Statistics 139 Final Project

Mirai Shah, Paul Stainier, Ramtin Talebi, and Rajet Vatsa

Appendix

The following pages contain the source R code for our group's final project analyses. The appendix is broken down into sub-sections that help narrate the story of our full analysis, which we eventually distilled into our final paper.

Preliminary Data Analysis

We began by importing our district-level dataset from Kaggle's Education in India database. Based on a preliminary examination of the metadata, we filtered out the variables that we deemed unimportant:

```
# Set up caching
knitr::opts_chunk$set(cache = TRUE)

# Import Kaggle dataset
districtdata <- read.csv("2015_16_Districtwise.csv")

# Determine variables of interest
myvars <- c("DISTCD", "STATCD", "DISTNAME",
            "TOTPOPULAT", "P_URB_POP",
            "POPULATION_0_6", "GROWTHRATE",
            "SEXRATIO", "P_SC_POP", "P_ST_POP",
            "OVERALL_LI", "FEMALE_LIT", "MALE_LIT",
            "AREA_SQKM", "TOT_6_10_15", "TOT_11_13_15",
            "SCHTOT", "SCHTOTG", "SCHTOTP", "SCHTOTM",
            "SCHTOTGR", "SCHTOTPR",
            "SCHBOYTOT", "SCHGIRTOT", "ENRTOT",
            "SCLSTOT", "STCHTOT", "ROADTOT", "SWATTOT",
            "SELETOT", "SCOMPTOT")

# Re-define predictor/column names
column_names <- c("dist_code", "state_code", "dist_name",
                  "total_pop", "p_urban_pop",
                  "pop0to6", "growth_rate",
                  "sex_ratio", "p_sched_castes",
                  "p_sched_tribes", "overall_lit",
                  "female_lit", "male_lit", "area_sqkm",
                  "pop6to10", "pop11to13", "tot_schools",
                  "tot_gov_schools", "tot_priv_schools",
                  "tot_unrec", "rural_gov_schools",
                  "rural_priv_schools", "boys_schools",
```

```r
                  "girls_schools", "elementary_enrollment",
                  "single_classroom", "single_teacher",
                  "tot_road_accessible", "tot_drinking_water",
                  "tot_electricity", "tot_computer")

# Filter out unimportant variables
newdistrictdata <- districtdata[myvars]

# Update predictor/column names
colnames(newdistrictdata) <- column_names

# Remove NA terms
newdistrictdata <- na.omit(newdistrictdata)
```

Considering our intended aim was to conduct a state-by-state analysis, we converted the majority of the predictors from absolute values per district to percentages. To do so, we divided by the total number of schools in or the total population of the respective district, based on the predictor.

```r
# Converting demographic data to percentages (per population)
newdistrictdata$p_pop0to6 <- newdistrictdata$pop0to6/
  newdistrictdata$total_pop * 100
newdistrictdata$p_pop6to10 <- newdistrictdata$pop6to10/
  newdistrictdata$total_pop * 100
newdistrictdata$p_pop11to13 <- newdistrictdata$pop11to13/
  newdistrictdata$total_pop * 100
newdistrictdata$p_elementary_enrollment <-
  newdistrictdata$elementary_enrollment/
  newdistrictdata$total_pop * 100

# Converting school data to percentage (per population)
newdistrictdata$p_capita_schools <- newdistrictdata$tot_schools/
  newdistrictdata$total_pop * 100

# Converting school data to percentage (per total # schools)
newdistrictdata$p_gov_school <- newdistrictdata$tot_gov_schools/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_priv_school <- newdistrictdata$tot_priv_schools/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_unrec <- newdistrictdata$tot_unrec/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_gov_rur <- newdistrictdata$rural_gov_schools/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_priv_rur <- newdistrictdata$rural_priv_schools/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_boy_school <- newdistrictdata$boys_schools/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_girl_school <- newdistrictdata$girls_schools/
  newdistrictdata$tot_schools * 100
```

```
newdistrictdata$p_single_class <- newdistrictdata$single_classroom/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_single_teacher <- newdistrictdata$single_teacher/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_road_accessible <- newdistrictdata$tot_road_accessible/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_drink_water <- newdistrictdata$tot_drinking_water/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_electricity <- newdistrictdata$tot_electricity/
  newdistrictdata$tot_schools * 100
newdistrictdata$p_computer <- newdistrictdata$tot_computer/
  newdistrictdata$tot_schools * 100
```

Accordingly, we re-defined the variables in our dataset.

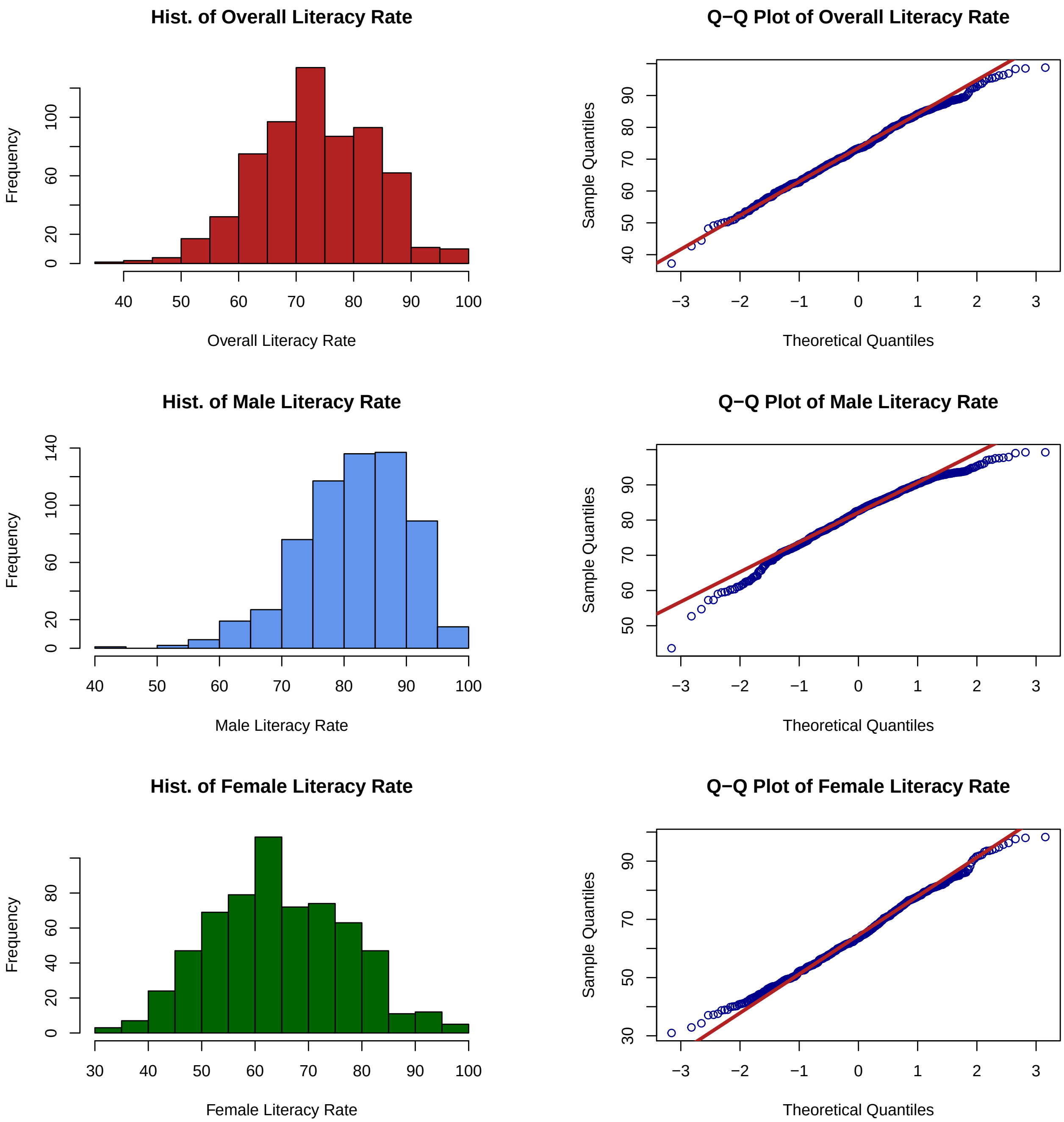
```
# Re-defined variable of interest
new_vars <- c("dist_code", "state_code", "dist_name",
                  "total_pop", "p_urban_pop",
                  "p_pop0to6", "growth_rate",
                  "sex_ratio", "p_sched_castes",
                  "p_sched_tribes", "overall_lit",
                  "female_lit", "male_lit", "area_sqkm",
                  "p_pop6to10", "p_pop11to13", "p_capita_schools",
                  "p_gov_school", "p_priv_school",
                  "p_unrec", "p_gov_rur",
                  "p_priv_rur", "p_boy_school",
                  "p_girl_school", "p_elementary_enrollment",
                  "p_single_class", "p_single_teacher",
                  "p_road_accessible", "p_drink_water",
                  "p_electricity", "p_computer")

# Filter out unimportant variables
newdistrictdata <- newdistrictdata[new_vars]
```

Response Variable Transformations

We then examined the distributions of our three response variables: overall literacy rate, male literacy rate, and female literacy rate.

```
# QQ-plot and hist of untransformed literacy rate response vars
par(mfrow = c(3, 2), mai=c(0.75,0.75,0.75,0.75), cex = 0.85)

# Histogram of overall literacy rate
hist(newdistrictdata$overall_lit,
     main="Hist. of Overall Literacy Rate",
     xlab="Overall Literacy Rate",col="firebrick")

# Q-Q plot for overall literacy rate
```

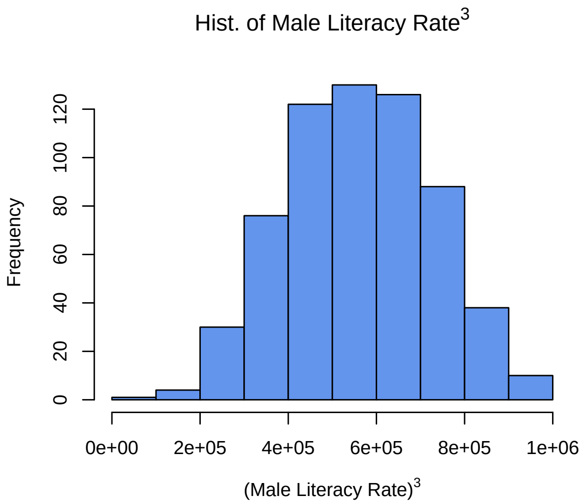
```r
qqnorm(newdistrictdata$overall_lit,
       main="Q-Q Plot of Overall Literacy Rate",
       col="darkblue")
qqline(newdistrictdata$overall_lit,col="firebrick",lwd=3)

# Histogram of male literacy rate
hist(newdistrictdata$male_lit,
     main="Hist. of Male Literacy Rate",
     xlab="Male Literacy Rate",col="cornflowerblue")

# Q-Q plot for male literacy rate
qqnorm(newdistrictdata$male_lit,
       main="Q-Q Plot of Male Literacy Rate",
       col="darkblue")
qqline(newdistrictdata$male_lit,col="firebrick",lwd=3)

# Histogram of female literacy rate
hist(newdistrictdata$female_lit,
     main="Hist. of Female Literacy Rate",
     xlab="Female Literacy Rate",col="darkgreen")

# Q-Q plot for female literacy rate
qqnorm(newdistrictdata$female_lit,
       main="Q-Q Plot of Female Literacy Rate",
       col="darkblue")
qqline(newdistrictdata$female_lit,col="firebrick",lwd=3)
```

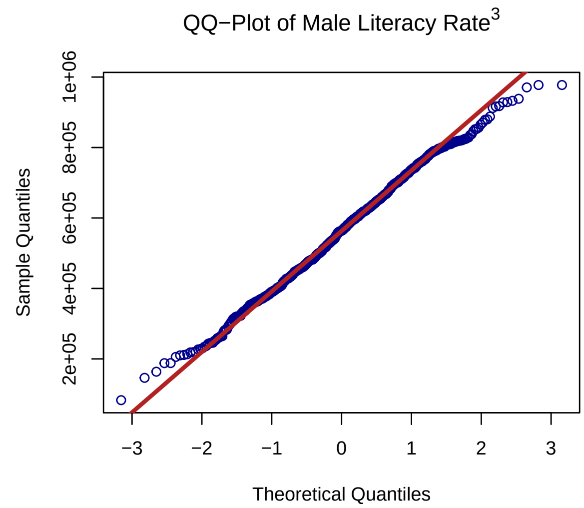


Based on these distributions, we left overall literacy rate and female literacy rate untransformed. Given that the QQ-plot for male literacy rate curved below the Normal QQ-line (indicating left-skew), we transformed the male literacy rate data with a higher-power cube transformation.

```r
# QQ-plot and hist of transformed male literacy rate
par(mfrow = c(1, 2), mai=c(0.75,0.75,0.75,0.75), cex = 0.85)

# Histogram of transformed male literacy rate
hist(newdistrictdata$male_lit^3,
     main=expression("Hist. of Male Literacy Rate"^3),
     xlab=expression("(Male Literacy Rate)"^3),
```

```
     col="cornflowerblue")

# Q-Q plot for transformed male literacy rate
qqnorm(newdistrictdata$male_lit^3,
       main=expression("QQ-Plot of Male Literacy Rate"^3),
       col="darkblue")
qqline(newdistrictdata$male_lit^3,col="firebrick",lwd=3)
```



