Untitled

```
set.seed(1)
x = rnorm(100)
noise = rnorm(100)
y=10+3*x+4*x^2+5*x^3+noise
newdata=data.frame(cbind(x,y))
```

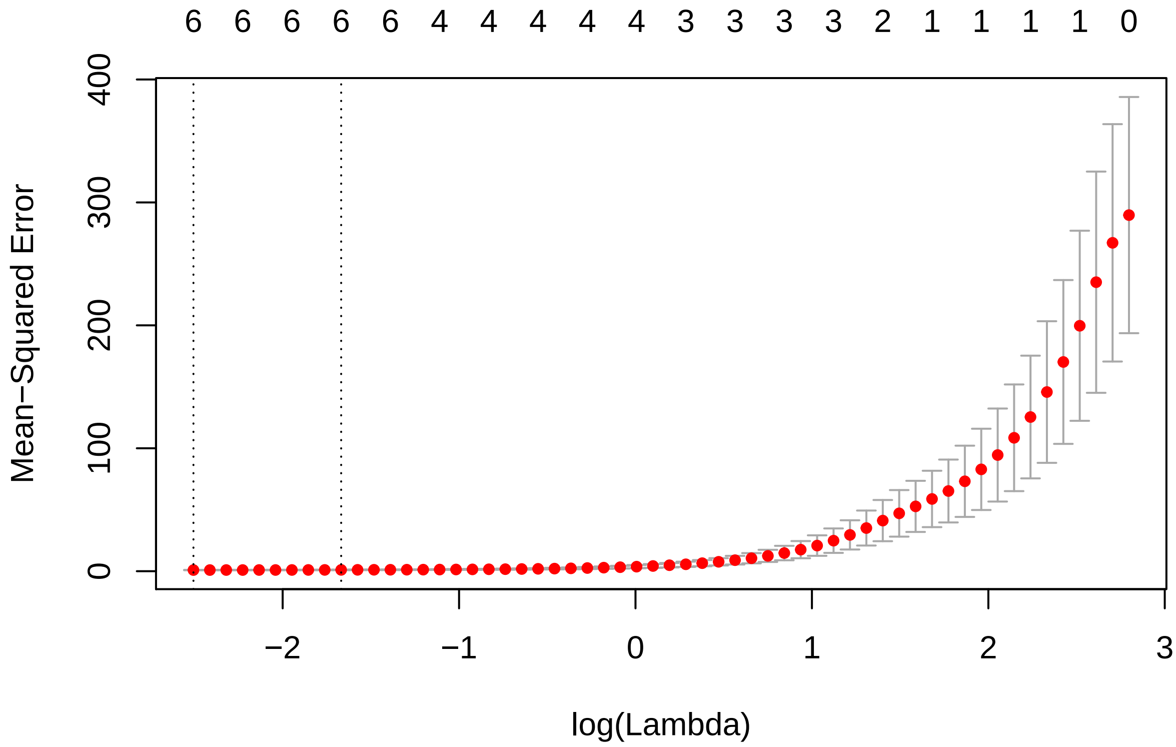
```
#(e)
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-18
```

```
mol.data=model.matrix(y~poly(x,10,raw=T),data=newdata)[,-1]
grid = 10^seq(10,-2,length=100)
lasso.mod=glmnet(mol.data,y,alpha = 1,lambda = grid,thresh = 1e-12)
set.seed(1)
cv.lasso = cv.glmnet(mol.data, y, alpha = 1)
plot(cv.lasso)
```



```
bestlam.lasso = cv.lasso$lambda.min
bestlam.lasso
```

```
## [1] 0.08161757
```

```
lasso.coef=predict(lasso.mod, s = bestlam.lasso, type = "coefficients")
lasso.coef
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                  1
## (Intercept)            10.18241209
## poly(x, 10, raw = T)1   3.21256596
## poly(x, 10, raw = T)2   3.62800230
## poly(x, 10, raw = T)3   4.69515841
## poly(x, 10, raw = T)4   0.04061086
## poly(x, 10, raw = T)5   0.05255439
## poly(x, 10, raw = T)6   .
## poly(x, 10, raw = T)7   .
## poly(x, 10, raw = T)8   .
## poly(x, 10, raw = T)9   .
## poly(x, 10, raw = T)10  .
```

According to the lasso, we can see that the coefficients for x1,x2,x3 is 3.1,3.6,4.8 respectively, and for other predictors, including x4,x5,x7, the coeffivients are 0.04,0.009,0.0037 repectively, which is close to 0. Therefore, the result of lasso is pretty close to the original equation. It shows that the predictor x1,x2,x3 have big coeffients, which are also close to the original coefficients 3,4,5.

