

6.5 Lab: Subset Selection Methods

Zongyi Liu

2023-05-30

6.5 Lab: Subset Selection Methods

Here we would use the `Hitters` data.

- `is.na()` function can be used to identify the missing observations. It returns a vector of the same length as the input vector, `TRUE` means the value is missing whereas `FALSE` means it is not missing.

```
library(ISLR)
fix(Hitters)
names(Hitters)
```

```
## [1] "AtBat"      "Hits"       "HmRun"      "Runs"       "RBI"        "Walks"
## [7] "Years"      "CAtBat"     "CHits"      "CHmRun"     "CRuns"      "CRBI"
## [13] "CWalks"     "League"     "Division"   "PutOuts"    "Assists"    "Errors"
## [19] "Salary"     "NewLeague"
```

```
dim(Hitters)
```

```
## [1] 322 20
```

```
sum(is.na(Hitters$Salary))
```

```
## [1] 59
```

So we know from here that the `Salary` column has 59 missing values, then try to remove them

```
Hitters=na.omit(Hitters)
```

The `regsubsets()` function (part of the `leaps` library) performs best subset selection by identifying the best model that contains a given number of predictors, where best is quantified using RSS.

```
library(leaps)
regfit.full=regsubsets(Salary~.,Hitters)
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., Hitters)
## 19 Variables (and intercept)
##           Forced in Forced out
## AtBat      FALSE      FALSE
## Hits       FALSE      FALSE
## HmRun       FALSE      FALSE
## Runs       FALSE      FALSE
## RBI        FALSE      FALSE
## Walks      FALSE      FALSE
## Years      FALSE      FALSE
## CAtBat     FALSE      FALSE
## CHits      FALSE      FALSE
## CHmRun     FALSE      FALSE
## CRuns      FALSE      FALSE
## CRBI       FALSE      FALSE
## CWalks     FALSE      FALSE
## LeagueN    FALSE      FALSE
## DivisionW  FALSE      FALSE
## PutOuts    FALSE      FALSE
## Assists    FALSE      FALSE
## Errors     FALSE      FALSE
## NewLeagueN FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##           AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " "
## 2 ( 1 ) " " "*" " " " " " " " " " " " " " " " "
## 3 ( 1 ) " " "*" " " " " " " " " " " " " " " " "
## 4 ( 1 ) " " "*" " " " " " " " " " " " " " " " "
## 5 ( 1 ) "*" "*" " " " " " " " " " " " " " " " "
## 6 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " "
## 7 ( 1 ) " " "*" " " " " " " "*" " " "*" "*" " " "
## 8 ( 1 ) "*" "*" " " " " " " "*" " " " " "*" "*" " "
##           CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1 ( 1 ) " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " "*" " " " " "
## 4 ( 1 ) " " " " "*" "*" " " " " " "
## 5 ( 1 ) " " " " "*" "*" " " " " " "
## 6 ( 1 ) " " " " "*" "*" " " " " " "
## 7 ( 1 ) " " " " "*" "*" " " " " " "
## 8 ( 1 ) "*" " " "*" "*" " " " " " "
```

The asterisk indicates that a given variable is included in the corresponding model. In this case, the best two-variable model contains only in Hits and CRBI.

By default, `regsubsets()` only reports results up to the best eight-variable model. But the `nvmax` option can be used in order to return as many variables as are desired.

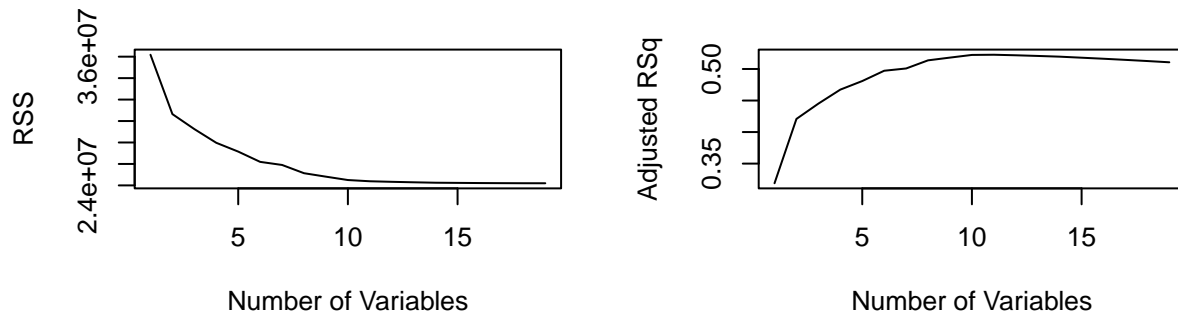
```
regfit.full=regsubsets(Salary~.,data=Hitters, nvmax=19)
reg.summary=summary(regfit.full)
reg.summary$rsq
```

```
## [1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146 0.5141227
```

```
## [8] 0.5285569 0.5346124 0.5404950 0.5426153 0.5436302 0.5444570 0.5452164
## [15] 0.5454692 0.5457656 0.5459518 0.5460945 0.5461159
```

