC"="PORS","ROLS"="ROL",
                       "ROVE"="RROV","RRVR"="RROV","SABA"="SAAB","SAA"="SAAB","SATR"="STRN","SATU"="STRN","SMAR"="SMRT","SUB"="SUBA","SUBU"="SUBA",
                       "SUSU"="SUZI","SUZ"="SUZI","SUZK"="SUZI","SUZU"="SUZI","TELS"="TESL","TOYO"="TOYT","TOYOT"="TOYT","VOL"="VOLK","VW"="VOLK",
                       'OTHR'='OTHR','UNK'='OTHR','MASE'='OTHR','HD'='OTHR','CHEC'='OTHR','COOP'='OTHR','MZ'='OTHR','SUNR'='OTHR',
                       'HNO'='OTHR','PTRB'='OTHR','DAEW'='OTHR','KW'='OTHR','IND'='OTHR','AUBU'='OTHR','JENS'='OTHR',
                       'FREI'='OTHR','STLG'='OTHR','EGLE'='OTHR','GRUM'='OTHR','WHIT'='OTHR','PEUG'='OTHR','HARL'='OTHR','INTE'='OTHR',
                       'LAND'='OTHR','FLEE'='OTHR','OTHE'='OTHR','VESP'='OTHR','AUHE'='OTHR','DUES'='OTHR','FSKR'='OTHR','RENA'='OTHR','STU'='OTHR',
                       'UN'='OTHR','FALC'='OTHR','GENE'='OTHR','LXS'='OTHR','PACK'='OTHR','RR'='OTHR','SHAS'='OTHR','VINO'='OTHR','ALLE'='OTHR',
                       'AVTI'='OTHR','BMER'='OTHR','BOUN'='OTHR','BRAV'='OTHR','BROU'='OTHR','CITR'='OTHR','COAC'='OTHR','COLU'='OTHR','CUSH'='OTHR',
                       'DOLP'='OTHR','DUCA'='OTHR','EXPC'='OTHR','FLAI'='OTHR','FRLI'='OTHR','GENS'='OTHR','HINO'='OTHR','HRLY'='OTHR','ITAS'='OTHR',
                       'KENW'='OTHR','LAZY'='OTHR','LDRR'='OTHR','LDRV'='OTHR','LNAR'='OTHR','MALR'='OTHR','MER'='OTHR','MERB'='OTHR','MONA'='OTHR',
                       'MRCB'='OTHR','MTZ'='OTHR','OTAR'='OTHR','PCH'='OTHR','PETE'='OTHR','PRRO'='OTHR','RANG'='OTHR','REO'='OTHR','SAND'='OTHR',
                       'SANY'='OTHR','SFAR'='OTHR','STEL'='OTHR','SUPR'='OTHR','TIOG'='OTHR','UKN'='OTHR','UNKN'='OTHR','VALI'='OTHR','VAUA'='OTHR',
                       'WILD'='OTHR','ZNEN'='OTHR','ZZ'='OTHR'))

# Convert the missing values to "OTHR"
p[which(p$Make==""),10] <- "OTHR"

# Build flag for domestic vs. imported vehicles (per https://en.wikipedia.org/wiki/Passenger_vehicles_in_the_United_States#Import_makes,_with_some_assembly_in_the_U.S.)
imp <- c("ACUR","HYUN","LEXS","SUBA","VOLV","INFI","KIA","vOLK","NISS","FERR","LAMO","ROL","BMW","BENZ",
         "PORS","MAZD","HOND","ALFA","BUGA","LNDR","ASTO","JAGU","MNNI","AUDI","MITS","BENT","FIAT","SMRT")

p$Make.Import.Flag <- ifelse(p$Make %in% imp,1,0)

lux <- c("INFI","ALFA","LNDR","ACUR","JAGU","LINC","CADI","LEXS","VOLV","BENZ","BMW","AUDI","PORS","ROL","FERR","LAMO","TESL","BUGA","ASTO","BENT","RROV")

p$car.make.luxury.flag <- ifelse(p$car.make %in% lux,1,0)

remove(imp,lux)

# Convert colors into most prevalant representation, and then set all others to "OT"
tapply(p$X, p$Color, NROW)
```

```
p <-
  p %>%
  mutate(Color = recode(Color,
                        "BE"="BG","BI"="BG","BL"="BK","BR"="BN","BU"="OT","CO"="OT","CR"="OT","GR"="GN","MA"="MR","MN"="MR","PL"="OT","PU"="PR","RE"="RD",
                        "RU"="OT","SI"="SL","TA"="TN","TU"="OT","UN"="OT","UT"="OT","WI"="WT","WH"="WT","YL"="YE"))

