**Demographic Influencers Elementary Public School Performance in Florida**

Team No. 002

Summer 2020

Ana Schambach

Clinton McConnell

Gabe Uhrig

Keeley Ables

Louise Co

Matthew Stockhaus

Contents

[**Project Background** 3](#_Toc50937520)

[**Project Scope** 3](#_Toc50937521)

[**Business Problem** 3](#_Toc50937522)

[**Data Acquisition and Wrangling Process** 5](#_Toc50937523)

[Fields/Variables and Structure of Base Data 5](#_Toc50937524)

[Fields/Variables and Structure of Performance Data 6](#_Toc50937525)

[Statistical Process Control: XmR charts 7](#_Toc50937526)

[**Techniques Used for Analysis and Results** 9](#_Toc50937527)

[1): Descriptive Statistics 9](#_Toc50937528)

[2): General and Map Visualizations 11](#_Toc50937529)

[3): Linear Model and Results 14](#_Toc50937530)

[4): Machine Learning Classification and Results 17](#_Toc50937531)

[**Business Insights and Conclusion** 25](#_Toc50937532)

[**References** 26](#_Toc50937533)

[**Appendix – R Code** 27](#_Toc50937534)

# **Project Background**

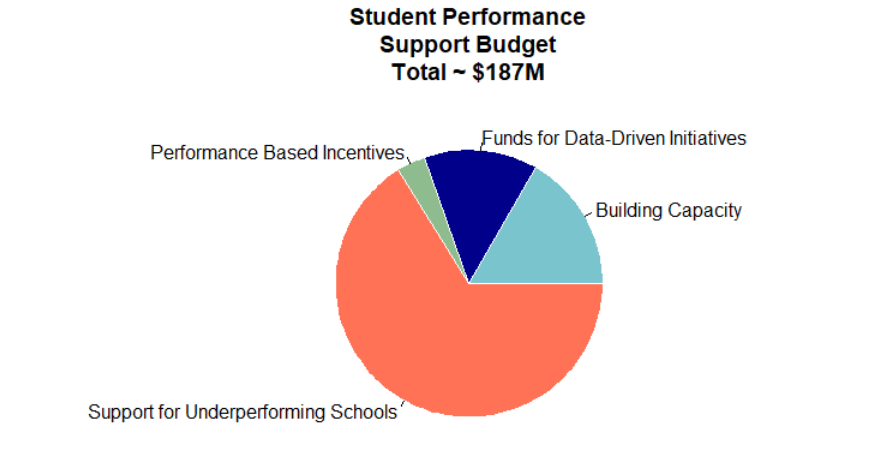
The group project was an opportunity to challenge our skills in R. We selected a comprehensive dataset, determined an appropriate set of business questions, and performed various data analyses and visualizations to address the business problem.

# **Project Scope**

Our mission was to analyze Florida school district grades and improvement metrics from 2008-2019 to determine if, and which, external factors had the largest impact on a school’s change in performance.

# **Business Problem**

The budget allotted for student support in the state of Florida equated to approximately $187,289,696 for the 2019-2020 school year.  The following pie chart is a high-level, visual, itemization of the allocation of funds for that academic year.



A brief elaboration on the fiscal categories in the above visualization is as follows: “Total Building Capacity” refers to monies appropriated for professional development to build capacity among teachers, administrators, and district personnel. The category “Total Funds Initiatives Driven by Data” refers to funds allocated for programs (initiatives) implemented due to a signaled area of weakness found in student data. For example, student data may show a weakness in Reading comprehension in a school district. The district may address the problem by purchasing new Reading curriculum with enhanced differentiated student supports/interventions infused into the program. The “Total Performance-Based Initiatives” category budgets money for yearly bonuses to reward teachers and administrators whose students meet a set criterion for standardized testing scores. Finally, monies set aside for “Total Support for Underperforming Schools” was meant to support schools whose school grades were a “D” or an “F” to aid students in accelerated learning. This would enable schools to fill the gap between students who traditionally perform on grade-level and students traditionally perform more than two levels below grade level.

In addition, according to the Florida Department of Education, data concerning academic performance, student growth, and proficiency statewide, on average, have stagnated in past years. This is despite the monies allocated to support student success. As a result, our team has deemed it prudent to identify the schools that are improving and declining. This identification would allow for the most appropriate optimization of resources for maximized support of student achievement. To summarize our objective, we want to know what schools/districts are consistently meeting or exceeding performance expectations and which schools/districts traditionally enjoy limited academic success. We also want to determine the potential factors for this academic success, or lack thereof. This knowledge will give insight into how and where to best distribute funds to help Florida’s struggling students fill the performance gap.

In order to objectively measure student academic achievement, the Florida Department of Education quantifies this success using percent of students meeting proficiency at grade level, percent of students making learning gains, and percent of students advancing out of the lowest quartile of student testing scores. An amalgamation of these percentages is included in a school grade given to every school in the state of Florida. Therefore, for the purpose of this team’s objective, we are utilizing school grades for schools across Florida to quantify “improving” and “declining” schools. Our variables potentially driving our independent variable (school grade) are whether or not the school is a charter school, whether or not the school is a Title 1 school, percent minority students attending the school, percent of economically disadvantaged students in attendance, and average income of families whose students attend the school.

For clarification, it is necessary to define some of the education jargon included in our business problem.

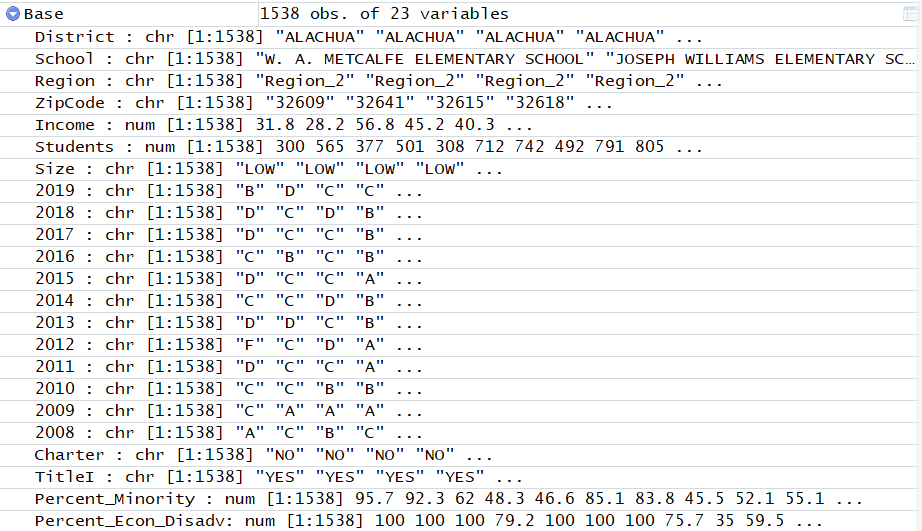
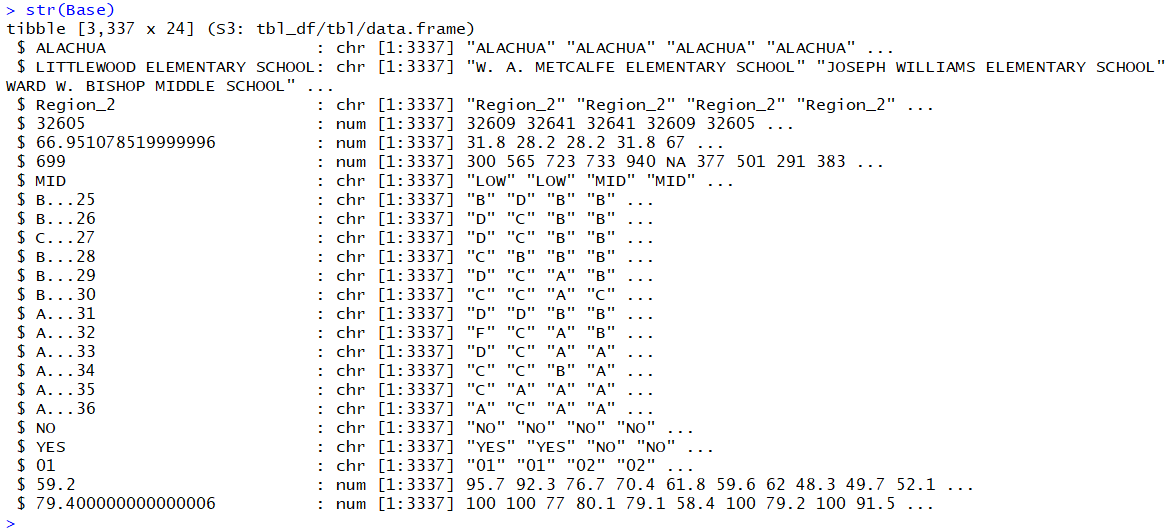
* **Differentiation**: Customized instruction to address the different levels of abilities within a classroom/grade-level/school. This is necessary because even within one classroom containing students in one grade level, performance abilities can range from three levels below grade level to three levels above. This is just an example. It is common for an even greater range of abilities to exist within one classroom. Therefore, instruction must be customized to support all student needs.
* **Charter schools**: Schools that are state funded, but usually receive less state funding. They are privately run and are not held to the same government regulations as public schools. However, they still implement standardized testing which holds them accountable to the state.
* **Title 1 schools**: Schools which serve a significant number of economically disadvantaged students within their student population. Therefore, these schools receive additional federal funds to support students and their families.
* **Standardized Testing:** A common state assessment given to students across the state of Florida used to measure student proficiency at grade level, as well as student growth. Each state has their own. In Florida, state testing is called Florida Standardized Assessment (FSA).
* **Scale Score/Levels (in the context of describing student performance on state testing):** Students’ performance on standardized testing is measured by a scale score. Ranges of scale scores are clustered together and assigned levels. Student performance is expressed in levels achieved on the FSA. Students who achieve a level 3 or higher are considered “proficient at grade level.”

# **Data Acquisition and Wrangling Process**

We utilized data provided by team member Keeley Ables, that focused on Florida Grade 5 Test Data spanning 2008 – 2019. The datasets were originally divided by year and each file pertained to a certain subject matter, i.e., Math or Reading, each school’s overall grade by subject matter, as well as additional columns pertaining to the school. Examples are District Name and Number, School Number, Number of Students, Mean Scale Score, Percentage in Level 3 or Above, ad final Percentage at Each Achievement Level (1:5) with an associating score. We then merged in geographical information as well as socioeconomic information pertaining to each school within the regions of Florida.

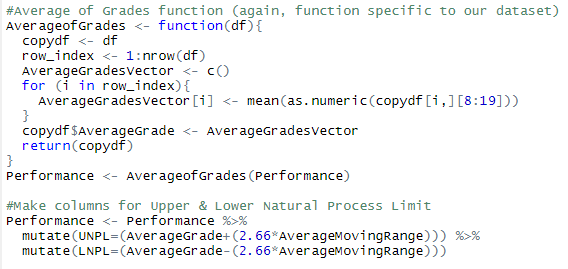
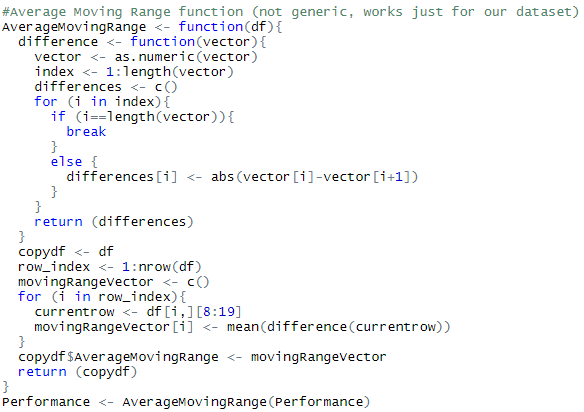
Once a final dataset was established in Excel, we utilized R to read the raw excel data file and begin data wrangling processes. We modified the final dataset column names to ones that are more streamlined and easier to include in testing, R functions, and general column header length. We further refined our dataset in R by selecting only elementary schools from the list of schools, and then by only using complete cases—not including NA values that were introduced through merging of data or from original dataset. We called this our “Base” set.

# Fields/Variables and Structure of Base Data

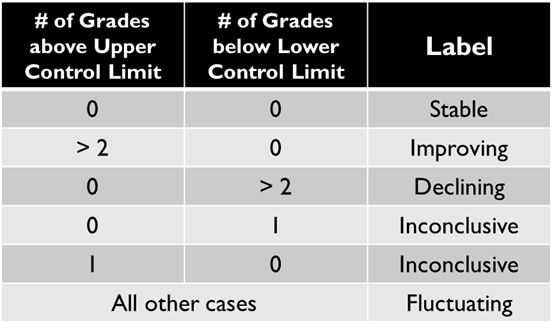
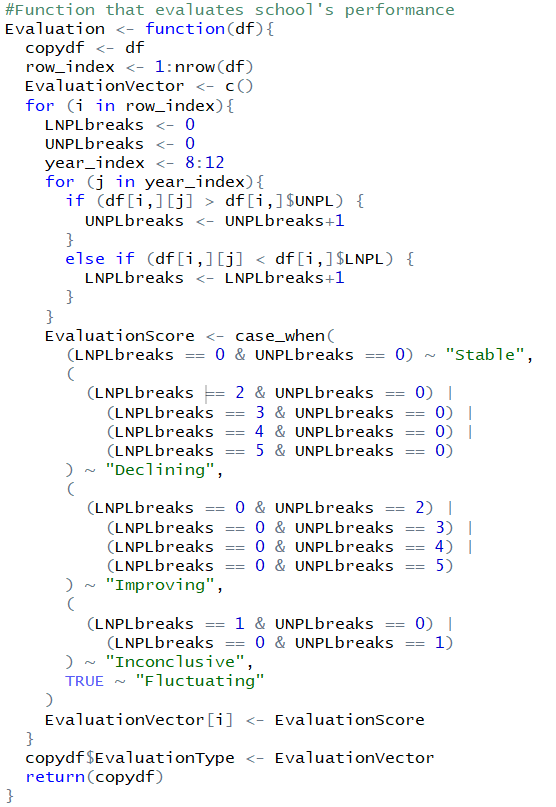
 

To proceed with our analysis, we decided to recode the letter grades as "A"=4,"B"=3,"C"=2,"D"=1,"F"=0. We then employed the concept of statistical process controls (as described by Donald Wheeler in Understanding Variation: The Key to Managing Chaos) to help us determine if a school’s performance was improving or declining.

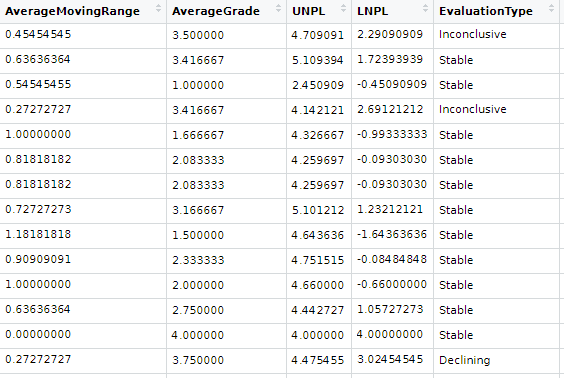
We then calculated average grades, average moving ranges, upper control limits, and lower control limits for each school in R with customized functions as seen below:



We then analyzed each school’s performance over the past 5 years in order to determine whether a school is improving or declining with an iterating function that holds the case when following the rules we established. ​

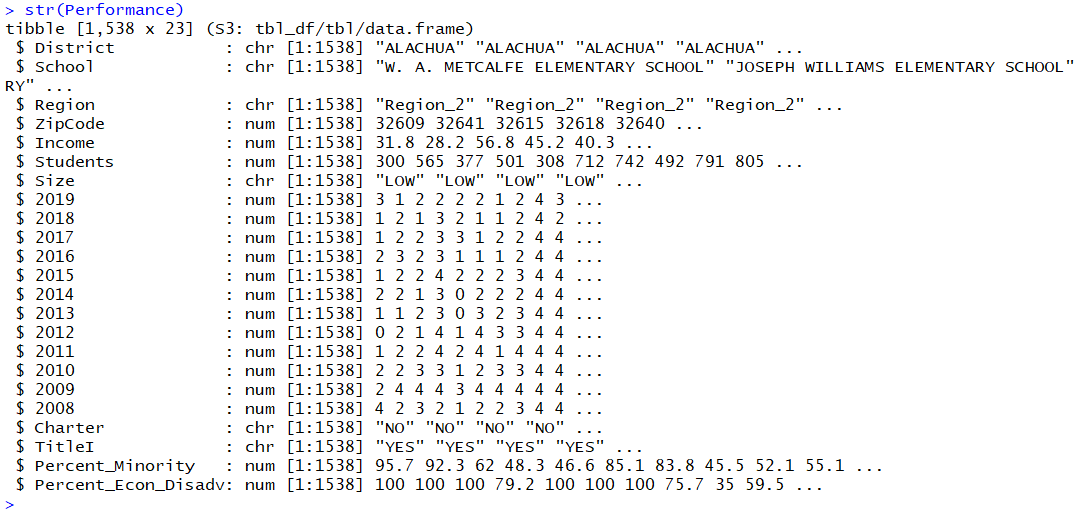
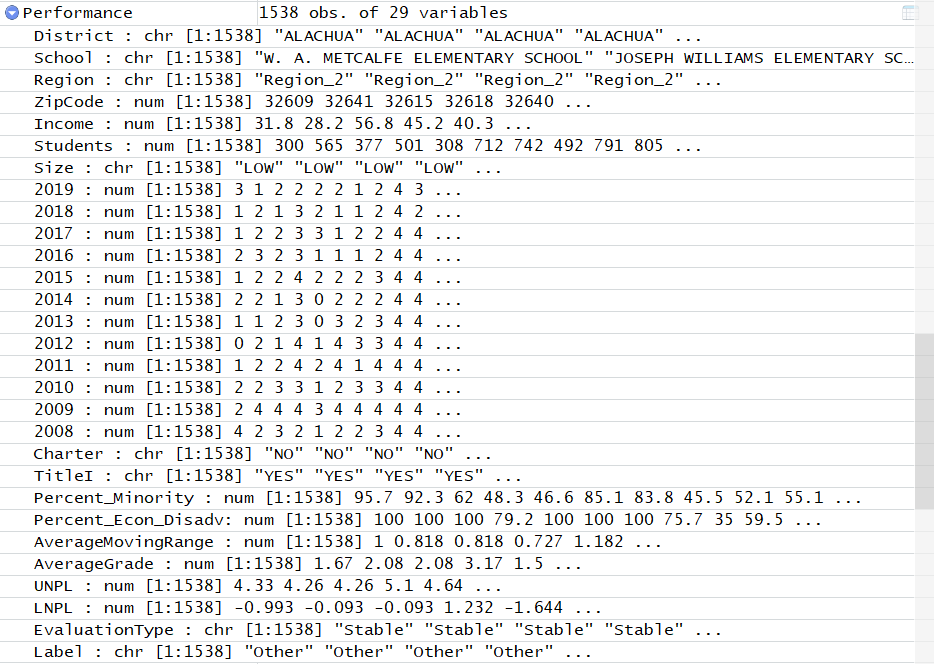


Example of output:

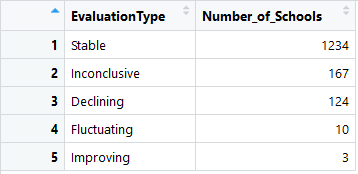


This became our new dataset named “Performance”.

