ELA-Visualizations

John Bartlett, Ebo Edonu, Raya Young

12/16/2019

In order to make the raw data file more digestible we first need to break down the excel file into smaller, more digestible chunks. The preceding r code loads the data into r and extracts the summary data from the dataframe. There are 6 different subgroups of data, Statewide, NRC, NRC by economic needs, County, School District, down to individual schools.

```
#state total data frame
STATE_ELA <- sqldf("SELECT * from ELA where NAME like 'STATEWIDE -%'")
STATE_ELA <- STATE_ELA[,c(1,6:23)] #trim off columns 2,3,4,5 due to NA

#NRC total data frame
NRC_ELA <- sqldf("SELECT * from ELA where NAME like 'NRC -%'")
NRC_ELA <- NRC_ELA[,c(1:3,6:23)] #trim off columns 4 and 5 due to NA

#pull out only NRC needs from the data frame
NRC_NEEDS_ELA <- sqldf("SELECT * from NRC_ELA where NAME like '%NEEDS%'")
NRC_ELA <- sqldf("SELECT * from NRC_ELA where NAME NOT like '%NEEDS%'")#restore NRC with just the non N
```

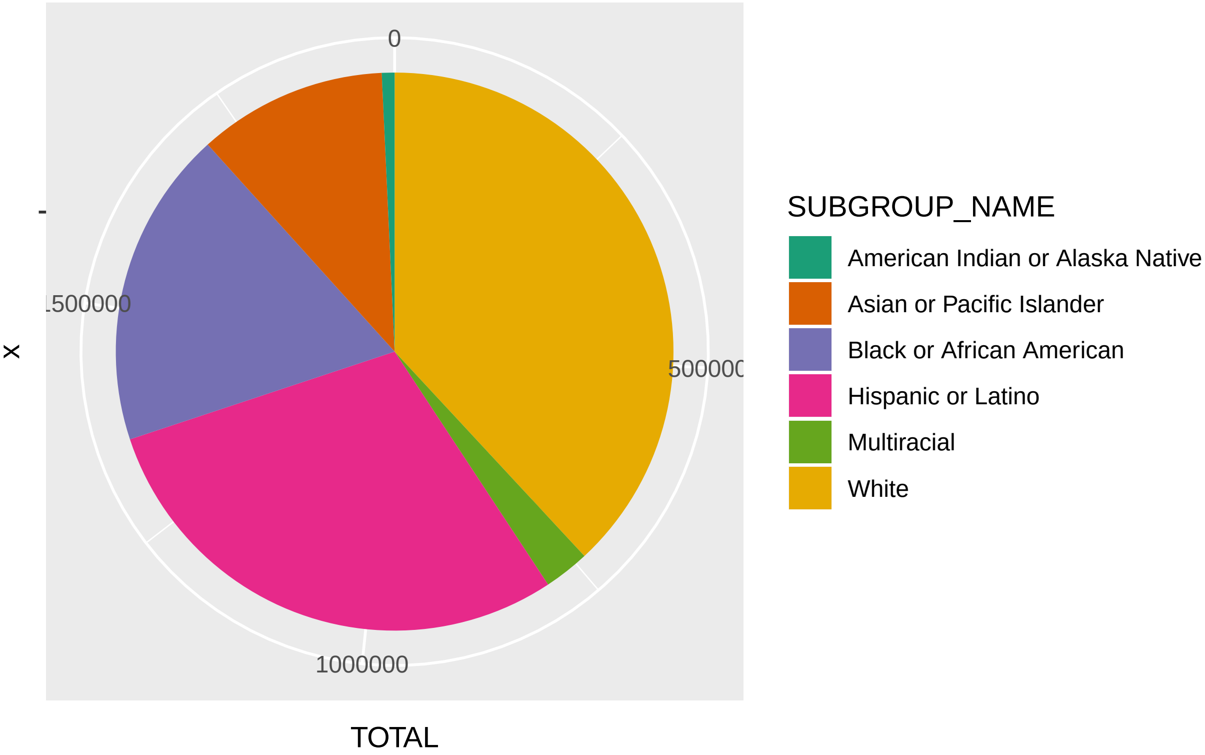
```
#county total data frame
COUNTY_ELA <-  sqldf("SELECT * from ELA where NAME like '%COUNTY%'")
COUNTY_ELA <- COUNTY_ELA[,c(1,4,6:23)]
```

```
#district total data frame
DISTRICT_ELA <- sqldf("SELECT * from ELA where NAME like '%District%'")
DISTRICT_ELA <- DISTRICT_ELA[,c(1:4,6:23)]

# convertiing the test count to numeric
cols.nms <- colnames(ELA[,12:23])
ELA[cols.nms] <- sapply(ELA[cols.nms], as.numeric)
sapply(ELA, class)


# dropping unwanted columns
ELA <- ELA[,-c(14,16,18,20,21,22)]



sch_all <- sqldf("select * from ELA where NAME LIKE '%SCHOOL' and NAME not like '%DISTRICT'
             AND BEDSCODE <> 1")
sch_race <- sqldf("select NRC_CODE
                    ,NRC_DESC
                    ,COUNTY_DESC
                    ,BEDSCODE
                    ,NAME
                    ,ITEM_SUBJECT_AREA
                    ,ITEM_DESC
                    ,SUBGROUP_CODE
                    ,SUBGROUP_NAME
                    ,TOTAL_TESTED
                    ,L1_COUNT, L2_COUNT, L3_COUNT, L4_COUNT, MEAN_SCALE_SCORE
                    ,CASE WHEN SUBGROUP_CODE = 4 THEN 1 ELSE 0 END AS AMERICAN_INDIAN
                    ,CASE WHEN SUBGROUP_CODE = 5 THEN 1 ELSE 0 END AS BLACK_AMERICAN
                    ,CASE WHEN SUBGROUP_CODE = 6 THEN 1 ELSE 0 END AS LATINO
                    ,CASE WHEN SUBGROUP_CODE = 7 THEN 1 ELSE 0 END AS ASIAN
                    ,CASE WHEN SUBGROUP_CODE = 6 THEN 1 ELSE 0 END AS WHITE
                    ,CASE WHEN SUBGROUP_CODE = 6 THEN 1 ELSE 0 END AS MULTIRIACIAL
                  FROM sch_all where SUBGROUP_CODE in (4,5,6,7,8,9) ")

#grouping examples
```

Racial Identity Overview

In this first example we break down the State_Ela data frame into a more digestible chunk and use it to create a pie chart of the total student population sub divided by racial identity.

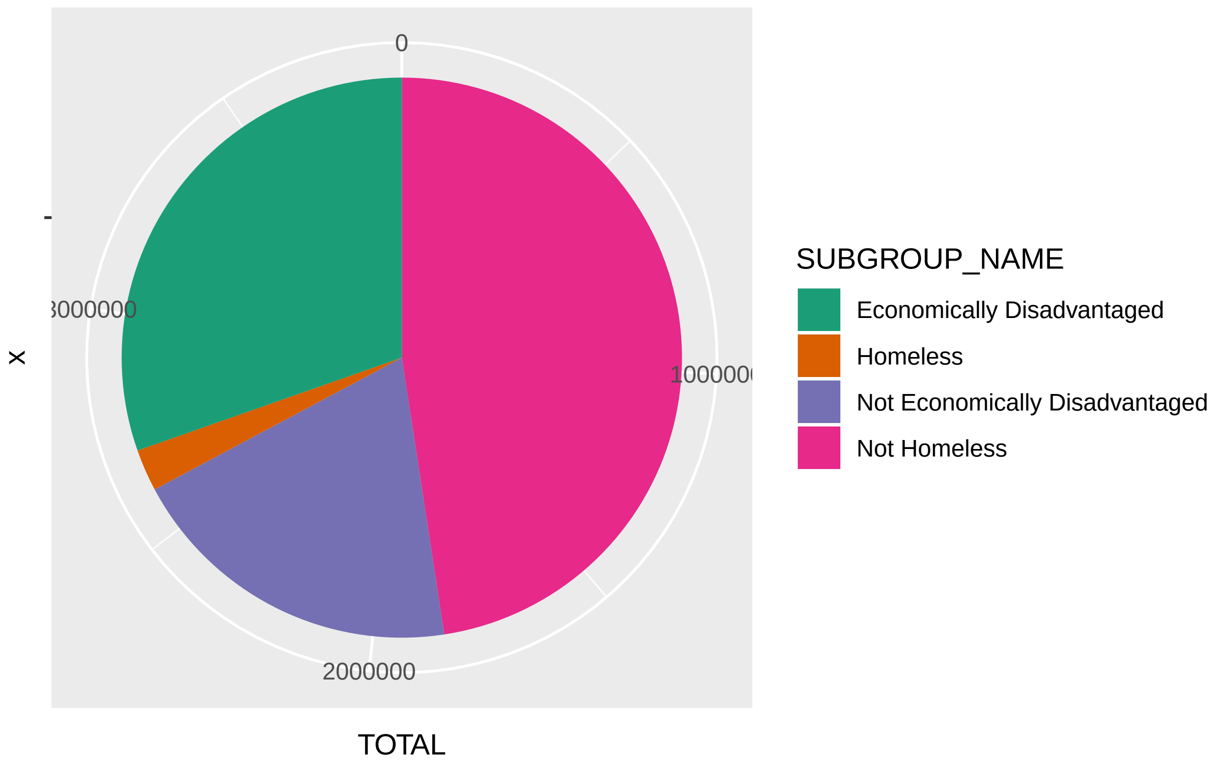
```
##########################################################################################
################# Race breakdown PIE
##########################################################################################

RACE_DF <- sqldf("SELECT SUBGROUP_NAME, SUM(TOTAL_TESTED) from STATE_ELA
                   where SUBGROUP_CODE = 4 or SUBGROUP_CODE = 5 or SUBGROUP_CODE = 6 or
                   SUBGROUP_CODE = 7 or SUBGROUP_CODE = 8 or SUBGROUP_CODE = 9
                   GROUP BY SUBGROUP_NAME
                   ORDER BY SUM(TOTAL_TESTED)")

colnames(RACE_DF) <- c('SUBGROUP_NAME', 'TOTAL')

ggplot(RACE_DF, aes(x="", y=TOTAL, fill=SUBGROUP_NAME)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start=0) +
  scale_fill_brewer(palette="Dark2")
```



Homelessness Overview

This second pie chart is made with the same strategies we used above, but this time on a dataframe broken down by the living status of the student.

