HW4 Solution CS6220-Data Mining

2. JWHT problem 8, p. 262 ; In (c), only use R2 and BIC, Skip (e), Skip (f)

(a)

```
set.seed(321)
X = rnorm(100)
eps = rnorm(100)
```

(b)

```
Y <- 3 + 4*X + 1.9 * I(X^2) + 0.7 * I(X^3) + eps
```

(c)

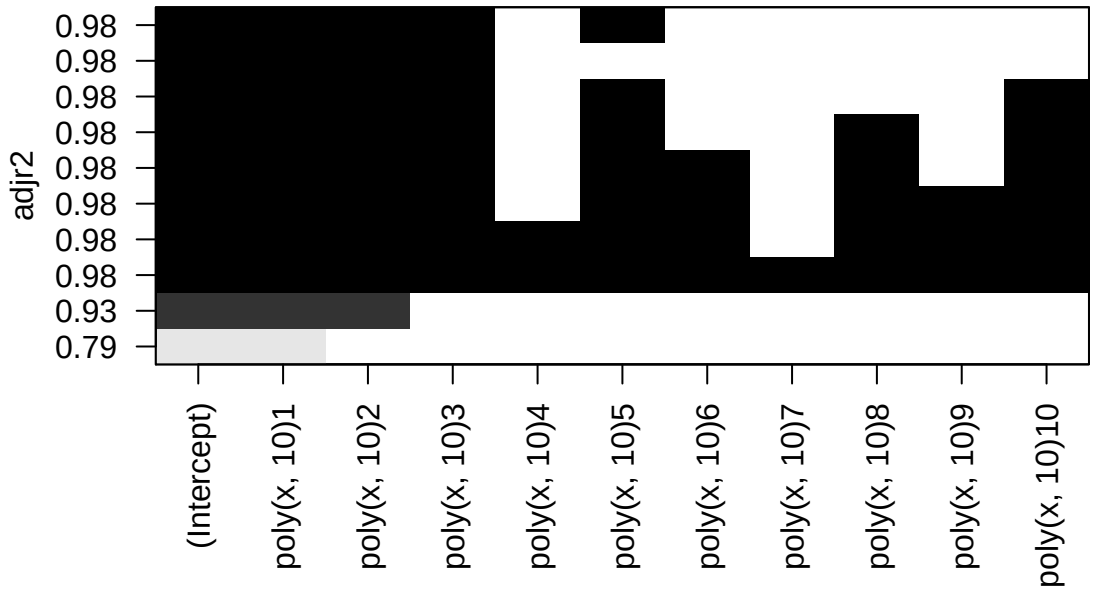
```
#Create the dataset
mydata <- data.frame(x = X, y = Y)

library(leaps)
myset <- regsubsets(y~poly(x, 10), data = mydata, nvmax = 10)
mysummary <- summary(myset)

#The model with the maximum value of Adjusted R-Square
myadjr2 <- which.max(mysummary$adjr2)
myadjr2
```

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

```
#Coefficients
coef(myset, myadjr2)
```

```
##  (Intercept) poly(x, 10)1 poly(x, 10)2 poly(x, 10)3 poly(x, 10)5
##    4.6923977   55.9925942   23.7598866   13.2193634    0.9247187
```

```
plot(myset, scale="adjr2")
```



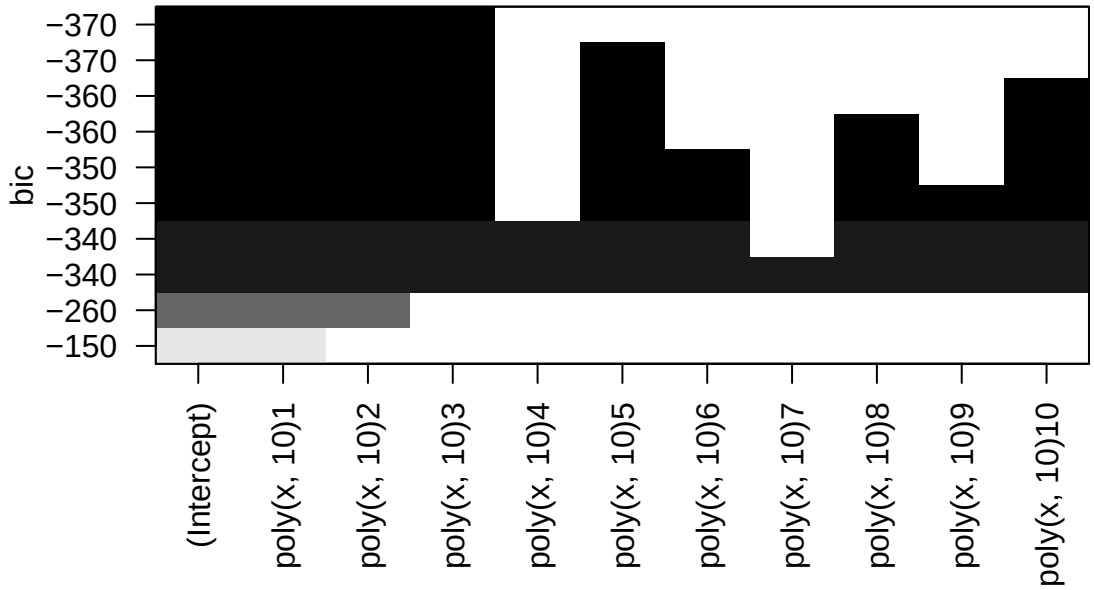

```
#The model with the least value of BIC
mybic <- which.min(mysummary$bic)
mybic
```

```
## [1] 3
```

