Stat 151a Final Project R Code

```
suppressMessages(library(tidyverse))
suppressMessages(library(usmap))
suppressMessages(library(scales))
suppressMessages(library(mice))
suppressMessages(library(glmnet))
suppressMessages(library(boot))
suppressMessages(library(grid))
suppressMessages(library(gridExtra))
suppressMessages(library(cowplot))
```

Data Pre Processing

```
# Read data in and seprate State and County in to seperate columns, then edit any problema
cancer_raw = read.csv("cancer_reg.csv")
cancer_edit = cancer_raw
cancer_edit <- cancer_edit %>% mutate(Target_div_Income = TARGET_deathRate/medIncome)
cancer_geo = cbind(cancer_edit, str_match(cancer_edit$Geography,"(.+), (.+)")[ ,-1])
colnames(cancer_geo)[37] ="State"
colnames(cancer_geo)[36] = "County"
cancer_geo[167,36] <- "Dona Ana County"
cancer_geo[821,36] <- "La Salle Parish"
codes <- rep(NULL, length(cancer_geo$County))

for (i in 1:length(cancer_geo$avgAnnCount)){
 codes[i] = fips(state = cancer_geo$State[i], county = cancer_geo$County[i])
}
cancer_final = cbind(cancer_geo, fips = codes)
```

Modeling Code

```
moddat <- cancer_final

(colMeans(is.na(moddat)))*100
```

```
            avgAnnCount        avgDeathsPerYear        TARGET_deathRate
               0.000000                0.000000                0.000000
          incidenceRate               medIncome              popEst2015
               0.000000                0.000000                0.000000
        povertyPercent             studyPerCap               binnedInc
               0.000000                0.000000                0.000000
             MedianAge          MedianAgeMale        MedianAgeFemale
               0.000000                0.000000                0.000000
             Geography       AvgHouseholdSize          PercentMarried
               0.000000                0.000000                0.000000
           PctNoHS18_24             PctHS18_24        PctSomeCol18_24
               0.000000                0.000000               74.991795
       PctBachDeg18_24            PctHS25_Over      PctBachDeg25_Over
               0.000000                0.000000                0.000000
    PctEmployed16_Over    PctUnemployed16_Over     PctPrivateCoverage
               4.988513                0.000000                0.000000
PctPrivateCoverageAlone      PctEmpPrivCoverage      PctPublicCoverage
              19.986872                0.000000                0.000000