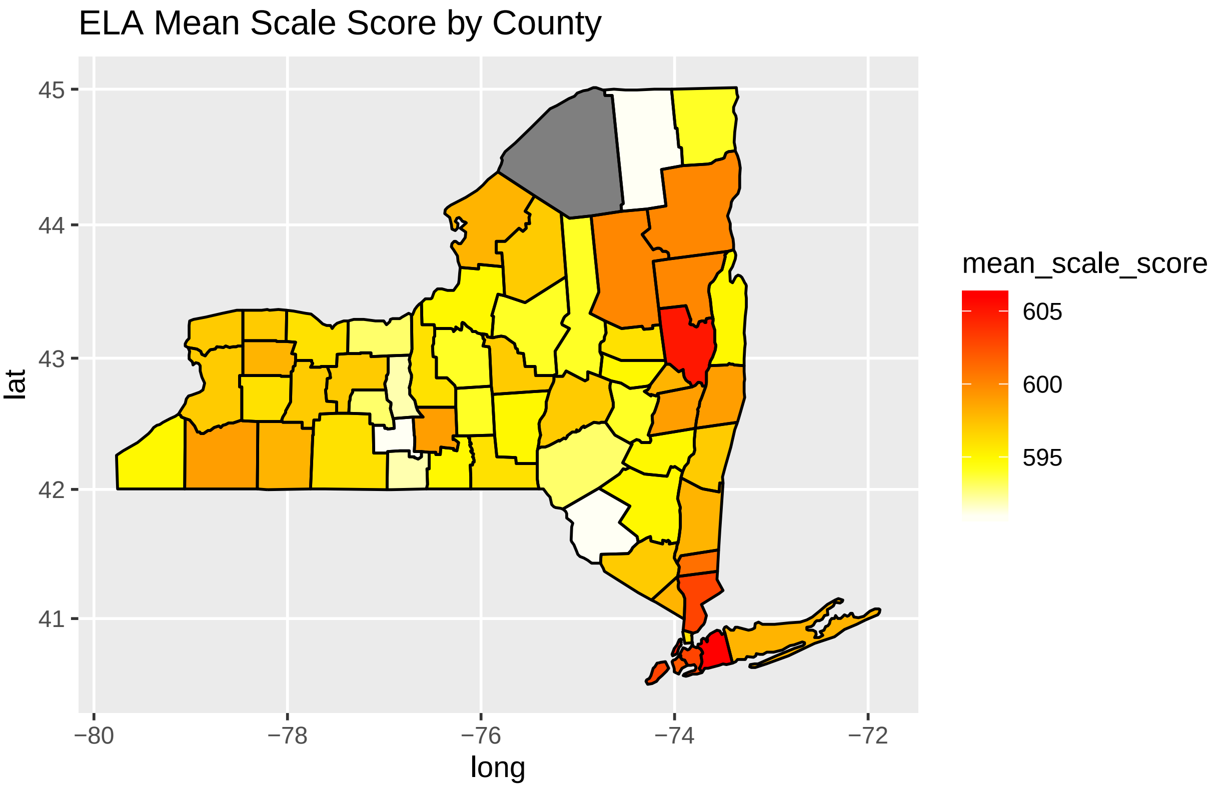
```
HOMES_DF <- sqldf("SELECT SUBGROUP_NAME, SUM(TOTAL_TESTED) from STATE_ELA
                   where SUBGROUP_CODE = 15 or SUBGROUP_CODE = 16 or SUBGROUP_CODE = 20 or
                   SUBGROUP_CODE = 21
                   GROUP BY SUBGROUP_NAME
                   ORDER BY SUM(TOTAL_TESTED)")

colnames(HOMES_DF) <- c('SUBGROUP_NAME', 'TOTAL')

options(scipen=10000)

ggplot(HOMES_DF, aes(x="", y=TOTAL, fill=SUBGROUP_NAME)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start=0) +
  scale_fill_brewer(palette="Dark2")
```



## ELA Overview by County This second example uses the County_Ela data frame and connects it to a data frame of counties within NY state. We then plot a county map with a fill that is representative of the ELA Mean Scale Score of each respective county.

Converting Mean Scale Score to numeric makes the map fill values plot correctly. The map is pulled from the mapdata library.

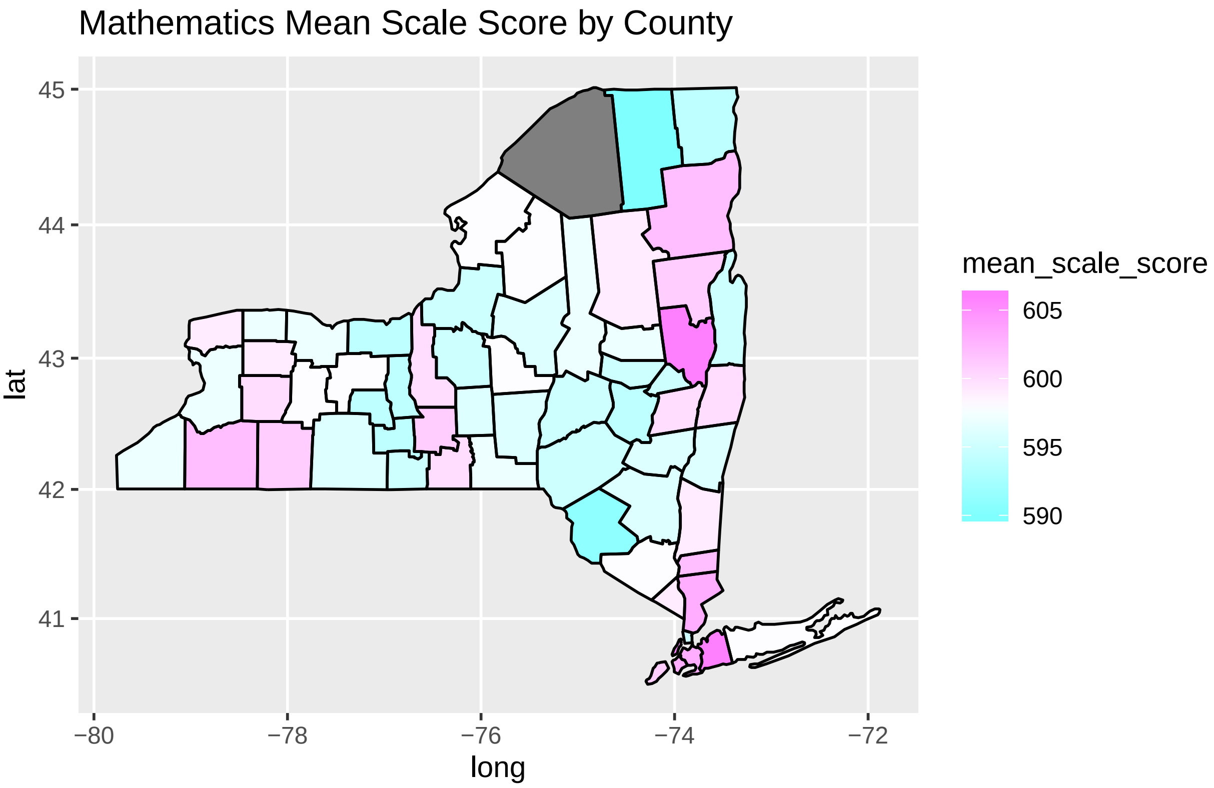
```
##########################################################################################
################## ELA Mean Scale Score by County
##########################################################################################

#create a df based off of a selection from the main dataset
COUNTY_SUM_DF <- sqldf("SELECT NAME, SUBGROUP_NAME, TOTAL_TESTED, MEAN_SCALE_SCORE
                 from COUNTY_ELA
                 WHERE SUBGROUP_CODE is 1 AND ITEM_SUBJECT_AREA like 'ELA'
                 GROUP BY NAME
                 ORDER BY NAME")

COUNTY_SUM_DF <- data.frame(tolower(str_remove(COUNTY_SUM_DF$NAME, ' COUNTY')),
                            COUNTY_SUM_DF$SUBGROUP_NAME, COUNTY_SUM_DF$TOTAL_TESTED, as.numeric(COUNTY_

colnames(COUNTY_SUM_DF) <- c('name', 'subgroup', 'total', 'mean_scale_score')

library(ggplot2)
library(mapdata)
library(tidyverse)

#create map
ny <- map_data('county', 'new york')
COUNTY_SUM_DF_MAP <- merge(ny, COUNTY_SUM_DF, by.x = 'subregion', by.y = 'name', all.x = TRUE, all.y =
COUNTY_SUM_DF_MAP <- COUNTY_SUM_DF_MAP[order(COUNTY_SUM_DF_MAP$order),]


#create map with data
ggplot(COUNTY_SUM_DF_MAP, aes(x=long,y=lat,group=group)) +
  geom_polygon(aes(fill=mean_scale_score))+
  scale_fill_gradientn(colours = rev(heat.colors(50))) +
  geom_path() +
  coord_map() + ggtitle('ELA Mean Scale Score by County')
```


ELA Mean Scale Score by County
45
44
43
42
41
lat
−80
−78
−76
−74
−72
long
mean_scale_score
605
600
595

## Math Overview by County

In this Example we plot the same map in the previous plot but this time with Math scores instead of ELA scores.

```
##########################################################################################
################## Mathematics Mean Scale Score by County
##########################################################################################

#create a df based off of a selection from the main dataset
COUNTY_SUM_DF <- sqldf("SELECT NAME, SUBGROUP_NAME, TOTAL_TESTED, MEAN_SCALE_SCORE
                 from COUNTY_ELA
                 WHERE SUBGROUP_CODE is 1 AND ITEM_SUBJECT_AREA like 'Mathematics'
                 GROUP BY NAME
                 ORDER BY NAME")

COUNTY_SUM_DF <- data.frame(tolower(str_remove(COUNTY_SUM_DF$NAME, ' COUNTY')),
                            COUNTY_SUM_DF$SUBGROUP_NAME, COUNTY_SUM_DF$TOTAL_TESTED, as.numeric(COUNTY_S

colnames(COUNTY_SUM_DF) <- c('name', 'subgroup', 'total', 'mean_scale_score')

library(ggplot2)
library(mapdata)
library(tidyverse)

#create map
ny <- map_data('county', 'new york')
COUNTY_SUM_DF_MAP <- merge(ny, COUNTY_SUM_DF, by.x = 'subregion', by.y = 'name', all.x = TRUE, all.y = 
COUNTY_SUM_DF_MAP <- COUNTY_SUM_DF_MAP[order(COUNTY_SUM_DF_MAP$order),]


#create map with data
ggplot(COUNTY_SUM_DF_MAP, aes(x=long,y=lat,group=group)) +
  geom_polygon(aes(fill=mean_scale_score))+
  scale_fill_gradientn(colours = cm.colors(50)) +
  geom_path() +
  coord_map() + ggtitle('Mathematics Mean Scale Score by County')
```


Mathematics Mean Scale Score by County
45
44
43
42
41
lat
−80
−78
−76
−74
−72
long
mean_scale_score
605
600
595
590

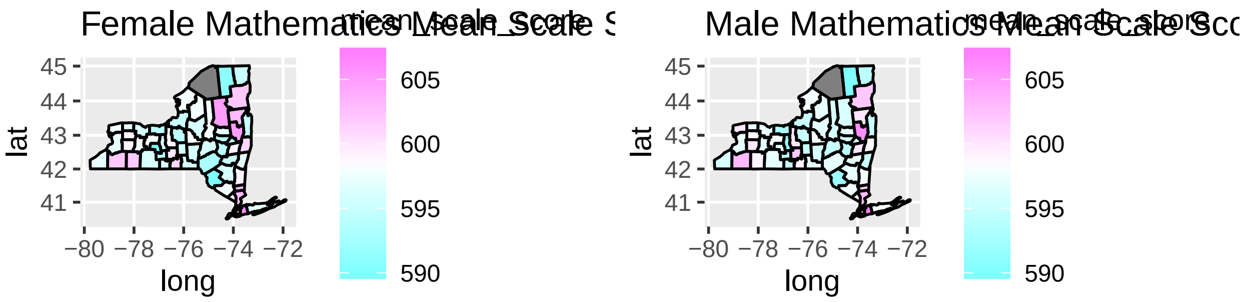
Math Overview by County Gender Comparison

In this Example we plot two map plots of Math scores, but this time broken into two different data frames, one for specifically female scores and the other for male. We then plot both data frames joining them with NY county map data to visually compare different counties that differ on Male vs Female Mean scale scores.

