Report Code

This report contains all the code and visualizations from the original report.!1

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")

df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
library("reshape2")

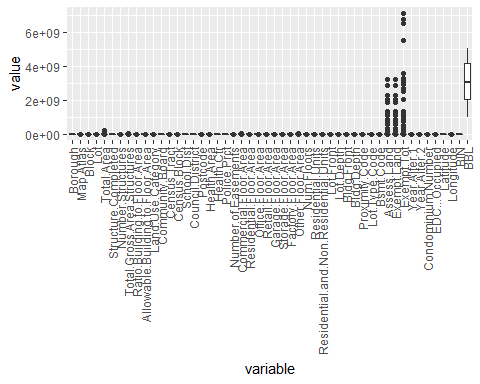
## Warning: package 'reshape2' was built under R version 4.1.2

library(ggplot2)  
  
data\_long <- melt(df) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 351291 rows containing non-finite values (stat\_boxplot).

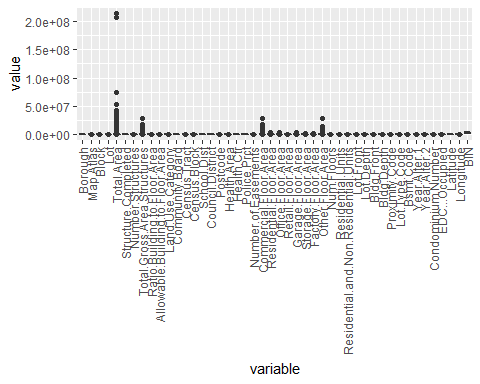


col1 <- c('Assess.Land', 'Exempt.Land', 'Exempt.Tot', 'BBL')  
df1 <- df[!(names(df)) %in% col1]  
  
#removing columns with large numbers   
data\_long <- melt(df1) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

#head(data\_long)   
ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 337684 rows containing non-finite values (stat\_boxplot).

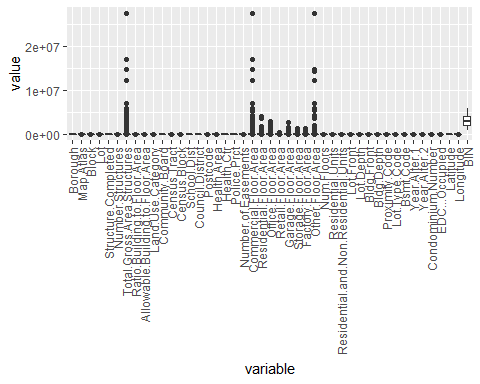


col2 <- c('Total.Area')  
df2 <- df1[!(names(df1)) %in% col2]  
  
#removing columns with large numbers   
data\_long <- melt(df2) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

#head(data\_long)   
ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 337684 rows containing non-finite values (stat\_boxplot).

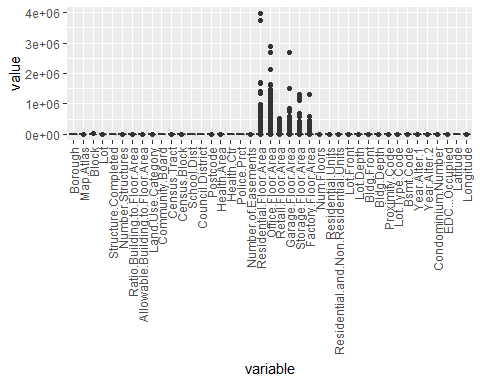


col3 <- c('Total.Gross.Area.Structures', 'Commercial.Floor.Area', 'Other.Floor.Area', 'BIN')  
df3 <- df2[!(names(df2)) %in% col3]  
  
#removing columns with large numbers   
data\_long <- melt(df3) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 303486 rows containing non-finite values (stat\_boxplot).

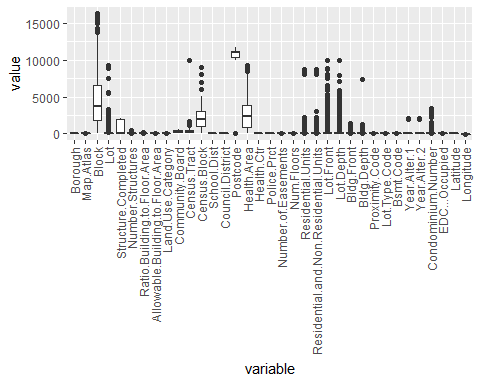


col4 <- c('Residential.Floor.Area', 'Office.Floor.Area', 'Retail.Floor.Area', 'Garage.Floor.Area', 'Storage.Floor.Area', 'Factory.Floor.Area')  
df4 <- df3[!(names(df3)) %in% col4]  
  
#removing columns with large numbers   
data\_long <- melt(df4) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

#head(data\_long)   
ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 303486 rows containing non-finite values (stat\_boxplot).

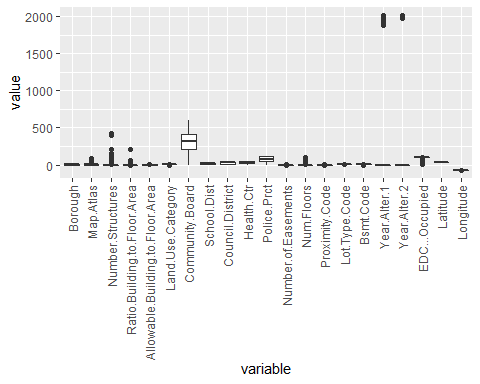


col5 <- c('Block', 'Lot', 'Structure.Completed', 'Census.Block', 'Census.Tract', 'Postcode', 'Health.Area', 'Residential.Units', 'Residential.and.Non.Residential.Units', 'Lot.Front', 'Lot.Depth', 'Bldg.Front', 'Bldg.Depth', 'Condominium.Number')  
df5 <- df4[!(names(df4)) %in% col5]  
  
#removing columns with large numbers   
data\_long <- melt(df5) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

#head(data\_long)   
ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 229579 rows containing non-finite values (stat\_boxplot).

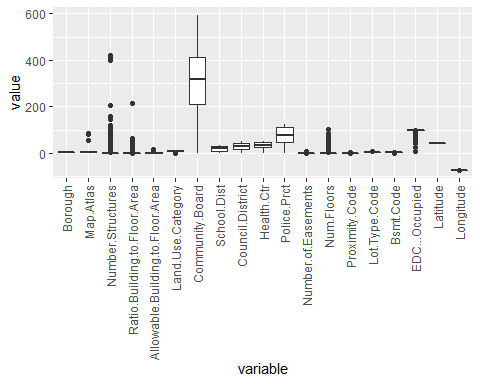


col6 <- c('Year.Alter.1', 'Year.Alter.2')  
df6 <- df5[!(names(df5)) %in% col6]  
  
#removing columns with large numbers   
data\_long <- melt(df6) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

#head(data\_long)   
ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 229539 rows containing non-finite values (stat\_boxplot).

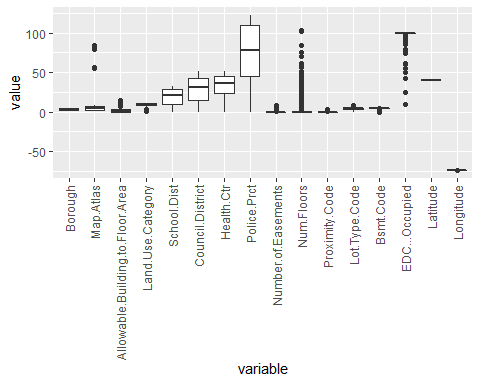


col7 <- c('Number.Structures', 'Ratio.Building.to.Floor.Area', 'Community.Board')  
df7 <- df6[!(names(df6)) %in% col7]  
  
#removing columns with large numbers   
data\_long <- melt(df7) # Reshaping data frame

## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

#head(data\_long)   
ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 229539 rows containing non-finite values (stat\_boxplot).

