DPS Hiring Project

Travis Froberg

12/8/2021

```r
library("readxl")
library(dplyr)
library(data.table)
library(kableExtra)
```

Data Preparation

```r
# Set working directory so could import data
setwd("C:\\Users\\malin\\Documents\\DPS Project")

# Imported StudentScores file and EducatorEffectivenessSnapshot file
scoreData <- read_excel("StudentScores.xlsx")
# View(scoreData)
snapData <- read_excel("EducatorEffectivenessSnapshot.xlsx")
# View(snapData)

# Removed first five rows from snapData
# because they were not a part of the dataset
snapData <- snapData[-(1:5),]
# View(snapData)

# searched school names in snapData to see if there were any
# names that were not schools. Removed obsevation 1 and 109 for this reason.
names(snapData) <- snapData[1,]
snapData <- snapData[-c(1, 109),]

# Explored the score data by checking the levels of some of the
# factor variables
# levels(factor(scoreData$ISDname))
# levels(factor(scoreData$ISDcode))
# levels(factor(scoreData$district_code))
# levels(factor(scoreData$school_year))
# levels(factor(scoreData$grade))
# levels(factor(scoreData$building_name))

# Removed values in the score data for any proficiency scores less than 10
# class(scoreData$average_scaled_score)
scoreData2 <- subset(scoreData, scoreData$average_scaled_score != "< 10")
# View(scoreData2)
```

```
# nrow(scoreData2)

# Removed values in score data that corresponded with individual schools
# that were not in snap data
loc_vector <- c(snapData$location)
build_name_vector <- c(scoreData2$building_name)
build_name_vector2 <- unique(build_name_vector)
# build_name_vector2
mismatch_vector <- build_name_vector2[!(build_name_vector2 %in% loc_vector)]
# mismatch_vector
remove <- which(scoreData2$building_name %in% mismatch_vector)
scoreData3 <- scoreData2[-remove,]
# nrow(scoreData3)
# View(scoreData3)

# Double checked the previous step
is.element("Burton International School", scoreData3$building_name)
```

```
## [1] TRUE
```

```
is.element("Pulaski Elementary-Middle School", scoreData3$building_name)
```

```
## [1] FALSE
```

The following is the code corresponding to project question number 1

```
# Created a subset of the score data that included only the value "All Students" in
# the variable subgroup and "Mathematics" in the variable subject_name
tempData <- subset(scoreData3, subgroup == "All Students" & subject_name == "Mathematics")
# View(tempData)

# Created data frame of the average math proficiency scores per school
tempData2 <- aggregate(x = as.numeric(tempData$percent_proficient),
          by = list(tempData$building_name),
          FUN = mean)

# Arranged math proficiency percentages in descending order and created appropriate
# column names
mean.prof.df <- tempData2 %>% arrange(desc(tempData2$x))
colnames(mean.prof.df) <- c("School", "Average Math Proficiency Rate")

# Double checked to make sure the previous step was done correctly
# nrow(mean.prof.df)
# max(tempData2$x)
# nrow(tempData2)

# Created table of the top ten schools in terms of mean math proficiency scores
table.math.prof <- mean.prof.df[1:10,]
table.math.prof
```

```
##                                     School Average Math Proficiency Rate
## 1                   Wright, Charles School                     0.6215000
## 2                Chrysler Elementary School                     0.5633333
## 3                             Bates Academy                     0.4856667
## 4          Davison Elementary-Middle School                     0.4661667
## 5                    Dixon Elementary School                     0.3670000
## 6               Burton International School                     0.3541667
## 7                  Pasteur Elementary School                     0.2557500
## 8  Greenfield Union Elementary-Middle School                     0.2406667
## 9                 Thirkell Elementary School                     0.2327500
## 10                   Cooke Elementary School                     0.2292500
```