The plots are also arranged together on one plot for easier viewing.

```
#create a df based off of a selection from the main dataset
COUNTY_SUM_DF <- sqldf("SELECT NAME, SUBGROUP_NAME, TOTAL_TESTED, MEAN_SCALE_SCORE
                 from COUNTY_ELA
                 WHERE SUBGROUP_CODE is 2 AND ITEM_SUBJECT_AREA like 'Mathematics'
                 GROUP BY NAME
                 ORDER BY NAME")

COUNTY_SUM_DF <- data.frame(tolower(str_remove(COUNTY_SUM_DF$NAME, ' COUNTY')),
                            COUNTY_SUM_DF$SUBGROUP_NAME, COUNTY_SUM_DF$TOTAL_TESTED, as.numeric(COUNTY_S

colnames(COUNTY_SUM_DF) <- c('name', 'subgroup', 'total', 'mean_scale_score')

library(ggplot2)
library(mapdata)
library(tidyverse)

#create map
ny <- map_data('county', 'new york')
COUNTY_SUM_DF_MAP <- merge(ny, COUNTY_SUM_DF, by.x = 'subregion', by.y = 'name', all.x = TRUE, all.y =
COUNTY_SUM_DF_MAP <- COUNTY_SUM_DF_MAP[order(COUNTY_SUM_DF_MAP$order),]


#create map with data
f <- ggplot(COUNTY_SUM_DF_MAP, aes(x=long,y=lat,group=group)) +
  geom_polygon(aes(fill=mean_scale_score))+
  scale_fill_gradientn(colours = cm.colors(50)) +
  geom_path() +
  coord_map() + ggtitle('Female Mathematics Mean Scale Score by County')

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

#create a df based off of a selection from the main dataset
COUNTY_SUM_DF <- sqldf("SELECT NAME, SUBGROUP_NAME, TOTAL_TESTED, MEAN_SCALE_SCORE
                 from COUNTY_ELA
                 WHERE SUBGROUP_CODE is 3 AND ITEM_SUBJECT_AREA like 'Mathematics'
                 GROUP BY NAME
                 ORDER BY NAME")

COUNTY_SUM_DF <- data.frame(tolower(str_remove(COUNTY_SUM_DF$NAME, ' COUNTY')),
                            COUNTY_SUM_DF$SUBGROUP_NAME, COUNTY_SUM_DF$TOTAL_TESTED, as.numeric(COUNTY_S

colnames(COUNTY_SUM_DF) <- c('name', 'subgroup', 'total', 'mean_scale_score')

library(ggplot2)
library(mapdata)
library(tidyverse)
```

```
#create map
ny <- map_data('county', 'new york')
COUNTY_SUM_DF_MAP <- merge(ny, COUNTY_SUM_DF, by.x = 'subregion', by.y = 'name', all.x = TRUE, all.y =
COUNTY_SUM_DF_MAP <- COUNTY_SUM_DF_MAP[order(COUNTY_SUM_DF_MAP$order),]


#create map with data
m <- ggplot(COUNTY_SUM_DF_MAP, aes(x=long,y=lat,group=group)) +
  geom_polygon(aes(fill=mean_scale_score))+
  scale_fill_gradientn(colours = cm.colors(50)) +
  geom_path() +
  coord_map() + ggtitle('Male Mathematics Mean Scale Score by County')

library(gridExtra)

grid.arrange(f,m, ncol = 2)
```

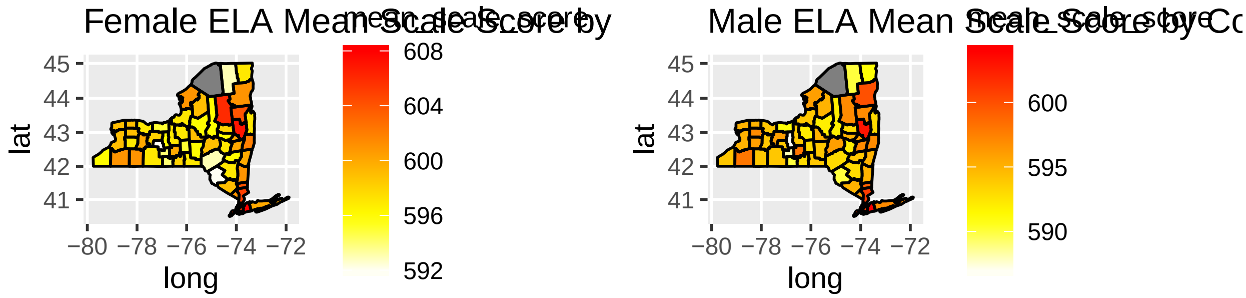


ELA Overview by County Gender Comparison

Repeating the steps above we can create an additonal map group of county scores by ELA instead of Math Scores.

```
##########################################################################################
################## Male vs Female ELA
##########################################################################################

#create a df based off of a selection from the main dataset
COUNTY_SUM_DF <- sqldf("SELECT NAME, SUBGROUP_NAME, TOTAL_TESTED, MEAN_SCALE_SCORE
                 from COUNTY_ELA
                 WHERE SUBGROUP_CODE is 2 AND ITEM_SUBJECT_AREA like 'ELA'
                 GROUP BY NAME
                 ORDER BY NAME")

COUNTY_SUM_DF <- data.frame(tolower(str_remove(COUNTY_SUM_DF$NAME, ' COUNTY')),
                            COUNTY_SUM_DF$SUBGROUP_NAME, COUNTY_SUM_DF$TOTAL_TESTED, as.numeric(COUNTY_

colnames(COUNTY_SUM_DF) <- c('name', 'subgroup', 'total', 'mean_scale_score')


#create map
ny <- map_data('county', 'new york')
COUNTY_SUM_DF_MAP <- merge(ny, COUNTY_SUM_DF, by.x = 'subregion', by.y = 'name', all.x = TRUE, all.y = 
COUNTY_SUM_DF_MAP <- COUNTY_SUM_DF_MAP[order(COUNTY_SUM_DF_MAP$order),]


#create map with data
f <- ggplot(COUNTY_SUM_DF_MAP, aes(x=long,y=lat,group=group)) +
  geom_polygon(aes(fill=mean_scale_score))+
  scale_fill_gradientn(colours = rev(heat.colors(50))) +
  geom_path() +
  coord_map() + ggtitle('Female ELA Mean Scale Score by County')

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

COUNTY_SUM_DF <- sqldf("SELECT NAME, SUBGROUP_NAME, TOTAL_TESTED, MEAN_SCALE_SCORE
                 from COUNTY_ELA
                 WHERE SUBGROUP_CODE is 3 AND ITEM_SUBJECT_AREA like 'ELA'
                 GROUP BY NAME
                 ORDER BY NAME")

COUNTY_SUM_DF <- data.frame(tolower(str_remove(COUNTY_SUM_DF$NAME, ' COUNTY')),
                            COUNTY_SUM_DF$SUBGROUP_NAME, COUNTY_SUM_DF$TOTAL_TESTED, as.numeric(COUNTY_

colnames(COUNTY_SUM_DF) <- c('name', 'subgroup', 'total', 'mean_scale_score')


#create map
ny <- map_data('county', 'new york')
COUNTY_SUM_DF_MAP <- merge(ny, COUNTY_SUM_DF, by.x = 'subregion', by.y = 'name', all.x = TRUE, all.y = 
COUNTY_SUM_DF_MAP <- COUNTY_SUM_DF_MAP[order(COUNTY_SUM_DF_MAP$order),]
```

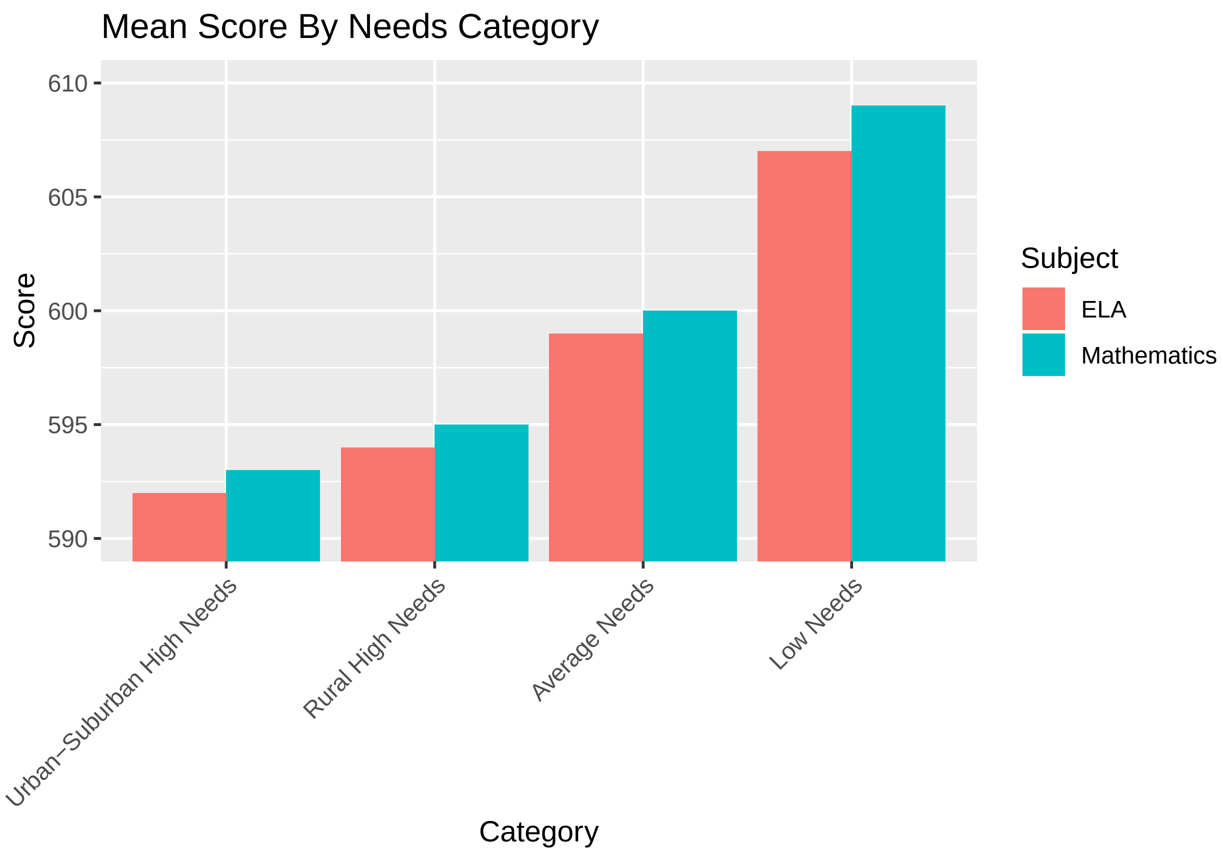
```r
#create map with data
m <- ggplot(COUNTY_SUM_DF_MAP, aes(x=long,y=lat,group=group)) +
  geom_polygon(aes(fill=mean_scale_score))+
  scale_fill_gradientn(colours = rev(heat.colors(50))) +
  geom_path() +
  coord_map() + ggtitle('Male ELA Mean Scale Score by County')


grid.arrange(f,m,ncol = 2)
```



Score Comparison based on NRC categorization

This next chunk of r-code creates two data frames, groupings of ELA and MATH mean scale scores grouped by NRC categorization. These two data frames are joined and their order readjusted in order to create a barplot that shows both math and ELA mean scale score by NRC categorization. Color coded according to the test type.

