HA2 Problem 2.2

Ishwara Hegde, Jonathan Neiman and Aleksei Samkov

10/15/2020

Section 1: Prep

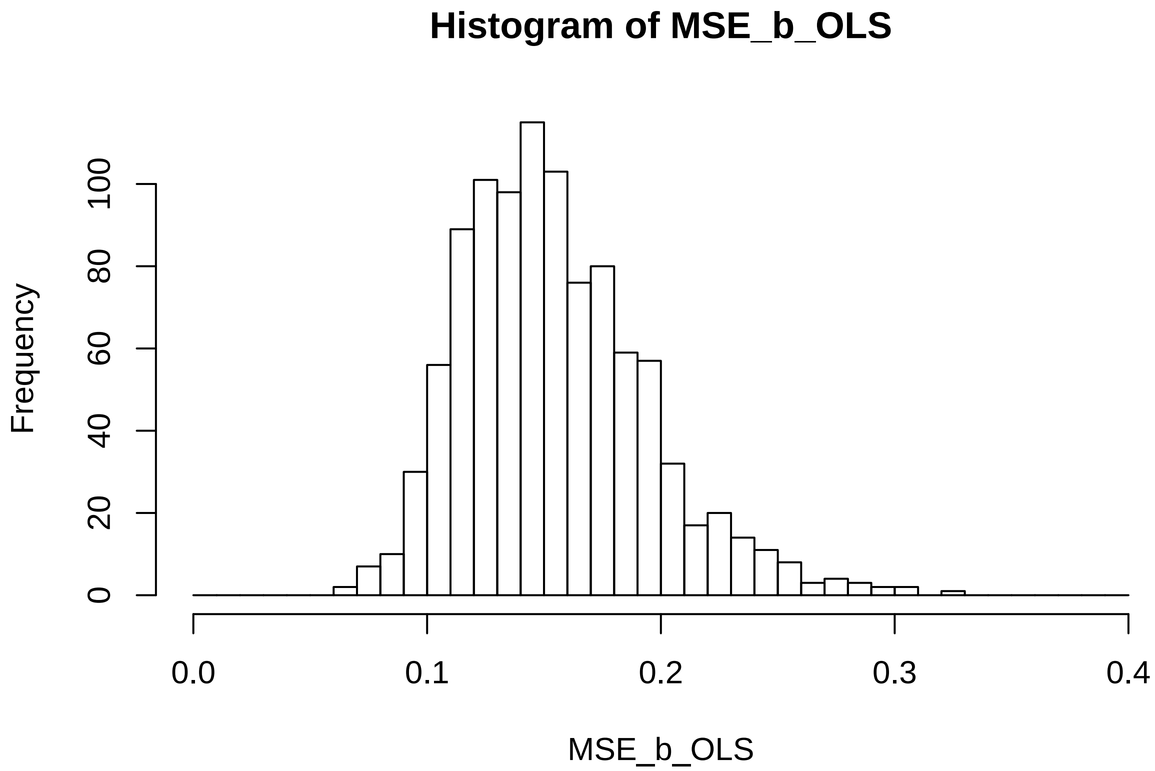
Here we import the data and define the relevant splines, interaction variables, and estimate the OLS fitted values which will serve as our “true” values of the conditional mean function. Lastly, we set the bootstrap parameters which will be used in the three subquestions.