Here, `reg.summary$rsq` can be used to get R^2 , we can also plot those parameters:

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



The `points()` command works like the `plot()` command, except that it puts points on a plot that has already been created, instead of creating a new plot.

The `which.max()` function can be used to identify the location of the maximum point of a vector.

```
which.max(reg.summary$adjr2)
```

```
## [1] 11
```

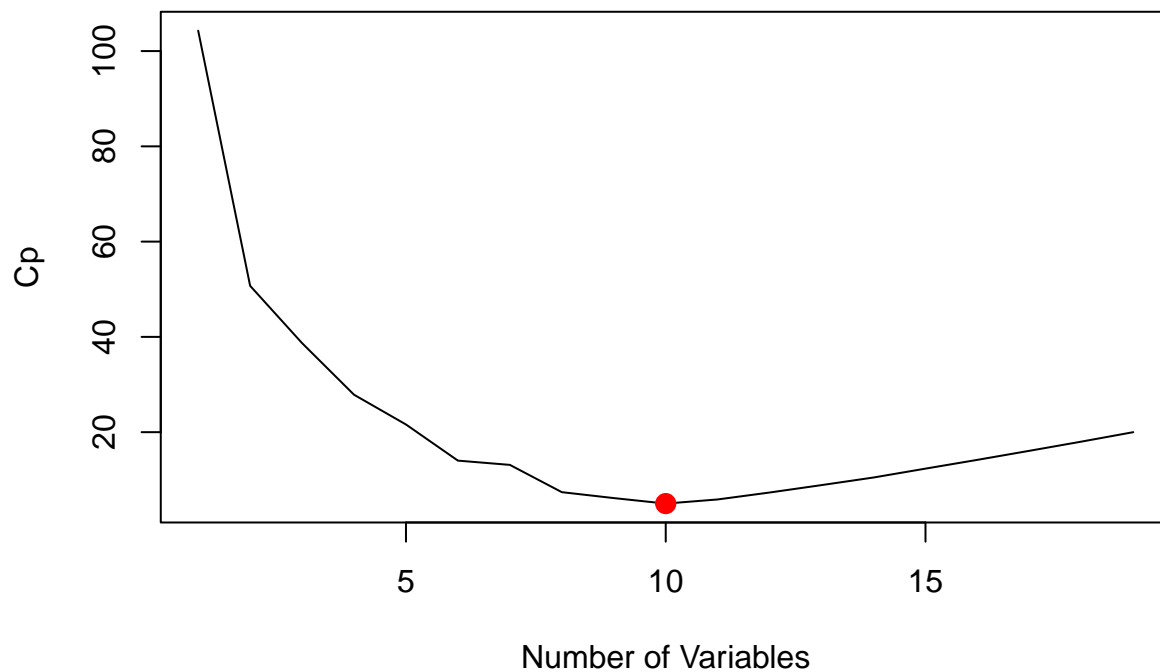
```
# points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)
```

Some extension of plotting:

```
plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
which.min(reg.summary$cp)
```