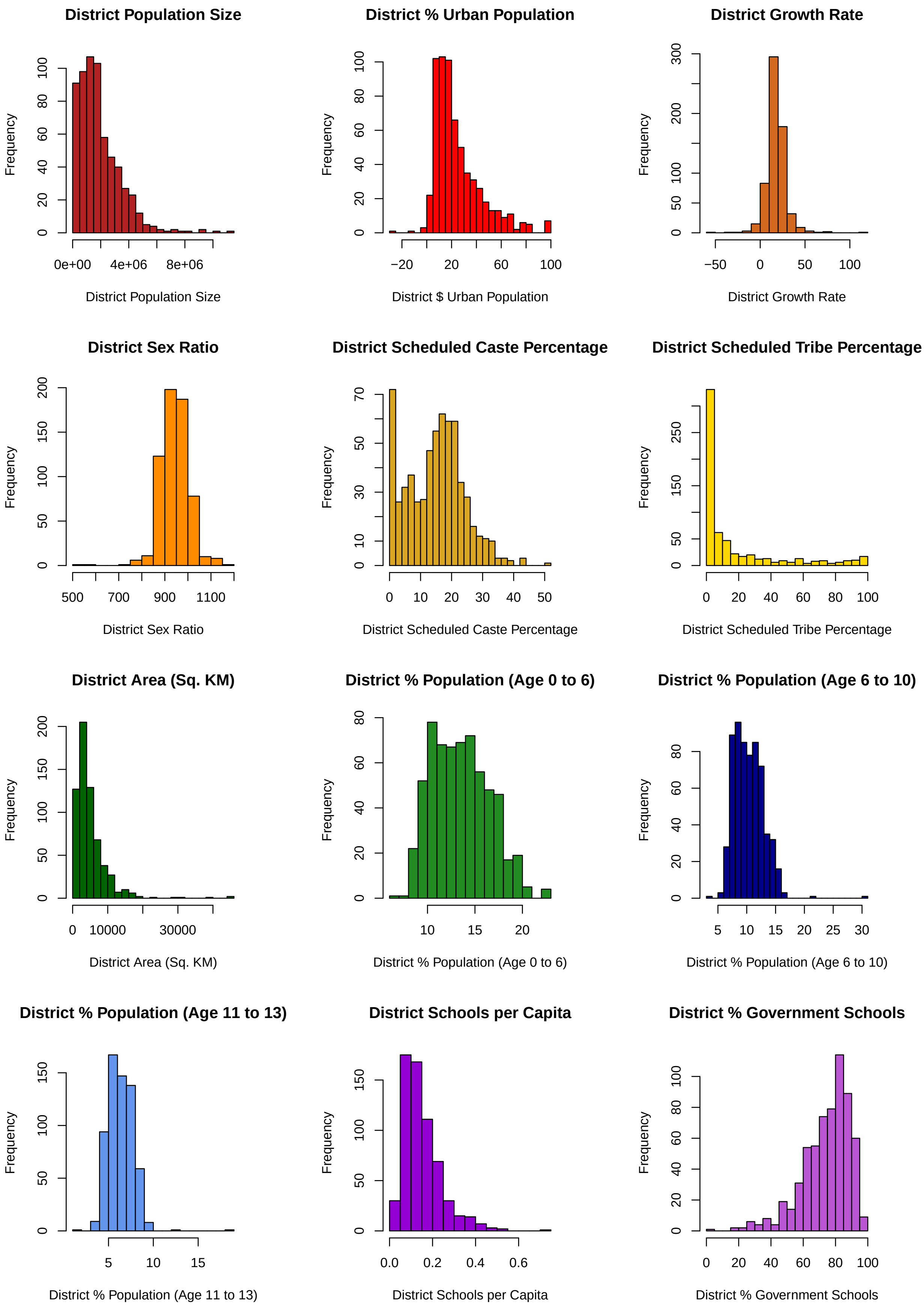


Predictor Variable Transformations

We began by plotting histograms of each of the predictor variables (at the district level). These included all of the filtered variables except for the three response variables (overall literacy rate, female literacy rate, and male literacy rate) and district code, state code, and district name.

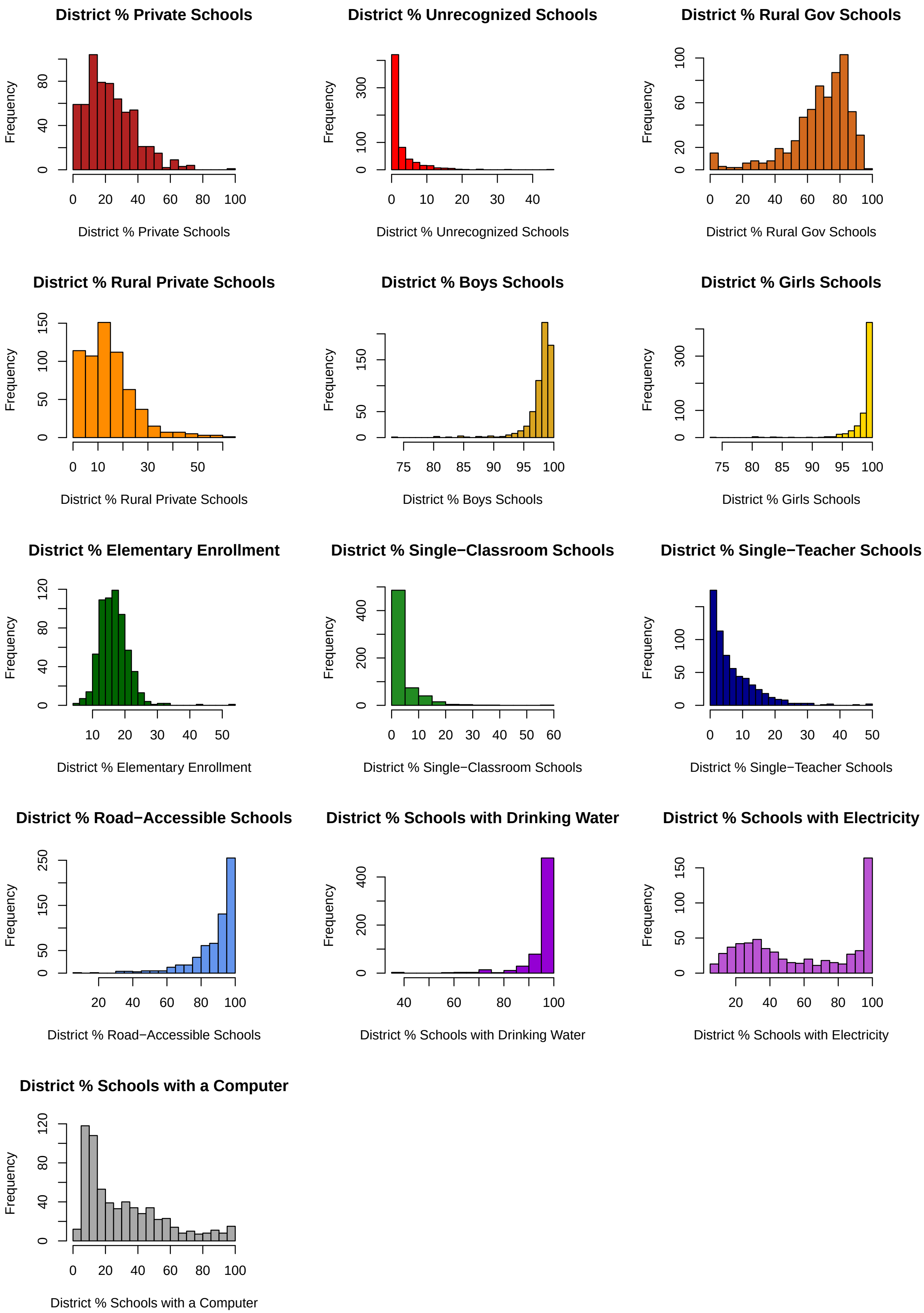
```
# Histograms of un-transformed predictors
par(mfrow = c(4, 3),mai=c(0.75,0.75,0.75,0.75), cex = .85)

hist(newdistrictdata$total_pop,
     main = 'District Population Size',
     xlab = 'District Population Size',
     breaks = 20, col = 'firebrick')
hist(newdistrictdata$p_urban_pop,
     main = 'District % Urban Population',
     xlab = 'District $ Urban Population',
     breaks = 20, col = 'red')
hist(newdistrictdata$growth_rate,
     main = 'District Growth Rate',
     xlab = 'District Growth Rate',
     breaks = 20, col = 'chocolate')
hist(newdistrictdata$sex_ratio,
     main = 'District Sex Ratio',
     xlab = 'District Sex Ratio',
```

```r
     breaks = 20, col = 'darkorange')
hist(newdistrictdata$p_sched_castes,
     main = 'District Scheduled Caste Percentage',
     xlab = 'District Scheduled Caste Percentage',
     breaks = 20, col = 'goldenrod')
hist(newdistrictdata$p_sched_tribes,
     main = 'District Scheduled Tribe Percentage',
     xlab = 'District Scheduled Tribe Percentage',
     breaks = 20, col = 'gold')
hist(newdistrictdata$area_sqkm,
     main = 'District Area (Sq. KM)',
     xlab = 'District Area (Sq. KM)',
     breaks = 20, col = 'darkgreen')
hist(newdistrictdata$p_pop0to6,
     main = 'District % Population (Age 0 to 6)',
     xlab = 'District % Population (Age 0 to 6)',
     breaks = 20, col = 'forestgreen')
hist(newdistrictdata$p_pop6to10,
     main = 'District % Population (Age 6 to 10)',
     xlab = 'District % Population (Age 6 to 10)',
     breaks = 20, col = 'darkblue')
hist(newdistrictdata$p_pop11to13,
     main = 'District % Population (Age 11 to 13)',
     xlab = 'District % Population (Age 11 to 13)',
     breaks = 20, col = 'cornflowerblue')
hist(newdistrictdata$p_capita_schools,
     main = 'District Schools per Capita',
     xlab = 'District Schools per Capita',
     breaks = 20, col = 'darkviolet')
hist(newdistrictdata$p_gov_school,
     main = 'District % Government Schools',
     xlab = 'District % Government Schools',
     breaks = 20, col = 'mediumorchid')
```


District Population Size
Frequency
District Population Size
District % Urban Population
Frequency
District $ Urban Population
District Growth Rate
Frequency
District Growth Rate
District Sex Ratio
Frequency
District Sex Ratio
District Scheduled Caste Percentage
Frequency
District Scheduled Caste Percentage
District Scheduled Tribe Percentage
Frequency
District Scheduled Tribe Percentage
District Area (Sq. KM)
Frequency
District Area (Sq. KM)
District % Population (Age 0 to 6)
Frequency
District % Population (Age 0 to 6)
District % Population (Age 6 to 10)
Frequency
District % Population (Age 6 to 10)
District % Population (Age 11 to 13)
Frequency
District % Population (Age 11 to 13)
District Schools per Capita
Frequency
District Schools per Capita
District % Government Schools
Frequency
District % Government Schools

```r
# Histograms of un-transformed predictors
par(mfrow = c(5, 3),mai=c(0.75,0.75,0.75,0.75), cex = .85)

hist(newdistrictdata$p_priv_school,
     main = 'District % Private Schools',
     xlab = 'District % Private Schools',
     breaks = 20, col = 'firebrick')
hist(newdistrictdata$p_unrec,
     main = 'District % Unrecognized Schools',
     xlab = 'District % Unrecognized Schools',
     breaks = 20, col = 'red')
hist(newdistrictdata$p_gov_rur,
     main = 'District % Rural Gov Schools',
     xlab = 'District % Rural Gov Schools',
     breaks = 20, col = 'chocolate')
hist(newdistrictdata$p_priv_rur,
     main = 'District % Rural Private Schools',
     xlab = 'District % Rural Private Schools',
     breaks = 20, col = 'darkorange')
hist(newdistrictdata$p_boy_school,
     main = 'District % Boys Schools',
     xlab = 'District % Boys Schools',
     breaks = 20, col = 'goldenrod')
hist(newdistrictdata$p_girl_school,
     main = 'District % Girls Schools',
     xlab = 'District % Girls Schools',
     breaks = 20, col = 'gold')
hist(newdistrictdata$p_elementary_enrollment,
     main = 'District % Elementary Enrollment',
     xlab = 'District % Elementary Enrollment',
     breaks = 20, col = 'darkgreen')
hist(newdistrictdata$p_single_class,
     main = 'District % Single-Classroom Schools',
     xlab = 'District % Single-Classroom Schools',
     breaks = 20, col = 'forestgreen')
hist(newdistrictdata$p_single_teacher,
     main = 'District % Single-Teacher Schools',
     xlab = 'District % Single-Teacher Schools',
     breaks = 20, col = 'darkblue')
hist(newdistrictdata$p_road_accessible,
     main = 'District % Road-Accessible Schools',
     xlab = 'District % Road-Accessible Schools',
     breaks = 20, col = 'cornflowerblue')
hist(newdistrictdata$p_drink_water,
     main = 'District % Schools with Drinking Water',
     xlab = 'District % Schools with Drinking Water',
     breaks = 20, col = 'darkviolet')
```

```r
hist(newdistrictdata$p_electricity,
     main = 'District % Schools with Electricity',
     xlab = 'District % Schools with Electricity',
     breaks = 20, col = 'mediumorchid')
hist(newdistrictdata$p_computer,
     main = 'District % Schools with a Computer',
     xlab = 'District % Schools with a Computer',
     breaks = 20, col = 'darkgrey')
```


District % Private Schools
Frequency
District % Private Schools
District % Unrecognized Schools
Frequency
District % Unrecognized Schools
District % Rural Gov Schools
Frequency
District % Rural Gov Schools
District % Rural Private Schools
Frequency
District % Rural Private Schools
District % Boys Schools
Frequency
District % Boys Schools
District % Girls Schools
Frequency
District % Girls Schools
District % Elementary Enrollment
Frequency
District % Elementary Enrollment
District % Single−Classroom Schools
Frequency
District % Single−Classroom Schools
District % Single−Teacher Schools
Frequency
District % Single−Teacher Schools
District % Road−Accessible Schools
Frequency
District % Road−Accessible Schools
District % Schools with Drinking Water
Frequency
District % Schools with Drinking Water
District % Schools with Electricity
Frequency
District % Schools with Electricity
District % Schools with a Computer
Frequency
District % Schools with a Computer

Based on the above plots, we removed `p_boy_school` and `p_girl_school` as predictors because they do not sum to 1 at a district level. We then transformed the remaining variables, as follows, to ensure that they were more normally distributed.