# Fields/Variables and Structure of Performance Data



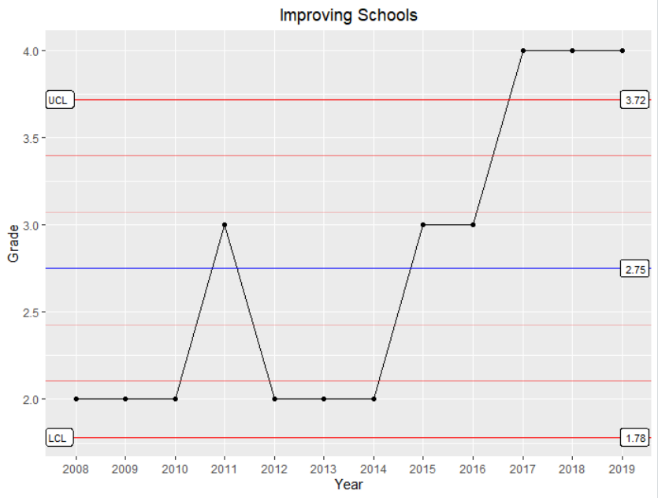
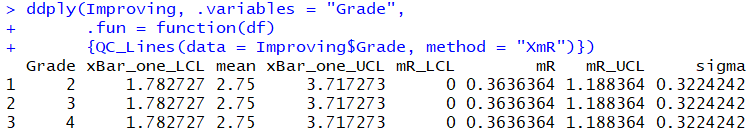
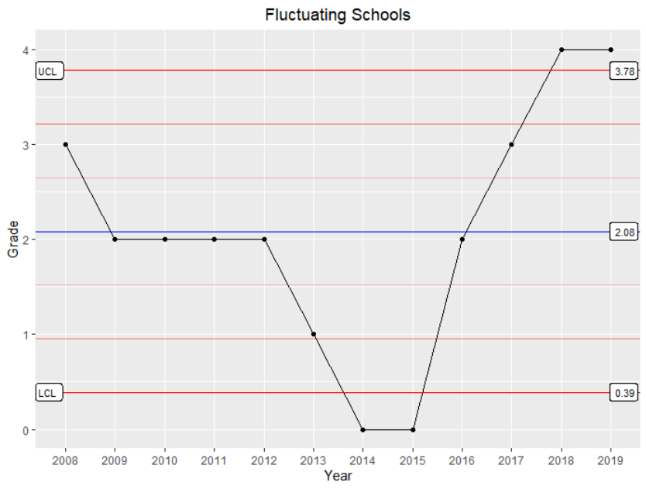
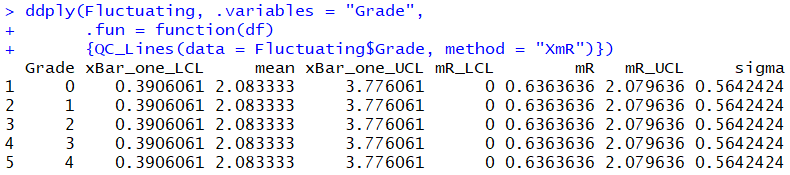
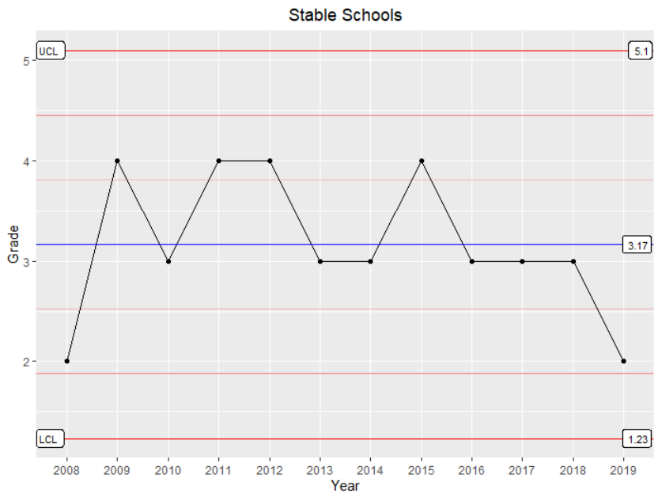
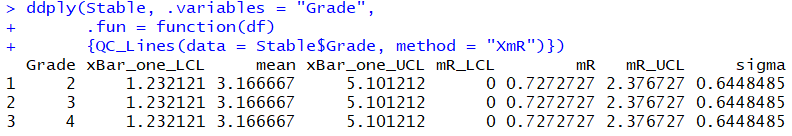
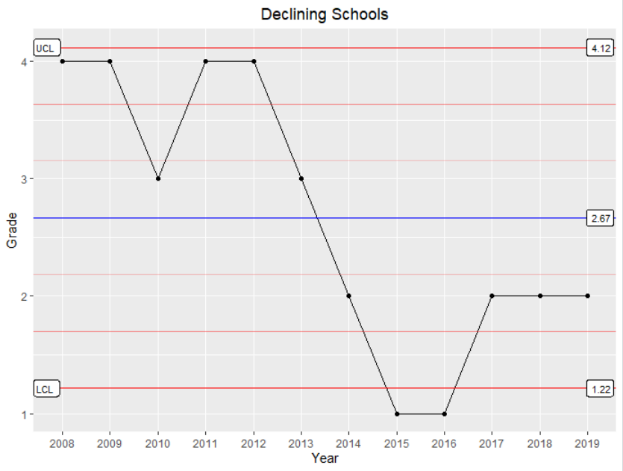
**Results**



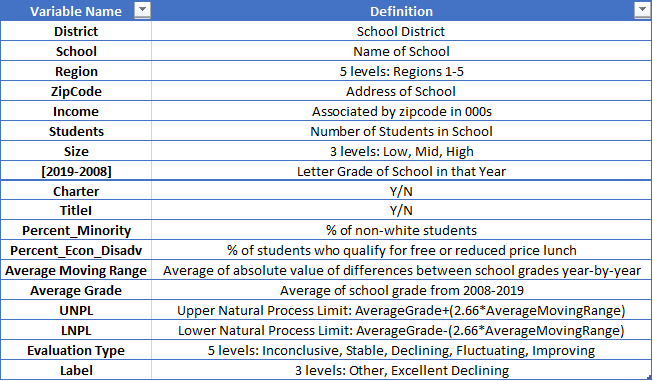
Most schools are stable, which allows us to have confidence in our identification method. However, only 3 schools were identified as improving while 124 schools were identified as declining.​ To proceed with our analysis, we decided to identify Excellent schools, which are schools that have received only A grades in the past 12 years. There are 164 such schools so this set will be used for comparison with declining schools.​

## Statistical Process Control: XmR charts

Upon visual inspection of XmR charts, we can have more confidence in our application of the method as these renderings are in line with the expected output (e.g. stable schools staying within upper and lower control limits and fluctuating schools breaking both limits).

* + Improving Schools
    - 
    - 
  + Fluctuating Schools
    - 
    - 
  + Stable Schools
    - 
    - 
  + Declining Schools​
    - 
    - 

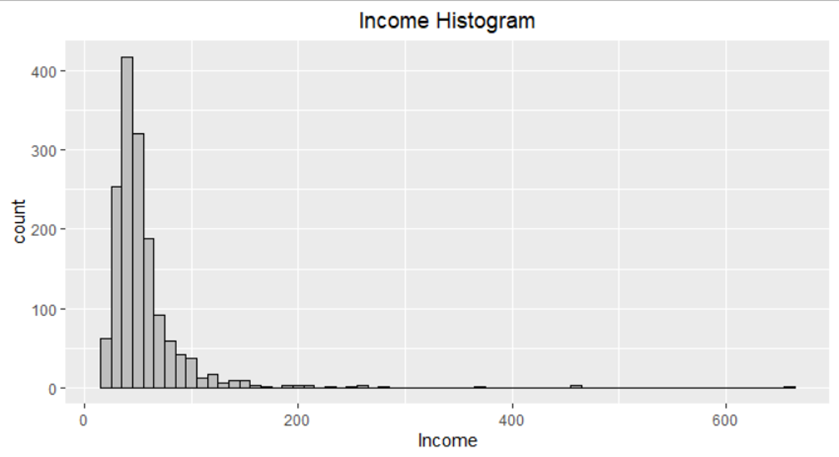
We end with the following Data Dictionary:



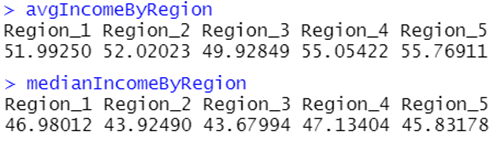
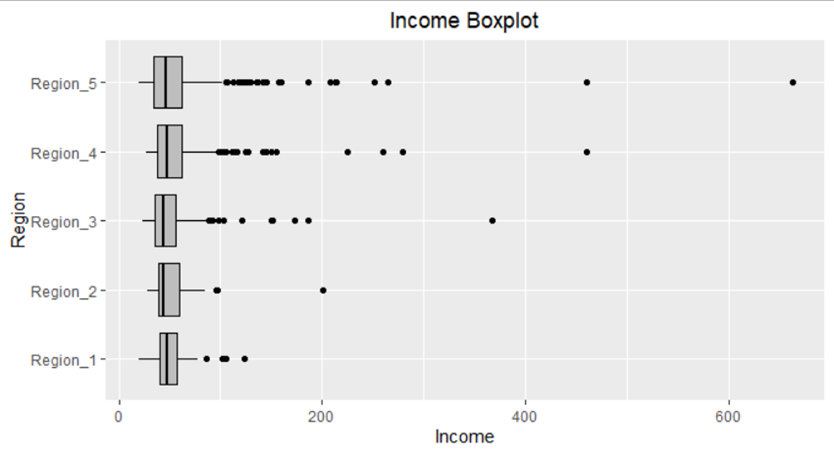
# **Techniques Used for Analysis and Results**

## 1): Descriptive Statistics

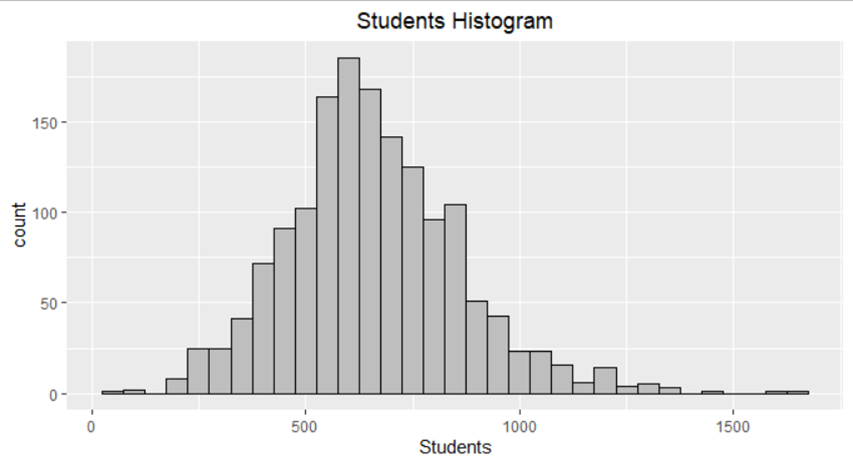
* Average Income: $54.0k
  + Histogram of Income – The distribution of income is skewed to the right. Most schools have an average adjusted gross income less than $100k; however, there are several region 4 and 5 schools in coastal cities (such as Palm Beach and Naples) with extremely high incomes.



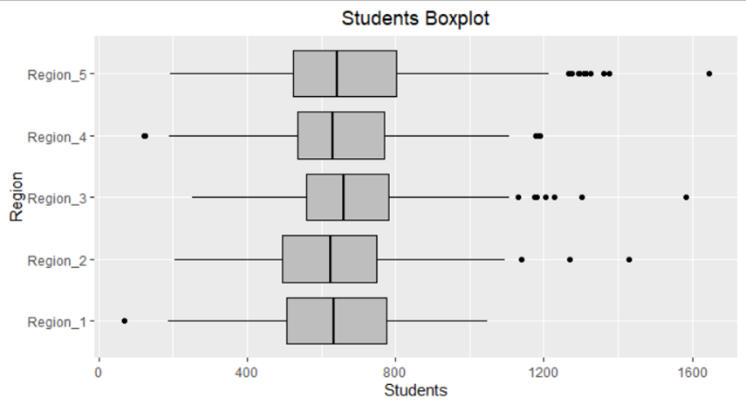
* Average Income by Region:
  + Boxplots – By running boxplots for each region, we are better able to see the distribution of income is concentrated in the inter-quartile range with some significant outliers, especially in Region 5 and Region 4.

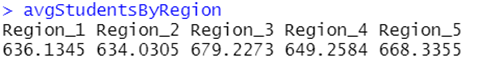


* Average # of Students: 659
  + Histogram of Students – The distribution of students normally distributed with most schools in the 500 to 800 range.

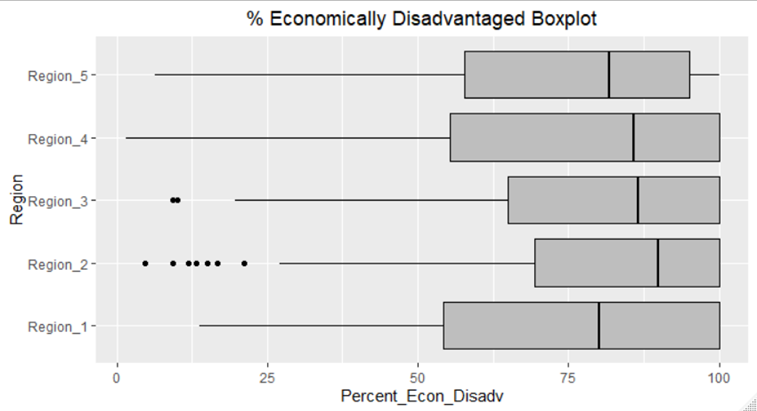


* Average # of Students by Region:
  + Boxplots – There are not major differences in the number of students per school between the regions. Region 5 and Region 4 are slightly more right-skewed, driven by some outliers. Region 3 has the highest average number of students with 679.



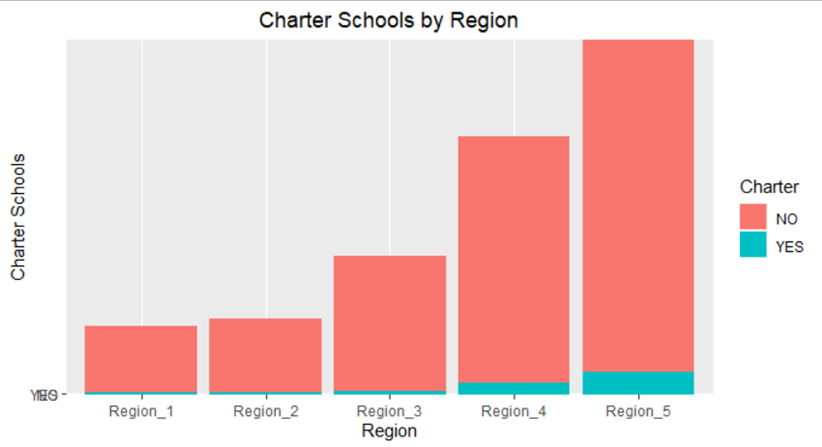


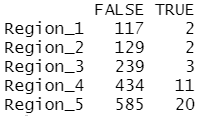
* Average % Economically Disadvantaged by Region:
  + Boxplots – Regions means are similar (~74%-79% on average); however, the inter-quartile range in Region 2 and Region 3 is more concentrated at the higher percentages (above 60%). In these two regions, it is an outlier for the % economically disadvantaged to be less than 25%, indicating that a lower percentage is only among a handful of schools.





* Count of charter schools by region
  + In the final dataset used for analysis, there were only a couple charter schools in Region 1, 2, and 3. Region 4 had 11 and Region 5 had 20 charter schools.

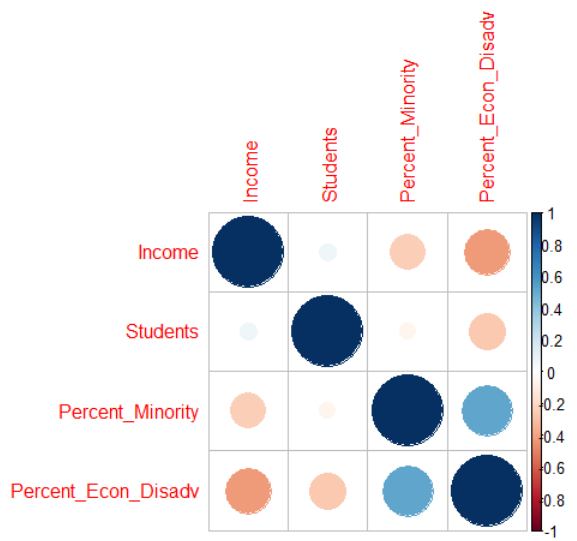




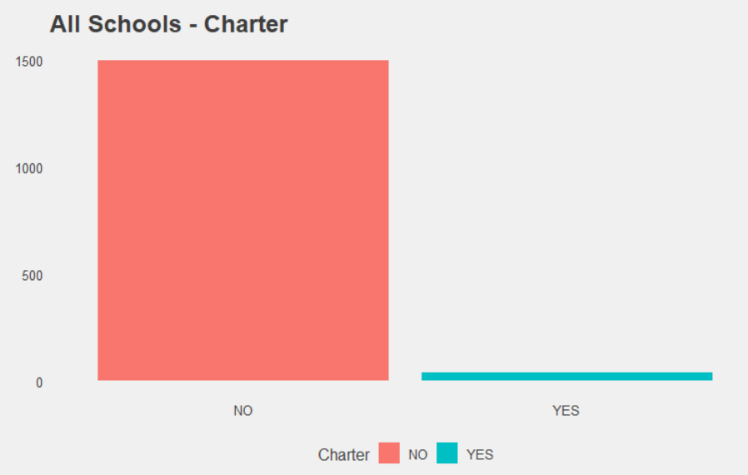
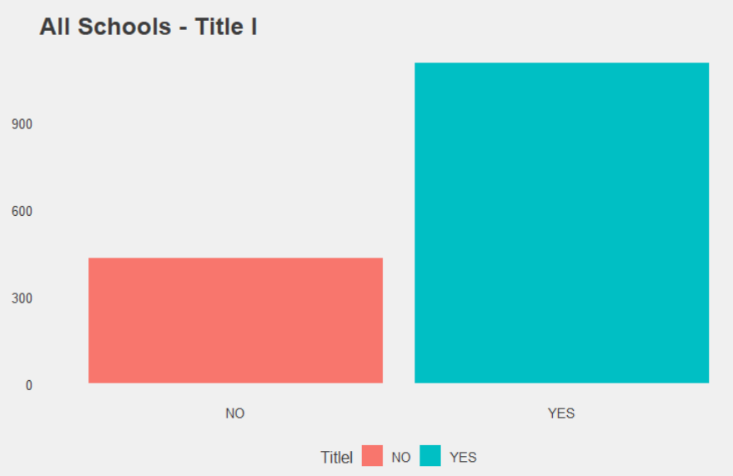
## 2): General and Map Visualizations

The following visualizations were created to get a better understanding of our dataset, address certain business questions, and to provide visual representations of subsets of dataset.

