# 664 Group 2 Final Project - Modeling Code

Group 2

2023-05-05

## R Code Details

This deliverable provides the code used to perform the analysis of the data. More details are provided in this document than exist in the paper as some analysis was excluded for brevity. Some column name were shortened to improve the graphs as labels were very long.

```
df <- read.csv('/home/rmaxseiner/DataspellProjects/pythonProject/664FinalProject/schooldigger_2022_Virg
colnames(df)[colnames(df) =="Expenditures.for.Public.Elementary.and.Secondary.Schools.per.Pupil"] <- "E
colnames(df)[colnames(df) =="Average.Standard.Score..2021.22."] <- "Standard.Score"
colnames(df)
```

```
##  [1] "County"                          "Total.Number.Students"
##  [3] "Percent.African.American"        "Percent.American.Indian"
##  [5] "Percent.Asian"                   "Percent.Hispanic"
##  [7] "Percent.Pacific.Islander"        "Percent.Two.or.More.Races"
##  [9] "Percent.White"                   "Standard.Score"
## [11] "Average.Standard.Score..2020.21." "Rank..2020.21."
## [13] "Rank.Change.from..2020.21."       "Percent.Free.Disc.Lunch"
## [15] "SchoolDigger.Star.Rating"         "Student.Teacher.Ratio"
## [17] "Number.Full.time.Teachers"        "Is.Title.I.."
## [19] "Is.Charter.."                     "Is.Magnet.."
## [21] "Is.Virtual.."                     "Expenditures.per.Pupil"
## [23] "Income.by.School.District"
```

### PCA Analysis

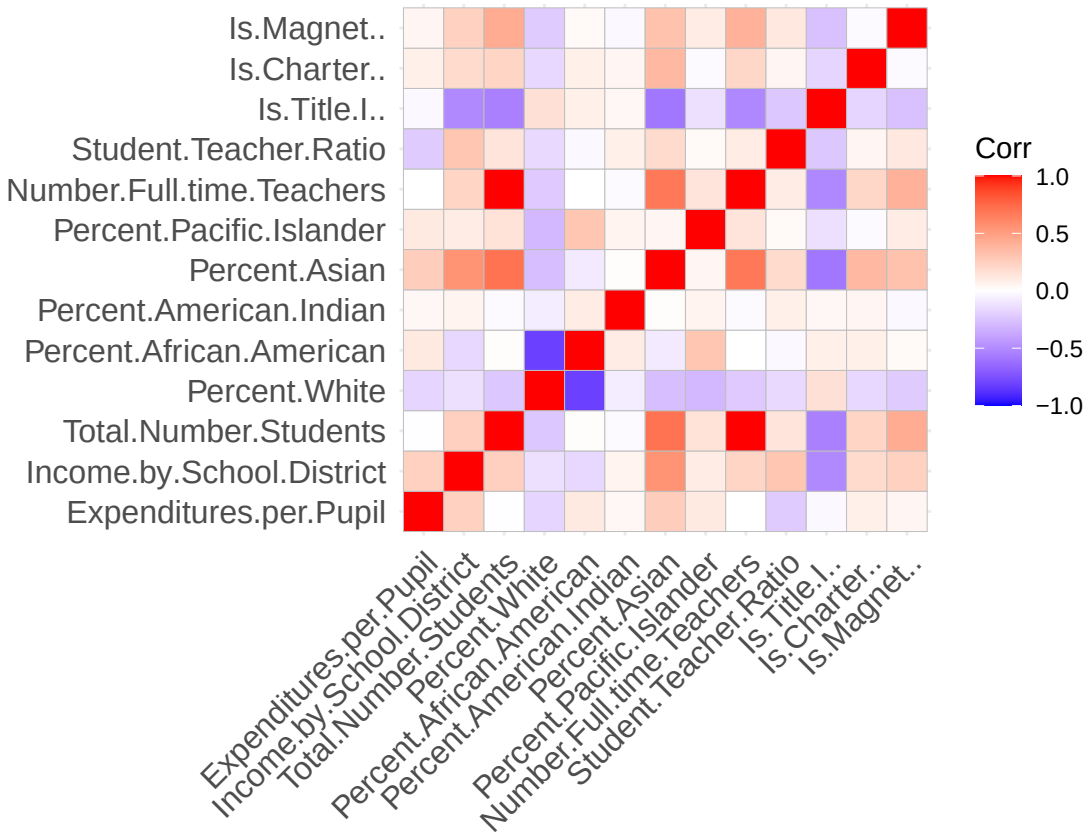
PCA was performed to determine if there was a way to improve the linear regression by reducing the number of predictor variables. While some reduction is possible overall the PCA confirmed that the data was not heavily cross correlated.

```
# PCA
# Define the response and predictor variables

library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
predictor_vars <- c("Expenditures.per.Pupil","Income.by.School.District",
                    "Total.Number.Students","Percent.White",
                    "Percent.African.American",
                    "Percent.American.Indian",
                    "Percent.Asian",
                    "Percent.Pacific.Islander",
                    "Number.Full.time.Teachers",
                    "Student.Teacher.Ratio",
```

```
                    "Is.Title.I..",
                    "Is.Charter..",
                    "Is.Magnet..")

# Create a data frame with the predictor and response variables
predictor_df <- data.frame(df[, predictor_vars])
data_scaled <- scale(predictor_df)


data_normalized <- scale(data_scaled)

corr_matrix <- cor(data_normalized)
ggcorrplot(corr_matrix)
```




```
data.pca <- princomp(corr_matrix)
summary(data.pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     0.917262 0.5467979 0.35336543 0.33786632 0.29244584
## Proportion of Variance 0.516793 0.1836468 0.07669693 0.07011641 0.05253162
## Cumulative Proportion  0.516793 0.7004399 0.77713678 0.84725319 0.89978481
##                           Comp.6     Comp.7     Comp.8      Comp.9     Comp.10
## Standard deviation     0.24546744 0.22787615 0.16492275 0.109808870 0.091981068
## Proportion of Variance 0.03700988 0.03189537 0.01670671 0.007406358 0.005196689
## Cumulative Proportion  0.93679468 0.96869005 0.98539676 0.992803123 0.997999812
##                           Comp.11      Comp.12      Comp.13
```

```
## Standard deviation     0.057058114 8.917170e-04 5.580436e-09
## Proportion of Variance 0.001999699 4.884093e-07 1.912785e-17
## Cumulative Proportion  0.999999512 1.000000e+00 1.000000e+00
```

```
data.pca$loadings[, 1:8]
```

```
##                                Comp.1      Comp.2      Comp.3      Comp.4
## Expenditures.per.Pupil     0.03418430  0.14579092  0.57228656  0.43871432
## Income.by.School.District  0.27943293 -0.13375889  0.50446153 -0.24062806
## Total.Number.Students      0.45020207 -0.04706833 -0.30605748  0.16717189
## Percent.White             -0.20405111 -0.60054608 -0.04425312  0.11049571
## Percent.African.American   0.01759432  0.67083964 -0.06390478 -0.03505426
## Percent.American.Indian   -0.05819274  0.10328256  0.13663885 -0.26828725
## Percent.Asian              0.43704603 -0.08297020  0.19975543  0.09458339
## Percent.Pacific.Islander   0.07689281  0.33650406 -0.05746887 -0.03568987
## Number.Full.time.Teachers  0.43802594 -0.04929422 -0.31789064  0.20018249
## Student.Teacher.Ratio      0.13574739 -0.05784660 -0.01647900 -0.75114441
## Is.Title.I..              -0.42355060  0.10161472 -0.13164603  0.13241362
## Is.Charter..               0.15196217  0.02922427  0.27717607  0.05758531
## Is.Magnet..                0.25105882  0.01465032 -0.24685009  0.03008510
##                                Comp.5      Comp.6       Comp.7      Comp.8
## Expenditures.per.Pupil     0.26369667  0.05726453  0.155098818  0.27911107
## Income.by.School.District  0.24534741 -0.04762211 -0.004313756 -0.14250353
## Total.Number.Students     -0.10212609  0.12487813 -0.050094455  0.17312739
## Percent.White              0.05057264  0.09900126 -0.201304525 -0.09236556
## Percent.African.American  -0.11182108 -0.07499085  0.120856763  0.06402914
## Percent.American.Indian   -0.18986539  0.90538120  0.097890565 -0.13290422
## Percent.Asian             -0.09558445  0.02895349  0.052950973  0.15499301
## Percent.Pacific.Islander   0.37099763  0.05428495 -0.757999016 -0.23369779
## Number.Full.time.Teachers -0.11473038  0.14179090 -0.055465610  0.20376775
## Student.Teacher.Ratio      0.02158604 -0.22184561  0.084612306  0.31236734
## Is.Title.I..              -0.08138205 -0.06281921  0.195900798  0.06509376
## Is.Charter..              -0.69269618 -0.25664983 -0.119701722 -0.48753328
## Is.Magnet..                0.40233429 -0.02824329  0.518069731 -0.61795142
```

### Full Linear Regression

This section provides details on a linear regression with all numerical attributes. Since there are a number of attributes that were not statistically significant, several ways of removing these were investigated. Below we have attempted to use ridge and lasso to eliminate predictor variables. In the end the best method was just to remove variable from the

```
# Define the response and predictor variables
response_var <- "Standard.Score"
predictor_vars <- c("Expenditures.per.Pupil","Income.by.School.District",
                    "Total.Number.Students","Percent.White",
                    "Percent.African.American",
                    "Percent.American.Indian",
                    "Percent.Asian",
                    "Percent.Pacific.Islander",
                    "Number.Full.time.Teachers",
                    "Student.Teacher.Ratio",
                    "Is.Title.I..",
                    "Is.Charter..",
                    "Is.Magnet..",
```

```
                    "Is.Virtual..")

# Define the formula for the linear regression
formula <- as.formula(paste(response_var, paste(predictor_vars, collapse = "+"), sep = " ~ "))

# Create a data frame with the predictor and response variables
df1 <- data.frame(df[, predictor_vars], df[, response_var])
colnames(df1)[length(predictor_vars) + 1] <- response_var

model_full <- lm(formula, data = df1)
summary(model_full)
```

```
##
## Call:
## lm(formula = formula, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.062  -8.174  -0.005   9.306  28.356
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.216e+00  2.072e+01   0.348 0.728310
## Expenditures.per.Pupil    -1.102e-03  6.801e-04  -1.621 0.107798
## Income.by.School.District  2.654e-04  7.647e-05   3.470 0.000732 ***
## Total.Number.Students     -1.111e-03  2.240e-03  -0.496 0.620831
## Percent.White              7.808e-01  1.157e-01   6.746 6.28e-10 ***
## Percent.African.American   3.175e-01  1.331e-01   2.387 0.018624 *
## Percent.American.Indian   -2.997e+00  2.740e+00  -1.094 0.276326
## Percent.Asian              1.207e+00  6.880e-01   1.754 0.082110 .
## Percent.Pacific.Islander   5.071e-01  9.204e+00   0.055 0.956153
## Number.Full.time.Teachers  1.678e-02  3.023e-02   0.555 0.579908
## Student.Teacher.Ratio     -3.380e-01  7.809e-01  -0.433 0.665977
## Is.Title.I..              -1.336e+01  7.165e+00  -1.865 0.064656 .
## Is.Charter..              -3.052e+02  3.002e+02  -1.016 0.311564
## Is.Magnet..                1.098e+01  1.582e+01   0.694 0.488901
## Is.Virtual..                      NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.62 on 116 degrees of freedom
## Multiple R-squared:  0.5724, Adjusted R-squared:  0.5245
## F-statistic: 11.94 on 13 and 116 DF,  p-value: 4.654e-16
```

```
model_output <- capture.output(summary(model_full))
writeLines(model_output,"linear_regression_output1.txt")
```

