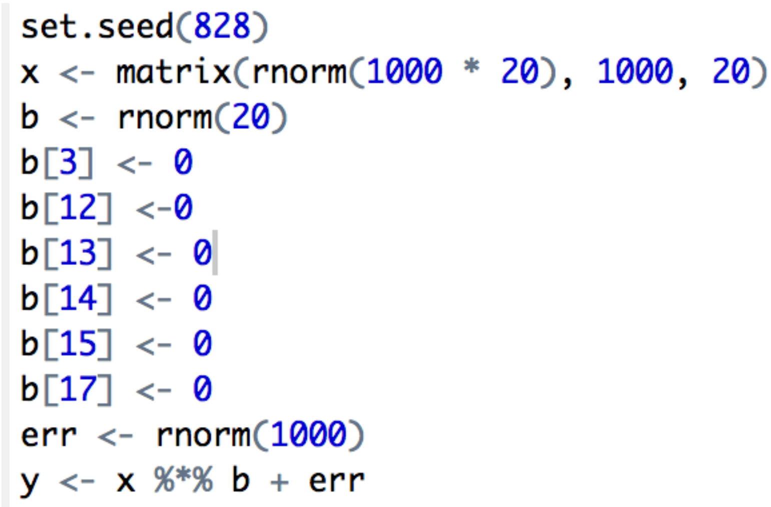
Problem Set 3

Shengwenxin Ni

February 10, 2020

1 Question 1

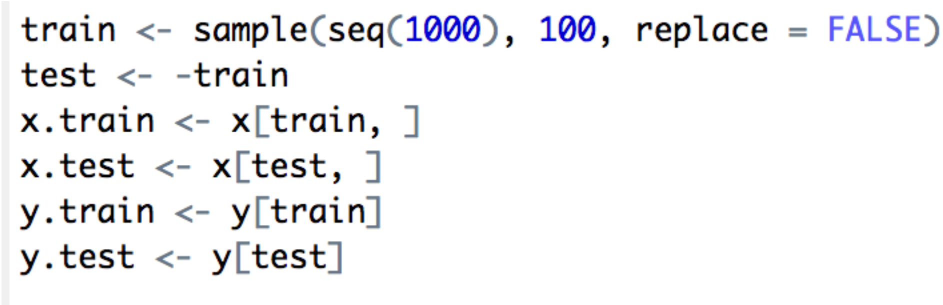
(1) Generate the dataset



```
set.seed(828)
x <- matrix(rnorm(1000 * 20), 1000, 20)
b <- rnorm(20)
b[3] <- 0
b[12] <-0
b[13] <- 0
b[14] <- 0
b[15] <- 0
b[17] <- 0
err <- rnorm(1000)
y <- x %*% b + err
```

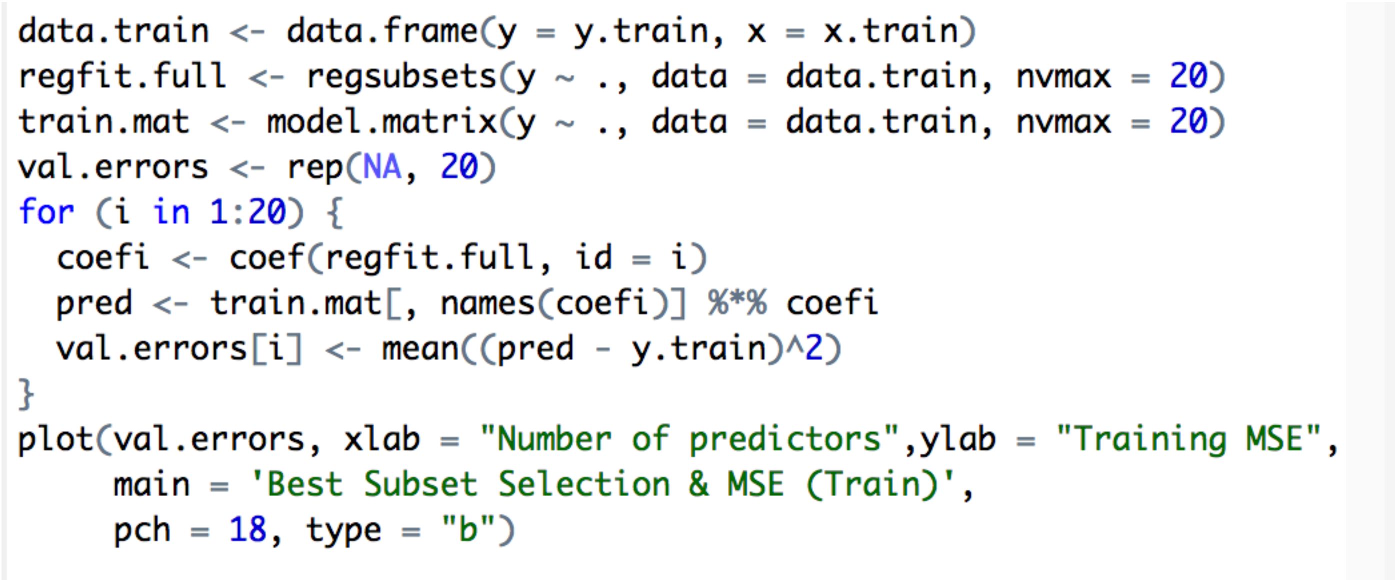
X: the data set with p = 20 features, n = 1000 observations
b: A list of numbers that store the values of betas including 6 zero values.
Y: The associated quantitative response vector generated according to the model

