ASSIGNMENT 2

- Msbk19

```
> ## testing
> library(glmnet)
> library(leaps)
> pc2_testing_ <- read.csv("~/Desktop/assignment2_testdata (1) (1).csv", header=TRUE)
> set.seed(20854690)
> train<- sample(1: nrow(pc2_testing_),nrow(pc2_testing_)/2)
> test<- (-train)
> x = model.matrix(LOGVALUE~., pc2_testing_)[,-1]
> y <- pc2_testing_$LOGVALUE
>
> #### performing the best subset selection (to see whether all variables should be considere
d)
> ridgefit.full <- regsubsets(LOGVALUE~.,pc2_testing_,nvmax=19)
> reg.summary<- summary(ridgefit.full)
```

Upon using the same code on Q1, I got the R^2 value as:

```
> reg.summary$rsq #### from this we can see how the r^2 statistic increases from
 [1] 0.1520176 0.2196040 0.2391299 0.2549027 0.2591834 0.2622017 0.2634006 0.2637973
 [9] 0.2641577 0.2643402 0.2644448 0.2645256 0.2645269
```

The r^2 statistic in the test set increases from 15% when 1 variable is being used to 26%when all the variables are being utilized when being compared to the 13% to 22% change I had gotten in the training set.

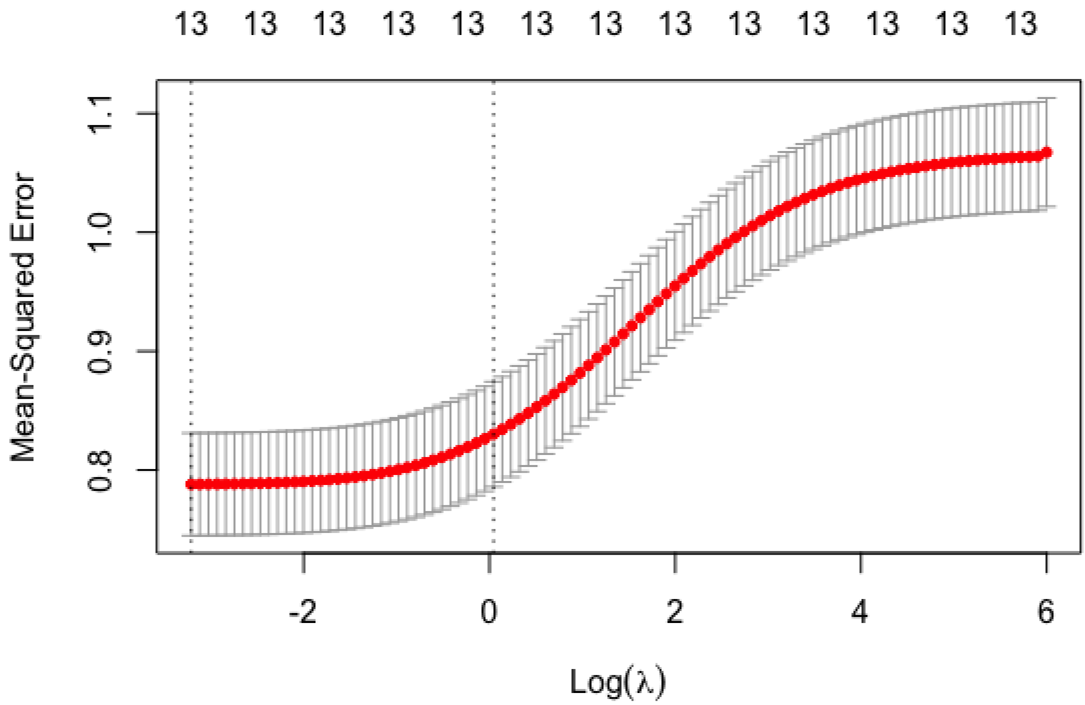
The MSE of the test training set is :

```
> View(best.fit)
> model <- lm(LOGVALUE~.,pc2_testing_)
> model_summ <-summary(model)
> mean(model_summ$residuals^2)
[1] 0.7850071
>
```

