# Statistical Learning/Lab3

*Sascha Strobl*

*5/16/2018*

Load dataset "CompanyBill" and remove missing data:

```
setwd("~/Documents/CGUClasses/ZOLDClasses/Statistical Learning/Lab 3 Apr 3");

CompanyBill = read.table("CompanyBill.txt", header = T);

library (ISLR);

CompanyBill = CompanyBill*1;
CompanyBill=CompanyBill[complete.cases(CompanyBill), ]
```

Show dimension of dataset:

```
dim(CompanyBill)
```

```
## [1] 7186    7
```

Find the best subset for this dataset after renaming the columns:

```
colnames(CompanyBill) <- c("V1","V2","V3","V4","V5","V6","V7")
library(leaps);
regfit.full=regsubsets(CompanyBill$V1~.,CompanyBill);
summary(regfit.full);
```

```
## Subset selection object
## Call: regsubsets.formula(CompanyBill$V1 ~ ., CompanyBill)
## 6 Variables  (and intercept)
##    Forced in Forced out
## V2     FALSE      FALSE
## V3     FALSE      FALSE
## V4     FALSE      FALSE
## V5     FALSE      FALSE
## V6     FALSE      FALSE
## V7     FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: exhaustive
##          V2  V3  V4  V5  V6  V7
## 1  ( 1 ) " " " " " " " " "*" " "
## 2  ( 1 ) " " " " "*" " " "*" " "
## 3  ( 1 ) " " " " "*" "*" "*" " "
## 4  ( 1 ) " " "*" "*" "*" "*" " "
## 5  ( 1 ) "*" "*" "*" "*" "*" " "
## 6  ( 1 ) "*" "*" "*" "*" "*" "*"
```

Use the forward stepwise selection method:

```
regfit.fwd=regsubsets (CompanyBill$V1~.,CompanyBill, method="forward");
reg.summary=summary(regfit.fwd);
```

Plot the Cp, BIC, AIC and Adj.R2 measures in relation to the number of variables:

```
par(mfrow =c(2,2));
plot(reg.summary$adjr2 ,xlab =" Number of Variables ", ylab="Adjusted RSq",type="l");
which.max(reg.summary$adjr2);
```

```
## [1] 6
```

```
points (6, reg.summary$adjr2[6], col ="red",cex =2, pch =20);
plot(reg.summary$rsq ,xlab =" Number of Variables ", ylab="RSq",type="l");
which.max(reg.summary$rsq);
```

```
## [1] 6
```

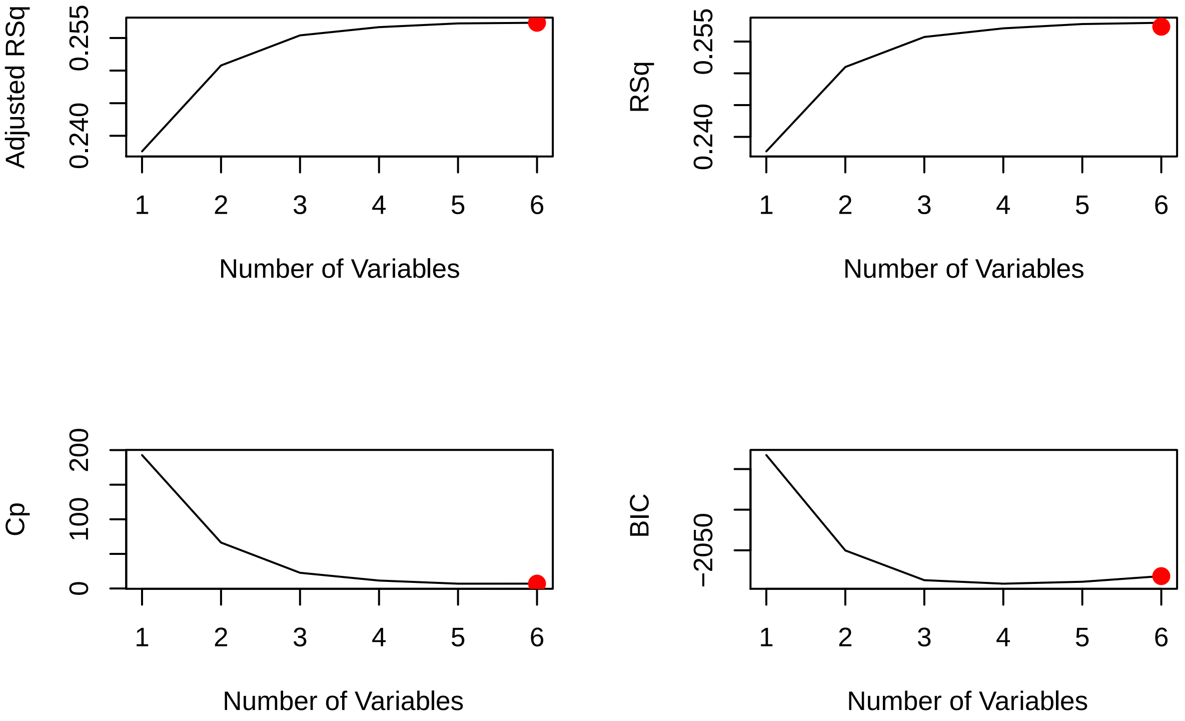
```
points (6, reg.summary$adjr2[6], col ="red",cex =2, pch =20);
plot(reg.summary$cp ,xlab =" Number of Variables",ylab="Cp", type="l");
which.min(reg.summary$cp);
```

```
## [1] 5
```

```
points (6, reg.summary$cp [6], col ="red",cex =2, pch=20);
plot(reg.summary$bic ,xlab=" Number of Variables",ylab=" BIC", type="l");
which.min(reg.summary$bic);
```

```
## [1] 4
```

```
points(6, reg.summary$bic[6], col ="red",cex=2,pch=20);
```

