36-402 Homework 6

James “Morgan” Hawkins

Friday March 3, 2023

Problem 1

```
house = read.csv("/Users/morganhawkins/Downloads/house.csv")
mobility = read.csv("/Users/morganhawkins/Downloads/mobility.csv")
```

Problem 1 (a)

```
house$isPA = as.numeric(house$Longitude >= -90)
```

```
set.seed(999)

#column i shows which rows in data set are in bootstrap sample i
bs.samples = matrix(nrow = 10605,ncol = 200)
for(i in 1:200){
  bs.samples[,i] = as.integer(runif(10605, 1, 10606))
}

#values correspond to index in bootstrap sample NOT in data set
bs.samples.folds = matrix(nrow = 10605, ncol = 200)

for(i in 1:200){
  bs.samples.folds[,i] = sample(1:10605)
}

#prediction.errors.mod.3 = matrix(nrow = 5, ncol = 200)
#prediction.errors.mod.4 = matrix(nrow = 5, ncol = 200)

#reading previously collected results into
prediction.errors.mod.3 = read.csv("/Users/morganhawkins/Desktop/36402_HW6_mod3.csv")[,-1] %>%
  as.matrix(nrow = 5)
colnames(prediction.errors.mod.3) = NULL

prediction.errors.mod.4 = read.csv("/Users/morganhawkins/Desktop/36402_HW6_mod4.csv")[,-1] %>%
  as.matrix(nrow = 5)
colnames(prediction.errors.mod.4) = NULL
```

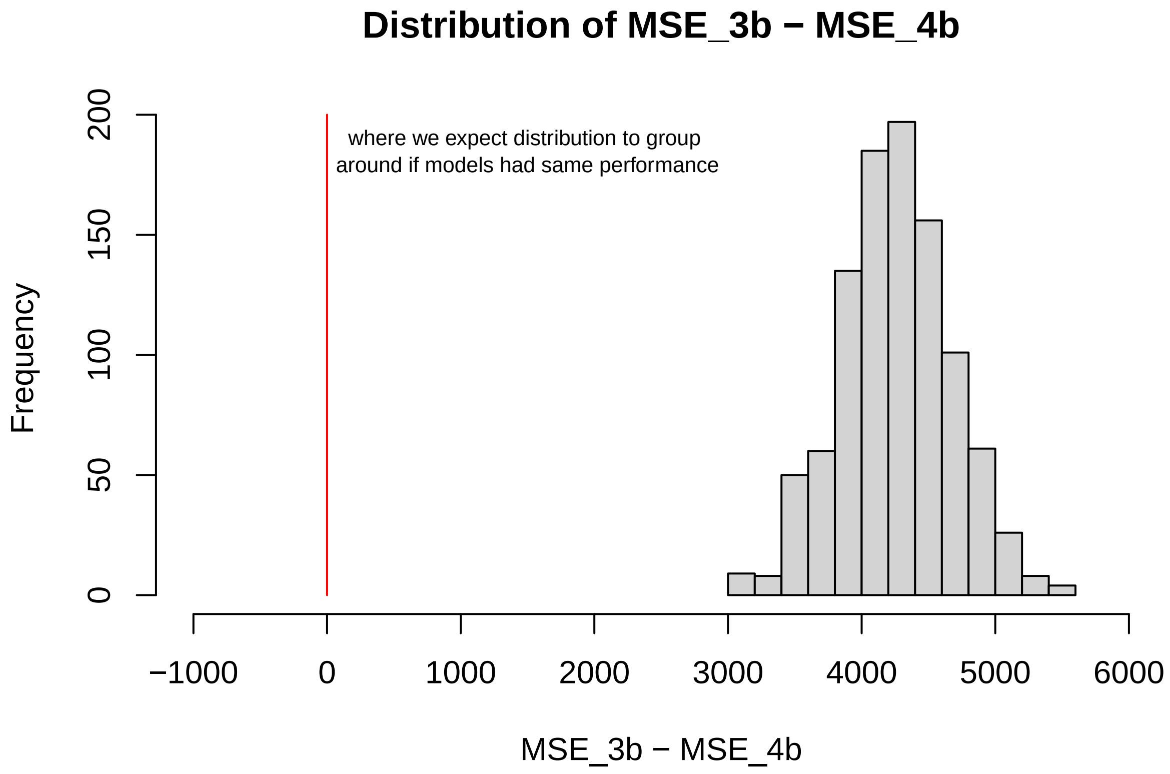
```r
#functions to clean up next code chunk
#just returns model fit from data passed
fit.mod.3 = function(f.data){
  return(lm(Median_house_value ~ Population + isPA + Latitude:isPA +
              Longitude:isPA + Median_household_income + Mean_household_income,
            data = f.data))
}

fit.mod.4 = function(f.data, f.data.test){
  f.bws = sapply(f.data[ ,c(1,2,3,5,6)], sd)/nrow(f.data)^(0.2)

  return(npreg(Median_house_value ~ Population + Latitude + Longitude +
                 Median_household_income + Mean_household_income,
               data = f.data,
               newdata = f.data.test,
               bws = f.bws
  ))
}
```

```r
#finding first sample with incomplete prediction error calculations
first.incomplete.column = 201
for(i in 1:200){
  temp.mod.3.col.incomplete = sum(is.na(prediction.errors.mod.3[,i])) > 0
  temp.mod.4.col.incomplete = sum(is.na(prediction.errors.mod.4[,i])) > 0

  if(temp.mod.3.col.incomplete | temp.mod.4.col.incomplete){
    first.incomplete.column = i
    break
  }
}


cat("starting at sample",first.incomplete.column,"\n\n")
```

```
## starting at sample 201
```

```r
for(sample in first.incomplete.column:200){
  #breaking loop if first incomplete column > number of columns in data set
  #true when lines 103-113 does not find an incomplete column
  if(sample > 200){
    break
  }

  cat("sample",sample,"\n")

  temp.full = house[bs.samples[,sample],]

  for(fold_num in 0:4){
    cat("  ","fold",fold_num+1)

    #splitting train and test
    test.fold.indices = ((fold_num*2121)+1):((fold_num+1)*2121)
    temp.test = temp.full[test.fold.indices,]
    temp.train = temp.full[-test.fold.indices,]

    #fitting models
    mod.3.temp = fit.mod.3(temp.train)
    mod.4.temp = fit.mod.4(temp.train, temp.test)


    #calculating errors
    mod.3.pred.errs = predict(mod.3.temp, temp.test) - temp.test$Median_house_value
    mod.4.pred.errs = mod.4.temp$mean - temp.test$Median_house_value
    prediction.errors.mod.3[(fold_num+1),sample] = mean(mod.3.pred.errs^2)
    prediction.errors.mod.4[(fold_num+1),sample] = mean(mod.4.pred.errs^2)

    cat(" - done\n")
  }
  cat("\n")

  #saving calculated errors
  write.csv(data.frame(prediction.errors.mod.3), "/Users/morganhawkins/Desktop/36402_HW6_mod3.csv")
  write.csv(data.frame(prediction.errors.mod.4), "/Users/morganhawkins/Desktop/36402_HW6_mod4.csv")

}
```