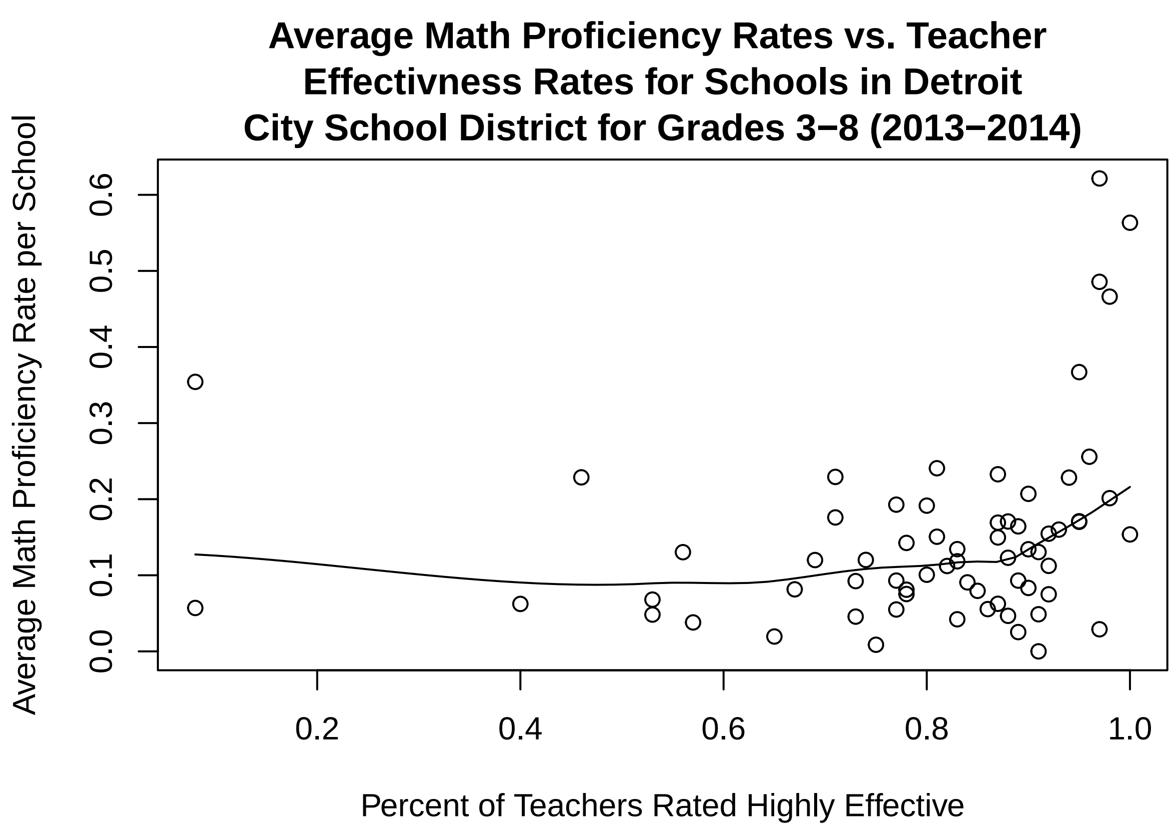
```
# table.math.prof %>%
#  kbl() %>%
#  kable_minimal() %>%
#  add_header_above(c("Top Ten Schools in Math Proficiency for Grades 3-8 in Detroit # City School Dist
```