 PctPublicCoverageAlone               PctWhite               PctBlack
               0.000000                0.000000                0.000000
              PctAsian           PctOtherRace    PctMarriedHouseholds
               0.000000                0.000000                0.000000
             BirthRate      Target_div_Income                  County
               0.000000                0.000000                0.000000
                 State                    fips
               0.000000                0.000000
```

```
# Set reproducability seed and then impute data
set.seed(1)
trim = moddat[,-18]
imp <- mice(trim, m = 5, maxit = 50, meth = "pmm")
```

```
Warning: Number of logged events: 505
```

```r
complete(imp)
```

```r
imputed <- complete(imp)
imputed_new <- imputed
```

```r
mod1 <- lm(data = imputed_new, TARGET_deathRate ~ povertyPercent + PctBlack + PctNoHS18_24
```

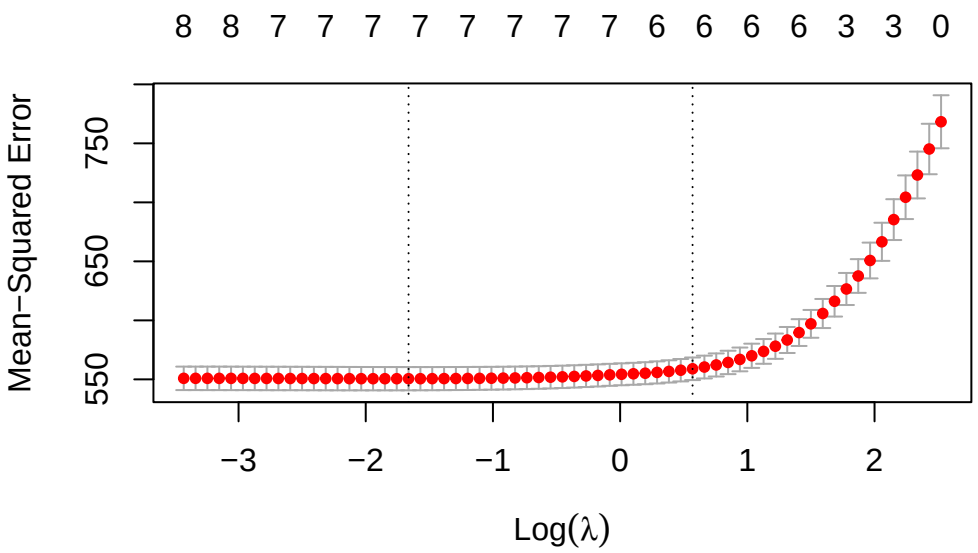
```r
summary(mod1)
```

```
Call:
lm(formula = TARGET_deathRate ~ povertyPercent + PctBlack + PctNoHS18_24 +
    PctHS18_24, data = imputed_new)

Residuals:
     Min       1Q   Median       3Q      Max
-106.595  -13.332    1.245   14.515  164.404

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    127.58734    2.10954  60.481  < 2e-16 ***
povertyPercent   1.66957    0.08312  20.087  < 2e-16 ***
PctBlack         0.13644    0.03527   3.869 0.000112 ***
PctNoHS18_24    -0.17345    0.05673  -3.058 0.002251 **
PctHS18_24       0.70898    0.04883  14.518  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.21 on 3042 degrees of freedom
Multiple R-squared:  0.2402,    Adjusted R-squared:  0.2392
F-statistic: 240.4 on 4 and 3042 DF,  p-value: < 2.2e-16
```

```r
#First, code the Southeast variable for future use
new_england <- c("Connecticut", "Maine", "Massachusetts", "New Hampshire", "Rhode Island",
mideast <- c("Delaware", "District of Columbia", "Maryland", "New Jersey", "New York", "Pe
great_lakes <- c("Illinois", "Indiana", "Michigan", "Ohio", "Wisconsin")
plains <- c("Iowa", "Kansas", "Minnesota", "Missouri", "Nebraska", "North Dakota", "South
southeast <- c("Alabama", "Arkansas", "Florida", "Georgia", "Kentucky", "Louisiana", "Miss
southwest <- c("Arizona", "New Mexico", "Oklahoma", "Texas")
```

```r
rocky_mountain <- c("Colorado", "Idaho", "Montana", "Utah", "Wyoming")
far_west <- c("Alaska", "California", "Hawaii", "Nevada", "Oregon", "Washington")