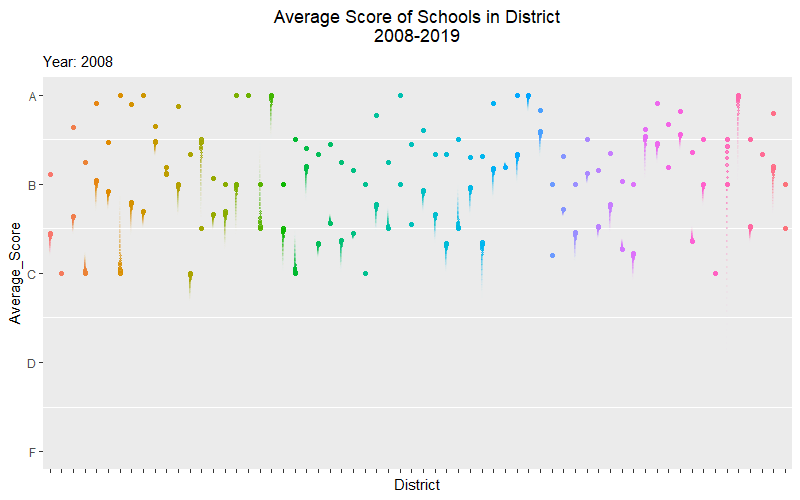
The correlation matrix below shows expected correlations such as higher Income correlating with lower % Economically Disadvantaged. However, solving for multicollinearity with tools like principal component analysis is beyond the scope of this analysis. We can note, though, that there’s a likelihood that if one variable is statistically significant while it’s highly correlated pairing with another variable is not, that there’s more to explore in what makes that statistically significant variable stands out.



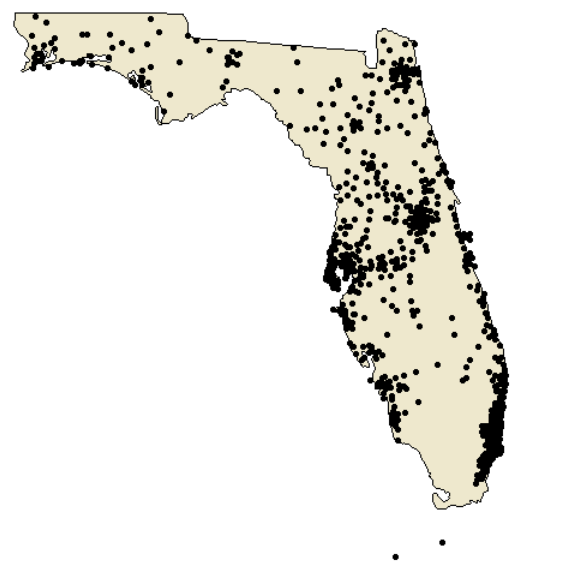
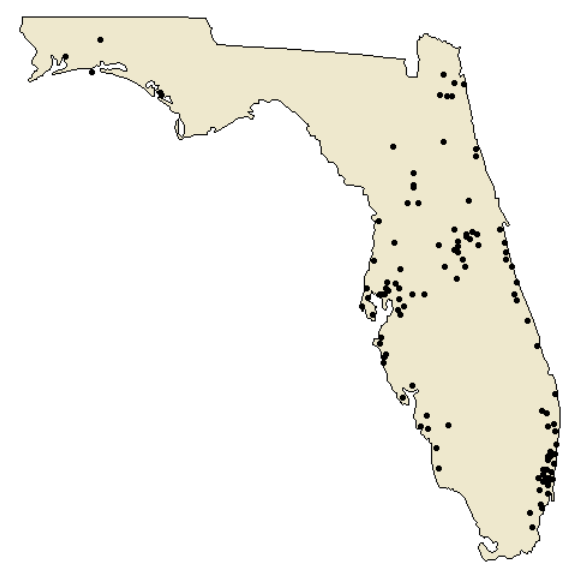
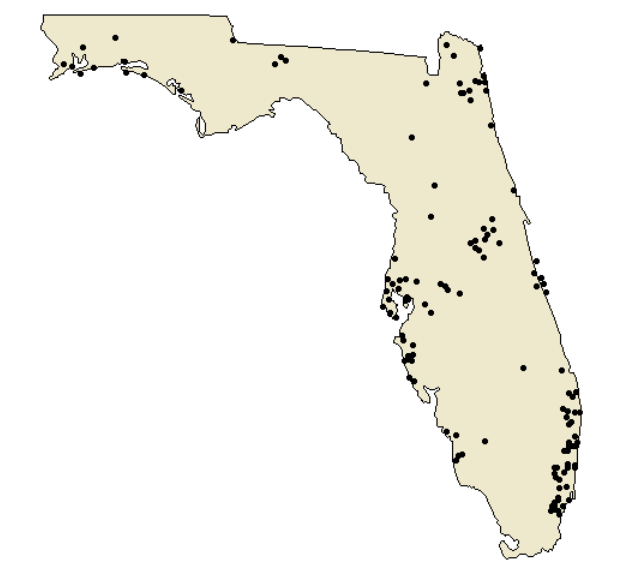
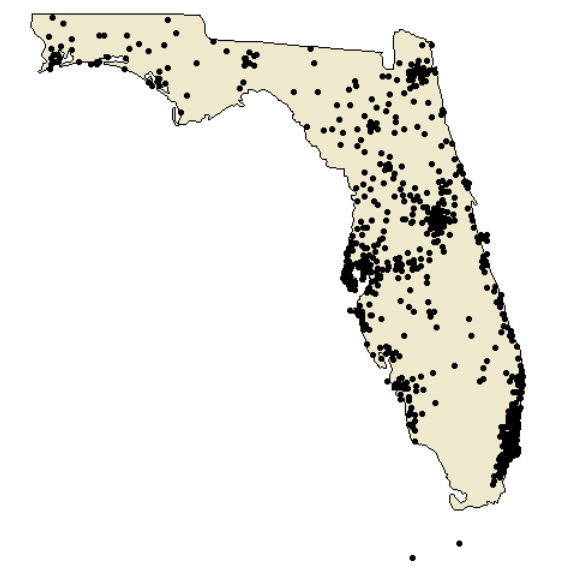
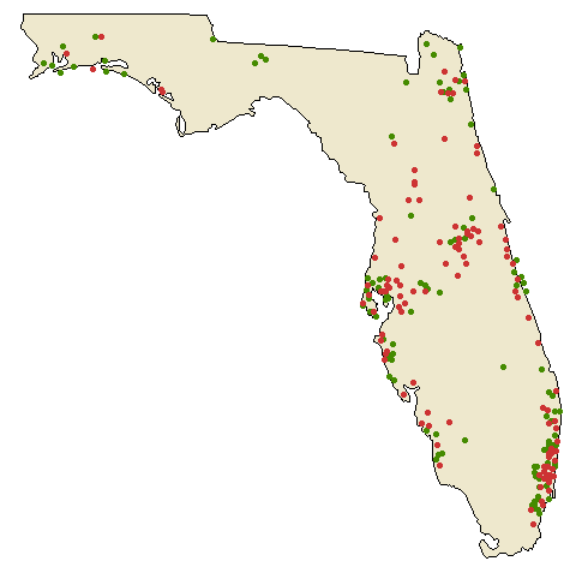
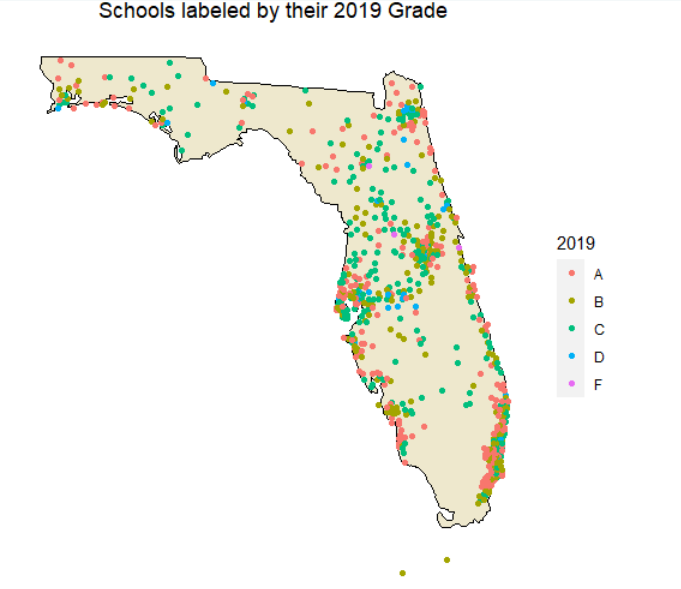
The bar charts below were used specifically to visualize any unbalanced issues with our binary variables requiring consideration when using machine learning models. From this, we knew to closely monitor the Charter variable.



The animated gif below (of which a larger version appears in the accompanying power point) helps to emphasize how grades fluctuate over time. Statistical process controls will allow us to differentiate between signal and noise.



The map visualizations below were helpful for us to see the location of schools participating in state standardized testing, as well as the location of schools which were categorized as excellent versus those that were categorized as declining. This method allows us to envision the urgency of the situation. At this level, there were no visibly evident patterns, such as clusters of declining schools along a coastline.

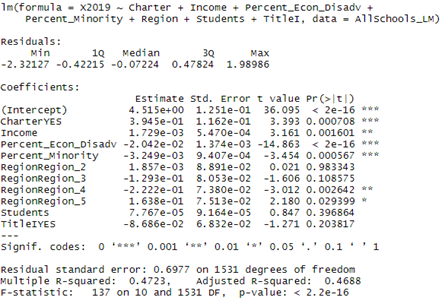
* Base Map of Florida Map of Florida Schools within our final dataset used for analysis
  + 
* Map of Schools that we identified as “Excellent” Map of Schools that we identified as “Declining”
  + 
* On the left: Plot points on the "FLbase" map for the schools being assessed where green represents Excellent and red represents Declining. On the right: Plot points on the "FLbase" map of all of the schools that were neither Excellent nor Declining
  + 
* Map of Florida Schools by their 2019 Grade
  + 

## 3): Linear Model and Results

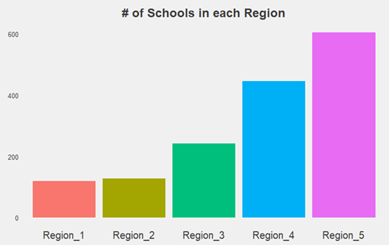
In order to optimize Florida’s Student Performance Support Budget, our team needs to pinpoint the driving factors for school grades. This will allow funds to be distributed to initiatives with the highest yields.

In our linear models that follow, % Economically Disadvantaged has the most explanatory power for predicting 2019 grade.

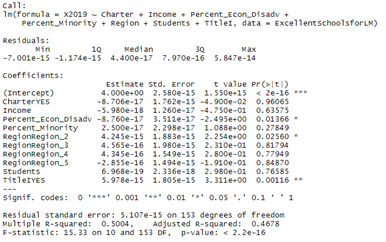
Linear Model to Predict Grade in 2019 for All Schools



* Adjusted R^2 in overall linear model: 0.4688
* The significant variables in the linear model include:
  + Charter – although a low % of schools overall; they have a positive effect on grades (p-value: 0.0007)
  + Income – higher income has positive effect on grades (p-value: 0.0016)
  + % Economically Disadvantaged – strong negative effect on grades (p-value: 0.0000)
  + % Minority - strong negative effect on grades (p-value: 0.0006)
  + Region 4 – negative relationship with grades in Region 4 (p-value: 0.0026)
  + Region 5 – positive relationship with grades in Region 5 (p-value: 0.0294)
  + Region variables may be significant due to their distribution as seen below:

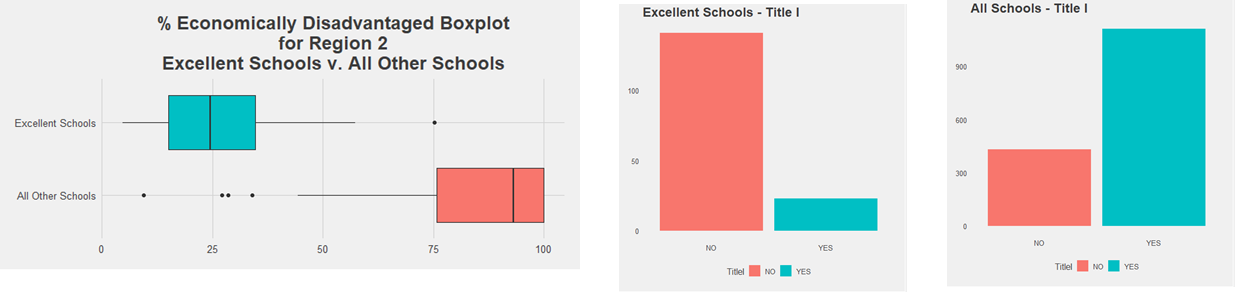


Linear Model to Predict Grade in 2019 for Excellent Schools



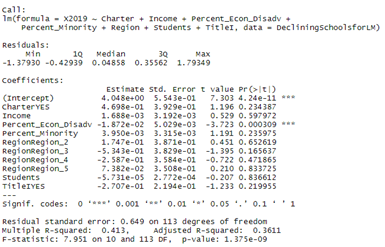
* Adjusted R^2 in Excellent Schools linear model: 0.4678
* The significant variables in the Excellent Schools linear model include:
  + % Economically Disadvantaged – larger negative coefficient (p-value: 0.0137)
  + Region 2 – positive coefficient (p-value: 0.0256)
  + Title 1 – positive coefficient (p-value: 0.0012)
* When we ran the Excellent Schools linear model again with only % Economically Disadvantaged as the independent variable the adjusted R^2 increases to 0.4947. However, while that model overall is still statistically significant, %Economically Disadvantaged as a variable was not significant. Adding back Title I as a variable resulted in similar values, with the difference being that both independent variables used were statistically significant.

Deeper Dive into the 3 significant variables for the linear model of Excellent Schools



* Excellent Schools in Region 2 have a lower portion of economically disadvantaged students than all other schools in Region 2
* Title 1 schools serve a significant number of economically disadvantaged students and therefore receive additional federal funds to support students and their families (it appears that this extra support is not enough to narrow the performance gap).

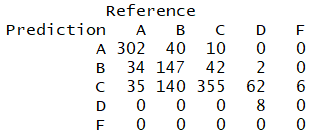
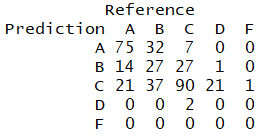
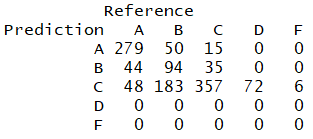
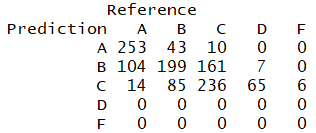
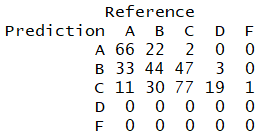
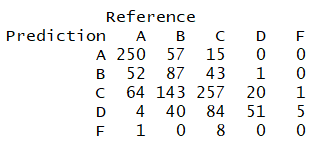
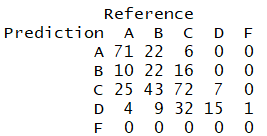
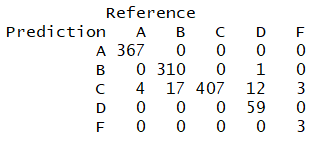
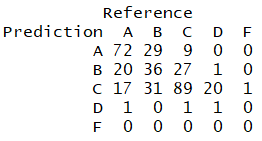
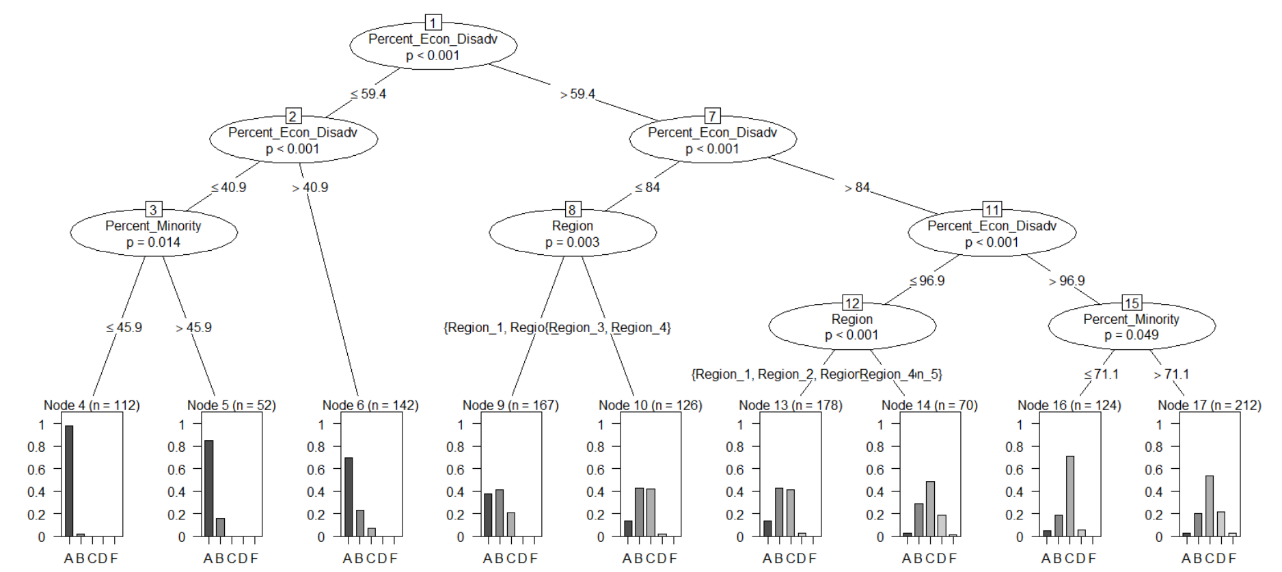
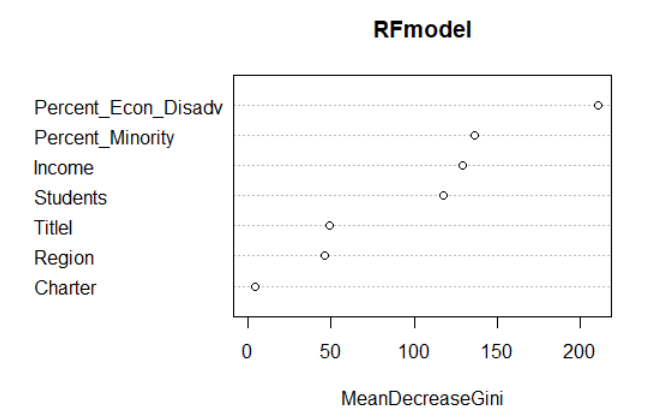
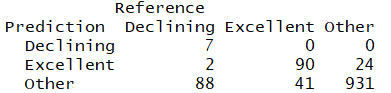
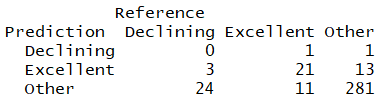
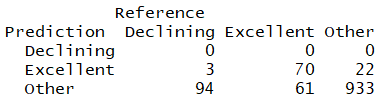
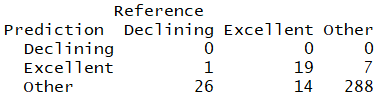
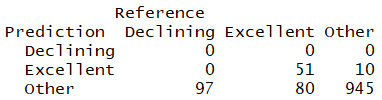
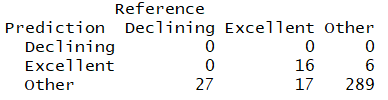
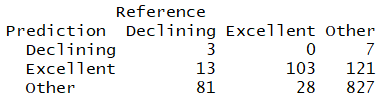
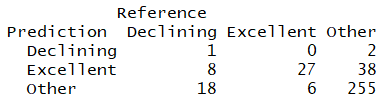
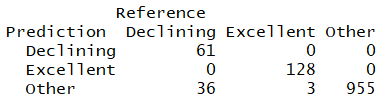
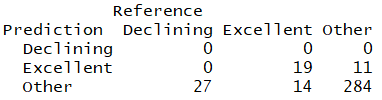
Linear Model to Predict Grade in 2019 for Declining Schools