### Reduction of Features

```
# Define the response and predictor variables
response_var <- "Standard.Score"
predictor_vars <- c("Expenditures.per.Pupil","Income.by.School.District",
                    "Total.Number.Students","Percent.White",
                    "Percent.African.American",
```

```
                    "Percent.American.Indian",
                    "Percent.Asian",
                    "Percent.Pacific.Islander",
                    "Number.Full.time.Teachers",
                    "Student.Teacher.Ratio",
                    "Is.Title.I..",
                    "Is.Charter..",
                    "Is.Magnet..",
                    "Is.Virtual..")

# Define the formula for the linear regression
formula <- as.formula(paste(response_var, paste(predictor_vars, collapse = "+"), sep = " ~ "))

# Create a data frame with the predictor and response variables
df1 <- data.frame(df[, predictor_vars], df[, response_var])
colnames(df1)[length(predictor_vars) + 1] <- response_var
model_full <- lm(formula, data = df1)

# View the model summary
summary(model_full)
```

```
##
## Call:
## lm(formula = formula, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.062  -8.174  -0.005   9.306  28.356
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.216e+00  2.072e+01   0.348 0.728310
## Expenditures.per.Pupil    -1.102e-03  6.801e-04  -1.621 0.107798
## Income.by.School.District  2.654e-04  7.647e-05   3.470 0.000732 ***
## Total.Number.Students     -1.111e-03  2.240e-03  -0.496 0.620831
## Percent.White              7.808e-01  1.157e-01   6.746 6.28e-10 ***
## Percent.African.American   3.175e-01  1.331e-01   2.387 0.018624 *
## Percent.American.Indian   -2.997e+00  2.740e+00  -1.094 0.276326
## Percent.Asian              1.207e+00  6.880e-01   1.754 0.082110 .
## Percent.Pacific.Islander   5.071e-01  9.204e+00   0.055 0.956153
## Number.Full.time.Teachers  1.678e-02  3.023e-02   0.555 0.579908
## Student.Teacher.Ratio     -3.380e-01  7.809e-01  -0.433 0.665977
## Is.Title.I..              -1.336e+01  7.165e+00  -1.865 0.064656 .
## Is.Charter..              -3.052e+02  3.002e+02  -1.016 0.311564
## Is.Magnet..                1.098e+01  1.582e+01   0.694 0.488901
## Is.Virtual..                      NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.62 on 116 degrees of freedom
## Multiple R-squared:  0.5724, Adjusted R-squared:  0.5245
## F-statistic: 11.94 on 13 and 116 DF,  p-value: 4.654e-16
```

```r
##############################################################################
# remove "Is Virtual" since the data is not valid
# remove "Number.Full.time.Teachers", "Student.Teacher.Ratio" and
#         "Percent.Pacific.Islander"since statistically they are not valid
# Define the response and predictor variables
response_var <- "Standard.Score"
predictor_vars <- c("Expenditures.per.Pupil","Income.by.School.District",
                    "Total.Number.Students","Percent.White",
                    "Percent.African.American",
                    "Percent.American.Indian",
                    "Percent.Asian",
                    "Is.Title.I..",
                    "Is.Charter..",
                    "Is.Magnet..")

# Define the formula for the linear regression
formula <- as.formula(paste(response_var, paste(predictor_vars, collapse = "+"), sep = " ~ "))

# Create a data frame with the predictor and response variables
df1 <- data.frame(df[, predictor_vars], df[, response_var])
colnames(df1)[length(predictor_vars) + 1] <- response_var
model_full <- lm(formula, data = df1)

# View the model summary
summary(model_full)
```

```
##
## Call:
## lm(formula = formula, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.935  -9.080   0.129   8.766  29.470
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -9.410e-01  1.454e+01  -0.065  0.94849
## Expenditures.per.Pupil    -8.931e-04  6.262e-04  -1.426  0.15641
## Income.by.School.District  2.495e-04  7.317e-05   3.409  0.00089 ***
## Total.Number.Students      1.315e-04  2.045e-04   0.643  0.52146
## Percent.White              7.899e-01  1.111e-01   7.110  9.3e-11 ***
## Percent.African.American   3.192e-01  1.279e-01   2.497  0.01390 *
## Percent.American.Indian   -3.089e+00  2.699e+00  -1.145  0.25468
## Percent.Asian              1.128e+00  6.579e-01   1.715  0.08904 .
## Is.Title.I..              -1.219e+01  6.899e+00  -1.767  0.07980 .
## Is.Charter..              -3.276e+02  2.927e+02  -1.119  0.26529
## Is.Magnet..                7.167e+00  1.417e+01   0.506  0.61391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.49 on 119 degrees of freedom
## Multiple R-squared:  0.5695, Adjusted R-squared:  0.5333
## F-statistic: 15.74 on 10 and 119 DF,  p-value: < 2.2e-16
```

```r
##############################################################################
# not much shift in statistical significance
# remove "Is.Charter..","Is.Magnet..", "Total.Number.Students",
# and "Percent.American.Indian"

# Define the response and predictor variables
response_var <- "Standard.Score"
predictor_vars <- c("Expenditures.per.Pupil",
                    "Income.by.School.District",
                    "Percent.White",
                    "Percent.African.American",
                    "Percent.Asian",
                    "Is.Title.I..")

# Define the formula for the linear regression
formula <- as.formula(paste(response_var, paste(predictor_vars, collapse = "+"), sep = " ~ "))

# Create a data frame with the predictor and response variables
df1 <- data.frame(df[, predictor_vars], df[, response_var])
colnames(df1)[length(predictor_vars) + 1] <- response_var
model_full <- lm(formula, data = df1)

# View the model summary
summary(model_full)
```

```
##
## Call:
## lm(formula = formula, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.497  -9.393   0.812   8.658  28.526
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                5.395e+00  1.377e+01   0.392  0.69596
## Expenditures.per.Pupil    -9.588e-04  6.100e-04  -1.572  0.11855
## Income.by.School.District  2.260e-04  6.972e-05   3.241  0.00153 **
## Percent.White              7.632e-01  1.093e-01   6.981 1.61e-10 ***
## Percent.African.American   2.823e-01  1.253e-01   2.252  0.02609 *
## Percent.Asian              1.198e+00  5.048e-01   2.373  0.01917 *
## Is.Title.I..              -1.482e+01  6.590e+00  -2.249  0.02630 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.49 on 123 degrees of freedom
## Multiple R-squared:  0.555,  Adjusted R-squared:  0.5333
## F-statistic: 25.57 on 6 and 123 DF,  p-value: < 2.2e-16
```

```r
##############################################################################
# not much shift in statistical significance
# remove "Is.Charter..","Is.Magnet..", "Total.Number.Students",
# and "Percent.American.Indian"
```

```r
# Define the response and predictor variables
response_var <- "Percent.African.American"
predictor_vars <- c("Income.by.School.District", "Percent.Asian")

# Define the formula for the linear regression
formula <- as.formula(paste(response_var, paste(predictor_vars, collapse = "+"), sep = " ~ "))

# Create a data frame with the predictor and response variables
df1 <- data.frame(df[, predictor_vars], df[, response_var])
colnames(df1)[length(predictor_vars) + 1] <- response_var
model_full <- lm(formula, data = df1)

# View the model summary
summary(model_full)
```

```
##
## Call:
## lm(formula = formula, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.180 -16.153  -5.645  10.705  63.960
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.127e+01  5.854e+00   5.341 4.12e-07 ***
## Income.by.School.District -1.644e-04  9.774e-05  -1.682   0.0951 .
## Percent.Asian              5.157e-02  6.151e-01   0.084   0.9333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.86 on 127 degrees of freedom
## Multiple R-squared:  0.02919,    Adjusted R-squared:  0.0139
## F-statistic: 1.909 on 2 and 127 DF,  p-value: 0.1524
```

```r
################################################################################
# not much shift in statistical significance
# Thsi is to determine the relationship between Expenditures and Income

# Define the response and predictor variables
response_var <- "Expenditures.per.Pupil"
predictor_vars <- c("Income.by.School.District")

# Define the formula for the linear regression
formula <- "Expenditures.per.Pupil ~ Income.by.School.District"

# Create a data frame with the predictor and response variables
df1 <- data.frame(df[, predictor_vars, drop=FALSE], df[, response_var])
colnames(df1)[length(predictor_vars) + 1] <- response_var
model_full <- lm(formula, data = df1)

# View the model summary
summary(model_full)
```

```
##
## Call:
## lm(formula = formula, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3172.7 -1308.1  -451.5   665.0  8391.2
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.105e+04  5.366e+02   20.60  < 2e-16 ***
## Income.by.School.District 2.276e-02  8.044e-03    2.83  0.00541 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2048 on 128 degrees of freedom
## Multiple R-squared:  0.05887,    Adjusted R-squared:  0.05151
## F-statistic: 8.006 on 1 and 128 DF,  p-value: 0.005414
```