MSE value using residuals - 0.7850071

The lambda value sees a minor change from 0.0008686703 received in the training set to 0.0008835256 received in the test set.

```
> pred = predict(lasso.tr,x[-train,])
> rmse = sqrt(apply((y[-train]-pred)^2,2,mean))
> plot(log(lasso.tr$lambda),rmse,type="b",xlab = "log (lambda)")
> lam.best = lasso.tr$lambda[order(rmse)[1]] #### best value of lambda(training data) : [1]
 0.0008686703
> lam.best ## lambda
[1] 0.0008835256
```

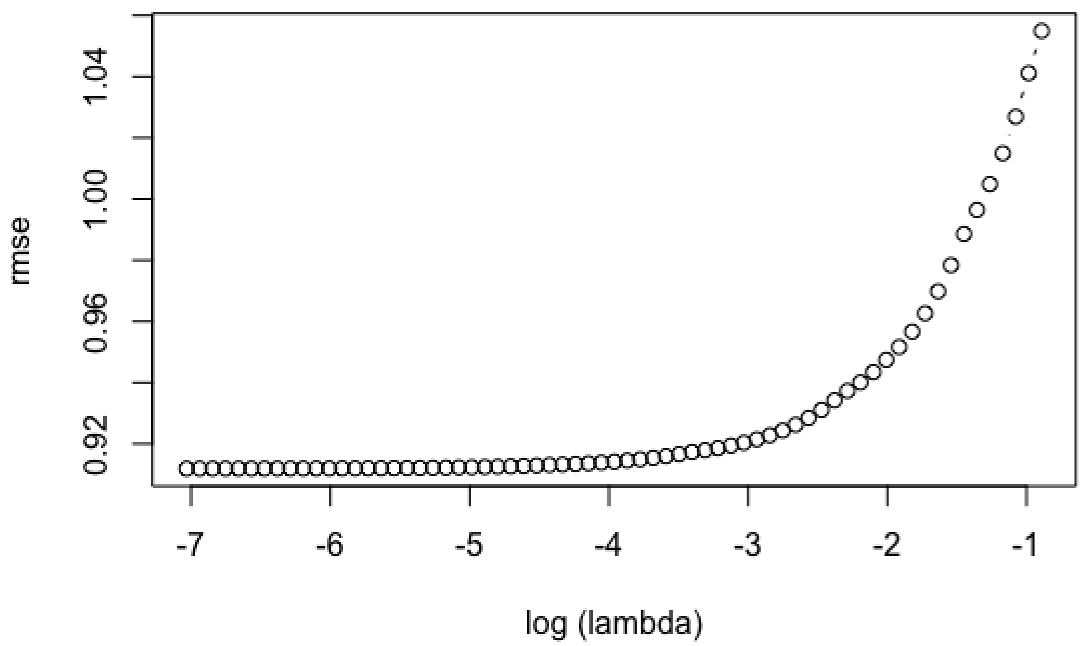


This is synonymous with the graph received in the training set and the lambda value when we perform a cross validation mean squared error of all the variables.

We receive nearly the same values as when the LASSO is compared between the training and test set.

```
> ##### Lasso
> fit.lasso = glmnet(x,y,alpha=1)
> plot(fit.lasso,xvar= "lambda", label = TRUE)
> plot(fit.lasso,xvar= "dev", label = TRUE)
> cv.lasso = cv.glmnet(x,y)
> plot(cv.lasso)
> coef(cv.lasso)
14 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) 1.051348e+01
BATHS       1.856450e-01
BEDRMS      .
BUILT       .
UNITSF      3.403625e-06
LOT         .
ROOMS       1.253176e-01
REGION      1.282508e-01
KITCHEN     .
FLOORS      1.496869e-03
LAUNDY      .
RECRM       .
METRO       .
METRO3      .
```

For Q2, we receive the same graph as the one in the training set:



Rest of the code and the values received with the algorithm and the testing set, on the next page :