# Convert the missing values to "OT"
p[which(p$Color==""),12] <- "OT"

# Convert body styls into most prevalant representation
tapply(p$X, p$Body.Style, NROW)

p <-
  p %>%
  mutate(Body.Style = recode(Body.Style,
                             'LM'='OT','RV'='OT','SC'='OT','TL'='OT','SW'='OT','UT'='OT','VA'='OT','20'='OT','PP'='OT','VV'='OT','|'='OT','BK'='OT',
                             'BO'='OT','JE'='OT','MI'='OT','MO'='OT','MT'='OT','MY'='OT','PV'='OT','PY'='OT','S'='OT','SP'='OT','U'='OT','WI'='OT'))

p[which(p$Body.Style==""),11] <- "OT"

## Clean up violation code, so that all are consistent
tapply(p$X, p$Violation.code, NROW)

p[which(p$Violation.code=="8069BS"),16] <- "80.69BS"

## Convert and clean up the latitude and longitude columns
library(proj4)
pj <- "+proj=lcc +lat_1=34.03333333333333 +lat_2=35.46666666666667 +lat_0=33.5 +lon_0=-118 +x_0=2000000 +y_0=500000.0000000002 +ellps=GRS80 +datum=NAD83
+to_meter=0.3048006096012192 no_defs"

p <- cbind(p, data.frame(project(data.frame(p$Latitude, p$Longitude), proj = pj, inverse = TRUE)))

names(p)[c(30, 31)] <- c('lat', 'lon') # rename the columns...they come in as x and y

remove(pj)

## Drop unnecessary columns and reorder/rename columns for ease of use with the rest of the project
drop <- c(1,6,9,19:20)
p<- p[,-drop]
remove(drop)

col.order <- c(1:2,16:19,3,20,12,4,11,10,25:26,13:15,6,21:23,5,7,24,8,9)
p <- p[,col.order]
remove(col.order)


col.name <- c('ticket.number','issue.date','issue.year','issue.month','issue.day','issue.weekday','issue.time','issue.time.bin',
              'agency.id','meter.id','route.id','issue.address','issue.address.lat','issue.address.lon','violation.id','violation.desc',
              'violation.fine.amt','plate.expire.date','plate.expire.year','plate.expire.month','plate.expire.flag','plate.state','car.make',
              'car.make.import.flag','car.bodystyle','car.color')

colnames(p) <- col.name
remove(col.name)

## Handle NA and null values/levels in data
# Convert bad data
levels(p$meter.id)[levels(p$meter.id)==""] <- "0"
levels(p$route.id)[levels(p$route.id)==""] <- "0"
levels(p$issue.address)[levels(p$issue.address)==""] <- "No Address"
p[which(p$issue.address == "No Address"),13] <- 0 # convert lat to 0 when no address is present
p[which(p$issue.address == "No Address"),14] <- 0 # convert lon to 0 when no address is present

# Check conversion worked
NROW(which(p$meter.id == "0"))
NROW(which(p$route.id == "0"))
NROW(which(p$issue.address == "No Address"))

## Write clean data set out for group share
write.csv(p,"streetsweeping-citations-2018-clean.csv")


##########################################
## Street Sweeping Citations Description ##
##########################################

## Read in the clean data set
p <- read.csv("streetsweeping-citations-2018-clean.csv", header=T)
str(p)
View(head(p),15)

mean(p$car.make.import.flag)

# Remove observations with relevant missing data
p <- p[which(p$issue.time.bin!=0),]
p <- p[which(p$route.id!=0),]
p <- p[which(p$issue.address.lat!=0),]
p <- p[which(p$plate.expire.date!=0),]

# Convert factor formatted columns to numeric
p$meter.id.n <- as.numeric(p$meter.id)
p$route.id.n <- as.numeric(p$route.id)
p$plate.state.n <- as.numeric(p$plate.state)
p$car.make.n <- as.numeric(p$car.make)
p$car.bodystyle.n <- as.numeric(p$car.bodystyle)
p$car.color.n <- as.numeric(p$car.color)
```