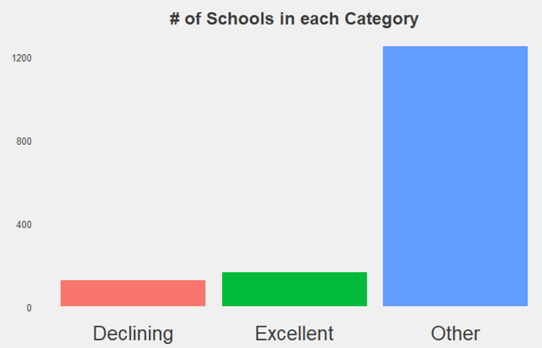


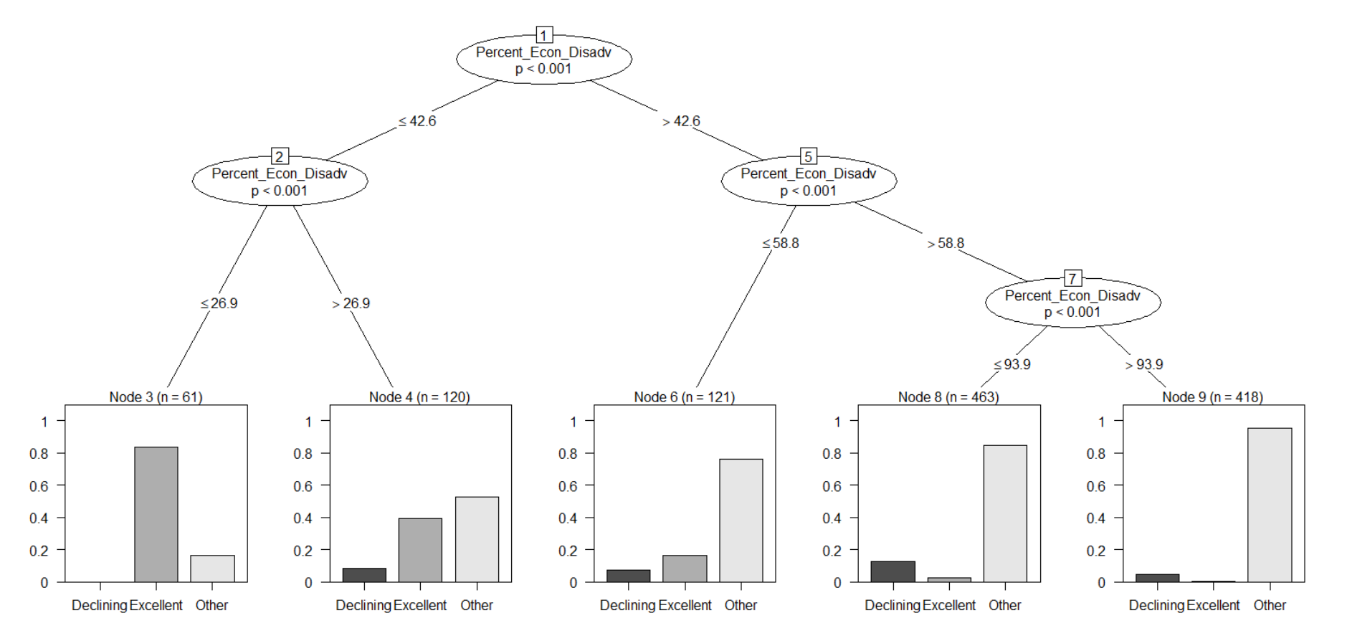
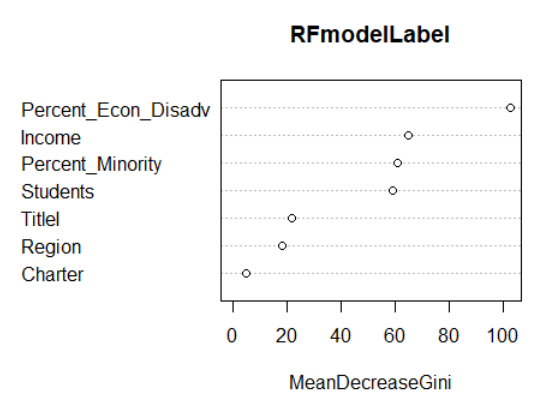
* Adjusted R^2 in Declining Schools linear model: 0.3611
* The significant variables in the Declining schools linear model include:
  + % Economically Disadvantaged – negative coefficient (p-value: 0.0003)
* When we ran the Declining schools linear model again with only % Economically Disadvantaged as the independent variable the adjusted R^2 decreases to 0.3248.

## 4): Machine Learning Classification and Results

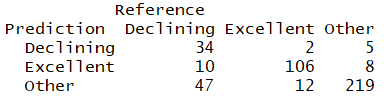
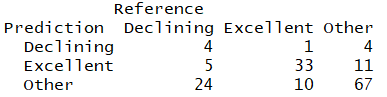
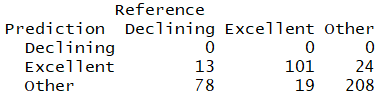
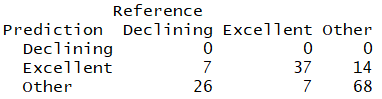
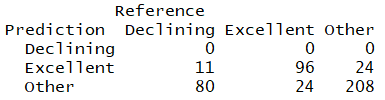
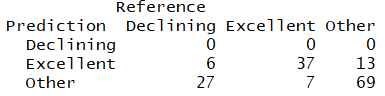
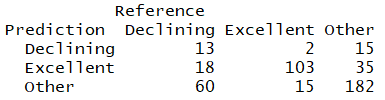
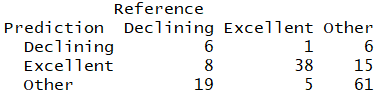
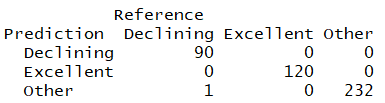
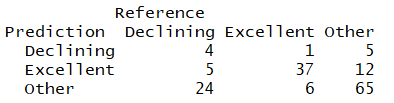
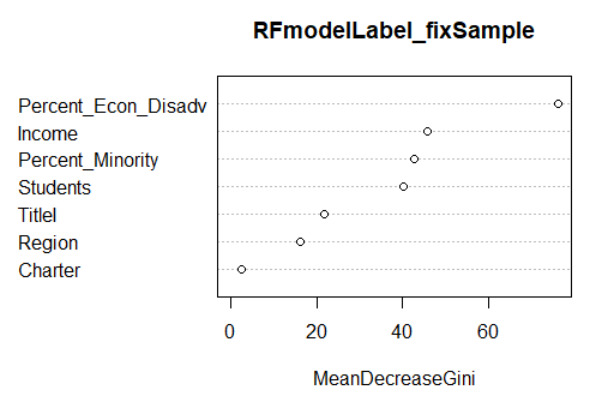
For the following machine learning models, we used 75% of our data for training, and reserved 25% of our data for testing.

* Machine Learning to Predict Grades
  + KSVM
    - Train Accuracy
      * 
      * 68.64% Accuracy
    - Test Accuracy
      * 
      * 54.08% Accuracy
  + SVM
    - Train Accuracy
      * 
      * 61.71% Accuracy
    - Test Accuracy
      * 
      * 56.06% Accuracy
  + Decision Tree
    - Train Accuracy
      * 
      * 58.16% Accuracy
    - Test Accuracy
      * 
      * 52.68% Accuracy
  + NaiveBayes
    - Train Accuracy
      * 
      * 54.52% Accuracy
    - Test Accuracy
      * 
      * 50.70% Accuracy
  + Random Forest
    - Train Accuracy
      * 
      * 96.87% Accuracy
    - Test Accuracy
      * 
      * 55.77% Accuracy
  + Findings
    - We would like to affirm our findings from our regression analysis by using other prediction models. ​However, we do acknowledge that some of these methods are like a “black box” and are not as interpretable as linear modeling. ​In addition, these can be re-run and produce different results due to randomness. The PowerPoint that accompanies this document has results from a different run on purpose to illustrate this expected difference. In addition, the general trends discussed here are also relatively reflected by the PowerPoint results.
* Machine Learning Visualizations for Predicting Grades
  + Decision Tree
    - 
  + Random Forest (Importance of Variables)
    - 
  + Findings
    - Fortunately, Decision Trees can be easily visualized and by following the path of variables, we can determine which variables have greater impact on predicting grades.​ In addition, the Random Forest model (which is the combination of multiple randomly created Decision Trees) has a visual plot in R that shows the relative importance of each variable to the prediction. ​Both these visualizations confirm % Economically Disadvantaged students as the most influential variable. Note, how when scanning the decision tree from left to right, the histograms change shape with lower %Economically Disadvantaged having higher grades.
* Machine Learning to Predict Label (Excellent, Declining, Other)
  + KSVM
    - Train Accuracy
      * 
      * 86.90% Accuracy
    - Test Accuracy
      * 
      * 85.07% Accuracy
  + SVM
    - Train Accuracy
      * 
      * 84.78% Accuracy
    - Test Accuracy
      * 
      * 86.48% Accuracy
  + Decision Tree
    - Train Accuracy
      * 
      * 84.19% Accuracy
    - Test Accuracy
      * 
      * 85.92% Accuracy
  + NaïveBayes
    - Train Accuracy
      * 
      * 78.87% Accuracy
    - Test Accuracy
      * 
      * 79.72% Accuracy
  + Random Forest
    - Train Accuracy
      * 
      * 96.70% Accuracy
    - Test Accuracy
      * 
      * 85.35% Accuracy
  + Findings
    - Upon closer observation of the confusion matrices, the models’ accuracy which hovers around 80% may be driven by the fact that ~ 81% of the data belongs in one class - “Other” as illustrated by the bar chart below.



* Machine Learning Visualizations for Predicting Label (Excellent, Declining, Other)
  + Decision Tree
    - 
  + Random Forest (Importance of Variables)
    - 
  + Findings
    - Once again, % Economically Disadvantaged appears to be the most influential variable for prediction. ​In contrast to the Decision Tree in the PowerPoint (which uses 2 other variables: Charter and %Minority for nodes with n totaling to 40 ~ 3.5% of training data), the one above only uses % Economically Disadvantaged
* Machine Learning to Predict Label – Undersample

With 124 Declining schools & 164 Excellent schools, we decided to randomly sample only 25% of all Other schools for a pool of 315 schools in order to address the imbalance of these classes.

* + KSVM
    - Train Accuracy – FixSample
      * 
      * 81.04% Accuracy
    - Test Accuracy – FixSample
      * 
      * 65.41% Accuracy
  + SVM
    - Train Accuracy – FixSample
      * 
      * 69.75% Accuracy
    - Test Accuracy – FixSample
      * 
      * 66.04% Accuracy
  + Decision Tree
    - Train Accuracy – FixSample
      * 
      * 68.62% Accuracy
    - Test Accuracy – FixSample
      * 
      * 66.67% Accuracy
  + NaiveBayes
    - Train Accuracy – FixSample
      * 
      * 67.27% Accuracy
    - Test Accuracy – FixSample
      * 
      * 66.04% Accuracy
  + Random Forest
    - Train Accuracy – FixSample
      * 
      * 99.77% Accuracy
    - Test Accuracy – FixSample
      * 
      * 66.67% Accuracy
  + Findings
    - ​While overall, the models were less accurate compared to when we did not undersample, they did not simply predict “Other” for everything. However, these models suggest that we should investigate other variables to include as the explanatory power of our current variables of interest are not sufficient especially for Declining schools. The models perform better in predicting for Excellent schools. ​
* Machine Learning Visualizations for Predictive Label – Undersample
  + Decision Tree
    - 
  + Random Forest (Importance of Variables)
    - 
  + Findings
    - As the trend continues, % Economically Disadvantaged appears to be the most influential variable for prediction. ​Note again how Excellent Schools decrease from left to right on the bar charts on the Decision Tree.

# **Business Insights and Conclusion**

Theteam’s analysis suggests a consistent theme throughout. Our models build a narrative suggesting that the most consistent factor for predicting student success is the percent of student population that is classified as economically disadvantaged (our % Economically Disadvantaged variable). Recall that these are students determined to be eligible for free and reduced-price meals under the National School Lunch Program, and their struggles with academic performance are well-known in academic literature which concludes that “Economically disadvantaged children enter school with less developed cognitive skills than their peers.” [2]

Students eligible for this category face obstacles to learning that their middle- or upper-class peers may not experience. Economically disadvantaged students could suffer from limited parent engagement, hunger, homelessness, and more. Learning disabilities that are often hereditary or resulting from malnutrition, extreme lack of sleep, or a host of other factors, can greatly hinder students’ opportunity for academic achievement. Our analyses above could also be boiled down to the simple visual comparison below:



Although our team recognizes that the state of Florida currently allots the majority of its Student Improvement Support Budget to support underperforming schools, to address this problem, it may be necessary for the state to focus on districts with high percentages of economically disadvantaged students and redistribute funds to give more support to these regions. It may also benefit schools to address this factor by reflecting upon whether maximum utilization of available resources is currently implemented. However, we do note that Title I schools which receive additional federal funds to solve for these inequities do not seem to be performing significantly better despite additional support according to our analysis (for example, it wasn’t a significant contributor to decision trees). Our analyses suggests that schools likely need additional federal, state, and district support, should experiment with more novel solutions using current resources available, and may benefit from more research into learning and development in order to accelerate student learning and reduce this proficiency gap.

Possible insights/questions for thought:

\*Are Districts/schools/teachers engaging their communities for support with extra food for when students are not in school? Student clothing?

\*Could business partnerships be formed for medical services for students?

\*Could community churches be solicited to aid students with homework support, or other tutoring services, free of charge to families?

\*How could schools/districts support parents who do not speak English so they could be more involved in their children’s education?

\*In general, what could districts/schools do support parent engagement with their children?

\*At the state and district level, are the amount of funds spent on economically disadvantaged students proportionate to the magnitude of correlation seen in our analyses?

\*Is there a more efficient way to utilize district and state personnel (instructional coaches, district specialists, school support specialists at the state level) to address the obstacles that economically disadvantaged students face?

# **References**

**[1]** Florida Department of Education. (2008-2019). *Florida Standards Assessments Results*.

**[2]** Crosnoe, Robert , and Cooper Carey E.. 2010. “Economically Disadvantaged Children's Transitions Into Elementary School: Linking Family Processes, School Contexts, and Educational Policy.” American Educational Research Journal 47 (2): 258–91. 10.3102/0002831209351564.

# **Appendix – R Code**

* Unexecuted Code

# CREATE BASE DATASET ####

install.packages("ggplot2")

install.packages("ggQC")

install.packages("plyr")

install.packages("gridExtra")

install.packages("tidyverse")

install.packages("readxl")

install.packages("kernlab")

install.packages("caret")

install.packages("e1071")

install.packages("party")

install.packages("randomForest")

install.packages("corrplot")

install.packages("ggthemes")

install.packages("gganimate")

install.packages("gifski")

install.packages("reshape2")

install.packages("devtools")

install.packages("dplyr")

install.packages("gdata")

install.packages("ggmap")

install.packages("mapdata")

install.packages("maps")

install.packages("openintro")

install.packages("sqldf")

install.packages("stringr")

install.packages("zipcode")

library(plyr)

library(tidyverse)

library(ggQC)

library(gridExtra)

library(readxl)

library(kernlab)

library(caret)

library(e1071)

library(party)

library(randomForest)

library(corrplot)

library(ggthemes)

library(gganimate)

library(gifski)

library(reshape2)

library(devtools)

library(ggmap)

library(mapdata)

library(maps)

library(openintro)

library(sqldf)

library(zipcode)

#Excel file should be in working directory, named as Data\_Raw.xlsx

#assumes only one sheet in Excel and there are no formulas, so copy-paste-as-values

#ignore Warning message for expecting logical since it's not a column we'll be using

Base <- read\_excel("Data\_Raw.xlsx",range = "A5:BA3342")

#Keep only columns of interest, check structure

Base <- Base[,c(2,4,5,6,7,8,9,25,26,27,28,29,30,31,32,33,34,35,36,48,49,51,52,53)]

str(Base)

#Change column names

colnames(Base) <- c("District","School","Region","ZipCode","Income","Students","Size","2019","2018","2017","2016","2015","2014","2013","2012","2011","2010","2009","2008","Charter","TitleI","Type","Percent\_Minority","Percent\_Econ\_Disadv")

#Select for only elementary schools, and then remove that column

Base <- Base %>%

filter(Type =="01") %>%

select(-contains("Type"))

#Use only complete cases

Base <- Base %>%

filter(complete.cases(.))

# CREATE PERFORMANCE SET FOR EVALUATING SCHOOLS ####

#function to convert letter grades into numbers (works specifically with Base, note year\_index as column numbers)

recode\_col <- function(df){

copydf <- df

year\_index <- 8:19

for (i in year\_index){

copydf[i] <- recode(pull(df,i),"A"=4,"B"=3,"C"=2,"D"=1,"F"=0)

}

return (copydf)

}

Performance <- recode\_col(Base)

str(Performance)

#Average Moving Range function (not generic, works just for our dataset)

AverageMovingRange <- function(df){

difference <- function(vector){

vector <- as.numeric(vector)

index <- 1:length(vector)

differences <- c()

for (i in index){

if (i==length(vector)){

break

}

else {

differences[i] <- abs(vector[i]-vector[i+1])

}

}

return (differences)

}

copydf <- df

row\_index <- 1:nrow(df)

movingRangeVector <- c()

for (i in row\_index){

currentrow <- df[i,][8:19]

movingRangeVector[i] <- mean(difference(currentrow))

}

copydf$AverageMovingRange <- movingRangeVector

return (copydf)

}

Performance <- AverageMovingRange(Performance)

#Average of Grades function (again, function specific to our dataset)

AverageofGrades <- function(df){

copydf <- df

row\_index <- 1:nrow(df)

AverageGradesVector <- c()

for (i in row\_index){

AverageGradesVector[i] <- mean(as.numeric(copydf[i,][8:19]))

}

copydf$AverageGrade <- AverageGradesVector

return(copydf)

}

Performance <- AverageofGrades(Performance)

#Make columns for Upper and Lower Natural Process Limit

Performance <- Performance %>%

mutate(UNPL=(AverageGrade+(2.66\*AverageMovingRange))) %>%

mutate(LNPL=(AverageGrade-(2.66\*AverageMovingRange)))

#Function that evaluates school's performance

Evaluation <- function(df){

copydf <- df

row\_index <- 1:nrow(df)

EvaluationVector <- c()

for (i in row\_index){

LNPLbreaks <- 0

UNPLbreaks <- 0

year\_index <- 8:12

for (j in year\_index){

if (df[i,][j] > df[i,]$UNPL) {

UNPLbreaks <- UNPLbreaks+1

}

else if (df[i,][j] < df[i,]$LNPL) {

LNPLbreaks <- LNPLbreaks+1

}

}

EvaluationScore <- case\_when(

(LNPLbreaks == 0 and UNPLbreaks == 0) ~ "Stable",

(

(LNPLbreaks == 2 and UNPLbreaks == 0) |

(LNPLbreaks == 3 and UNPLbreaks == 0) |

(LNPLbreaks == 4 and UNPLbreaks == 0) |

(LNPLbreaks == 5 and UNPLbreaks == 0)

) ~ "Declining",

(

(LNPLbreaks == 0 and UNPLbreaks == 2) |

(LNPLbreaks == 0 and UNPLbreaks == 3) |

(LNPLbreaks == 0 and UNPLbreaks == 4) |

(LNPLbreaks == 0 and UNPLbreaks == 5)