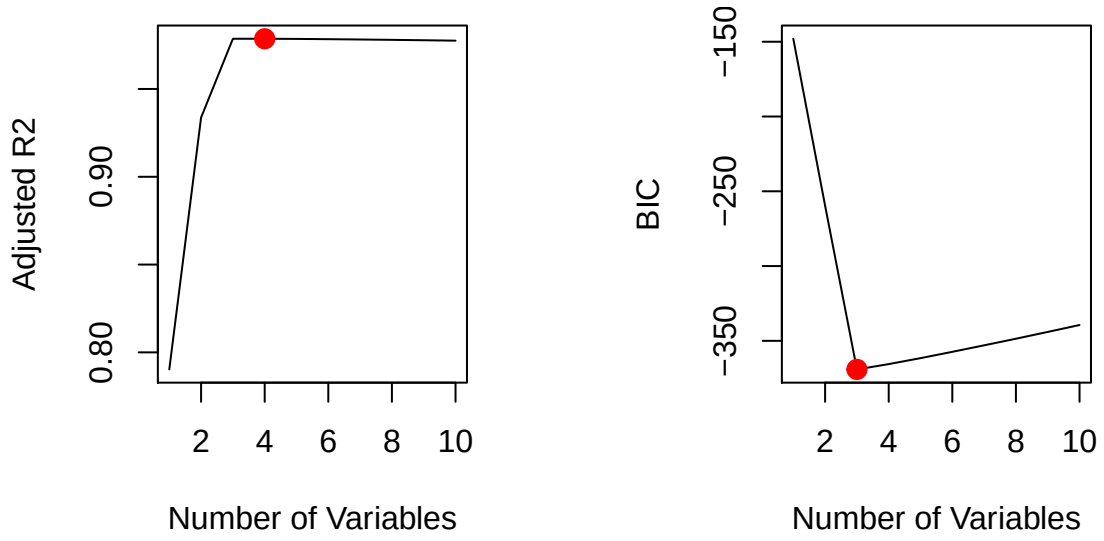
```
#Coefficients
coef(myset, mybic)
```

```
##  (Intercept) poly(x, 10)1 poly(x, 10)2 poly(x, 10)3
##     4.692398    55.992594    23.759887    13.219363
```

```
plot(myset, scale="bic")
```



```
par(mfrow=c(1, 2))
plot(mysummary$adjr2 ,xlab="Number of Variables",ylab="Adjusted R2", type="l")
points(myadjr2, mysummary$adjr2[myadjr2], col="red",cex=2,pch=20)
plot(mysummary$bic ,xlab="Number of Variables",ylab="BIC", type="l")
points(mybic, mysummary$bic[mybic], col="red",cex=2,pch=20)
```



The problem illustrates the choice of predictors from a larger subset. We had 10 predictors $(X, X^2, \ldots, X^{10})$, but only the first 3 (X, X^2, X^3) explain the variation in Y. BIC performed better than $adjR^2$ in selecting the right number of predictors.

(d)

Forward stepwise selection

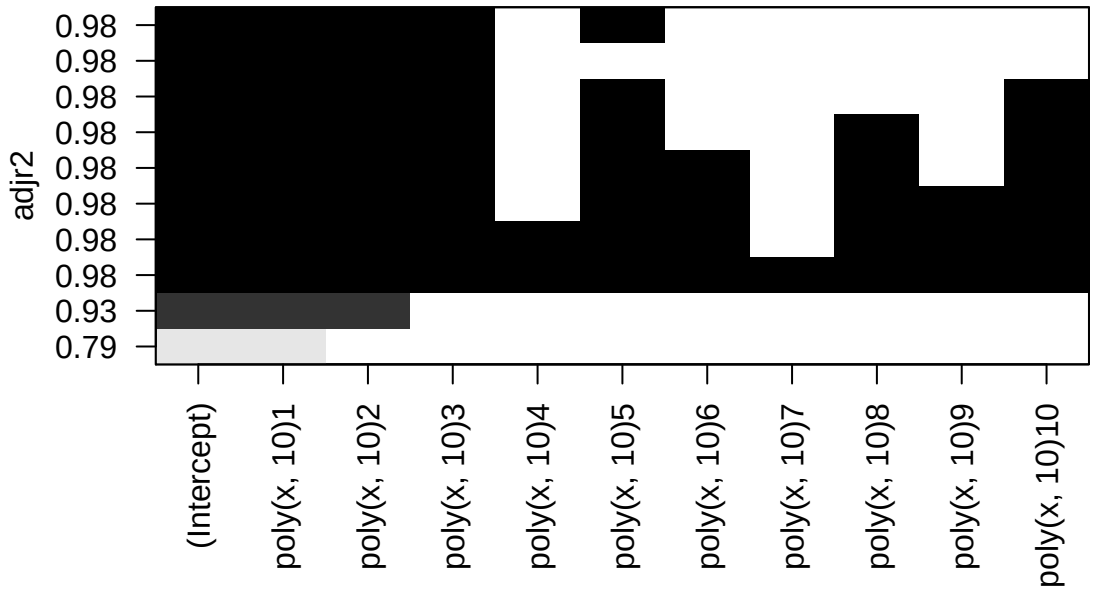
```
##Forward stepwise selection##
fwdset <- regsubsets(y~poly(x, 10), data = mydata, nvmax = 10, method="forward")
fwdsummary <- summary(fwdset)

#Adjusted R2
fwdadjr2 <- which.max(fwdsummary$adjr2)
# Number of predictors
fwdadjr2
```