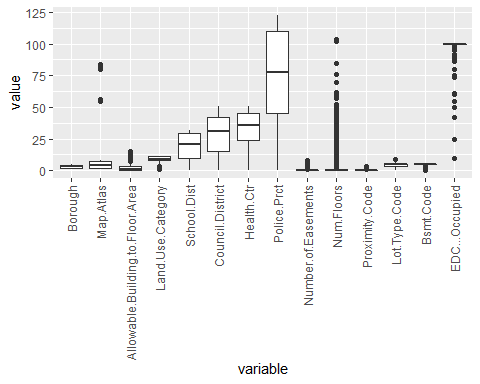


col8 <- c('Longitude', 'Latitude')  
df8 <- df7[!(names(df7)) %in% col8]  
  
#removing columns with large numbers   
data\_long <- melt(df8) # Reshaping data frame

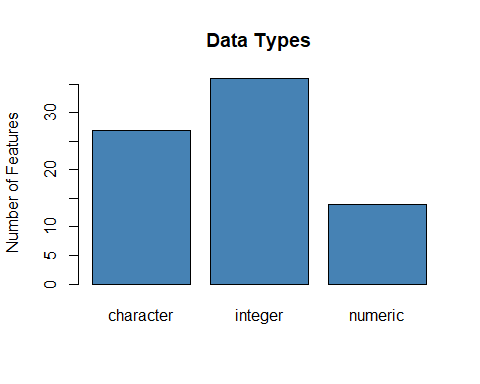
## Using Date.Created, Address, Street.Number, Street.Name, Parcel.Name, Agency, Current.Uses, Open.Petroleum.Spill, Govt.Clean.Up.Program, Fire.Comp, Major.Use, Irr.Lot.Code, His.Dist, Landmark, Coordinates, E.Designation.Number, Industrial.Business.Zone, Zone.Dist.1, Zone.Dist.2, Overlay.1, Overlay.2, SP.Dist.1, SP.Dist.2, Potential.Urban.Ag, Contact, Pluto.Version, NTA as id variables

ggplot(data\_long, aes(x = variable, y = value)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 164621 rows containing non-finite values (stat\_boxplot).



df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
data\_types <- function(frame) {  
 res <- lapply(frame, class)  
 res\_frame <- data.frame(unlist(res))  
 barplot(table(res\_frame), main="Data Types", col="steelblue", ylab="Number of Features")  
}  
  
data\_types(df)

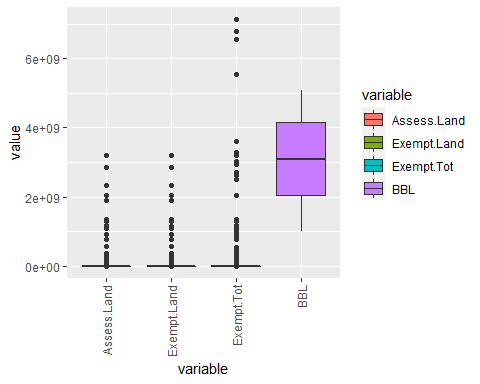


df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
col = c("Assess.Land", "Exempt.Land", "Exempt.Tot", "BBL")  
df1 <- df[,col]  
  
data\_long <- melt(df1) # Reshaping data frame

## No id variables; using all as measure variables

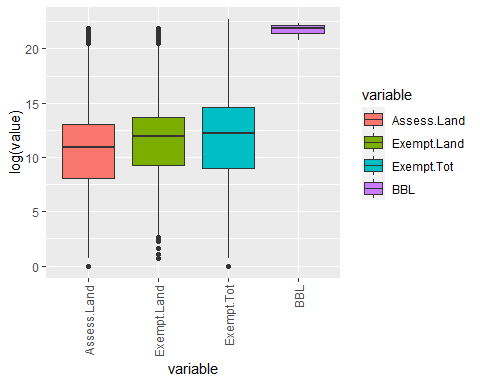
ggplot(data\_long, aes(x = variable, y = value, fill = variable)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 13607 rows containing non-finite values (stat\_boxplot).



ggplot(data\_long, aes(x = variable, y = log(value), fill = variable)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 66328 rows containing non-finite values (stat\_boxplot).

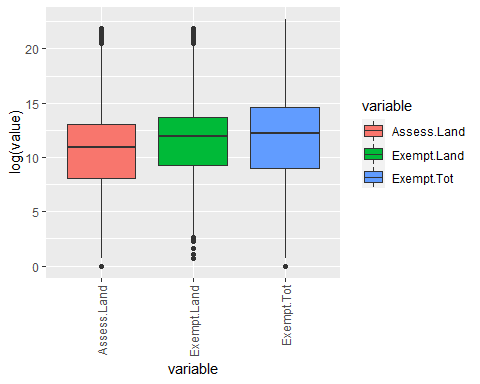


col = c("Assess.Land", "Exempt.Land", "Exempt.Tot")  
df1 <- df[,col]  
  
data\_long <- melt(df1) # Reshaping data frame

## No id variables; using all as measure variables

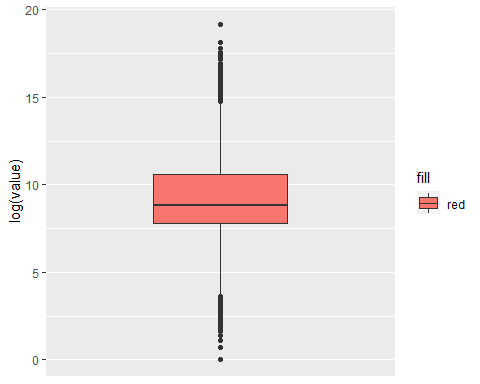
ggplot(data\_long, aes(x = variable, y = log(value), fill = variable)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 66309 rows containing non-finite values (stat\_boxplot).



col = c("Total.Area")  
df1 <- df[,col]  
  
data\_long <- melt(df1) # Reshaping data frame  
   
ggplot(data\_long, aes(y =log(value), fill = 'red')) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 1645 rows containing non-finite values (stat\_boxplot).

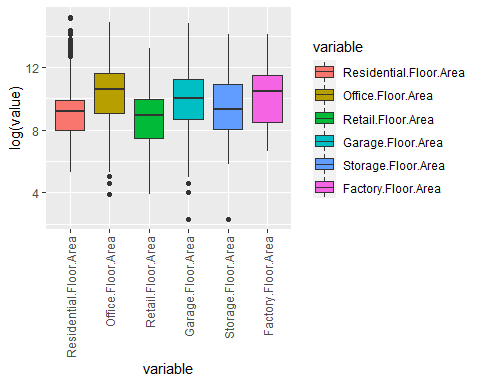


col <- c('Residential.Floor.Area', 'Office.Floor.Area', 'Retail.Floor.Area', 'Garage.Floor.Area', 'Storage.Floor.Area', 'Factory.Floor.Area')  
  
df1 <- df[,col]  
  
data\_long <- melt(df1) # Reshaping data frame

## No id variables; using all as measure variables

ggplot(data\_long, aes(x = variable, y = log(value), fill = variable)) + # Applying ggplot function  
 geom\_boxplot() +  
 scale\_x\_discrete(guide = guide\_axis(angle = 90))