) ~ "Improving",

(

(LNPLbreaks == 1 and UNPLbreaks == 0) |

(LNPLbreaks == 0 and UNPLbreaks == 1)

) ~ "Inconclusive",

TRUE ~ "Fluctuating"

)

EvaluationVector[i] <- EvaluationScore

}

copydf$EvaluationType <- EvaluationVector

return(copydf)

}

Performance <- Evaluation(Performance)

#View how many schools fall into what category

EvaluationTab <- Performance %>%

group\_by(EvaluationType) %>%

summarise(Number\_of\_Schools = n()) %>%

arrange(desc(Number\_of\_Schools))

#Create new column called "Label" to be used for prediction along with 2019 Grades

Performance <- Performance %>%

mutate(Label=case\_when(

EvaluationType=="Declining" ~ "Declining",

AverageGrade==4 ~ "Excellent",

TRUE ~ "Other"))

#XMR Chart####

Improving <- Performance %>%

filter(EvaluationType=="Improving") %>%

slice(3) %>%

select(starts\_with("20")) %>%

gather(key="Year",value="Grade") %>%

arrange(Year)

(imp\_xmr\_plot <-

ggplot(Improving, aes(x = Year, y = Grade,group=1)) +

geom\_point() + geom\_line() + ggtitle("Improving Schools") +

theme(plot.title = element\_text(hjust = 0.5)) +

stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))

Fluctuating <- Performance %>%

filter(EvaluationType=="Fluctuating") %>%

slice(2) %>%

select(starts\_with("20")) %>%

gather(key="Year",value="Grade") %>%

arrange(Year)

(fluc\_xmr\_plot <- ggplot(Fluctuating, aes(x = Year, y = Grade,group=1)) +

geom\_point() + geom\_line() + ggtitle("Fluctuating Schools") +

theme(plot.title = element\_text(hjust = 0.5)) +

stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))

Stable <- Performance %>%

filter(EvaluationType=="Stable") %>%

slice(4) %>%

select(starts\_with("20")) %>%

gather(key="Year",value="Grade") %>%

arrange(Year)

(stab\_xmr\_plot <- ggplot(Stable, aes(x = Year, y = Grade,group=1)) +

geom\_point() + geom\_line() + ggtitle("Stable Schools") +

theme(plot.title = element\_text(hjust = 0.5)) +

stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))

Declining <- Performance %>%

filter(EvaluationType=="Declining") %>%

slice(2) %>%

select(starts\_with("20")) %>%

gather(key="Year",value="Grade") %>%

arrange(Year)

(decl\_xmr\_plot <- ggplot(Declining, aes(x = Year, y = Grade,group=1)) +

geom\_point() + geom\_line() + ggtitle("Declining Schools") +

theme(plot.title = element\_text(hjust = 0.5)) +

stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))

grid.arrange(imp\_xmr\_plot, fluc\_xmr\_plot, stab\_xmr\_plot, decl\_xmr\_plot, nrow = 2)

ddply(Improving, .variables = "Grade",

.fun = function(df)

{QC\_Lines(data = Improving$Grade, method = "XmR")})

ddply(Fluctuating, .variables = "Grade",

.fun = function(df)

{QC\_Lines(data = Fluctuating$Grade, method = "XmR")})

ddply(Stable, .variables = "Grade",

.fun = function(df)

{QC\_Lines(data = Stable$Grade, method = "XmR")})

ddply(Declining, .variables = "Grade",

.fun = function(df)

{QC\_Lines(data = Declining$Grade, method = "XmR")})

#LINEAR MODELS FOR CLUSTERS ####

#Isolate 3 clusters into different datasets

ExcellentSchools <- Performance %>%

filter(AverageGrade==4)

DecliningSchools <- Performance %>%

filter(EvaluationType=="Declining")

AllOtherSchools <- Performance %>%

filter(!(AverageGrade==4 | EvaluationType=="Declining"))

#Clean datasets to retain only columns to be used for linear models

CleanforLM <- function(df){

removethiscolumns <- c(1:2,4,7,9:19,24:29)

df <- df[,-removethiscolumns]

return(df)

}

ExcellentSchoolsforLM <- CleanforLM(ExcellentSchools)

DecliningSchoolsforLM <- CleanforLM(DecliningSchools)

AllOtherSchoolsforLM <- CleanforLM(AllOtherSchools)

#create linear models

names(ExcellentSchoolsforLM) <- make.names(names(ExcellentSchoolsforLM))

ExcellentSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv +

Percent\_Minority + Region + Students + TitleI, data=ExcellentSchoolsforLM)

summary(ExcellentSchools\_LinearModel)

names(DecliningSchoolsforLM) <- make.names(names(DecliningSchoolsforLM))

DecliningSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv +

Percent\_Minority + Region + Students + TitleI, data=DecliningSchoolsforLM)

summary(DecliningSchools\_LinearModel)

names(AllOtherSchoolsforLM) <- make.names(names(AllOtherSchoolsforLM))

AllOtherSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv +

Percent\_Minority + Region + Students + TitleI, data=AllOtherSchoolsforLM)

summary(AllOtherSchools\_LinearModel)

#Linear Model for all Schools

AllSchools\_LM <- CleanforLM(Performance)

names(AllSchools\_LM) <- make.names(names(AllSchools\_LM))

AllSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv + Percent\_Minority + Region + Students + TitleI, data=AllSchools\_LM)

summary(AllSchools\_LinearModel)

#Models for 3 clusters only using significant variables from above results (except Region since it's a 5-level factor)

ExcellentSchools\_LinearModel\_sigonly <- lm(X2019 ~ Percent\_Econ\_Disadv + TitleI, data=ExcellentSchoolsforLM)

summary(ExcellentSchools\_LinearModel\_sigonly)

DecliningSchools\_LinearModel\_sigonly <- lm(X2019 ~ Percent\_Econ\_Disadv, data=DecliningSchoolsforLM)

summary(DecliningSchools\_LinearModel\_sigonly)

AllOtherSchools\_LinearModel\_sigonly <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv + Percent\_Minority, data=AllOtherSchoolsforLM)

summary(AllOtherSchools\_LinearModel\_sigonly)

#MACHINE LEARNING FOR PREDICTING GRADES (2019)####

MLData <- Base[,-c(1:2,4,7,9:19)]

colnames(MLData)[colnames(MLData) == "2019"] <- "Grade"

#Convert to Factor

MLData <- MLData %>%

mutate\_at(c(1,4:6),factor)

#Setup test and train data

set.seed(5678)

pd <- sample(2,nrow(MLData), replace = TRUE, prob = c(.75,.25))

train <- MLData[pd==1,]

test <- MLData[pd==2,]

#KSVM

ksvmresults <- ksvm(Grade~., data = train, kernel = "rbfdot",

kpar="automatic",C=10, cross=10, prob.model=TRUE)

ksvmresults

#Confusion Matrix for Train

confusionMatrix(predict(ksvmresults,train),train$Grade)

#Confusion Matrix for Test

confusionMatrix(predict(ksvmresults,test),test$Grade)

#SVM

svmresults <- svm(Grade~., data = train, kernel = "radial",

kpar="automatic",C=10, cross=10, prob.model=TRUE)

svmresults

#Confusion Matrix for Train

confusionMatrix(predict(svmresults,train),train$Grade)

#Confusion Matrix for Test

confusionMatrix(predict(svmresults,test),test$Grade)

#Decision Tree

tree <- ctree (Grade ~ .,train)

plot(tree)

#Confusion Matrix for Train

confusionMatrix(predict(tree,train),train$Grade)

#Confusion Matrix for Test

confusionMatrix(predict(tree,test),test$Grade)

#Naive Bayes

NBmodel <- naiveBayes(Grade~., data=train)

#Confusion Matrix for Train

confusionMatrix(predict(NBmodel,train),train$Grade)

#Confusion Matrix for Test

confusionMatrix(predict(NBmodel,test),test$Grade)

#Random Forest

set.seed(9876)

RFmodel <- randomForest(Grade~., data=train)

#Confusion matrix with train data

confusionMatrix(predict(RFmodel,train),train$Grade)

#Confusion matrix with test data

confusionMatrix(predict(RFmodel,test),test$Grade)

#Plot Variables of Importance

varImpPlot(RFmodel, sort = TRUE)

#MODELS TO PREDICT "LABEL"####

Label <- Performance %>%

select(Region, Income, Students, Charter, TitleI, Percent\_Minority,Percent\_Econ\_Disadv,Label) %>%

mutate\_at(c(1,4:5,8),factor)

#Setup test and train data

trainLabel <- Label[pd==1,]

testLabel <- Label[pd==2,]

#KSVM

ksvmLabel <- ksvm(Label~., data = trainLabel, kernel = "rbfdot",

kpar="automatic",C=10, cross=10, prob.model=TRUE)

ksvmLabel

#Confusion matrix with train data

confusionMatrix(predict(ksvmLabel,trainLabel),trainLabel$Label)

#Confusion matrix with test data

confusionMatrix(predict(ksvmLabel,testLabel),testLabel$Label)

#SVM

svmLabel <- svm(Label~., data = trainLabel, kernel = "radial",

kpar="automatic",C=5, cross=5, prob.model=TRUE)

svmLabel

#Confusion matrix with train data

confusionMatrix(predict(svmLabel,trainLabel),trainLabel$Label)

#Confusion matrix with test data

confusionMatrix(predict(svmLabel,testLabel),testLabel$Label)

#Decision Tree

treeLabel <- ctree (Label ~ .,trainLabel)

(plot(treeLabel))

#Confusion matrix with train data

confusionMatrix(predict(treeLabel,trainLabel),trainLabel$Label)

#Confusion matrix with test data

confusionMatrix(predict(treeLabel,testLabel),testLabel$Label)

#NaiveBayes

NBmodelLabel <- naiveBayes(Label~., data=trainLabel)

#Confusion matrix with train data

confusionMatrix(predict(NBmodelLabel,trainLabel),trainLabel$Label)

#Confusion matrix with test data

confusionMatrix(predict(NBmodelLabel,testLabel),testLabel$Label)

#Random Forest

set.seed(9943)

RFmodelLabel <- randomForest(Label~., data=trainLabel)

#Confusion matrix with train data)

confusionMatrix(predict(RFmodelLabel,trainLabel),trainLabel$Label)

#Confusion matrix with test data

confusionMatrix(predict(RFmodelLabel,testLabel),testLabel$Label)

#Plot Variables of Importance

varImpPlot(RFmodelLabel, sort = TRUE)

#UNDERSAMPLING FOR PREDICTING LABEL####

fixSample\_Label <- Label %>%

filter(Label=="Excellent" | Label =="Declining")

OtherOnly <- Label %>%

filter(Label=="Other")

set.seed(384)

pd\_Other <- sample(2,nrow(OtherOnly), replace = TRUE, prob = c(.75,.25))

OtherOnly <- OtherOnly[pd\_Other==2,]

fixSample\_Label <- rbind(fixSample\_Label,OtherOnly)

#Setup test and train data

pd\_fixSample <- sample(2,nrow(fixSample\_Label), replace = TRUE, prob = c(.75,.25))

trainLabel\_fixSample <- fixSample\_Label[pd\_fixSample==1,]

testLabel\_fixSample <- fixSample\_Label[pd\_fixSample==2,]

#KSVM

ksvmlabelunder <- ksvm(Label~., data = trainLabel\_fixSample, kernel = "rbfdot",

kpar="automatic",C=10, cross=10, prob.model=TRUE)

ksvmlabelunder

#Confusion Matrix for Train and Test Data

confusionMatrix(predict(ksvmlabelunder,trainLabel\_fixSample),trainLabel\_fixSample$Label)

confusionMatrix(predict(ksvmlabelunder,testLabel\_fixSample),testLabel\_fixSample$Label)

#SVM

svmlabelunder <- svm(Label~., data = trainLabel\_fixSample, kernel = "radial",

kpar="automatic",C=5, cross=5, prob.model=TRUE)

svmlabelunder

#Confusion Matrix for Train and Test Data

confusionMatrix(predict(svmlabelunder,trainLabel\_fixSample),trainLabel\_fixSample$Label)

confusionMatrix(predict(svmlabelunder,testLabel\_fixSample),testLabel\_fixSample$Label)

#Decision Tree

treeLabel\_fixSample <- ctree (Label ~ .,trainLabel\_fixSample)

(plot(treeLabel\_fixSample))

#Confusion Matrix for Train and Test Data

confusionMatrix(predict(treeLabel\_fixSample,trainLabel\_fixSample),trainLabel\_fixSample$Label)

confusionMatrix(predict(treeLabel\_fixSample,testLabel\_fixSample),testLabel\_fixSample$Label)

#NaiveBayes

NBmodelLabel\_fixSample <- naiveBayes(Label~., data=trainLabel\_fixSample)

#Confusion Matrix for Train and Test Data

confusionMatrix(predict(NBmodelLabel\_fixSample,trainLabel\_fixSample),trainLabel\_fixSample$Label)

confusionMatrix(predict(NBmodelLabel\_fixSample,testLabel\_fixSample),testLabel\_fixSample$Label)

#Random Forest

set.seed(9543)

RFmodelLabel\_fixSample <- randomForest(Label~., data=trainLabel\_fixSample)

#Confusion matrix with train data)

confusionMatrix(predict(RFmodelLabel\_fixSample,trainLabel\_fixSample),trainLabel\_fixSample$Label)

#Confusion matrix with test data

confusionMatrix(predict(RFmodelLabel\_fixSample,testLabel\_fixSample),testLabel\_fixSample$Label)

#Plot Variables of Importance

varImpPlot(RFmodelLabel\_fixSample, sort = TRUE)

#VISUALIZATIONS GENERAL####

#Piechart for business problem

myPalette <- c("cadetblue3","darkblue","darkseagreen","coral1")

PieChartLabels <- c("Building Capacity","Funds for Data-Driven Initiatives", "Performance Based Incentives", "Support for Underperforming Schools")

BudgetInDollars <- c(31219426, 25819420, 6500000, 123750850)

pie(BudgetInDollars , labels = PieChartLabels, border="white",col=myPalette,main="Student Performance\nSupport Budget\nTotal ~ $187M")

#Correlation Matrix for continuous variables

corrplot(cor(Base[,-c(1:4,7:21)]))

#Title I and Charter Schools viz

TitleI\_AllSchools <- ggplot(data=Base, aes(x=TitleI,fill=TitleI)) + geom\_bar() + ggtitle("All Schools - Title I") + theme\_fivethirtyeight()+ theme(panel.grid.major = element\_blank())

TitleI\_AllSchools

Charter\_AllSchools <- ggplot(data=Base, aes(x=Charter,fill=Charter)) + geom\_bar() + ggtitle("All Schools - Charter") + theme\_fivethirtyeight()+ theme(panel.grid.major = element\_blank())

Charter\_AllSchools

#Histogram of Percent Economically Disadvantaged for Excellent Schools

hist\_PerEconDis\_ExcellentSchools <- Performance %>%

filter(Label=="Excellent") %>%

ggplot(aes(x=Percent\_Econ\_Disadv)) + geom\_histogram(bins=12) + theme\_fivethirtyeight()+ theme(panel.grid.major = element\_blank(),axis.text.y=element\_blank()) + ggtitle("Histogram of % Economically Disadvantaged\n for Excellent Schools") + scale\_x\_continuous(labels = function(x) paste0(x, '%'))

hist\_PerEconDis\_ExcellentSchools

#Histogram of Percent Economically Disadvantaged for Declining Schools

hist\_PerEconDis\_DecliningSchools <- Performance %>%

filter(Label=="Declining") %>%

ggplot(aes(x=Percent\_Econ\_Disadv)) + geom\_histogram(bins=12) + theme\_fivethirtyeight()+ theme(panel.grid.major = element\_blank(),axis.text.y=element\_blank()) + ggtitle("Histogram of % Economically Disadvantaged\n for Declining Schools") + scale\_x\_continuous(labels = function(x) paste0(x, '%'))

hist\_PerEconDis\_DecliningSchools

#Bar graph showing distribution of Region

Region\_Viz <- ggplot(data=Base, aes(x=Region,fill=Region)) + geom\_bar() + theme\_fivethirtyeight() + theme(plot.title = element\_text(hjust=0.5),axis.text.x = element\_text(size = 15),legend.position = "none",panel.grid.major = element\_blank()) + ggtitle("# of Schools in each Region")

Region\_Viz

#Visualization for Title I schools in Excellent Schools

(TitleI\_Excellent <- ggplot(data=ExcellentSchools, aes(x=TitleI,fill=TitleI)) + geom\_bar() + ggtitle("Excellent Schools - Title I") + ylab("Title I") + theme\_fivethirtyeight() + theme(panel.grid.major = element\_blank()))

#Bar graph showing distribution of Excellent, Declining, Other

Label\_Viz <- ggplot(data=Performance, aes(x=Label,fill=Label)) + geom\_bar() + theme\_fivethirtyeight() + theme(plot.title = element\_text(hjust=0.5),axis.text.x = element\_text(size = 20),legend.position = "none",panel.grid.major = element\_blank()) + ggtitle("# of Schools in each Category")

Label\_Viz

#Animation of District Average Grades

#Prep data, use only Grades over years and District, average Grades by District

avgscoreDistrict <- Performance[,c(1,8:19)] %>%

group\_by(District) %>%

summarise(across(everything(),mean)) %>%

melt(id="District")

colnames(avgscoreDistrict) <- c("District","Year","Average\_Score")

avgscoreDistrict$Year <- as.integer(as.character(avgscoreDistrict$Year))

#Animated GIF to show average scores by District have movement over time

graph1 <- ggplot(avgscoreDistrict) +

geom\_point(aes(x=District, y=Average\_Score, color=District)) +

theme(panel.grid.major = element\_blank(),axis.text.x=element\_blank(),plot.title = element\_text(hjust = 0.5)) +

scale\_y\_continuous(labels=c("0" = "F", "1" = "D","2" = "C","3"="B","4"="A")) +

ggtitle("Average Score of Schools in District\n 2008-2019")

graph1.animation <- graph1 +

transition\_time(Year) +

labs(subtitle = "Year: {frame\_time}") +

shadow\_wake(wake\_length = 0.1) + theme(legend.position = "none")

graph2.animation <- animate(graph1.animation, height = 500, width = 800, fps = 30, duration = 15, end\_pause = 60, res =100, renderer = gifski\_renderer())

#This functions saves into working directory, latest animation created

anim\_save("district\_avg\_score.gif")

#MAP VISUALIZATIONS####

#Clean the "Base$ZipCode" data using "zipcode" dataset

data(zipcode)

Base$ZipCode <- clean.zipcodes(Base$ZipCode)

#Merge the "Base" and "zipcode" datasets to obtain latlon info

BaseNew <- merge(Base, zipcode, by.x="ZipCode", by.y="zip", all.x=TRUE, all.y=FALSE)

#Call the FL state boundaries data for our map

states <- map\_data("state")

FLstate <- subset(states, region=="florida")

#Create a base map of FL

FLbase <- ggplot(data=FLstate, mapping=aes(x=long, y=lat)) + coord\_fixed(1.3) + geom\_polygon(color="black", fill="cornsilk2") + theme\_nothing()

FLbase

#Plot points on the "FLbase" map for all of the schools being assessed

map.all.schools <- FLbase + geom\_point(data=BaseNew, aes(x=longitude, y=latitude)) + ggtitle("title")

map.all.schools

#Merge the "ExcellentSchools" and "zipcode" datasets to obtain latlon info

ExcellentSchoolsNew <- merge(ExcellentSchools, zipcode, by.x="ZipCode", by.y="zip", all.x=TRUE, all.y=FALSE)