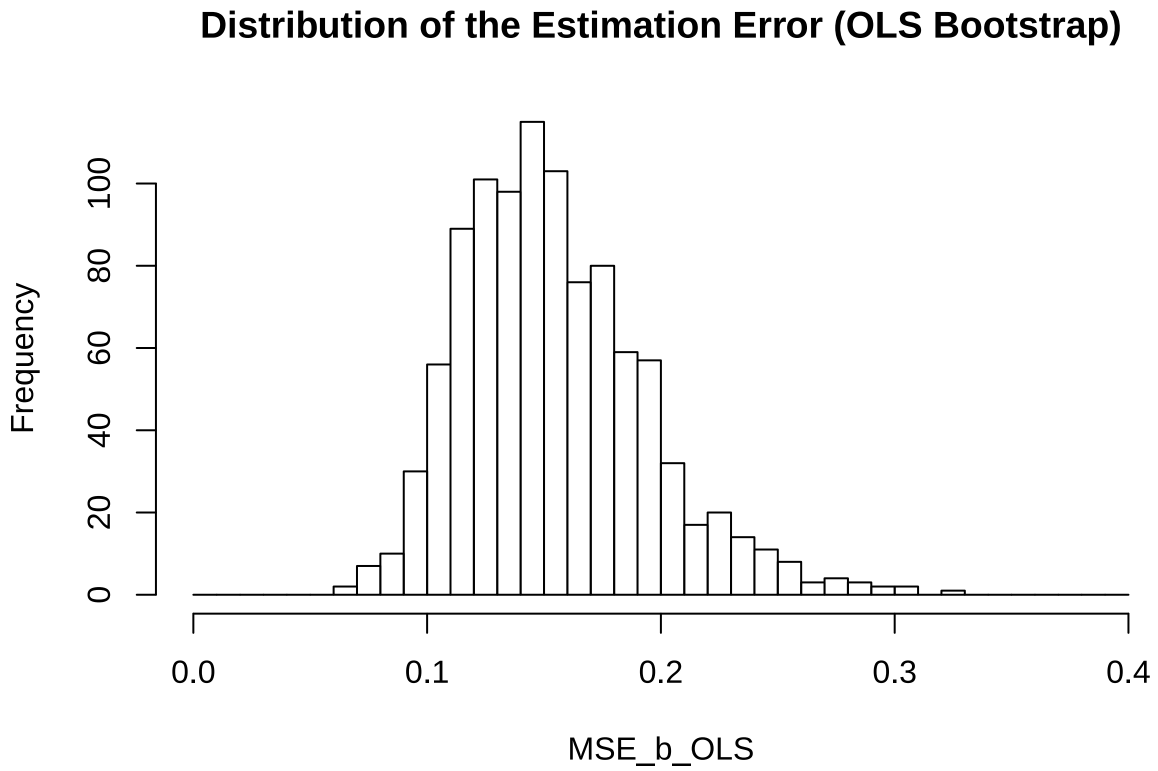
```
# Import data
setwd("../Data")
census00 <- read.csv("../Data/census00.csv")

# Determine Y variable
logwk <- census00$logwk

# Split variables into splines
df_af = 5
educ_s  <- ns(census00$educ, df = df_af)
age_s   <- ns(census00$age, df = df_af)
exper_s <- ns(census00$exper, df = df_af)

# Interactions
educ_s_blk  <- educ_s * census00$black
age_s_blk   <- age_s * census00$black
exper_s_blk <- exper_s * census00$black

# Linear regression and fitted values
model1 <- lm(logwk ~ educ_s + age_s + exper_s + educ_s_blk + age_s_blk +
               exper_s_blk)
Y_pred <- model1$fitted.values

# Bootstrap Parameters
n <- 300
B <- 1000
```

Section 2: OLS Bootstrap

```
MSE_b_OLS <- rep(0, B)

for (b in 1:B){
  index_b <- sample(1:n,n,replace = TRUE)
  Y_b   <- logwk[index_b]
  X_1sb <- educ_s[index_b,]
  X_2sb <- age_s[index_b,]
  X_3sb <- exper_s[index_b,]
```

```r
  X_4sb <- educ_s_blk[index_b,]
  X_5sb <- age_s_blk[index_b,]
  X_6sb <- exper_s_blk[index_b,]

  model2_b <- lm(Y_b ~ X_1sb + X_2sb + X_3sb + X_4sb + X_5sb + X_6sb)
  Y_pred_b <- model2_b$fitted.values

  f_b <- Y_pred[index_b]

  MSE_b_OLS[b] <- mean((Y_pred_b - f_b)^2)
}

plot(hist(MSE_b_OLS, breaks=seq(0,0.4,0.01)),
     main = 'Distribution of the Estimation Error (OLS Bootstrap)')
```