The following is the code corresponding to project question number 2

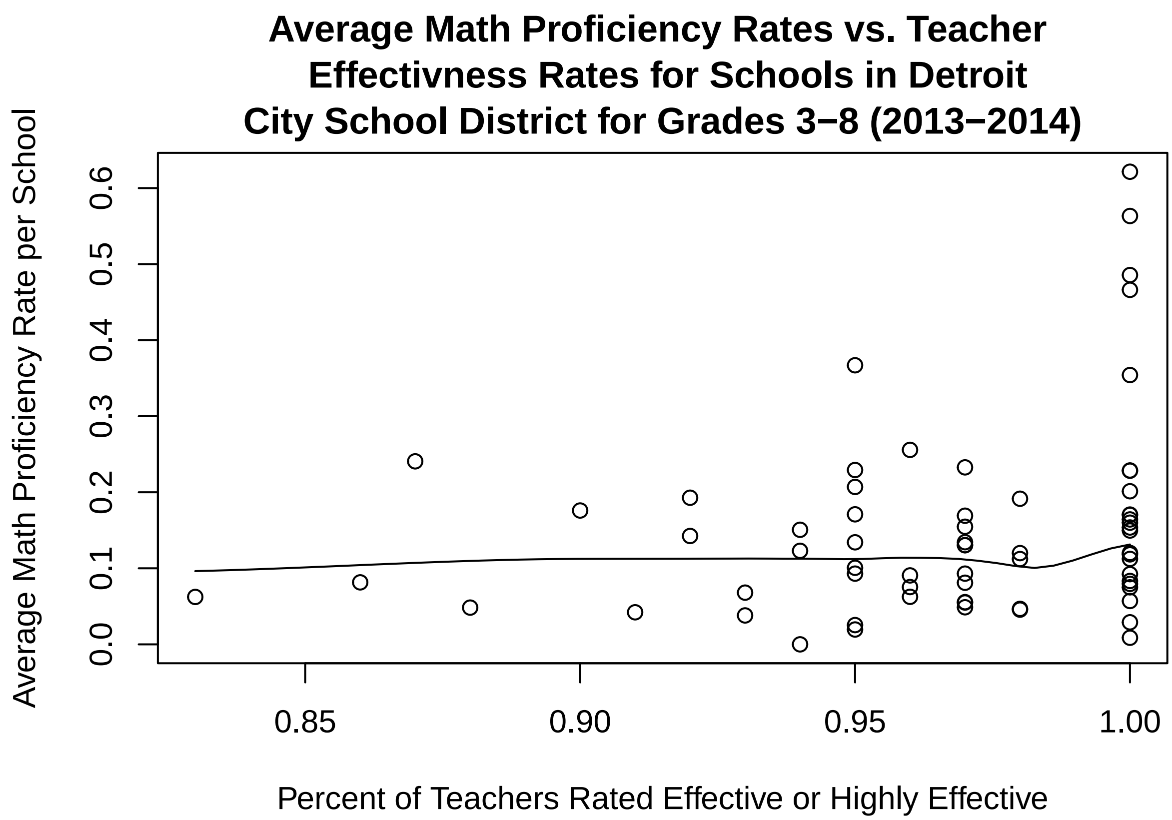
```
# Removed first two rows in snap data because they did not
# correspond to individual schools
snapData2 <- snapData[-c(1,2),]

# Removed all rows in snap data where there was not
# a matching school in score data. For example, if snap data included a
# school that was not in the score data set, this observation would be removed
# from snap data in the final data set.
snapData3 <- snapData2[match(mean.prof.df$School, snapData2$location, nomatch = 0),]

# Checked to make sure previous step was done correctly
# nrow(snapData3)
# nrow(mean.prof.df)
# View(snapData3)

# Ordered snap data by school (called location) and score data by school alphebetically
snapData3 <- snapData3[order(snapData3$location),]
scoreData4 <- mean.prof.df[order(mean.prof.df$School),]
# View(scoreData4)
# nrow(scoreData4)

# Checked to make sure schools all matched up between the two data sets, then merged
# the two datasets
snapData3$location == scoreData4$School
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [61] TRUE TRUE TRUE TRUE TRUE TRUE
```