## Warning: Removed 405077 rows containing non-finite values (stat\_boxplot).



#install.packages(naivebayes)  
#install.packages('tidyverse')  
#library(tidyverse)  
#install.packages('ggplot2')  
library(ggplot2)  
#install.packages('caret')  
library(caret)

## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: lattice

#install.packages('caretEnsemble')  
library(caretEnsemble)

## Warning: package 'caretEnsemble' was built under R version 4.1.3

##   
## Attaching package: 'caretEnsemble'

## The following object is masked from 'package:ggplot2':  
##   
## autoplot

#install.packages('psych')  
library(psych)

## Warning: package 'psych' was built under R version 4.1.3

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

#install.packages('Amelia')  
library(Amelia)

## Warning: package 'Amelia' was built under R version 4.1.3

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.8.0, built: 2021-05-26)  
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

#install.packages('mice')  
library(mice)

## Warning: package 'mice' was built under R version 4.1.3

##   
## Attaching package: 'mice'

## The following object is masked from 'package:stats':  
##   
## filter

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

#install.packages('GGally')  
library(GGally)

## Warning: package 'GGally' was built under R version 4.1.3

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

#install.packages('rpart')  
library(rpart)  
#install.packages('randomForest')  
library(randomForest)

## Warning: package 'randomForest' was built under R version 4.1.3

## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':  
##   
## outlier

## The following object is masked from 'package:ggplot2':  
##   
## margin

## The following object is masked from 'package:dplyr':  
##   
## combine

library(naivebayes)

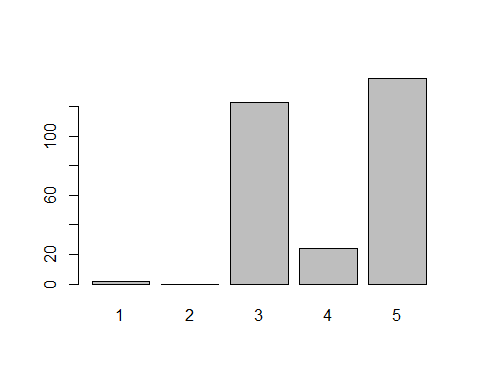
## Warning: package 'naivebayes' was built under R version 4.1.3

## naivebayes 0.9.7 loaded

df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
df$Borough <- factor(df$Borough)  
locmodel <- naive\_bayes(Borough ~ Assess.Land, data = df)  
  
# Predict  
pr <- predict(locmodel, df[which(df$Num.Floors > 20),])

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

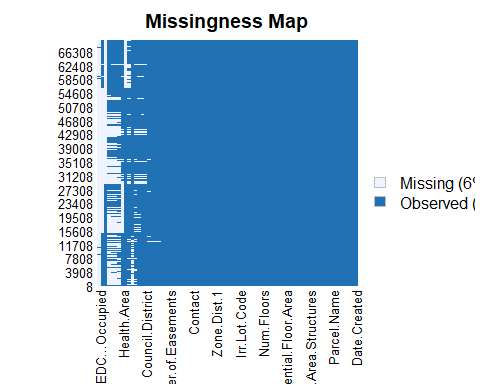
plot(pr)



# Predict  
df$pr <- predict(locmodel, df)

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

#describe(df)  
missmap(df)



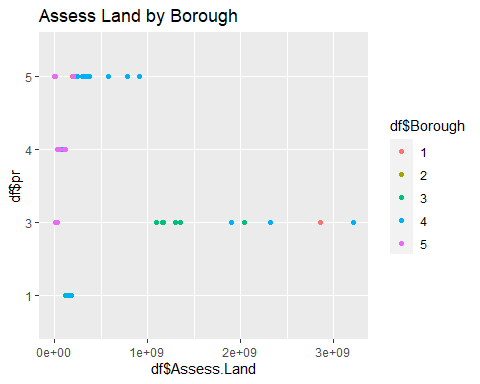
ggplot(df, aes(df$Assess.Land, df$pr, colour = df$Borough)) +  
 geom\_point() +   
 labs(title="Assess Land by Borough")

## Warning: Use of `df$Assess.Land` is discouraged. Use `Assess.Land` instead.

## Warning: Use of `df$pr` is discouraged. Use `pr` instead.

## Warning: Use of `df$Borough` is discouraged. Use `Borough` instead.

## Warning: Removed 1 rows containing missing values (geom\_point).

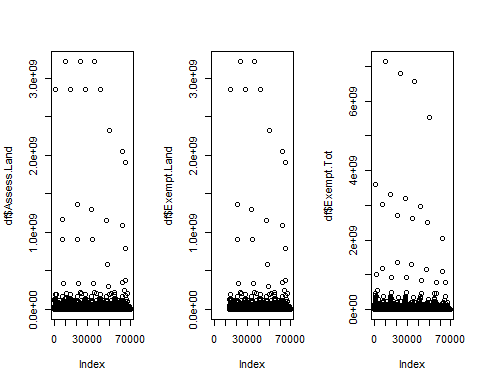


library(dplyr)  
library(ggplot2)  
library(viridis)

## Warning: package 'viridis' was built under R version 4.1.2

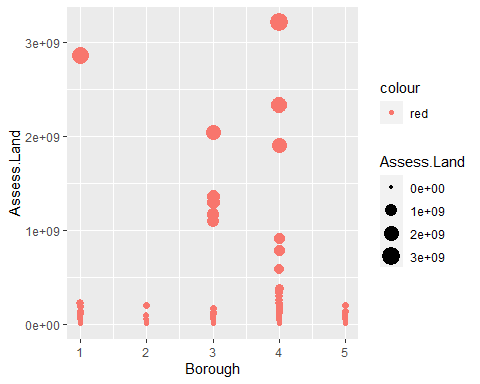
## Loading required package: viridisLite

df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
par(mfrow = c(1, 3))   
plot(df$Assess.Land)  
plot(df$Exempt.Land)  
plot(df$Exempt.Tot)



ggplot(df, aes(Borough, Assess.Land, size = Assess.Land, col = "red")) +  
 geom\_point()

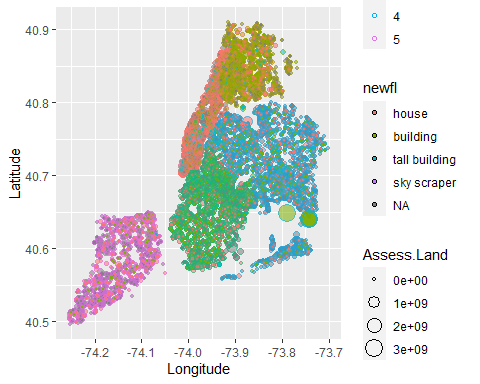
## Warning: Removed 1 rows containing missing values (geom\_point).