(2) Split the dataset

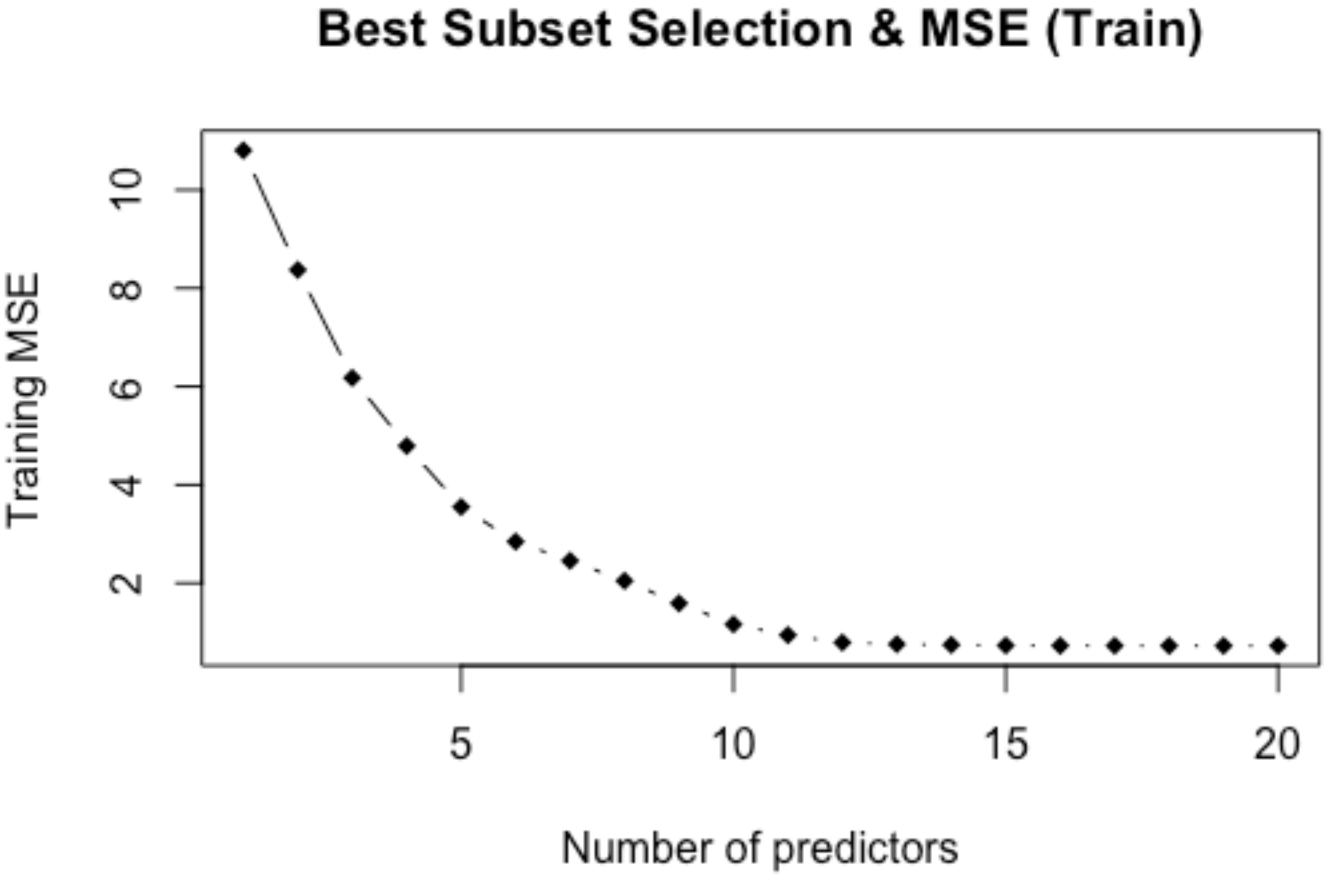


```
train <- sample(seq(1000), 100, replace = FALSE)
test <- -train
x.train <- x[train, ]
x.test <- x[test, ]
y.train <- y[train]
y.test <- y[test]
```

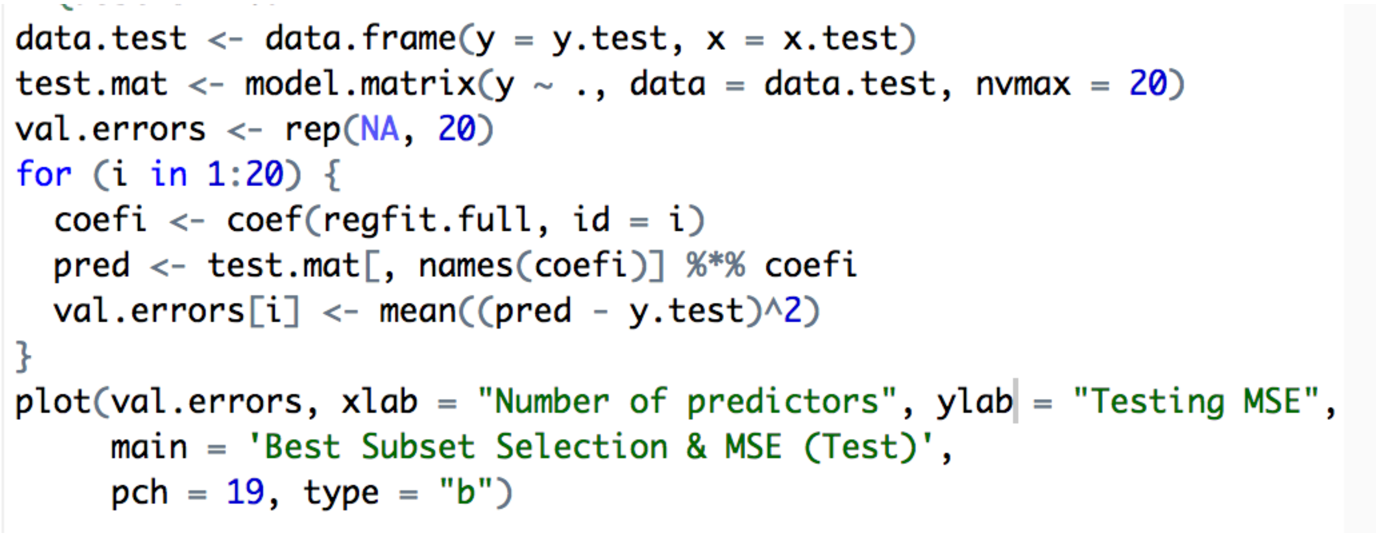
(3) Best subset selection & training MSE