```
p.n.o <- c(25,5,7,9,14:15,22,30,32:33)
p.n <- p[,p.n.o]
remove(p.n.o)

# Create the train and test data sets
p.n.rand <- sample(1:dim(p.n)[1])
p.n.cut <- round((nrow(p.n)/3)*2,digits=0)
p.n.train <- p.n[p.n.rand[1:p.n.cut],]
p.n.test <- p.n[p.n.rand[(p.n.cut+1):nrow(p.n)],]

# XG Boost Model Attempt
library(xgboost)
library(caret)
library(stringr)
library(readr)

xgb <- xgboost(data=data.matrix(p.n.train[,-1]),
               label=p.n.train[,1],
               #silent=1,
               eta=1,
               max_depth=50000,
               nrounds=3,
               #subsample=0.5,
               #colsample_bytree=0.5,
               #seed=1,
               #eval_metric="auc",
               objective="binary:logistic",
               #num_class=2,
               nthread=2)

xgb.p <- predict(xgb, data.matrix(p.n.test[,-1]))

# Test Results
model <- xgb.dump(xgb, with_stats = T)
names <- dimnames(data.matrix(p.n.test[,-1][[2]]))
importance <- xgb.importance(names, model=xgb)
xgb.plot.importance(importance[1:10,])

xgb.p.n.df <- data.frame(p.n.test[,1],round(xgb.p))
xgb.p.n.t <- table(xgb.p.n.df)
xgb.p.n.t
(xgb.p.n.t[1,1]+xgb.p.n.t[2,2])/sum(xgb.p.n.t)

# Random Forest Model Attempt
library(randomForest)
?randomForest
rf <- randomForest(p.n.train[,-1],p.n.train[,1])
rf.p <- predict(rf, p.n.test)
rf$importance

rf.p.n.df <- data.frame(p.n.test[,1],round(rf.p))
rf.p.n.t <- table(rf.p.n.df)
rf.p.n.t
(rf.p.n.t[1,1]+rf.p.n.t[2,2])/sum(rf.p.n.t)

# Logistic Regression Model Attempt
library("aod")
summary(p.n)
sapply(p.n, sd)
xtabs(~p.n$car.make.import.flag + p.n$issue.month, data = p.n)

p.n.train.glm <- p.n.train[,-c(2,4)]
p.n.test.glm <- p.n.test[,-c(2,4)]

glm <- glm(car.make.import.flag ~ ., data = p.n.train.glm, family = "binomial")
summary(glm)

glm.p <- predict(glm, p.n.test.glm)
glm.p.n.df <- data.frame(p.n.test.glm[,1],ifelse(glm.p <= 0, 0, 1))
glm.p.n.t <- table(glm.p.n.df)
glm.p.n.t
(glm.p.n.t[1,1]+glm.p.n.t[2,2])/sum(glm.p.n.t)

# PIVOT -- Attempt to predict weekly revenue
ll.test <- within(p.n, {
  grp.lat = cut(issue.address.lat, 10, labels=FALSE)
  grp.lon = cut(issue.address.lon, 10, labels=FALSE)
})

head(ll.test)
unique(ll.test$grp.lat)

colnames(p.n) <-
c("car.make.import.flag","issue.month","issue.weekday","issue.time.bin","issue.address.lon","issue.address.lat","plate.expire.flag","plate.state.n","car.bodystyle.n","car.
color.n")
head(p.n)

# Van Nuys center
p$vn.lon <- -118.4514
p$vn.lat <- 34.189857

# Hollywood center
p$hw.lon <- -118.3287
p$hw.lat <- 34.0928
```

```
# San Pedro center
p$sp.lon <- -118.2922
p$sp.lat <- 33.7361

p$vn.d <- sqrt((p$vn.lat-p$issue.address.lon)^2 + (p$vn.lon-p$issue.address.lat)^2)
p$hw.d <- sqrt((p$hw.lat-p$issue.address.lon)^2 + (p$hw.lon-p$issue.address.lat)^2)
p$sp.d <- sqrt((p$sp.lat-p$issue.address.lon)^2 + (p$sp.lon-p$issue.address.lat)^2)

summary(p.n)
tail(p.n.train)

p.n$issue.cityCenter <- ifelse(apply(p.n[17:19],1,FUN=min)==p.n$vn.d,1,
                               ifelse(apply(p.n[17:19],1,FUN=min)==p.n$hw.d,2,3))