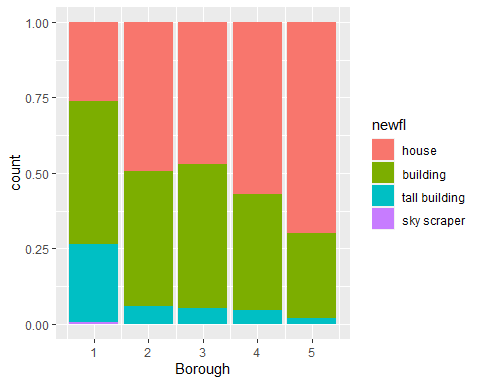
df$newfl <- cut(df$Num.Floors,  
 breaks=c(0, 2, 5, 50, 500),  
 labels=c('house', 'building', 'tall building', 'sky scraper'))   
#na.exclude(df$newfl)   
  
df %>%  
 group\_by(Borough) %>%  
 group\_by(Num.Floors) %>%  
 ggplot(aes(Longitude, Latitude, size = Assess.Land)) +  
 geom\_point(shape = 21, aes(alpha = 0.25, color = factor(Borough), fill = newfl)) +  
 scale\_color\_brewer(palette = "Dark2") +  
 scale\_colour\_discrete(na.translate = F)

## Scale for 'colour' is already present. Adding another scale for 'colour',  
## which will replace the existing scale.

## Warning: Removed 32460 rows containing missing values (geom\_point).



filter(df, !is.na(newfl)) %>%   
 ggplot() +  
 geom\_bar(mapping = aes(x = Borough, fill = newfl), position = "fill")



library(ggplot2)  
library(plyr)

## Warning: package 'plyr' was built under R version 4.1.2

## ------------------------------------------------------------------------------

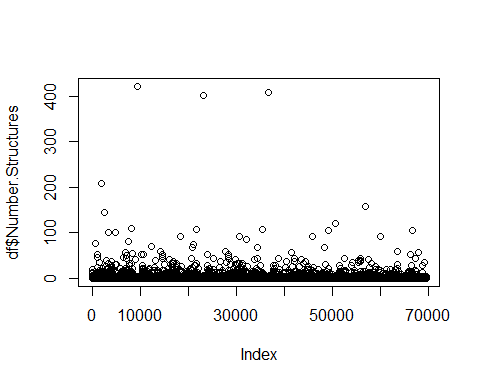
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

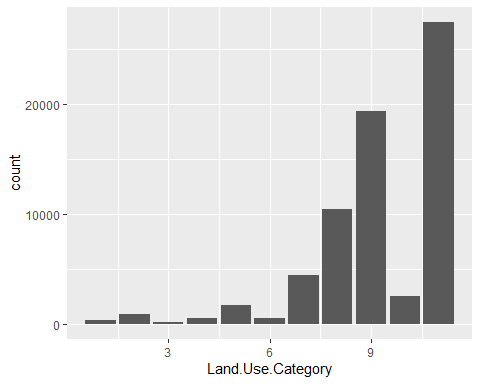
## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

library(dplyr)  
  
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
plot(df$Number.Structures)

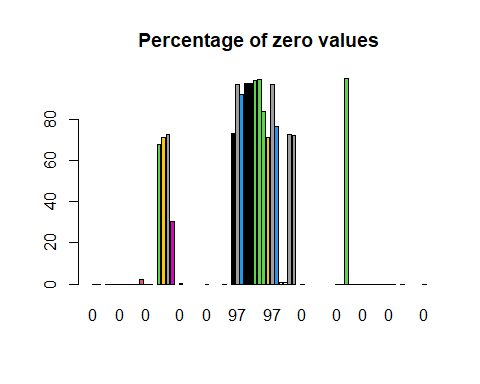


ggplot(df, aes(Land.Use.Category)) +  
 geom\_bar()

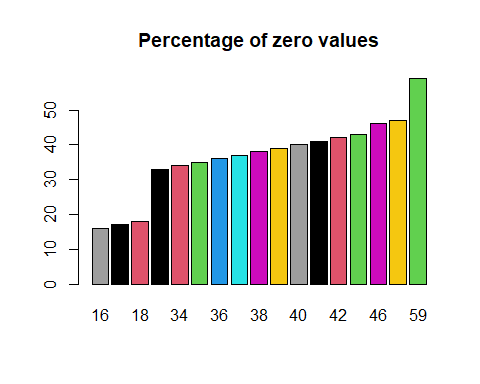
## Warning: Removed 1020 rows containing non-finite values (stat\_count).



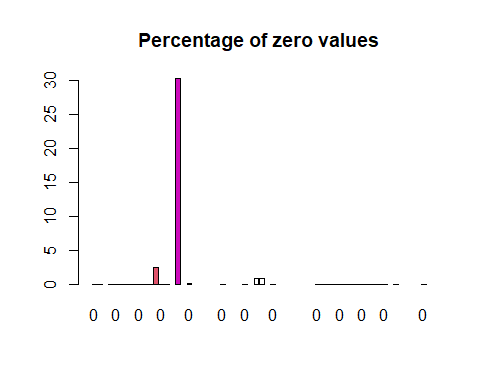
zerosPercentage <- colSums(df==0)/nrow(df)\*100   
  
barplot(zerosPercentage, main = "Percentage of zero values", col = zerosPercentage, names.arg = round(zerosPercentage, 0))



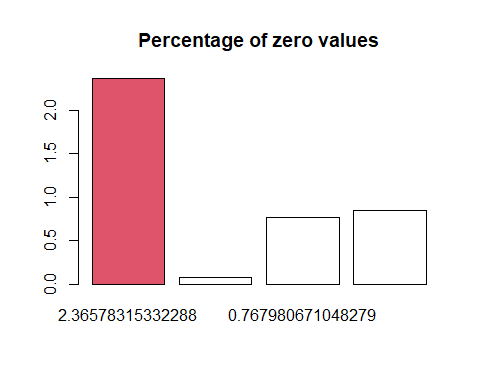
greaterThan50 <- which((colSums(df==0)/nrow(df)\*100) > 50)  
  
barplot(greaterThan50, main = "Percentage of zero values", col = greaterThan50, names.arg = round(greaterThan50, 0))



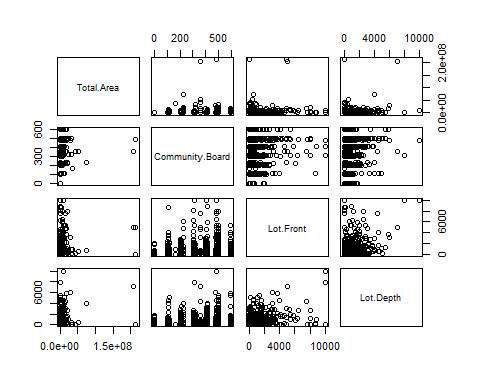
unwantedColumns <- names(which((colSums(df==0)/nrow(df)\*100) > 50))  
  
cleanedDf = df[!(names(df)) %in% unwantedColumns]  
  
zerosPercentage <- colSums(cleanedDf==0)/nrow(cleanedDf)\*100   
barplot(zerosPercentage, main = "Percentage of zero values", col = zerosPercentage, names.arg = round(zerosPercentage, 0))



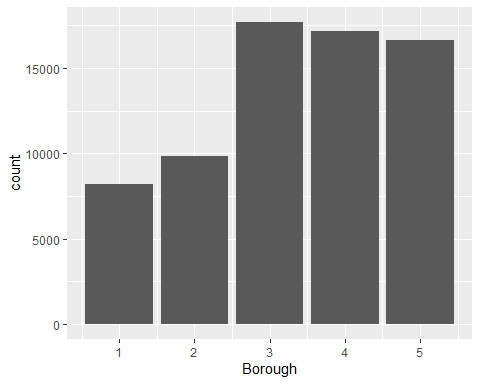
#names(which((colSums(cleanedDf==0)/nrow(cleanedDf)\*100) > 0))  
  
col <- c("Total.Area", "Community.Board", "Lot.Front", "Lot.Depth")  
  
cleanedDf = cleanedDf[!(names(cleanedDf)) %in% c("Allowable.Building.to.Floor.Area")]  
  
barplot(zerosPercentage[col], main = "Percentage of zero values", col = zerosPercentage[col], names.arg = zerosPercentage[col])



plot(cleanedDf[col])