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

```
#Coefficients
coef(fwdset, fwdadjr2)
```

```
##  (Intercept) poly(x, 10)1 poly(x, 10)2 poly(x, 10)3 poly(x, 10)5
##    4.6923977   55.9925942   23.7598866   13.2193634    0.9247187
```

```
plot(fwdset, scale="adjr2")
```



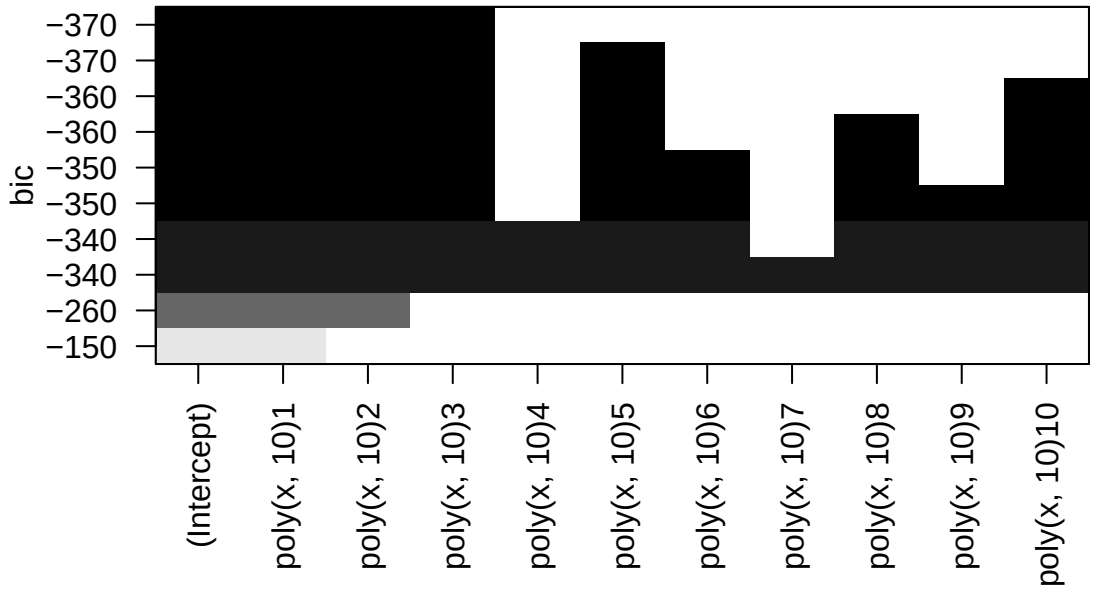

```
#BIC
fwdbic <- which.min(fwdsummary$bic)
# Number of predictors
fwdbic
```

```
## [1] 3
```

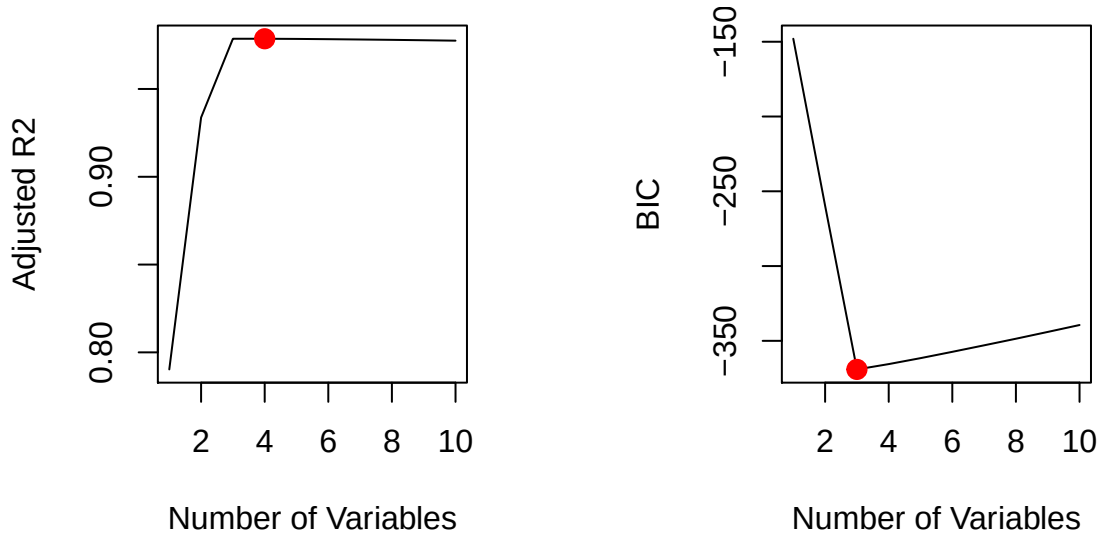
```
#Coefficients
coef(fwdset, fwdbic)
```

```
##  (Intercept) poly(x, 10)1 poly(x, 10)2 poly(x, 10)3
##     4.692398    55.992594    23.759887    13.219363
```

```
plot(fwdset, scale="bic")
```



```
par(mfrow=c(1, 2))
plot(fwdsummary$adjr2 ,xlab="Number of Variables",ylab="Adjusted R2", type="l")
points(fwdadjr2, fwdsummary$adjr2[fwdadjr2], col="red",cex=2,pch=20)
plot(fwdsummary$bic ,xlab="Number of Variables",ylab="BIC", type="l")
points(fwdbic, fwdsummary$bic[fwdbic], col="red",cex=2,pch=20)
```



