project

Erin Lyons, Yujian Tang, Norma Techarukpong, Ryan Thomas, Jason Yu

April 3, 2019

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:neuralnet':  
##   
## compute

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

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

library(MASS)

##   
## Attaching package: 'MASS'

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

library(ggplot2)  
library(purrr)  
library(leaps)  
library(glmnet)

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 3.5.3

## Loading required package: foreach

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loaded glmnet 2.0-16

library(neuralnet)  
library(tidyr)

## Warning: package 'tidyr' was built under R version 3.5.2

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:Matrix':  
##   
## expand

library(pls)

## Warning: package 'pls' was built under R version 3.5.3

##   
## Attaching package: 'pls'

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

library(devtools)

## Warning: package 'usethis' was built under R version 3.5.3

install\_github("vqv/ggbiplot")

## Skipping install of 'ggbiplot' from a github remote, the SHA1 (7325e880) has not changed since last install.  
## Use `force = TRUE` to force installation

library(ggbiplot)

## Loading required package: plyr

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

## 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 object is masked from 'package:purrr':  
##   
## compact

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

## Loading required package: scales

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## Loading required package: grid

# Data Cleaning

#Uploading Dataset  
CPS.data <- read.csv("Chicago\_Public\_Schools\_-\_Progress\_Report\_Cards\_\_2011-2012\_.csv")  
  
#Cleaning Data  
CPS.data$Link <- NULL  
CPS.data$Phone.Number <- NULL  
CPS.data$State <- NULL  
CPS.data$Street.Address <- NULL  
CPS.data$Location <- NULL  
CPS.data$City <- NULL  
CPS.data$RCDTS.Code <- NULL  
CPS.data$Latitude <- NULL  
CPS.data$Longitude <- NULL  
CPS.data$X\_COORDINATE <- NULL  
CPS.data$Y\_COORDINATE <- NULL

#Removing NDA values from variables and changing the values to NA  
for(i in names(CPS.data)) {  
 for(j in 1:nrow(CPS.data)) {  
 if(isTRUE(CPS.data[j,i] == "NDA")) { CPS.data[j, i] = NA}  
 }  
}  
  
#View  
head(CPS.data)