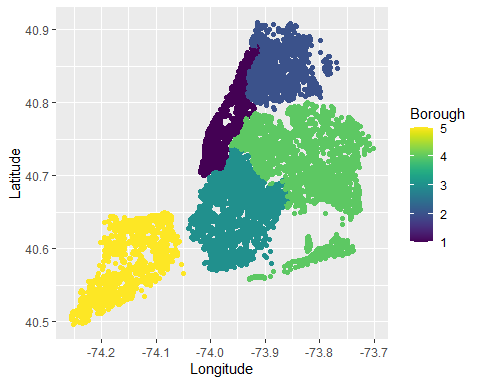


library(dplyr)  
library(ggplot2)  
library(viridis)  
  
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
df %>%  
 group\_by(Borough) %>%  
 ggplot(aes(Borough)) +  
 geom\_bar()

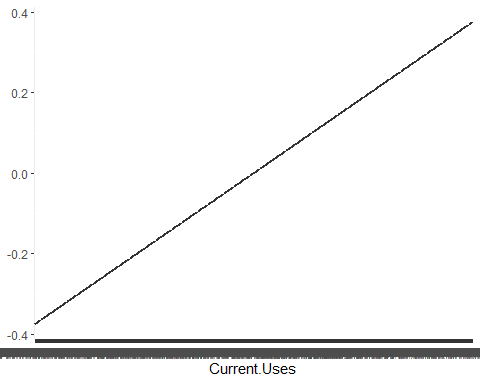


clr = c('yellow', 'red', 'blue', 'brown', 'green')  
  
df %>%  
 group\_by(Borough) %>%  
 ggplot(aes(Longitude, Latitude, col = Borough, )) +  
 geom\_point() +  
 scale\_color\_viridis(option = "D")

## Warning: Removed 32459 rows containing missing values (geom\_point).

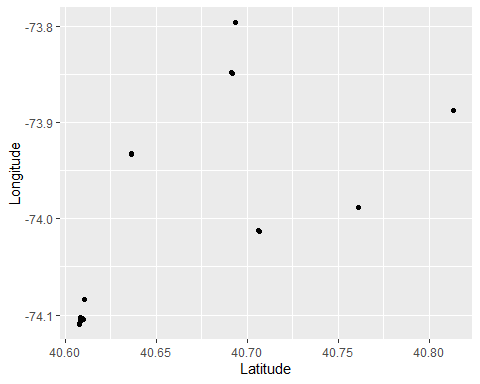


#colors()  
df %>%  
 group\_by(Borough) %>%  
 filter(Borough == 1) %>%  
 group\_by(Current.Uses) %>%  
 ggplot(aes(Current.Uses)) +   
 geom\_boxplot()



#unique(df$Current.Uses)  
 df <- df[which(df$Current.Uses == "PUBLIC SCHOOL"),]  
   
 ggplot(df, aes(Latitude, Longitude)) +  
 geom\_point()

## Warning: Removed 4 rows containing missing values (geom\_point).



library(maps)

## Warning: package 'maps' was built under R version 4.1.2

##   
## Attaching package: 'maps'

## The following object is masked from 'package:plyr':  
##   
## ozone

## The following object is masked from 'package:viridis':  
##   
## unemp

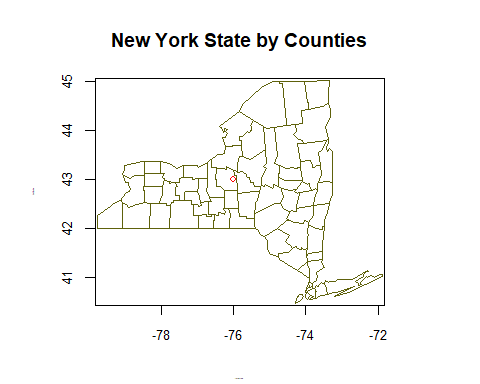
library(ggplot2)  
  
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
maps::map('county', region = 'new york', city = "new york", col = "#5E610B")

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "city" is not a graphical  
## parameter

#map.cities(us.cities, country="NY", col = "#642EFE", cex = 0.6) # map cities recorded in us.cities to NY State  
map.axes(cex.axis=0.8)   
ylim(40, 41)

## <ScaleContinuousPosition>  
## Range:   
## Limits: 40 -- 41

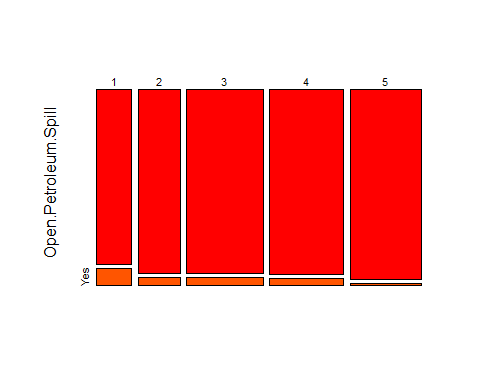
title(main = "New York State by Counties", xlab = "Longitude", ylab = "Latitude",  
 cex.lab = 0.1)   
points(-76, 43,   
 col = "red", cex = .8)



install.packages("naivebayes")

## Warning: package 'naivebayes' is in use and will not be installed

library(naivebayes)  
  
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
# Build the location prediction model  
df$Borough <- as.factor(df$Borough)  
  
model <- naive\_bayes(Borough ~ Open.Petroleum.Spill, data = df)  
  
# Predict Thursday's 9am location  
plot(model) #model 4 has more oil spills



#model1  
  
# Predict Saturdays's 9am location  
#predict(locmodel, as.data.frame(df$Major\_Use))

library(dplyr)  
library(ggplot2)  
library(viridis)  
library(WVPlots)

## Warning: package 'WVPlots' was built under R version 4.1.3

## Loading required package: wrapr

## Warning: package 'wrapr' was built under R version 4.1.3

##   
## Attaching package: 'wrapr'

## The following object is masked from 'package:dplyr':  
##   
## coalesce

#install.packages(naivebayes)  
#install.packages('tidyverse')  
#library(tidyverse)  
#install.packages('ggplot2')  
library(ggplot2)  
#install.packages('caret')  
library(caret)  
#install.packages('caretEnsemble')  
library(caretEnsemble)  
#install.packages('psych')  
library(psych)  
#install.packages('Amelia')  
library(Amelia)  
#install.packages('mice')  
library(mice)  
#install.packages('GGally')  
library(GGally)  
#install.packages('rpart')  
library(rpart)  
#install.packages('randomForest')  
library(randomForest)  
library(naivebayes)  
  
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
cols <- c("Major.Use", "Potential.Urban.Ag", "Landmark", "Open.Petroleum.Spill", "Land.Use.Category", "Govt.Clean.Up.Program", "Current.Uses", "Borough")  
data <- df[,cols]  
  
data[is.na(data) | data == ""] <- NA  
#data$Potential.Urban.Ag  
fmla <- Potential.Urban.Ag ~ .  
  
  
model <- naive\_bayes(fmla, data = data, laplace = 1)

## Warning: naive\_bayes(): y contains NAs. They are excluded from the estimation  
## process.

data$pred <- predict(model, data, type = "prob")

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

plot(predict(model, data, type = "prob"), col = unique(data$Land.Use.Category), pch = 0, size = unique(data$Land.Use.Category))

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

## Warning in plot.window(...): "size" is not a graphical parameter

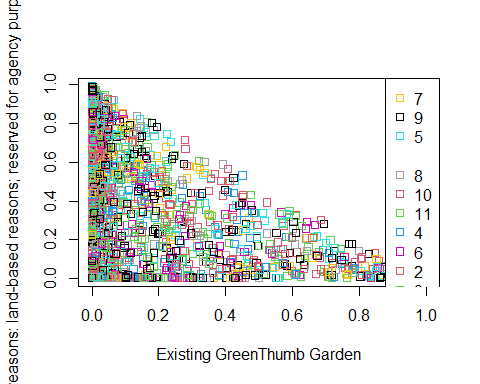
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter