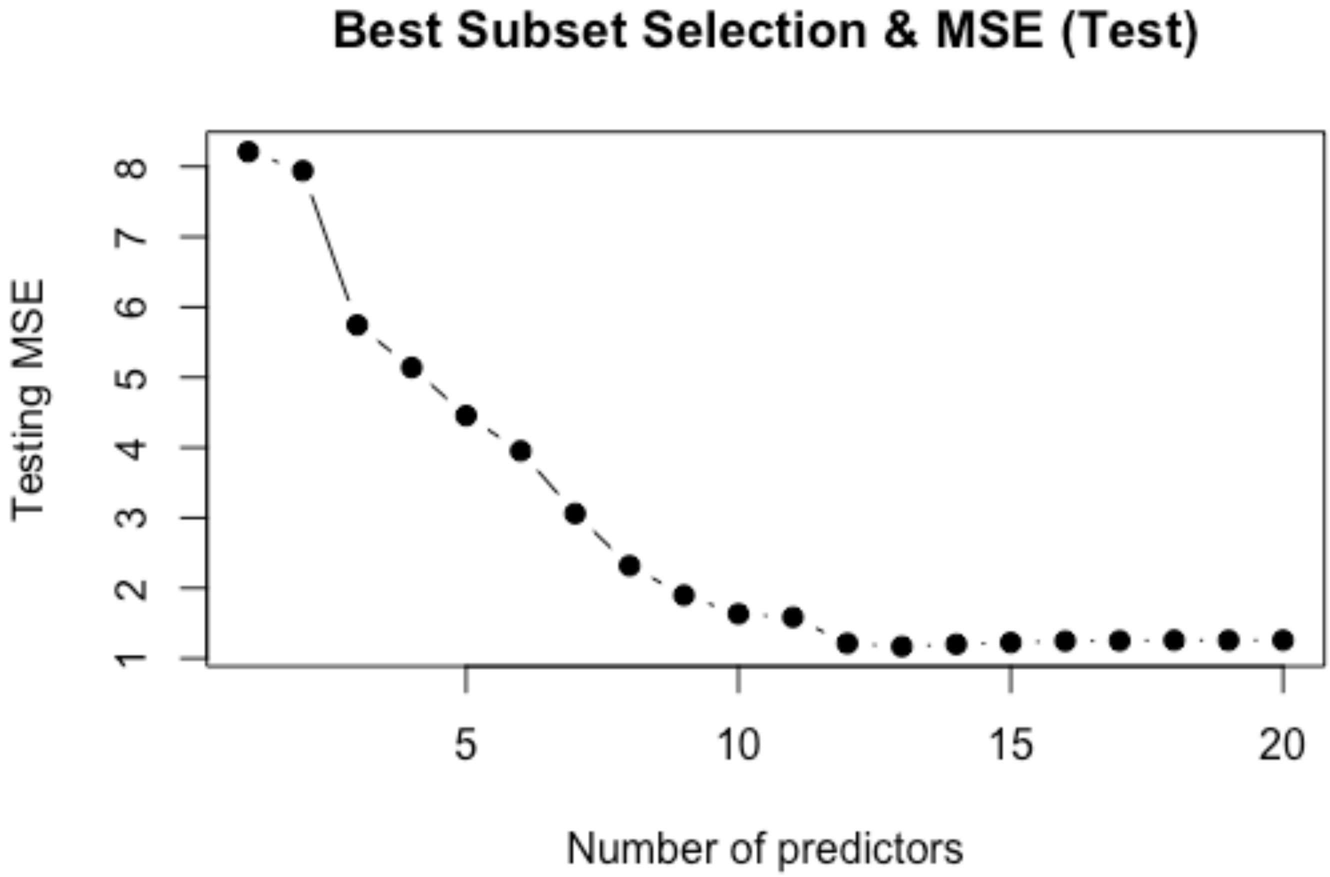
```r
data.train <- data.frame(y = y.train, x = x.train)
regfit.full <- regsubsets(y ~ ., data = data.train, nvmax = 20)
train.mat <- model.matrix(y ~ ., data = data.train, nvmax = 20)
val.errors <- rep(NA, 20)
for (i in 1:20) {
  coefi <- coef(regfit.full, id = i)
  pred <- train.mat[, names(coefi)] %*% coefi
  val.errors[i] <- mean((pred - y.train)^2)
}
plot(val.errors, xlab = "Number of predictors",ylab = "Training MSE",
     main = 'Best Subset Selection & MSE (Train)',
     pch = 18, type = "b")
```



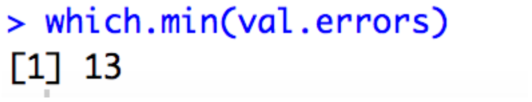
(4) Best subset selection & Testing MSE



```
data.test <- data.frame(y = y.test, x = x.test)
test.mat <- model.matrix(y ~ ., data = data.test, nvmax = 20)
val.errors <- rep(NA, 20)
for (i in 1:20) {
  coefi <- coef(regfit.full, id = i)
  pred <- test.mat[, names(coefi)] %*% coefi
  val.errors[i] <- mean((pred - y.test)^2)
}
plot(val.errors, xlab = "Number of predictors", ylab = "Testing MSE",
     main = 'Best Subset Selection & MSE (Test)',
     pch = 19, type = "b")
```