Backward stepwise selection

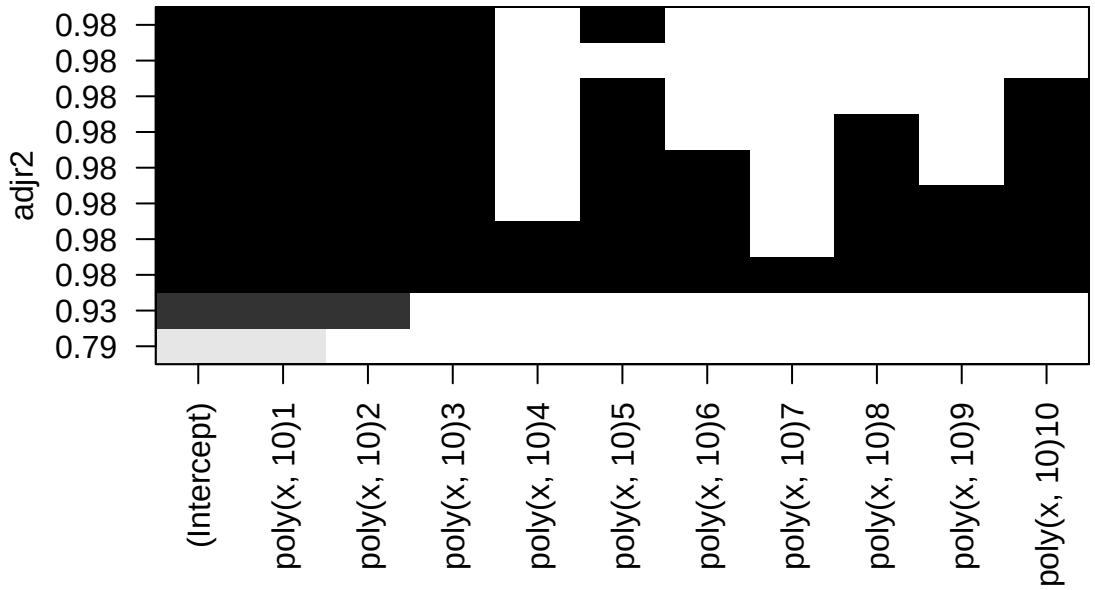
```
bwdset <- regsubsets(y~poly(x, 10), data = mydata, nvmax = 10, method="backward")
bwdsummary <- summary(bwdset)

#Adjusted R2
bwdadjr2 <- which.max(bwdsummary$adjr2)
# Number of predictors
bwdadjr2
```

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

```
#Coefficients
coef(bwdset, bwdadjr2)
```

```
##  (Intercept) poly(x, 10)1 poly(x, 10)2 poly(x, 10)3 poly(x, 10)5
##    4.6923977   55.9925942   23.7598866   13.2193634    0.9247187
```

```
plot(bwdset, scale="adjr2", xlab="Number of Variables",ylab="Adjusted R2")
```



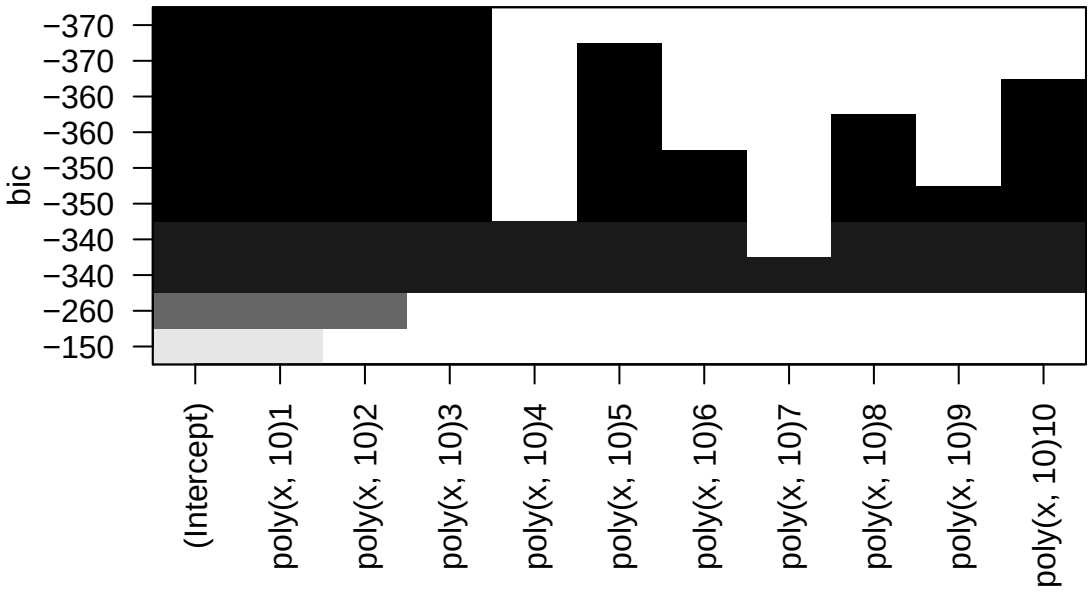

```
#BIC
bwdbic <- which.min(bwdsummary$bic)
# Number of predictors
bwdbic
```

```
## [1] 3
```