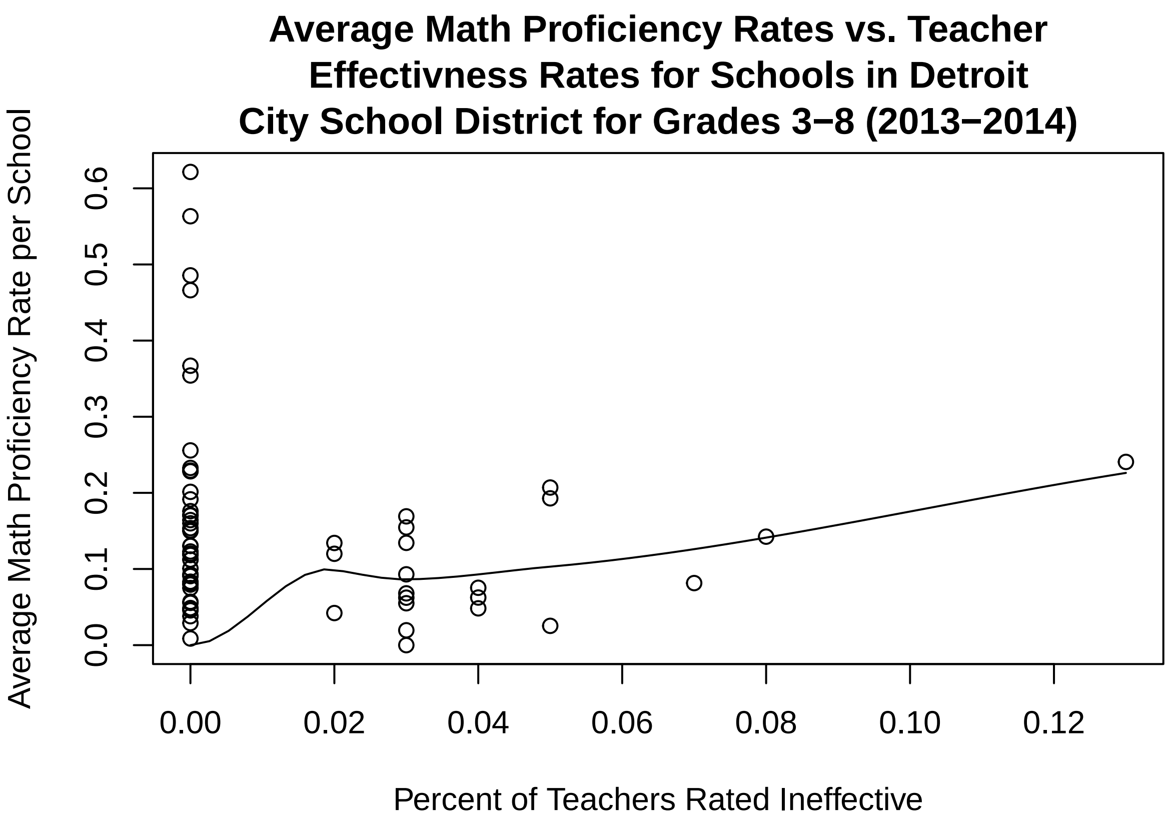
```r
mergedData <- data.frame(snapData3, scoreData4)
# View(mergedData)

# Ordered the merged dataset by mean math proficiency rate to make plots easier to interpret
mergedData2 <- mergedData %>% arrange(desc(mergedData$Average.Math.Proficiency.Rate))
# View(mergedData2)

# Created scatterplot of mean proficiency rate and scatter plots of teacher effectiveness
# with mean proficiency rate on the y-axis
# Used scatter.smooth so a line would be added to the scatter plots. This line was not created
# from a sophisticated model, but is rather used to get a general glance at any trends
# in the data
scatter.smooth(mergedData2$highly_effective_percent, mergedData2$Average.Math.Proficiency.Rate,
               xlab = "Percent of Teachers Rated Highly Effective",
               ylab = "Average Math Proficiency Rate per School",
               main = "Average Math Proficiency Rates vs. Teacher
 Effectivness Rates for Schools in Detroit
City School District for Grades 3-8 (2013-2014)")
```




```r
scatter.smooth(mergedData2$effective_or_more_percent, mergedData2$Average.Math.Proficiency.Rate,
               xlab = "Percent of Teachers Rated Effective or Highly Effective",
               ylab = "Average Math Proficiency Rate per School",
               main = "Average Math Proficiency Rates vs. Teacher
 Effectivness Rates for Schools in Detroit
City School District for Grades 3-8 (2013-2014)")
```




```
scatter.smooth(mergedData2$ineffective_percent, mergedData2$Average.Math.Proficiency.Rate,
               xlab = "Percent of Teachers Rated Ineffective",
               ylab = "Average Math Proficiency Rate per School",
               main = "Average Math Proficiency Rates vs. Teacher
  Effectivness Rates for Schools in Detroit
City School District for Grades 3-8 (2013-2014)")
```




```r
# plot(mergedData2$Average.Math.Proficiency.Rate)

# Performed hypothesis test to determine whether mean proficiency rate was
# correlated with the variable effective_or_more_percent
test <- cor.test(mergedData2$Average.Math.Proficiency.Rate,
                 as.numeric(mergedData2$effective_or_more_percent))
test
```