## Warning in box(...): "size" is not a graphical parameter

## Warning in title(...): "size" is not a graphical parameter

legend("topright", legend = unique(data$Land.Use.Category),  
 col = unique(data$Land.Use.Category),  
 pch = 0)



plot(predict(model, data[which(data$Borough == 1),], type = "prob"), col = unique(data$Land.Use.Category), pch = 0, size = unique(data$Land.Use.Category))

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

## Warning in plot.window(...): "size" is not a graphical parameter

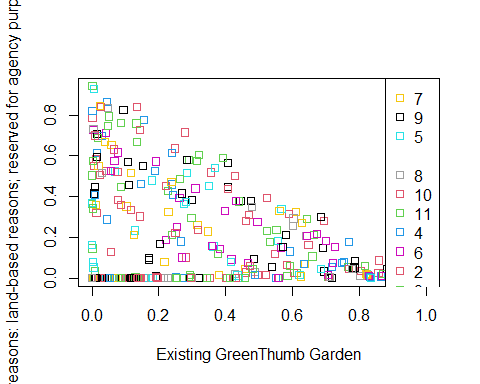
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter

## Warning in box(...): "size" is not a graphical parameter

## Warning in title(...): "size" is not a graphical parameter

legend("topright", legend = unique(data$Land.Use.Category),  
 col = unique(data$Land.Use.Category),  
 pch = 0)



plot(predict(model, data[which(data$Borough == 2),], type = "prob"), col = unique(data$Land.Use.Category), pch = 0, size = unique(data$Land.Use.Category))

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

## Warning in plot.window(...): "size" is not a graphical parameter

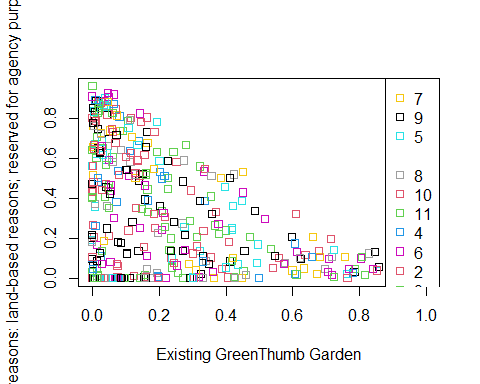
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter

## Warning in box(...): "size" is not a graphical parameter

## Warning in title(...): "size" is not a graphical parameter

legend("topright", legend = unique(data$Land.Use.Category),  
 col = unique(data$Land.Use.Category),  
 pch = 0)



plot(predict(model, data[which(data$Borough == 3),], type = "prob"), col = unique(data$Land.Use.Category), pch = 0, size = unique(data$Land.Use.Category))

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

## Warning in plot.window(...): "size" is not a graphical parameter

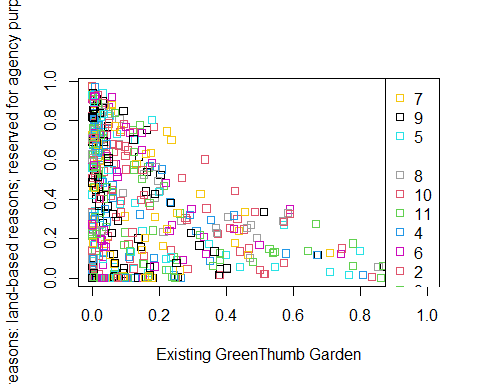
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter

## Warning in box(...): "size" is not a graphical parameter

## Warning in title(...): "size" is not a graphical parameter

legend("topright", legend = unique(data$Land.Use.Category),  
 col = unique(data$Land.Use.Category),  
 pch = 0)



plot(predict(model, data[which(data$Borough == 4),], type = "prob"), col = unique(data$Land.Use.Category), pch = 0, size = unique(data$Land.Use.Category))

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

## Warning in plot.window(...): "size" is not a graphical parameter

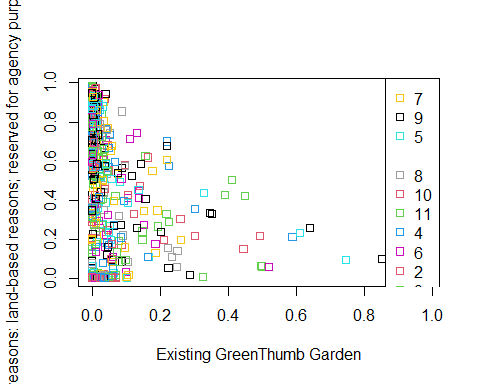
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter

## Warning in box(...): "size" is not a graphical parameter

## Warning in title(...): "size" is not a graphical parameter

legend("topright", legend = unique(data$Land.Use.Category),  
 col = unique(data$Land.Use.Category),  
 pch = 0)



plot(predict(model, data[which(data$Borough == 5),], type = "prob"), col = unique(data$Land.Use.Category), pch = 0, size = unique(data$Land.Use.Category))

## Warning: predict.naive\_bayes(): more features in the newdata are provided as  
## there are probability tables in the object. Calculation is performed based on  
## features to be found in the tables.

## Warning in plot.window(...): "size" is not a graphical parameter

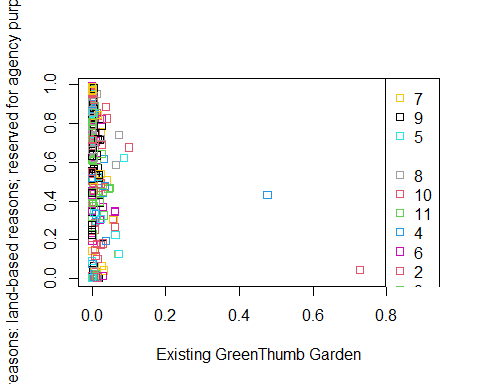
## Warning in plot.xy(xy, type, ...): "size" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "size" is not a  
## graphical parameter

## Warning in box(...): "size" is not a graphical parameter

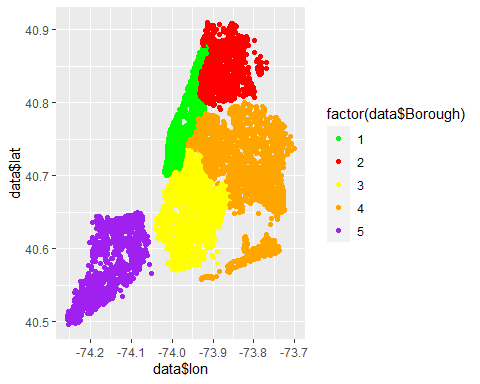
## Warning in title(...): "size" is not a graphical parameter

legend("topright", legend = unique(data$Land.Use.Category),  
 col = unique(data$Land.Use.Category),  
 pch = 0)



#unique(data$Borough)  
#unique(data$Land.Use.Category)  
  
data$lat <- df$Latitude  
data$lon <- df$Longitude  
  
ggplot(df, aes(x = data$lon, y = data$lat, color = factor(data$Borough))) +   
 geom\_point() +  
 scale\_color\_manual(values = c('green', 'red', 'yellow', 'orange', 'purple'))