### Linear Regression with statistically significant features

This section provides details on a linear regression with all numerical attributes. This is the basis for much of the other analysis.

```r
# Define the response and predictor variables
response_var <- "Standard.Score"
predictor_vars <- c("Income.by.School.District",
                    "Percent.White",
                    "Percent.African.American",
                    "Percent.Asian",
                    "Is.Title.I..")

# Define the formula for the linear regression
formula <- as.formula(paste(response_var, paste(predictor_vars, collapse = "+"), sep = " ~ "))

# Create a data frame with the predictor and response variables
df1 <- data.frame(df[, predictor_vars], df[, response_var])
colnames(df1)[length(predictor_vars) + 1] <- response_var


# View the mean squared errors for each fold

model_full <- lm(formula, data = df1)
summary(model_full)
```

```
##
## Call:
## lm(formula = formula, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.427  -9.492   0.702   9.754  30.122
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
```

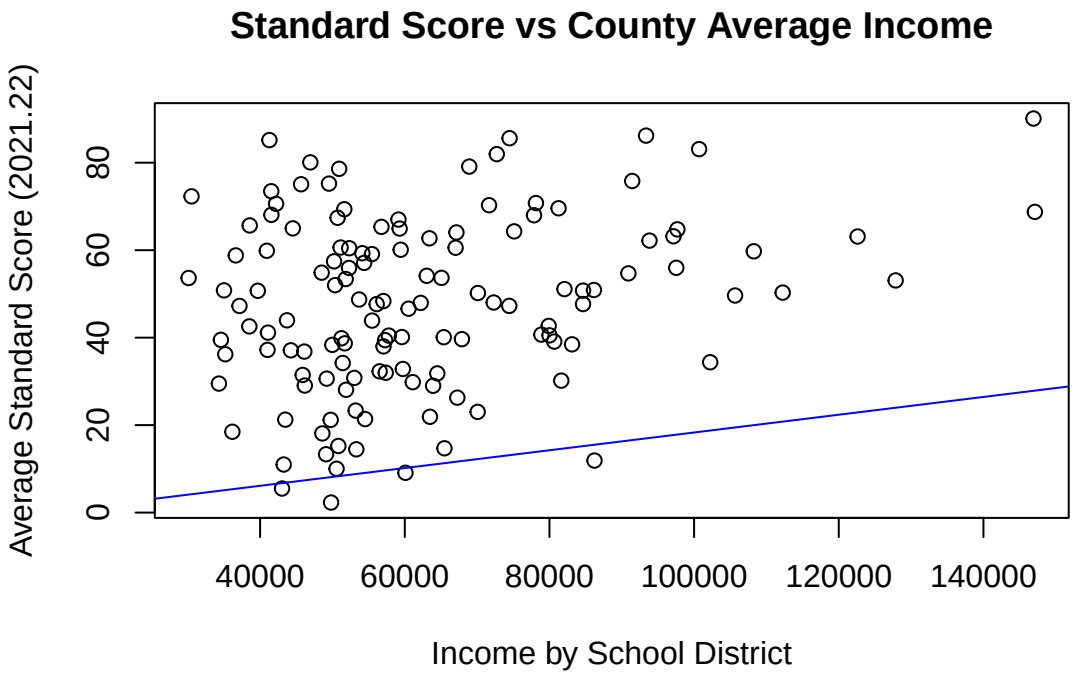
```
## (Intercept)               -1.999e+00  1.302e+01  -0.154  0.87824
## Income.by.School.District  2.031e-04  6.859e-05   2.961  0.00367 **
## Percent.White              7.546e-01  1.098e-01   6.870 2.76e-10 ***
## Percent.African.American   2.590e-01  1.252e-01   2.069  0.04066 *
## Percent.Asian              1.020e+00  4.949e-01   2.062  0.04132 *
## Is.Title.I..              -1.694e+01  6.489e+00  -2.610  0.01017 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.57 on 124 degrees of freedom
## Multiple R-squared:  0.546,  Adjusted R-squared:  0.5277
## F-statistic: 29.83 on 5 and 124 DF,  p-value: < 2.2e-16
```

```
# View the model summary
print('Create Scatter Plot')
```

**Scatter plot of Standard Score vs County Average Income**

```
## [1] "Create Scatter Plot"
```

```
plot(  df1$"Income.by.School.District" ,
  df1$"Standard.Score",
  xlab="Income by School District",
  ylab = "Average Standard Score (2021.22)",
  main = "Standard Score vs County Average Income")

abline(coef(model_full)[1], coef(model_full)[["Income.by.School.District"]], col = "blue")
```

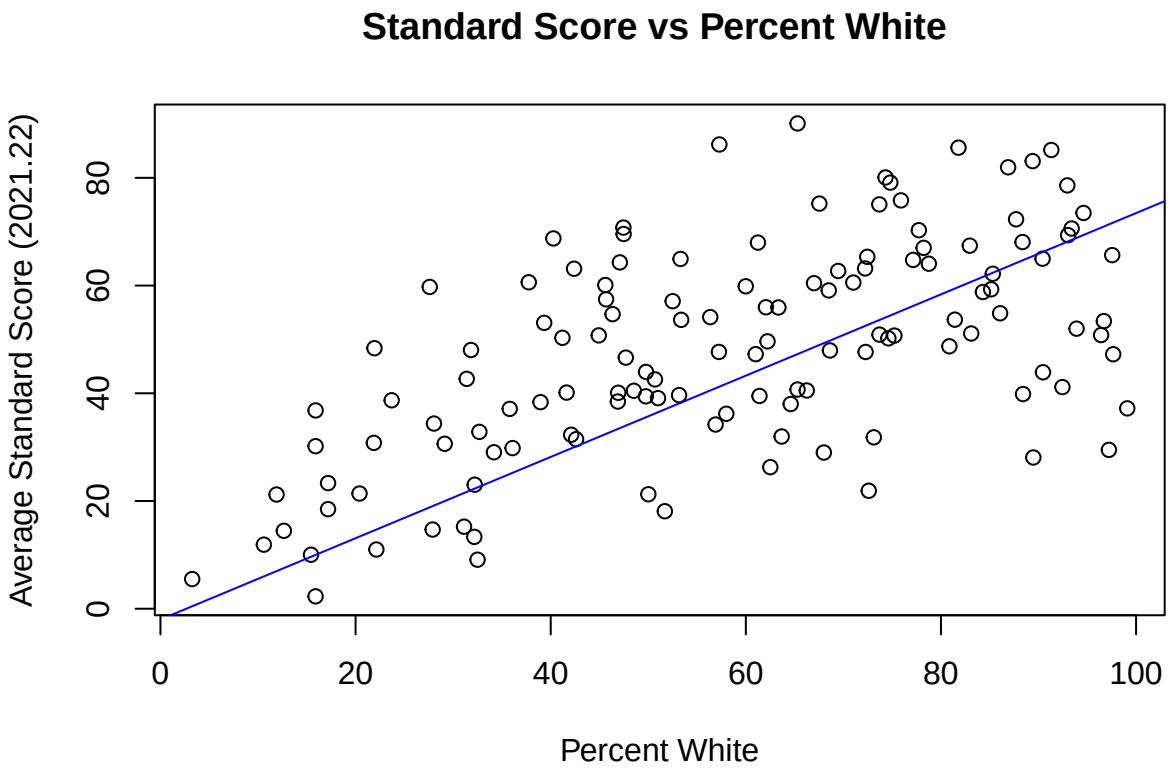



```
print('Create Scatter Plot')
```

**Scatter plot of Standard Score vs Standard Score**

```
## [1] "Create Scatter Plot"
```

```
plot(
  df1$"Percent.White" ,
  df1$"Standard.Score",
  xlab="Percent White",
  ylab = "Average Standard Score (2021.22)",
  main = "Standard Score vs Percent White ")

abline(coef(model_full)[1], coef(model_full)[["Percent.White"]], col= "blue")
```

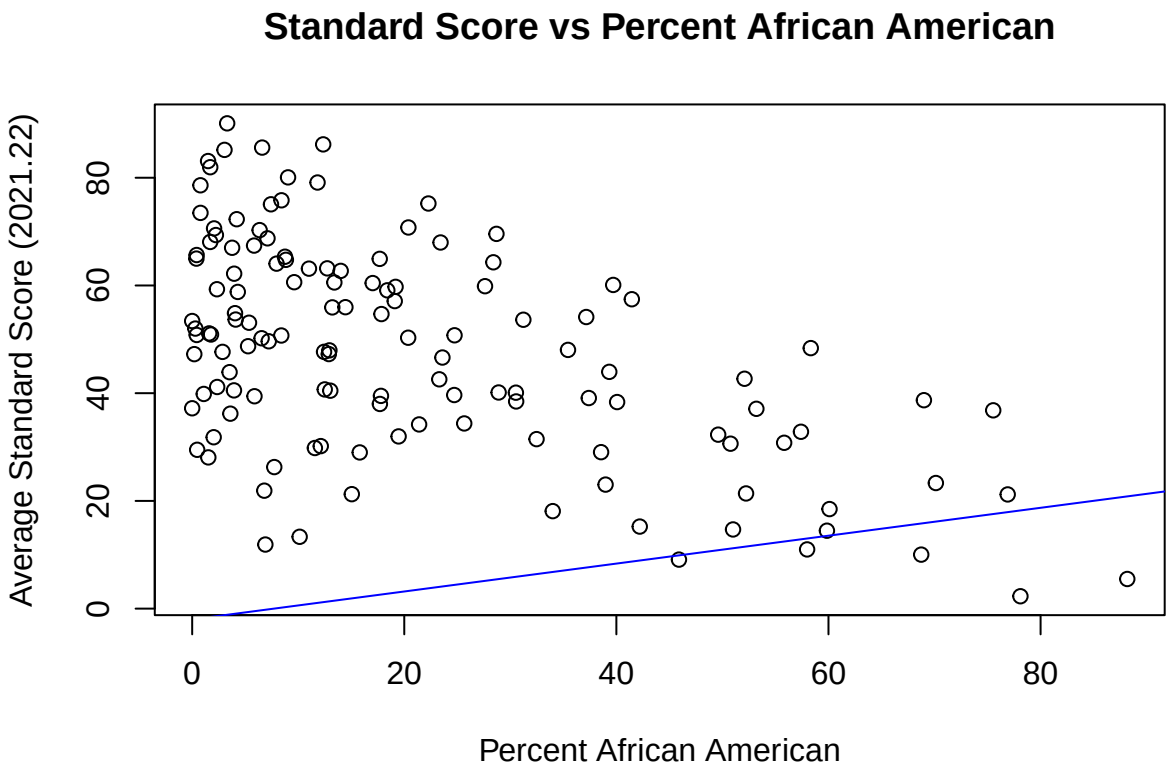



```
print('Create Scatter Plot')
```

**Scatter plot of Standard Score vs Percent African American**

```
## [1] "Create Scatter Plot"
```

```
plot(
  df1$"Percent.African.American" ,
  df1$"Standard.Score",
  xlab="Percent African American",
  ylab = "Average Standard Score (2021.22)",
  main = "Standard Score vs Percent African American")

abline(coef(model_full)[1], coef(model_full)[["Percent.African.American"]], col= "blue")
```