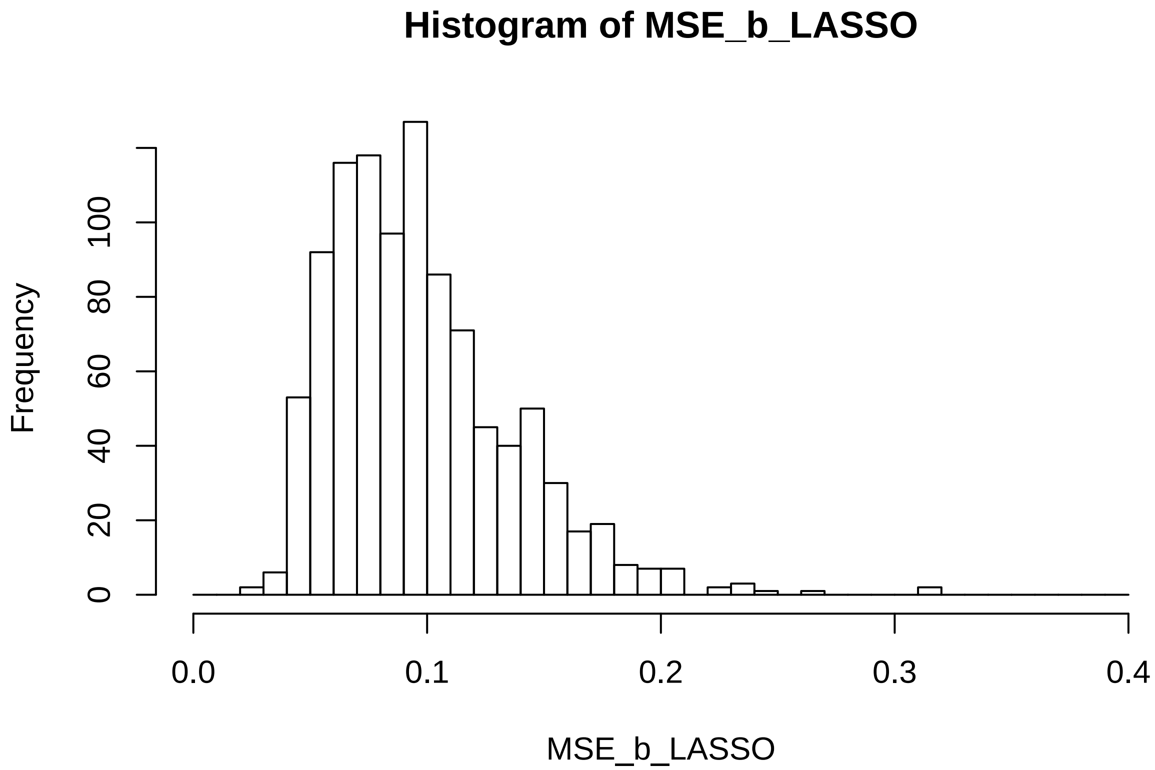


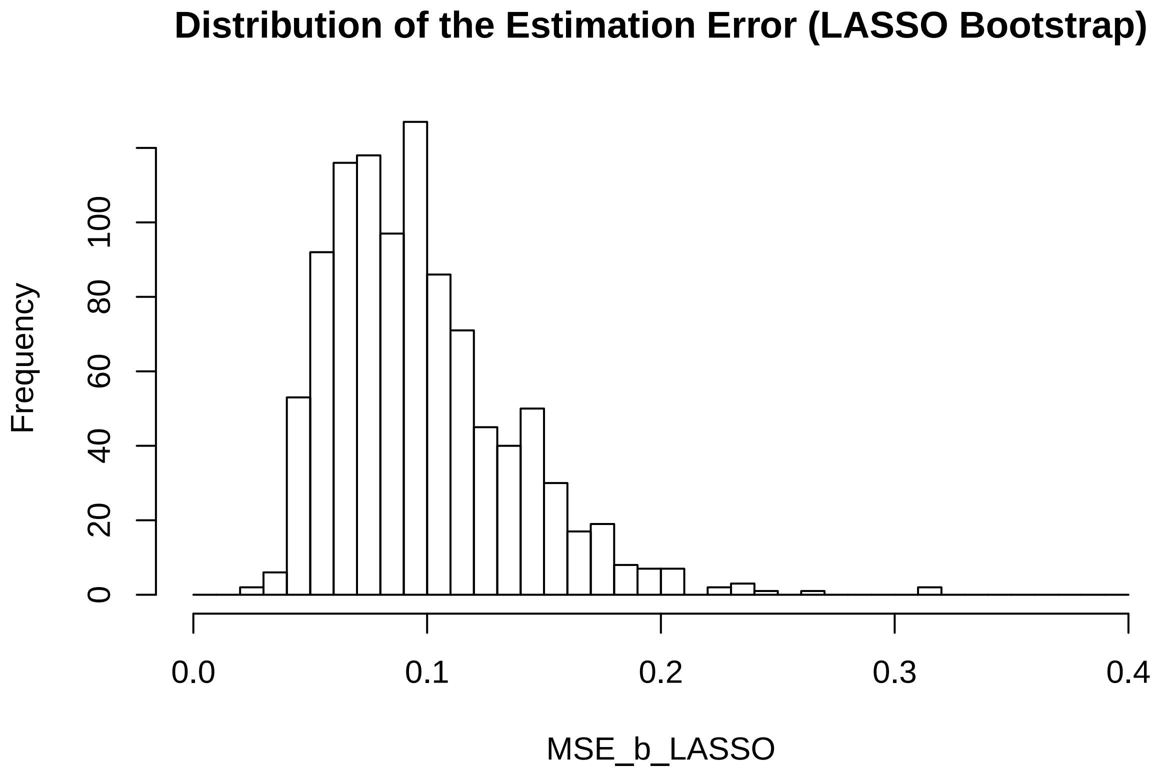
Section 3: LASSO with 5-fold CV Bootstrap