#Plot points on the "FLbase" map for all of the schools that were excelling

map.excellent.schools <- FLbase + geom\_point(data=ExcellentSchoolsNew, aes(x=longitude, y=latitude))

map.excellent.schools

#Merge the "DecliningSchools" and "zipcode" datasets to obtain latlon info

DecliningSchoolsNew <- merge(DecliningSchools, zipcode, by.x="ZipCode", by.y="zip", all.x=TRUE, all.y=FALSE)

#Plot points on the "FLbase" map for all of the schools that were declining

map.declining.schools <- FLbase + geom\_point(data=DecliningSchoolsNew, aes(x=longitude, y=latitude))

map.declining.schools

#Plot points on the "FLbase" map for the schools being assessed where green represents excelling and red represents declining

map.colors.schools <- FLbase + geom\_point(data=ExcellentSchoolsNew, aes(x=longitude, y=latitude), color="chartreuse4") + geom\_point(data=DecliningSchoolsNew, aes(x=longitude, y=latitude), color="brown3")

map.colors.schools

#Merge the "AllOtherSchools" and "zipcode" datasets to obtain latlon info

AllOtherSchoolsNew <- merge(AllOtherSchools, zipcode, by.x="ZipCode", by.y="zip", all.x=TRUE, all.y=FALSE)

#Plot points on the "FLbase" map for all of the schools that were neither excelling nor declining

map.allother.schools <- FLbase+ geom\_point(data=AllOtherSchoolsNew, aes(x=longitude, y=latitude))

map.allother.schools

#Show map by 2019 Grade

ForMap <- merge(Base, zipcode, by.x="ZipCode", by.y="zip", all.x=TRUE, all.y=FALSE)

FL\_Map <- ggplot(data=FLstate, mapping=aes(x=long, y=lat)) + coord\_fixed(1.3) + geom\_polygon(color="black", fill="cornsilk2") + ggtitle("title") + theme\_nothing(legend=TRUE)

(Map\_Grades <- FL\_Map + geom\_point(data=ForMap, aes(x=longitude, y=latitude,color=`2019`)) + ggtitle("Schools labeled by their 2019 Grade") + theme(plot.title = element\_text(hjust = 0.5)))

#DESCRIPTIVE STATISTICS ####

#Average income by region

avgIncome <- mean(Performance$Income)

avgIncome

#Histogram of income

incomeHist <- ggplot(Performance, aes(x=Income)) +

geom\_histogram(binwidth=10, color="black", fill="gray") +

theme(plot.title = element\_text(hjust=.5)) + labs(title = "Income Histogram")

incomeHist

#Boxplot of income

incomeBoxplot <- ggplot(Performance, aes(x=Income, y=Region)) +

geom\_boxplot(color="black", fill="gray") +

theme(plot.title = element\_text(hjust=.5)) + labs(title = "Income Boxplot")

incomeBoxplot

#Avg Income by Region

avgIncomeByRegion <- tapply(Performance$Income, Performance$Region, mean)

avgIncomeByRegion

#Median Income by Region

medianIncomeByRegion <- tapply(Performance$Income, Performance$Region, median)

medianIncomeByRegion

#Histogram of students

studentsHist <- ggplot(Performance, aes(x=Students)) +

geom\_histogram(binwidth=50, color="black", fill="gray") +

theme(plot.title = element\_text(hjust=.5)) + labs(title = "Students Histogram")

studentsHist

#Boxplot of students

studentsBoxplot <- ggplot(Performance, aes(x=Students, y=Region)) +

geom\_boxplot(color="black", fill="gray") +

theme(plot.title = element\_text(hjust=.5)) + labs(title = "Students Boxplot")

studentsBoxplot

#Avg Students by Region

avgStudentsByRegion <- tapply(Performance$Students, Performance$Region, mean)

avgStudentsByRegion

avgstureg <- data.frame(Region=names(avgStudentsByRegion),Mean=avgStudentsByRegion)

str(avgstureg)

View(avgstureg)

#Plot Avg Students by Region

plotAvgStudentsByRegion <- ggplot(avgstureg, aes(x=Region, y=Mean)) +

geom\_bar(position="stack", stat="identity") +

theme(plot.title = element\_text(hjust=.5)) + labs(x="Region", y="Charter Schools", title = "Avg. of Students by Region")

plotAvgStudentsByRegion

#Count Schools by Region

countSchoolsByRegion <- tapply(Performance$School, Performance$Region, length)

countSchoolsByRegion

#Count Charter Schools by Region

countCharterByRegion <- tapply(Performance$School, list(Performance$Region, Performance$Charter=="YES"), length)

countCharterByRegion

#Plot Charter Schools by Region

plotCharterByRegion <- ggplot(Performance, aes(fill=Charter, x=Region, y=Charter)) +

geom\_bar(position="stack", stat="identity") +

theme(plot.title = element\_text(hjust=.5)) + labs(x="Region", y="Charter Schools", title = "Charter Schools by Region")

plotCharterByRegion

#Avg % Economically Disadvantaged by Region

avgEconDisadvByRegion <- tapply(Performance$Percent\_Econ\_Disadv, Performance$Region, mean)

avgEconDisadvByRegion

#Boxplot of % Economically Disadvantaged

disadvantagedBoxplot <- ggplot(Performance, aes(x=Percent\_Econ\_Disadv, y=Region)) +

geom\_boxplot(color="black", fill="gray") +

theme(plot.title = element\_text(hjust=.5)) + labs(title = "% Economically Disadvantaged Boxplot")

disadvantagedBoxplot

(Region2\_Focus <- Performance %>%

filter(Region=="Region\_2") %>%

mutate(Excellent=case\_when(Label=="Excellent"~"Excellent Schools",TRUE~"All Other Schools")) %>%

ggplot(aes(x=Percent\_Econ\_Disadv,y=Excellent,fill=Excellent)) + geom\_boxplot() + theme\_fivethirtyeight() +

theme(plot.title = element\_text(hjust=.5),axis.title.y=element\_blank(),axis.text.y=element\_text(size=10),legend.position = "none") + labs(title = "% Economically Disadvantaged Boxplot\nfor Region 2\nExcellent Schools v. All Other Schools"))