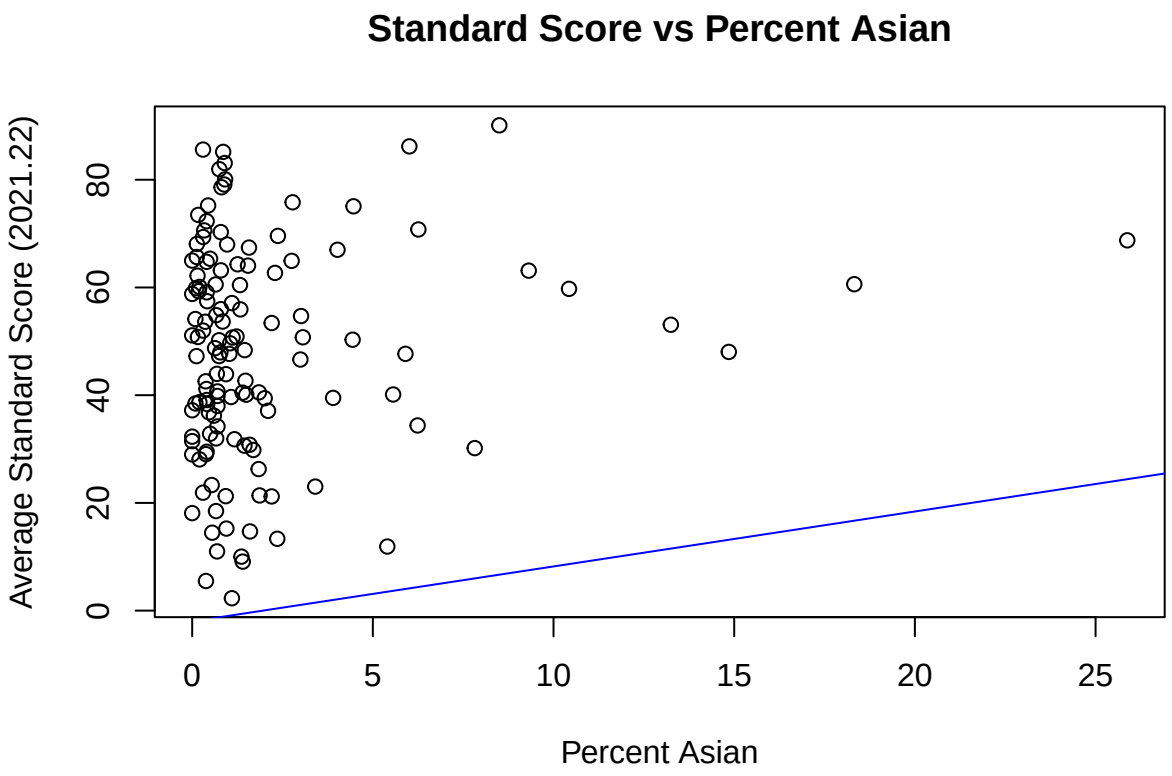



```
print('Create Scatter Plot')
```

**Scatter plot of Standard Score vs Percent Asian**

```
## [1] "Create Scatter Plot"
```

```
plot(
  df1$"Percent.Asian" ,
  df1$"Standard.Score",
  xlab="Percent Asian",
  ylab = "Average Standard Score (2021.22)",
  main = "Standard Score vs Percent Asian")

abline(coef(model_full)[1], coef(model_full)[["Percent.Asian"]], col= "blue")
```

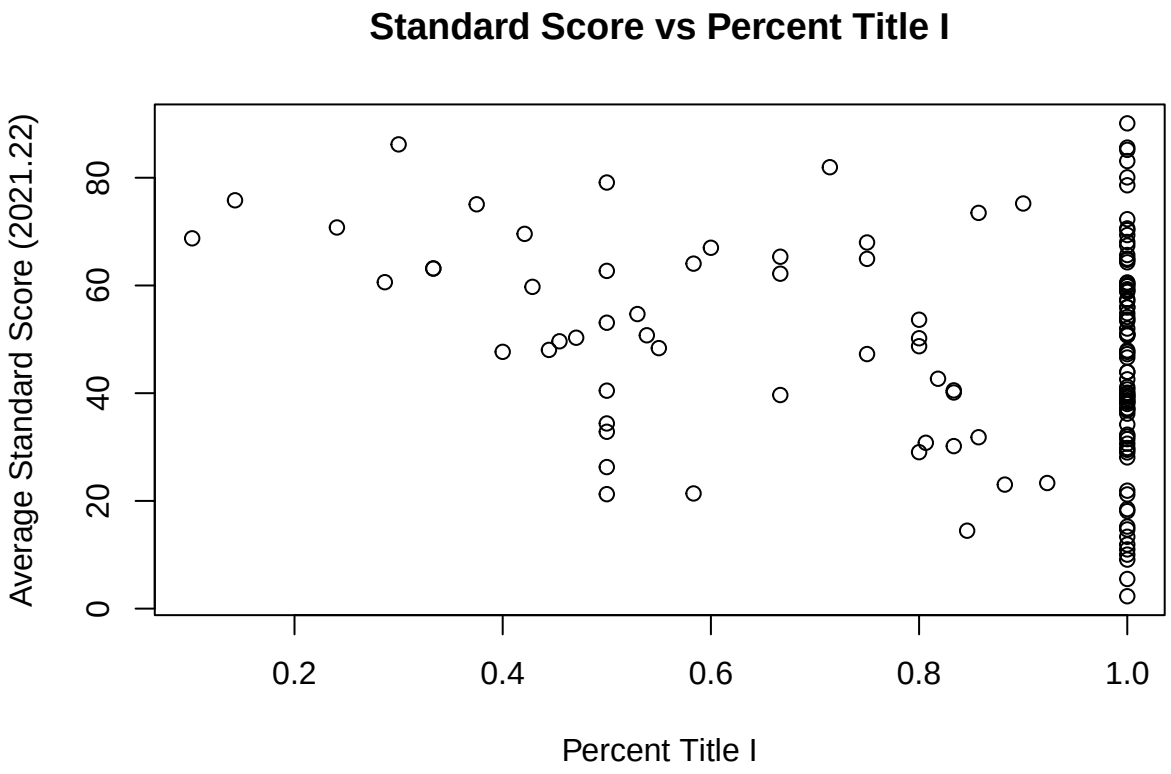



```
print('Create Scatter Plot')
```

**Scatter plot of Standard Score vs Percent Title I**

```
## [1] "Create Scatter Plot"
```

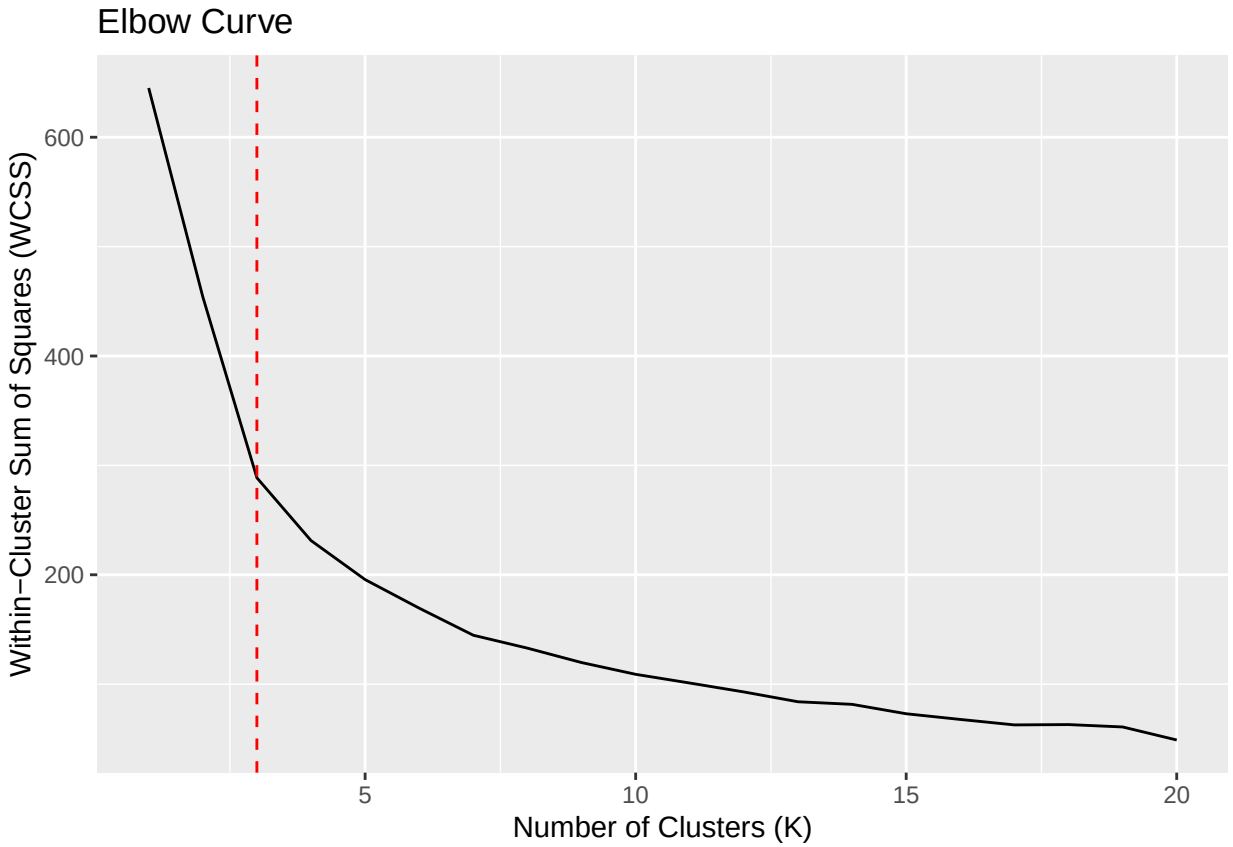
```
plot(
  df1$"Is.Title.I.." ,
  df1$"Standard.Score",
  xlab="Percent Title I",
  ylab = "Average Standard Score (2021.22)",
  main = "Standard Score vs Percent Title I")

abline(coef(model_full)[1], coef(model_full)[["Is.Title.I.."]], col= "blue")
```