get_region <- function(state) {
  if (state %in% new_england) {
    return("New England")
  } else if (state %in% mideast) {
    return("Mideast")
  } else if (state %in% great_lakes) {
    return("Great Lakes")
  } else if (state %in% plains) {
    return("Plains")
  } else if (state %in% southeast) {
    return("Southeast")
  } else if (state %in% southwest) {
    return("Southwest")
  } else if (state %in% rocky_mountain) {
    return("Rocky Mountain")
  } else if (state %in% far_west) {
    return("Far West")
  } else {
    return(NA)
  }
}

imputed_new$Region <- sapply(imputed_new$State, get_region)

imputed_new$isSoutheast <- ifelse(imputed_new$Region == "Southeast", "Yes", "No")
```

```r
#Create Lasso Lambda graph
set.seed(1)
y = imputed_new$TARGET_deathRate
x = data.matrix(imputed_new[, c('povertyPercent', 'PctBlack', 'PctHS18_24','PctNoHS18_24',
cv_model <- cv.glmnet(x, y, alpha = 1)
plot(cv_model)
```



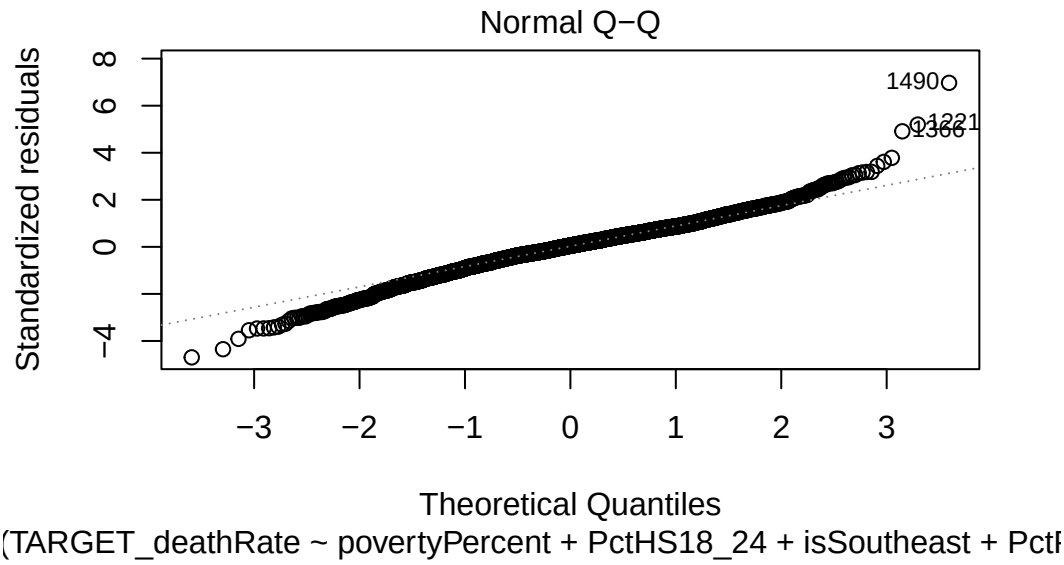

```
#Assign 1se lambda and then run Lasso using it
min_lambda <- cv_model$lambda.min
se_lambda <- cv_model$lambda.1se
best_model <- glmnet(x, y, alpha = 1, lambda = se_lambda)
coef(best_model)
```

```
9 x 1 sparse Matrix of class "dgCMatrix"
                                s0
(Intercept)            124.2552434
povertyPercent           0.4504766
PctBlack                 .
PctHS18_24               0.3404665
PctNoHS18_24             .
isSoutheast              9.5552934
PctPublicCoverage        0.1653959
PctPublicCoverageAlone   0.6901461
PctUnemployed16_Over     0.3542458
```

```
finmod <- lm(data = imputed_new, TARGET_deathRate ~ povertyPercent + PctHS18_24 + isSouthe
summary(finmod)
```

```
Call:
lm(formula = TARGET_deathRate ~ povertyPercent + PctHS18_24 +
    isSoutheast + PctPublicCoverage + PctPublicCoverageAlone +
    PctUnemployed16_Over, data = imputed_new)

Residuals:
     Min       1Q   Median       3Q      Max
-110.107  -13.052    1.293   14.243  163.409

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)             122.3032     2.4841  49.234  < 2e-16 ***
povertyPercent            0.5638     0.1207   4.672 3.12e-06 ***
PctHS18_24                0.4931     0.0497   9.922  < 2e-16 ***
isSoutheastYes           11.6651     1.0157  11.485  < 2e-16 ***
PctPublicCoverage         0.2811     0.1109   2.535  0.01128 *
PctPublicCoverageAlone    0.5704     0.1827   3.123  0.00181 **