* Executed Code

|  |
| --- |
| > #Excel file should be in working directory, named as Data\_Raw.xlsx  > #assumes only one sheet in Excel and there are no formulas, so copy-paste-as-values  > #ignore Warning message for expecting logical since it's not a column we'll be using  > Base <- read\_excel("Data\_Raw.xlsx",range = "A5:BA3342")  New names:  \* `61` -> `61...11`  \* `50` -> `50...12`  \* `61` -> `61...13`  \* `50` -> `50...15`  \* `` -> ...17  \* ...  Warning messages:  1: In read\_fun(path = enc2native(normalizePath(path)), sheet\_i = sheet, :  Expecting logical in AU1766 / R1766C47: got '350801359022'  2: In read\_fun(path = enc2native(normalizePath(path)), sheet\_i = sheet, :  Expecting logical in AU1769 / R1769C47: got '350801359022'  3: In read\_fun(path = enc2native(normalizePath(path)), sheet\_i = sheet, :  Expecting numeric in H3340 / R3340C8: got 'High'  4: In read\_fun(path = enc2native(normalizePath(path)), sheet\_i = sheet, :  Expecting numeric in H3341 / R3341C8: got 'MID'  5: In read\_fun(path = enc2native(normalizePath(path)), sheet\_i = sheet, :  Expecting numeric in H3342 / R3342C8: got 'LOW'  >  > #Keep only columns of interest, check structure  > Base <- Base[,c(2,4,5,6,7,8,9,25,26,27,28,29,30,31,32,33,34,35,36,48,49,51,52,53)]  > str(Base)  tibble [3,337 x 24] (S3: tbl\_df/tbl/data.frame)  $ ALACHUA : chr [1:3337] "ALACHUA" "ALACHUA" "ALACHUA" "ALACHUA" ...  $ LITTLEWOOD ELEMENTARY SCHOOL: chr [1:3337] "W. A. METCALFE ELEMENTARY SCHOOL" "JOSEPH  $ Region\_2 : chr [1:3337] "Region\_2" "Region\_2" "Region\_2" "Region\_2" ...  $ 32605 : num [1:3337] 32609 32641 32641 32609 32605 ...  $ 66.951078519999996 : num [1:3337] 31.8 28.2 28.2 31.8 67 ...  $ 699 : num [1:3337] 300 565 723 733 940 NA 377 501 291 383 ...  $ MID : chr [1:3337] "LOW" "LOW" "MID" "MID" ...  $ B...25 : chr [1:3337] "B" "D" "B" "B" ...  $ B...26 : chr [1:3337] "D" "C" "B" "B" ...  $ C...27 : chr [1:3337] "D" "C" "B" "B" ...  $ B...28 : chr [1:3337] "C" "B" "B" "B" ...  $ B...29 : chr [1:3337] "D" "C" "A" "B" ...  $ B...30 : chr [1:3337] "C" "C" "A" "C" ...  $ A...31 : chr [1:3337] "D" "D" "B" "B" ...  $ A...32 : chr [1:3337] "F" "C" "A" "B" ...  $ A...33 : chr [1:3337] "D" "C" "A" "A" ...  $ A...34 : chr [1:3337] "C" "C" "B" "A" ...  $ A...35 : chr [1:3337] "C" "A" "A" "A" ...  $ A...36 : chr [1:3337] "A" "C" "A" "A" ...  $ NO : chr [1:3337] "NO" "NO" "NO" "NO" ...  $ YES : chr [1:3337] "YES" "YES" "NO" "NO" ...  $ 01 : chr [1:3337] "01" "01" "02" "02" ...  $ 59.2 : num [1:3337] 95.7 92.3 76.7 70.4 61.8 59.6 62 48.3 49.7  $ 79.400000000000006 : num [1:3337] 100 100 77 80.1 79.1 58.4 100 79.2 100  >  > #Change column names  > colnames(Base) <- c("District","School","Region","ZipCode","Income","Students","Size",  "2019","2018","2017","2016","2015","2014","2013","2012","2011","2010","2009","2008",  "Charter","TitleI","Type","Percent\_Minority","Percent\_Econ\_Disadv")  >  > #Select for only elementary schools, and then remove that column  > Base <- Base %>%  + filter(Type =="01") %>%  + select(-contains("Type"))  >  > #Use only complete cases  > Base <- Base %>%  + filter(complete.cases(.))  >  > # CREATE PERFORMANCE SET FOR EVALUATING SCHOOLS ####  > #function to convert letter grades into numbers (works specifically with Base,  note year\_index as column numbers)  > recode\_col <- function(df){  + copydf <- df  + year\_index <- 8:19  + for (i in year\_index){  + copydf[i] <- recode(pull(df,i),"A"=4,"B"=3,"C"=2,"D"=1,"F"=0)  + }  + return (copydf)  + }  > Performance <- recode\_col(Base)  > str(Performance)  tibble [1,538 x 23] (S3: tbl\_df/tbl/data.frame)  $ District : chr [1:1538] "ALACHUA" "ALACHUA" "ALACHUA" "ALACHUA" ...  $ School : chr [1:1538] "W. A. METCALFE ELEMENTARY SCHOOL" "JOSEPH  $ Region : chr [1:1538] "Region\_2" "Region\_2" "Region\_2" "Region\_2" ...  $ ZipCode : num [1:1538] 32609 32641 32615 32618 32640 ...  $ Income : num [1:1538] 31.8 28.2 56.8 45.2 40.3 ...  $ Students : num [1:1538] 300 565 377 501 308 712 742 492 791 805 ...  $ Size : chr [1:1538] "LOW" "LOW" "LOW" "LOW" ...  $ 2019 : num [1:1538] 3 1 2 2 2 2 1 2 4 3 ...  $ 2018 : num [1:1538] 1 2 1 3 2 1 1 2 4 2 ...  $ 2017 : num [1:1538] 1 2 2 3 3 1 2 2 4 4 ...  $ 2016 : num [1:1538] 2 3 2 3 1 1 1 2 4 4 ...  $ 2015 : num [1:1538] 1 2 2 4 2 2 2 3 4 4 ...  $ 2014 : num [1:1538] 2 2 1 3 0 2 2 2 4 4 ...  $ 2013 : num [1:1538] 1 1 2 3 0 3 2 3 4 4 ...  $ 2012 : num [1:1538] 0 2 1 4 1 4 3 3 4 4 ...  $ 2011 : num [1:1538] 1 2 2 4 2 4 1 4 4 4 ...  $ 2010 : num [1:1538] 2 2 3 3 1 2 3 3 4 4 ...  $ 2009 : num [1:1538] 2 4 4 4 3 4 4 4 4 4 ...  $ 2008 : num [1:1538] 4 2 3 2 1 2 2 3 4 4 ...  $ Charter : chr [1:1538] "NO" "NO" "NO" "NO" ...  $ TitleI : chr [1:1538] "YES" "YES" "YES" "YES" ...  $ Percent\_Minority : num [1:1538] 95.7 92.3 62 48.3 46.6 85.1 83.8 45.5 52.1 55.1 ...  $ Percent\_Econ\_Disadv: num [1:1538] 100 100 100 79.2 100 100 100 75.7 35 59.5 ...  >  > #Average Moving Range function (not generic, works just for our dataset)  > AverageMovingRange <- function(df){  + difference <- function(vector){  + vector <- as.numeric(vector)  + index <- 1:length(vector)  + differences <- c()  + for (i in index){  + if (i==length(vector)){  + break  + }  + else {  + differences[i] <- abs(vector[i]-vector[i+1])  + }  + }  + return (differences)  + }  + copydf <- df  + row\_index <- 1:nrow(df)  + movingRangeVector <- c()  + for (i in row\_index){  + currentrow <- df[i,][8:19]  + movingRangeVector[i] <- mean(difference(currentrow))  + }  + copydf$AverageMovingRange <- movingRangeVector  + return (copydf)  + }  > Performance <- AverageMovingRange(Performance)  >  > #Average of Grades function (again, function specific to our dataset)  > AverageofGrades <- function(df){  + copydf <- df  + row\_index <- 1:nrow(df)  + AverageGradesVector <- c()  + for (i in row\_index){  + AverageGradesVector[i] <- mean(as.numeric(copydf[i,][8:19]))  + }  + copydf$AverageGrade <- AverageGradesVector  + return(copydf)  + }  > Performance <- AverageofGrades(Performance)  >  > #Make columns for Upper and Lower Natural Process Limit  > Performance <- Performance %>%  + mutate(UNPL=(AverageGrade+(2.66\*AverageMovingRange))) %>%  + mutate(LNPL=(AverageGrade-(2.66\*AverageMovingRange)))  >  > #Function that evaluates school's performance  > Evaluation <- function(df){  + copydf <- df  + row\_index <- 1:nrow(df)  + EvaluationVector <- c()  + for (i in row\_index){  + LNPLbreaks <- 0  + UNPLbreaks <- 0  + year\_index <- 8:12  + for (j in year\_index){  + if (df[i,][j] > df[i,]$UNPL) {  + UNPLbreaks <- UNPLbreaks+1  + }  + else if (df[i,][j] < df[i,]$LNPL) {  + LNPLbreaks <- LNPLbreaks+1  + }  + }  + EvaluationScore <- case\_when(  + (LNPLbreaks == 0 and UNPLbreaks == 0) ~ "Stable",  + (  + (LNPLbreaks == 2 and UNPLbreaks == 0) |  + (LNPLbreaks == 3 and UNPLbreaks == 0) |  + (LNPLbreaks == 4 and UNPLbreaks == 0) |  + (LNPLbreaks == 5 and UNPLbreaks == 0)  + ) ~ "Declining",  + (  + (LNPLbreaks == 0 and UNPLbreaks == 2) |  + (LNPLbreaks == 0 and UNPLbreaks == 3) |  + (LNPLbreaks == 0 and UNPLbreaks == 4) |  + (LNPLbreaks == 0 and UNPLbreaks == 5)  + ) ~ "Improving",  + (  + (LNPLbreaks == 1 and UNPLbreaks == 0) |  + (LNPLbreaks == 0 and UNPLbreaks == 1)  + ) ~ "Inconclusive",  + TRUE ~ "Fluctuating"  + )  + EvaluationVector[i] <- EvaluationScore  + }  + copydf$EvaluationType <- EvaluationVector  + return(copydf)  + }  > Performance <- Evaluation(Performance)  >  > #View how many schools fall into what category  > EvaluationTab <- Performance %>%  + group\_by(EvaluationType) %>%  + summarise(Number\_of\_Schools = n()) %>%  + arrange(desc(Number\_of\_Schools))  `summarise()` ungrouping output (override with `.groups` argument)  >  > #Create new column called "Label" to be used for prediction along with 2019 Grades  > Performance <- Performance %>%  + mutate(Label=case\_when(  + EvaluationType=="Declining" ~ "Declining",  + AverageGrade==4 ~ "Excellent",  + TRUE ~ "Other")) |
|  |
| |  | | --- | | > | |
| > #XMR Chart####  >  > Improving <- Performance %>%  + filter(EvaluationType=="Improving") %>%  + slice(3) %>%  + select(starts\_with("20")) %>%  + gather(key="Year",value="Grade") %>%  + arrange(Year)  > (imp\_xmr\_plot <-  + ggplot(Improving, aes(x = Year, y = Grade,group=1)) +  + geom\_point() + geom\_line() + ggtitle("Improving Schools") +  + theme(plot.title = element\_text(hjust = 0.5)) +  + stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))  >  > Fluctuating <- Performance %>%  + filter(EvaluationType=="Fluctuating") %>%  + slice(2) %>%  + select(starts\_with("20")) %>%  + gather(key="Year",value="Grade") %>%  + arrange(Year)  > (fluc\_xmr\_plot <- ggplot(Fluctuating, aes(x = Year, y = Grade,group=1)) +  + geom\_point() + geom\_line() + ggtitle("Fluctuating Schools") +  + theme(plot.title = element\_text(hjust = 0.5)) +  + stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))  >  > Stable <- Performance %>%  + filter(EvaluationType=="Stable") %>%  + slice(4) %>%  + select(starts\_with("20")) %>%  + gather(key="Year",value="Grade") %>%  + arrange(Year)  > (stab\_xmr\_plot <- ggplot(Stable, aes(x = Year, y = Grade,group=1)) +  + geom\_point() + geom\_line() + ggtitle("Stable Schools") +  + theme(plot.title = element\_text(hjust = 0.5)) +  + stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))  >  > Declining <- Performance %>%  + filter(EvaluationType=="Declining") %>%  + slice(2) %>%  + select(starts\_with("20")) %>%  + gather(key="Year",value="Grade") %>%  + arrange(Year)  > (decl\_xmr\_plot <- ggplot(Declining, aes(x = Year, y = Grade,group=1)) +  + geom\_point() + geom\_line() + ggtitle("Declining Schools") +  + theme(plot.title = element\_text(hjust = 0.5)) +  + stat\_QC(method = "XmR", auto.label = T, label.digits = 2, show.1n2.sigma = T))  >  > grid.arrange(imp\_xmr\_plot, fluc\_xmr\_plot, stab\_xmr\_plot, decl\_xmr\_plot, nrow = 2)  >  > ddply(Improving, .variables = "Grade",  + .fun = function(df)  + {QC\_Lines(data = Improving$Grade, method = "XmR")})  Grade xBar\_one\_LCL mean xBar\_one\_UCL mR\_LCL mR mR\_UCL sigma  1 2 1.782727 2.75 3.717273 0 0.3636364 1.188364 0.3224242  2 3 1.782727 2.75 3.717273 0 0.3636364 1.188364 0.3224242  3 4 1.782727 2.75 3.717273 0 0.3636364 1.188364 0.3224242  >  > ddply(Fluctuating, .variables = "Grade",  + .fun = function(df)  + {QC\_Lines(data = Fluctuating$Grade, method = "XmR")})  Grade xBar\_one\_LCL mean xBar\_one\_UCL mR\_LCL mR mR\_UCL sigma  1 0 0.3906061 2.083333 3.776061 0 0.6363636 2.079636 0.5642424  2 1 0.3906061 2.083333 3.776061 0 0.6363636 2.079636 0.5642424  3 2 0.3906061 2.083333 3.776061 0 0.6363636 2.079636 0.5642424  4 3 0.3906061 2.083333 3.776061 0 0.6363636 2.079636 0.5642424  5 4 0.3906061 2.083333 3.776061 0 0.6363636 2.079636 0.5642424  >  > ddply(Stable, .variables = "Grade",  + .fun = function(df)  + {QC\_Lines(data = Stable$Grade, method = "XmR")})  Grade xBar\_one\_LCL mean xBar\_one\_UCL mR\_LCL mR mR\_UCL sigma  1 2 1.232121 3.166667 5.101212 0 0.7272727 2.376727 0.6448485  2 3 1.232121 3.166667 5.101212 0 0.7272727 2.376727 0.6448485  3 4 1.232121 3.166667 5.101212 0 0.7272727 2.376727 0.6448485  >  > ddply(Declining, .variables = "Grade",  + .fun = function(df)  + {QC\_Lines(data = Declining$Grade, method = "XmR")})  Grade xBar\_one\_LCL mean xBar\_one\_UCL mR\_LCL mR mR\_UCL sigma  1 1 1.215758 2.666667 4.117576 0 0.5454545 1.782545 0.4836364  2 2 1.215758 2.666667 4.117576 0 0.5454545 1.782545 0.4836364  3 3 1.215758 2.666667 4.117576 0 0.5454545 1.782545 0.4836364  4 4 1.215758 2.666667 4.117576 0 0.5454545 1.782545 0.4836364 |
|  |
| |  | | --- | | > | |
| > #LINEAR MODELS FOR CLUSTERS ####  > #Isolate 3 clusters into different datasets  > ExcellentSchools <- Performance %>%  + filter(AverageGrade==4)  > DecliningSchools <- Performance %>%  + filter(EvaluationType=="Declining")  > AllOtherSchools <- Performance %>%  + filter(!(AverageGrade==4 | EvaluationType=="Declining"))  > #Clean datasets to retain only columns to be used for linear models  > CleanforLM <- function(df){  + removethiscolumns <- c(1:2,4,7,9:19,24:29)  + df <- df[,-removethiscolumns]  + return(df)  + }  > ExcellentSchoolsforLM <- CleanforLM(ExcellentSchools)  > DecliningSchoolsforLM <- CleanforLM(DecliningSchools)  > AllOtherSchoolsforLM <- CleanforLM(AllOtherSchools)  > #create linear models  > names(ExcellentSchoolsforLM) <- make.names(names(ExcellentSchoolsforLM))  > ExcellentSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  + Percent\_Minority + Region + Students + TitleI,  data=ExcellentSchoolsforLM)  > summary(ExcellentSchools\_LinearModel)  Call:  lm(formula = X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  Percent\_Minority + Region + Students + TitleI, data = ExcellentSchoolsforLM)  Residuals:  Min 1Q Median 3Q Max  -4.873e-15 -6.640e-17 -3.700e-18 9.780e-17 5.835e-16  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.000e+00 2.150e-16 1.861e+16 < 2e-16 \*\*\*  CharterYES 7.255e-18 1.468e-16 4.900e-02 0.96065  Income 4.983e-19 1.050e-18 4.750e-01 0.63575  Percent\_Econ\_Disadv 7.300e-18 2.926e-18 2.495e+00 0.01366 \*  Percent\_Minority -2.083e-18 1.915e-18 -1.088e+00 0.27849  RegionRegion\_2 -3.538e-16 1.569e-16 -2.254e+00 0.02560 \*  RegionRegion\_3 -3.804e-17 1.650e-16 -2.310e-01 0.81794  RegionRegion\_4 -3.621e-17 1.291e-16 -2.800e-01 0.77949  RegionRegion\_5 2.379e-17 1.245e-16 1.910e-01 0.84870  Students -5.807e-20 1.946e-19 -2.980e-01 0.76585  TitleIYES -4.982e-16 1.505e-16 -3.311e+00 0.00116 \*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 4.256e-16 on 153 degrees of freedom  Multiple R-squared: 0.4606, Adjusted R-squared: 0.4254  F-statistic: 13.07 on 10 and 153 DF, p-value: 2.4e-16  Warning message:  In summary.lm(ExcellentSchools\_LinearModel) :  essentially perfect fit: summary may be unreliable  >  > names(DecliningSchoolsforLM) <- make.names(names(DecliningSchoolsforLM))  > DecliningSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  + Percent\_Minority + Region + Students + TitleI,  data=DecliningSchoolsforLM)  > summary(DecliningSchools\_LinearModel)  Call:  lm(formula = X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  Percent\_Minority + Region + Students + TitleI, data = DecliningSchoolsforLM)  Residuals:  Min 1Q Median 3Q Max  -1.37930 -0.42939 0.04858 0.35562 1.79349  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.048e+00 5.543e-01 7.303 4.24e-11 \*\*\*  CharterYES 4.698e-01 3.929e-01 1.196 0.234387  Income 1.688e-03 3.192e-03 0.529 0.597972  Percent\_Econ\_Disadv -1.872e-02 5.029e-03 -3.723 0.000309 \*\*\*  Percent\_Minority 3.950e-03 3.315e-03 1.191 0.235975  RegionRegion\_2 1.747e-01 3.871e-01 0.451 0.652619  RegionRegion\_3 -5.343e-01 3.829e-01 -1.395 0.165637  RegionRegion\_4 -2.587e-01 3.584e-01 -0.722 0.471865  RegionRegion\_5 7.382e-02 3.508e-01 0.210 0.833725  Students -5.731e-05 2.772e-04 -0.207 0.836612  TitleIYES -2.707e-01 2.194e-01 -1.233 0.219955  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.649 on 113 degrees of freedom  Multiple R-squared: 0.413, Adjusted R-squared: 0.3611  F-statistic: 7.951 on 10 and 113 DF, p-value: 1.375e-09  > names(AllOtherSchoolsforLM) <- make.names(names(AllOtherSchoolsforLM))  > AllOtherSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  + Percent\_Minority + Region + Students + TitleI,  data=AllOtherSchoolsforLM)  > summary(AllOtherSchools\_LinearModel)  Call:  lm(formula = X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  Percent\_Minority + Region + Students + TitleI, data = AllOtherSchoolsforLM)  Residuals:  Min 1Q Median 3Q Max  -2.27265 -0.44052 -0.06868 0.51357 2.05142  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.5400649 0.1512158 30.024 < 2e-16 \*\*\*  CharterYES 0.3788368 0.1498855 2.528 0.011611 \*  Income 0.0021239 0.0006100 3.482 0.000515 \*\*\*  Percent\_Econ\_Disadv -0.0215686 0.0016721 -12.899 < 2e-16 \*\*\*  Percent\_Minority -0.0038362 0.0010835 -3.541 0.000414 \*\*\*  RegionRegion\_2 0.0199533 0.1027190 0.194 0.846011  RegionRegion\_3 -0.0985030 0.0907504 -1.085 0.277943  RegionRegion\_4 -0.2284377 0.0839970 -2.720 0.006628 \*\*  RegionRegion\_5 0.1785451 0.0866085 2.062 0.039462 \*  Students 0.0001606 0.0001064 1.509 0.131573  TitleIYES -0.0546862 0.0781521 -0.700 0.484221  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.7194 on 1239 degrees of freedom  Multiple R-squared: 0.4042, Adjusted R-squared: 0.3994  F-statistic: 84.06 on 10 and 1239 DF, p-value: < 2.2e-16  >  > #Linear Model for all Schools  > AllSchools\_LM <- CleanforLM(Performance)  > names(AllSchools\_LM) <- make.names(names(AllSchools\_LM))  > AllSchools\_LinearModel <- lm(X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  Percent\_Minority + Region + Students + TitleI, data=AllSchools\_LM)  > summary(AllSchools\_LinearModel)  Call:  lm(formula = X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  Percent\_Minority + Region + Students + TitleI, data = AllSchools\_LM)  Residuals:  Min 1Q Median 3Q Max  -2.32292 -0.42301 -0.07304 0.47988 1.98856  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.515e+00 1.251e-01 36.078 < 2e-16 \*\*\*  CharterYES 3.935e-01 1.163e-01 3.384 0.000733 \*\*\*  Income 1.723e-03 5.472e-04 3.149 0.001669 \*\*  Percent\_Econ\_Disadv -2.040e-02 1.376e-03 -14.826 < 2e-16 \*\*\*  Percent\_Minority -3.194e-03 9.439e-04 -3.383 0.000734 \*\*\*  RegionRegion\_2 5.707e-03 8.968e-02 0.064 0.949263  RegionRegion\_3 -1.306e-01 8.055e-02 -1.621 0.105194  RegionRegion\_4 -2.233e-01 7.382e-02 -3.025 0.002527 \*\*  RegionRegion\_5 1.622e-01 7.517e-02 2.157 0.031133 \*  Students 7.545e-05 9.169e-05 0.823 0.410694  TitleIYES -8.909e-02 6.839e-02 -1.303 0.192897  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.6978 on 1527 degrees of freedom  Multiple R-squared: 0.4722, Adjusted R-squared: 0.4687  F-statistic: 136.6 on 10 and 1527 DF, p-value: < 2.2e-16  >  > #Models for 3 clusters only using significant variables from above results (except  Region since it's a 5-level factor)  > ExcellentSchools\_LinearModel\_sigonly <- lm(X2019 ~ Percent\_Econ\_Disadv + TitleI,  data=ExcellentSchoolsforLM)  > summary(ExcellentSchools\_LinearModel\_sigonly)  Call:  lm(formula = X2019 ~ Percent\_Econ\_Disadv + TitleI, data = ExcellentSchoolsforLM)  Residuals:  Min 1Q Median 3Q Max  -5.161e-15 -5.110e-17 2.790e-17 9.750e-17 4.848e-16  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.000e+00 8.193e-17 4.882e+16 < 2e-16 \*\*\*  Percent\_Econ\_Disadv 7.330e-18 2.487e-18 2.948e+00 0.003677 \*\*  TitleIYES -5.654e-16 1.445e-16 -3.914e+00 0.000134 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 4.27e-16 on 161 degrees of freedom  Multiple R-squared: 0.5101, Adjusted R-squared: 0.5041  F-statistic: 83.83 on 2 and 161 DF, p-value: < 2.2e-16  Warning message:  In summary.lm(ExcellentSchools\_LinearModel\_sigonly) :  essentially perfect fit: summary may be unreliable  >  > DecliningSchools\_LinearModel\_sigonly <- lm(X2019 ~ Percent\_Econ\_Disadv,  data=DecliningSchoolsforLM)  > summary(DecliningSchools\_LinearModel\_sigonly)  Call:  lm(formula = X2019 ~ Percent\_Econ\_Disadv, data = DecliningSchoolsforLM)  Residuals:  Min 1Q Median 3Q Max  -1.3956 -0.4842 -0.1062 0.4284 1.6897  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.348601 0.223617 19.447 < 2e-16 \*\*\*  Percent\_Econ\_Disadv -0.022423 0.002891 -7.757 2.95e-12 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.6672 on 122 degrees of freedom  Multiple R-squared: 0.3303, Adjusted R-squared: 0.3248  F-statistic: 60.17 on 1 and 122 DF, p-value: 2.953e-12  >  > AllOtherSchools\_LinearModel\_sigonly <- lm(X2019 ~ Charter + Income +  Percent\_Econ\_Disadv + Percent\_Minority, data=AllOtherSchoolsforLM)  > summary(AllOtherSchools\_LinearModel\_sigonly)  Call:  lm(formula = X2019 ~ Charter + Income + Percent\_Econ\_Disadv +  Percent\_Minority, data = AllOtherSchoolsforLM)  Residuals:  Min 1Q Median 3Q Max  -2.2412 -0.4007 -0.1573 0.5742 1.8578  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 4.6288374 0.1052271 43.989 < 2e-16 \*\*\*  CharterYES 0.3400559 0.1524703 2.230 0.025905 \*  Income 0.0022347 0.0006167 3.624 0.000302 \*\*\*  Percent\_Econ\_Disadv -0.0239391 0.0011897 -20.122 < 2e-16 \*\*\*  Percent\_Minority -0.0016759 0.0009174 -1.827 0.067964 .  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 0.7362 on 1245 degrees of freedom  Multiple R-squared: 0.3732, Adjusted R-squared: 0.3712  F-statistic: 185.3 on 4 and 1245 DF, p-value: < 2.2e-16 |
|  |
| |  | | --- | | > | |