## Warning: Removed 32459 rows containing missing values (geom\_point).



#logistic regression  
  
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
cols <- c("Assess.Land", "Exempt.Tot")  
df <- df[cols]  
  
df$Assess.Land <- as.numeric(df$Assess.Land)  
df$Exempt.Tot <- as.numeric(df$Exempt.Tot)  
  
  
normalize<-function(x) {  
 sapply(x, function(y) {if(all(is.na(y))) {y}  
 else if(sum(!is.na(y))!=1) {helper<- (y-min(y, na.rm=TRUE))/(max(y, na.rm=TRUE)-min(y, na.rm=TRUE))   
 helper}  
 else if(sum(!is.na(y))==1) {helper<-ifelse(is.na(y),0,1)  
 helper}  
 })  
}  
  
df <- as.data.frame(lapply(df, normalize))  
  
nrow(df)

## [1] 69533

fmla <- Potential.Urban.Ag ~ .  
  
model <- glm(Assess.Land ~ Exempt.Tot, data = df, family = "binomial")

## Warning: glm.fit: algorithm did not converge

df$pred <- predict(model, df)

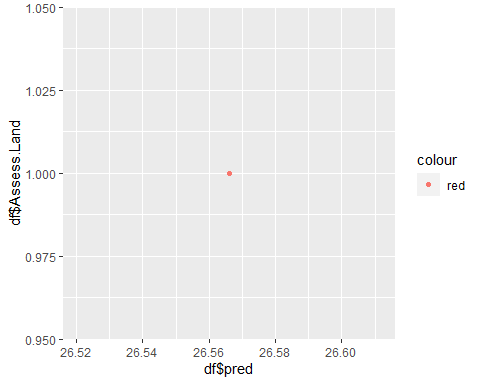
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from a rank-deficient fit may be misleading

ggplot(df, aes(x = df$pred, y = df$Assess.Land, color = "red")) +   
 geom\_point() +  
 geom\_abline(color = "blue")

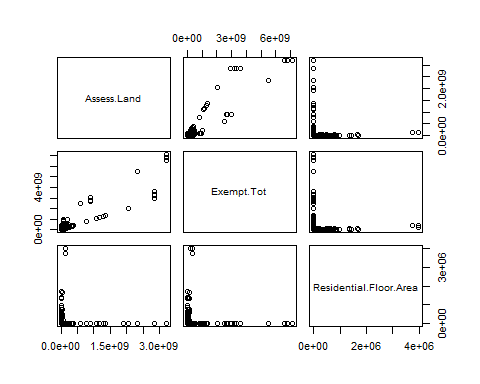
## Warning: Use of `df$pred` is discouraged. Use `pred` instead.

## Warning: Use of `df$Assess.Land` is discouraged. Use `Assess.Land` instead.

## Warning: Removed 1 rows containing missing values (geom\_point).



#install.packages("WVPlots")  
library(dplyr)  
library(ggplot2)  
library(viridis)  
library(WVPlots)  
#install.packages(naivebayes)  
#install.packages('tidyverse')  
#library(tidyverse)  
#install.packages('ggplot2')  
library(ggplot2)  
#install.packages('caret')  
library(caret)  
#install.packages('caretEnsemble')  
library(caretEnsemble)  
#install.packages('psych')  
library(psych)  
#install.packages('Amelia')  
library(Amelia)  
#install.packages('mice')  
library(mice)  
#install.packages('GGally')  
library(GGally)  
#install.packages('rpart')  
library(rpart)  
#install.packages('randomForest')  
library(randomForest)  
library(naivebayes)  
  
  
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
cols <- c("Assess.Land", "Exempt.Tot", "Residential.Floor.Area")  
data <- df[cols]  
  
plot(data)



nrow(data)

## [1] 69533

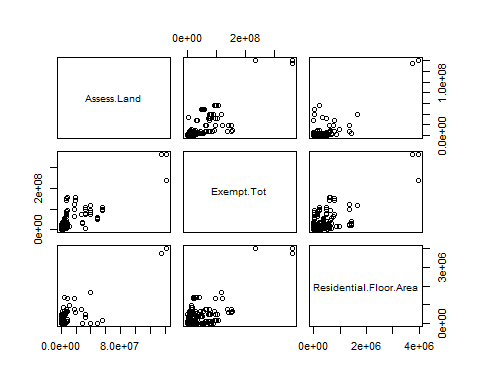
count(data[which(data == 0),])

## Assess.Land Exempt.Tot Residential.Floor.Area freq  
## 1 0 0 0 3351  
## 2 0 1892700 0 1  
## 3 NA NA NA 92504

data <- data[which(data$Residential.Floor.Area != 0),]  
data <- data[which(data$Assess.Land != 0),]  
data <- data[which(data$Exempt.Tot != 0),]  
  
  
nrow(data)

## [1] 1106

plot(data)



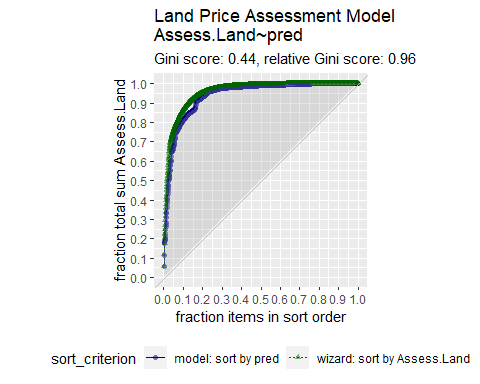
fmla <- Assess.Land ~ Exempt.Tot + Residential.Floor.Area  
fmla

## Assess.Land ~ Exempt.Tot + Residential.Floor.Area

model <- lm(fmla, data = data)  
  
#data <- data[-2,]  
nrow(data)

## [1] 1106

data$pred <- predict(model)  
  
#glance(model)  
  
GainCurvePlot(data, "pred", "Assess.Land", "Land Price Assessment Model")



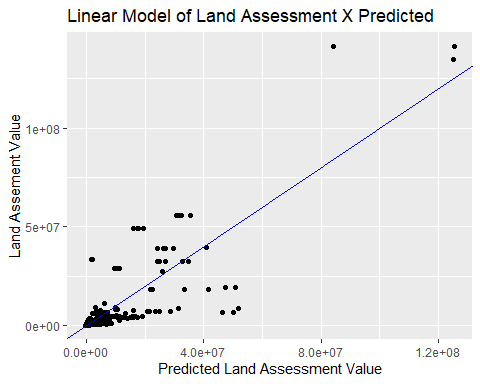
nrow(data)

## [1] 1106

ggplot(data, aes(x = data$pred, y = data$Assess.Land)) +   
 geom\_point() +  
 geom\_abline(color = "blue") +  
 xlab("Predicted Land Assessment Value") +  
 ylab("Land Assement Value") +  
 ggtitle("Linear Model of Land Assessment X Predicted")

## Warning: Use of `data$pred` is discouraged. Use `pred` instead.

## Warning: Use of `data$Assess.Land` is discouraged. Use `Assess.Land` instead.



ggplot(data, aes(x = data$Residential.Floor.Area, y = data$pred)) +   
 geom\_point() +  
 geom\_line(aes(data$Residential.Floor.Area, data$Assess.Land, color = 'Asses Land')) +  
 labs(color = 'Land Assess') +  
 geom\_line(aes(data$Residential.Floor.Area, data$Exempt.Tot, color = 'Total Exemtion')) +  
 xlab("Residential Floor Area") +  
 ylab("Price Assessment of Land") +  
 ggtitle("Linear Model of Land Assessment X Residentil Floor Area")

## Warning: Use of `data$Residential.Floor.Area` is discouraged. Use  
## `Residential.Floor.Area` instead.

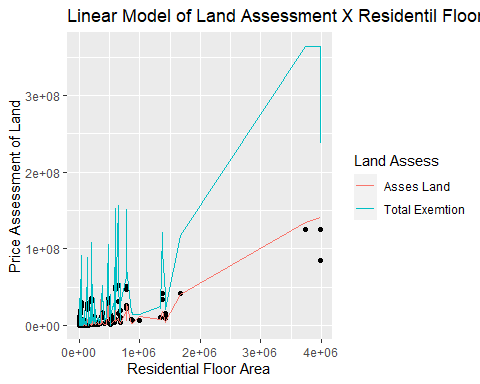
## Warning: Use of `data$pred` is discouraged. Use `pred` instead.