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

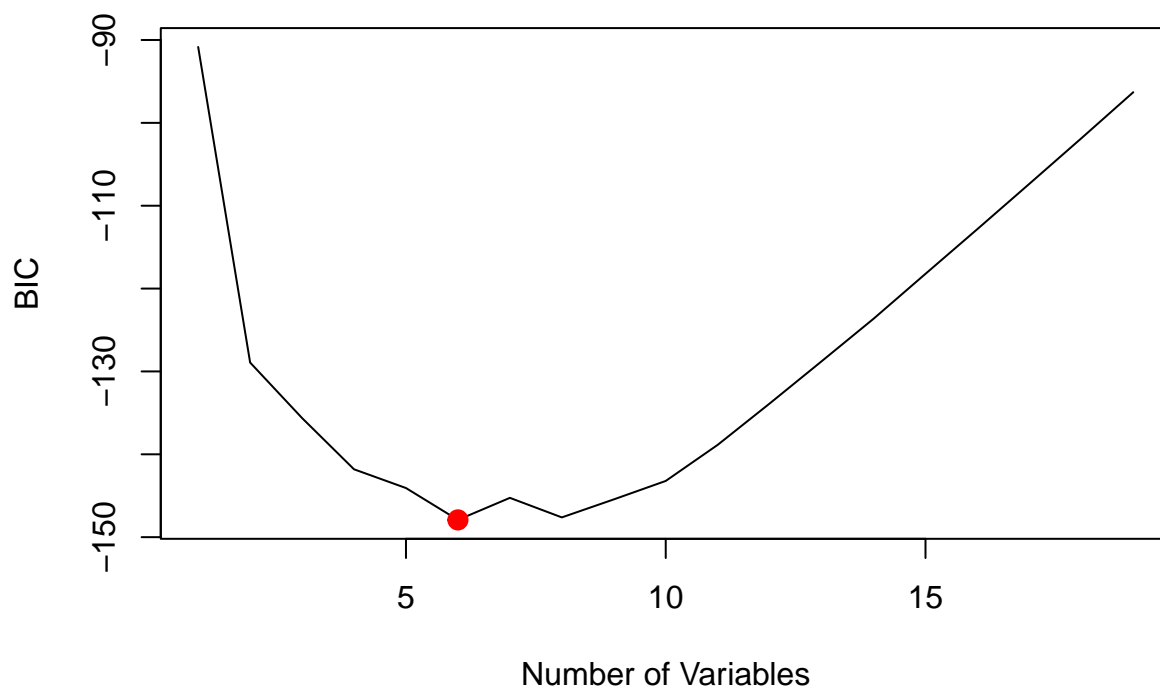
```
points(10,reg.summary$cp[10],col="red",cex=2,pch=20)
```



```
which.min(reg.summary$bic)
```