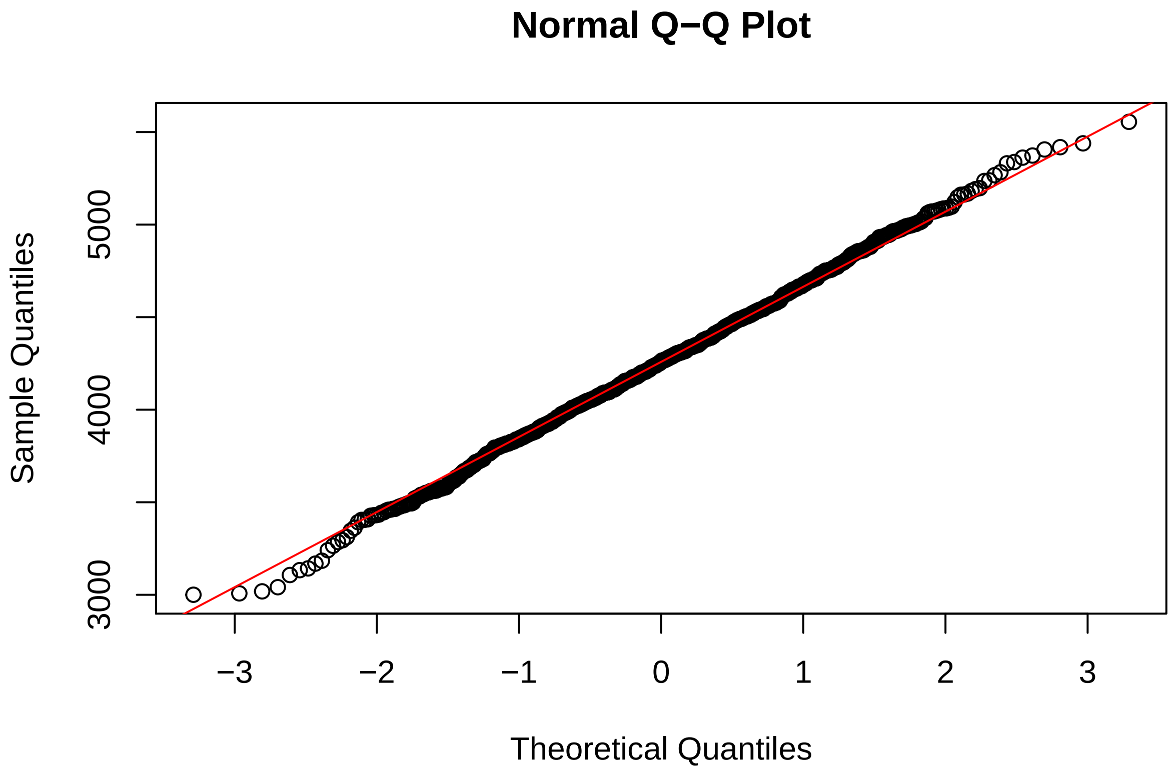
```r
mse.3b = sapply(prediction.errors.mod.3,mean)
mse.4b = sapply(prediction.errors.mod.4,mean)

hist((mse.3b - mse.4b), main = "Distribution of MSE_3b - MSE_4b",
     xlab = "MSE_3b - MSE_4b", xlim = c(-1000,6000))
segments(0,0,0,200,col = "red")
"where we expect distribution to group \naround if models had same performance" %>%
  text(1500,185,., cex = .7)
```