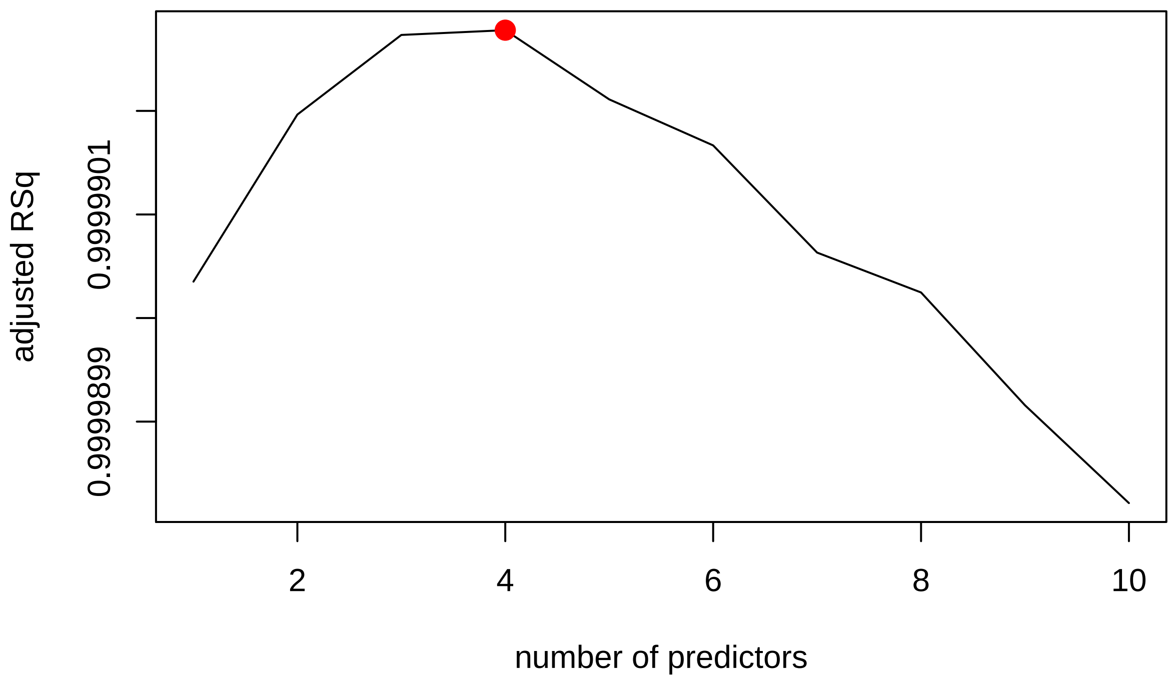
```r
#(f)
set.seed(1)
x = rnorm(100)
noise = rnorm(100)
y = 30 + 5*(x^7) + noise
newdata2=data.frame(cbind(y,poly(x,10,raw=T)))
```

```r
#using best sbubset selection
library(leaps)
regfit.best=regsubsets(y~.,data = newdata2, nvmax = 10)
reg.summary = summary(regfit.best)

#check with adjusted R square
which.max(reg.summary$adjr2)
```

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

```r
plot(reg.summary$adjr2,xlab="number of predictors",ylab="adjusted RSq",type="l")
points(4,reg.summary$adjr2[4],col="red",cex=2,pch=20)
```