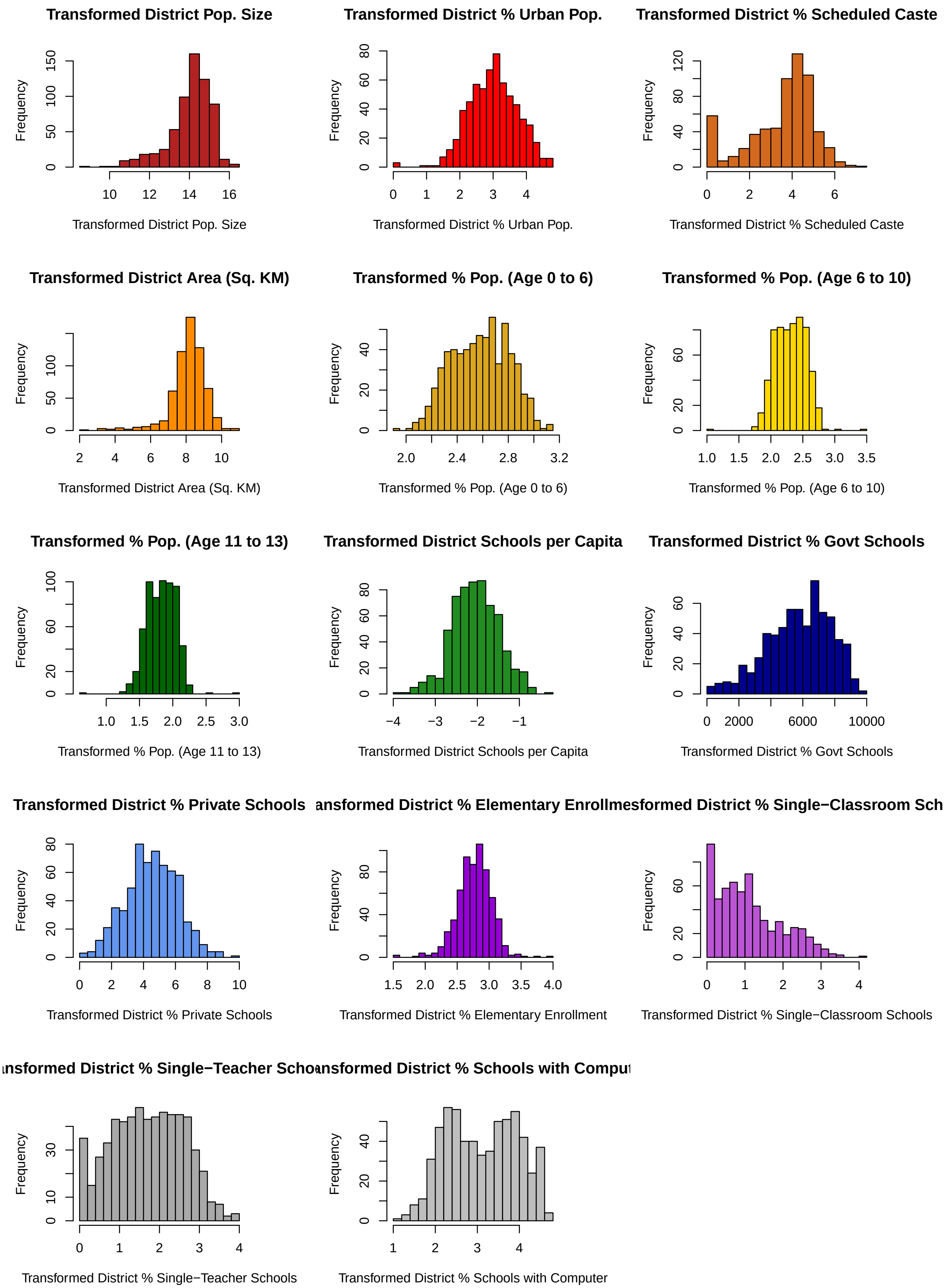
```
# Initialize new data frame with transformed predictors
districtdata.transform = newdistrictdata

# Transform variables depending on skew
# Higher-order (square/cube) transformations for left-skewed data
# Lower-order (log/sqrt) transformations for right-skewed data
districtdata.transform$total_pop =
  log(newdistrictdata$total_pop)
districtdata.transform$p_urban_pop =
  log(abs(newdistrictdata$p_urban_pop)+1)
districtdata.transform$p_sched_castes =
  sqrt(newdistrictdata$p_sched_castes)
districtdata.transform$area_sqkm =
  log(newdistrictdata$area_sqkm)
districtdata.transform$p_pop0to6 =
  log(newdistrictdata$p_pop0to6)
districtdata.transform$p_pop6to10 =
  log(newdistrictdata$p_pop6to10)
districtdata.transform$p_pop11to13 =
  log(newdistrictdata$p_pop11to13)
districtdata.transform$p_capita_schools =
  log(newdistrictdata$p_capita_schools)
districtdata.transform$p_gov_school =
  newdistrictdata$p_gov_school^2
districtdata.transform$p_priv_school =
  sqrt(newdistrictdata$p_priv_school)
districtdata.transform$p_elementary_enrollment =
  log(newdistrictdata$p_elementary_enrollment)
districtdata.transform$p_single_class =
  log(newdistrictdata$p_single_class+1)
districtdata.transform$p_single_teacher =
  log(newdistrictdata$p_single_teacher+1)
districtdata.transform$p_computer =
  log(newdistrictdata$p_computer)
```

Once the variables were transformed, we plotted their histograms to confirm that they were more normally distributed.

```
# Histograms of transformed predictors
par(mfrow = c(5, 3),mai=c(0.75,0.75,0.75,0.75), cex = .8)

hist(districtdata.transform$total_pop,
     main = 'Transformed District Pop. Size',
     xlab = 'Transformed District Pop. Size',
     breaks = 20, col = 'firebrick')
```

```r
hist(districtdata.transform$p_urban_pop,
     main = 'Transformed District % Urban Pop.',
     xlab = 'Transformed District % Urban Pop.',
     breaks = 20, col = 'red')
hist(districtdata.transform$p_sched_castes,
     main = 'Transformed District % Scheduled Caste',
     xlab = 'Transformed District % Scheduled Caste',
     breaks = 20, col = 'chocolate')
hist(districtdata.transform$area_sqkm,
     main = 'Transformed District Area (Sq. KM)',
     xlab = 'Transformed District Area (Sq. KM)',
     breaks = 20, col = 'darkorange')
hist(districtdata.transform$p_pop0to6,
     main = 'Transformed % Pop. (Age 0 to 6)',
     xlab = 'Transformed % Pop. (Age 0 to 6)',
     breaks = 20, col = 'goldenrod')
hist(districtdata.transform$p_pop6to10,
     main = 'Transformed % Pop. (Age 6 to 10)',
     xlab = 'Transformed % Pop. (Age 6 to 10)',
     breaks = 20, col = 'gold')
hist(districtdata.transform$p_pop11to13,
     main = 'Transformed % Pop. (Age 11 to 13)',
     xlab = 'Transformed % Pop. (Age 11 to 13)',
     breaks = 20, col = 'darkgreen')
hist(districtdata.transform$p_capita_schools,
     main = 'Transformed District Schools per Capita',
     xlab = 'Transformed District Schools per Capita',
     breaks = 20, col = 'forestgreen')
hist(districtdata.transform$p_gov_school,
     main = 'Transformed District % Govt Schools',
     xlab = 'Transformed District % Govt Schools',
     breaks = 20, col = 'darkblue')
hist(districtdata.transform$p_priv_school,
     main = 'Transformed District % Private Schools',
     xlab = 'Transformed District % Private Schools',
     breaks = 20, col = 'cornflowerblue')
hist(districtdata.transform$p_elementary_enrollment,
     main = 'Transformed District % Elementary Enrollment',
     xlab = 'Transformed District % Elementary Enrollment',
     breaks = 20, col = 'darkviolet')
hist(districtdata.transform$p_single_class,
     main = 'Transformed District % Single-Classroom Schools',
     xlab = 'Transformed District % Single-Classroom Schools',
     breaks = 20, col = 'mediumorchid')
hist(districtdata.transform$p_single_teacher,
     main = 'Transformed District % Single-Teacher Schools',
     xlab = 'Transformed District % Single-Teacher Schools',
```

```
     breaks = 20, col = 'darkgrey')
hist(districtdata.transform$p_computer,
     main = 'Transformed District % Schools with Computer',
     xlab = 'Transformed District % Schools with Computer',
     breaks = 20, col = 'grey')
```


Transformed District Pop. Size
Frequency
Transformed District Pop. Size
Transformed District % Urban Pop.
Frequency
Transformed District % Urban Pop.
Transformed District % Scheduled Caste
Frequency
Transformed District % Scheduled Caste
Transformed District Area (Sq. KM)
Frequency
Transformed District Area (Sq. KM)
Transformed % Pop. (Age 0 to 6)
Frequency
Transformed % Pop. (Age 0 to 6)
Transformed % Pop. (Age 6 to 10)
Frequency
Transformed % Pop. (Age 6 to 10)
Transformed % Pop. (Age 11 to 13)
Frequency
Transformed % Pop. (Age 11 to 13)
Transformed District Schools per Capita
Frequency
Transformed District Schools per Capita
Transformed District % Govt Schools
Frequency
Transformed District % Govt Schools
Transformed District % Private Schools
Frequency
Transformed District % Private Schools
ansformed District % Elementary Enrollme
Frequency
Transformed District % Elementary Enrollment
sformed District % Single−Classroom Sch
Frequency
Transformed District % Single−Classroom Schools
nsformed District % Single−Teacher Scho
Frequency
Transformed District % Single−Teacher Schools
ansformed District % Schools with Comput
Frequency
Transformed District % Schools with Computer

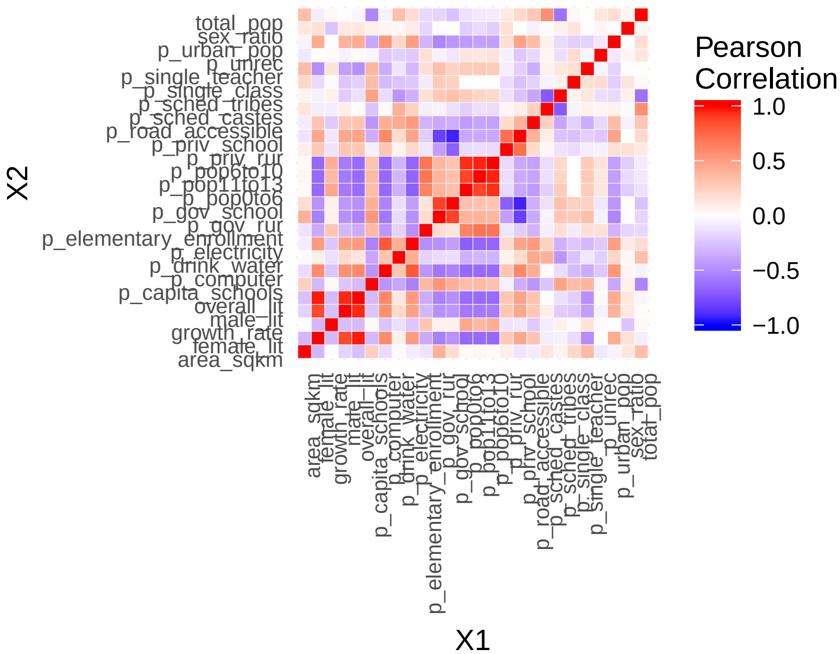
Accordingly, we generated our final dataset with the appropriately transformed predictor variables.

```
# Define final variable names
final_vars <- c("dist_name", "dist_code", "state_code",
                "overall_lit", "female_lit", "male_lit",
                "total_pop", "p_urban_pop",
                "growth_rate",
                "sex_ratio", "p_sched_castes",
                "p_sched_tribes", "area_sqkm",
                "p_pop0to6", "p_pop6to10", "p_pop11to13",
                "p_capita_schools",
                "p_gov_school", "p_priv_school",
                "p_unrec", "p_gov_rur",
                "p_priv_rur", "p_elementary_enrollment",
                "p_single_class", "p_single_teacher",
                "p_road_accessible", "p_drink_water",
                "p_electricity", "p_computer")

# Generate final data set
finaldistrictdata <- districtdata.transform[final_vars]
```

Correlation Matrix of Predictors

In order to conduct a preliminary investigation of predictor variables that were associated, we generated the following correlation heatmap using the transformed dataset.

```
library(ggplot2)
library(reshape)
cormat <- round(cor(finaldistrictdata[,4:29]),5)
melted_cormat <- melt(cormat)
ggplot(data=melted_cormat, aes(X1,X2,fill=value))+
 geom_tile(color="white")+
 scale_fill_gradient2(low="blue",high="red",mid="white",
                      midpoint=0,limit=c(-1,1),space="Lab",
                      name="Pearson\nCorrelation")+
  theme_minimal()+
  theme(axis.text.x=element_text(angle=90,vjust=1,
                                 size=8,hjust=1),
        axis.text.y=element_text(vjust=1,
                                 size=8,hjust=1))+
  coord_fixed()
```



ANOVA of Literacy Rate by State

To continue our preliminary investigation of the data, we conducted an ANOVA test on overall literacy rates in India, with the data segregated by state. We did this to determine whether the state a district was in should be incorporated as a potentially useful predictor in subsequent regression analyses.

```
state_model = aov(overall_lit~state_code,data=finaldistrictdata)
anova(state_model)
```

```
## Analysis of Variance Table
##
## Response: overall_lit
##              Df Sum Sq Mean Sq F value    Pr(>F)
## state_code    1   2698 2697.94  27.456 2.202e-07 ***
## Residuals   623  61218   98.26
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As indicated by the p-value above, the result of the ANOVA test was strongly statistically significant.

Regression Analysis

Finalizing Dataset

In order to begin our regression analysis, we further modified our dataset to only include the overall literacy rate response variable (not male and female literacy rates) and also added the `state_code` predictor as a factor variable.