We can see on the histogram of $\widehat{MSE}^*_{3b} - \widehat{MSE}^*_{4b}$ that model 3 appears to have a higher MSE. This is shown by the fact that there are 0 boot trap samples where model 4 has a higher MSE.

Problem 1 (b)

```r
qqnorm((mse.3b - mse.4b))
qqline((mse.3b - mse.4b), col = "red")
```



This sample does appear to be normally distributed as the data points appear to fall close to the line.

```
t.test(mse.3b, mse.4b)
```

```
##
##  Welch Two Sample t-test
##
## data:  mse.3b and mse.4b
## t = 167.8, df = 1942.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  4206.657 4306.154
## sample estimates:
## mean of x mean of y
## 14217.198  9960.792
```

The t test reveals that model 4 is likely a better model than model 3. The p-value associated with the null hypothesis that MSE_3b = MSe_4b has a p-value of under 2.2e-16.

Problem 2

Problem 2 (a)

```
mobility = na.omit(mobility)

mod.2a = lm(Mobility ~ State + Longitude + Latitude + Seg_racial + Gini +
              Social_capital + School_spending,
            data = mobility)

mod.2a$coefficients["Social_capital"]
```

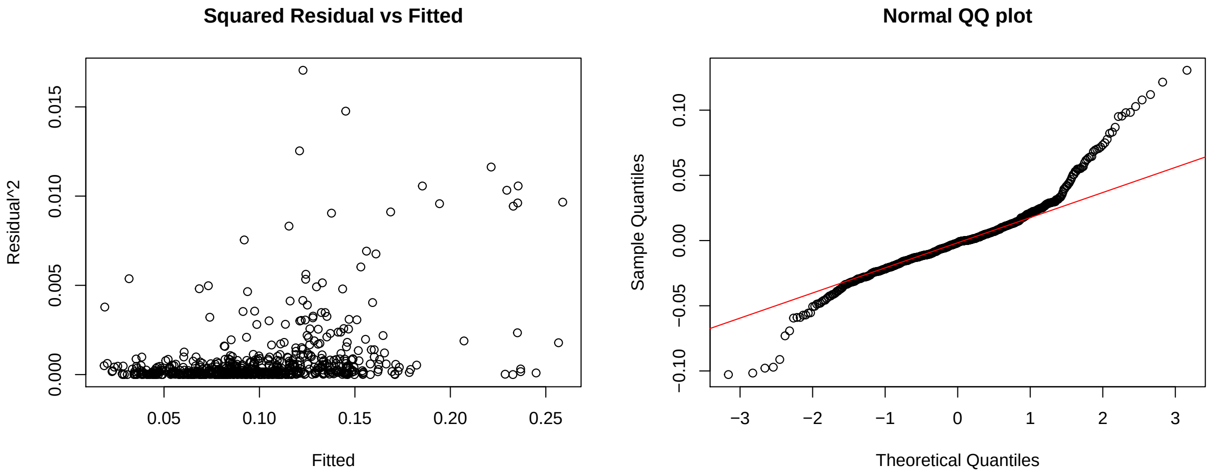
```
## Social_capital
##    0.004331108
```

```
confint(mod.2a, "Social_capital", level = 0.95)
```

```
##                        2.5 %      97.5 %
## Social_capital 0.0005712503 0.008090965
```