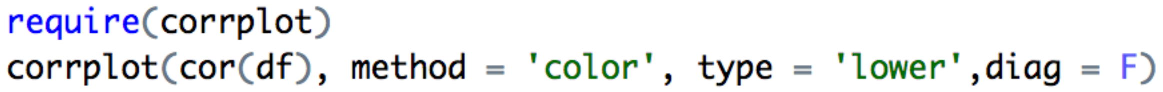


(5) Model size with min MSE

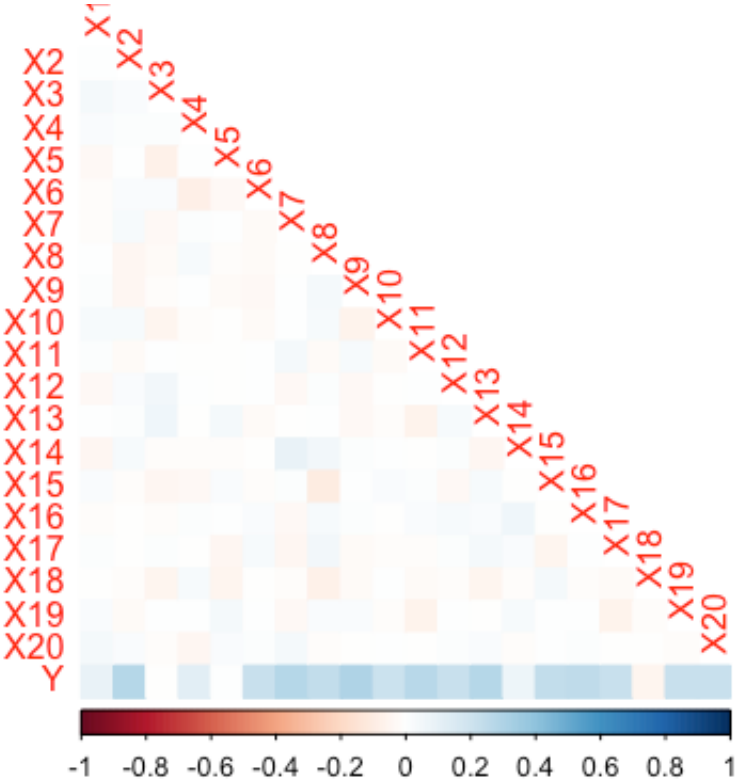


```
> which.min(val.errors)
[1] 13
```

From the above code, it's clear that when the model takes 13 coefficients, the test set MSE take on its minimum value.



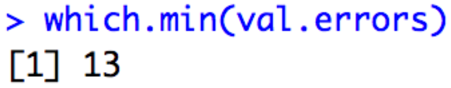
```
require(corrplot)
corrplot(cor(df), method = 'color', type = 'lower',diag = F)
```



From the above corrplot, we can see that there are around 5 variables (light blue or light red) that do not correlate well with the response variable. If we include these variables in prediction, we may overfit the data with the noises included and therefore, did not decreases MSE in the testing dataset.

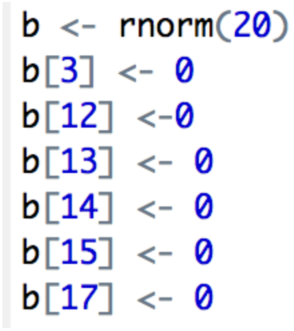
(6) Model with min MSE vs true model

Below is the variables (and its coefficients) for the model with minimum MSE.

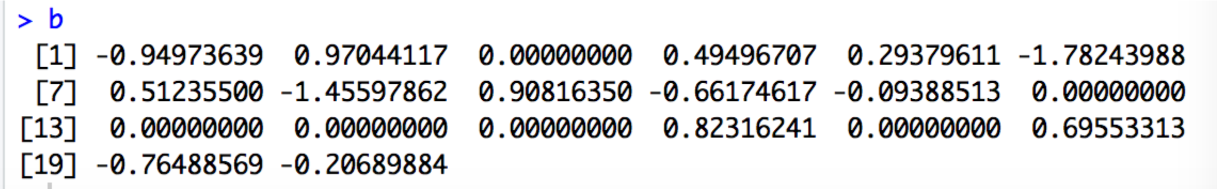


```
> which.min(val.errors)
[1] 13
```

We may compare it with the βs:



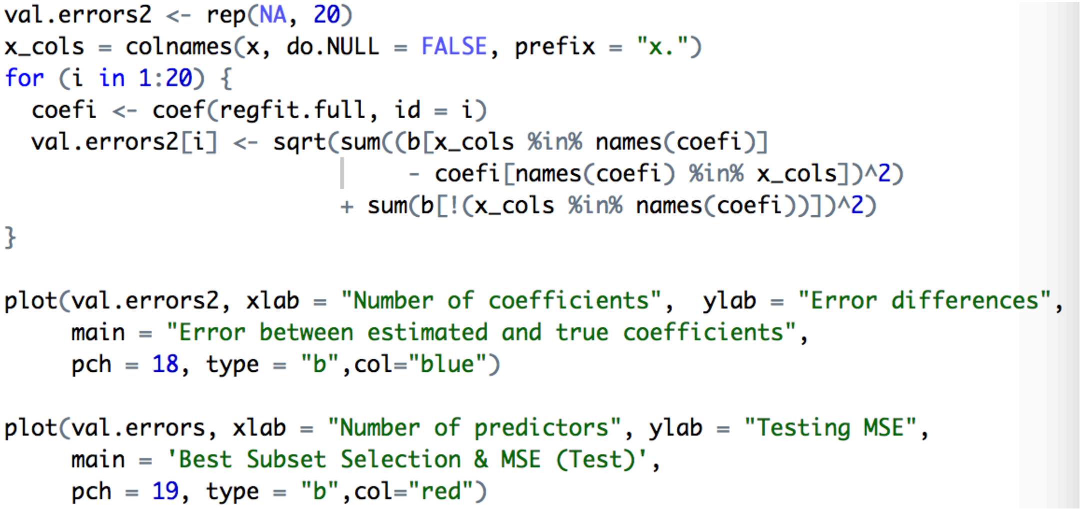
```
b <- rnorm(20)
b[3] <- 0
b[12] <-0
b[13] <- 0
b[14] <- 0
b[15] <- 0
b[17] <- 0
```



```
> b
 [1] -0.94973639  0.97044117  0.00000000  0.49496707  0.29379611 -1.78243988
 [7]  0.51235500 -1.45597862  0.90816350 -0.66174617 -0.09388513  0.00000000
[13]  0.00000000  0.00000000  0.00000000  0.82316241  0.00000000  0.69553313
[19] -0.76488569 -0.20689884
```