```r
finaldistrictdata$state_code = factor(finaldistrictdata$state_code)
overall_vars <- c("dist_name", "dist_code", "state_code",
                "overall_lit",
                "total_pop", "p_urban_pop",
                "growth_rate",
                "sex_ratio", "p_sched_castes",
                "p_sched_tribes", "area_sqkm",
                "p_pop0to6", "p_pop6to10", "p_pop11to13",
                "p_capita_schools",
                "p_gov_school", "p_priv_school",
                "p_unrec", "p_gov_rur",
                "p_priv_rur", "p_elementary_enrollment",
                "p_single_class", "p_single_teacher",
                "p_road_accessible", "p_drink_water",
                "p_electricity", "p_computer")

overall_data = finaldistrictdata[overall_vars]
```

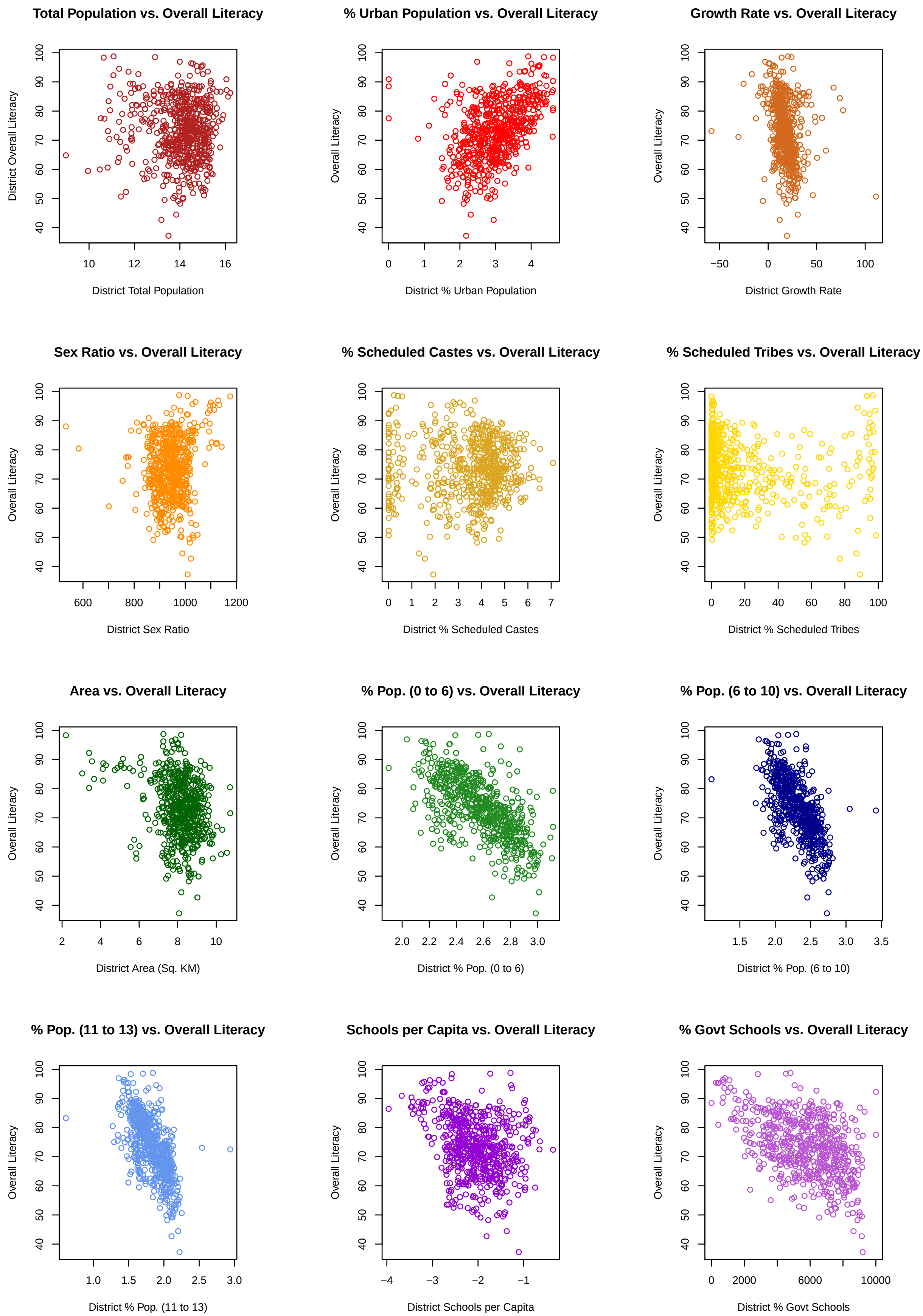
Checking Linearity of Predictors

The next step before generating our regression models was to check the linearity of each of our predictors with the `overall_lit` response variable.

```r
# Plots of predictors vs. overall literacy response
par(mfrow = c(4, 3),mai=c(0.75,0.75,0.75,0.75), cex = .7)

plot(overall_data$total_pop,
     overall_data$overall_lit,
     main = 'Total Population vs. Overall Literacy',
     xlab = 'District Total Population',
     ylab = 'District Overall Literacy',
     col = 'firebrick')

plot(overall_data$p_urban_pop,
     overall_data$overall_lit,
     main = '% Urban Population vs. Overall Literacy',
     xlab = 'District % Urban Population',
     ylab = 'Overall Literacy',
     col = 'red')

plot(overall_data$growth_rate,
     overall_data$overall_lit,
     main = 'Growth Rate vs. Overall Literacy',
     xlab = 'District Growth Rate',
     ylab = 'Overall Literacy',
     col = 'chocolate')
```

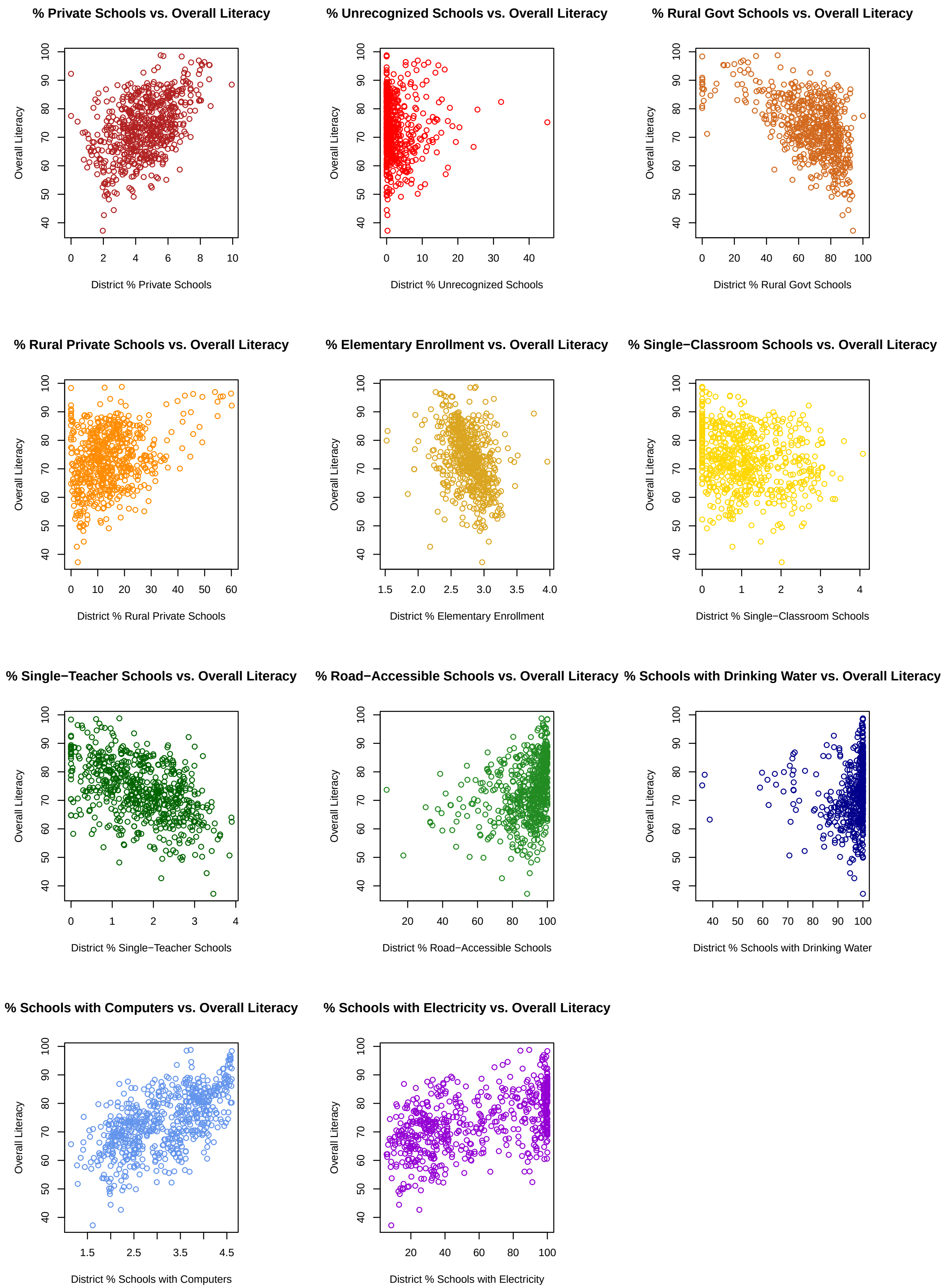
```
plot(overall_data$sex_ratio,
     overall_data$overall_lit,
     main = 'Sex Ratio vs. Overall Literacy',
     xlab = 'District Sex Ratio',
     ylab = 'Overall Literacy',
     col = 'darkorange')

plot(overall_data$p_sched_castes,
     overall_data$overall_lit,
     main = '% Scheduled Castes vs. Overall Literacy',
     xlab = 'District % Scheduled Castes',
     ylab = 'Overall Literacy',
     col = 'goldenrod')

plot(overall_data$p_sched_tribes,
     overall_data$overall_lit,
     main = '% Scheduled Tribes vs. Overall Literacy',
     xlab = 'District % Scheduled Tribes',
     ylab = 'Overall Literacy',
     col = 'gold')

plot(overall_data$area_sqkm,
     overall_data$overall_lit,
     main = 'Area vs. Overall Literacy',
     xlab = 'District Area (Sq. KM)',
     ylab = 'Overall Literacy',
     col = 'darkgreen')

plot(overall_data$p_pop0to6,
     overall_data$overall_lit,
     main = '% Pop. (0 to 6) vs. Overall Literacy',
     xlab = 'District % Pop. (0 to 6)',
     ylab = 'Overall Literacy',
     col = 'forestgreen')

plot(overall_data$p_pop6to10,
     overall_data$overall_lit,
     main = '% Pop. (6 to 10) vs. Overall Literacy',
     xlab = 'District % Pop. (6 to 10)',
     ylab = 'Overall Literacy',
     col = 'darkblue')

plot(overall_data$p_pop11to13,
     overall_data$overall_lit,
     main = '% Pop. (11 to 13) vs. Overall Literacy',
     xlab = 'District % Pop. (11 to 13)',
     ylab = 'Overall Literacy',
```

```
     col = 'cornflowerblue')

plot(overall_data$p_capita_schools,
     overall_data$overall_lit,
     main = 'Schools per Capita vs. Overall Literacy',
     xlab = 'District Schools per Capita',
     ylab = 'Overall Literacy',
     col = 'darkviolet')

plot(overall_data$p_gov_school,
     overall_data$overall_lit,
     main = '% Govt Schools vs. Overall Literacy',
     xlab = 'District % Govt Schools',
     ylab = 'Overall Literacy',
     col = 'mediumorchid')
```


Total Population vs. Overall Literacy
District Overall Literacy
District Total Population
% Urban Population vs. Overall Literacy
Overall Literacy
District % Urban Population
Growth Rate vs. Overall Literacy
Overall Literacy
District Growth Rate
Sex Ratio vs. Overall Literacy
Overall Literacy
District Sex Ratio
% Scheduled Castes vs. Overall Literacy
Overall Literacy
District % Scheduled Castes
% Scheduled Tribes vs. Overall Literacy
Overall Literacy
District % Scheduled Tribes
Area vs. Overall Literacy
Overall Literacy
District Area (Sq. KM)
% Pop. (0 to 6) vs. Overall Literacy
Overall Literacy
District % Pop. (0 to 6)
% Pop. (6 to 10) vs. Overall Literacy
Overall Literacy
District % Pop. (6 to 10)
% Pop. (11 to 13) vs. Overall Literacy
Overall Literacy
District % Pop. (11 to 13)
Schools per Capita vs. Overall Literacy
Overall Literacy
District Schools per Capita
% Govt Schools vs. Overall Literacy
Overall Literacy
District % Govt Schools

```r
# Plots of predictors vs. overall literacy response
par(mfrow = c(4, 3),mai=c(0.75,0.75,0.75,0.75), cex = .7)

plot(overall_data$p_priv_school,
     overall_data$overall_lit,
     main = '% Private Schools vs. Overall Literacy',
     xlab = 'District % Private Schools',
     ylab = 'Overall Literacy',
     col = 'firebrick')

plot(overall_data$p_unrec,
     overall_data$overall_lit,
     main = '% Unrecognized Schools vs. Overall Literacy',
     xlab = 'District % Unrecognized Schools',
     ylab = 'Overall Literacy',
     col = 'red')

plot(overall_data$p_gov_rur,
     overall_data$overall_lit,
     main = '% Rural Govt Schools vs. Overall Literacy',
     xlab = 'District % Rural Govt Schools',
     ylab = 'Overall Literacy',
     col = 'chocolate')

plot(overall_data$p_priv_rur,
     overall_data$overall_lit,
     main = '% Rural Private Schools vs. Overall Literacy',
     xlab = 'District % Rural Private Schools',
     ylab = 'Overall Literacy',
     col = 'darkorange')

plot(overall_data$p_elementary_enrollment,
     overall_data$overall_lit,
     main = '% Elementary Enrollment vs. Overall Literacy',
     xlab = 'District % Elementary Enrollment',
     ylab = 'Overall Literacy',
     col = 'goldenrod')

plot(overall_data$p_single_class,
     overall_data$overall_lit,
     main = '% Single-Classroom Schools vs. Overall Literacy',
     xlab = 'District % Single-Classroom Schools',
     ylab = 'Overall Literacy',
     col = 'gold')

plot(overall_data$p_single_teacher,
     overall_data$overall_lit,
```

```
     main = '% Single-Teacher Schools vs. Overall Literacy',
     xlab = 'District % Single-Teacher Schools',
     ylab = 'Overall Literacy',
     col = 'darkgreen')

plot(overall_data$p_road_accessible,
     overall_data$overall_lit,
     main = '% Road-Accessible Schools vs. Overall Literacy',
     xlab = 'District % Road-Accessible Schools',
     ylab = 'Overall Literacy',
     col = 'forestgreen')