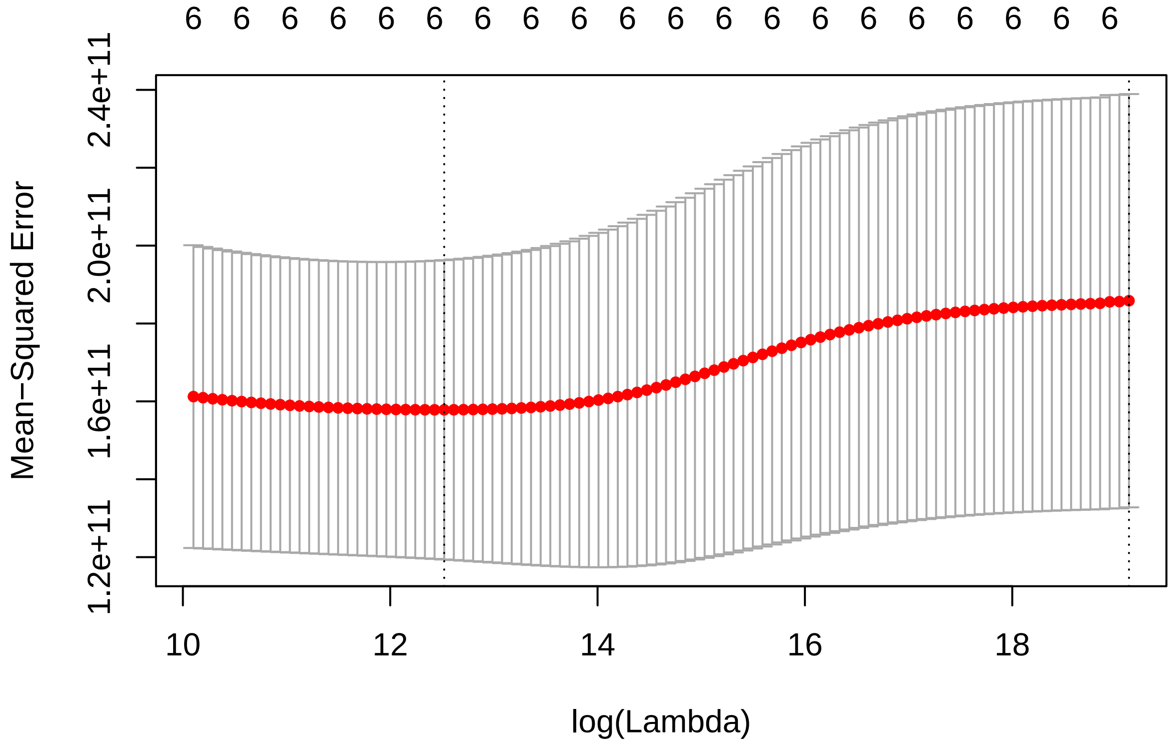


Run a Ridge Regression, tune the lambda hyperparameter:

```
library(Matrix);
x=model.matrix (V1~.,CompanyBill )[,-1];
y=CompanyBill$V1;
library(glmnet);
```

```
## Loading required package: foreach
## Loaded glmnet 2.0-13
```

```
set.seed(1);
train = sample(1 : nrow(x), nrow(x)/2);
test=(- train );
y.test=y[test];
cv.out =cv.glmnet (x[train ,],y[train],alpha =0);
plot(cv.out);
```



```
bestlam1=cv.out$lambda.min;
bestlam1;
```

```
## [1] 273949
```