It's clear from the comparison that the best model does not include item 3, 12, 13, 14, 15, 17 of the β-list, which are all zeros. It successfully catches all 0-coefficients, which are non-existing in the true model.

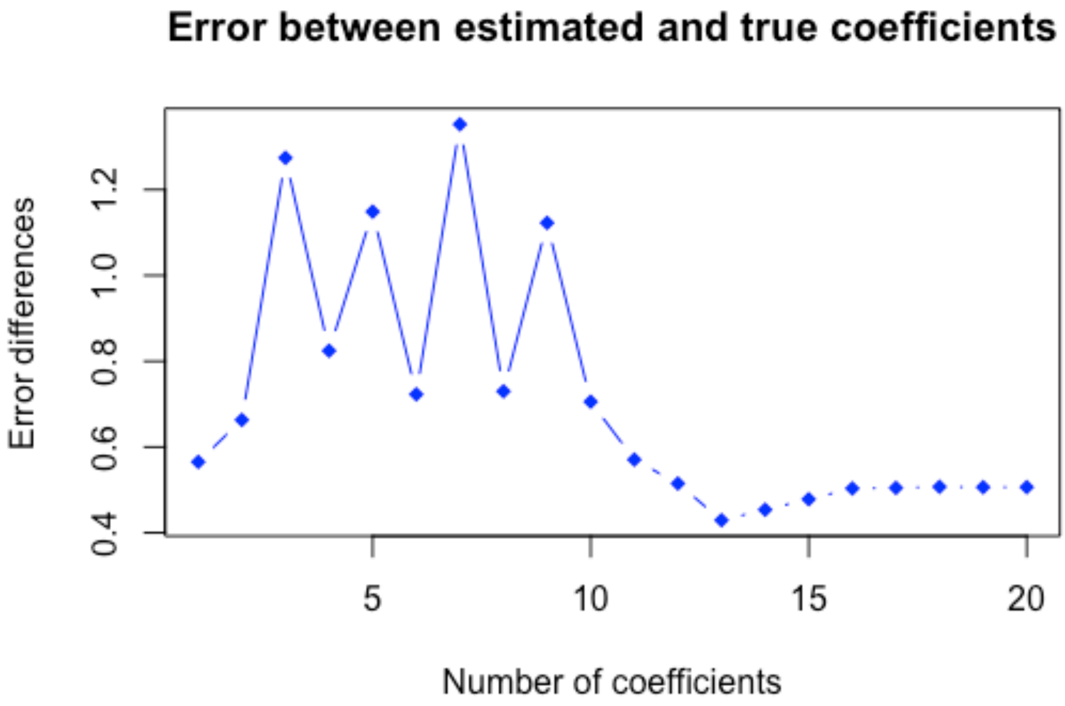
(7) Error Differences and test MSE

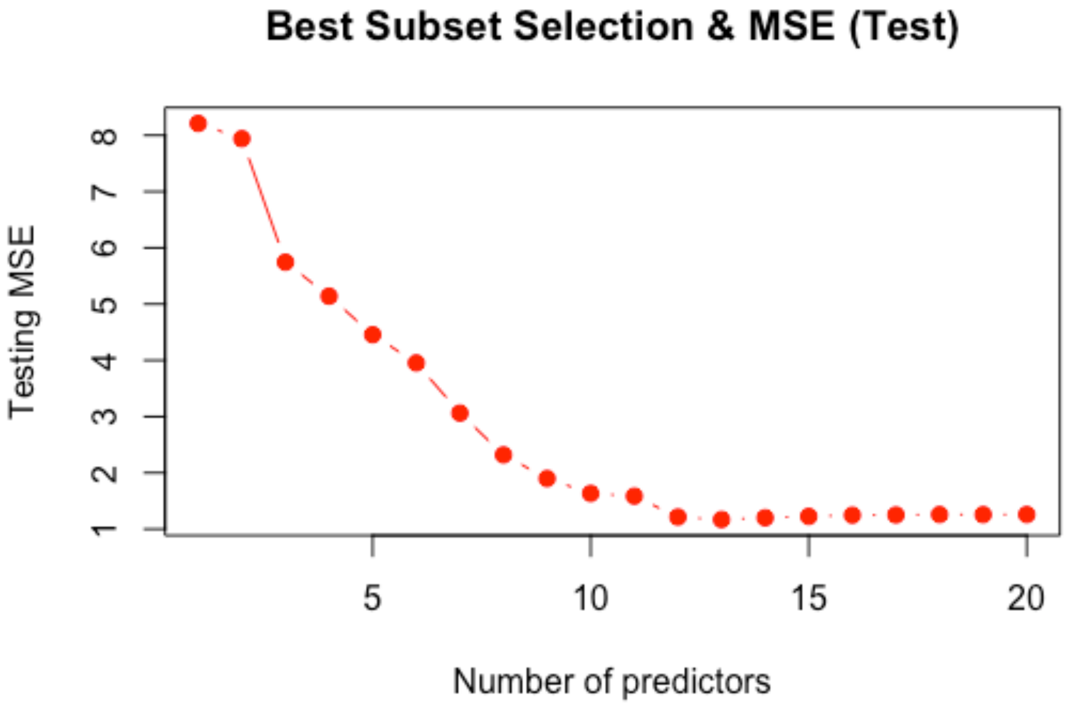


```
val.errors2 <- rep(NA, 20)
x_cols = colnames(x, do.NULL = FALSE, prefix = "x.")
for (i in 1:20) {
  coefi <- coef(regfit.full, id = i)
  val.errors2[i] <- sqrt(sum((b[x_cols %in% names(coefi)]
                                  - coefi[names(coefi) %in% x_cols])^2)
                             + sum(b[!(x_cols %in% names(coefi))])^2)
}

plot(val.errors2, xlab = "Number of coefficients",  ylab = "Error differences",
     main = "Error between estimated and true coefficients",
     pch = 18, type = "b",col="blue")

plot(val.errors, xlab = "Number of predictors", ylab = "Testing MSE",
     main = 'Best Subset Selection & MSE (Test)',
     pch = 19, type = "b",col="red")
```