**Kmeans Clustering**

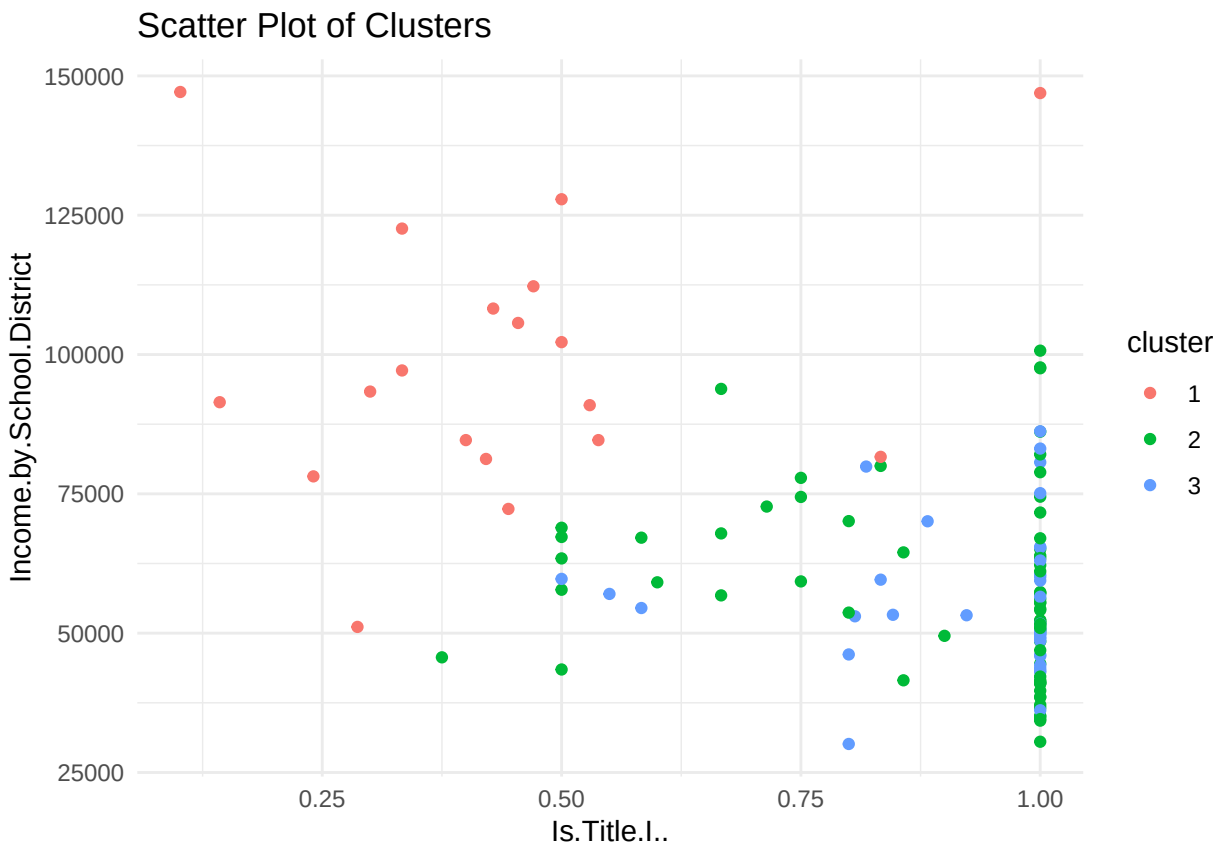
Perform K means Clustering to identify clusters. The resulting diagram doesn't have an elbow. No real clusters.

```
################################################################################

# Subset the data to include only the predictor variables
vars <- c("Income.by.School.District",
                    "Percent.White",
                    "Percent.African.American",
                    "Percent.Asian",
                    "Is.Title.I..")
cluster_df <- df[, vars]

# Standardize the data
data_scaled <- scale(cluster_df)

# Run K-means clustering for k = 1 to 10
wcss <- sapply(1:20, function(k){kmeans(data_scaled, k, nstart=20)$tot.withinss})

# Plot the elbow curve
plot_data <- data.frame(K = 1:20, WCSS = wcss)

# Create the plot
ggplot(plot_data, aes(x = K, y = WCSS)) +
  geom_line() +
  geom_vline(xintercept = 3, linetype = "dashed", color = "red") +
  labs(title = "Elbow Curve", x = "Number of Clusters (K)", y = "Within-Cluster Sum of Squares (WCSS)")
```

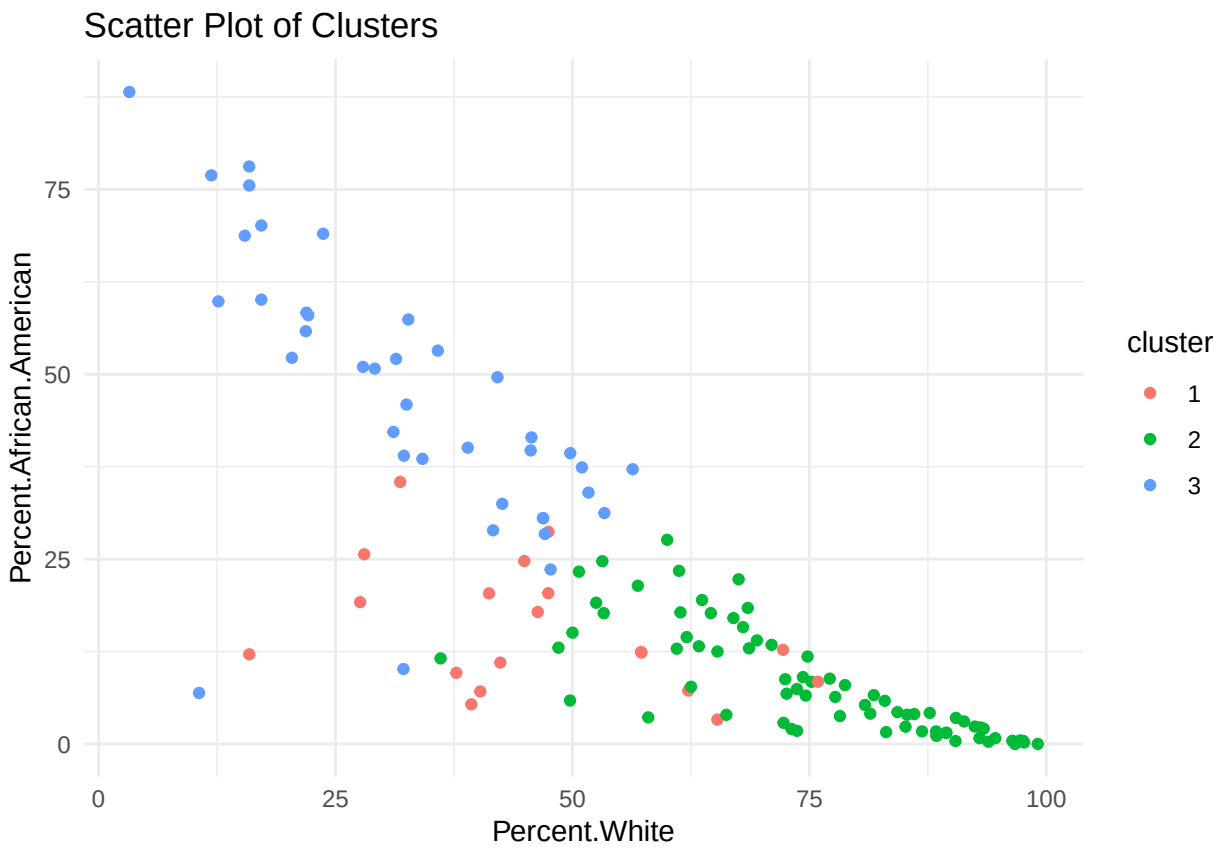



```
# Assuming you've chosen the optimal k based on the elbow curve
optimal_k <- 3 # Replace 3 with the optimal k value you've determined from the elbow curve

# Perform k-means clustering using the optimal k
kmeans_result <- kmeans(data_scaled, centers = optimal_k, nstart = 20)

# Add cluster assignment to the original data frame
df$cluster <- as.factor(kmeans_result$cluster)

# Choose variable pairs for the scatter plot (replace these with your desired variable pairs)
var1 <- "Is.Title.I.."
var2 <- "Income.by.School.District"

# Create the scatter plot using ggplot2
ggplot(df, aes_string(x = var1, y = var2, color = "cluster")) +
  geom_point() +
  labs(x = var1, y = var2) +
  ggtitle("Scatter Plot of Clusters") +
  theme_minimal()
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

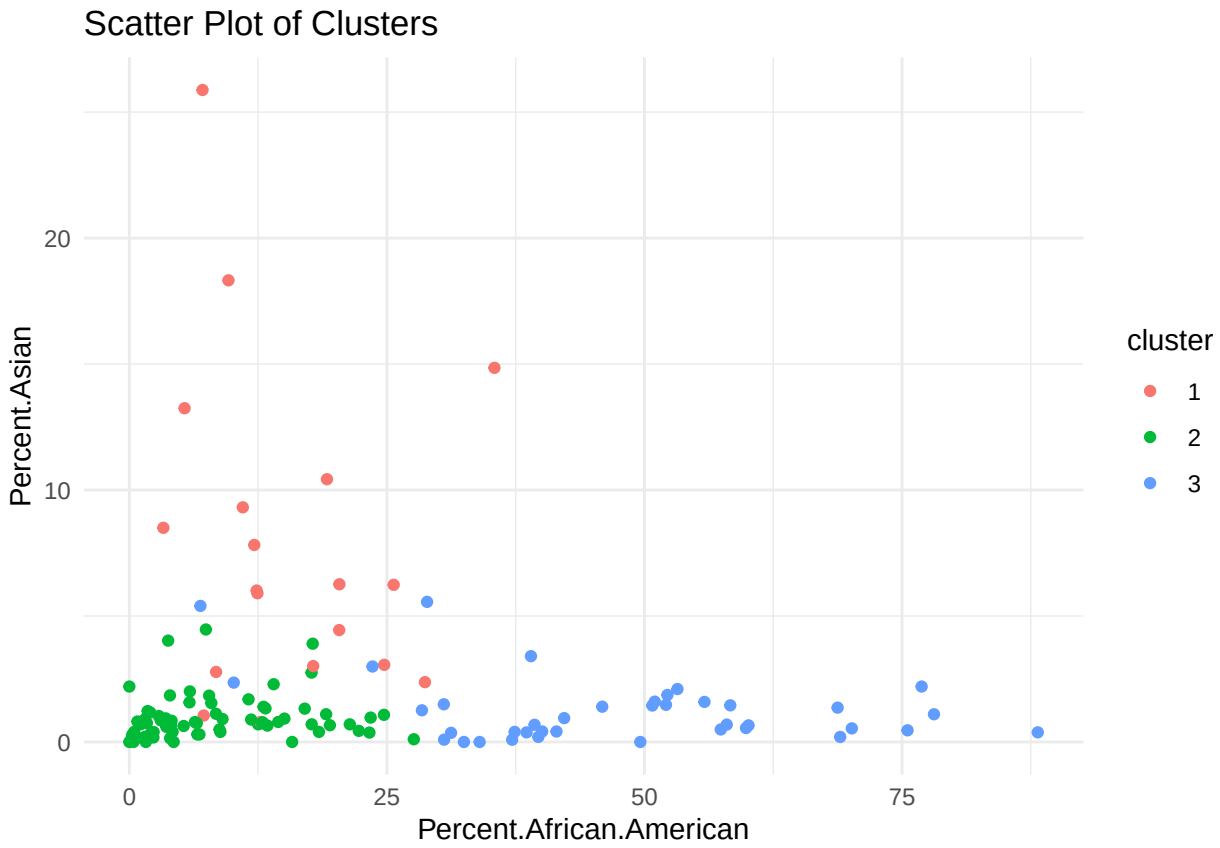