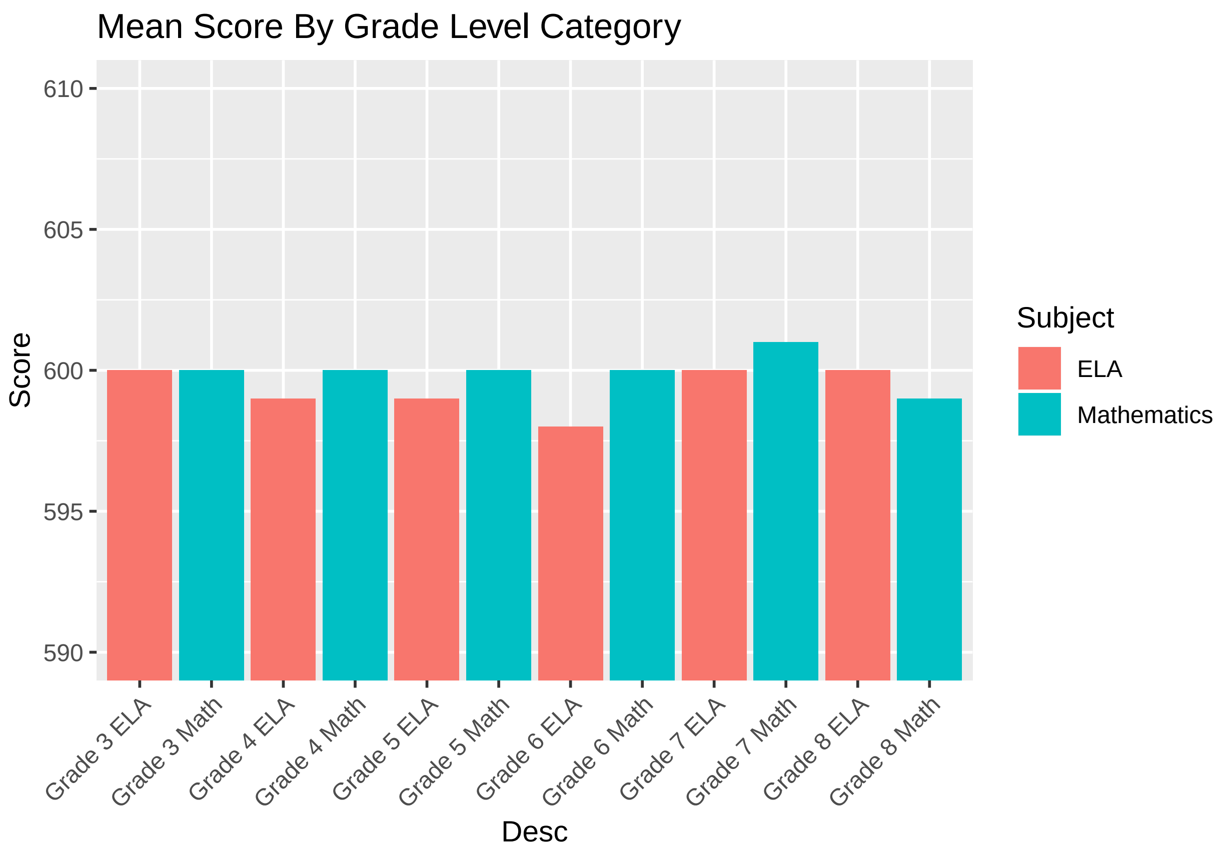
```
##########################################################################################
################## Needs Mean Score Barplot
##########################################################################################

#create a df based off of a selection from the main dataset
NRC_NEEDS_MATH <- sqldf("SELECT NRC_DESC, ITEM_SUBJECT_AREA, MEAN_SCALE_SCORE
                        from NRC_NEEDS_ELA
                        WHERE SUBGROUP_CODE is 1 AND ITEM_SUBJECT_AREA like 'Mathematics'
                        GROUP BY NRC_CODE")

#create a df based off of a selection from the main dataset
NRC_NEEDS_ELA_S <- sqldf("SELECT NRC_DESC, ITEM_SUBJECT_AREA, MEAN_SCALE_SCORE
                          from NRC_NEEDS_ELA
                          WHERE SUBGROUP_CODE is 1 AND ITEM_SUBJECT_AREA like 'ELA'
                          GROUP BY NRC_CODE")

NEEDS_SUMMARY <- rbind(NRC_NEEDS_ELA_S, NRC_NEEDS_MATH)

NEEDS_SUMMARY <- data.frame(paste(NEEDS_SUMMARY$NRC_DESC, NEEDS_SUMMARY$ITEM_SUBJECT_AREA),
                            as.numeric(NEEDS_SUMMARY$MEAN_SCALE_SCORE), NEEDS_SUMMARY$NRC_DESC, NEEDS_SU

colnames(NEEDS_SUMMARY) <- c('Desc', 'Score', 'Category', 'Subject')

order_nrc <- c('Urban-Suburban High Needs', 'Rural High Needs', 'Average Needs','Low Needs')

NEEDS_SUMMARY$Category <- factor(NEEDS_SUMMARY$Category, levels = order_nrc)

ggplot(NEEDS_SUMMARY, aes(x = Category, y = Score, fill = Subject)) +
  geom_bar(stat = 'identity', position=position_dodge()) +
  coord_cartesian(ylim = c(590,610)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle('Mean Score By Needs Category')
```


Mean Score By Needs Category
Score
610
605
600
595
590
Urban–Suburban High Needs
Rural High Needs
Average Needs
Low Needs
Category
Subject
ELA
Mathematics

Score Comparison based on Grade Level

Then we repeated the previous ggplot to create an additional plot that uses the student grade level as the grouping instead of NRC categorization.

```
##########################################################################################
################## Grades Summary Mean Score Barplot
##########################################################################################

#create a df based off of a selection from the main dataset
STATE_G_MATH <- sqldf("SELECT ITEM_DESC, ITEM_SUBJECT_AREA, MEAN_SCALE_SCORE
                        from STATE_ELA
                        WHERE SUBGROUP_CODE is 1 AND ITEM_SUBJECT_AREA like 'Mathematics'
                        GROUP BY ITEM_DESC")

#create a df based off of a selection from the main dataset
STATE_G_ELA <- sqldf("SELECT ITEM_DESC, ITEM_SUBJECT_AREA, MEAN_SCALE_SCORE
                          from STATE_ELA
                          WHERE SUBGROUP_CODE is 1 AND ITEM_SUBJECT_AREA like 'ELA'
                          GROUP BY ITEM_DESC")

GRADES_SUMMARY <- rbind(STATE_G_ELA, STATE_G_MATH)

GRADES_SUMMARY <- data.frame(GRADES_SUMMARY$ITEM_DESC, GRADES_SUMMARY$ITEM_SUBJECT_AREA,
                             as.numeric(GRADES_SUMMARY$MEAN_SCALE_SCORE))

colnames(GRADES_SUMMARY) <- c('Desc', 'Subject', 'Score')

ggplot(GRADES_SUMMARY, aes(x = Desc, y = Score, fill = Subject)) +
  geom_bar(stat = 'identity', position=position_dodge()) +
  coord_cartesian(ylim = c(590,610)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle('Mean Score By Grade Level Category')
```

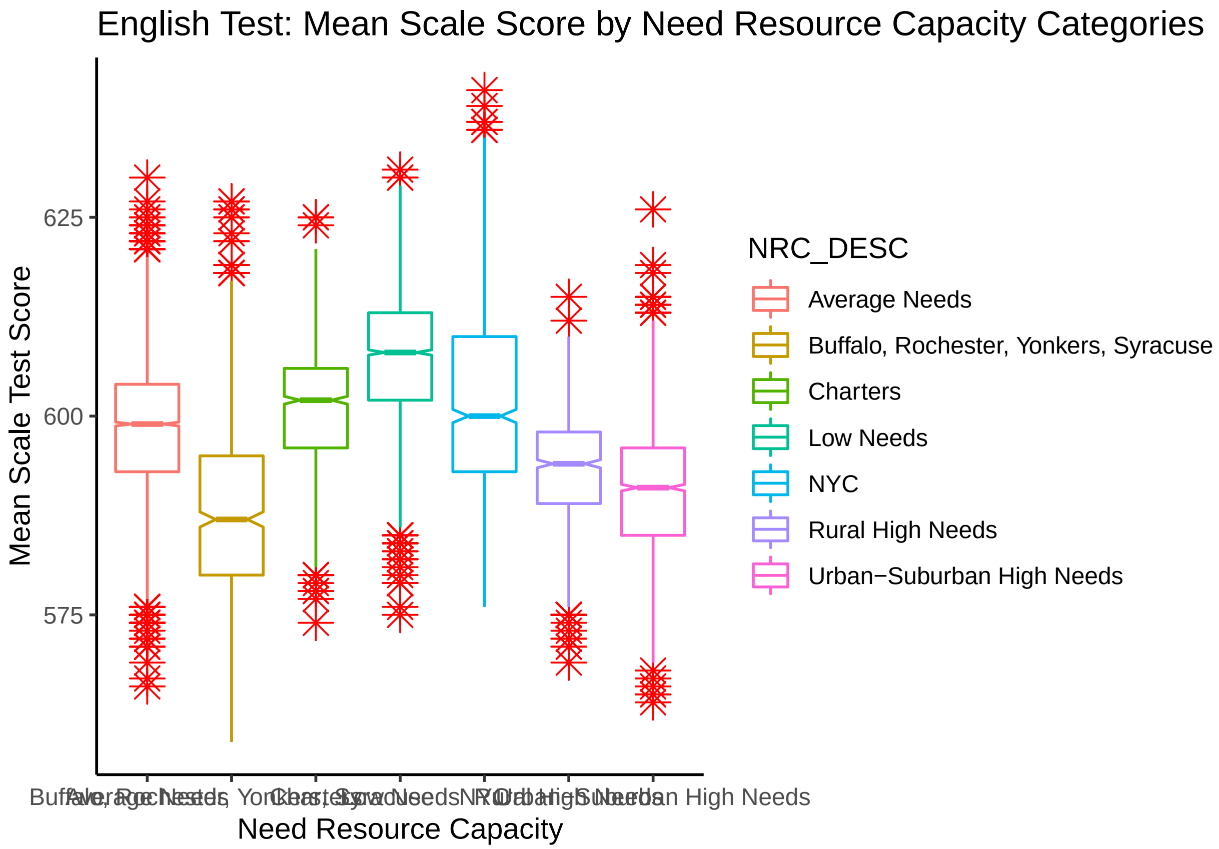

Mean Score By Grade Level Category
Score
610
605
600
595
590
Grade 3 ELA
Grade 3 Math
Grade 4 ELA
Grade 4 Math
Grade 5 ELA
Grade 5 Math
Grade 6 ELA
Grade 6 Math
Grade 7 ELA
Grade 7 Math
Grade 8 ELA
Grade 8 Math
Desc
Subject
ELA
Mathematics

## Score Comparison based on Race

Next we create a few boxplots that show the mean scale scores statistics by race. In this you can clearly see how the different scores differ amongst different racial identities.