Unsurprisingly, the LASSO bootstrap performs somewhat better compared to the OLS bootstrap.

```
## Warning: from glmnet Fortran code (error code -95); Convergence for 95th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

```
## Warning: from glmnet Fortran code (error code -94); Convergence for 94th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

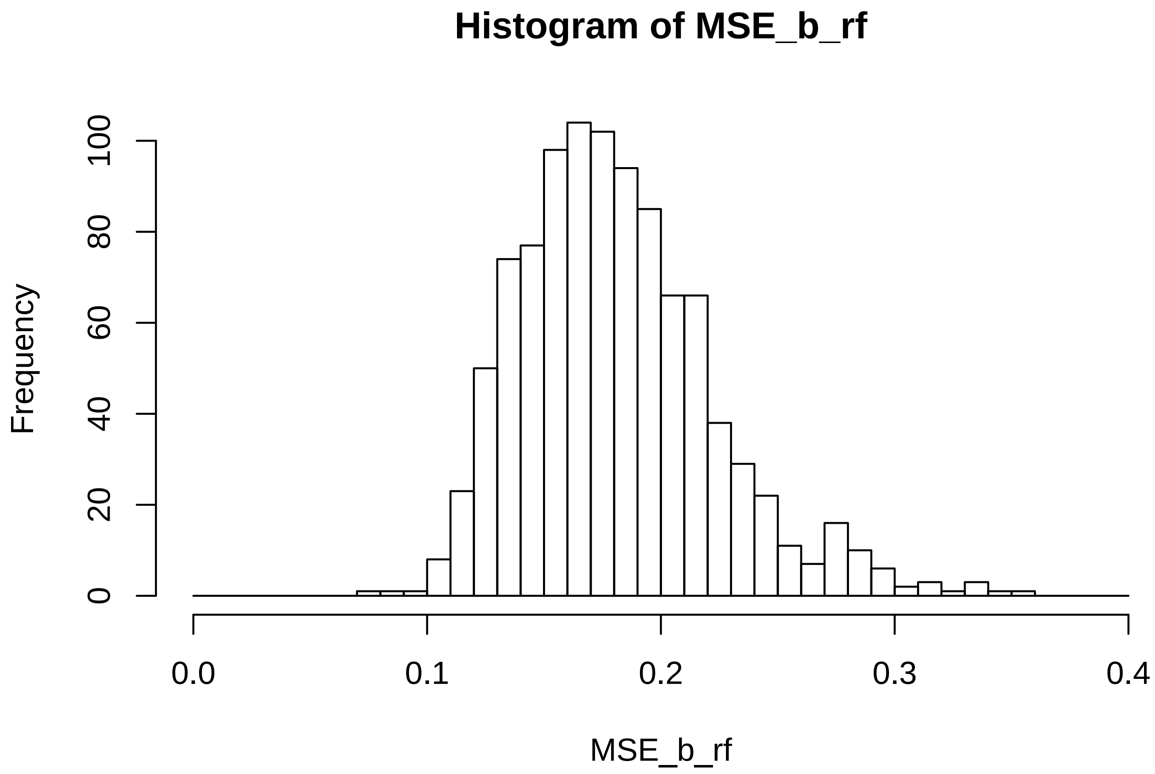

Histogram of MSE_b_LASSO
Frequency
0
20
40
60
80
100
0.0
0.1
0.2
0.3
0.4
MSE_b_LASSO