## School.ID  
## 1 610038  
## 2 610281  
## 3 610185  
## 4 609993  
## 5 610513  
## 6 610212  
## Name.of.School  
## 1 Abraham Lincoln Elementary School  
## 2 Adam Clayton Powell Paideia Community Academy Elementary School  
## 3 Adlai E Stevenson Elementary School  
## 4 Agustin Lara Elementary Academy  
## 5 Air Force Academy High School  
## 6 Albany Park Multicultural Academy  
## Elementary..Middle..or.High.School ZIP.Code  
## 1 ES 60614  
## 2 ES 60649  
## 3 ES 60652  
## 4 ES 60609  
## 5 HS 60609  
## 6 MS 60625  
## Network.Manager Collaborative.Name  
## 1 Fullerton Elementary Network NORTH-NORTHWEST SIDE COLLABORATIVE  
## 2 Skyway Elementary Network SOUTH SIDE COLLABORATIVE  
## 3 Midway Elementary Network SOUTHWEST SIDE COLLABORATIVE  
## 4 Pershing Elementary Network SOUTHWEST SIDE COLLABORATIVE  
## 5 Southwest Side High School Network SOUTHWEST SIDE COLLABORATIVE  
## 6 O'Hare Elementary Network NORTH-NORTHWEST SIDE COLLABORATIVE  
## Adequate.Yearly.Progress.Made. Track.Schedule  
## 1 No Standard  
## 2 No Track\_E  
## 3 No Standard  
## 4 No Track\_E  
## 5 <NA> Standard  
## 6 Yes Standard  
## CPS.Performance.Policy.Status CPS.Performance.Policy.Level  
## 1 Not on Probation Level 1  
## 2 Not on Probation Level 1  
## 3 Not on Probation Level 2  
## 4 Not on Probation Level 1  
## 5 Not on Probation Not Enough Data  
## 6 Not on Probation Level 1  
## Healthy.Schools.Certified. Safety.Icon Safety.Score  
## 1 Yes Very Strong 99  
## 2 No Average 54  
## 3 No Strong 61  
## 4 No Average 56  
## 5 Yes Average 49  
## 6 No Strong 66  
## Family.Involvement.Icon Family.Involvement.Score Environment.Icon  
## 1 Very Strong 99 Strong  
## 2 Strong 66 Strong  
## 3 <NA> <NA> Average  
## 4 Average 44 Average  
## 5 Strong 60 Strong  
## 6 Weak 37 Strong  
## Environment.Score Instruction.Icon Instruction.Score Leaders.Icon  
## 1 74 Strong 66 Strong  
## 2 74 Very Strong 84 Strong  
## 3 50 Weak 36 <NA>  
## 4 45 Weak 37 Strong  
## 5 60 Average 55 Average  
## 6 66 Strong 71 Average  
## Leaders.Score Teachers.Icon Teachers.Score Parent.Engagement.Icon  
## 1 65 Strong 70 Strong  
## 2 63 Strong 76 Weak  
## 3 <NA> <NA> <NA> Average  
## 4 65 Average 48 Average  
## 5 45 Average 54 Average  
## 6 43 Average 50 Weak  
## Parent.Engagement.Score Parent.Environment.Icon Parent.Environment.Score  
## 1 56 Average 47  
## 2 46 Average 50  
## 3 47 Weak 41  
## 4 53 Strong 58  
## 5 53 Average 49  
## 6 46 Average 51  
## Average.Student.Attendance Rate.of.Misconducts..per.100.students.  
## 1 96.0 2.0  
## 2 95.6 15.7  
## 3 95.7 2.3  
## 4 95.5 10.4  
## 5 93.3 15.6  
## 6 97.0 2.3  
## Average.Teacher.Attendance  
## 1 96.4  
## 2 95.3  
## 3 94.7  
## 4 95.8  
## 5 96.9  
## 6 96.9  
## Individualized.Education.Program.Compliance.Rate Pk.2.Literacy..  
## 1 95.8 80.1  
## 2 100.0 62.4  
## 3 98.3 53.7  
## 4 100.0 76.9  
## 5 100.0 <NA>  
## 6 100.0 <NA>  
## Pk.2.Math.. Gr3.5.Grade.Level.Math.. Gr3.5.Grade.Level.Read..  
## 1 43.3 89.6 84.9  
## 2 51.7 21.9 15.1  
## 3 26.6 38.3 34.7  
## 4 <NA> 26 24.7  
## 5 <NA> <NA> <NA>  
## 6 <NA> <NA> <NA>  
## Gr3.5.Keep.Pace.Read.. Gr3.5.Keep.Pace.Math.. Gr6.8.Grade.Level.Math..  
## 1 60.7 62.6 81.9  
## 2 29 42.8 38.5  
## 3 43.7 57.3 48.8  
## 4 61.8 49.7 39.2  
## 5 <NA> <NA> <NA>  
## 6 <NA> <NA> 60.7  
## Gr6.8.Grade.Level.Read.. Gr6.8.Keep.Pace.Math. Gr6.8.Keep.Pace.Read..  
## 1 85.2 52 62.4  
## 2 27.4 44.8 42.7  
## 3 39.2 46.8 44  
## 4 27.2 69.7 60.6  
## 5 <NA> <NA> <NA>  
## 6 39.8 53.7 59.8  
## Gr.8.Explore.Math.. Gr.8.Explore.Read.. ISAT.Exceeding.Math..  
## 1 66.3 77.9 69.7  
## 2 14.1 34.4 16.8  
## 3 7.5 21.9 18.3  
## 4 9.1 18.2 11.1  
## 5 <NA> <NA> NA  
## 6 17.5 20.8 34.5  
## ISAT.Exceeding.Reading.. ISAT.Value.Add.Math ISAT.Value.Add.Read  
## 1 64.4 0.2 0.9  
## 2 16.5 0.7 1.4  
## 3 15.5 -0.9 -1.0  
## 4 9.6 0.9 2.4  
## 5 NA NA NA  
## 6 15.6 0.2 0.3  
## ISAT.Value.Add.Color.Math ISAT.Value.Add.Color.Read  
## 1 Yellow Green  
## 2 Green Green  
## 3 Red Red  
## 4 Green Green  
## 5 <NA> <NA>  
## 6 Yellow Yellow  
## Students.Taking..Algebra.. Students.Passing..Algebra..  
## 1 67.1 54.5  
## 2 17.2 27.3  
## 3 <NA> <NA>  
## 4 42.9 25  
## 5 <NA> <NA>  
## 6 29.2 50  
## X9th.Grade.EXPLORE..2009. X9th.Grade.EXPLORE..2010.  
## 1 <NA> <NA>  
## 2 <NA> <NA>  
## 3 <NA> <NA>  
## 4 <NA> <NA>  
## 5 14.6 14.8  
## 6 <NA> <NA>  
## X10th.Grade.PLAN..2009. X10th.Grade.PLAN..2010.  
## 1 <NA> <NA>  
## 2 <NA> <NA>  
## 3 <NA> <NA>  
## 4 <NA> <NA>  
## 5 <NA> 16  
## 6 <NA> <NA>  
## Net.Change.EXPLORE.and.PLAN X11th.Grade.Average.ACT..2011.  
## 1 <NA> <NA>  
## 2 <NA> <NA>  
## 3 <NA> <NA>  
## 4 <NA> <NA>  
## 5 1.4 <NA>  
## 6 <NA> <NA>  
## Net.Change.PLAN.and.ACT College.Eligibility.. Graduation.Rate..  
## 1 <NA> <NA> <NA>  
## 2 <NA> <NA> <NA>  
## 3 <NA> <NA> <NA>  
## 4 <NA> <NA> <NA>  
## 5 <NA> <NA> <NA>  
## 6 <NA> <NA> <NA>  
## College.Enrollment.Rate.. College.Enrollment..number.of.students.  
## 1 <NA> 813  
## 2 <NA> 521  
## 3 <NA> 1324  
## 4 <NA> 556  
## 5 <NA> 302  
## 6 <NA> 266  
## General.Services.Route Freshman.on.Track.Rate.. Community.Area.Number  
## 1 33 <NA> 7  
## 2 46 <NA> 43  
## 3 44 <NA> 70  
## 4 42 <NA> 61  
## 5 40 91.8 34  
## 6 31 <NA> 14  
## Community.Area.Name Ward Police.District  
## 1 LINCOLN PARK 43 18  
## 2 SOUTH SHORE 7 4  
## 3 ASHBURN 13 8  
## 4 NEW CITY 20 9  
## 5 ARMOUR SQUARE 11 9  
## 6 ALBANY PARK 39 17