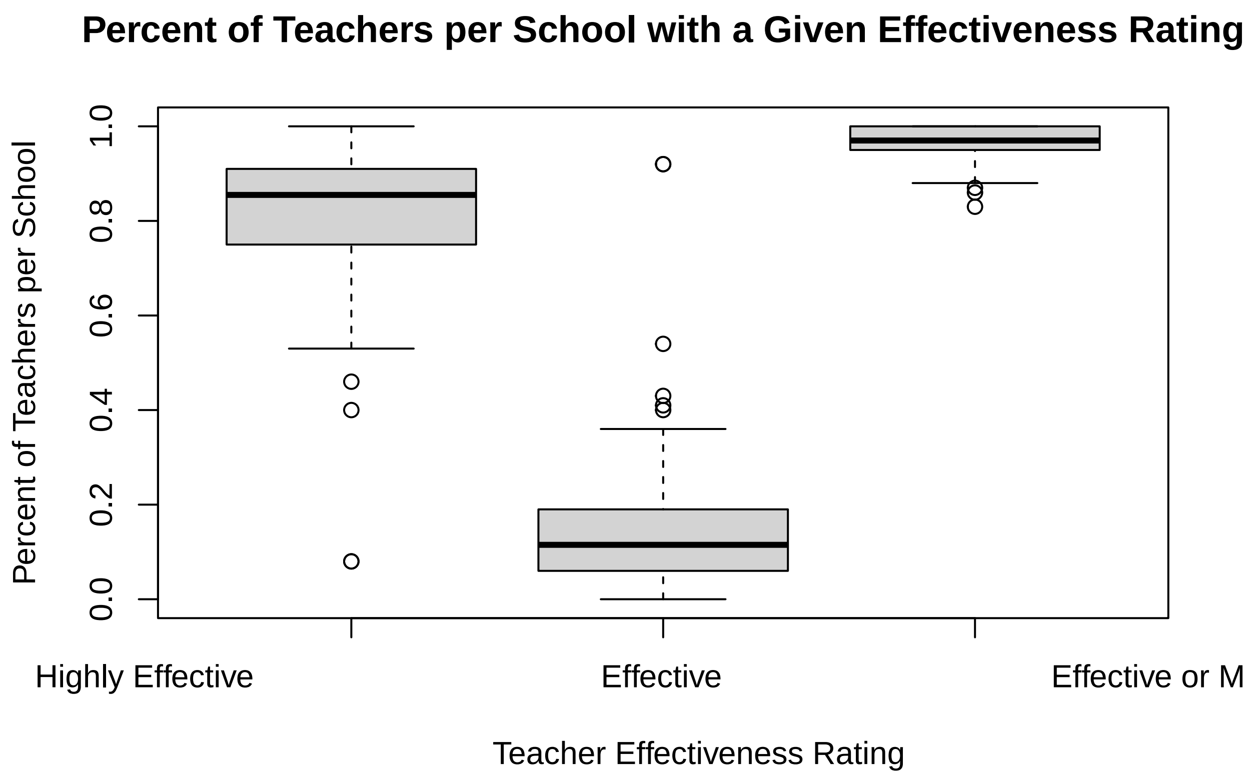
```
##
##  Pearson's product-moment correlation
##
## data:  mergedData2$Average.Math.Proficiency.Rate and as.numeric(mergedData2$effective_or_more_percent
## t = 1.9641, df = 64, p-value = 0.05386
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.003816311  0.454254667
## sample estimates:
##       cor
## 0.2384367
```

```r
# Checked to see if previous step calculated the correct correlation coefficient by
# calculating it with a different function
cor(mergedData2$Average.Math.Proficiency.Rate, as.numeric(mergedData2$effective_or_more_percent))
```