| > #MACHINE LEARNING FOR PREDICTING GRADES (2019)####  > MLData <- Base[,-c(1:2,4,7,9:19)]  > colnames(MLData)[colnames(MLData) == "2019"] <- "Grade"  > #Convert to Factor  > MLData <- MLData %>%  + mutate\_at(c(1,4:6),factor)  > #Setup test and train data  > set.seed(5678)  > pd <- sample(2,nrow(MLData), replace = TRUE, prob = c(.75,.25))  > train <- MLData[pd==1,]  > test <- MLData[pd==2,]  >  > #KSVM  > ksvmresults <- ksvm(Grade~., data = train, kernel = "rbfdot",  + kpar="automatic",C=10, cross=10, prob.model=TRUE)  > ksvmresults  Support Vector Machine object of class "ksvm"  SV type: C-svc (classification)  parameter : cost C = 10  Gaussian Radial Basis kernel function.  Hyperparameter : sigma = 0.23085774923009  Number of Support Vectors : 910  Objective Function Value : -2565.124 -1621.085 -375.5647 -76.0723 -4333.204 -1071.834  -117.1747 -1357.907 -118.254 -115.6979  Training error : 0.313609  Cross validation error : 0.425167  Probability model included.  > #Confusion Matrix for Train  > confusionMatrix(predict(ksvmresults,train),train$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 302 40 10 0 0  B 34 147 42 2 0  C 35 140 355 62 6  D 0 0 0 8 0  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.6864  95% CI : (0.6591, 0.7128)  No Information Rate : 0.344  P-Value [Acc > NIR] : < 2.2e-16    Kappa : 0.5387    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.8140 0.4495 0.8722 0.111111 0.000000  Specificity 0.9384 0.9089 0.6869 1.000000 1.000000  Pos Pred Value 0.8580 0.6533 0.5936 1.000000 NaN  Neg Pred Value 0.9170 0.8121 0.9111 0.945532 0.994928  Prevalence 0.3136 0.2764 0.3440 0.060862 0.005072  Detection Rate 0.2553 0.1243 0.3001 0.006762 0.000000  Detection Prevalence 0.2975 0.1902 0.5055 0.006762 0.000000  Balanced Accuracy 0.8762 0.6792 0.7795 0.555556 0.500000  > #Confusion Matrix for Test  > confusionMatrix(predict(ksvmresults,test),test$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 75 32 7 0 0  B 14 27 27 1 0  C 21 37 90 21 1  D 0 0 2 0 0  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.5408  95% CI : (0.4874, 0.5936)  No Information Rate : 0.3549  P-Value [Acc > NIR] : 6.915e-13    Kappa : 0.3224    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.6818 0.28125 0.7143 0.000000 0.000000  Specificity 0.8408 0.83784 0.6507 0.993994 1.000000  Pos Pred Value 0.6579 0.39130 0.5294 0.000000 NaN  Neg Pred Value 0.8548 0.75874 0.8054 0.937677 0.997183  Prevalence 0.3099 0.27042 0.3549 0.061972 0.002817  Detection Rate 0.2113 0.07606 0.2535 0.000000 0.000000  Detection Prevalence 0.3211 0.19437 0.4789 0.005634 0.000000  Balanced Accuracy 0.7613 0.55954 0.6825 0.496997 0.500000  >  > #SVM  > svmresults <- svm(Grade~., data = train, kernel = "radial",  + kpar="automatic",C=10, cross=10, prob.model=TRUE)  > svmresults  Call:  svm(formula = Grade ~ ., data = train, kernel = "radial", kpar = "automatic", C = 10,  cross = 10, prob.model = TRUE)  Parameters:  SVM-Type: C-classification  SVM-Kernel: radial  cost: 1  Number of Support Vectors: 954  >  > #Confusion Matrix for Train  > confusionMatrix(predict(svmresults,train),train$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 279 50 15 0 0  B 44 94 35 0 0  C 48 183 357 72 6  D 0 0 0 0 0  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.6171  95% CI : (0.5887, 0.6449)  No Information Rate : 0.344  P-Value [Acc > NIR] : < 2.2e-16    Kappa : 0.4324    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.7520 0.28746 0.8771 0.00000 0.000000  Specificity 0.9200 0.90771 0.6018 1.00000 1.000000  Pos Pred Value 0.8110 0.54335 0.5360 NaN NaN  Neg Pred Value 0.8903 0.76931 0.9033 0.93914 0.994928  Prevalence 0.3136 0.27642 0.3440 0.06086 0.005072  Detection Rate 0.2358 0.07946 0.3018 0.00000 0.000000  Detection Prevalence 0.2908 0.14624 0.5630 0.00000 0.000000  Balanced Accuracy 0.8360 0.59759 0.7395 0.50000 0.500000  > #Confusion Matrix for Test  > confusionMatrix(predict(svmresults,test),test$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 70 26 6 0 0  B 16 23 14 0 0  C 24 47 106 22 1  D 0 0 0 0 0  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.5606  95% CI : (0.5072, 0.6129)  No Information Rate : 0.3549  P-Value [Acc > NIR] : 2.396e-15    Kappa : 0.3447    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.6364 0.23958 0.8413 0.00000 0.000000  Specificity 0.8694 0.88417 0.5895 1.00000 1.000000  Pos Pred Value 0.6863 0.43396 0.5300 NaN NaN  Neg Pred Value 0.8419 0.75828 0.8710 0.93803 0.997183  Prevalence 0.3099 0.27042 0.3549 0.06197 0.002817  Detection Rate 0.1972 0.06479 0.2986 0.00000 0.000000  Detection Prevalence 0.2873 0.14930 0.5634 0.00000 0.000000  Balanced Accuracy 0.7529 0.56188 0.7154 0.50000 0.500000  >  > #Decision Tree  > tree <- ctree (Grade ~ .,train)  > plot(tree)  > #Confusion Matrix for Train  > confusionMatrix(predict(tree,train),train$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 253 43 10 0 0  B 104 199 161 7 0  C 14 85 236 65 6  D 0 0 0 0 0  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.5816  95% CI : (0.5529, 0.6099)  No Information Rate : 0.344  P-Value [Acc > NIR] : < 2.2e-16    Kappa : 0.3942    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.6819 0.6086 0.5799 0.00000 0.000000  Specificity 0.9347 0.6822 0.7809 1.00000 1.000000  Pos Pred Value 0.8268 0.4225 0.5813 NaN NaN  Neg Pred Value 0.8655 0.8202 0.7799 0.93914 0.994928  Prevalence 0.3136 0.2764 0.3440 0.06086 0.005072  Detection Rate 0.2139 0.1682 0.1995 0.00000 0.000000  Detection Prevalence 0.2587 0.3981 0.3432 0.00000 0.000000  Balanced Accuracy 0.8083 0.6454 0.6804 0.50000 0.500000  > #Confusion Matrix for Test  > confusionMatrix(predict(tree,test),test$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 66 22 2 0 0  B 33 44 47 3 0  C 11 30 77 19 1  D 0 0 0 0 0  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.5268  95% CI : (0.4734, 0.5797)  No Information Rate : 0.3549  P-Value [Acc > NIR] : 2.831e-11    Kappa : 0.3109    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.6000 0.4583 0.6111 0.00000 0.000000  Specificity 0.9020 0.6795 0.7336 1.00000 1.000000  Pos Pred Value 0.7333 0.3465 0.5580 NaN NaN  Neg Pred Value 0.8340 0.7719 0.7742 0.93803 0.997183  Prevalence 0.3099 0.2704 0.3549 0.06197 0.002817  Detection Rate 0.1859 0.1239 0.2169 0.00000 0.000000  Detection Prevalence 0.2535 0.3577 0.3887 0.00000 0.000000  Balanced Accuracy 0.7510 0.5689 0.6724 0.50000 0.500000  >  > #Naive Bayes  > NBmodel <- naiveBayes(Grade~., data=train)  > #Confusion Matrix for Train  > confusionMatrix(predict(NBmodel,train),train$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 250 57 15 0 0  B 52 87 43 1 0  C 64 143 257 20 1  D 4 40 84 51 5  F 1 0 8 0 0  Overall Statistics    Accuracy : 0.5452  95% CI : (0.5163, 0.5739)  No Information Rate : 0.344  P-Value [Acc > NIR] : < 2.2e-16    Kappa : 0.3695    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.6739 0.26606 0.6314 0.70833 0.000000  Specificity 0.9113 0.88785 0.7062 0.88029 0.992353  Pos Pred Value 0.7764 0.47541 0.5299 0.27717 0.000000  Neg Pred Value 0.8595 0.76000 0.7851 0.97898 0.994889  Prevalence 0.3136 0.27642 0.3440 0.06086 0.005072  Detection Rate 0.2113 0.07354 0.2172 0.04311 0.000000  Detection Prevalence 0.2722 0.15469 0.4100 0.15554 0.007608  Balanced Accuracy 0.7926 0.57695 0.6688 0.79431 0.496177  > #Confusion Matrix for Test  > confusionMatrix(predict(NBmodel,test),test$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 71 22 6 0 0  B 10 22 16 0 0  C 25 43 72 7 0  D 4 9 32 15 1  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.507  95% CI : (0.4538, 0.5602)  No Information Rate : 0.3549  P-Value [Acc > NIR] : 3.218e-09    Kappa : 0.3148    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.6455 0.22917 0.5714 0.68182 0.000000  Specificity 0.8857 0.89961 0.6725 0.86186 1.000000  Pos Pred Value 0.7172 0.45833 0.4898 0.24590 NaN  Neg Pred Value 0.8477 0.75896 0.7404 0.97619 0.997183  Prevalence 0.3099 0.27042 0.3549 0.06197 0.002817  Detection Rate 0.2000 0.06197 0.2028 0.04225 0.000000  Detection Prevalence 0.2789 0.13521 0.4141 0.17183 0.000000  Balanced Accuracy 0.7656 0.56439 0.6220 0.77184 0.500000  >  > #Random Forest  > set.seed(9876)  > RFmodel <- randomForest(Grade~., data=train)  > #Confusion matrix with train data  > confusionMatrix(predict(RFmodel,train),train$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 367 0 0 0 0  B 0 310 0 1 0  C 4 17 407 12 3  D 0 0 0 59 0  F 0 0 0 0 3  Overall Statistics    Accuracy : 0.9687  95% CI : (0.9571, 0.9779)  No Information Rate : 0.344  P-Value [Acc > NIR] : < 2.2e-16    Kappa : 0.9552    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.9892 0.9480 1.0000 0.81944 0.500000  Specificity 1.0000 0.9988 0.9536 1.00000 1.000000  Pos Pred Value 1.0000 0.9968 0.9187 1.00000 1.000000  Neg Pred Value 0.9951 0.9805 1.0000 0.98843 0.997458  Prevalence 0.3136 0.2764 0.3440 0.06086 0.005072  Detection Rate 0.3102 0.2620 0.3440 0.04987 0.002536  Detection Prevalence 0.3102 0.2629 0.3745 0.04987 0.002536  Balanced Accuracy 0.9946 0.9734 0.9768 0.90972 0.750000  > #Confusion matrix with test data  > confusionMatrix(predict(RFmodel,test),test$Grade)  Confusion Matrix and Statistics  Reference  Prediction A B C D F  A 72 29 9 0 0  B 20 36 27 1 0  C 17 31 89 20 1  D 1 0 1 1 0  F 0 0 0 0 0  Overall Statistics    Accuracy : 0.5577  95% CI : (0.5044, 0.6101)  No Information Rate : 0.3549  P-Value [Acc > NIR] : 5.565e-15    Kappa : 0.3511    Mcnemar's Test P-Value : NA  Statistics by Class:  Class: A Class: B Class: C Class: D Class: F  Sensitivity 0.6545 0.3750 0.7063 0.045455 0.000000  Specificity 0.8449 0.8147 0.6987 0.993994 1.000000  Pos Pred Value 0.6545 0.4286 0.5633 0.333333 NaN  Neg Pred Value 0.8449 0.7786 0.8122 0.940341 0.997183  Prevalence 0.3099 0.2704 0.3549 0.061972 0.002817  Detection Rate 0.2028 0.1014 0.2507 0.002817 0.000000  Detection Prevalence 0.3099 0.2366 0.4451 0.008451 0.000000  Balanced Accuracy 0.7497 0.5948 0.7025 0.519724 0.500000  > #Plot Variables of Importance  > varImpPlot(RFmodel, sort = TRUE) |
|  |
| |  | | --- | | > | |
| > #VISUALIZATIONS GENERAL####  >  > #Piechart for business problem  > myPalette <- c("cadetblue3","darkblue","darkseagreen","coral1")  > PieChartLabels <- c("Building Capacity","Funds for Data-Driven Initiatives",  "Performance Based Incentives", "Support for Underperforming Schools")  > BudgetInDollars <- c(31219426, 25819420, 6500000, 123750850)  > pie(BudgetInDollars , labels = PieChartLabels, border="white",col=myPalette,main=  "Student Performance\nSupport Budget\nTotal ~ $187M")  >  > #Correlation Matrix for continuous variables  > corrplot(cor(Base[,-c(1:4,7:21)]))  >  > #Title I and Charter Schools viz  > TitleI\_AllSchools <- ggplot(data=Base, aes(x=TitleI,fill=TitleI)) + geom\_bar() +  ggtitle("All Schools - Title I") + theme\_fivethirtyeight()+ theme(panel.grid.major =  element\_blank())  > TitleI\_AllSchools  > Charter\_AllSchools <- ggplot(data=Base, aes(x=Charter,fill=Charter)) + geom\_bar() +  ggtitle("All Schools - Charter") + theme\_fivethirtyeight()+ theme(panel.grid.major =  element\_blank())  > Charter\_AllSchools  >  > #Histogram of Percent Economically Disadvantaged for Excellent Schools  > hist\_PerEconDis\_ExcellentSchools <- Performance %>%  + filter(Label=="Excellent") %>%  + ggplot(aes(x=Percent\_Econ\_Disadv)) + geom\_histogram(bins=12) +  theme\_fivethirtyeight()+ theme(panel.grid.major = element\_blank(),axis.text.  y=element\_blank()) + ggtitle("Histogram of % Economically Disadvantaged\n for  Excellent Schools") + scale\_x\_continuous(labels = function(x) paste0(x, '%'))  > hist\_PerEconDis\_ExcellentSchools  >  > #Histogram of Percent Economically Disadvantaged for Declining Schools  > hist\_PerEconDis\_DecliningSchools <- Performance %>%  + filter(Label=="Declining") %>%  + ggplot(aes(x=Percent\_Econ\_Disadv)) + geom\_histogram(bins=12) +  theme\_fivethirtyeight()+ theme(panel.grid.major = element\_blank(),axis.text.  y=element\_blank()) + ggtitle("Histogram of % Economically Disadvantaged\n for  Declining Schools") + scale\_x\_continuous(labels = function(x) paste0(x, '%'))  > hist\_PerEconDis\_DecliningSchools  >  > #Bar graph showing distribution of Region  > Region\_Viz <- ggplot(data=Base, aes(x=Region,fill=Region)) + geom\_bar() +  theme\_fivethirtyeight() + theme(plot.title = element\_text(hjust=0.5),axis.text.  x = element\_text(size = 15),legend.position = "none",panel.grid.major = element\_blank())  + ggtitle("# of Schools in each Region")  > Region\_Viz  >  > #Visualization for Title I schools in Excellent Schools  > (TitleI\_Excellent <- ggplot(data=ExcellentSchools, aes(x=TitleI,fill=TitleI))  + geom\_bar() + ggtitle("Excellent Schools - Title I") + ylab("Title I") +  theme\_fivethirtyeight() + theme(panel.grid.major = element\_blank()))  >  > #Bar graph showing distribution of Excellent, Declining, Other  > Label\_Viz <- ggplot(data=Performance, aes(x=Label,fill=Label)) + geom\_bar() +  theme\_fivethirtyeight() + theme(plot.title = element\_text(hjust=0.5),axis.text.  x = element\_text(size = 20),legend.position = "none",panel.grid.major = element\_blank())  + ggtitle("# of Schools in each Category")  > Label\_Viz  >  > #Animation of District Average Grades  >  > #Prep data, use only Grades over years and District, average Grades by District  > avgscoreDistrict <- Performance[,c(1,8:19)] %>%  + group\_by(District) %>%  + summarise(across(everything(),mean)) %>%  + melt(id="District")  `summarise()` ungrouping output (override with `.groups` argument)  > colnames(avgscoreDistrict) <- c("District","Year","Average\_Score")  > avgscoreDistrict$Year <- as.integer(as.character(avgscoreDistrict$Year))  >  > #Animated GIF to show average scores by District have movement over time  > graph1 <- ggplot(avgscoreDistrict) +  + geom\_point(aes(x=District, y=Average\_Score, color=District)) +  + theme(panel.grid.major = element\_blank(),axis.text.x=element\_blank(),plot.title  = element\_text(hjust = 0.5)) +  + scale\_y\_continuous(labels=c("0" = "F", "1" = "D","2" = "C","3"="B","4"="A")) +  + ggtitle("Average Score of Schools in District\n 2008-2019")  > graph1.animation <- graph1 +  + transition\_time(Year) +  + labs(subtitle = "Year: {frame\_time}") +  + shadow\_wake(wake\_length = 0.1) + theme(legend.position = "none")  > graph2.animation <- animate(graph1.animation, height = 500, width = 800, fps = 30,  duration = 15, end\_pause = 60, res =100, renderer = gifski\_renderer())    Frame 450 (100%)  Finalizing encoding... done!  > #This functions saves into working directory, latest animation created  > anim\_save("district\_avg\_score.gif") |
|  |
| |  | | --- | | > | |
| > #MAP VISUALIZATIONS####  >  > #Clean the "Base$ZipCode" data using "zipcode" dataset  > data(zipcode)  > Base$ZipCode <- clean.zipcodes(Base$ZipCode)  >  > #Merge the "Base" and "zipcode" datasets to obtain latlon info  > BaseNew <- merge(Base, zipcode, by.x="ZipCode", by.y="zip", all.x=TRUE, all.y=FALSE)  >  > #Call the FL state boundaries data for our map  > states <- map\_data("state")  > FLstate <- subset(states, region=="florida")  >  > #Create a base map of FL  > FLbase <- ggplot(data=FLstate, mapping=aes(x=long, y=lat)) + coord\_fixed(1.3) +  geom\_polygon(color="black", fill="cornsilk2") + theme\_nothing()  > FLbase  >  > #Plot points on the "FLbase" map for all of the schools being assessed  > map.all.schools <- FLbase + geom\_point(data=BaseNew, aes(x=longitude, y=latitude)) +  ggtitle("title")  > map.all.schools  >  > #Merge the "ExcellentSchools" and "zipcode" datasets to obtain latlon info  > ExcellentSchoolsNew <- merge(ExcellentSchools, zipcode, by.x="ZipCode", by.y="zip",  all.x=TRUE, all.y=FALSE)  >  > #Plot points on the "FLbase" map for all of the schools that were excelling  > map.excellent.schools <- FLbase + geom\_point(data=ExcellentSchoolsNew, aes(x=longitude,  y=latitude))  > map.excellent.schools  >  > #Merge the "DecliningSchools" and "zipcode" datasets to obtain latlon info  > DecliningSchoolsNew <- merge(DecliningSchools, zipcode, by.x="ZipCode", by.y="zip",  all.x=TRUE, all.y=FALSE)  >  > #Plot points on the "FLbase" map for all of the schools that were declining  > map.declining.schools <- FLbase + geom\_point(data=DecliningSchoolsNew, aes(x=longitude,  y=latitude))  > map.declining.schools  >  > #Plot points on the "FLbase" map for the schools being assessed where green  represents excelling and red represents declining  > map.colors.schools <- FLbase + geom\_point(data=ExcellentSchoolsNew, aes(x=longitude,  y=latitude), color="chartreuse4") + geom\_point(data=DecliningSchoolsNew, aes(x=longitude,  y=latitude), color="brown3")  > map.colors.schools  >  > #Merge the "AllOtherSchools" and "zipcode" datasets to obtain latlon info  > AllOtherSchoolsNew <- merge(AllOtherSchools, zipcode, by.x="ZipCode", by.y="zip",  all.x=TRUE, all.y=FALSE)  >  > #Plot points on the "FLbase" map for all of the schools that were neither excelling  nor declining  > map.allother.schools <- FLbase+ geom\_point(data=AllOtherSchoolsNew, aes(x=longitude,  y=latitude))  > map.allother.schools  >  > #Show map by 2019 Grade  > ForMap <- merge(Base, zipcode, by.x="ZipCode", by.y="zip", all.x=TRUE, all.y=FALSE)  > FL\_Map <- ggplot(data=FLstate, mapping=aes(x=long, y=lat)) + coord\_fixed(1.3) +  geom\_polygon(color="black", fill="cornsilk2") + ggtitle("title") +  theme\_nothing(legend=TRUE)  > (Map\_Grades <- FL\_Map + geom\_point(data=ForMap, aes(x=longitude, y=latitude,  color=`2019`)) + ggtitle("Schools labeled by their 2019 Grade") +  theme(plot.title = element\_text(hjust = 0.5))) |
|  |
| |  | | --- | | > | |
| > #DESCRIPTIVE STATISTICS ####  > #Average income by region  > avgIncome <- mean(Performance$Income)  > avgIncome  [1] 54.06825  >  > #Histogram of income  > incomeHist <- ggplot(Performance, aes(x=Income)) +  + geom\_histogram(binwidth=10, color="black", fill="gray") +  + theme(plot.title = element\_text(hjust=.5)) + labs(title = "Income Histogram")  > incomeHist  >  > #Boxplot of income  > incomeBoxplot <- ggplot(Performance, aes(x=Income, y=Region)) +  + geom\_boxplot(color="black", fill="gray") +  + theme(plot.title = element\_text(hjust=.5)) + labs(title = "Income Boxplot")  > incomeBoxplot  >  > #Avg Income by Region  > avgIncomeByRegion <- tapply(Performance$Income, Performance$Region, mean)  > avgIncomeByRegion  Region\_1 Region\_2 Region\_3 Region\_4 Region\_5  51.99250 52.34426 49.92849 55.05422 55.76911  >  > #Median Income by Region  > medianIncomeByRegion <- tapply(Performance$Income, Performance$Region, median)  > medianIncomeByRegion  Region\_1 Region\_2 Region\_3 Region\_4 Region\_5  46.98012 44.57479 43.67994 47.13404 45.83178  >  > #Histogram of students  > studentsHist <- ggplot(Performance, aes(x=Students)) +  + geom\_histogram(binwidth=50, color="black", fill="gray") +  + theme(plot.title = element\_text(hjust=.5)) + labs(title = "Students Histogram")  > studentsHist  >  > #Boxplot of students  > studentsBoxplot <- ggplot(Performance, aes(x=Students, y=Region)) +  + geom\_boxplot(color="black", fill="gray") +  + theme(plot.title = element\_text(hjust=.5)) + labs(title = "Students Boxplot")  > studentsBoxplot  >  > #Avg Students by Region  > avgStudentsByRegion <- tapply(Performance$Students, Performance$Region, mean)  > avgStudentsByRegion  Region\_1 Region\_2 Region\_3 Region\_4 Region\_5  636.1345 635.4094 679.2273 649.2584 668.3355  > avgstureg <- data.frame(Region=names(avgStudentsByRegion),Mean=avgStudentsByRegion)  > str(avgstureg)  'data.frame': 5 obs. of 2 variables:  $ Region: Factor w/ 5 levels "Region\_1","Region\_2",..: 1 2 3 4 5  $ Mean : num 636 635 679 649 668  > View(avgstureg)  >  > #Plot Avg Students by Region  > plotAvgStudentsByRegion <- ggplot(avgstureg, aes(x=Region, y=Mean)) +  + geom\_bar(position="stack", stat="identity") +  + theme(plot.title = element\_text(hjust=.5)) + labs(x="Region", y="Charter Schools",  title = "Avg. of Students by Region")  > plotAvgStudentsByRegion  >  > #Count Schools by Region  > countSchoolsByRegion <- tapply(Performance$School, Performance$Region, length)  > countSchoolsByRegion  Region\_1 Region\_2 Region\_3 Region\_4 Region\_5  119 127 242 445 605  >  > #Count Charter Schools by Region  > countCharterByRegion <- tapply(Performance$School, list(Performance$Region,  Performance$Charter=="YES"), length)  > countCharterByRegion  FALSE TRUE  Region\_1 117 2  Region\_2 125 2  Region\_3 239 3  Region\_4 434 11  Region\_5 585 20  >  > #Plot Charter Schools by Region  > plotCharterByRegion <- ggplot(Performance, aes(fill=Charter, x=Region, y=Charter)) +  + geom\_bar(position="stack", stat="identity") +  + theme(plot.title = element\_text(hjust=.5)) + labs(x="Region", y="Charter Schools",  title = "Charter Schools by Region")  > plotCharterByRegion  >  > #Avg % Economically Disadvantaged by Region  > avgEconDisadvByRegion <- tapply(Performance$Percent\_Econ\_Disadv, Performance$Region, mean)  > avgEconDisadvByRegion  Region\_1 Region\_2 Region\_3 Region\_4 Region\_5  74.42689 79.03150 79.31942 75.64562 74.35686  >  > #Boxplot of % Economically Disadvantaged  > disadvantagedBoxplot <- ggplot(Performance, aes(x=Percent\_Econ\_Disadv, y=Region)) +  + geom\_boxplot(color="black", fill="gray") +  + theme(plot.title = element\_text(hjust=.5)) + labs(title = "% Economically  Disadvantaged Boxplot")  > disadvantagedBoxplot  >  > (Region2\_Focus <- Performance %>%  + filter(Region=="Region\_2") %>%  + mutate(Excellent=case\_when(Label=="Excellent"~"Excellent Schools",TRUE~"All Other  Schools")) %>%  + ggplot(aes(x=Percent\_Econ\_Disadv,y=Excellent,fill=Excellent)) + geom\_boxplot() +  theme\_fivethirtyeight() +  + theme(plot.title = element\_text(hjust=.5),axis.title.y=element\_blank(),  axis.text.y=element\_text(size=10),legend.position = "none") + labs(title =  "% Economically Disadvantaged Boxplot\nfor Region 2\nExcellent Schools v.  All Other Schools")) |
|  |
| |  | | --- | | > | |