#Data that only describes the numeric variables  
numeric.data <- CPS.data[, map\_lgl(CPS.data, is.numeric)]  
head(numeric.data)

## School.ID ZIP.Code Safety.Score Environment.Score Instruction.Score  
## 1 610038 60614 99 74 66  
## 2 610281 60649 54 74 84  
## 3 610185 60652 61 50 36  
## 4 609993 60609 56 45 37  
## 5 610513 60609 49 60 55  
## 6 610212 60625 66 66 71  
## Average.Student.Attendance Rate.of.Misconducts..per.100.students.  
## 1 96.0 2.0  
## 2 95.6 15.7  
## 3 95.7 2.3  
## 4 95.5 10.4  
## 5 93.3 15.6  
## 6 97.0 2.3  
## Average.Teacher.Attendance  
## 1 96.4  
## 2 95.3  
## 3 94.7  
## 4 95.8  
## 5 96.9  
## 6 96.9  
## Individualized.Education.Program.Compliance.Rate ISAT.Exceeding.Math..  
## 1 95.8 69.7  
## 2 100.0 16.8  
## 3 98.3 18.3  
## 4 100.0 11.1  
## 5 100.0 NA  
## 6 100.0 34.5  
## ISAT.Exceeding.Reading.. ISAT.Value.Add.Math ISAT.Value.Add.Read  
## 1 64.4 0.2 0.9  
## 2 16.5 0.7 1.4  
## 3 15.5 -0.9 -1.0  
## 4 9.6 0.9 2.4  
## 5 NA NA NA  
## 6 15.6 0.2 0.3  
## College.Enrollment..number.of.students. General.Services.Route  
## 1 813 33  
## 2 521 46  
## 3 1324 44  
## 4 556 42  
## 5 302 40  
## 6 266 31  
## Community.Area.Number Ward Police.District  
## 1 7 43 18  
## 2 43 7 4  
## 3 70 13 8  
## 4 61 20 9  
## 5 34 11 9  
## 6 14 39 17

# Erin's section

# Charles' section  
# set seed  
set.seed(100)  
# First obtain numeric values only  
numcols <- unlist(lapply(CPS.data, is.numeric))  
CPS.num <- na.omit(CPS.data[,numcols])  
# Second check correlations  
corrs <- cor(CPS.num)  
corrs[upper.tri(corrs)] <- 0  
diag(corrs) <- 0  
CPS.num.new <- scale(CPS.num[,!apply(corrs, 2, function(x) any (abs(x) > 0.7))])  
train\_size <- floor(0.7\*nrow(CPS.num.new))  
train\_ind <- sample(seq\_len(nrow(CPS.num.new)), size = train\_size)  
CPS\_train <- CPS.num.new[train\_ind,]  
CPS\_test <- CPS.num.new[-train\_ind,]  
  
# use a simple neural network to predict Instruction Score  
nn1 = neuralnet(Instruction.Score ~ ., data=CPS\_train)  
  
# check we get somewhat reasonable predictions  
nnpred <- compute(nn1, CPS\_test)  
nnpred1\_results <- ifelse(nnpred$net.result > iscore\_med, 1, 0)  
aboveavg <- ifelse(CPS\_test[,3] > iscore\_med, 1, 0)  
sum(abs(nnpred1\_results - aboveavg))  
# we see our naive neural network doesn't perform very well, misclassifying 50/131 test points

# Norma's section

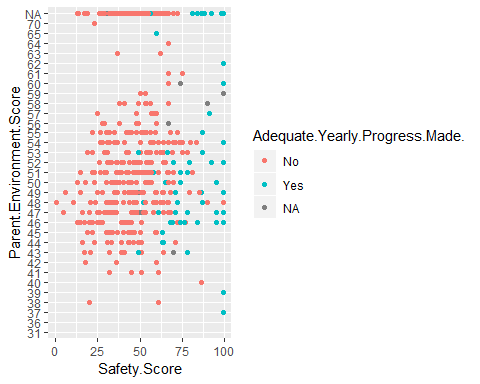
# Ryan’s Section

Ideas:

* Other Variables connection to classificaiton
* Are there certain ways to categorize elementary middle and high school?