```r
# Choose variable pairs for the scatter plot (replace these with your desired variable pairs)
var1 <- "Percent.White"
var2 <- "Percent.African.American"

# Create the scatter plot using ggplot2
ggplot(df, aes_string(x = var1, y = var2, color = "cluster")) +
  geom_point() +
  labs(x = var1, y = var2) +
  ggtitle("Scatter Plot of Clusters") +
  theme_minimal()
```

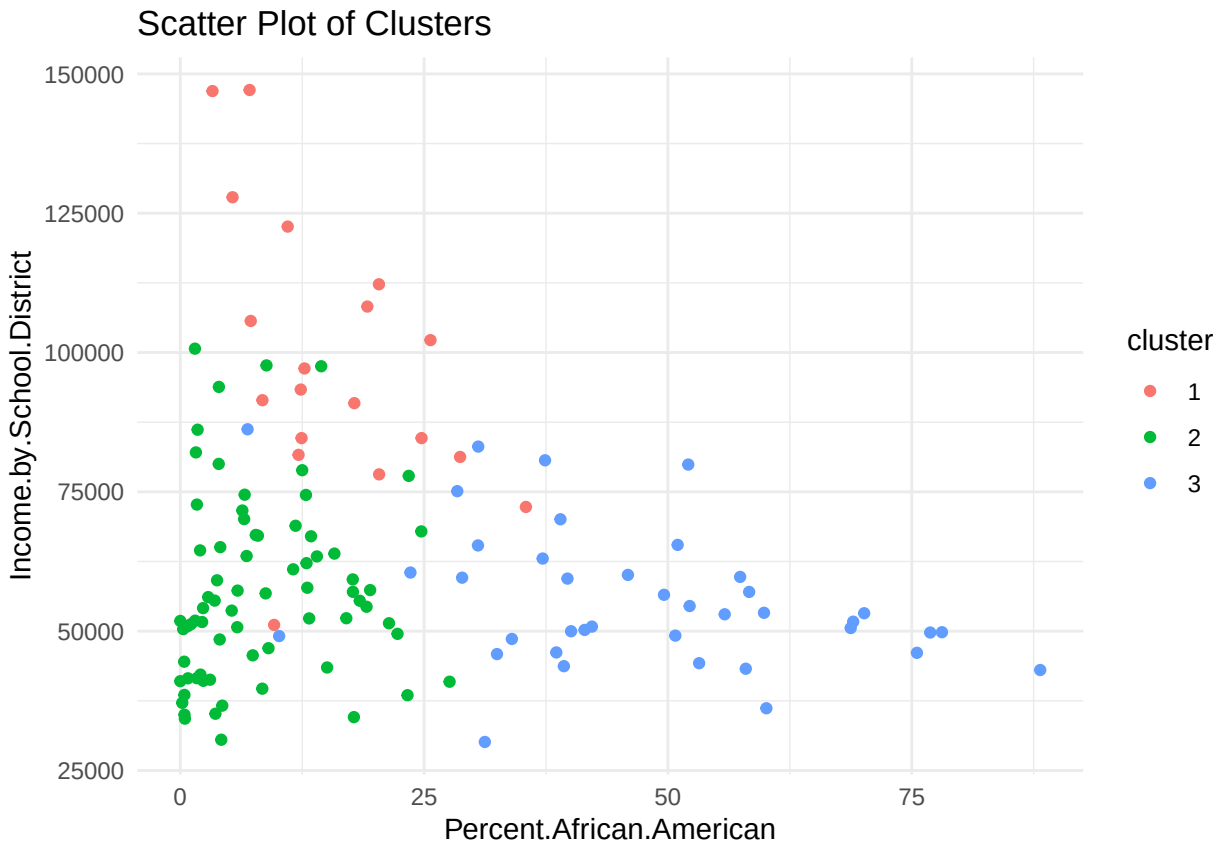



```
# Choose variable pairs for the scatter plot (replace these with your desired variable pairs)
var1 <- "Percent.White"
var2 <- "Percent.Asian"

# Create the scatter plot using ggplot2
ggplot(df, aes_string(x = var1, y = var2, color = "cluster")) +
  geom_point() +
  labs(x = var1, y = var2) +
  ggtitle("Scatter Plot of Clusters") +
  theme_minimal()
```


```
# Choose variable pairs for the scatter plot (replace these with your desired variable pairs)
var1 <- "Percent.African.American"
var2 <- "Percent.Asian"

ggplot(df, aes_string(x = var1, y = var2, color = "cluster")) +
  geom_point() +
  labs(x = var1, y = var2) +
  ggtitle("Scatter Plot of Clusters") +
  theme_minimal()
```

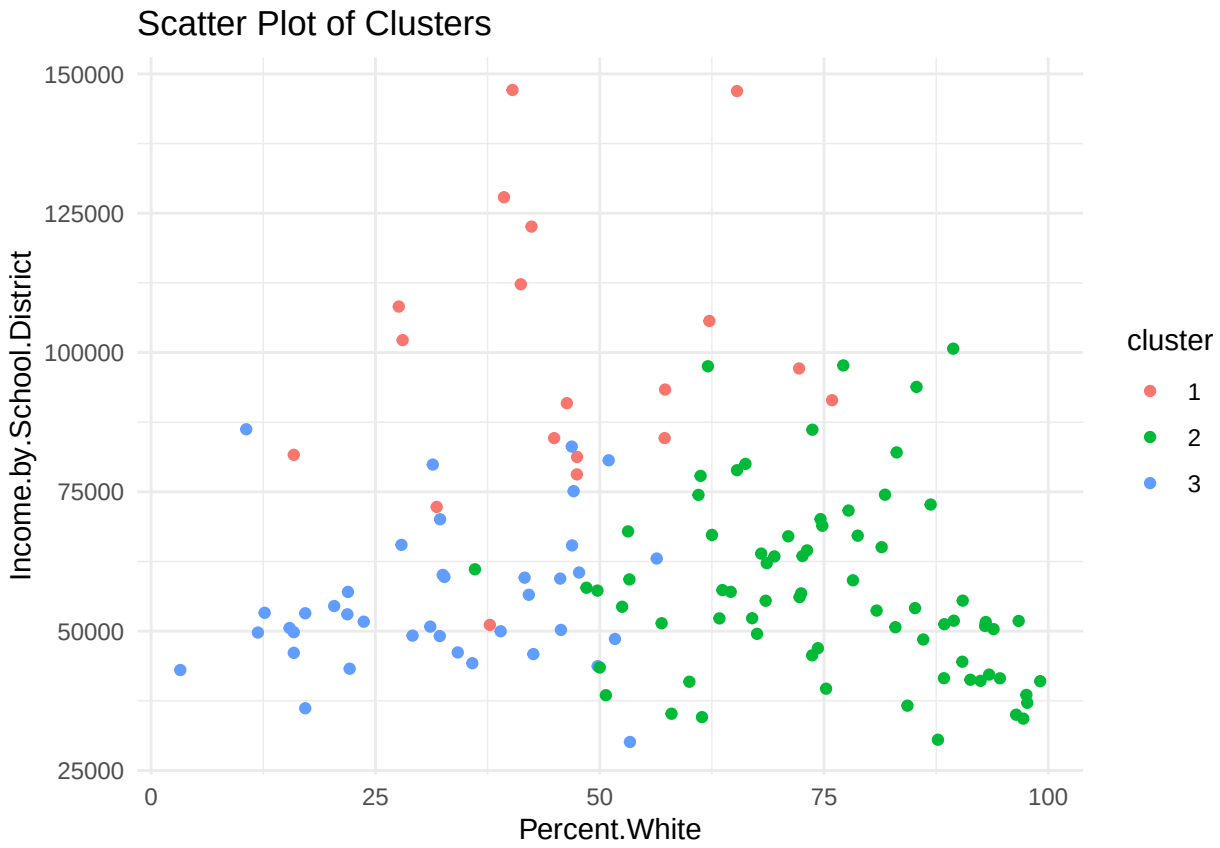



```
# Choose variable pairs for the scatter plot (replace these with your desired variable pairs)
var1 <- "Percent.African.American"
var2 <- "Income.by.School.District"

ggplot(df, aes_string(x = var1, y = var2, color = "cluster")) +
  geom_point() +
  labs(x = var1, y = var2) +
  ggtitle("Scatter Plot of Clusters") +
  theme_minimal()
```