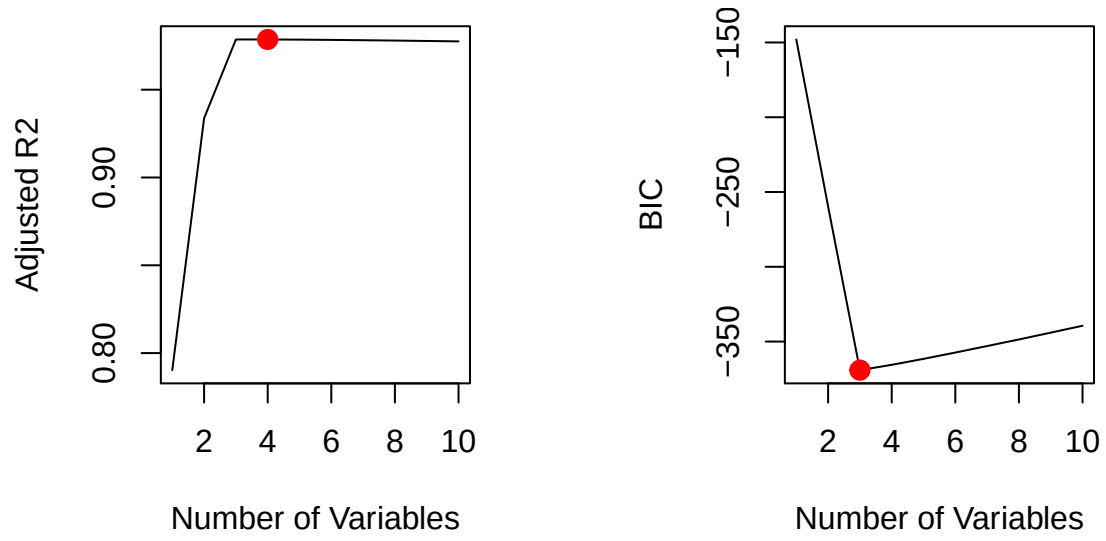
```
#Coefficients
coef(bwdset, bwdadjr2)
```

```
##  (Intercept) poly(x, 10)1 poly(x, 10)2 poly(x, 10)3 poly(x, 10)5
##    4.6923977   55.9925942   23.7598866   13.2193634    0.9247187
```

```
plot(bwdset, scale="bic",xlab="Number of Variables",ylab="BIC")
```




```
par(mfrow=c(1, 2))
plot(bwdsummary$adjr2 ,xlab="Number of Variables",ylab="Adjusted R2", type="l")
points(bwdadjr2, bwdsummary$adjr2[bwdadjr2], col="red",cex=2,pch=20)
plot(bwdsummary$bic ,xlab="Number of Variables",ylab="BIC", type="l")
points(bwdbic, bwdsummary$bic[bwdbic], col="red",cex=2,pch=20)
```



In this example, the exhaustive search, forward and backward stepwise selection don't differ a lot. For other problems the result may vary.

3. JWHT problem 11, p. 264 ; In (a), use exhaustive search and/or forward selection and backward elimination to find a sequence of candidate models.

(a)

```
library(MASS)
k = 10
set.seed(321)
folds = sample(1:k,nrow(Boston),replace=TRUE)
```

```
library(leaps)
predict.regsubsets =function (object ,newdata ,id ,...)
{ form=as.formula (object$call [[2]])
  mat=model.matrix(form ,newdata )
  coefi=coef(object ,id=id)
  xvars=names(coefi)
  mat[,xvars]%*%coefi
}
```

Exhaustive search

```
# Calculating MSE for each fold, and for each model in the exhaustive search
count = ncol(Boston) - 1
exh = matrix(NA, k, count)
for(j in 1:k){
  exh.fit = regsubsets(crim~., data = Boston[folds!=j,], nvmax=count)
  for(i in 1:count){
    pred = predict(exh.fit, Boston[folds == j,], id = i)
    exh[j, i] = mean((Boston$crim[folds == j] - pred)^2)
  }
}
# Reporting average MSE for each model in the exhaustive search
mean.exh=apply(exh, 2,mean)
mean.exh
```

```
##  [1] 41.16534 39.31654 39.64000 39.64598 39.21234 39.23664 39.22460
##  [8] 38.70249 38.24350 38.37417 38.40841 38.35474 38.48824
```

```
# Coefficients of the model that minimizes the cross-validated MSE
reg.exh=regsubsets (crim~.,data=Boston , nvmax=13)
coef(reg.exh ,which.min(mean.exh))
```

```
##   (Intercept)            zn         indus           nox           dis
##  19.124636156   0.042788127  -0.099385948 -10.466490364  -1.002597606
##           rad       ptratio         black         lstat          medv
##   0.539503547  -0.270835584  -0.008003761   0.117805932  -0.180593877
```