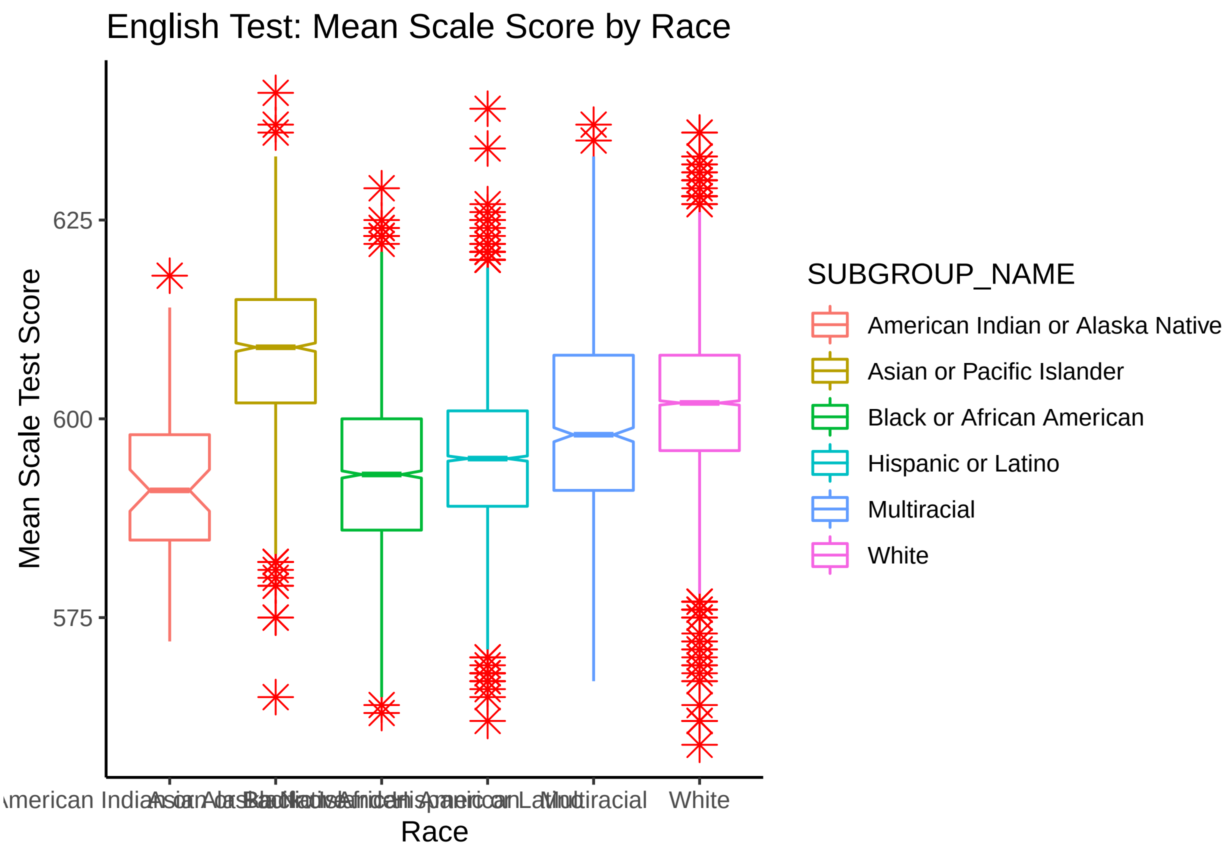
```
# English Test
sch_race_eng <- sch_race[sch_race$ITEM_SUBJECT_AREA=="ELA",] # English
# boxplot Mean scale test score by NRC
ggplot(sch_race_eng, aes(x=NRC_DESC, y=MEAN_SCALE_SCORE, color=NRC_DESC))+
  geom_boxplot(outlier.colour="red", outlier.shape=8,
               outlier.size=4,notch=TRUE)+
  labs(title="English Test: Mean Scale Score by Need Resource Capacity Categories",x="Need Resource Capa
  theme_classic()
```

```
## Warning: Removed 17492 rows containing non-finite values (stat_boxplot).
```



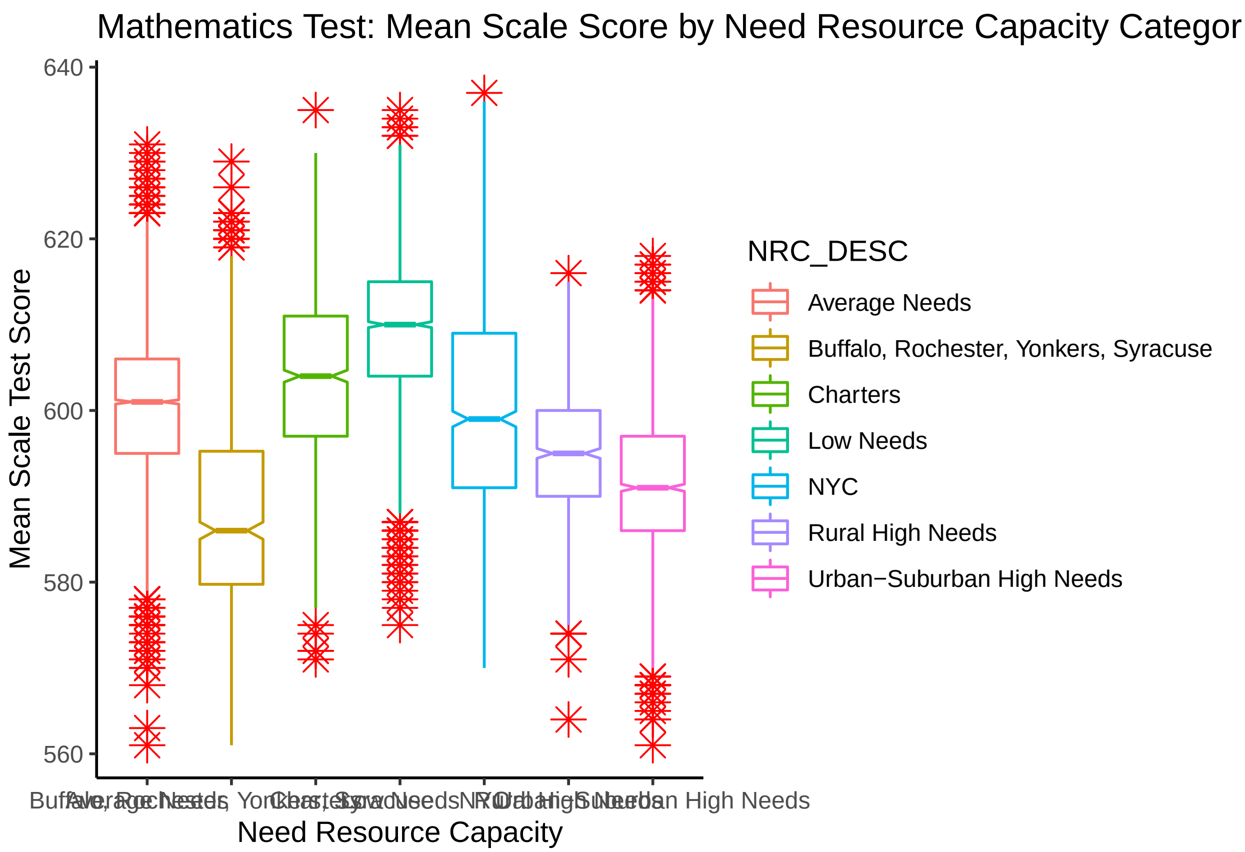
```
ggplot(sch_race_eng, aes(x=SUBGROUP_NAME, y=MEAN_SCALE_SCORE, color=SUBGROUP_NAME))+
  geom_boxplot(outlier.colour="red", outlier.shape=8,
               outlier.size=4,notch=TRUE)+
  labs(title="English Test: Mean Scale Score by Race",x="Race", y = "Mean Scale Test Score")+
  theme_classic()
```

```
## Warning: Removed 17492 rows containing non-finite values (stat_boxplot).
```