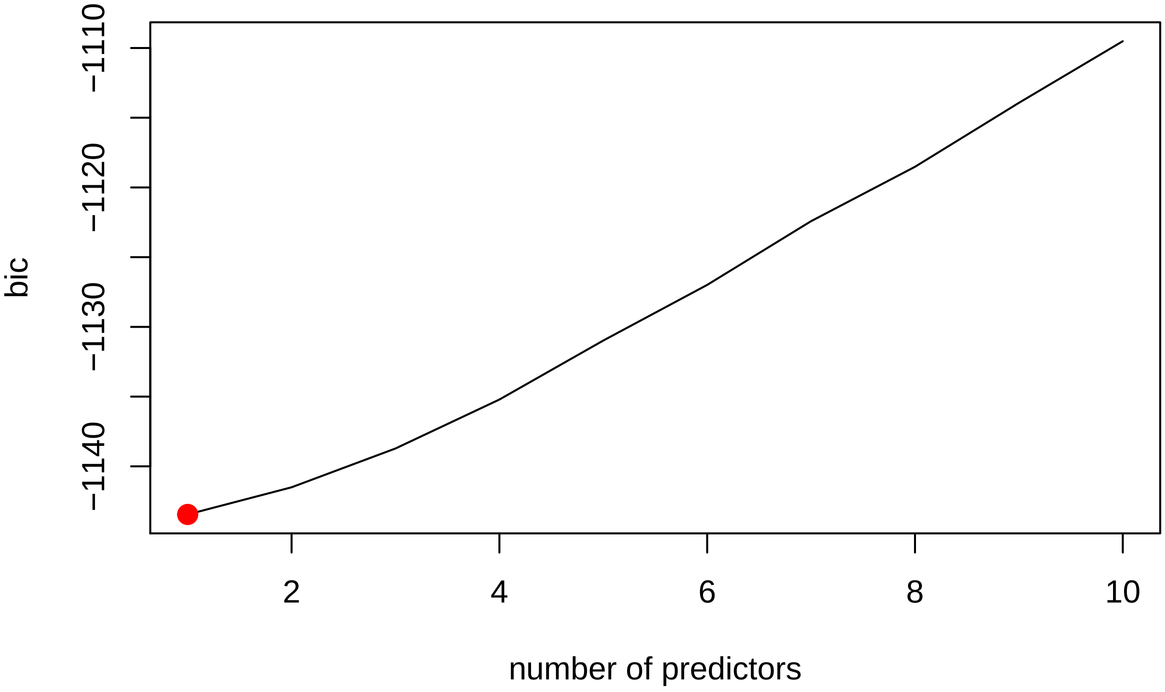
```
coef(regfit.best,id=4)
```

```
## (Intercept)          X1          X2          X3          X7
##  30.0762524   0.2914016  -0.1617671  -0.2526527   5.0091338
```

```
#check with bic
which.min(reg.summary$bic)
```

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

```
plot(reg.summary$bic,xlab="number of predictors",ylab="bic",type="l")
points(1,reg.summary$bic[1],col="red",cex=2,pch=20)
```



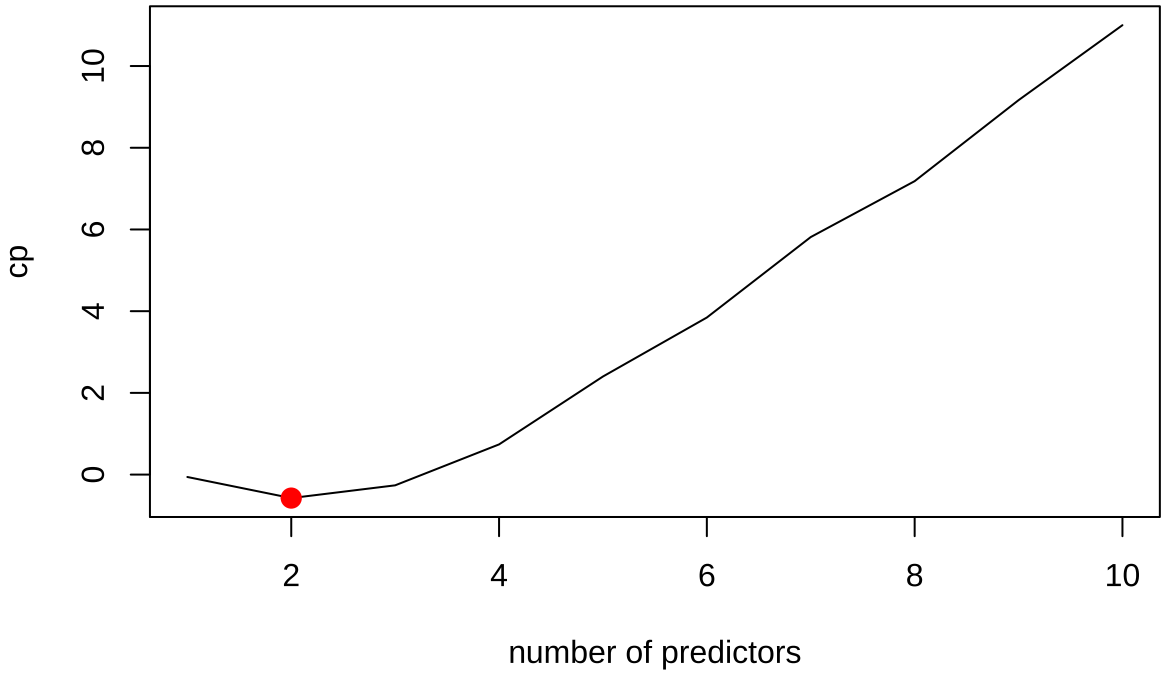
```
coef(regfit.best,id=1)
```

```
## (Intercept)          X7
##    29.95894     5.00077
```

```
#check with cp
which.min(reg.summary$cp)
```

```
## [1] 2
```

```
plot(reg.summary$cp,xlab="number of predictors",ylab="cp",type="l")
points(2,reg.summary$cp[2],col="red",cex=2,pch=20)
```