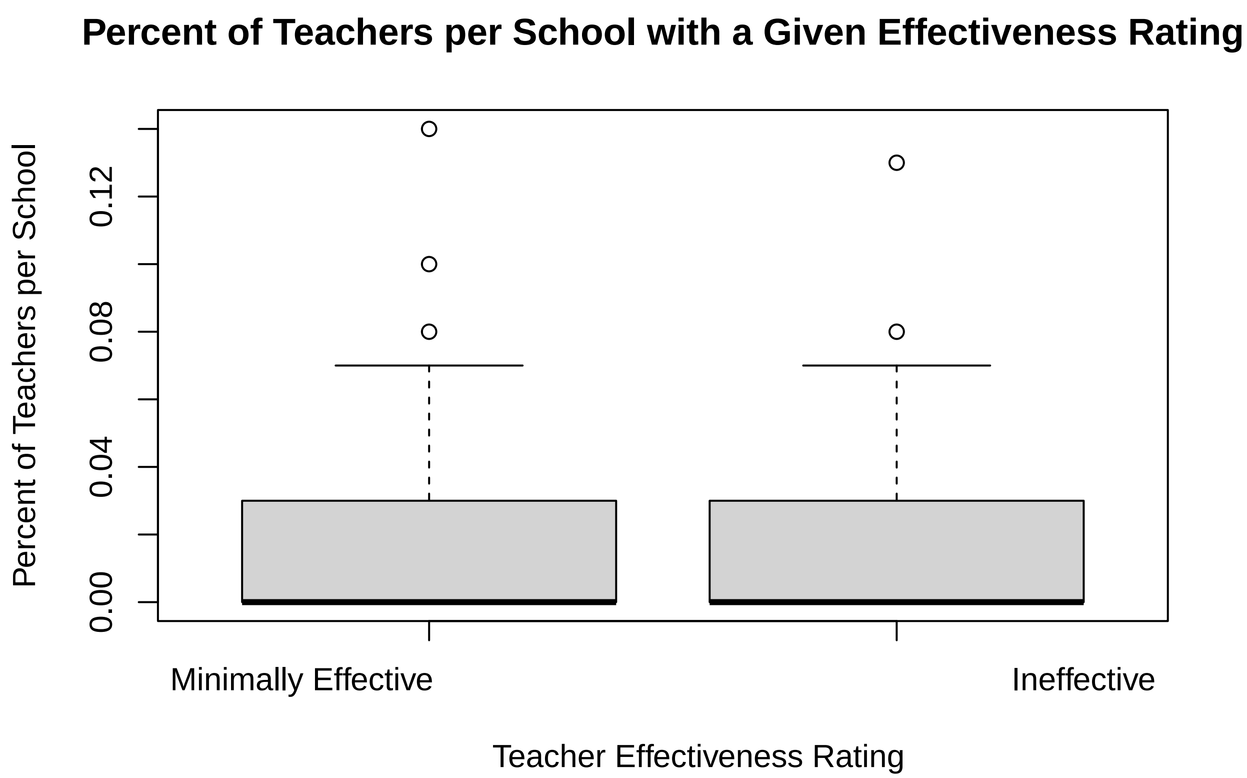
```
## [1] 0.2384367
```

The following is the code corresponding to project question number 3

```
# Created box-plots of teacher effectivness percentages to create a visual for
# determining the general spread of the data
boxplot(as.numeric(mergedData2$highly_effective_percent), as.numeric(mergedData2$effective_percent),
        as.numeric(mergedData2$effective_or_more_percent),
        xlab = "Highly Effective                                         Effective

        Teacher Effectiveness Rating",
        ylab = "Percent of Teachers per School",
        main = "Percent of Teachers per School with a Given Effectiveness Rating")
```




```
boxplot(as.numeric(mergedData2$minimally_effective_percent),
        as.numeric(mergedData2$ineffective_percent),
        xlab = "Minimally Effective                                                          Ine

        Teacher Effectiveness Rating",
        ylab = "Percent of Teachers per School",
        main = "Percent of Teachers per School with a Given Effectiveness Rating")
```




```r
# Created data frame containing statistics that measure spread
first_col <- c(var(as.numeric(mergedData2$highly_effective_percent)),
                var(as.numeric(mergedData2$effective_percent)),
                var(as.numeric(mergedData2$minimally_effective_percent)),
                var(as.numeric(mergedData2$ineffective_percent)),
                var(as.numeric(mergedData2$effective_or_more_percent)))
one.5_col <- c(sqrt(var(as.numeric(mergedData2$highly_effective_percent))),
                sqrt(var(as.numeric(mergedData2$effective_percent))),
                sqrt(var(as.numeric(mergedData2$minimally_effective_percent))),
                sqrt(var(as.numeric(mergedData2$ineffective_percent))),
                sqrt(var(as.numeric(mergedData2$effective_or_more_percent))))
second_col <- c(min(as.numeric(mergedData2$highly_effective_percent)),
               min(as.numeric(mergedData2$effective_percent)),
               min(as.numeric(mergedData2$minimally_effective_percent)),
               min(as.numeric(mergedData2$ineffective_percent)),
               min(as.numeric(mergedData2$effective_or_more_percent)))
third_col <- c(max(as.numeric(mergedData2$highly_effective_percent)),
                max(as.numeric(mergedData2$effective_percent)),
                max(as.numeric(mergedData2$minimally_effective_percent)),
                max(as.numeric(mergedData2$ineffective_percent)),
                max(as.numeric(mergedData2$effective_or_more_percent)))
four_col <- c(third_col-second_col)

row.names <- c("Highly Effective Percent", "Effective Percent", "Minimally Effective Percent", "Ineffec
spread.table <- data.frame(first_col, one.5_col, second_col, third_col, four_col, row.names = row.names
spread.table <- setNames(spread.table,c("Variance", "Standard Deviation", "Minimum", "Maximum", "Range"
```

```r
spread.table
```

```
##                                 Variance Standard Deviation Minimum Maximum
## Highly Effective Percent    0.0338594172         0.18400929    0.08    1.00
## Effective Percent           0.0317115152         0.17807727    0.00    0.92
## Minimally Effective Percent 0.0008704196         0.02950287    0.00    0.14
## Ineffective Percent         0.0005856410         0.02420002    0.00    0.13
## Effective or More Percent   0.0014465501         0.03803354    0.83    1.00
##                             Range
## Highly Effective Percent     0.92
## Effective Percent            0.92
## Minimally Effective Percent  0.14
## Ineffective Percent          0.13
## Effective or More Percent    0.17
```