We estimate that for every unit increase in social capital, the expected increase in mobility is .00433. We are 95% confident that if we were to recreate this experiment with newly sampled data, this estimate would fall between .00057 and .00809.

```
par(mfrow = c(1,2))
plot(fitted(mod.2a), residuals(mod.2a)^2, xlab = "Fitted", ylab = "Residual^2",
     main = "Squared Residual vs Fitted")
qqnorm(residuals(mod.2a), main = "Normal QQ plot")
qqline(residuals(mod.2a), col = "red")
```

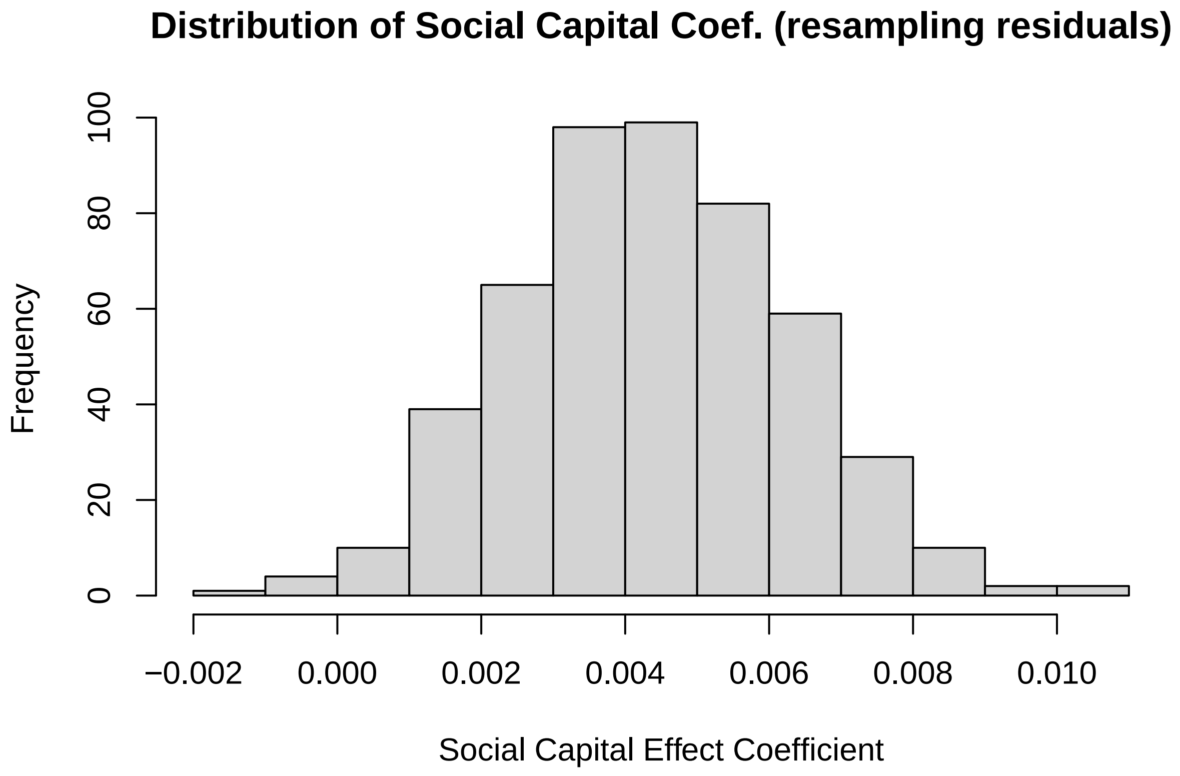


On the squared residual vs fitted plot we see that residuals appear to have higher variance with larger fitted values. This violates our assumption of our residuals having constant variance. Our assumption of normally distributed residuals also appears to be violated as shown by the deviations from the line on the normal qq plot. Both of these issues suggest that the standard error estimated in the regression model does not make sense to use.