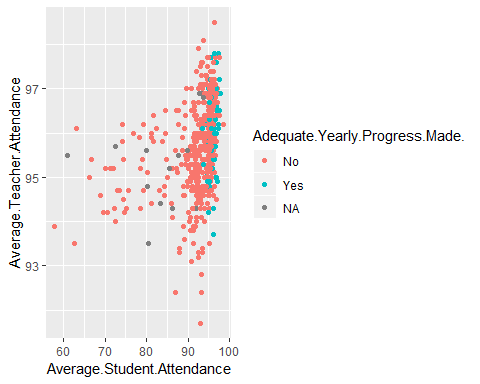
#Safety and Parents  
ggplot(CPS.data) +  
 geom\_point(aes(x = Safety.Score, y = Parent.Environment.Score, col = Adequate.Yearly.Progress.Made.))

## Warning: Removed 53 rows containing missing values (geom\_point).



#Attendance  
ggplot(filter(CPS.data, Average.Teacher.Attendance != 0)) +  
 geom\_point(aes(x = Average.Student.Attendance, y = Average.Teacher.Attendance, col = Adequate.Yearly.Progress.Made.))

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



#

1st graph: High safety scores indicate higher progress.

2nd graph: Most progress happened when attendance was high for both parties.

head(numeric.data)

## School.ID ZIP.Code Safety.Score Environment.Score Instruction.Score  
## 1 610038 60614 99 74 66  
## 2 610281 60649 54 74 84  
## 3 610185 60652 61 50 36  
## 4 609993 60609 56 45 37  
## 5 610513 60609 49 60 55  
## 6 610212 60625 66 66 71  
## Average.Student.Attendance Rate.of.Misconducts..per.100.students.  
## 1 96.0 2.0  
## 2 95.6 15.7  
## 3 95.7 2.3  
## 4 95.5 10.4  
## 5 93.3 15.6  
## 6 97.0 2.3  
## Average.Teacher.Attendance  
## 1 96.4  
## 2 95.3  
## 3 94.7  
## 4 95.8  
## 5 96.9  
## 6 96.9  
## Individualized.Education.Program.Compliance.Rate ISAT.Exceeding.Math..  
## 1 95.8 69.7  
## 2 100.0 16.8  
## 3 98.3 18.3  
## 4 100.0 11.1  
## 5 100.0 NA  
## 6 100.0 34.5  
## ISAT.Exceeding.Reading.. ISAT.Value.Add.Math ISAT.Value.Add.Read  
## 1 64.4 0.2 0.9  
## 2 16.5 0.7 1.4  
## 3 15.5 -0.9 -1.0  
## 4 9.6 0.9 2.4  
## 5 NA NA NA  
## 6 15.6 0.2 0.3  
## College.Enrollment..number.of.students. General.Services.Route  
## 1 813 33  
## 2 521 46  
## 3 1324 44  
## 4 556 42  
## 5 302 40  
## 6 266 31  
## Community.Area.Number Ward Police.District  
## 1 7 43 18  
## 2 43 7 4  
## 3 70 13 8  
## 4 61 20 9  
## 5 34 11 9  
## 6 14 39 17

# Numerical Analysis

#Correlation of all Variables  
correlation.matrix <- cor(numeric.data, use = "complete.obs")  
correlation.dataset <- as.data.frame(correlation.matrix)  
most.correlation <- which(abs(correlation.matrix) < 1 & abs(correlation.matrix) > .70)  
  
rows.list <- list()  
columns.list <- list()  
j <- 1  
#For Loop for finding maximum correlations  
for(i in most.correlation) {  
 #Correlation  
 row <- ceiling(i/ncol(numeric.data))  
 column <- ifelse(i%%ncol(numeric.data) == 0, ncol(numeric.data), i%%ncol(numeric.data))  
 rows.list[[j]] <- rownames(correlation.dataset[row,])  
 columns.list[[j]] <- names(correlation.dataset)[column]  
 j = j+1  
}  
  
row.column.names = cbind(rows.list, columns.list)  
  
#New Dataframe To hold Rows/ Columns  
row.column.dataframe <- data.frame(  
 row = NULL,  
 column = NULL  
)  
#putting values in dataframe  
for(i in 1:length(rows.list)) {  
 newRow = data.frame(row = rows.list[[i]], column = columns.list[[i]])  
 row.column.dataframe <- rbind(row.column.dataframe, newRow)  
}  
  
#Changing variable types to character  
row.column.dataframe$row <- as.character(row.column.dataframe$row)  
row.column.dataframe$column <- as.character(row.column.dataframe$column)  
  