Forward stepwise selection

```
# Calculating MSE for each fold, and for each model in the forward stepwise search
fwd = matrix(NA,k,13, dimnames =list(NULL , paste(1:13)))

for(j in 1:k){
  fwd.fit = regsubsets(crim~., data = Boston[folds!=j,], nvmax=13, method = "forward")
  for(i in 1:13){
    pred = predict(fwd.fit, Boston[folds == j,], id = i)
    fwd[j, i] = mean((Boston$crim[folds == j] - pred)^2)
  }
}

# Reporting average MSE for each model in the forward stepwise search
mean.fwd = apply(fwd ,2,mean)
mean.fwd
```

```
##        1        2        3        4        5        6        7        8
## 41.16534 39.31654 39.86157 39.78226 39.35021 39.12909 39.04827 38.66130
##        9       10       11       12       13
## 38.31276 38.34549 38.39462 38.35474 38.48824
```

```
# Coefficients of the model that minimizes the cross-validated MSE
reg.fwd = regsubsets(crim~.,data=Boston , nvmax=13, method="forward")
coef(reg.fwd, which.min(mean.fwd))
```

```
##   (Intercept)            zn         indus           nox           dis
##  19.124636156   0.042788127  -0.099385948 -10.466490364  -1.002597606
##           rad       ptratio         black         lstat          medv
##   0.539503547  -0.270835584  -0.008003761   0.117805932  -0.180593877
```

Backward stepwise selection

```
# Calculating MSE for each fold, and for each model in the backward stepwise search
bwd = matrix(NA,k,13, dimnames =list(NULL , paste(1:13)))
for(j in 1:k){
  bwd.fit = regsubsets(crim~., data = Boston[folds!=j,], nvmax=13, method = "backward")
  for(i in 1:13){
    pred = predict(bwd.fit, Boston[folds == j,], id = i)
    bwd[j, i] = mean((Boston$crim[folds == j] - pred)^2)
  }
}
# Reporting average MSE for each model in the backward stepwise search
mean.bwd = apply(bwd ,2,mean)
mean.bwd
```

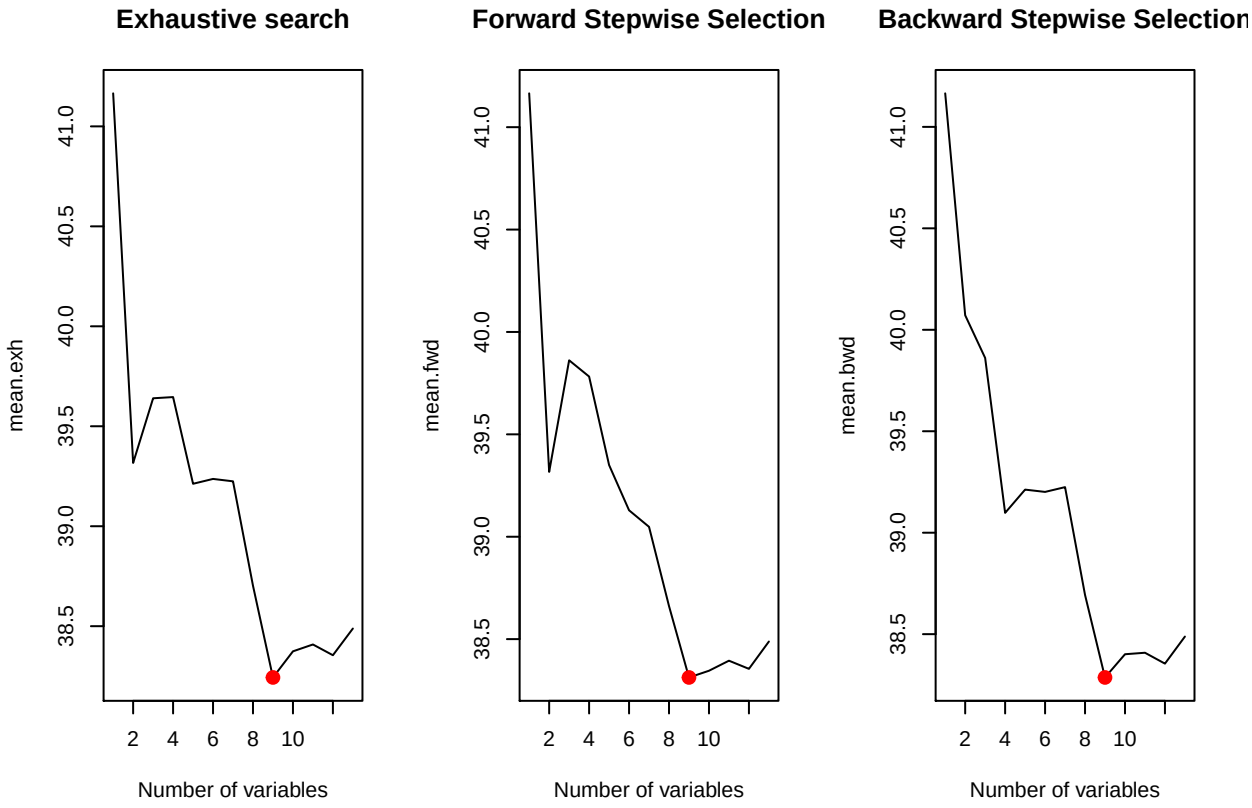
```
##        1        2        3        4        5        6        7        8
## 41.16534 40.07154 39.86116 39.09771 39.21234 39.20101 39.22460 38.69348
##        9       10       11       12       13
## 38.28661 38.40107 38.40841 38.35474 38.48824
```

```
# Coefficients of the model that minimizes the cross-validated MSE
reg.bwd = regsubsets(crim~.,data=Boston , nvmax=13, method="backward")
coef(reg.bwd, which.min(mean.bwd))
```

```
##   (Intercept)            zn         indus           nox           dis
##  19.124636156   0.042788127  -0.099385948 -10.466490364  -1.002597606
##           rad       ptratio         black         lstat          medv
##   0.539503547  -0.270835584  -0.008003761   0.117805932  -0.180593877
```

```
par(mfrow=c(1, 3))
plot(mean.exh ,type= "l", xlab='Number of variables', main = "Exhaustive search")
points(which.min(mean.exh),
mean.exh[which.min(mean.exh)],
col = "red", cex = 2, pch = 20)

plot(mean.fwd ,type= "l", xlab='Number of variables', main = "Forward Stepwise Selection")
points(which.min(mean.fwd),
```

```
mean.fwd[which.min(mean.fwd)],
col = "red", cex = 2, pch = 20)

plot(mean.bwd ,type= "l", xlab='Number of variables', main = "Backward Stepwise Selection")
points(which.min(mean.bwd),
mean.bwd[which.min(mean.bwd)],
col = "red", cex = 2, pch = 20)
```