Problem 2 (b)

```
predictions.2b = fitted(mod.2a)
residuals.2b = residuals(mod.2a)
temp.data = mobility
```

```
set.seed(999)
bs.coefs = rep(0, 500)

for(i in 1:500){
  temp.data$Mobility = predictions.2b + sample(residuals.2b, replace = TRUE)

  temp.mod.2b = lm(Mobility ~ State + Longitude + Latitude + Seg_racial + Gini +
                     Social_capital + School_spending,
                   data = temp.data)

  bs.coefs[i] = temp.mod.2b$coefficients["Social_capital"]
}


hist(bs.coefs, main = "Distribution of Social Capital Coef. (resampling residuals)",
     xlab = "Social Capital Effect Coefficient", breaks = 15)
```

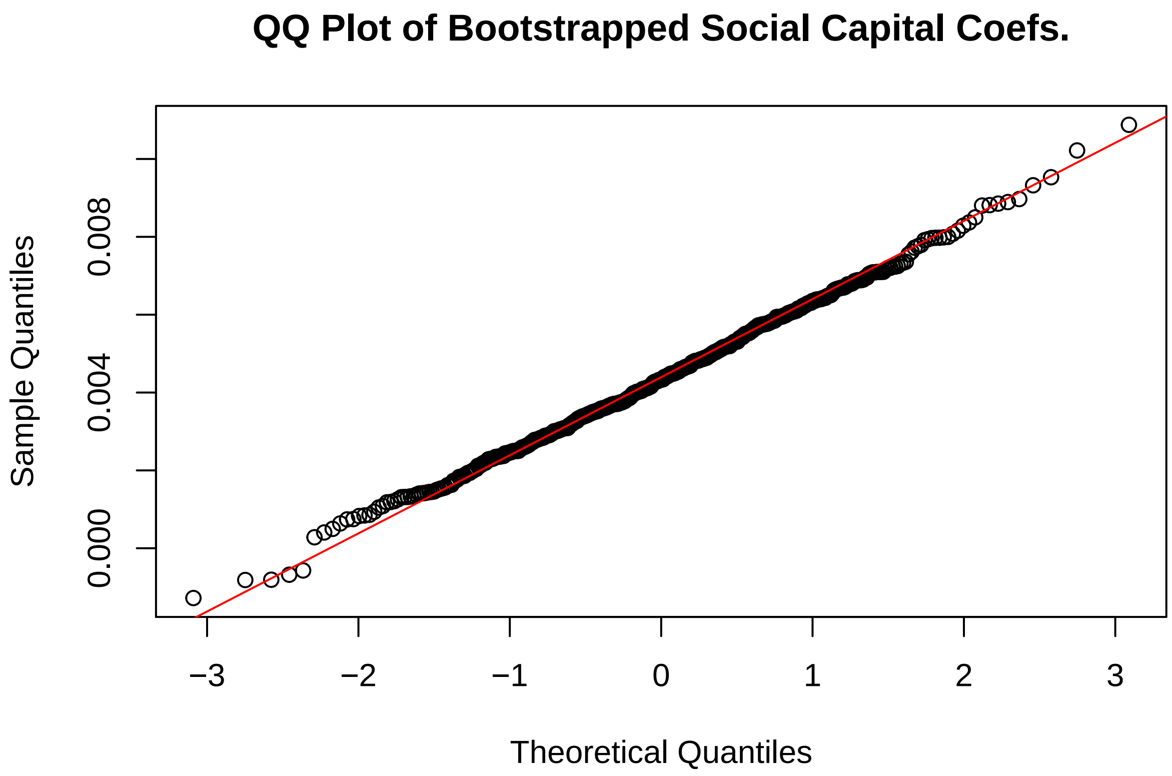


```
bs.coefs.se = sd(bs.coefs)

cat("(", 0.004331108 + bs.coefs.se*qnorm(.025), ",",
    0.004331108 + bs.coefs.se*qnorm(.975), ")")
```

```
## ( 0.0005487724 , 0.008113444 )
```

Bootstrapping by resampling residuals gives us an estimated standard error of 0.0019. Using this, we now estimate that if we were to recreate this experiment with newly sampled data, we are 95% confident that our coefficient for social capital would fall in the interval [0.0005487724, 0.008113444]. The width of this interval is only about 0.6% wider than our confidence interval computed in part (a).

```
qqnorm(bs.coefs, main = "QQ Plot of Bootstrapped Social Capital Coefs.")
qqline(bs.coefs, col = 'red')
```