plot(overall_data$p_drink_water,
     overall_data$overall_lit,
     main = '% Schools with Drinking Water vs. Overall Literacy',
     xlab = 'District % Schools with Drinking Water',
     ylab = 'Overall Literacy',
     col = 'darkblue')

plot(overall_data$p_computer,
     overall_data$overall_lit,
     main = '% Schools with Computers vs. Overall Literacy',
     xlab = 'District % Schools with Computers',
     ylab = 'Overall Literacy',
     col = 'cornflowerblue')

plot(overall_data$p_electricity,
     overall_data$overall_lit,
     main = '% Schools with Electricity vs. Overall Literacy',
     xlab = 'District % Schools with Electricity',
     ylab = 'Overall Literacy',
     col = 'darkviolet')
```



% Private Schools vs. Overall Literacy
Overall Literacy
District % Private Schools
0 2 4 6 8 10
40 50 60 70 80 90 100
% Unrecognized Schools vs. Overall Literacy
Overall Literacy
District % Unrecognized Schools
0 10 20 30 40
% Rural Govt Schools vs. Overall Literacy
Overall Literacy
District % Rural Govt Schools
0 20 40 60 80 100
% Rural Private Schools vs. Overall Literacy
Overall Literacy
District % Rural Private Schools
0 10 20 30 40 50 60
% Elementary Enrollment vs. Overall Literacy
Overall Literacy
District % Elementary Enrollment
1.5 2.0 2.5 3.0 3.5 4.0
% Single−Classroom Schools vs. Overall Literacy
Overall Literacy
District % Single−Classroom Schools
0 1 2 3 4
% Single−Teacher Schools vs. Overall Literacy
Overall Literacy
District % Single−Teacher Schools
0 1 2 3 4
% Road−Accessible Schools vs. Overall Literacy
Overall Literacy
District % Road−Accessible Schools
20 40 60 80 100
% Schools with Drinking Water vs. Overall Literacy
Overall Literacy
District % Schools with Drinking Water
40 50 60 70 80 90 100
% Schools with Computers vs. Overall Literacy
Overall Literacy
District % Schools with Computers
1.5 2.5 3.5 4.5
% Schools with Electricity vs. Overall Literacy
Overall Literacy
District % Schools with Electricity
20 40 60 80 100

Based on these scatterplots, we identified predictors that may have quadratic relationships with the `overall_lit` response variable. These included `area_sqkm`, `p_gov_rur`, `p_priv_rur`.

Regression Models

<u>**Model 1:**</u> Our first regression model consisted of the main effects of all predictor terms, except for the `state_code` factor variable.

```r
# Generate multiple regression Model 1
model1 = lm(overall_lit ~ ., overall_data[,4:27])
summary(model1)
```

```
##
## Call:
## lm(formula = overall_lit ~ ., data = overall_data[, 4:27])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.1911  -3.5614   0.4822   4.1313  19.3855
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.597e+02  1.469e+01  10.868  < 2e-16 ***
## total_pop                8.004e-01  5.261e-01   1.521 0.128674
## p_urban_pop              1.436e+00  4.495e-01   3.195 0.001474 **
## growth_rate             -2.527e-02  2.589e-02  -0.976 0.329337
## sex_ratio                1.656e-03  4.673e-03   0.354 0.723187
## p_sched_castes          -3.546e-01  2.618e-01  -1.355 0.176029
## p_sched_tribes          -8.432e-03  1.634e-02  -0.516 0.605962
## area_sqkm               -1.298e+00  4.052e-01  -3.204 0.001429 **
## p_pop0to6               -1.483e+01  3.732e+00  -3.973 7.97e-05 ***
## p_pop6to10               5.782e+00  6.106e+00   0.947 0.344070
## p_pop11to13             -1.965e+01  5.021e+00  -3.914 0.000101 ***
## p_capita_schools         3.522e+00  9.746e-01   3.613 0.000328 ***
## p_gov_school            -3.594e-04  1.020e-03  -0.352 0.724727
## p_priv_school           -1.313e+00  1.165e+00  -1.127 0.260336
## p_unrec                 -2.414e-02  1.597e-01  -0.151 0.879932
## p_gov_rur               -9.249e-02  4.631e-02  -1.997 0.046247 *
## p_priv_rur               1.836e-01  4.811e-02   3.816 0.000150 ***
## p_elementary_enrollment  2.671e+00  1.488e+00   1.795 0.073157 .
## p_single_class          -1.742e+00  3.650e-01  -4.773 2.29e-06 ***
## p_single_teacher        -2.759e+00  3.320e-01  -8.310 6.40e-16 ***
## p_road_accessible       -3.718e-02  2.488e-02  -1.495 0.135566
## p_drink_water           -1.342e-01  4.143e-02  -3.238 0.001270 **
## p_electricity           -5.426e-03  1.633e-02  -0.332 0.739785
## p_computer               1.390e+00  7.358e-01   1.889 0.059362 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 6.048 on 601 degrees of freedom
## Multiple R-squared:  0.656,  Adjusted R-squared:  0.6428
## F-statistic: 49.83 on 23 and 601 DF,  p-value: < 2.2e-16
```

Model 2: Our second regression model consisted of the main effects of all predictor terms, including the `state_code` factor variable.

```
# Generate multiple regression Model 2
model2 = lm(overall_lit ~ ., overall_data[,3:27])
summary(model2)
```

```
##
## Call:
## lm(formula = overall_lit ~ ., data = overall_data[, 3:27])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.8649  -2.7450   0.1825   3.0541  14.4897
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.705e+02  1.494e+01  11.409  < 2e-16 ***
## state_code2             3.158e+00  2.444e+00   1.292 0.196862
## state_code3            -7.065e+00  2.385e+00  -2.963 0.003177 **
## state_code4             5.948e+00  5.498e+00   1.082 0.279799
## state_code5             1.627e+00  2.184e+00   0.745 0.456821
## state_code6            -1.075e+00  2.214e+00  -0.486 0.627382
## state_code7             1.864e+00  3.059e+00   0.609 0.542501
## state_code8            -4.830e+00  1.831e+00  -2.638 0.008579 **
## state_code9             1.949e-01  1.880e+00   0.104 0.917468
## state_code10            6.143e+00  2.435e+00   2.523 0.011912 *
## state_code11           -2.259e-01  3.122e+00  -0.072 0.942353
## state_code12            3.778e+00  2.330e+00   1.621 0.105512
## state_code13            1.365e+01  2.642e+00   5.164 3.35e-07 ***
## state_code14            6.051e+00  2.324e+00   2.604 0.009465 **
## state_code15            2.313e+01  2.658e+00   8.704  < 2e-16 ***
## state_code16            1.954e+01  3.113e+00   6.276 6.92e-10 ***
## state_code17            9.706e+00  2.653e+00   3.659 0.000277 ***
## state_code18            5.619e+00  1.834e+00   3.064 0.002289 **
## state_code19           -6.139e-01  2.337e+00  -0.263 0.792911
## state_code20            3.252e+00  2.290e+00   1.420 0.156120
## state_code21            1.927e+00  1.925e+00   1.001 0.317109
## state_code22            1.801e+00  2.187e+00   0.823 0.410580
## state_code23            3.299e+00  1.682e+00   1.961 0.050387 .
## state_code24            2.759e+00  2.514e+00   1.098 0.272850
## state_code25            7.015e+00  4.133e+00   1.697 0.090209 .
## state_code26            9.801e+00  5.233e+00   1.873 0.061596 .
## state_code27            1.059e+00  2.093e+00   0.506 0.613020
```

```
## state_code28           -1.417e+01  2.526e+00  -5.610 3.16e-08 ***
## state_code29           -6.758e+00  2.287e+00  -2.954 0.003265 **
## state_code30            1.602e+00  4.013e+00   0.399 0.689816
## state_code31            1.429e+01  6.089e+00   2.347 0.019249 *
## state_code32            4.424e+00  3.185e+00   1.389 0.165397
## state_code33           -9.734e+00  2.363e+00  -4.119 4.37e-05 ***
## state_code34            9.292e-01  3.501e+00   0.265 0.790799
## state_code35            8.570e+00  3.715e+00   2.307 0.021439 *
## state_code36           -1.477e+01  2.590e+00  -5.705 1.88e-08 ***
## total_pop               2.064e+00  5.845e-01   3.532 0.000447 ***
## p_urban_pop             1.120e+00  4.002e-01   2.799 0.005301 **
## growth_rate             8.180e-03  2.422e-02   0.338 0.735722
## sex_ratio               6.878e-03  5.052e-03   1.361 0.173913
## p_sched_castes          7.053e-02  3.053e-01   0.231 0.817367
## p_sched_tribes         -6.351e-02  1.674e-02  -3.793 0.000165 ***
## area_sqkm              -1.309e+00  4.338e-01  -3.017 0.002670 **
## p_pop0to6              -2.385e+01  3.252e+00  -7.335 7.75e-13 ***
## p_pop6to10              5.938e+00  5.530e+00   1.074 0.283390
## p_pop11to13            -1.670e+01  4.879e+00  -3.423 0.000665 ***
## p_capita_schools        5.471e+00  1.365e+00   4.007 6.96e-05 ***
## p_gov_school           -2.806e-03  9.864e-04  -2.845 0.004609 **
## p_priv_school          -2.372e+00  1.087e+00  -2.183 0.029452 *
## p_unrec                -3.851e-01  1.531e-01  -2.515 0.012179 *
## p_gov_rur               2.670e-02  4.725e-02   0.565 0.572296
## p_priv_rur              3.743e-02  5.613e-02   0.667 0.505157
## p_elementary_enrollment  1.437e+00  1.517e+00   0.948 0.343719
## p_single_class         -1.306e+00  4.784e-01  -2.729 0.006551 **
## p_single_teacher       -6.516e-01  3.829e-01  -1.702 0.089376 .
## p_road_accessible      -4.215e-02  2.588e-02  -1.628 0.104010
## p_drink_water          -1.844e-01  3.897e-02  -4.731 2.83e-06 ***
## p_electricity           4.512e-02  2.330e-02   1.936 0.053358 .
## p_computer              2.197e+00  8.342e-01   2.633 0.008688 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.625 on 566 degrees of freedom
## Multiple R-squared:  0.8106, Adjusted R-squared:  0.7912
## F-statistic: 41.77 on 58 and 566 DF,  p-value: < 2.2e-16
```

To determine if the second model had more explanatory power than the first model, we performed an Extra Sum-of-Squares (ESS) F-test, as follows. We also determined the Bayes Information Criterion (BIC) for both models.

```
# ESS F-test
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: overall_lit ~ total_pop + p_urban_pop + growth_rate + sex_ratio +
```

```
##     p_sched_castes + p_sched_tribes + area_sqkm + p_pop0to6 +
##     p_pop6to10 + p_pop11to13 + p_capita_schools + p_gov_school +
##     p_priv_school + p_unrec + p_gov_rur + p_priv_rur + p_elementary_enrollment +
##     p_single_class + p_single_teacher + p_road_accessible + p_drink_water +
##     p_electricity + p_computer
## Model 2: overall_lit ~ state_code + total_pop + p_urban_pop + growth_rate +
##     sex_ratio + p_sched_castes + p_sched_tribes + area_sqkm +
##     p_pop0to6 + p_pop6to10 + p_pop11to13 + p_capita_schools +
##     p_gov_school + p_priv_school + p_unrec + p_gov_rur + p_priv_rur +
##     p_elementary_enrollment + p_single_class + p_single_teacher +
##     p_road_accessible + p_drink_water + p_electricity + p_computer
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1    601 21987
## 2    566 12106 35    9881.1 13.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Calculate BIC for each model
n = dim(finaldistrictdata)[1]
extractAIC(model1, k = log(n))[2]
```

```
## [1] 2379.782
```

```r
extractAIC(model2, k = log(n))[2]
```

```
## [1] 2232.12
```

Model 3: In our third regression model, we performed a step-wise sequential selection, starting with Model 2. In our sequential selection method, we set the intercept-only model as the "lower bound" and a model with certain interaction/quadratic terms as the "upper bound". The quadratic terms were determined from our prior linearity analysis and the interaction terms were determined based on what intuitively made sense.

The interaction terms that we included were all predictors interacting with `state_code` factors, among others.