(b & c) As we see, different approaches came to choose 9 predictors among the 13 available ones. Using more predictors doesn't improve the predictive ability of the model in this dataset.

4. Repeat Question 3, but for the surgical unit dataset given as an example in the class.

(a)

```
setwd('/Users/ovitek/Dropbox/Olga/Teaching/CS6220/Fall15/Homeworks/Hw4')
sdata <- read.table("surgical.txt")
dimnames(sdata)[[2]] <- c('blood', 'prog', 'enz', 'liver', 'age', 'female', 'modAlc', 'heavyAlc', 'surv
dim(sdata)
```

```
## [1] 54 10
```

```
head(sdata)
```

```
##   blood prog enz liver age female modAlc heavyAlc surv lsurv
## 1   6.7   62  81  2.59  50      0      1        0  695 6.544
## 2   5.1   59  66  1.70  39      0      0        0  403 5.999
```

```
## 3   7.4   57  83  2.16  55      0      0        0  710 6.565
## 4   6.5   73  41  2.01  48      0      0        0  349 5.854
## 5   7.8   65 115  4.30  45      0      0        1 2343 7.759
## 6   5.8   38  72  1.42  65      1      1        0  348 5.852
```

```
# Check for NA values
sum(is.na(sdata))
```

```
## [1] 0
```

```
library(leaps)
k = 10
set.seed(1)
folds=sample(1:k,nrow(sdata),replace=TRUE)
```

Exhaustive search

```
# Calculating MSE for each fold, and for each model in the exhaustive search
exh = matrix(NA,k,8, dimnames =list(NULL , paste(1:8)))
for(j in 1:k){
  exh.fit = regsubsets(lsurv~., data = sdata[folds!=j, -9], nvmax= 8 )
  for(i in 1:8){
    pred = predict(exh.fit, sdata[folds == j, -9], id = i)
    exh[j, i] = mean((sdata$lsurv[folds == j] - pred)^2)
  }
}
# Reporting average MSE for each model in the exhaustive search
mean.exh=apply(exh,2,mean)
mean.exh
```

```
##         1         2         3         4         5         6
## 0.22327582 0.11966364 0.06804331 0.05304368 0.06207444 0.05963498
##         7         8
## 0.05660512 0.05656102
```

```
regfit.full <- regsubsets(lsurv ~ ., data=sdata[,-9])
coef(regfit.full, which.min(mean.exh))
```

```
## (Intercept)       blood        prog         enz    heavyAlc
##  3.85241856  0.07332263  0.01418507  0.01545270  0.35296762
```

Forward stepwise selection

```
fwd = matrix(NA,k,8, dimnames =list(NULL , paste(1:8)))
for(j in 1:k){
  fwd.fit = regsubsets(lsurv~., data = sdata[folds!=j, -9], nvmax= 8, method = "forward" )
  for(i in 1:8){
    pred = predict(fwd.fit, sdata[folds == j, -9], id = i)
    fwd[j, i] = mean((sdata$lsurv[folds == j] - pred)^2)
  }
}
mean.fwd=apply(fwd ,2,mean)
mean.fwd
```

```
##          1          2          3          4          5          6
## 0.22327582 0.12253753 0.07360553 0.05950694 0.05280154 0.06339662
##          7          8
## 0.05905524 0.05656102
```

```
reg.fwd = regsubsets(lsurv~., data = sdata[, -9], method = "forward")
coef(reg.fwd, which.min(mean.fwd))
```

```
## (Intercept)       blood        prog         enz      female    heavyAlc
##  3.86709541  0.07124119  0.01389038  0.01511505  0.08690962  0.36267739
```