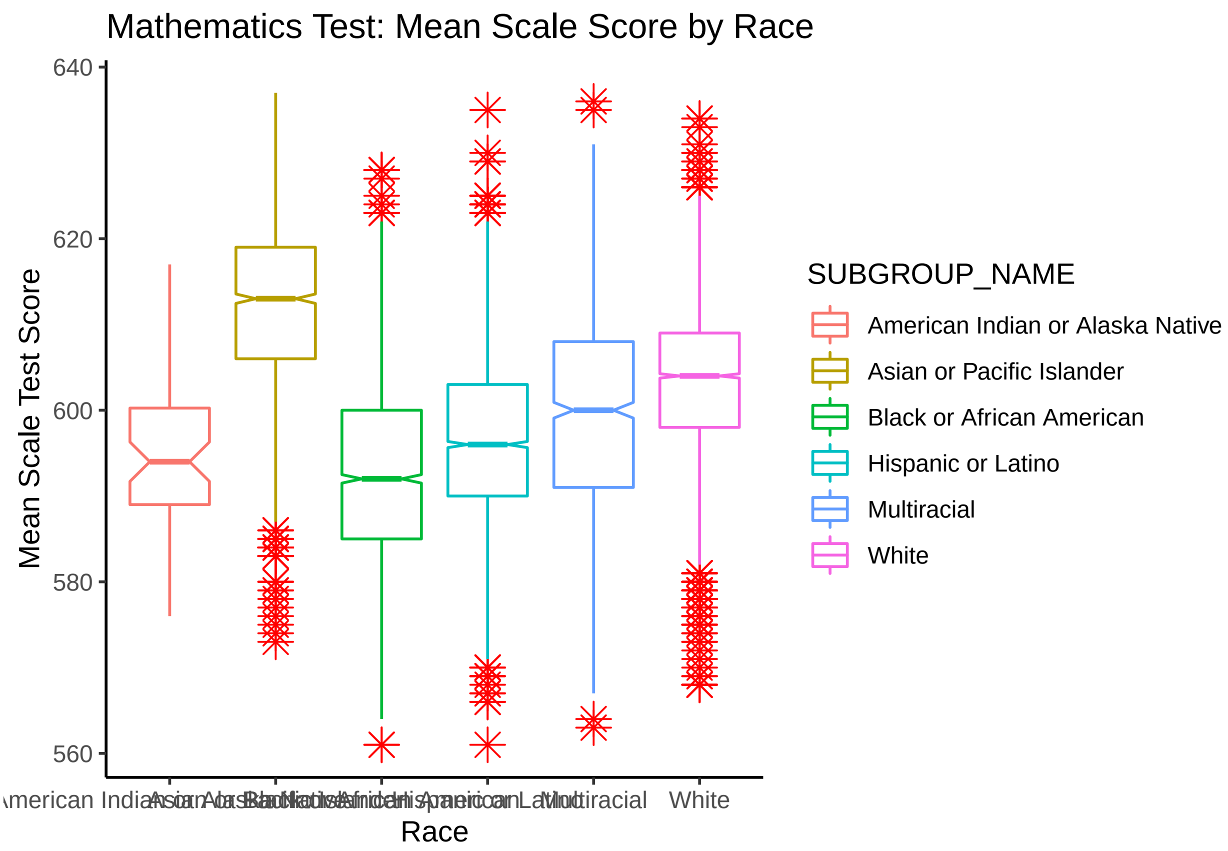
```
# English Test
sch_race_math <- sch_race[sch_race$ITEM_SUBJECT_AREA=="Mathematics",] # Mathematics

# boxplot Mean scale test score by NRC
ggplot(sch_race_math, aes(x=NRC_DESC, y=MEAN_SCALE_SCORE, color=NRC_DESC))+
  geom_boxplot(outlier.colour="red", outlier.shape=8,
               outlier.size=4,notch=TRUE)+
  labs(title="Mathematics Test: Mean Scale Score by Need Resource Capacity Categories",x="Need Resource
  theme_classic()
```

```
## Warning: Removed 17434 rows containing non-finite values (stat_boxplot).
```



```r
ggplot(sch_race_math, aes(x=SUBGROUP_NAME, y=MEAN_SCALE_SCORE, color=SUBGROUP_NAME))+
  geom_boxplot(outlier.colour="red", outlier.shape=8,
               outlier.size=4,notch=TRUE)+
  labs(title="Mathematics Test: Mean Scale Score by Race",x="Race", y = "Mean Scale Test Score")+
  theme_classic()
```

```
## Warning: Removed 17434 rows containing non-finite values (stat_boxplot).
```


Mathematics Test: Mean Scale Score by Race
Mean Scale Test Score
640
620
600
580
560
American Indian or Alaska Native
Asian or Pacific Islander
Black or African American
Hispanic or Latino
Multiracial
White
Race
SUBGROUP_NAME
American Indian or Alaska Native
Asian or Pacific Islander
Black or African American
Hispanic or Latino
Multiracial
White

# Machine Learning Attempts at machine learning reveals some unexpended results. Firstly take some ggplots of regular mean score and l1,l2 etc for deeper understanding. Then run some linear regressions and attempt to run neural networks and Random Forest.

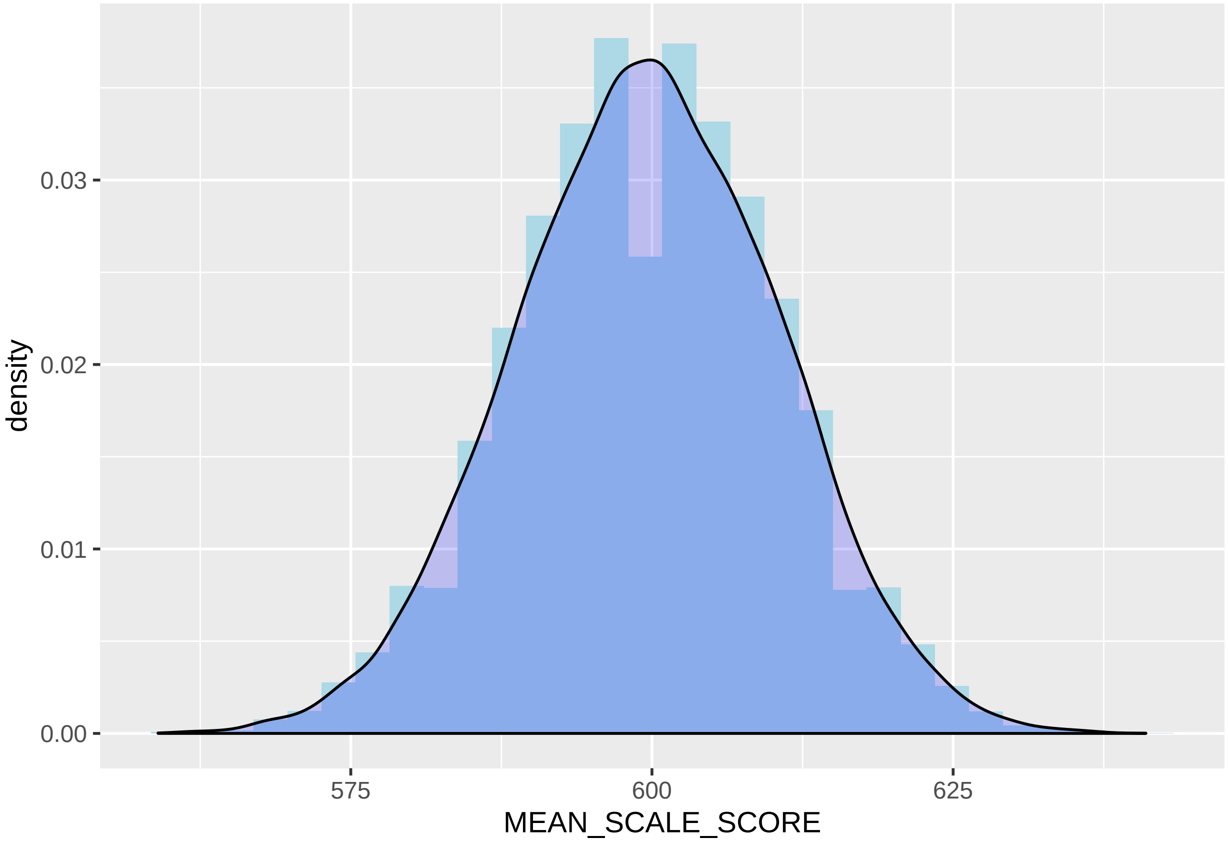
Firstly here is a normal distribution of the mean scale scores, just to get a better picture of the data in your head.

```
# 3. Preliminary Data Viz School level data - Race counts.
ggplot(sch_race, aes(x=MEAN_SCALE_SCORE))+
  geom_histogram(aes(y=..density..),fill="lightblue")+
  geom_density(alpha=.2,fill="blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 34926 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 34926 rows containing non-finite values (stat_density).
```



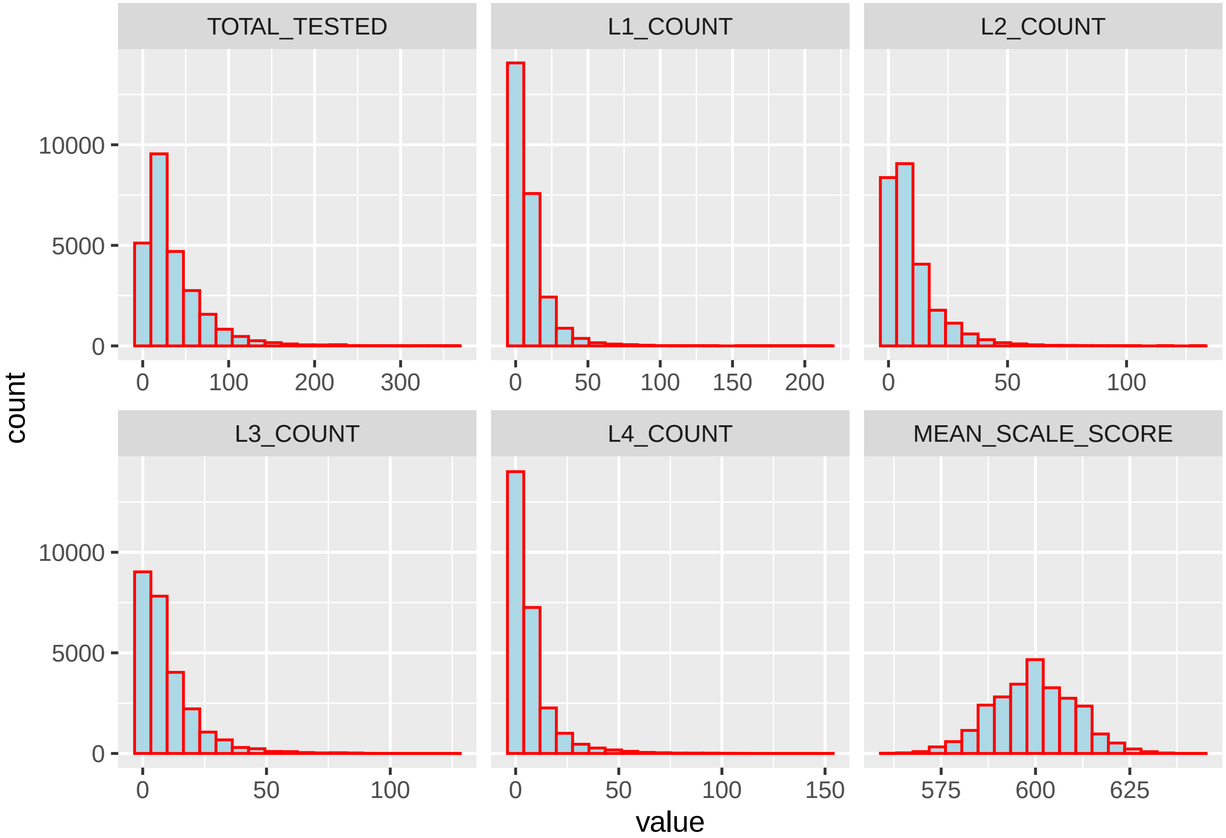
Look at Test Counts and Level Test Counts

Now we attempt to plot the different testing levels within the data frame in order to gain some deeper understanding of how everything is 'shaped.'

```
# sch_race_1 <- sch_race[,11:15]
# Facet wrap for the Test level count and mean Scale score for subject and all grade
ggplot(data = melt(sch_race[,10:15]), mapping = aes(x = value)) +
  geom_histogram(bins = 20, fill="lightblue",color="red") +
  facet_wrap(~variable, scales = 'free_x')
```

```
## No id variables; using all as measure variables
```

```
## Warning: Removed 209556 rows containing non-finite values (stat_bin).
```