```r
library("MASS")
# Define upper and lower bounds
model_lower = lm(overall_lit ~ 1, overall_data[,3:27])
model_upper = lm(overall_lit ~ .+state_code:.+
                   total_pop:growth_rate+
                   p_urban_pop:growth_rate+
                   p_urban_pop:sex_ratio+
                   p_urban_pop:p_sched_castes+
                   p_urban_pop:p_sched_tribes+
                   p_urban_pop:p_gov_rur+
                   p_urban_pop:p_priv_rur+
                   p_urban_pop:p_single_class+
                   p_urban_pop:p_single_teacher+
                   p_urban_pop:p_road_accessible+
                   p_urban_pop:p_computer+
```

```r
                   growth_rate:p_single_class+
                   growth_rate:p_single_teacher+
                   p_sched_tribes:p_gov_rur+
                   p_sched_tribes:p_priv_rur+
                   p_sched_tribes:p_gov_school+
                   p_sched_tribes:p_priv_school+
                   area_sqkm:p_road_accessible+
                   p_capita_schools:p_single_teacher+
                   p_capita_schools:p_gov_school+
                   p_capita_schools:p_priv_school+
                   p_capita_schools:p_unrec+
                   p_capita_schools:p_gov_rur+
                   p_capita_schools:p_priv_rur+
                   p_capita_schools:p_single_class+
                   p_capita_schools:p_single_teacher+
                   p_capita_schools:p_road_accessible+
                   p_capita_schools:p_drink_water+
                   p_capita_schools:p_electricity+
                   p_capita_schools:p_computer+
                   area_sqkm^2+p_gov_rur^2+p_priv_rur^2,
                 overall_data[,3:27])

# Step-wise selection of useful features
model3=stepAIC(model2,
               scope=list(lower=model_lower,
                          upper = model_upper),
               direction="both", k=log(n),trace=FALSE)
summary(model3)
```

```
##
## Call:
## lm(formula = overall_lit ~ state_code + total_pop + p_urban_pop +
##     p_sched_tribes + area_sqkm + p_pop0to6 + p_pop11to13 + p_capita_schools +
##     p_gov_school + p_priv_school + p_unrec + p_single_class +
##     p_drink_water + p_computer + p_sched_tribes:p_priv_school,
##     data = overall_data[, 3:27])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8950  -2.6980   0.4525   2.9726  17.3773
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.927e+02  1.371e+01  14.055  < 2e-16 ***
## state_code2                 6.857e+00  1.911e+00   3.588 0.000362 ***
## state_code3                -3.762e+00  1.764e+00  -2.133 0.033361 *
## state_code4                 9.305e+00  5.078e+00   1.832 0.067425 .
```

```
## state_code5                     4.372e+00  1.810e+00   2.415 0.016045 *
## state_code6                     2.612e+00  1.695e+00   1.541 0.123830
## state_code7                     4.446e+00  2.455e+00   1.811 0.070609 .
## state_code8                    -3.134e+00  1.512e+00  -2.073 0.038574 *
## state_code9                     1.383e+00  1.546e+00   0.895 0.371378
## state_code10                    7.421e+00  1.974e+00   3.759 0.000188 ***
## state_code11                    4.122e-01  2.850e+00   0.145 0.885054
## state_code12                    4.705e+00  2.018e+00   2.331 0.020074 *
## state_code13                    1.101e+01  2.419e+00   4.551 6.52e-06 ***
## state_code14                    4.906e+00  2.105e+00   2.330 0.020146 *
## state_code15                    1.932e+01  2.601e+00   7.429 3.99e-13 ***
## state_code16                    2.359e+01  2.786e+00   8.467  < 2e-16 ***
## state_code17                    2.998e+00  3.027e+00   0.990 0.322418
## state_code18                    7.350e+00  1.609e+00   4.568 6.04e-06 ***
## state_code19                    2.707e+00  1.900e+00   1.425 0.154843
## state_code20                    4.855e+00  1.859e+00   2.612 0.009247 **
## state_code21                    4.321e+00  1.564e+00   2.762 0.005930 **
## state_code22                    5.503e+00  1.762e+00   3.124 0.001876 **
## state_code23                    3.186e+00  1.351e+00   2.359 0.018648 *
## state_code24                    7.661e+00  1.987e+00   3.855 0.000129 ***
## state_code25                    9.552e+00  3.828e+00   2.495 0.012863 *
## state_code26                    1.281e+01  4.844e+00   2.645 0.008394 **
## state_code27                    4.646e+00  1.685e+00   2.758 0.005997 **
## state_code28                   -1.075e+01  2.006e+00  -5.359 1.22e-07 ***
## state_code29                   -2.688e+00  1.618e+00  -1.662 0.097079 .
## state_code30                    4.291e+00  3.539e+00   1.212 0.225884
## state_code31                    2.748e+01  5.921e+00   4.642 4.29e-06 ***
## state_code32                    1.215e+01  2.323e+00   5.230 2.37e-07 ***
## state_code33                   -4.849e+00  1.819e+00  -2.665 0.007917 **
## state_code34                    4.618e+00  2.964e+00   1.558 0.119804
## state_code35                    1.439e+01  3.323e+00   4.329 1.76e-05 ***
## state_code36                   -1.152e+01  2.074e+00  -5.555 4.25e-08 ***
## total_pop                       2.252e+00  5.386e-01   4.181 3.36e-05 ***
## p_urban_pop                     1.383e+00  3.775e-01   3.664 0.000271 ***
## p_sched_tribes                 -2.045e-01  3.019e-02  -6.773 3.13e-11 ***
## area_sqkm                      -1.730e+00  3.702e-01  -4.675 3.67e-06 ***
## p_pop0to6                      -2.067e+01  2.277e+00  -9.078  < 2e-16 ***
## p_pop11to13                    -1.108e+01  2.176e+00  -5.092 4.81e-07 ***
## p_capita_schools                6.962e+00  1.005e+00   6.926 1.16e-11 ***
## p_gov_school                   -4.006e-03  8.215e-04  -4.875 1.41e-06 ***
## p_priv_school                  -4.232e+00  1.091e+00  -3.879 0.000117 ***
## p_unrec                        -6.307e-01  1.485e-01  -4.248 2.51e-05 ***
## p_single_class                 -1.468e+00  4.393e-01  -3.342 0.000885 ***
## p_drink_water                  -1.831e-01  3.680e-02  -4.976 8.58e-07 ***
## p_computer                      2.449e+00  7.770e-01   3.151 0.001709 **
## p_sched_tribes:p_priv_school  3.884e-02  7.823e-03   4.965 9.08e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.551 on 575 degrees of freedom
## Multiple R-squared:  0.8137, Adjusted R-squared:  0.7978
## F-statistic: 51.24 on 49 and 575 DF,  p-value: < 2.2e-16
```

```
# Calculate BIC
extractAIC(model3,k = log(n))[2]
```

```
## [1] 2163.966
```

Model 4: In our fourth regression analysis, we performed a multilevel mixed-effects model, using the predictors from Model 3 (without the `state_code` predictors). Given the distract-state hierarchy inherent to our dataset, we included `state_code` in our model as a "random" factor variable. In particular, we fit our `overall_lit` response data with a varying intercept model for the districts in each state.

```
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:reshape':
##
##     expand
```

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step
```

```
# Genereate regression model
model4 = lmer(overall_lit ~ total_pop + p_urban_pop +
                p_sched_tribes + area_sqkm + p_pop0to6 +
                p_pop11to13 + p_capita_schools +
                p_gov_school + p_priv_school +
                p_unrec + p_single_class +
                p_drink_water + p_computer +
                p_sched_tribes:p_priv_school +
                (1|state_code), data = overall_data)
summary(model4)
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
```

```
## Formula:
## overall_lit ~ total_pop + p_urban_pop + p_sched_tribes + area_sqkm +
##     p_pop0to6 + p_pop11to13 + p_capita_schools + p_gov_school +
##     p_priv_school + p_unrec + p_single_class + p_drink_water +
##     p_computer + p_sched_tribes:p_priv_school + (1 | state_code)
##    Data: overall_data
##
## REML criterion at convergence: 3801.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.2941 -0.6061  0.1015  0.6629  3.5077
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  state_code (Intercept) 47.01    6.856
##  Residual               20.83    4.564
## Number of obs: 625, groups:  state_code, 36
##
## Fixed effects:
##                                Estimate Std. Error         df t value
## (Intercept)                   2.041e+02  1.335e+01  6.100e+02  15.294
## total_pop                     1.859e+00  5.199e-01  6.079e+02   3.575
## p_urban_pop                   1.487e+00  3.717e-01  6.023e+02   4.000
## p_sched_tribes               -1.923e-01  2.873e-02  6.016e+02  -6.694
## area_sqkm                    -1.879e+00  3.576e-01  6.089e+02  -5.254
## p_pop0to6                    -2.033e+01  2.253e+00  5.966e+02  -9.022
## p_pop11to13                  -1.102e+01  2.163e+00  5.890e+02  -5.094
## p_capita_schools              6.187e+00  9.631e-01  6.043e+02   6.424
## p_gov_school                 -4.217e-03  8.074e-04  6.041e+02  -5.223
## p_priv_school                -4.637e+00  1.068e+00  6.070e+02  -4.341
## p_unrec                      -6.545e-01  1.457e-01  6.057e+02  -4.491
## p_single_class               -1.622e+00  4.275e-01  6.100e+02  -3.795
## p_drink_water                -1.839e-01  3.637e-02  5.995e+02  -5.057
## p_computer                    2.500e+00  7.506e-01  6.078e+02   3.331
## p_sched_tribes:p_priv_school  3.799e-02  7.283e-03  5.568e+02   5.216
##                              Pr(>|t|)
## (Intercept)                   < 2e-16 ***
## total_pop                    0.000378 ***
## p_urban_pop                  7.11e-05 ***
## p_sched_tribes               4.97e-11 ***
## area_sqkm                    2.06e-07 ***
## p_pop0to6                     < 2e-16 ***
## p_pop11to13                  4.73e-07 ***
## p_capita_schools             2.69e-10 ***
## p_gov_school                 2.43e-07 ***
## p_priv_school                1.66e-05 ***
## p_unrec                      8.48e-06 ***
```

```
## p_single_class               0.000163 ***
## p_drink_water                5.67e-07 ***
## p_computer                   0.000918 ***
## p_sched_tribes:p_priv_school 2.59e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 15 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

Lasso/Ridge Regressions and Model Comparisons

Our next step was to conduct a cross-validation of Models 1-4, in addition to models incorporating lasso regression (Model 5) and ridge regression (Model 6).

```r
library(glmnet)
```

```
## Loading required package: foreach
## Loaded glmnet 2.0-13
```

```r
library(splitstackshape)
```

```
## Loading required package: data.table
##
## Attaching package: 'data.table'
## The following object is masked from 'package:reshape':
##
##     melt
```

```r
set.seed(12345)
nsims=200

# Define range of lambdas for lasso/ridge regressions
lambdas_lasso = seq(0,.00002,.000001)
lambdas_ridge = seq(0,.02,.001)

# Set-up sum of squared errors vectors
sse1=sse2=sse3=sse4=rep(NA,nsims)
sse_lasso = matrix(NA,nrow=nsims,ncol=length(lambdas_lasso))
sse_ridge = matrix(NA,nrow=nsims,ncol=length(lambdas_ridge))

# Subset data to only contain states with sufficient number of
# districts (need enough data points for train and test sets)
```

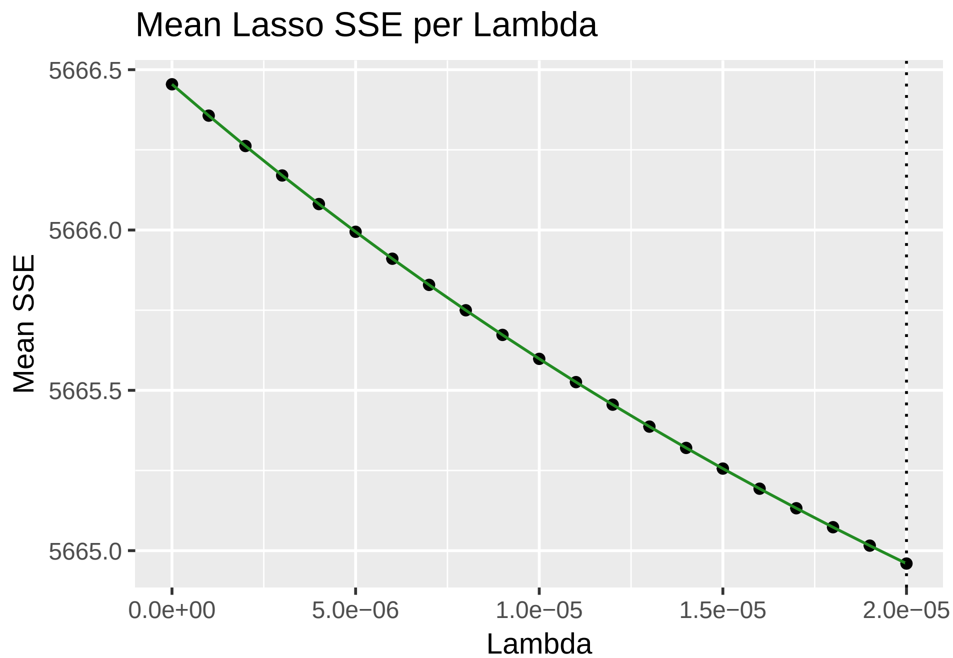
```r
temp_overall_data = subset(overall_data, state_code != 26 &
                             state_code != 31 & state_code != 4)
unique_codes = unique(temp_overall_data$state_code)
X = model.matrix(model3)
y = overall_data$overall_lit

n.train = 404 # Define number of data points for training set
n = nrow(temp_overall_data)

# Conduct cross-validation of regression models
for(i in 1:nsims){
  reorder=sample(n)

  # Split data into train and test sets
  train = stratified(temp_overall_data, 'state_code',
                   select =
                     list('state_code' =
                            c(unique_codes)),
                   size = .65)
  test = temp_overall_data[!duplicated(
    rbind(train,
          temp_overall_data))[
            -seq_len(nrow(train))], ]

  # Fit1: Result of Model 1
  fit1 = lm(formula(model1),data=train)
  # Fit2: Result of Model 2
  fit2 = lm(formula(model2),data=train)
  # Fit3: Result of Model 3
  fit3 = lm(formula(model3),data=train)
  # Fit4: result of Model 4
  fit4=lmer(formula(model4),data=train)

  # Calculate SSEs for Models 1-4
  sse1[i]=sum((test$overall_lit-predict(fit1,new=test))^2)
  sse2[i]=sum((test$overall_lit-predict(fit2,new=test))^2)
  sse3[i]=sum((test$overall_lit-predict(fit3,new=test))^2)
  sse4[i]=sum((test$overall_lit-predict(fit4,newdata=test))^2)

  # Re-order train and test sets
  X_train=X[reorder[1:n.train],]
  y_train=y[reorder[1:n.train]]
  X_test=X[reorder[n.train:n],]
  y_test=y[reorder[n.train:n]]

  # Calculate lasso and ridge regressions
  lassos = glmnet(X_train,y_train, alpha = 1, lambda = lambdas_lasso)
```

```r
  ridges = glmnet(X_train,y_train, alpha = 0, lambda = lambdas_ridge)

  # Calculate yhats for test set to get SSEs in test set
  yhat_test_lassos = predict(lassos,newx=X_test)
  yhat_test_ridges = predict(ridges,newx=X_test)
  sse_lasso[i,]=apply((y_test-yhat_test_lassos)^2,2,sum)
  sse_ridge[i,]=apply((y_test-yhat_test_ridges)^2,2,sum)

  if(i%%100==0){cat("Finished Iteration #",i,"\n")}
}
```

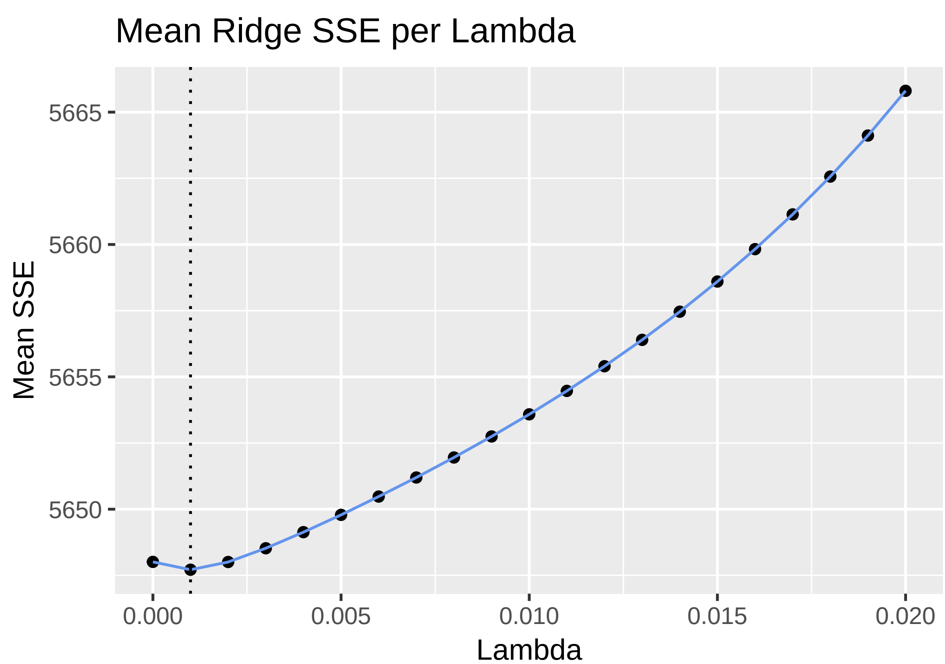
```
## Finished Iteration # 100
## Finished Iteration # 200
```

We then performed an analysis to determine which lambda values resulted in the lowest mean lasso and ridge regression sum of squared errors (SSEs).