p.n$vn.d.q <- within(p.n, quartile <- as.integer(cut(vn.d, quantile(vn.d, probs=0:4/4), include.lowest=TRUE)))$quartile
p.n$hw.d.q <- within(p.n, quartile <- as.integer(cut(hw.d, quantile(hw.d, probs=0:4/4), include.lowest=TRUE)))$quartile
p.n$sp.d.q <- within(p.n, quartile <- as.integer(cut(sp.d, quantile(sp.d, probs=0:4/4), include.lowest=TRUE)))$quartile

p.n$cityCenter <- ifelse(p.n$issue.cityCenter==1){print(p.n$vn.d.q)}{
       ifelse(p.n$issue.cityCenter==2){print(p.n$hw.d.q)}}{print(p.n$sp.d.q)}

p.n$issue.CityCenter.dist <-
if (p.n$issue.cityCenter==1) {
  print(p.n$vn.d.q)
} else {
  if (p.n$issue.cityCenter==2)  {
    print(p.n$hw.d.q)
  } else {
    print(p.n$sp.d.q)
  }
}

p.n <- p.n[,-c(11:19,21:23)]
p.n <- p.n[,-c(4:6)]

p$issue.cityCenter <- p.n$issue.cityCenter
p$issue.cityCenter.dist <- p.n$issue.cityCenter.dist

tapply(p$ticket.number,list(p$issue.weekday,p$issue.cityCenter, p$issue.cityCenter.dist),NROW)

p$issue.week <- as.numeric(format(as.Date(p$issue.date), '%V'))

p.n <- data.frame(week.total=tapply(p$ticket.number,p$issue.week,NROW),stringsAsFactors = F)

week.mean <- mean(p.n$week.total)

p.n$above.avg <- ifelse(p.n$week.total >= week.mean,1,0)

remove(week.mean)
tapply(p$X,p$issue.cityCenter.dist,NROW)

p$issue.cityCenter.dist <- as.numeric(p$issue.cityCenter.dist)
p$issue.calday <- as.Date(p$issue.date)

library(plyr)
p.summary2 <- ddply(p,"issue.calday",summarise,
      fineCnt = NROW(which(X>=0)),
      fineAmt = NROW(which(X>=0))*73,
      time3 = NROW(which(issue.time.bin==3)),
      time4 = NROW(which(issue.time.bin==4)),
      wkDay2 = NROW(which(issue.weekday==2)),
      wkDay3 = NROW(which(issue.weekday==3)),
      wkDay4 = NROW(which(issue.weekday==4)),
      wkDay5 = NROW(which(issue.weekday==5)),
      wkDay6 = NROW(which(issue.weekday==6)),
      holiday = NROW(which(holiday.ind==1)),
      cityCnt12 = NROW(which(issue.cityCenter==1 & issue.cityCenter.dist==2)),
      cityCnt13 = NROW(which(issue.cityCenter==1 & issue.cityCenter.dist==3)),
      cityCnt14 = NROW(which(issue.cityCenter==1 & issue.cityCenter.dist==4)),
      cityCnt21 = NROW(which(issue.cityCenter==2 & issue.cityCenter.dist==1)),
      cityCnt22 = NROW(which(issue.cityCenter==2 & issue.cityCenter.dist==2)),
      cityCnt23 = NROW(which(issue.cityCenter==2 & issue.cityCenter.dist==3)),
      cityCnt24 = NROW(which(issue.cityCenter==2 & issue.cityCenter.dist==4)),
      cityCnt34 = NROW(which(issue.cityCenter==3 & issue.cityCenter.dist==4)),
      cityCntVn1 = NROW(which(issue.cityCenter.clust==1 & issue.cityCenter.subclust==1)),
      cityCntVn2 = NROW(which(issue.cityCenter.clust==1 & issue.cityCenter.subclust==2)),
      cityCntVn3 = NROW(which(issue.cityCenter.clust==1 & issue.cityCenter.subclust==3)),
      cityCntHw1 = NROW(which(issue.cityCenter.clust==3 & issue.cityCenter.subclust==1)),
      cityCntHw2 = NROW(which(issue.cityCenter.clust==3 & issue.cityCenter.subclust==2)),
      cityCntHw3 = NROW(which(issue.cityCenter.clust==3 & issue.cityCenter.subclust==3)),
      cityCntHw4 = NROW(which(issue.cityCenter.clust==3 & issue.cityCenter.subclust==4)),
      cityCntSp1 = NROW(which(issue.cityCenter.clust==2 & issue.cityCenter.subclust==1)),
      cityCntSp2 = NROW(which(issue.cityCenter.clust==2 & issue.cityCenter.subclust==2)),
      cityCntSp3 = NROW(which(issue.cityCenter.clust==2 & issue.cityCenter.subclust==3)),
      cityCntSp4 = NROW(which(issue.cityCenter.clust==2 & issue.cityCenter.subclust==4)),
      cityCntSp5 = NROW(which(issue.cityCenter.clust==2 & issue.cityCenter.subclust==5)),
      agency51 = NROW(which(agency.id==51)),
      agency53 = NROW(which(agency.id==53)),
      agency54 = NROW(which(agency.id==54)),
      agency55 = NROW(which(agency.id==55)),
      agency56 = NROW(which(agency.id==56)),
      plateExp = NROW(which(plate.expire.flag==1)),
      plateCA = NROW(which(plate.state=="CA")),
      plateNCA = NROW(which(plate.state!="CA")),
      carImport = NROW(which(car.make.import.flag==1)),
      carLuxury = NROW(which(car.make.luxury.flag==1)),
```