Backward stepwise selection

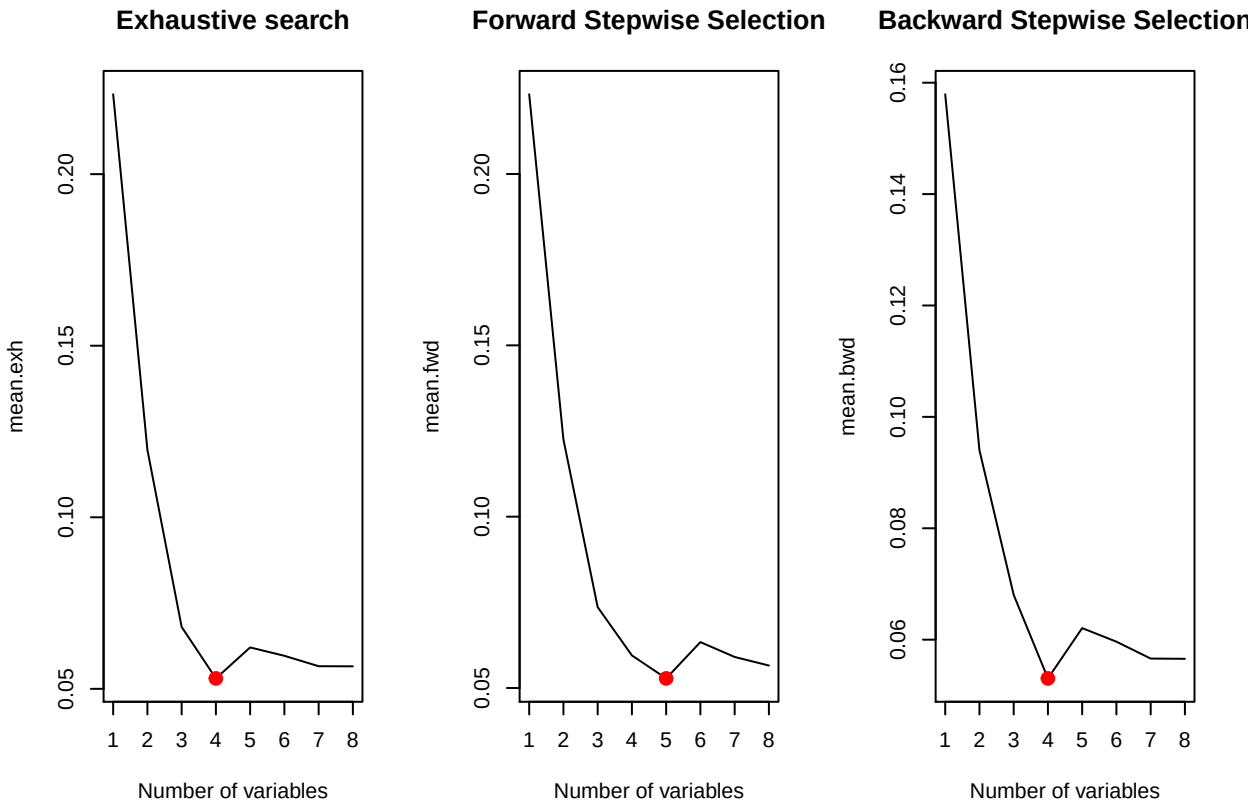
```
bwd = matrix(NA,k,8, dimnames =list(NULL , paste(1:8)))
for(j in 1:k){
  bwd.fit = regsubsets(lsurv~., data = sdata[folds!=j, -9], nvmax= 8, method = "backward" )
  for(i in 1:8){
    pred = predict(bwd.fit, sdata[folds == j, -9], id = i)
    bwd[j, i] = mean((sdata$lsurv[folds == j] - pred)^2)
  }
}
mean.bwd=apply(bwd, 2,mean)
mean.bwd
```

```
##          1          2          3          4          5          6
## 0.15792325 0.09400147 0.06804331 0.05304368 0.06207444 0.05963498
##          7          8
## 0.05660512 0.05656102
```

```
reg.bwd = regsubsets(lsurv~., data = sdata[, -9], method = "backward")
coef(reg.bwd, which.min(mean.bwd))
```

```
## (Intercept)       blood        prog         enz    heavyAlc
##  3.85241856  0.07332263  0.01418507  0.01545270  0.35296762
```

```
par(mfrow=c(1, 3))

plot(mean.exh, xlab='Number of variables', type = "l", main = "Exhaustive search")
points(which.min(mean.exh),
mean.exh[which.min(mean.exh)],
col = "red", cex = 2, pch = 20)

plot(mean.fwd, xlab='Number of variables', type = "l", main = "Forward Stepwise Selection")
points(which.min(mean.fwd),
mean.fwd[which.min(mean.fwd)],
col = "red", cex = 2, pch = 20)

plot(mean.bwd, xlab='Number of variables', type = "l", main = "Backward Stepwise Selection")
points(which.min(mean.bwd),
mean.bwd[which.min(mean.bwd)],
col = "red", cex = 2, pch = 20)
```



(b & c) As we see, different approaches came to choose 4 or 5 predictors among the 8 available ones. If we prefer the simpler model, we may select the exhustive search approach to select 4 predictors and have the best fit. It is also important to examine the stability of the predictors (i.e., whether all the models of a same size that are selected according to different algorithms do in fact contain the same predictors).