#Removing Duplicates  
for (i in 1:nrow(row.column.dataframe))  
{  
 row.column.dataframe[i, ] = sort(row.column.dataframe[i, ])  
}  
row.column.dataframe <- row.column.dataframe[!duplicated(row.column.dataframe),]  
  
#Viewing Dataframe of Variables with the most Correlation  
row.column.dataframe

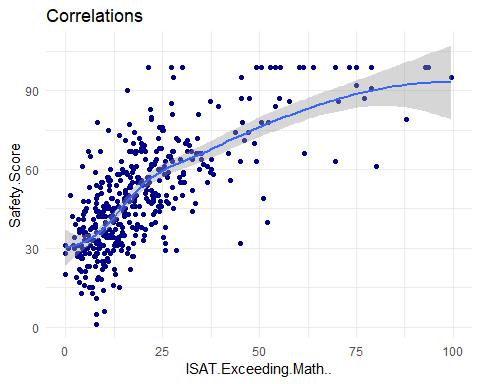
## row column  
## 1 ISAT.Exceeding.Math.. Safety.Score  
## 2 ISAT.Exceeding.Reading.. Safety.Score  
## 3 Environment.Score Instruction.Score  
## 6 ISAT.Exceeding.Math.. ISAT.Exceeding.Reading..  
## 9 Community.Area.Number General.Services.Route  
## 11 Police.District Ward

#For Loop For Creating Scatterplots of data with the Greatest Correlation  
for(i in 1:nrow(row.column.dataframe)) {  
 #Extract column and row variable  
 row <- row.column.dataframe[[i, 1]]  
 column <- row.column.dataframe[[i, 2]]  
 #Plot  
 print(ggplot(CPS.data) +  
 geom\_point(aes\_string(x = row, y = column), color = "blue4") +  
 geom\_smooth(aes\_string(x = row, y = column)) +  
 theme\_minimal() +  
 ggtitle("Correlations"))  
}

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

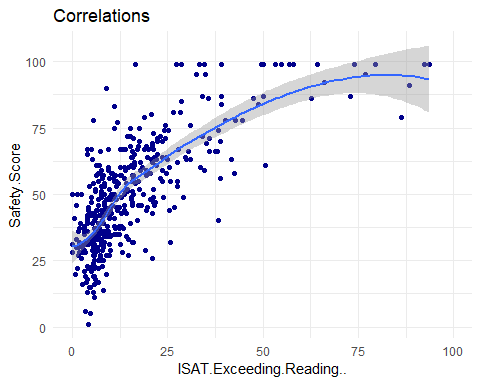
## Warning: Removed 130 rows containing non-finite values (stat\_smooth).

## Warning: Removed 130 rows containing missing values (geom\_point).



## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

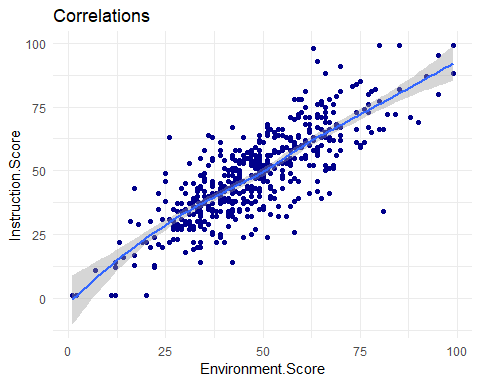
## Warning: Removed 130 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 130 rows containing missing values (geom\_point).



## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 53 rows containing non-finite values (stat\_smooth).

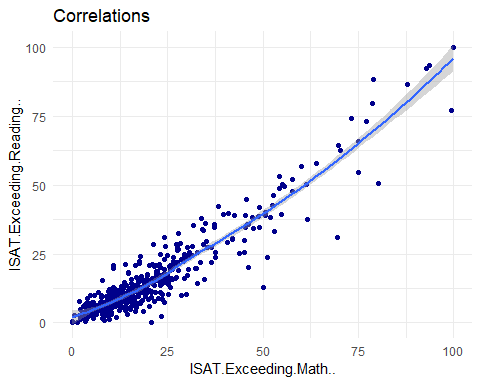
## Warning: Removed 53 rows containing missing values (geom\_point).



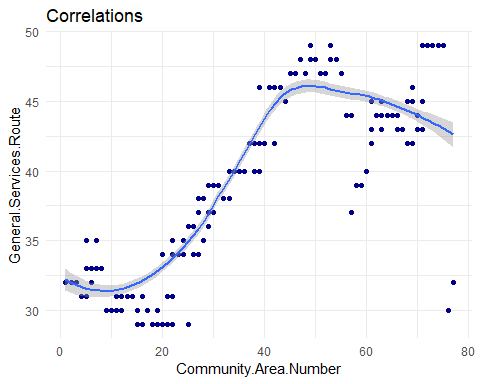
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 90 rows containing non-finite values (stat\_smooth).