## Warning: Use of `data$Residential.Floor.Area` is discouraged. Use  
## `Residential.Floor.Area` instead.

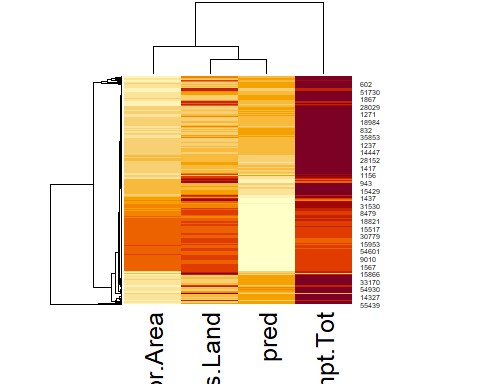
## Warning: Use of `data$Assess.Land` is discouraged. Use `Assess.Land` instead.

## Warning: Use of `data$Residential.Floor.Area` is discouraged. Use  
## `Residential.Floor.Area` instead.

## Warning: Use of `data$Exempt.Tot` is discouraged. Use `Exempt.Tot` instead.



heatmap(as.matrix(data))



#install.packages("WVPlots")  
library(dplyr)  
library(ggplot2)  
library(viridis)  
library(WVPlots)  
#install.packages('vtreat')  
library(vtreat)

## Warning: package 'vtreat' was built under R version 4.1.3

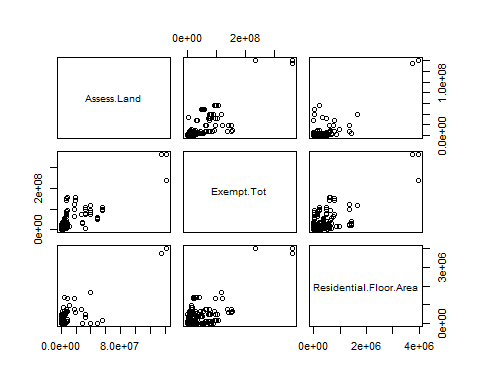
df <- read.csv("C:/Users/jeffb/OneDrive/Desktop/WSU-Spring-2022/Big Data Analitics/data-capstone-final-project/nysus.csv")  
  
cols <- c("Assess.Land", "Exempt.Tot", "Residential.Floor.Area")  
data <- df[cols]  
  
nrow(data)

## [1] 69533

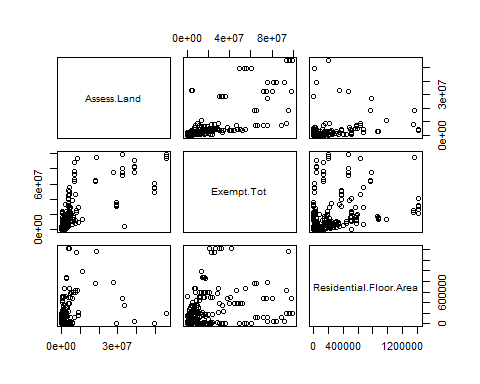
count(data[which(data == 0),])

## Assess.Land Exempt.Tot Residential.Floor.Area freq  
## 1 0 0 0 3351  
## 2 0 1892700 0 1  
## 3 NA NA NA 92504

data <- data[which(data$Residential.Floor.Area != 0),]  
data <- data[which(data$Assess.Land != 0),]  
data <- data[which(data$Exempt.Tot != 0),]  
  
plot(data)



data <- data[which(data$Residential.Floor.Area < 1e+08),]  
data <- data[which(data$Assess.Land < 1e+08),]  
data <- data[which(data$Exempt.Tot < 1e+08),]  
  
plot(data)



fmla <- Assess.Land ~ Exempt.Tot + Residential.Floor.Area  
  
model <- lm(fmla, data = data)  
  
#data <- data[-2,]  
nrow(data)

## [1] 1094

data$pred <- predict(model)  
#unique(data$Potential.Urban.Ag)  
nrow(data)

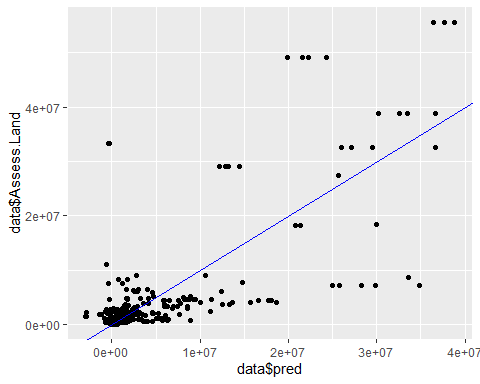
## [1] 1094

#glance(model)  
  
ggplot(data, aes(x = data$pred, y = data$Assess.Land, color = data$Potential.Urban.Ag)) +   
 geom\_point() +  
 geom\_abline(color = "blue")

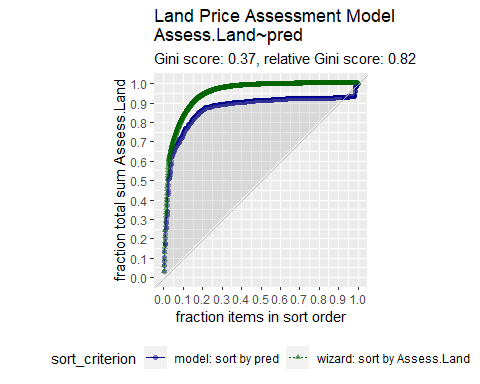
## Warning: Use of `data$pred` is discouraged. Use `pred` instead.

## Warning: Use of `data$Assess.Land` is discouraged. Use `Assess.Land` instead.

## Warning: Use of `data$Potential.Urban.Ag` is discouraged. Use  
## `Potential.Urban.Ag` instead.



GainCurvePlot(data, "pred", "Assess.Land", "Land Price Assessment Model")



ggplot(data, aes(x = data$Residential.Floor.Area, y = data$pred)) +   
 geom\_point() +  
 geom\_line(aes(data$Residential.Floor.Area, data$Assess.Land, color = 'Asses Land')) +  
 labs(color = 'Land Assess') +  
 geom\_line(aes(data$Residential.Floor.Area, data$Exempt.Tot, color = 'Total Exemtion')) +  
 xlab("Residential Floor Area") +  
 ylab("Price Assessment of Land") +  
 ggtitle("Linear Model of Land Assessment X Residentil Floor Area")

## Warning: Use of `data$Residential.Floor.Area` is discouraged. Use  
## `Residential.Floor.Area` instead.

## Warning: Use of `data$pred` is discouraged. Use `pred` instead.

## Warning: Use of `data$Residential.Floor.Area` is discouraged. Use  
## `Residential.Floor.Area` instead.

## Warning: Use of `data$Assess.Land` is discouraged. Use `Assess.Land` instead.

## Warning: Use of `data$Residential.Floor.Area` is discouraged. Use  
## `Residential.Floor.Area` instead.

## Warning: Use of `data$Exempt.Tot` is discouraged. Use `Exempt.Tot` instead.