```
      carNeutCol = NROW(which(car.color %in% c("BK","GY","WT"))),
      carOthCol = NROW(which(!(car.color %in% c("BK","GY","WT")))))

head(p.summary2)
summary(p.summary)

p.summary2 <- p.summary2[which((p.summary2$wkDay2+p.summary2$wkDay3+p.summary2$wkDay4+p.summary2$wkDay5+p.summary2$wkDay6)>0),]

mean <- mean(p.summary2$fineCnt)

p.summary2 <- data.frame(issueCalDay = p.summary2[,1], fineAavg = ifelse(p.summary2$fineCnt >= mean,1,0), p.summary2[,-1], stringsAsFactors = FALSE)

remove(mean)

p.summary2$time3p <- p.summary2$time3/p.summary2$fineCnt
p.summary2$time4p <- p.summary2$time4/p.summary2$fineCnt
p.summary2$wkDay2p <- p.summary2$wkDay2/p.summary2$fineCnt
p.summary2$wkDay3p <- p.summary2$wkDay3/p.summary2$fineCnt
p.summary2$wkDay4p <- p.summary2$wkDay4/p.summary2$fineCnt
p.summary2$wkDay5p <- p.summary2$wkDay5/p.summary2$fineCnt
p.summary2$wkDay6p <- p.summary2$wkDay6/p.summary2$fineCnt
p.summary2$holidayp <- p.summary2$holiday/p.summary2$fineCnt
p.summary2$cityCnt12p <- p.summary2$cityCnt12/p.summary2$fineCnt
p.summary2$cityCnt13p <- p.summary2$cityCnt13/p.summary2$fineCnt
p.summary2$cityCnt14p <- p.summary2$cityCnt14/p.summary2$fineCnt
p.summary2$cityCnt21p <- p.summary2$cityCnt21/p.summary2$fineCnt
p.summary2$cityCnt22p <- p.summary2$cityCnt22/p.summary2$fineCnt
p.summary2$cityCnt23p <- p.summary2$cityCnt23/p.summary2$fineCnt
p.summary2$cityCnt24p <- p.summary2$cityCnt24/p.summary2$fineCnt
p.summary2$cityCnt34p <- p.summary2$cityCnt34/p.summary2$fineCnt
p.summary2$cityCntVn1p <- p.summary2$cityCntVn1/p.summary2$fineCnt
p.summary2$cityCntVn2p <- p.summary2$cityCntVn2/p.summary2$fineCnt
p.summary2$cityCntVn3p <- p.summary2$cityCntVn3/p.summary2$fineCnt
p.summary2$cityCntHw1p <- p.summary2$cityCntHw1/p.summary2$fineCnt
p.summary2$cityCntHw2p <- p.summary2$cityCntHw2/p.summary2$fineCnt
p.summary2$cityCntHw3p <- p.summary2$cityCntHw3/p.summary2$fineCnt
p.summary2$cityCntHw4p <- p.summary2$cityCntHw4/p.summary2$fineCnt
p.summary2$cityCntSp1p <- p.summary2$cityCntSp1/p.summary2$fineCnt
p.summary2$cityCntSp2p <- p.summary2$cityCntSp2/p.summary2$fineCnt
p.summary2$cityCntSp3p <- p.summary2$cityCntSp3/p.summary2$fineCnt
p.summary2$cityCntSp4p <- p.summary2$cityCntSp4/p.summary2$fineCnt
p.summary2$cityCntSp5p <- p.summary2$cityCntSp5/p.summary2$fineCnt
p.summary2$agency51p <- p.summary2$agency51/p.summary2$fineCnt
p.summary2$agency53p <- p.summary2$agency53/p.summary2$fineCnt
p.summary2$agency54p <- p.summary2$agency54/p.summary2$fineCnt
p.summary2$agency55p <- p.summary2$agency55/p.summary2$fineCnt
p.summary2$agency56p <- p.summary2$agency56/p.summary2$fineCnt
p.summary2$plateExpp <- p.summary2$plateExp/p.summary2$fineCnt
p.summary2$plateCAp <- p.summary2$plateCA/p.summary2$fineCnt
p.summary2$plateNCAp <- p.summary2$plateNCA/p.summary2$fineCnt
p.summary2$carImportp<- p.summary2$carImport/p.summary2$fineCnt
p.summary2$carLuxuryp<- p.summary2$carLuxury/p.summary2$fineCnt
p.summary2$carNeutColp<- p.summary2$carNeutCol/p.summary2$fineCnt
p.summary2$carOthColp<- p.summary2$carOthCol/p.summary2$fineCnt