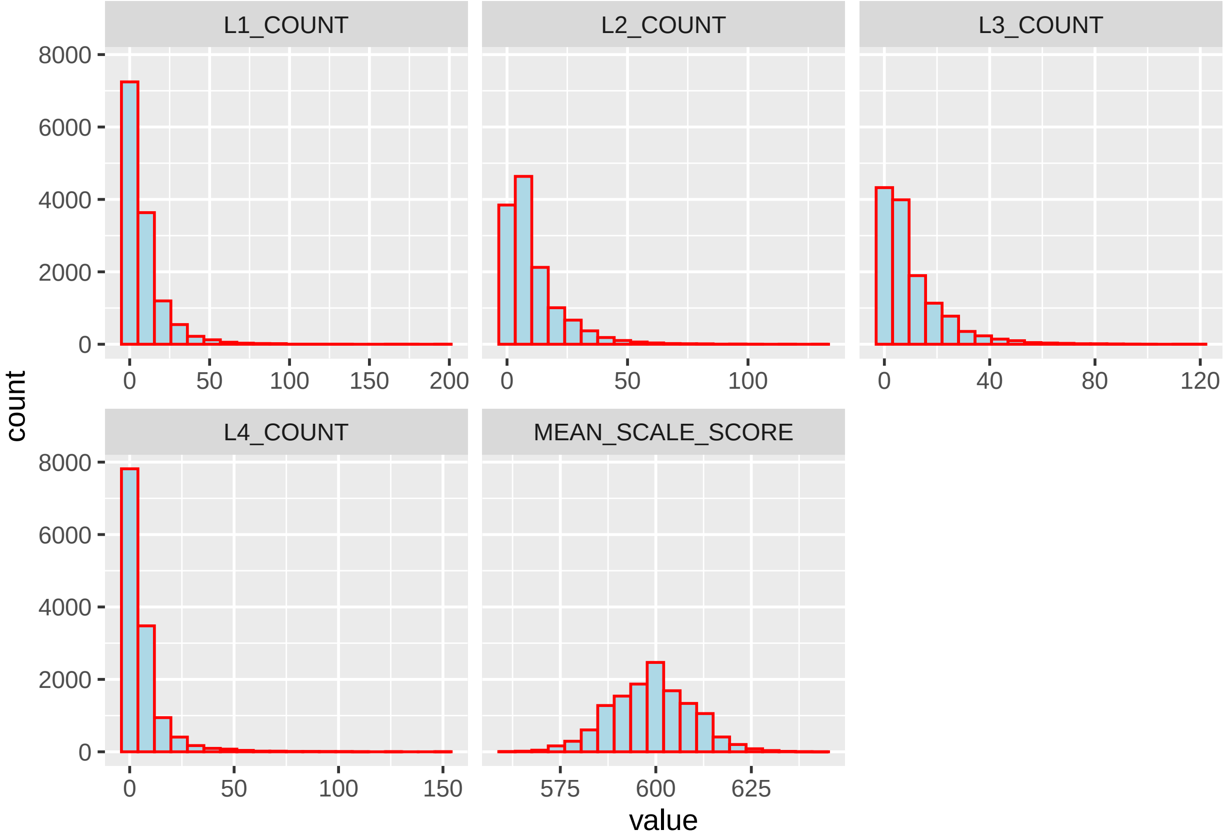
Narrow to ELA only

Preform a similar plot like the one above but instead focus on ELA scores

```
# subsetting by subject
# 1 English Test
sch_race_eng <- sch_race[sch_race$ITEM_SUBJECT_AREA=="ELA",] # English

ggplot(data = melt(sch_race_eng[,11:15]), mapping = aes(x = value)) +
  geom_histogram(bins = 20, fill="lightblue", color="red") +
  facet_wrap(~variable, scales = 'free_x')
```

```
## No id variables; using all as measure variables
```

```
## Warning: Removed 87460 rows containing non-finite values (stat_bin).
```



Machine Learning RF

In this chunk of code we attempt to build a linear regression using LM and plotting the results, which creates some very interesting visualizations.

Then we attempt some randomForests, iterate and view their results. Later ploting the results of our trained prediction

```
#clean the nas out of df
sch_race_eng <- sch_race_eng[,-c(3)]
sch_race_eng <- na.omit(sch_race_eng)



# building linear regression model to predict mean scale score for english across all 3 : 8
# testing for collinearity
R_cor_eng <- cor(sch_race_eng[, c( "L1_COUNT",
                                   "L2_COUNT", "L3_COUNT", "L4_COUNT","MEAN_SCALE_SCORE")], use = "compl
R_cor_eng
```

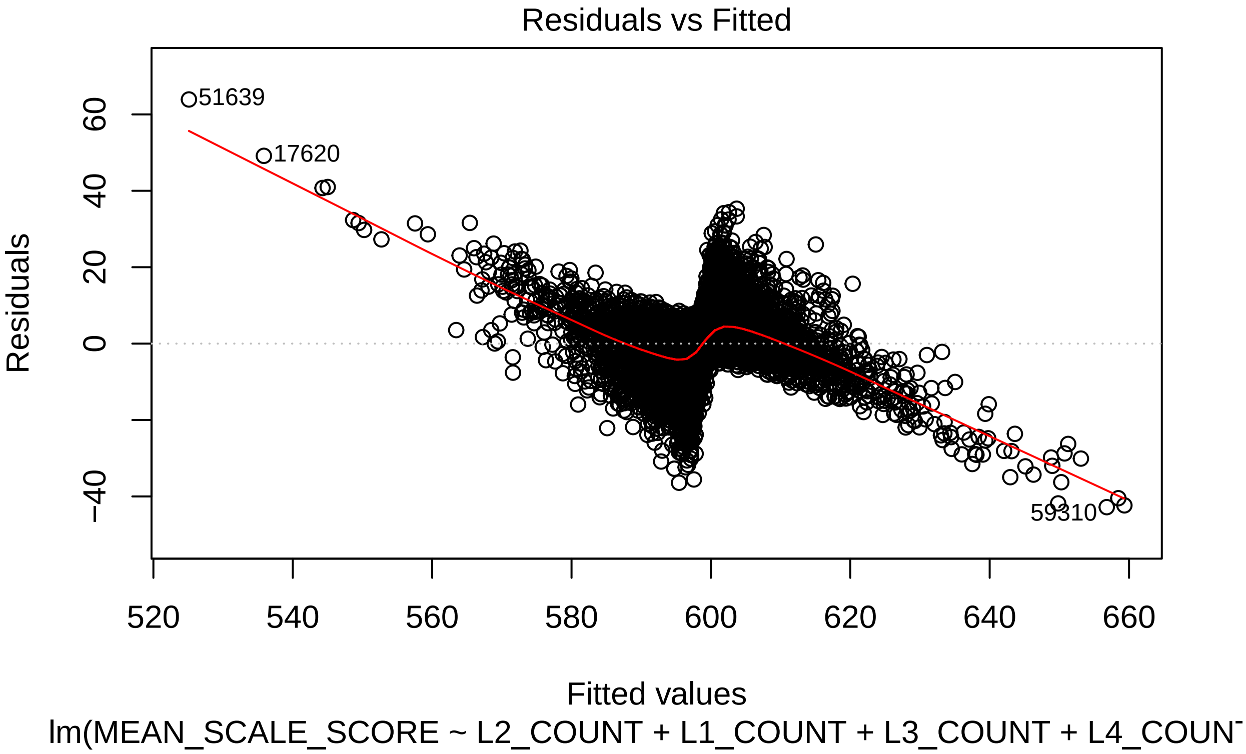
```
##                     L1_COUNT    L2_COUNT  L3_COUNT  L4_COUNT
## L1_COUNT           1.0000000  0.65456623 0.3046614 0.2055635
## L2_COUNT           0.6545662  1.00000000 0.7335029 0.4055438
## L3_COUNT           0.3046614  0.73350293 1.0000000 0.5562889
## L4_COUNT           0.2055635  0.40554378 0.5562889 1.0000000
## MEAN_SCALE_SCORE -0.3885904 -0.07682174 0.2657428 0.3887112
##                  MEAN_SCALE_SCORE
## L1_COUNT              -0.38859040
## L2_COUNT              -0.07682174
## L3_COUNT               0.26574284
## L4_COUNT               0.38871120
## MEAN_SCALE_SCORE       1.00000000
```

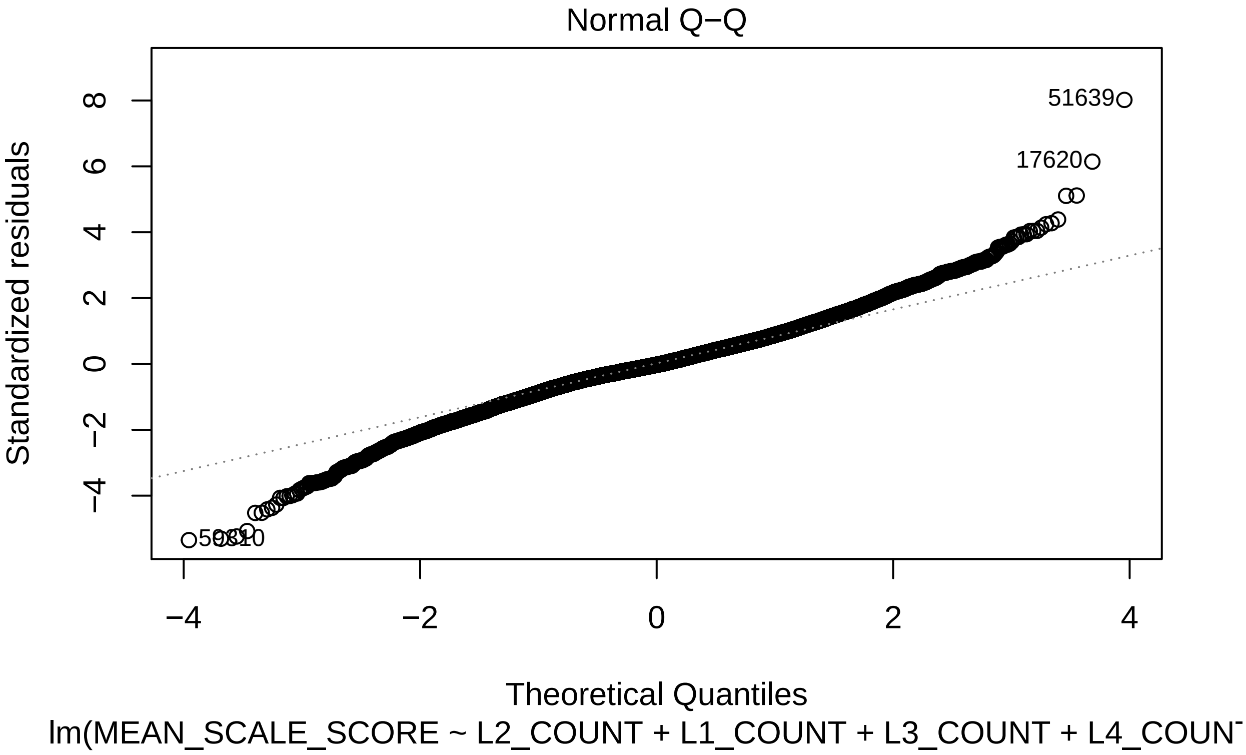
```
# there is collinearity between all vasriables except L2_count
# linear regression
linearm <- lm(MEAN_SCALE_SCORE~L2_COUNT + L1_COUNT + L3_COUNT + L4_COUNT , data=sch_race_eng)
summary(linearm)
```