```r
# Calculate mean lasso and ridge SSEs per lambda value
mean_lassos = apply(sse_lasso, 2, mean)
mean_ridges = apply(sse_ridge, 2, mean)

lasso_df = data.frame(t(rbind(lambdas_lasso, mean_lassos)))
ridge_df = data.frame(t(rbind(lambdas_ridge, mean_ridges)))

# Determine lambda value with lowest mean lasso/ridge SSE
best_lasso = lambdas_lasso[which.min(mean_lassos)]
best_ridge = lambdas_ridge[which.min(mean_ridges)]

# Plot mean lasso and ridge SSEs per lambda
ggplot(data = lasso_df, aes(x = lambdas_lasso,
                            y = mean_lassos)) +
  geom_point() + geom_line(color = 'forestgreen') +
  ggtitle('Mean Lasso SSE per Lambda') +
  xlab('Lambda') +
  ylab('Mean SSE') +
  geom_vline(xintercept = best_lasso, linetype = 'dotted')
```

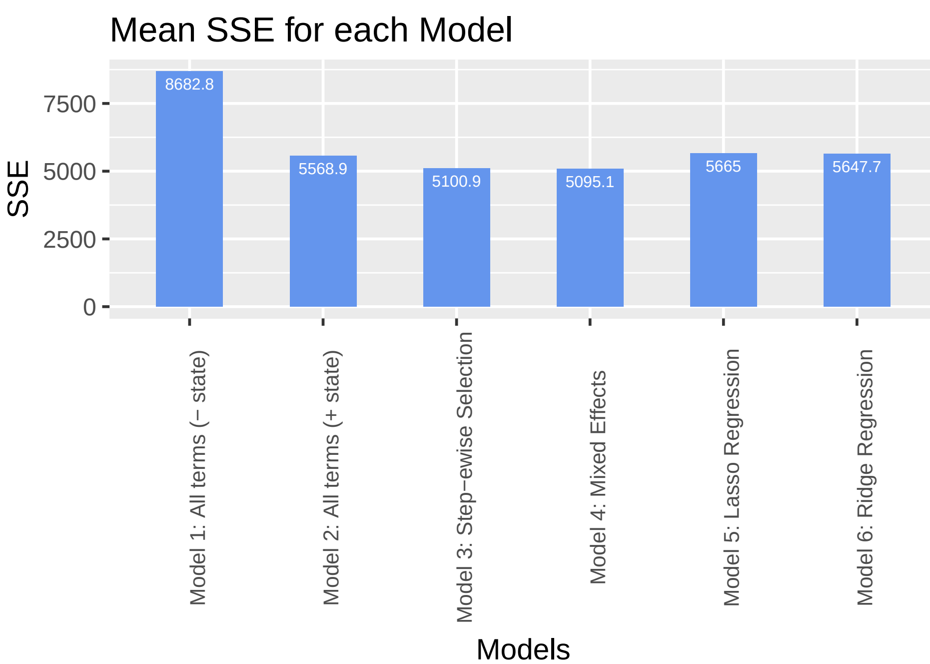



```
ggplot(data = ridge_df, aes(x = lambdas_ridge,
                            y = mean_ridges)) +
  geom_point() + geom_line(color = 'cornflowerblue') +
  ggtitle('Mean Ridge SSE per Lambda') +
  xlab('Lambda') +
  ylab('Mean SSE') +
  geom_vline(xintercept = best_ridge, linetype = 'dotted')
```



Finally, we compared the mean SSEs among models to identify the model with the lowest mean SSE (best predictive model).

```r
# Calculate mean SSE for Models 1-6
errors = c(mean(sse1), mean(sse2), mean(sse3), mean(sse4),
  mean_lassos[which.min(mean_lassos)],
  mean_ridges[which.min(mean_ridges)]
)

error_df <- data.frame(Models = c("Model 1: All terms (- state)",
                                  "Model 2: All terms (+ state)",
                                  "Model 3: Step-ewise Selection",
                                  "Model 4: Mixed Effects",
                                  "Model 5: Lasso Regression",
                                  "Model 6: Ridge Regression"),
                       SSE = errors)

# Plot mean SSE for each model
ggplot(error_df, aes(Models, SSE)) +
  geom_bar(stat="identity",fill='cornflowerblue',width=.5)+
  geom_text(aes(label=round(SSE,1)),vjust=1.6,
            color="white",size=2)+
  theme(axis.text.x = element_text(size=8, angle=90)) +
  ggtitle('Mean SSE for each Model')
```

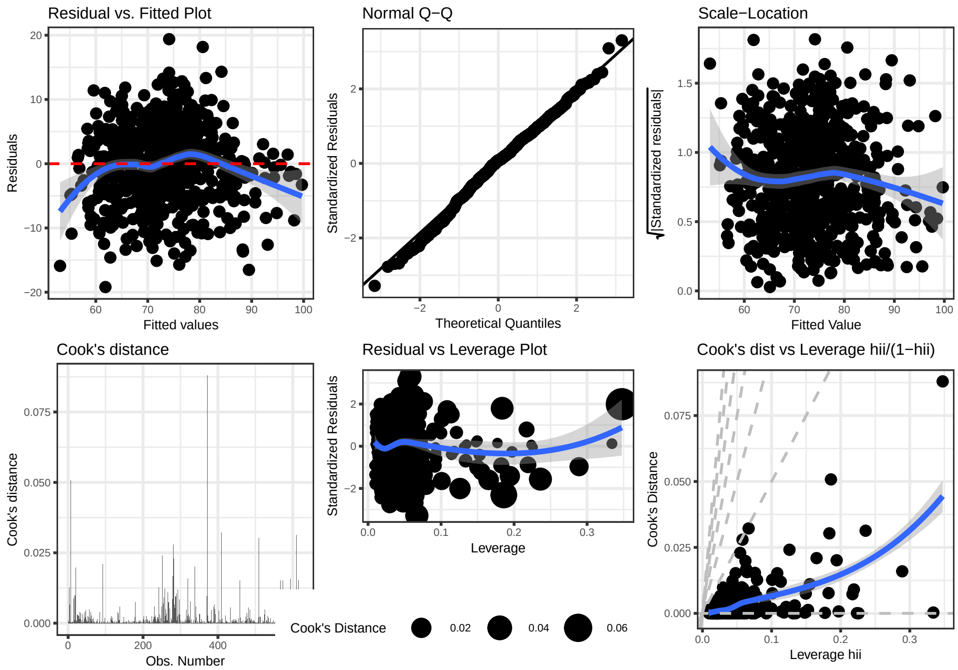


Checking Model Assumptions

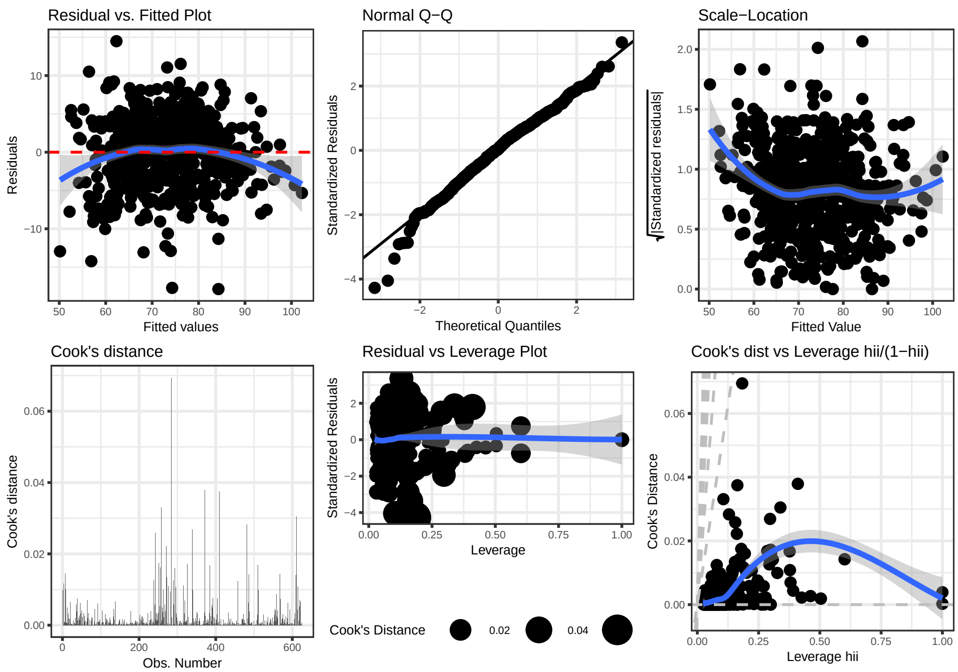
Before proceeding with any conclusions about our model cross-validation, we sought to check the assumptions of our regression models. Recall the the four key assumptions of multiple regression include: independence of errors, constant variance of errors, normality of errors, and linearity. Thus, our diagnostic plots enabled us to check for violations of each of these assumptions, in addition to the presence of outlier/leverage/influential points.

```
library(car)
library(GGally)
library(ggplot2)
library(grid)
library(gridExtra)
library(MASS)
library(reshape)

# Define function to create the following assumption-checking plots
diagPlot<-function(model){

    # Calculate QQ-line
    y = quantile(stdres(model),c(0.25,0.75)) # Find 1st, 3rd quartiles
    # Find the matching normal values on the x-axis
    x = qnorm( c(0.25, 0.75))
    slope = diff(y) / diff(x) # Compute the line slope
    int = y[1] - slope * x[1] # Compute the line intercept

    # Residual vs. Fitted Values Plot
    p1<-ggplot(model, aes(.fitted, .resid))+geom_point()
    p1<-p1+stat_smooth(method="loess")
    p1<-p1+geom_hline(yintercept=0, col="red", linetype="dashed")
    p1<-p1+xlab("Fitted values")+ylab("Residuals")
    p1<-p1+ggtitle("Residual vs. Fitted Plot")+
      theme_bw(base_size = 5)

    # Normal Q-Q Plot
    p2<-ggplot(model, aes(qqnorm(.stdresid)[[1]],
                          .stdresid))+geom_point(na.rm = TRUE)
    p2<-p2+geom_abline(intercept = int, slope = slope)
    p2<-p2+xlab("Theoretical Quantiles")+
      ylab("Standardized Residuals")
    p2<-p2+ggtitle("Normal Q-Q")+
      theme_bw(base_size = 5)

    # Standardized Residuals vs. Fitted Values Plot
    p3<-ggplot(model, aes(.fitted, sqrt(abs(.stdresid))))+
      geom_point(na.rm=TRUE)
    p3<-p3+stat_smooth(method="loess", na.rm = TRUE)+
      xlab("Fitted Value")
    p3<-p3+
      ylab(expression(sqrt("|Standardized residuals|")))
    p3<-p3+ggtitle("Scale-Location")+theme_bw(base_size = 5)

    # Cook's Distance Plot
    p4<-ggplot(model, aes(seq_along(.cooksd), .cooksd))
```

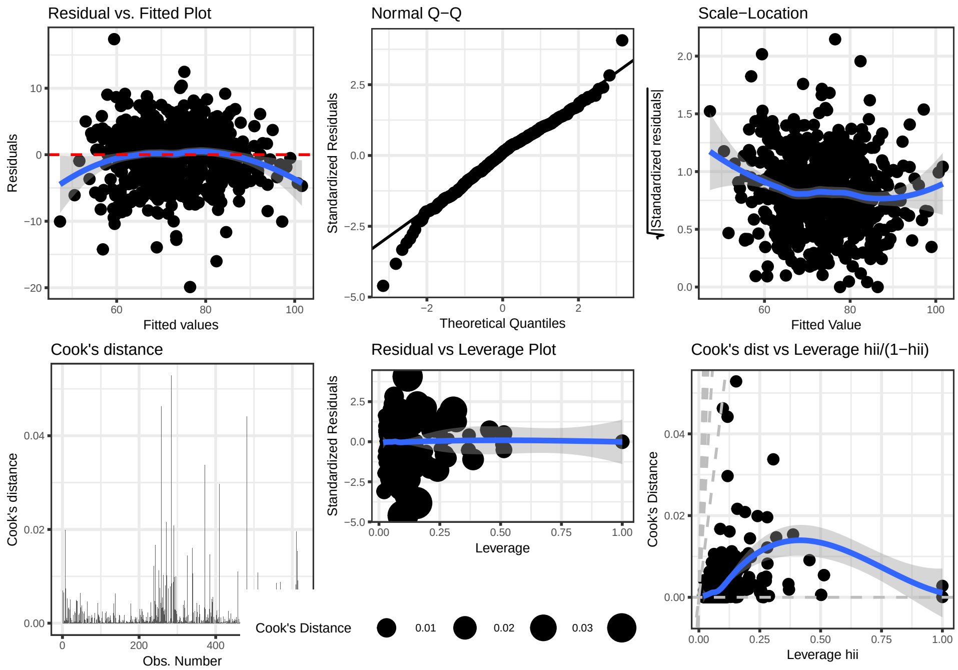
```r
    p4<-p4+geom_bar(stat="identity", position="identity")
    p4<-p4+xlab("Obs. Number")+ylab("Cook's distance")
    p4<-p4+ggtitle("Cook's distance")+theme_bw(base_size = 5)