head(p.summary2)
str(p.summary2)
summary(p.summary[,57:67])

mean(p.model$fineAavg)

## GLM 1 -- issue week
# Create the weekly, numeric data set
p.model <- p.summary[,c(2,44:58,71:82)]

# Create the train and test data sets
p.model.rand <- sample(1:dim(p.model)[1])
p.model.cut <- round((nrow(p.model)/3)*2,digits=0)
p.model.train <- p.model[p.model.rand[1:p.model.cut],]
p.model.test <- p.model[p.model.rand[(p.model.cut+1):nrow(p.model)],]

# Train the GLM model
glm <- glm(fineAavg ~ ., data = p.model.train, family = "binomial")
summary(glm)
head(glm)

# Test the GLM Model
glm.p <- predict(glm, p.model.test)

glm.p.df <- data.frame(p.model.test[,1],ifelse(glm.p <= 0, 0, 1))
glm.p.t <- table(glm.p.df)
glm.p.t
(glm.p.t[1,1]+glm.p.t[2,2])/sum(glm.p.t)

## GLM 2 -- issue week
p.model2 <- p.summary[,c(2,44:50,59:82)]

# Create the train and test data sets
p.model2.rand <- sample(1:dim(p.model2)[1])
p.model2.cut <- round((nrow(p.model2)/3)*2,digits=0)
p.model2.train <- p.model2[p.model2.rand[1:p.model2.cut],]
p.model2.test <- p.model2[p.model2.rand[(p.model2.cut+1):nrow(p.model2)],]

# Train the GLM model
glm.2 <- glm(fineAavg ~ ., data = p.model2.train, family = "binomial")
summary(glm.2)
```

```
head(glm.2)

glm.p2 <- predict(glm.2, p.model2.test)

glm.p.df2 <- data.frame(p.model2[,1],ifelse(glm.p2 <= 0, 0, 1))
glm.p.t2 <- table(glm.p.df2)
glm.p.t2
(glm.p.t2[1,1]+glm.p.t2[2,2])/sum(glm.p.t2)

## GLM 3 -- issue calendar day
p.model3 <- p.summary2[,c(2,44:50,59:82)]

# Create the train and test data sets
p.model3.rand <- sample(1:dim(p.model3)[1])
p.model3.cut <- round((NROW(p.model3)/3)*2,digits=0)
p.model3.train <- p.model3[p.model3.rand[1:p.model3.cut],]
p.model3.test <- p.model3[p.model3.rand[(p.model3.cut+1):nrow(p.model3)],]

# Train the GLM model
glm.3 <- glm(fineAavg ~ ., data = p.model3, family = "binomial")
summary(glm.3)
head(glm.3)

glm.p3 <- predict(glm.3,p.model3)

glm.p.df3 <- data.frame(p.model3[,1],ifelse(glm.p3 <= 0, 0, 1))
glm.p.t3 <- table(glm.p.df3)
glm.p.t3
(glm.p.t3[1,1]+glm.p.t3[2,2])/sum(glm.p.t3)