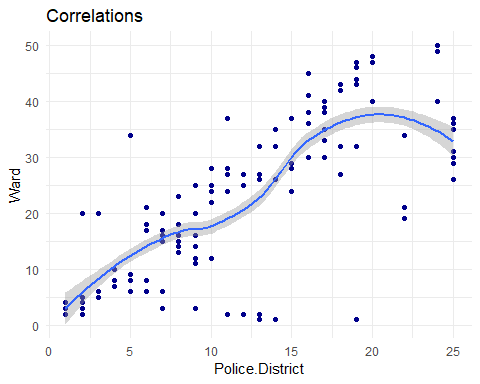
## Warning: Removed 90 rows containing missing values (geom\_point).



## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



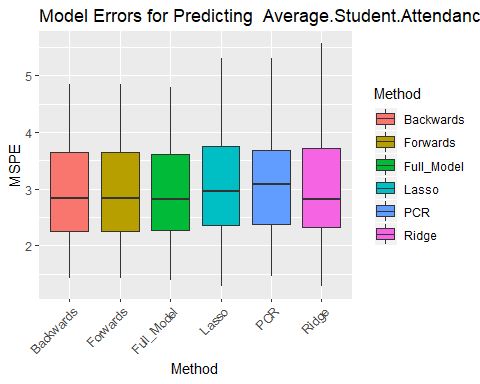
## How well can we predict student attendance?

Goal - determine best model between lasso regression, ridge regression, and least squares regression with variable selection.

which.model <- function(vrbl){  
full.error <- rep(0, 50)  
backwards.error <- rep(0, 50)  
forwards.error <- rep(0, 50)  
lasso.error <- rep(0, 50)  
ridge.error <- rep(0, 50)  
pcr.error <- rep(0, 50)  
  
#Data frame with no NA values  
no.na <- na.omit(numeric.data)  
  
for(i in 1:50){  
test.set <- sample\_n(no.na, 100)  
train.set <- suppressMessages(anti\_join(no.na, test.set))  
  
#Least Squares Model  
fullLinMod.SA <- lm(as.formula(paste(vrbl, "~.")), data = train.set)  
  
#Stepwise  
backwards <- step(fullLinMod.SA, direction = "backward", trace = FALSE)  
forwards <- step(fullLinMod.SA, direction = "backward", trace = FALSE)  
  
#Lasso Regression  
modMatrix <- model.matrix(as.formula(paste(vrbl, "~.")), data= train.set)  
lassoMod = cv.glmnet(modMatrix, y = train.set[,vrbl], alpha = 1, nfolds = 5)  
  
#Ridge Regression  
ridgeMod = cv.glmnet(modMatrix, y = train.set[,vrbl], alpha = 0, nfolds = 5)  
  
#PCR  
pcrMod <- pcr(as.formula(paste(vrbl, "~.")), data=train.set, scale=TRUE)  
  
#predictions  
full.predictions <- predict(fullLinMod.SA, test.set)  
backwards.predictions <- predict(backwards, test.set)  
forwards.predictions <- predict(forwards, test.set)  
  
newX <- model.matrix(as.formula(paste("~.-", vrbl)), data=test.set)  
lasso.predictions <- predict(lassoMod, s= lassoMod$lambda.min, newx= newX)  
ridge.predictions <- predict(ridgeMod, s= ridgeMod$lambda.min, newx= newX)  
pcr.predictions <- predict(pcrMod, test.set)  
  
MSE.full = mean((full.predictions - test.set[,vrbl])^2)  
MSE.back = mean((backwards.predictions - test.set[,vrbl])^2)  
MSE.for = mean((forwards.predictions - test.set[,vrbl])^2)  
MSE.lasso = mean((lasso.predictions - test.set[,vrbl])^2)  
MSE.ridge = mean((ridge.predictions - test.set[,vrbl])^2)  
MSE.pcr = mean((pcr.predictions - test.set[,vrbl])^2)  
  
full.error[[i]] <- MSE.full  
backwards.error[[i]] <- MSE.back  
forwards.error[[i]] <- MSE.for  
lasso.error[[i]] <- MSE.lasso  
ridge.error[[i]] <- MSE.ridge  
pcr.error[[i]] <- MSE.pcr  
}  
  
data = gather(data.frame(Full\_Model = full.error,  
 Backwards = backwards.error,  
 Forwards = forwards.error,  
 Lasso = lasso.error,  
 Ridge = ridge.error,  
 PCR = pcr.error), key = "Method", value = "MSPE")  
print(ggplot(data) +   
 geom\_boxplot(aes(x = Method, y = MSPE, fill = Method)) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 ggtitle(paste("Model Errors for Predicting ", vrbl)))  
}