```
# Choose variable pairs for the scatter plot (replace these with your desired variable pairs)
var1 <- "Percent.Asian"
var2 <- "Income.by.School.District"

ggplot(df, aes_string(x = var1, y = var2, color = "cluster")) +
  geom_point() +
  labs(x = var1, y = var2) +
  ggtitle("Scatter Plot of Clusters") +
  theme_minimal()
```


```
# Choose variable pairs for the scatter plot (replace these with your desired variable pairs)
var1 <- "Percent.White"
var2 <- "Income.by.School.District"

ggplot(df, aes_string(x = var1, y = var2, color = "cluster")) +
  geom_point() +
  labs(x = var1, y = var2) +
  ggtitle("Scatter Plot of Clusters") +
  theme_minimal()
```




## Ridge and Lasso Analysis

Ridge and Lasso do not provide and meaningful dimension reduction.

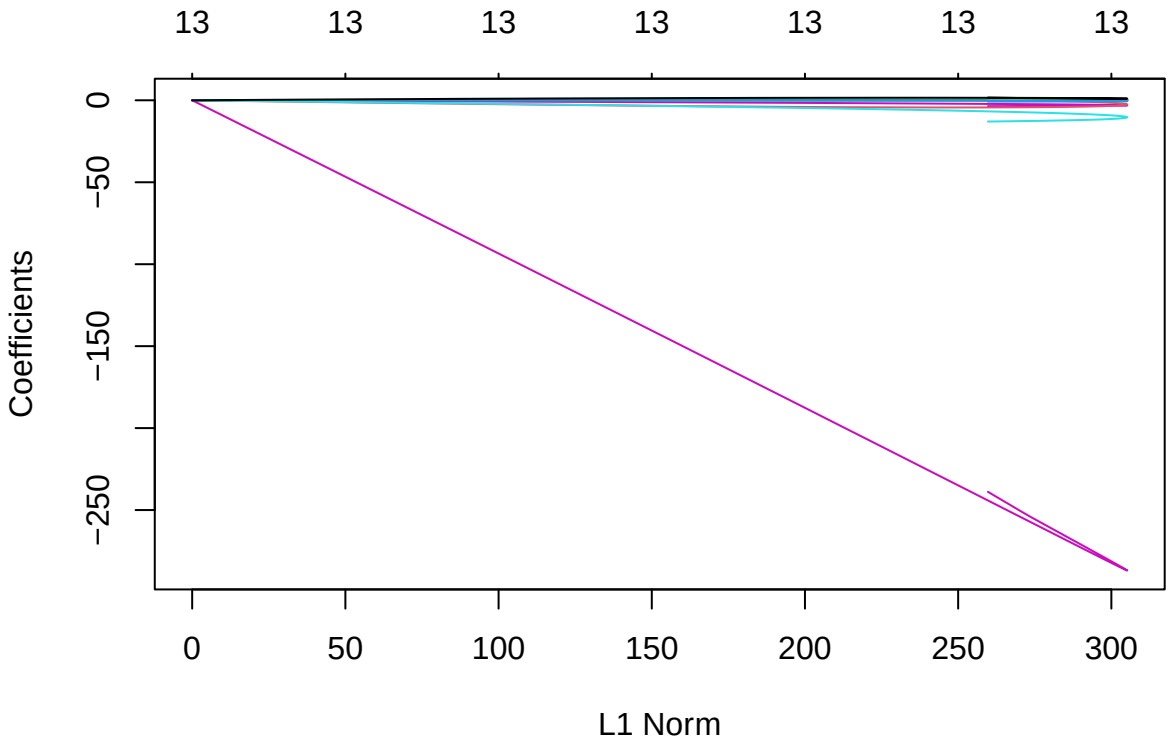
```
################################################################################
# Split the data into training and test sets
# Define the response and predictor variables
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
response_var <- "Standard.Score"
predictor_vars <- c("Expenditures.per.Pupil","Income.by.School.District",
                    "Total.Number.Students","Percent.White",
                    "Percent.African.American", "Percent.American.Indian",
                    "Percent.Asian", "Percent.Pacific.Islander",
                    "Number.Full.time.Teachers", "Student.Teacher.Ratio",
                    "Is.Title.I..","Is.Charter..","Is.Magnet..")

# Split the data into training and test sets
set.seed(123)
train_idx <- sample(nrow(df), 0.7 * nrow(df))
train_data <- df[train_idx, ]
test_data <- df[-train_idx, ]

# Fit ridge regression
x_train <- as.matrix(train_data[, predictor_vars])
```

```r
y_train <- train_data[, response_var]
fit_ridge <- glmnet(x_train, y_train, alpha = 0)
plot(fit_ridge)
```



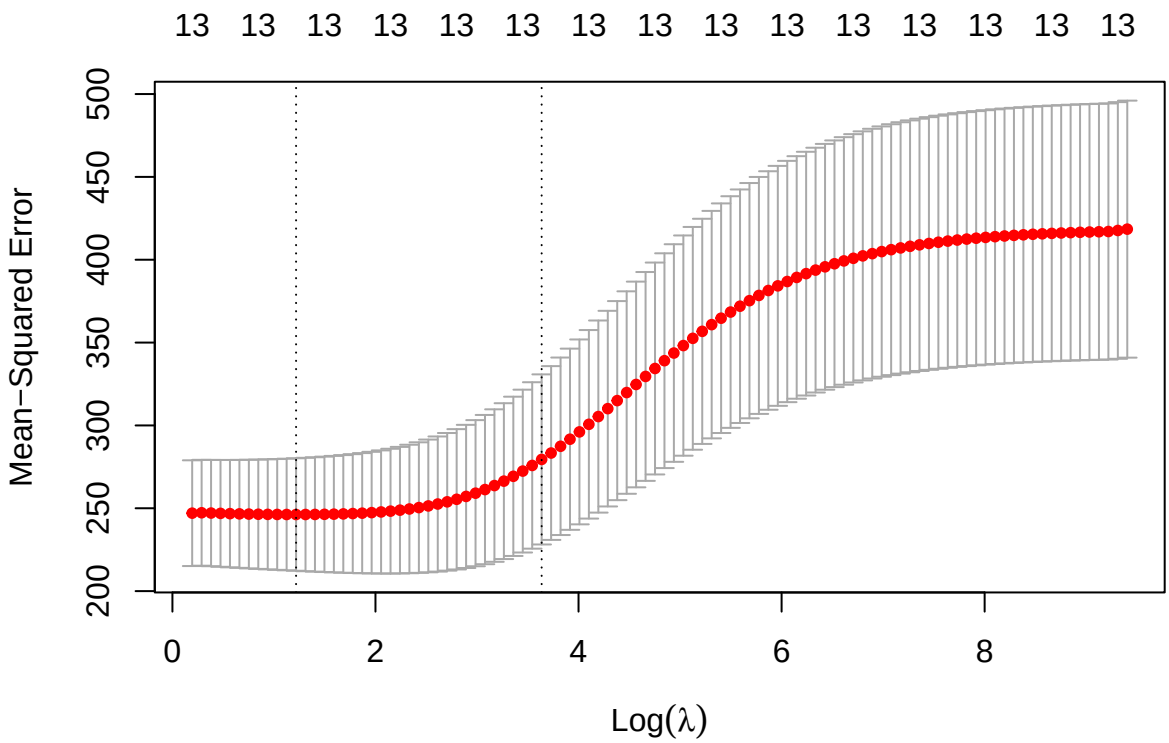

```r
print(fit_ridge)
```

```
##
## Call:  glmnet(x = x_train, y = y_train, alpha = 0)
##
##      Df  %Dev  Lambda
## 1    13  0.00 12150.0
## 2    13  0.38 11070.0
## 3    13  0.42 10090.0
## 4    13  0.46  9190.0
## 5    13  0.50  8373.0
## 6    13  0.55  7629.0
## 7    13  0.60  6952.0
## 8    13  0.66  6334.0
## 9    13  0.73  5771.0
## 10   13  0.80  5259.0
## 11   13  0.87  4791.0
## 12   13  0.96  4366.0
## 13   13  1.05  3978.0
## 14   13  1.15  3625.0
## 15   13  1.26  3303.0
## 16   13  1.38  3009.0
## 17   13  1.51  2742.0
## 18   13  1.65  2498.0
## 19   13  1.81  2276.0
## 20   13  1.98  2074.0
```

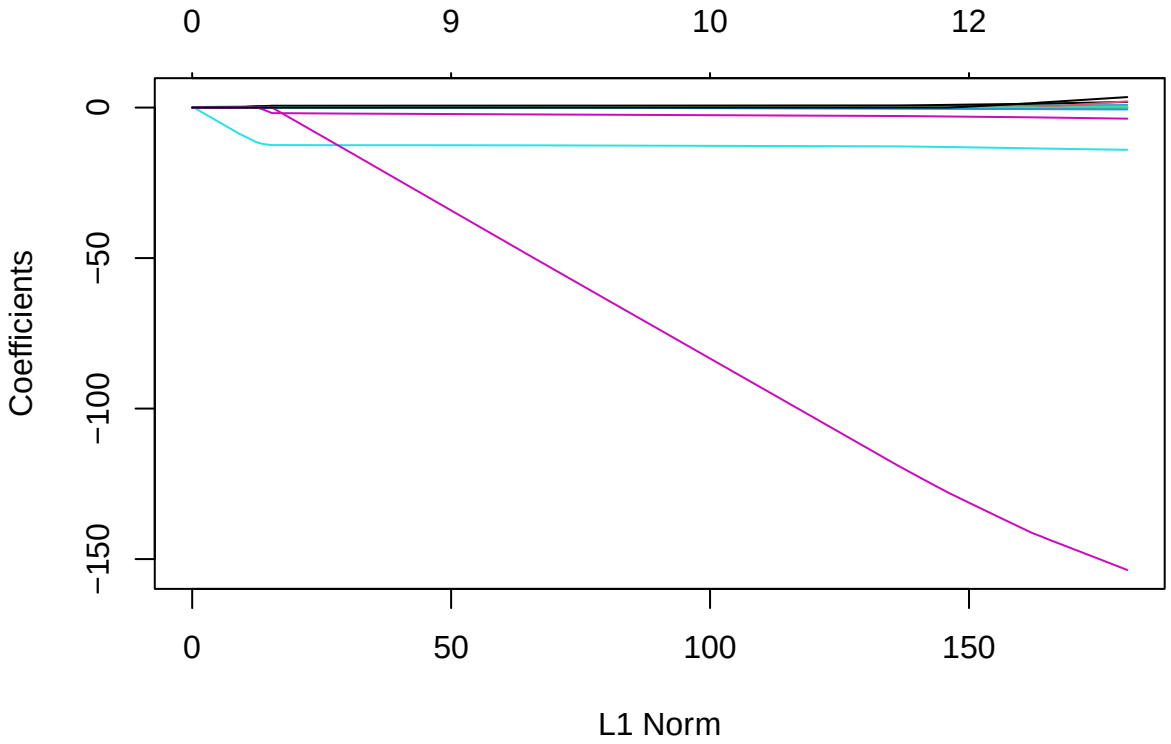
```
## 21  13  2.17  1890.0
## 22  13  2.37  1722.0
## 23  13  2.59  1569.0
## 24  13  2.83  1430.0
## 25  13  3.09  1303.0
## 26  13  3.38  1187.0
## 27  13  3.69  1081.0
## 28  13  4.03   985.4
## 29  13  4.39   897.8
## 30  13  4.79   818.1
## 31  13  5.22   745.4
## 32  13  5.68   679.2
## 33  13  6.18   618.8
## 34  13  6.72   563.9
## 35  13  7.30   513.8
## 36  13  7.92   468.1
## 37  13  8.59   426.5
## 38  13  9.30   388.7
## 39  13 10.06   354.1
## 40  13 10.87   322.7
## 41  13 11.73   294.0
## 42  13 12.65   267.9
## 43  13 13.61   244.1
## 44  13 14.62   222.4
## 45  13 15.69   202.6
## 46  13 16.80   184.6
## 47  13 17.96   168.2
## 48  13 19.16   153.3
## 49  13 20.40   139.7
## 50  13 21.68   127.3
## 51  13 22.99   116.0
## 52  13 24.32   105.7
## 53  13 25.68    96.3
## 54  13 27.05    87.7
## 55  13 28.42    79.9
## 56  13 29.80    72.8
## 57  13 31.16    66.4
## 58  13 32.51    60.5
## 59  13 33.84    55.1
## 60  13 35.14    50.2
## 61  13 36.41    45.7
## 62  13 37.64    41.7
## 63  13 38.82    38.0
## 64  13 39.95    34.6
## 65  13 41.03    31.5
## 66  13 42.06    28.7
## 67  13 43.04    26.2
## 68  13 43.95    23.9
## 69  13 44.82    21.7
## 70  13 45.63    19.8
## 71  13 46.39    18.0
## 72  13 47.09    16.4
## 73  13 47.75    15.0
## 74  13 48.37    13.7
```