PctUnemployed16_Over      0.5630     0.1730   3.253  0.00115 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.46 on 3040 degrees of freedom
Multiple R-squared:  0.2866,    Adjusted R-squared:  0.2852
F-statistic: 203.5 on 6 and 3040 DF,  p-value: < 2.2e-16
```

```r
plot(finmod, which =2)
```

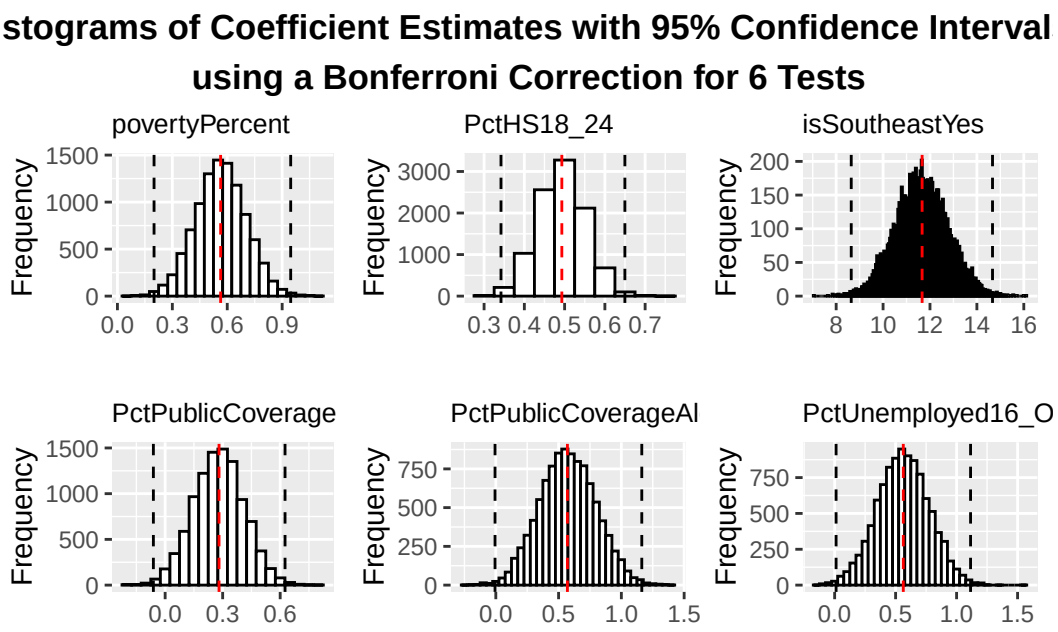



```
shapiro.test(finmod$residuals)
```

```
    Shapiro-Wilk normality test

data:  finmod$residuals
W = 0.98178, p-value < 2.2e-16
```

```
nboot <- 10000
set.seed(1)

# Create a function to calculate the coefficients using the bootstrap
coef.boot <- function(data, indices) {
  model <- lm(TARGET_deathRate ~ povertyPercent + PctHS18_24 + isSoutheast + PctPublicCove
  return(coef(model)[-1]) # exclude intercept column
}

# Perform the bootstrap using the defined function
boot.results <- boot(data = imputed_new, statistic = coef.boot, R = nboot)

# Convert bootstrap results to a data frame
```

```r
boot.df <- as.data.frame(boot.results$t)
colnames(boot.df) <- c("povertyPercent", "PctHS18_24", "isSoutheastYes","PctPublicCoverage

# Get coefficient estimates from original model
finmod <- lm(TARGET_deathRate ~ povertyPercent + PctHS18_24 + isSoutheast + PctPublicCover
coef.estimates <- coef(finmod)[-1 , drop = TRUE]

# Create a function to plot histograms with quantile and coefficient lines and a title
plot.hist <- function(x, coef.est, varname) {
  p <- ggplot(data.frame(x), aes(x = x)) +
    geom_histogram(binwidth = 0.05, color = "black", fill = "white") +
    geom_vline(xintercept = quantile(x, probs = c(0.004166667, 0.9958333)), linetype = "da
    geom_vline(xintercept = coef.est, color = "red", linetype = "dashed") +
    xlab("") + ylab("Frequency") +
    ggtitle(varname)+
    theme(plot.title = element_text(size = 9.5))
  return(p)
}

# Create a list of plots for each column in boot.df with titles
plot.list <- mapply(plot.hist, x = boot.df, coef.est = coef.estimates, varname = names(boo

# Combine the plots into a single figure with a title
grid.arrange(grobs = plot.list, ncol = 3, top = textGrob('Histograms of Coefficient Estima
)
```




```
# Create reduced data set and apply the Southeast column used previously
reduced <- na.omit(trim[,-21])
nrow(reduced) - nrow(imputed_new)
```

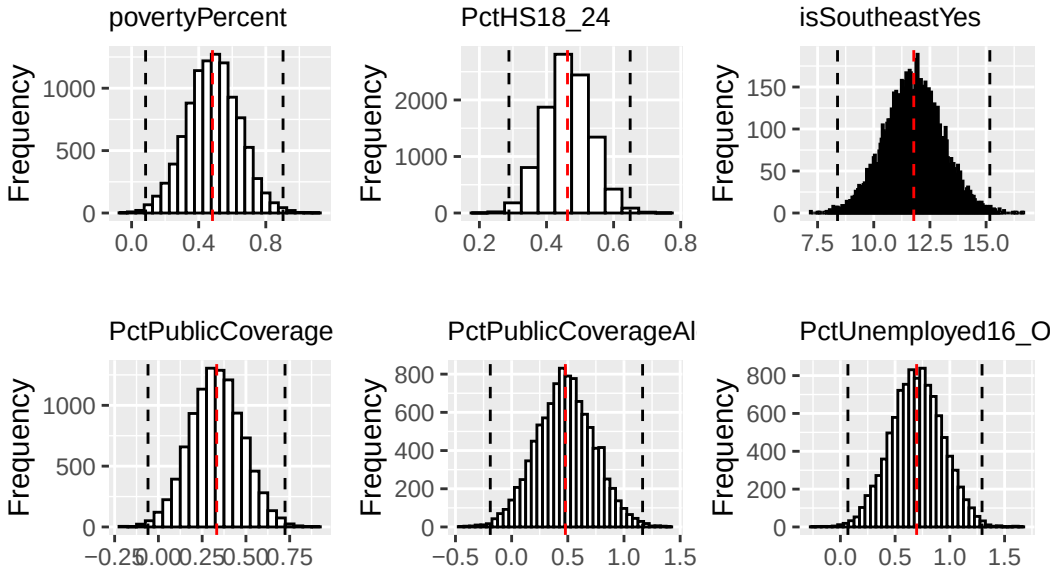
```
[1] -609
```

```
reduced$Region <- sapply(reduced$State, get_region)

reduced$isSoutheast <- ifelse(reduced$Region == "Southeast", "Yes", "No")