From the graph, we can notice that:

- The error between the estimated and true coefficients model first fluctuates significantly and then decreases after 9 predictors. Next, it increases slightly after 13 predictors.
- The error is minimized when the number of coefficients equals to 13.
- The error is low when the number of coefficients equals to: 1, 2, 6, 8, 10-20.
- MSE test error keeps decreasing and flattens around 12 predictors.
- MSE test error is minimized when the number of coefficients equals to 13
- MSE test error is low from 10-20.

From the above observations, we may conclude that:

- **A better fit of true coefficients doesnt necessarily mean a lower test MSE.**
- **Being close to the true model (around 14 predictors in this case) would result both low values for error differences and test MSE.**

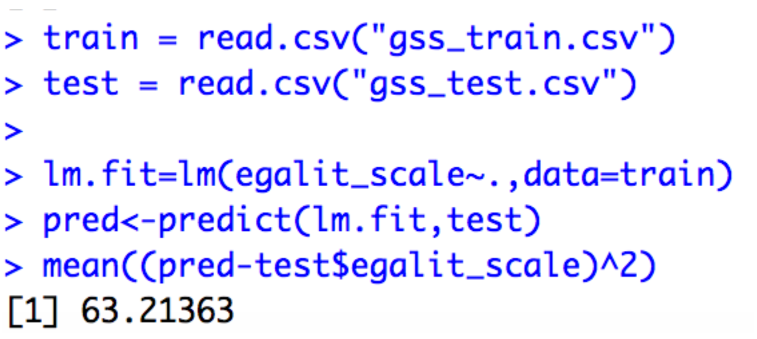
References

Here are two webpages that help with this question.

- https://rstudio-pubs-static.s3.amazonaws.com 65562_c062f4bb166140c6b7126b01adb27444.html
- https://rpubs.com/bgautijonsson/353732

2 Question 2

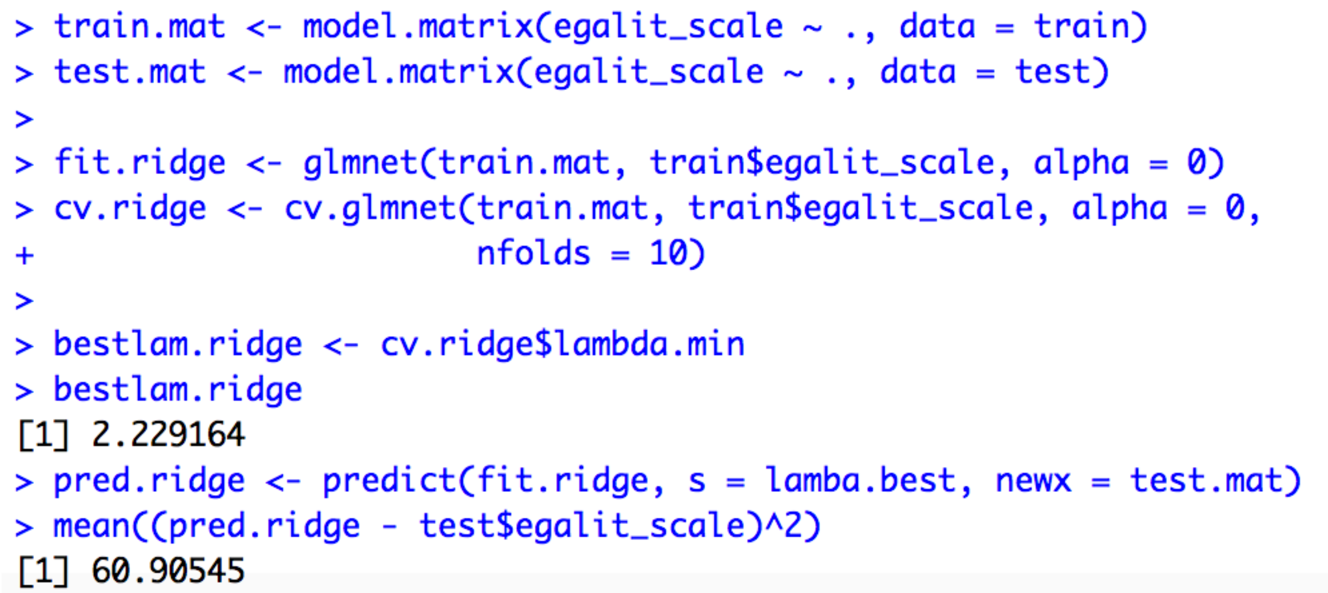
(1) Least squares linear model



```
> train = read.csv("gss_train.csv")
> test = read.csv("gss_test.csv")
>
> lm.fit=lm(egalit_scale~.,data=train)
> pred<-predict(lm.fit,test)
> mean((pred-test$egalit_scale)^2)
[1] 63.21363
```

The MSE value for this regression model is **63.21363**.