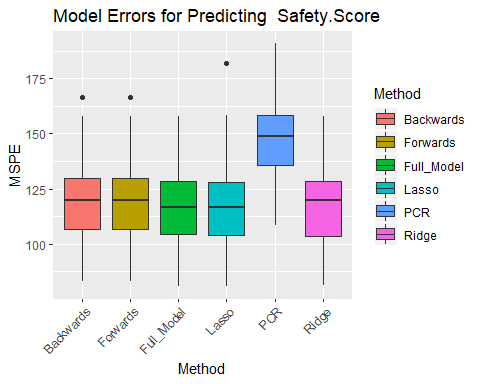
## Student Attendance

set.seed(123)  
which.model("Average.Student.Attendance")



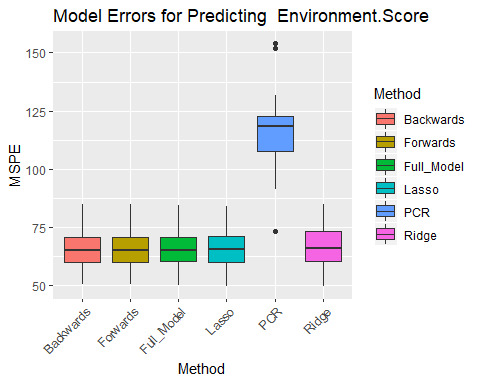
## Safety Score Best Model

set.seed(123)  
which.model("Safety.Score")



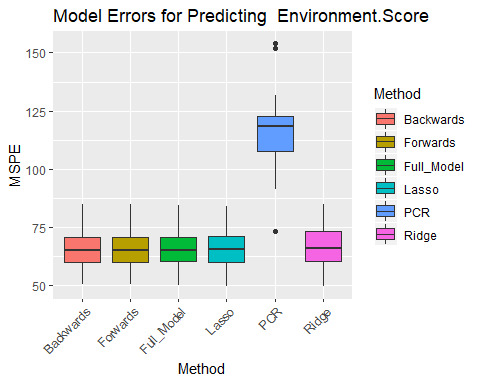
## Environment Score Best Model

set.seed(123)  
which.model("Environment.Score")



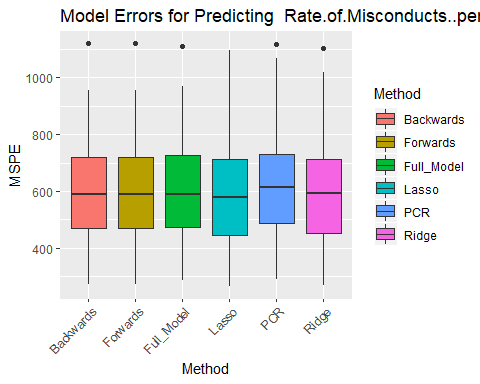
## Instruction Score Best Model

set.seed(123)  
which.model("Environment.Score")



# Rate of Misconduct

set.seed(123)  
which.model("Rate.of.Misconducts..per.100.students.")



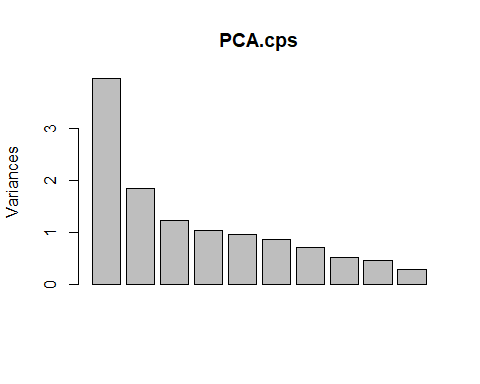
## Classification for School Grade-level

#Function that removes collinearity for data  
remove.collinearity <- function(data, c) {  
 #Correlation of all Variables  
 correlation.matrix <- cor(data, use = "complete.obs")  
 correlation.dataset <- as.data.frame(correlation.matrix)  
 most.correlation <- which(abs(correlation.matrix) < 1 & abs(correlation.matrix) > c)  
  
 rows.list <- list()  
 columns.list <- list()  
 j <- 1  
 #For Loop for finding maximum correlations  
 for(i in most.correlation) {  
 #Correlation  
 row <- ceiling(i/ncol(data))  
 column <- ifelse(i%%ncol(data) == 0, ncol(data), i%%ncol(data))  
 rows.list[[j]] <- rownames(correlation.dataset[row,])  
 columns.list[[j]] <- names(correlation.dataset)[column]  
 j = j+1  
 }  
  
 row.column.names = cbind(rows.list, columns.list)  
  
 #New Dataframe To hold Rows/ Columns  
 row.column.dataframe <- data.frame(  
 row = NULL,  
 column = NULL  
 )  
 #putting values in dataframe  
 for(i in 1:length(rows.list)) {  
 newRow = data.frame(row = rows.list[[i]], column = columns.list[[i]])  
 row.column.dataframe <- rbind(row.column.dataframe, newRow)  
 }  
  
 #Changing variable types to character  
 row.column.dataframe$row <- as.character(row.column.dataframe$row)  
 row.column.dataframe$column <- as.character(row.column.dataframe$column)  
  