All the other code values received :

```
> ## testing
> library(glmnet)
> library(leaps)
> pc2_testing_ <- read.csv("~/Desktop/assignment2_testdata (1) (1).csv", header=TRUE)
> set.seed(20854690)
> train<- sample(1: nrow(pc2_testing_),nrow(pc2_testing_)/2)
> test<- (-train)
> x = model.matrix(LOGVALUE~., pc2_testing_)[,-1]
> y <- pc2_testing_$LOGVALUE
>
> #### performing the best subset selection (to see whether all variables should be considere
d)
> ridgefit.full <- regsubsets(LOGVALUE~.,pc2_testing_,nvmax=19)
> reg.summary<- summary(ridgefit.full)
> reg.summary$rsq #### from this we can see how the r^2 statistic increases from
 [1] 0.1520176 0.2196040 0.2391299 0.2549027 0.2591834 0.2622017 0.2634006 0.2637973
 [9] 0.2641577 0.2643402 0.2644448 0.2645256 0.2645269
> ### 13% when only 1 variable is included in the model vs 22% when all are included.
>
> ##### ridge regression :::
> fit.ridge = glmnet(x,y,alpha=0)
> plot(fit.ridge,xvar = "lambda", label= TRUE)
> cv.ridge = cv.glmnet(x,y,alpha=0)
> plot.cv = plot(cv.ridge) ## plot shows how well the model works when all the variables are
 together
> bestlam <- cv.ridge$lambda.min
>
>
> ##### Lasso
> fit.lasso = glmnet(x,y,alpha=1)
> plot(fit.lasso,xvar= "lambda", label = TRUE)
> plot(fit.lasso,xvar= "dev", label = TRUE)
> cv.lasso = cv.glmnet(x,y)
> plot(cv.lasso)
> coef(cv.lasso)
14 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) 1.051348e+01
```


```
> ##### Lasso
> fit.lasso = glmnet(x,y,alpha=1)
> plot(fit.lasso,xvar= "lambda", label = TRUE)
> plot(fit.lasso,xvar= "dev", label = TRUE)
> cv.lasso = cv.glmnet(x,y)
> plot(cv.lasso)
> coef(cv.lasso)
14 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) 1.051348e+01
```


```
BATHS       1.856450e-01
BEDRMS      .
BUILT       .
UNITSF      3.403625e-06
LOT         .
ROOMS       1.253176e-01
REGION      1.282508e-01
KITCHEN     .
FLOORS      1.496869e-03
LAUNDY      .
RECRM       .
METRO       .
METRO3      .
>
> ### using training to gain a "lambda" value for the LASSO.
> lasso.tr = glmnet(x[train,],y[train])
> lasso.tr
```

```
> ### using training to gain a "lambda" value for the LASSO.
> lasso.tr = glmnet(x[train,],y[train])
> lasso.tr

Call:  glmnet(x = x[train, ], y = y[train])

   Df  %Dev  Lambda
1   0  0.00 0.41010
2   2  3.00 0.37370
3   2  5.94 0.34050
4   2  8.39 0.31020
5   2 10.42 0.28270
6   2 12.10 0.25760
7   3 13.65 0.23470
8   3 15.58 0.21380
9   3 17.19 0.19480
10  3 18.53 0.17750
11  3 19.64 0.16180
12  3 20.56 0.14740
13  3 21.32 0.13430
14  4 22.06 0.12240
15  4 22.68 0.11150
16  4 23.19 0.10160
17  5 23.73 0.09256
18  5 24.26 0.08434
19  5 24.70 0.07684
20  5 25.07 0.07002
21  5 25.37 0.06380
22  5 25.62 0.05813
23  5 25.83 0.05297
24  5 26.01 0.04826
25  5 26.15 0.04397
> pred = predict(lasso.tr,x[-train,])
> rmse = sqrt(apply((y[-train]-pred)^2,2,mean))
> plot(log(lasso.tr$lambda),rmse,type="b",xlab = "log (lambda)")
> lam.best = lasso.tr$lambda[order(rmse)[1]] #### best value of lambda(training data) : [1]
 0.0008686703
> lam.best ## lambda
[1] 0.0008835256
> coef(lasso.tr,s=lam.best)
14 x 1 sparse Matrix of class "dgCMatrix"
                       s1
(Intercept)  1.392156e+01
BATHS        2.044467e-01
BEDRMS       5.859686e-03
BUILT       -2.236950e-03
UNITSF       5.148544e-05
LOT          9.760204e-08
ROOMS        1.614951e-01
REGION       2.799443e-01
KITCHEN     -7.903225e-02
FLOORS       1.085617e-01
LAUNDRY     -2.450334e-02
RECRM       -1.310507e-01
METRO        1.326095e-02
METRO3      -1.045851e-02
>
>
>
```

```
> View(best.fit)
> model <- lm(LOGVALUE~.,pc2_testing_)
> model_summ <-summary(model)
> mean(model_summ$residuals^2)
[1] 0.7850071
```