```
## 75  13 48.94    12.4
## 76  13 49.47    11.3
## 77  13 49.97    10.3
## 78  13 50.43     9.4
## 79  13 50.87     8.6
## 80  13 51.28     7.8
## 81  13 51.66     7.1
## 82  13 52.03     6.5
## 83  13 52.37     5.9
## 84  13 52.70     5.4
## 85  13 53.02     4.9
## 86  13 53.32     4.5
## 87  13 53.61     4.1
## 88  13 53.89     3.7
## 89  13 54.16     3.4
## 90  13 54.42     3.1
## 91  13 54.67     2.8
## 92  13 54.92     2.6
## 93  13 55.15     2.3
## 94  13 55.38     2.1
## 95  13 55.60     1.9
## 96  13 55.82     1.8
## 97  13 56.02     1.6
## 98  13 56.21     1.5
## 99  13 56.40     1.3
## 100 13 56.57     1.2
```

```
cv_ridge <- cv.glmnet(x_train, y_train, alpha = 0)
plot(cv_ridge)
```



```r
# Fit lasso regression
fit_lasso <- glmnet(x_train, y_train, alpha = 1)
plot(fit_lasso)
```

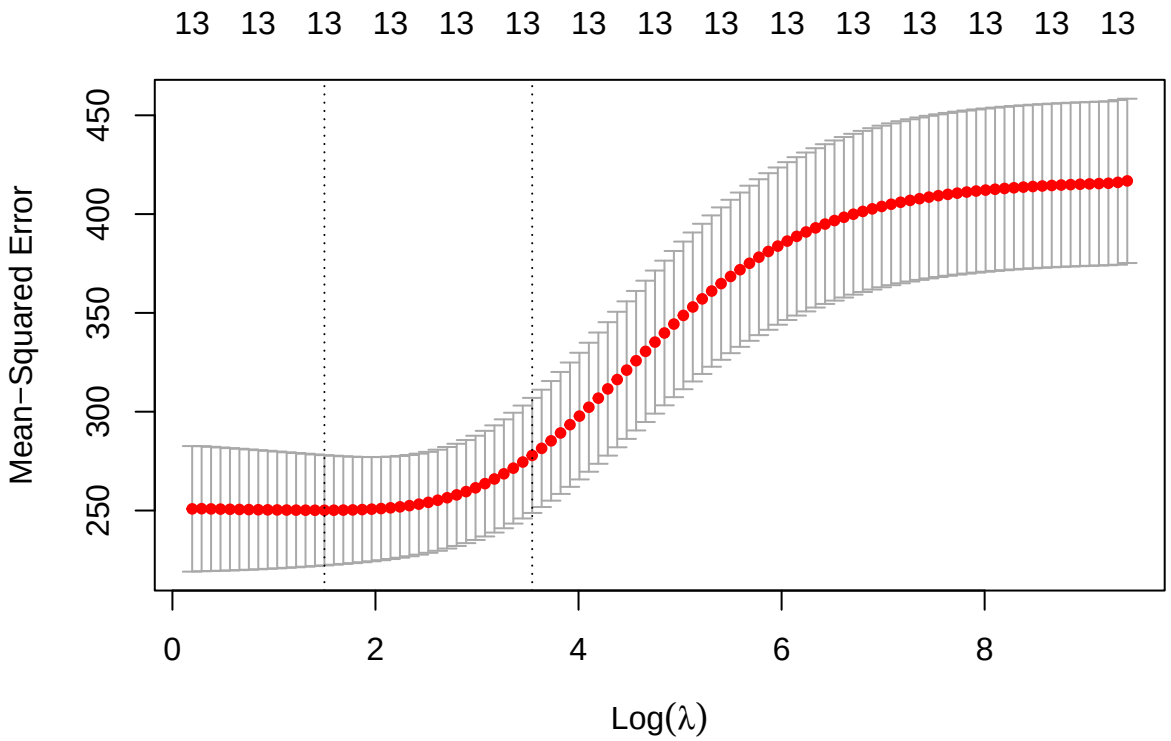



```r
print(fit_lasso)
```

```
##
## Call:  glmnet(x = x_train, y = y_train, alpha = 1)
##
##    Df  %Dev  Lambda
## 1   0  0.00 12.1500
## 2   2  6.41 11.0700
## 3   2 11.93 10.0900
## 4   2 16.51  9.1900
## 5   2 20.31  8.3730
## 6   2 23.46  7.6290
## 7   2 26.09  6.9520
## 8   2 28.26  6.3340
## 9   2 30.07  5.7710
## 10  4 32.02  5.2590
## 11  4 35.51  4.7910
## 12  4 38.40  4.3660
## 13  4 40.80  3.9780
## 14  4 42.79  3.6250
## 15  4 44.45  3.3030
## 16  5 46.28  3.0090
## 17  4 47.78  2.7420
## 18  4 48.78  2.4980
## 19  4 49.61  2.2760
## 20  5 50.41  2.0740
## 21  6 51.16  1.8900
```

```
## 22  7 51.82  1.7220
## 23  7 52.43  1.5690
## 24  7 52.94  1.4300
## 25  7 53.36  1.3030
## 26  7 53.71  1.1870
## 27  7 54.00  1.0810
## 28  8 54.27  0.9854
## 29  9 54.49  0.8978
## 30  9 54.68  0.8181
## 31 10 54.93  0.7454
## 32 10 55.13  0.6792
## 33 10 55.30  0.6188
## 34 10 55.44  0.5639
## 35 10 55.56  0.5138
## 36 10 55.66  0.4681
## 37 11 56.06  0.4265
## 38 11 56.42  0.3887
## 39 12 56.74  0.3541
## 40 12 56.99  0.3227
## 41 12 57.21  0.2940
## 42 12 57.39  0.2679
## 43 12 57.54  0.2441
## 44 13 57.67  0.2224
## 45 12 57.77  0.2026
## 46 12 57.86  0.1846
## 47 12 57.93  0.1682
## 48 12 57.99  0.1533
## 49 12 58.04  0.1397
## 50 12 58.08  0.1273
## 51 12 58.12  0.1160
## 52 12 58.15  0.1057
## 53 12 58.17  0.0963
## 54 12 58.19  0.0877
## 55 12 58.21  0.0799
## 56 12 58.22  0.0728
## 57 12 58.23  0.0664
## 58 12 58.24  0.0605
## 59 12 58.25  0.0551
## 60 12 58.26  0.0502
## 61 12 58.26  0.0457
## 62 12 58.27  0.0417
## 63 12 58.27  0.0380
## 64 12 58.27  0.0346
## 65 12 58.27  0.0315
## 66 12 58.28  0.0287
## 67 12 58.28  0.0262
## 68 12 58.28  0.0238
## 69 12 58.28  0.0217
## 70 12 58.28  0.0198
## 71 12 58.28  0.0180
## 72 12 58.28  0.0164
## 73 12 58.29  0.0150
## 74 12 58.29  0.0137
```

```r
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 0)
plot(cv_lasso)
```




```r
# Predict on test data
x_test <- as.matrix(test_data[, predictor_vars])
y_test <- test_data[, response_var]
pred_ridge <- predict(fit_ridge, newx = x_test)
pred_lasso <- predict(fit_lasso, newx = x_test)

# Calculate RMSE on test data
rmse_ridge <- sqrt(mean((y_test - pred_ridge)^2))
rmse_lasso <- sqrt(mean((y_test - pred_lasso)^2))

# Print the RMSE values
print(paste("RMSE (ridge):", rmse_ridge))
```

```
## [1] "RMSE (ridge): 16.2998719469293"
```

```r
print(paste("RMSE (lasso):", rmse_lasso))
```

```
## [1] "RMSE (lasso): 13.6172684811326"
```