Output the regression result with the optimal lambda:

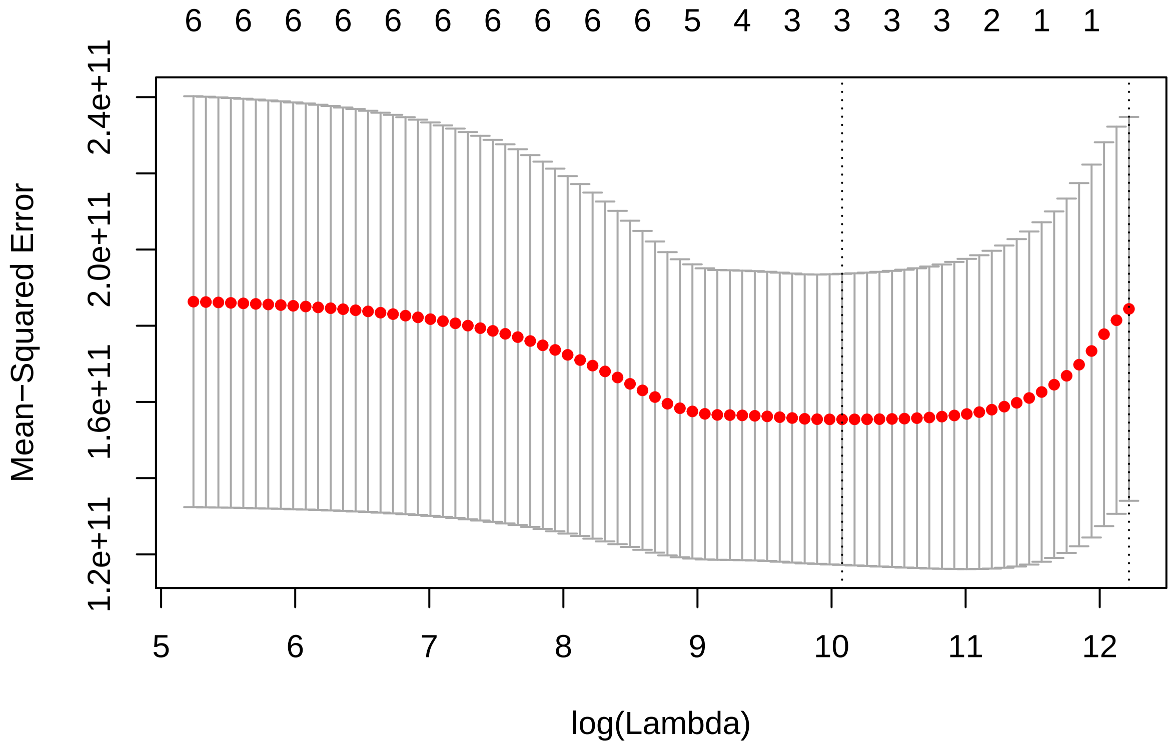
```
ridge.mod =glmnet (x,y,alpha =0, lambda =bestlam1);
coef(ridge.mod, s=bestlam1)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept)  9.094006e+04
## V2           2.249989e-03
## V3           6.307917e-02
## V4           2.599610e-02
## V5           6.859095e-02
## V6           7.531287e-02
## V7          -2.817214e+02
```

Redo for the Lasso:

```
set.seed (1);
cv.out =cv.glmnet (x[train ,],y[train],alpha =1);
```

```
plot(cv.out);
```



```
bestlam2 =cv.out$lambda.min;
bestlam2;
```

```
## [1] 23826.69
```

```
lasso.mod =glmnet (x,y,alpha =1, lambda =bestlam2);
coef(lasso.mod, s=bestlam2)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) 8.167587e+04
## V2          .
## V3          .
## V4          .
## V5          6.117684e-03
## V6          2.448676e-01
## V7          .
```

Cross validation for choosing lambda; calculating the test MSE.

```
ridge.pred=predict (ridge.mod ,s=bestlam1, newx=x[test,]);
mean(( ridge.pred -y.test)^2);
```

```
## [1] 95566315387
```

```
lasso.pred=predict (lasso.mod ,s=bestlam2 ,newx=x[test ,]);
mean(( lasso.pred -y.test)^2);
```

```
## [1] 93736326673
```

Conclusion: Lasso better