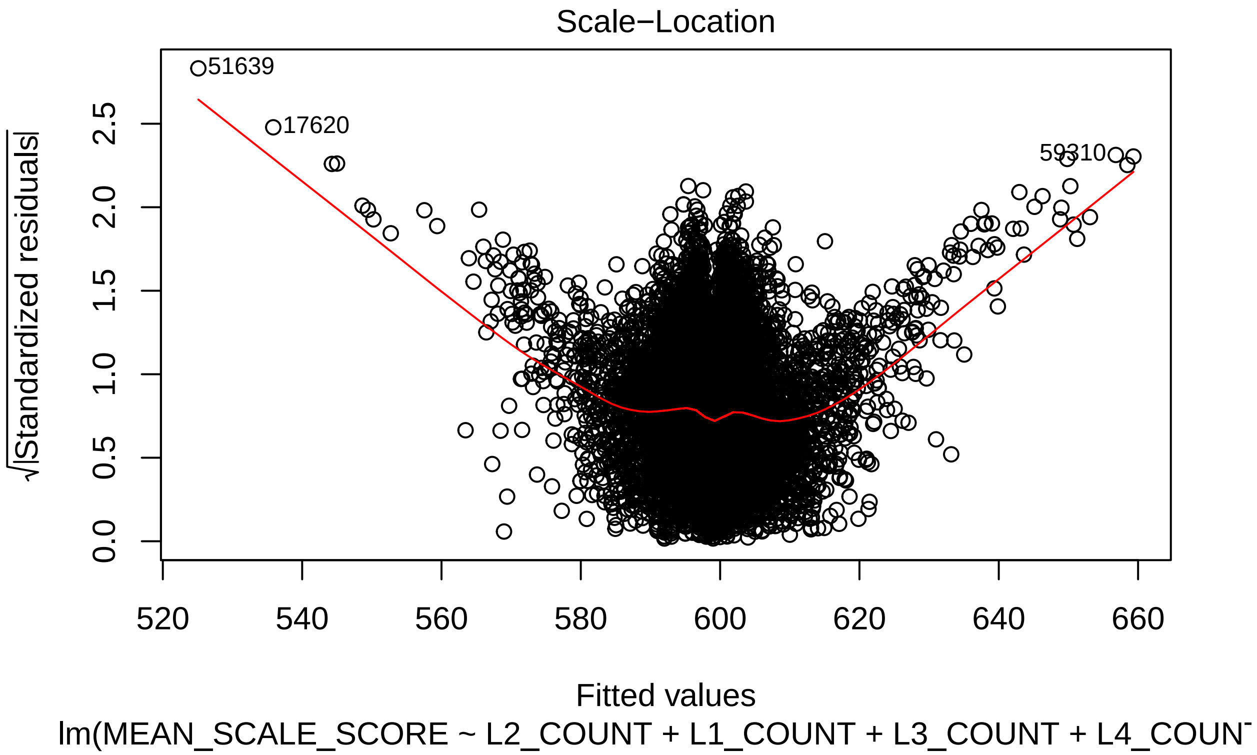
```
##
## Call:
## lm(formula = MEAN_SCALE_SCORE ~ L2_COUNT + L1_COUNT + L3_COUNT +
##     L4_COUNT, data = sch_race_eng)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.802  -4.283  -0.205   4.597  63.904
##
## Coefficients:
##               Estimate Std. Error t value            Pr(>|t|)
## (Intercept) 598.861510   0.099958 5991.13 <0.0000000000000002 ***
## L2_COUNT     -0.147296   0.012069  -12.20 <0.0000000000000002 ***
## L1_COUNT     -0.396527   0.008176  -48.50 <0.0000000000000002 ***
## L3_COUNT      0.281595   0.010229   27.53 <0.0000000000000002 ***
## L4_COUNT      0.388055   0.008386   46.28 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

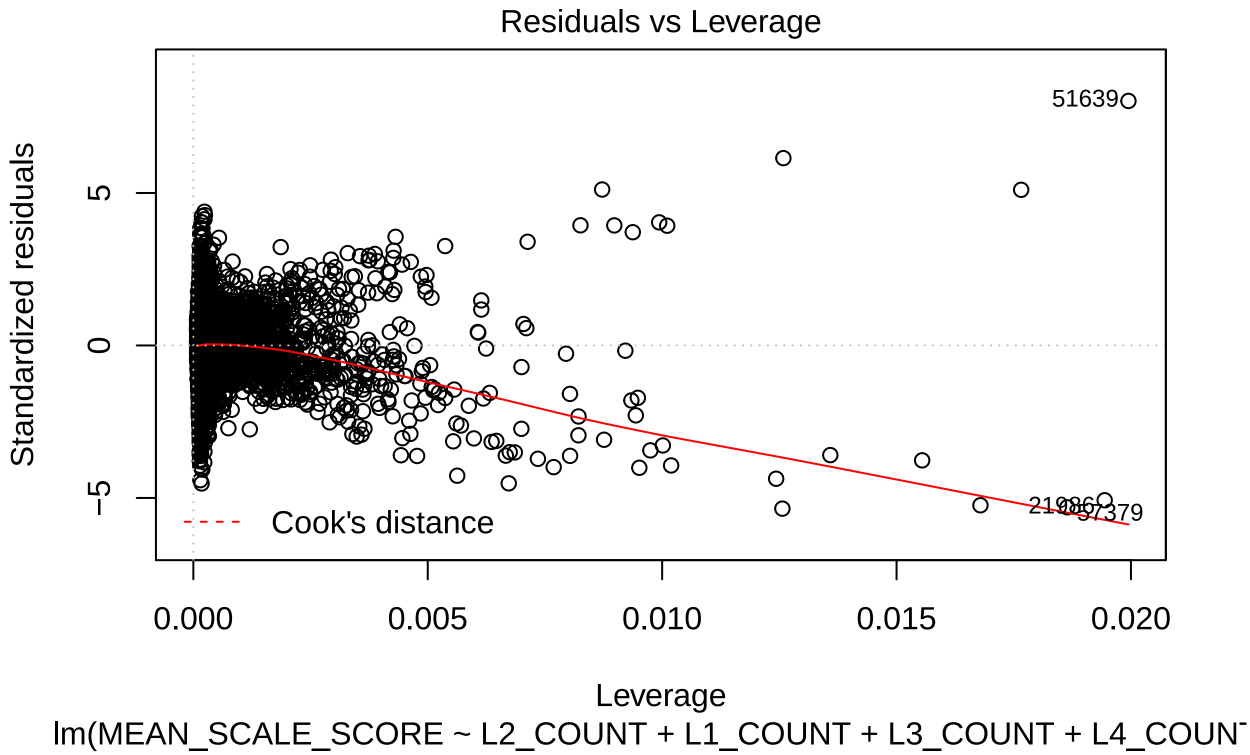
```
## Residual standard error: 8.051 on 13092 degrees of freedom
## Multiple R-squared:  0.418,  Adjusted R-squared:  0.4179
## F-statistic:  2351 on 4 and 13092 DF,  p-value: < 0.00000000000000022
```

```
sch_race_eng$Pred <- linearm$fitted.values
sch_race_eng$Resid <- linearm$residuals
plot(linearm)
```




Normal Q−Q
Standardized residuals
8
6
4
2
0
−2
−4
51639
17620
59810
−4
−2
0
2
4
Theoretical Quantiles
lm(MEAN_SCALE_SCORE ~ L2_COUNT + L1_COUNT + L3_COUNT + L4_COUN


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
51639
17620
59310
520
540
560
580
600
620
640
660
Fitted values
lm(MEAN_SCALE_SCORE ~ L2_COUNT + L1_COUNT + L3_COUNT + L4_COUN



```
# building maching learning model using RandomForest to predict mean scale score for english across all
sch_race_eng_1 <-sch_race_eng[,11:15]
#  Train classifer (clf)
eng_1 <- randomForest(sch_race_eng_1[,-5],sch_race_eng_1[,5])
```

```
## Warning in randomForest.default(sch_race_eng_1[, -5], sch_race_eng_1[, 5]):
## The response has five or fewer unique values. Are you sure you want to do
## regression?
```

```
eng_1
```

```
##
## Call:
##  randomForest(x = sch_race_eng_1[, -5], y = sch_race_eng_1[, 5])
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 0.004923544
##                     % Var explained: -1.25
```

```
#  Iterate
eng_1a <- randomForest(sch_race_eng_1[,-5],as.factor(sch_race_eng_1[,5]))
eng_1a
```

```
##
## Call:
##  randomForest(x = sch_race_eng_1[, -5], y = as.factor(sch_race_eng_1[,      5]))
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.49%
## Confusion matrix:
##       0 1 class.error
## 0 13033 0           0
## 1    64 0           1
```

```
num_exmps = nrow(sch_race_eng_1)
L = replace(integer(num_exmps), sch_race_eng_1[,5]>=600, 1)
M <- sch_race_eng_1[,-5]
```

```
#Use Cross validation to build model
train_eng <- sample(c(1:num_exmps), size = num_exmps * 0.7, replace = FALSE)
eng_3 <- randomForest(M[train_eng,],as.factor(L[train_eng]))
#Generate propsoed answers using Cross validation
pred <- predict(eng_3, M[-train_eng,],type="prob")

#9. Plot ROC metric
plot(roc(L[-train_eng], as.numeric(pred[,1])))
```

Machine Learning Nueral Net

Here we attempt to utilize nueral net for some additional training, we were not as effective in getting this to succesfully run, but include it at the tail of the report in any case.

```
# Build neural networt model



# neuralnetwork model
# eng_neutral <- neuralnet(MEAN_SCALE_SCORE ~ L1_COUNT + L2_COUNT + L3_COUNT  + L4_COUNT,
#                         sch_race_eng_1, hidden =3, lifesign = "minimal", linear.output = FALSE, act.fc
randIndex   <- sample(1:nrow(sch_race_eng_1))
head(randIndex)
nr<- nrow(sch_race_eng_1)
nr
eng_12_3    <- floor(2*nr/3)
eng_12_3

trainMSS <-sch_race_eng_1[randIndex[1:eng_12_3],]
testMSS <-sch_race_eng_1[randIndex[(eng_12_3+1):nr],]


eng_neutral <- neuralnet(MEAN_SCALE_SCORE ~ L1_COUNT + L2_COUNT + L3_COUNT  + L4_COUNT,
                         trainMSS, hidden =5, lifesign = "minimal", linear.output = FALSE,threshold = 0
summary(eng_neutral)
plot(eng_neutral)

pred <- compute(eng_neutral,testMSS)
head(pred$net.result)
summary(pred)
pred
```