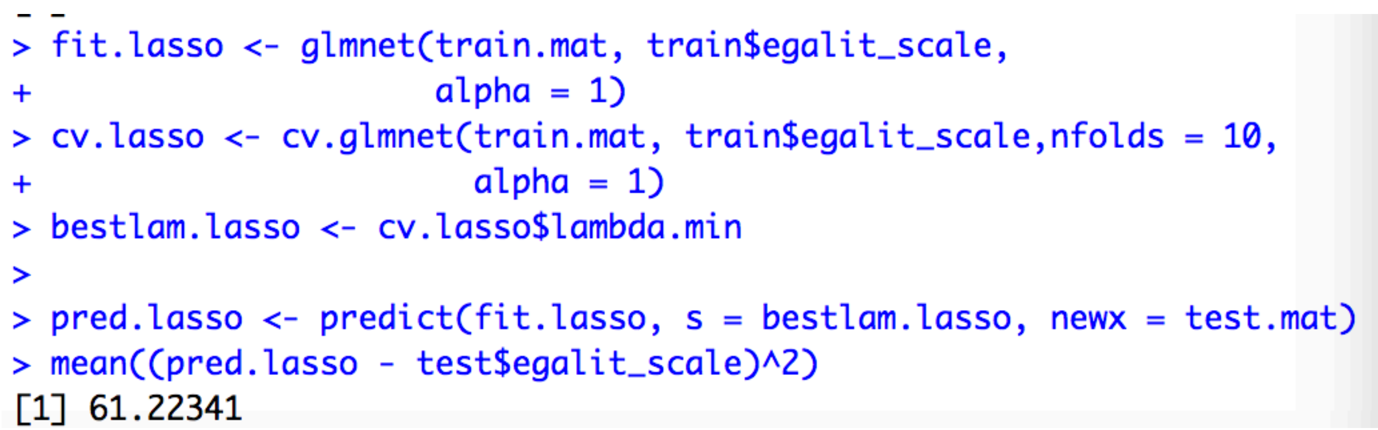
(2) Ridge regression model



```
> train.mat <- model.matrix(egalit_scale ~ ., data = train)
> test.mat <- model.matrix(egalit_scale ~ ., data = test)
>
> fit.ridge <- glmnet(train.mat, train$egalit_scale, alpha = 0)
> cv.ridge <- cv.glmnet(train.mat, train$egalit_scale, alpha = 0,
+                        nfolds = 10)
>
> bestlam.ridge <- cv.ridge$lambda.min
> bestlam.ridge
[1] 2.229164
> pred.ridge <- predict(fit.ridge, s = lamba.best, newx = test.mat)
> mean((pred.ridge - test$egalit_scale)^2)
[1] 60.90545
```

The MSE value for this regression model is **60.90545**.

(3) Lasso regression model



```
- -
> fit.lasso <- glmnet(train.mat, train$egalit_scale,
+                      alpha = 1)
> cv.lasso <- cv.glmnet(train.mat, train$egalit_scale,nfolds = 10,
+                        alpha = 1)
> bestlam.lasso <- cv.lasso$lambda.min
>
> pred.lasso <- predict(fit.lasso, s = bestlam.lasso, newx = test.mat)
> mean((pred.lasso - test$egalit_scale)^2)
[1] 61.22341
```

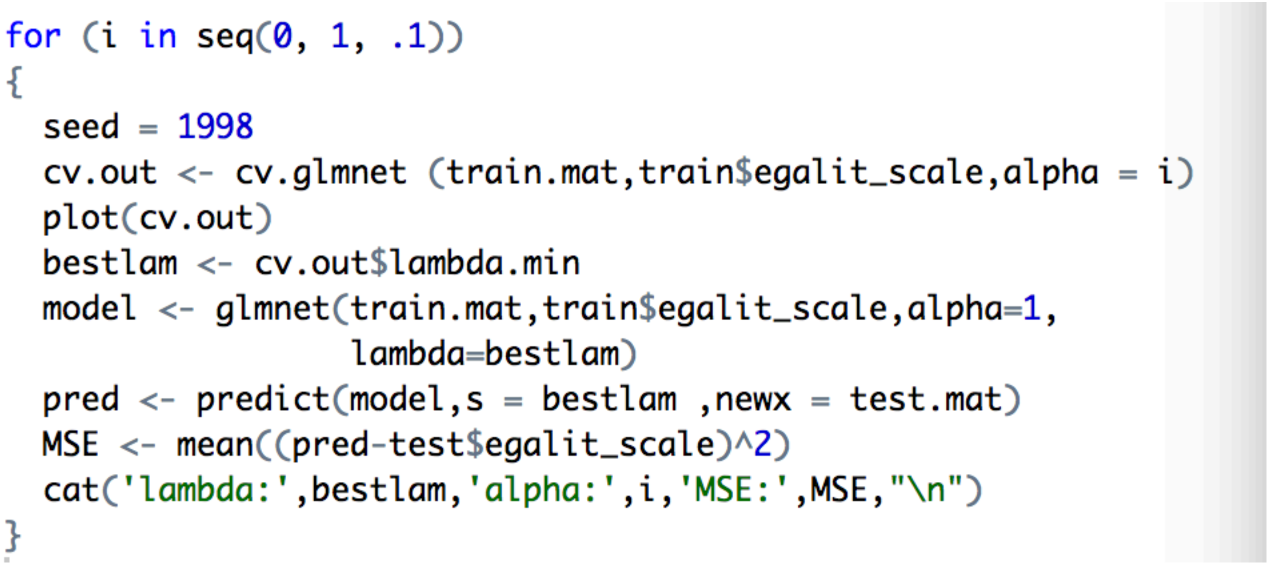
The MSE value for this regression model is **61.22341**.