    # Resdiual vs. Leverage Plot
    p5<-ggplot(model, aes(.hat, .stdresid))
    p5<-p5+geom_point(aes(size=.cooksd), na.rm=TRUE)
    p5<-p5+stat_smooth(method="loess", na.rm=TRUE)
    p5<-p5+xlab("Leverage")+ylab("Standardized Residuals")
    p5<-p5+ggtitle("Residual vs Leverage Plot")
    p5<-p5+scale_size_continuous("Cook's Distance",
                                 range=c(1,5))
    p5<-p5+theme_bw(base_size = 5)+
      theme(legend.position="bottom")

    # Cook's Distance vs. Leverage Plot
    p6<-ggplot(model, aes(.hat, .cooksd))
    p6<-p6+geom_point(na.rm=TRUE)+
      stat_smooth(method="loess", na.rm=TRUE)
    p6<-p6+xlab("Leverage hii")+ylab("Cook's Distance")
    p6<-p6+ggtitle("Cook's dist vs Leverage hii/(1-hii)")
    p6<-p6+geom_abline(slope=seq(0,3,0.5), color="gray",
                       linetype="dashed")
    p6<-p6+theme_bw(base_size = 5)

    return(list(rvfPlot=p1, qqPlot=p2, sclLocPlot=p3,
                cdPlot=p4, rvlevPlot=p5, cvlPlot=p6))
}

# Diagnostic plots for Model 1
diagPlts = diagPlot(model1)
grid.arrange(grobs = diagPlts, nrow=2)
```



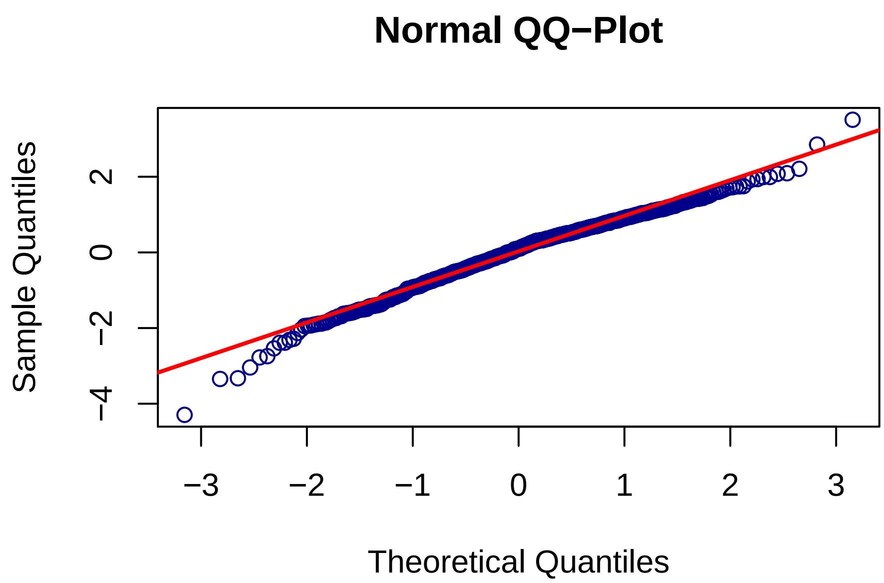

```r
# Diagnostic plots for Model 2
diagPlts = diagPlot(model2)
grid.arrange(grobs = diagPlts, nrow=2)
```



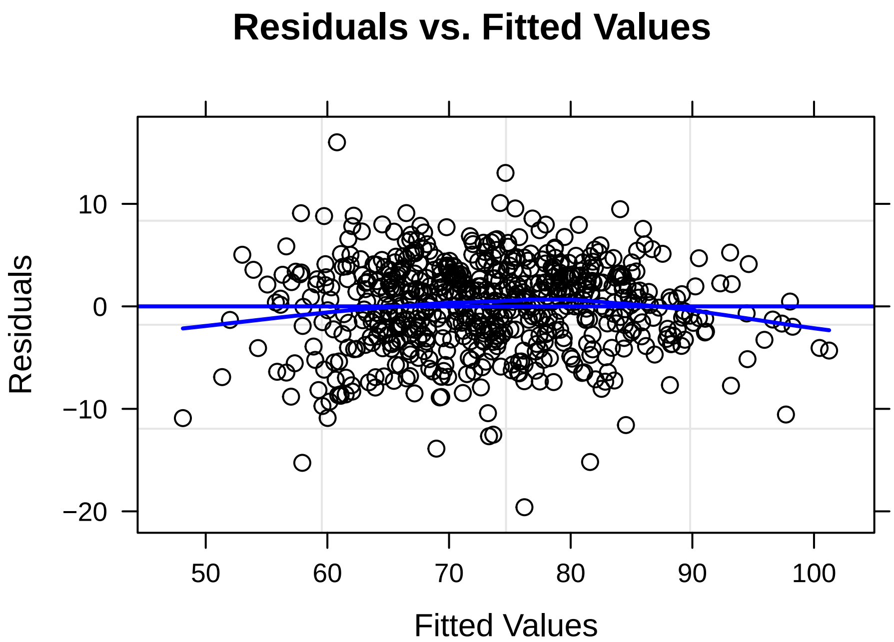

```r
# Diagnostic plots for Model 3
diagPlts = diagPlot(model3)
grid.arrange(grobs = diagPlts, nrow=2)
```




```
# Diagnostic plots for Model 4
qqnorm(summary(model4)$residuals,
       main="Normal QQ-Plot",col="darkblue") # QQ-Plot
qqline(summary(model4)$residuals,col="red",lwd=2)
```




```
# Residuals vs. Fitted Values plot
plot(model4,type=c("p","smooth"),
     col="black",col.line="blue",lwd=2,
     main="Residuals vs. Fitted Values",
     xlab="Fitted Values", ylab="Residuals")
```



Application to Male and Female Literacy Rates

Another question we sought to address was whether the "best" useful set of predictors for the overall literacy rate response was also the "best" useful set of predictors for the male and female literacy rate response variables. Thus, we began by generating the appropriate datasets.

```r
# Generate male literacy response dataset
finaldistrictdata$state_code = factor(finaldistrictdata$state_code)
male_vars <- c("dist_name", "dist_code", "state_code",
                "male_lit",
                "total_pop", "p_urban_pop",
                "growth_rate",
                "sex_ratio", "p_sched_castes",
                "p_sched_tribes", "area_sqkm",
                "p_pop0to6", "p_pop6to10", "p_pop11to13",
                "p_capita_schools",
                "p_gov_school", "p_priv_school",
                "p_unrec", "p_gov_rur",
                "p_priv_rur", "p_elementary_enrollment",
                "p_single_class", "p_single_teacher",
                "p_road_accessible", "p_drink_water",
                "p_electricity", "p_computer")
male_data = finaldistrictdata[male_vars]

# Generate female literacy response dataset
finaldistrictdata$state_code = factor(finaldistrictdata$state_code)
female_vars <- c("dist_name", "dist_code", "state_code",
                "female_lit",
                "total_pop", "p_urban_pop",
```

```
                 "growth_rate",
                 "sex_ratio", "p_sched_castes",
                 "p_sched_tribes", "area_sqkm",
                 "p_pop0to6", "p_pop6to10", "p_pop11to13",
                 "p_capita_schools",
                 "p_gov_school", "p_priv_school",
                 "p_unrec", "p_gov_rur",
                 "p_priv_rur", "p_elementary_enrollment",
                 "p_single_class", "p_single_teacher",
                 "p_road_accessible", "p_drink_water",
                 "p_electricity", "p_computer")
female_data = finaldistrictdata[female_vars]
```

We analyzed the applicability of the overall literacy rate predictors to the male and female literacy rate responses by: 1) fitting mixed effects models with the overall literacy rate predictors to the male and female data and 2) running a step-wise backward selection method to see if any predictors should be eliminated.

```
library(lme4)
library(lmerTest)
# Genereate regression Model 9 for male data
model9 = lmer(male_lit ~ total_pop + p_urban_pop +
                p_sched_tribes + area_sqkm + p_pop0to6 +
                p_pop11to13 + p_capita_schools +
                p_gov_school + p_priv_school +
                p_unrec + p_single_class +
                p_drink_water + p_computer +
                p_sched_tribes:p_priv_school +
                (1|state_code), data = male_data)
# Determine if any predictors are eliminated
step(model9)
```

```
##
## Random effects:
##            Chi.sq Chi.DF elim.num p.value
## state_code 245.66      1     kept < 1e-07
##
## Fixed effects:
##                                 Sum Sq   Mean Sq NumDF  DenDF F.value
## total_pop                     221.1014  221.1014     1 601.13 10.7774
## p_urban_pop                   154.2034  154.2034     1 605.47  7.5165
## p_sched_tribes               1113.8300 1113.8300     1 592.74 54.2926
## area_sqkm                     314.8223  314.8223     1 603.83 15.3457
## p_pop0to6                    1369.2547 1369.2547     1 600.14 66.7431
## p_pop11to13                   204.2876  204.2876     1 592.01  9.9578
## p_capita_schools             1165.9188 1165.9188     1 594.58 56.8316
## p_gov_school                  542.1248  542.1248     1 607.14 26.4254
## p_priv_school                 444.7279  444.7279     1 609.27 21.6778
```

```
## p_unrec                         484.1186  484.1186     1 608.50 23.5979
## p_single_class                  387.0420  387.0420     1 607.20 18.8660
## p_drink_water                   621.7473  621.7473     1 603.34 30.3065
## p_computer                      347.4012  347.4012     1 600.03 16.9338
## p_sched_tribes:p_priv_school  462.3286  462.3286     1 525.71 22.5358
##                               elim.num Pr(>F)
## total_pop                         kept 0.0011
## p_urban_pop                       kept 0.0063
## p_sched_tribes                    kept <1e-07
## area_sqkm                         kept  1e-04
## p_pop0to6                         kept <1e-07
## p_pop11to13                       kept 0.0017
## p_capita_schools                  kept <1e-07
## p_gov_school                      kept  0e+00
## p_priv_school                     kept  0e+00
## p_unrec                           kept  0e+00
## p_single_class                    kept  0e+00
## p_drink_water                     kept <1e-07
## p_computer                        kept  0e+00
## p_sched_tribes:p_priv_school      kept  0e+00
##
## Least squares means:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
##
##  Differences of LSMEANS:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
##
## Final model:
## lme4::lmer(formula = male_lit ~ total_pop + p_urban_pop + p_sched_tribes +
##     area_sqkm + p_pop0to6 + p_pop11to13 + p_capita_schools +
##     p_gov_school + p_priv_school + p_unrec + p_single_class +
##     p_drink_water + p_computer + p_sched_tribes:p_priv_school +
##     (1 | state_code), data = male_data)
```

```r
# Genereate regression Model 10 for female data
model10 = lmer(female_lit ~ total_pop + p_urban_pop +
                p_sched_tribes + area_sqkm + p_pop0to6 +
                p_pop11to13 + p_capita_schools +
                p_gov_school + p_priv_school +
                p_unrec + p_single_class +
                p_drink_water + p_computer +
                p_sched_tribes:p_priv_school +
                (1|state_code), data = female_data)
# Determine if any predictors are eliminated
step(model10)
```

```
##
## Random effects:
```

```
##            Chi.sq Chi.DF elim.num p.value
## state_code 327.51      1     kept < 1e-07
##
## Fixed effects:
##                                Sum Sq   Mean Sq NumDF  DenDF F.value
## total_pop                    423.4428  423.4428     1 609.63 15.9991
## p_urban_pop                  559.3667  559.3667     1 600.42 21.1348
## p_sched_tribes               698.0812  698.0812     1 605.41 26.3759
## area_sqkm                   1148.0272 1148.0272     1 609.91 43.3764
## p_pop0to6                   2553.1952 2553.1952     1 594.89 96.4685
## p_pop11to13                  643.7911  643.7911     1 587.84 24.3246
## p_capita_schools             708.8471  708.8471     1 607.65 26.7827
## p_gov_school                 558.9578  558.9578     1 602.31 21.1193
## p_priv_school                312.1568  312.1568     1 605.36 11.7944
## p_unrec                      337.8552  337.8552     1 603.82 12.7653
## p_single_class               214.5792  214.5792     1 609.65  8.1075
## p_drink_water                377.7068  377.7068     1 597.45 14.2711
## p_computer                   133.9555  133.9555     1 609.65  5.0613
## p_sched_tribes:p_priv_school 641.1879  641.1879     1 572.58 24.2263
##                             elim.num Pr(>F)
## total_pop                        kept  1e-04
## p_urban_pop                      kept  0e+00
## p_sched_tribes                   kept  0e+00
## area_sqkm                        kept <1e-07
## p_pop0to6                        kept <1e-07
## p_pop11to13                      kept  0e+00
## p_capita_schools                 kept  0e+00
## p_gov_school                     kept  0e+00
## p_priv_school                    kept 0.0006
## p_unrec                          kept 0.0004
## p_single_class                   kept 0.0046
## p_drink_water                    kept 0.0002
## p_computer                       kept 0.0248
## p_sched_tribes:p_priv_school     kept  0e+00
##
## Least squares means:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
##
##  Differences of LSMEANS:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
##
## Final model:
## lme4::lmer(formula = female_lit ~ total_pop + p_urban_pop + p_sched_tribes +
##     area_sqkm + p_pop0to6 + p_pop11to13 + p_capita_schools +
##     p_gov_school + p_priv_school + p_unrec + p_single_class +
##     p_drink_water + p_computer + p_sched_tribes:p_priv_school +
##     (1 | state_code), data = female_data)
```