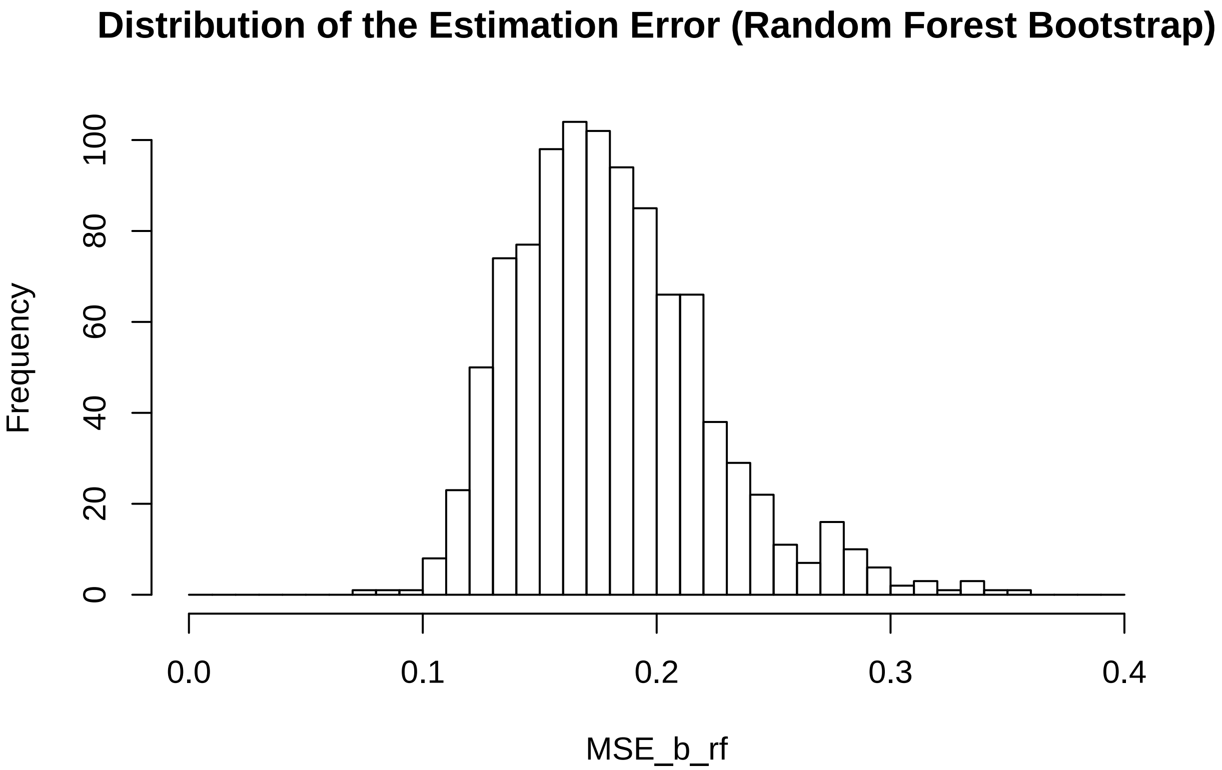


Section 4: Random Forest Bootstrap

The random forest here, despite being computationally intense, is actually worse than the initial two methods.Perhaps this is because of the large variance of decisiontrees. Random forests do in fact reduce this variance due to averaging but perhaps in this case the reduction was not sufficient to perform better than OLS annd LASSO.

```
MSE_b_rf <- rep(0, B)

educ  <- census00$educ
age   <- census00$age
exper <- census00$exper

for (b in 1:B){
  index_b <- sample(1:n,n,replace = TRUE)
  Y_b   <- logwk[index_b]
  X_1sb <- educ[index_b]
  X_2sb <- age[index_b]
  X_3sb <- exper[index_b]
  X_sb  <- cbind(X_1sb, X_2sb, X_3sb)

  model_rf <- randomForest(Y_b ~ X_1sb + X_2sb + X_3sb, importance = TRUE)

  f_b <- Y_pred[index_b]

  MSE_b_rf[b] <- mean((model_rf$predicted - f_b)^2)
}

plot(hist(MSE_b_rf, breaks=seq(0,0.4,0.01)),
     main = 'Distribution of the Estimation Error (Random Forest Bootstrap)')
```




Distribution of the Estimation Error (Random Forest Bootstrap)
Frequency
0
20
40
60
80
100
0.0
0.1
0.2
0.3
0.4
MSE_b_rf