# Attempt Random Forest w/ actual count
p.model4 <- p.summary2[,c(3,45:52,61:84)]

# Fit the rF model
rf4 <- randomForest(fineCnt ~ ., p.model4, ntree = 500)
rf4$importance
summary(rf4)
head(rf4)
str(rf4)

p.model4$fineCntP <- round(rf4$predicted,0)
p.model4$P.Residual <- p.model4$fineCnt-p.model4$fineCntP
p.model4$P.ResPcnt <- (p.model4$fineCnt/p.model4$fineCntP)-1

mean(p.model4$P.Residual)
mean(p.model4$P.ResPcnt)
p.model4$P.ResFlag <-
  ifelse(p.model4$P.Residual > 0,1,
       ifelse(p.model4$P.Residual < 0,-1,0))

head(p.model4)

# Create a function to calculate rmse
rmse <- function(error){
  sqrt(mean(error^2))
}

# Continue with RMSE step
rmse(p.model4$P.Residual)
mean(p.model4$fineCnt)
sd(p.model4$fineCnt)

## Remider the average daily ticket amount = 2048.873 and the sd = 563.1074
## Prior to addition of the city holidays flag, RMSE = 283.0865, Residual Rate = -0.0410028
## After the addition of the city holidays flag, RMSE = 283.4215, Residual Rate = -0.0413982
## Run 2 with city holidays flagged, RMSE = 156.981, Residual Rate = -0.04168207



# City Holidays
holiday <- as.Date(c("2018-01-01","2018-01-21","2018-02-18","2018-03-25","2018-05-27","2018-07-04","2018-09-02","2018-10-14","2018-11-11","2018-11-28","2018-11-29","2018-
12-25"))
p$holiday.ind <- ifelse(p$issue.calday %in% holiday,1,0)

str(p)
sum(p$holiday.ind)
unique(p[which(p$holiday.ind==1),52])

# Build kmeans clustering for Lat/Long
library(factoextra)
library(cluster)
library(gridExtra)

str(p)
fit <- kmeans(p[,44:46],centers=4, nstart=25)
summary(fit)
p1 <- fviz_cluster(fit, geom = "point", data = p[,44:46]) + ggtitle("k = 4") + coord_cartesian(xlim = c(-0.56,-0.63))
p1
p$issue.cityCenter.clust <- fit$cluster

fit.cC.vn <- kmeans(p[which(p$issue.cityCenter.clust==1),14:15], centers=3, nstart=25)
summary(fit.cC.vn)
p2 <- fviz_cluster(fit.cC.vn, geom = "point", data = p[which(p$issue.cityCenter.clust==1),14:15]) + ggtitle("vn.k = 3")
p2

fit.cC.hw <- kmeans(p[which(p$issue.cityCenter.clust==3),14:15], centers=4, nstart=25)
```

```
summary(fit.cC.hw)
p3 <- fviz_cluster(fit.cC.hw, geom = "point", data = p[which(p$issue.cityCenter.clust==3),14:15]) + ggtitle("hw.k = 4")
p3

fit.cC.sp <- kmeans(p[which(p$issue.cityCenter.clust==2),14:15], centers=5, nstart=25)
summary(fit.cC.sp)
p4 <- fviz_cluster(fit.cC.sp, geom = "point", data = p[which(p$issue.cityCenter.clust==2),14:15]) + ggtitle("sp.k = 5")
p4

grid.arrange(p1, p2, p3, p4, nrow = 2)

p <- p[,-(48:51)]

p$issue.cityCenter.subclus.vn <-
  ifelse(p$issue.cityCenter.clust==1,fit.cC.vn$cluster,0)
tapply(p$X, p$issue.cityCenter.subclus.vn, NROW)

p$issue.cityCenter.subclus.hw <-
  ifelse(p$issue.cityCenter.clust==3,fit.cC.hw$cluster,0)
tapply(p$X, p$issue.cityCenter.subclus.hw, NROW)

p$issue.cityCenter.subclus.sp <-
  ifelse(p$issue.cityCenter.clust==2,fit.cC.sp$cluster,0)
tapply(p$X, p$issue.cityCenter.subclus.sp, NROW)

p$issue.cityCenter.subclust <- apply(p[,48:50],1,max)
tapply(p$X,list(p$issue.cityCenter.clust,p$issue.cityCenter.subclust),NROW)
```