 #Removing Duplicates  
 for (i in 1:nrow(row.column.dataframe))  
 {  
 row.column.dataframe[i, ] = sort(row.column.dataframe[i, ])  
 }  
 row.column.dataframe <- row.column.dataframe[!duplicated(row.column.dataframe),]  
  
 #Dataframe of Variables with the most Correlation  
 table <- row.column.dataframe  
   
 #creates new dataframe without correlations above c  
 new.data = data  
 for(i in 1:nrow(table)) {  
 new.data[,table[i,2]] <- NULL  
 }  
 return(new.data)  
}

newdata = numeric.data %>%  
 select\_if(Negate(is.integer)) %>%  
 select\_if(is.numeric)  
  
newdata <- remove.collinearity(newdata, .1)  
newdata <- cbind(numeric.data, CPS.data$Elementary..Middle..or.High.School)  
newdata$Elementary..Middle..or.High.School <- newdata$`CPS.data$Elementary..Middle..or.High.School`  
newdata$`CPS.data$Elementary..Middle..or.High.School`<- NULL  
  
ldaMod <- lda(Elementary..Middle..or.High.School~., data = newdata)

# PCA to hopefully find separation in categories

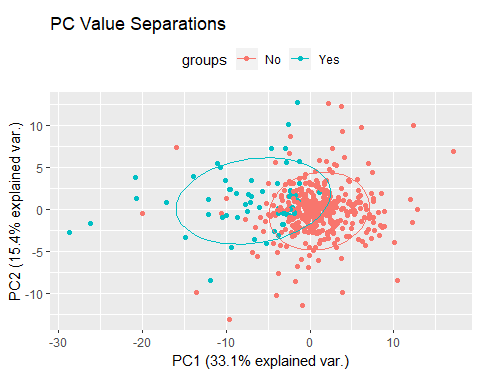
data.progress <- cbind(CPS.data$Adequate.Yearly.Progress.Made.,numeric.data)  
data.progress$Adequate.Yearly.Progress.Made. <- data.progress[,"CPS.data$Adequate.Yearly.Progress.Made."]  
data.progress[,"CPS.data$Adequate.Yearly.Progress.Made."] <- NULL  
data.progress[,"ZIP.Code"] <- NULL  
data.progress[,"Community.Area.Number"] <- NULL  
data.progress[,"CPS.data"] <- NULL  
data.progress[,"School.ID"] <- NULL  
data.progress[,"Ward"] <- NULL  
data.progress[,"General.Services.Route" ] <- NULL  
data.progress[,"Police.District" ] <- NULL  
data.progress <- na.omit(data.progress)  
  
PCA.cps <- prcomp(data.progress[1:12], center = TRUE, scale. = TRUE, rank = 3)  
  
plot(PCA.cps)



ggbiplot(PCA.cps, choices = c(1,2), obs.scale = 1, var.scale = 1, group = data.progress$Adequate.Yearly.Progress.Made., ellipse = TRUE, varname.size=0, var.axes = F) +   
 scale\_color\_discrete() +   
 theme(legend.direction = 'horizontal', legend.position = 'top') +  
 ggtitle("PC Value Separations")

## Warning in sweep(pcobj$x, 2, 1/(d \* nobs.factor), FUN = "\*"): STATS is  
## longer than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(v, 2, d^var.scale, FUN = "\*"): STATS is longer than the  
## extent of 'dim(x)[MARGIN]'

 ###Inspecting PC value weights

#In order of PC1 Vlaues  
PC.weights <- as.data.frame(PCA.cps$rotation)[,1:2]  
PC.weights <- PC.weights[order(abs(PC.weights$PC1), decreasing = TRUE),]  
head(PC.weights)

## PC1 PC2  
## Safety.Score -0.4284597 0.09436216  
## ISAT.Exceeding.Math.. -0.4195862 0.19323209  
## ISAT.Exceeding.Reading.. -0.4150908 0.18859388  
## Average.Student.Attendance -0.3336153 0.21546317  
## Environment.Score -0.3264192 -0.39813251  
## Rate.of.Misconducts..per.100.students. 0.2838580 -0.20630803

#In order of PC2 Values  
PC.weights <- PC.weights[order(abs(PC.weights$PC2), decreasing = TRUE),]  
head(PC.weights)

## PC1 PC2  
## Instruction.Score -0.25436362 -0.4772743  
## College.Enrollment..number.of.students. -0.09375909 0.4623220  
## ISAT.Value.Add.Math -0.18547723 -0.4268627  
## Environment.Score -0.32641920 -0.3981325  
## Average.Student.Attendance -0.33361528 0.2154632  
## Rate.of.Misconducts..per.100.students. 0.28385798 -0.2063080

\*PC1’s magnitude emphasizes better safety/environment and test taking. This definitely did the best job at separating the data between the two.

\*PC2 correlates with better teaching, college enrollment, math, and environment

# Jason's section