```

```
set.seed(1)
##DO NOT INCLUDE IN FINAL
nboot <- 10000

# Create a function to calculate the coefficients using the bootstrap
coef.boot <- function(data, indices) {
  model <- lm(TARGET_deathRate ~ povertyPercent + PctHS18_24 + isSoutheast + PctPublicCove
  return(coef(model)[-1]) # exclude intercept column
}
```

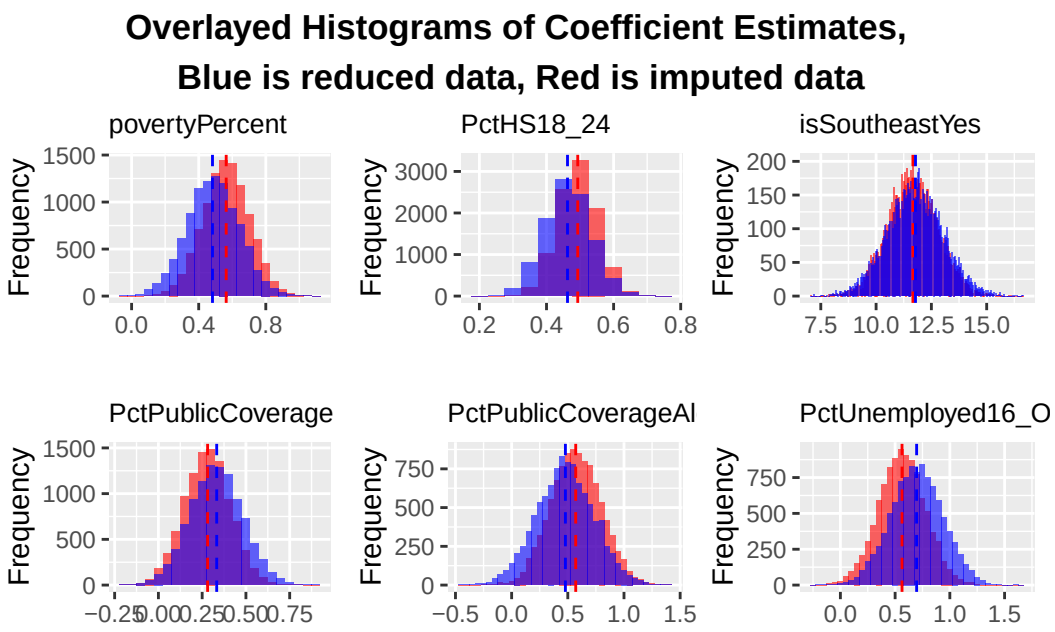
```r
# Perform the bootstrap using the defined function
boot.results.reduce <- boot(data = reduced, statistic = coef.boot, R = nboot)

# Convert bootstrap results to a data frame
boot.df.reduce <- as.data.frame(boot.results.reduce$t)
colnames(boot.df.reduce) <- c("povertyPercent", "PctHS18_24", "isSoutheastYes","PctPublicC

# Get coefficient estimates from original model
finmod.reduce <- lm(TARGET_deathRate ~ povertyPercent + PctHS18_24 + isSoutheast + PctPubl
coef.estimates.reduce <- coef(finmod.reduce)[-1 , drop = TRUE]

# Create a function to plot histograms with quantile and coefficient lines and a title
plot.hist <- function(x, coef.est, varname) {
  p <- ggplot(data.frame(x), aes(x = x)) +
    geom_histogram(binwidth = 0.05, color = "black", fill = "white") +
    geom_vline(xintercept = quantile(x, probs = c(0.004166667, 0.9958333)), linetype = "da
    geom_vline(xintercept = coef.est, color = "red", linetype = "dashed") +
    xlab("") + ylab("Frequency") +
    ggtitle(varname)+
    theme(plot.title = element_text(size = 9.5))
  return(p)
}

# Create a list of plots for each column in boot.df with titles
plot.list <- mapply(plot.hist, x = boot.df.reduce, coef.est = coef.estimates.reduce, varna

# Combine the plots into a single figure with a title
grid.arrange(grobs = plot.list, ncol = 3, top = textGrob('Reduced Data Histograms of Coeff
)
```

Data Histograms of Coefficient Estimates with 95% Confidence




```
# Combine the two bootstrapped data frames
boot.df_combined <- bind_rows(
  boot.df %>% mutate(group = "Original"),
  boot.df.reduce %>% mutate(group = "Reduced")
)

idx = 1:nrow(boot.df_combined)

# Create a function to plot histograms with quantile and coefficient lines and a title
plot.hist <- function(x, coef.est, coef.est.reduce, varname) {
  group <- ifelse(idx <= nrow(boot.df), "Original", "Reduced")
  p <- ggplot(data.frame(x, group = group), aes(x = x)) +
    geom_histogram(binwidth = 0.05, alpha = 0.6, aes(fill = group), position = "identity")