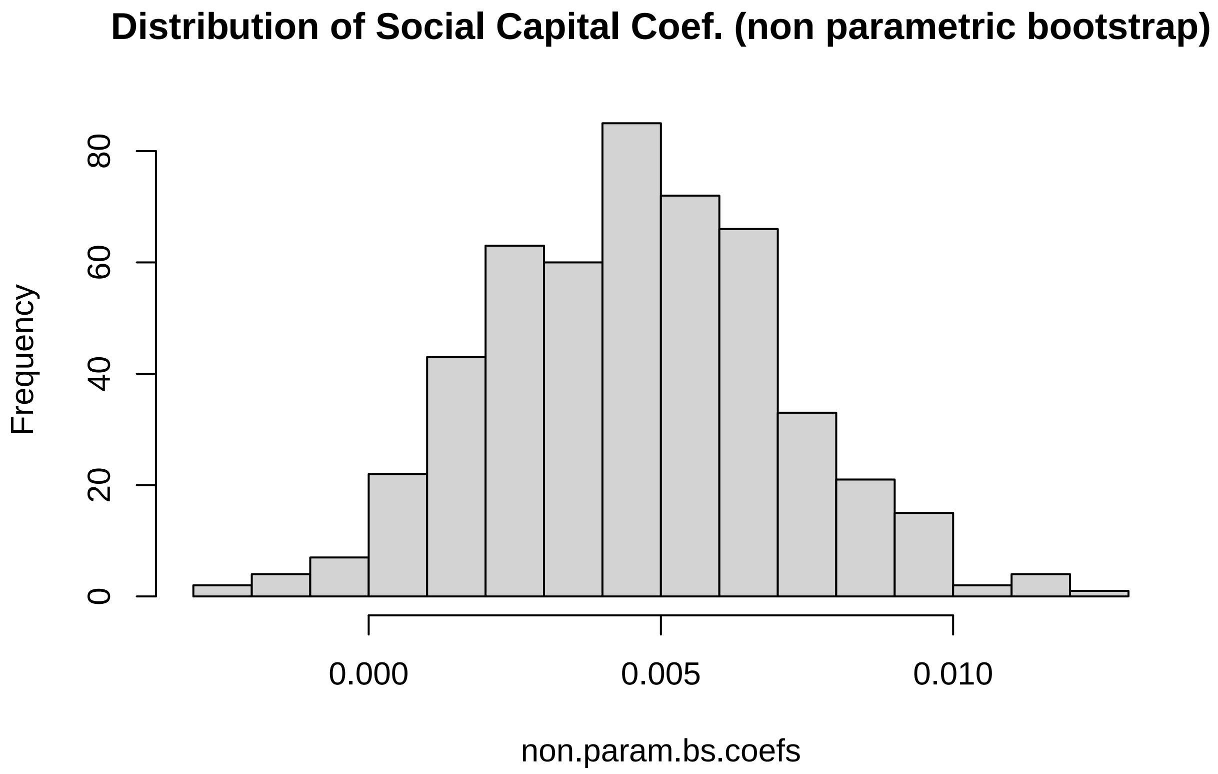


Our first concern with our previous confidence interval was our standard error being incorrect. Using bootstrapping addressed this issue. Our second concern was that our residuals were not normally distributed. This means that our social capital coefficient may not have been normally distributed. We can now confirm that social capital's coefficient is normally distributed through the qq plot above.

Resampling residuals may not be a good idea because we are not confident that our model is the correct shape. The kernel smoother was able to generalize far better than the linear model which had an additional explanatory variables. This means that the relationships between the response and the explanatory variables may not be linear so we should use case resampling instead of residual resampling.

Problem 2 (c)

```
non.param.bs.coefs = rep(0, 500)

for(i in 1:500){
  temp.data = mobility[sample(1:635, replace = TRUE),]

  temp.mod.2b = lm(Mobility ~ State + Longitude + Latitude + Seg_racial + Gini +
                     Social_capital + School_spending,
                   data = temp.data)

  non.param.bs.coefs[i] = temp.mod.2b$coefficients["Social_capital"]

}

hist(non.param.bs.coefs, breaks = 15,
     main = "Distribution of Social Capital Coef. (non parametric bootstrap)")
```



```
non.param.bs.se = sd(non.param.bs.coefs)

cat("(", 0.004331108 + non.param.bs.se*qnorm(.025), ",",
    0.004331108 + non.param.bs.se*qnorm(.975), ")")
```

```
## ( -0.0005657924 , 0.009228008 )
```