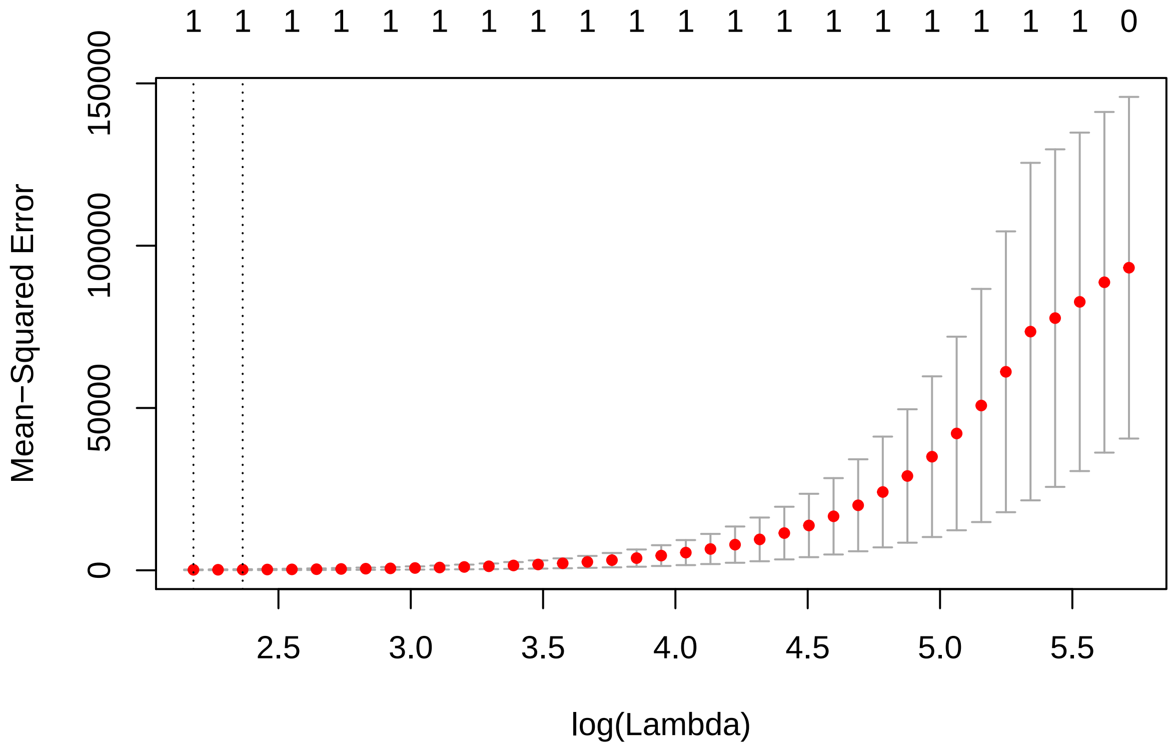
```
coef(regfit.best,id=2)
```

```
## (Intercept)          X2          X7 
##  30.0704904  -0.1417084   5.0015552
```

In this chunk, I use best subset selection. BIC,CP and adjusted R square give different results. And BIC's result is closest to the original equation. The coefficient is 5.00077, and intercept is 29.95, which are close to 5 and 30 in the equation. The other two also give pretty close results. They have more than one predictors in the final report, although the coefficients of those redundant predictors are very close to 0.

```
#using lasso
mol.data=model.matrix(y~poly(x,10,raw=T),data=newdata2)[,-1]
grid = 10^seq(10,-2,length=100)
lasso.mod=glmnet(mol.data,y,alpha = 1,lambda = grid,thresh = 1e-12)

set.seed(1)
cv.lasso = cv.glmnet(mol.data, y, alpha = 1)
plot(cv.lasso)
```



```
bestlam.lasso = cv.lasso$lambda.min
bestlam.lasso
```

```
## [1] 8.835273
```

```
lasso.coef=predict(lasso.mod,s = bestlam.lasso, type = "coefficients")
lasso.coef
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                1
## (Intercept)            30.574163
## poly(x, 10, raw = T)1   .
## poly(x, 10, raw = T)2   .
## poly(x, 10, raw = T)3   .
## poly(x, 10, raw = T)4   .
## poly(x, 10, raw = T)5   .
## poly(x, 10, raw = T)6   .
## poly(x, 10, raw = T)7   4.854995
## poly(x, 10, raw = T)8   .
## poly(x, 10, raw = T)9   .
## poly(x, 10, raw = T)10  .
```

According to the lasso, the results are close to original equation. The coefficient in lasso is 4.85, which is very close to 5. And intercept in lasso is 30.5, which is close to 30.

“‘