    geom_vline(xintercept = coef.est, color = "red", linetype = "dashed") +
    geom_vline(xintercept = coef.est.reduce, color = "blue", linetype = "dashed") +
    xlab("") + ylab("Frequency") +
    ggtitle(varname) +
    theme(plot.title = element_text(size = 9.5)) +
    scale_fill_manual(values = c("Original" = "red", "Reduced" = "blue")) + theme(legend.p
  return(p)
}
```

```
# Create a list of plots for each column in boot.df with titles
plot.list <- mapply(plot.hist, x = boot.df_combined[,-7], coef.est = coef.estimates, coef.

grid.arrange(grobs = plot.list, ncol = 3, top = textGrob('Overlayed Histograms of Coeffici
```




```
summary(finmod)
```

```
Call:
lm(formula = TARGET_deathRate ~ povertyPercent + PctHS18_24 +
    isSoutheast + PctPublicCoverage + PctPublicCoverageAlone +
    PctUnemployed16_Over, data = imputed_new)

Residuals:
     Min       1Q   Median       3Q      Max
-110.107  -13.052    1.293   14.243  163.409

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           122.3032     2.4841  49.234  < 2e-16 ***
povertyPercent          0.5638     0.1207   4.672 3.12e-06 ***
```

```
PctHS18_24                0.4931     0.0497   9.922  < 2e-16 ***
isSoutheastYes           11.6651     1.0157  11.485  < 2e-16 ***
PctPublicCoverage         0.2811     0.1109   2.535  0.01128 *
PctPublicCoverageAlone    0.5704     0.1827   3.123  0.00181 **
PctUnemployed16_Over      0.5630     0.1730   3.253  0.00115 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.46 on 3040 degrees of freedom
Multiple R-squared:  0.2866,    Adjusted R-squared:  0.2852
F-statistic: 203.5 on 6 and 3040 DF,  p-value: < 2.2e-16
```

```
quantile(boot.df$povertyPercent, probs = c(0.004166667, 0.9958333))
```

```
0.4166667%  99.58333%
 0.2000268  0.9466988
```

```
quantile(boot.df$PctHS18_24, probs = c(0.004166667, 0.9958333))
```

```
0.4166667%  99.58333%
 0.3419795  0.6497332
```

```
quantile(boot.df$isSoutheastYes, probs = c(0.004166667, 0.9958333))
```

```
0.4166667%  99.58333%
  8.642598  14.666479
```

```
quantile(boot.df$PctPublicCoverage, probs = c(0.004166667, 0.9958333))
```

```
 0.4166667%   99.58333%
-0.06139477  0.62534342
```

```
quantile(boot.df$PctPublicCoverageAlone, probs = c(0.004166667, 0.9958333))
```

```
  0.4166667%    99.58333%
-0.006147866  1.162750944
```

```r
quantile(boot.df$PctUnemployed16_Over, probs = c(0.004166667, 0.9958333))
```

```
0.4166667%  99.58333%
0.01195514 1.11447750
```