The out put shows **28** non-zero coefficients (exclude the intercept). They are:

(Intercept) 30.58973208
age -0.04126914
black 0.91512414
childs 0.17755861
colhomo 0.15439475

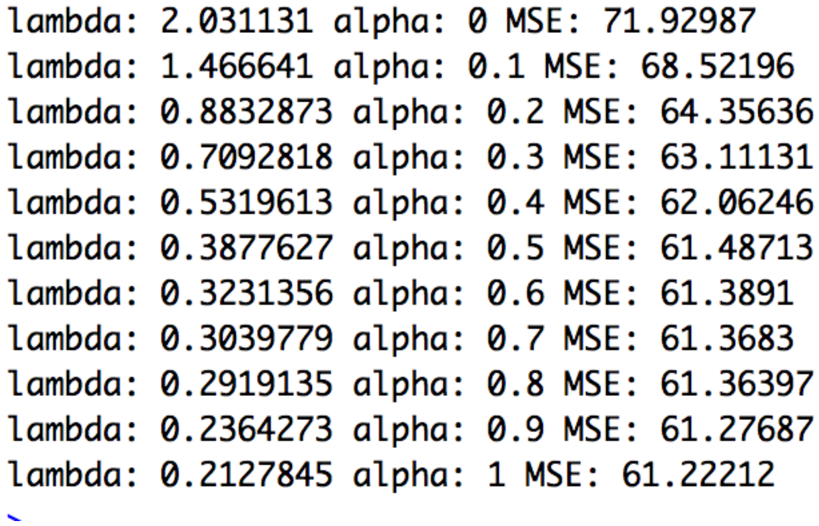
grass -1.35079986
happy 0.23987199
income06 -0.11076466
owngun 0.73225195
polviews -1.55739600
pres08 -4.27741977
science_quiz -0.06304310
sex 0.98560092
sibs 0.10097098
tolerance -0.20203266
tvhours 0.18426808
vetyears -0.08462341
degree_Junior.Coll -0.46922672
degree_Bachelor.deg -1.69215333
marital_Widowed -0.16406883
news_NEVER 0.14016794
partyid_3_Ind -1.02240341
partyid_3_Rep -2.59705850
relig_HINDUISM -1.75593400
relig_ORTHODOX.CHRISTIAN -0.10787052
spend3_Mod 0.13511572
spend3_Liberal 1.28376737
zodiac_VIRGO 0.40586995
zodiac_SCORPIO -0.22452741
zodiac_AQUARIUS 0.23106647

(4) Elastic net regression model



```
for (i in seq(0, 1, .1))
{
  seed = 1998
  cv.out <- cv.glmnet (train.mat,train$egalit_scale,alpha = i)
  plot(cv.out)
  bestlam <- cv.out$lambda.min
  model <- glmnet(train.mat,train$egalit_scale,alpha=1,
                  lambda=bestlam)
  pred <- predict(model,s = bestlam ,newx = test.mat)
  MSE <- mean((pred-test$egalit_scale)^2)
  cat('lambda:',bestlam,'alpha:',i,'MSE:',MSE,"\n")
}
```

Below is the output:



```
lambda: 2.031131 alpha: 0 MSE: 71.92987
lambda: 1.466641 alpha: 0.1 MSE: 68.52196
lambda: 0.8832873 alpha: 0.2 MSE: 64.35636
lambda: 0.7092818 alpha: 0.3 MSE: 63.11131
lambda: 0.5319613 alpha: 0.4 MSE: 62.06246
lambda: 0.3877627 alpha: 0.5 MSE: 61.48713
lambda: 0.3231356 alpha: 0.6 MSE: 61.3891
lambda: 0.3039779 alpha: 0.7 MSE: 61.3683
lambda: 0.2919135 alpha: 0.8 MSE: 61.36397
lambda: 0.2364273 alpha: 0.9 MSE: 61.27687
lambda: 0.2127845 alpha: 1 MSE: 61.22212
```

The best combination is $\alpha = \mathbf{1.0}$ and $\lambda = \mathbf{0.2127845}$

The MSE value for this regression model is **61.22212**.

The out put shows **28** non-zero coefficients (exclude the intercept). They are:

(Intercept) 30.58527416
age -0.03979472
black 0.94109267
childs 0.16003046
colhomo 0.08072144
grass -1.28294232
happy 0.21835366
income06 -0.11137199
owngun 0.71695643
polviews -1.55518887
pres08 -4.31120985
science_quiz -0.05630268
sex 0.96543510
sibs 0.09764766
tolerance -0.19834190
tvhours 0.17916102
vetyears -0.07150823
degree_Junior.Coll -0.38227955
degree_Bachelor.deg -1.66026565
marital_Widowed -0.07861622
news_NEVER 0.10220439
partyid_3_Ind -0.92011865
partyid_3_Rep -2.50177021

relig_HINDUISM -1.49863982

spend3_Mod 0.07551828

spend3_Liberal 1.24670553

zodiac_VIRGO 0.34170206

zodiac_SCORPIO -0.17387974

zodiac_AQUARIUS 0.16537497

(5) Comments

Regressions models in 2(a)-(d) do not differ from each other significantly in terms of MSE. Therefore, we can say that they almost perform equally well.

From 2(c) and 2(d), we may notice several significant variables among the generated coefficients. Based on my subjective judgement, I think the following variables are the most significant for predicting an individuals egalitarianism:

income, sex, race, tolerance, political believes, birthdary and education degrees

References

Here are two webpages that help with this question.

- https://rstudio-pubs-static.s3.amazonaws.com/384357_6b71ee9df6214ed5bf76f8de9ef5aba3.html
- https://rpubs.com/kimbrown345/286189