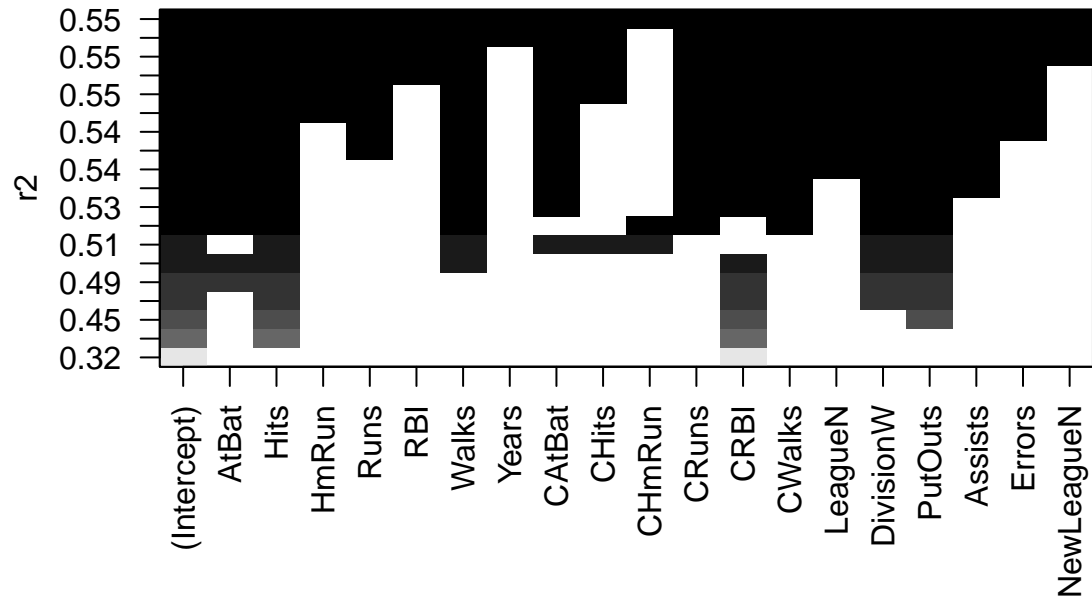
```
## [1] 6
```

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

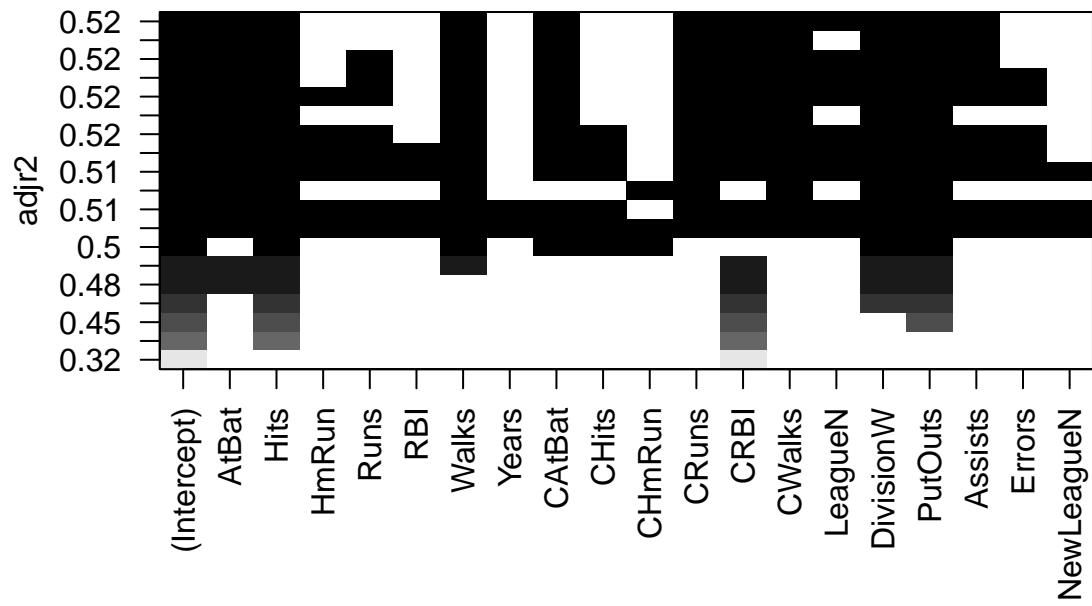


The `regsubsets()` function has a built-in `plot()` command which can be used to display the selected variables for the best model with a given number of predictors, can we can change the ranking criteria of those plots.

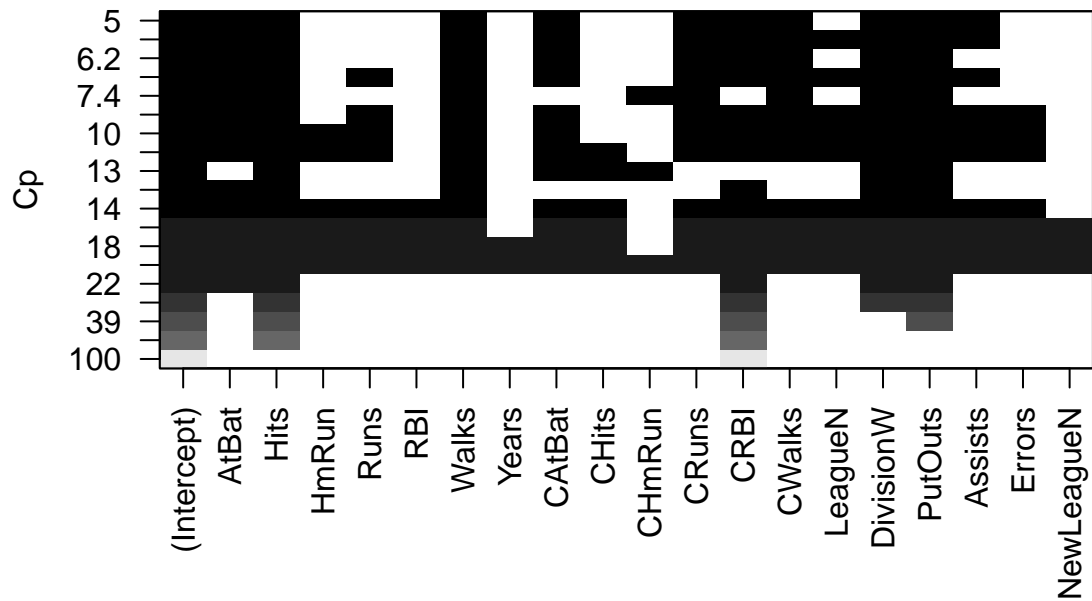
```
plot(regfit.full,scale="r2")
```



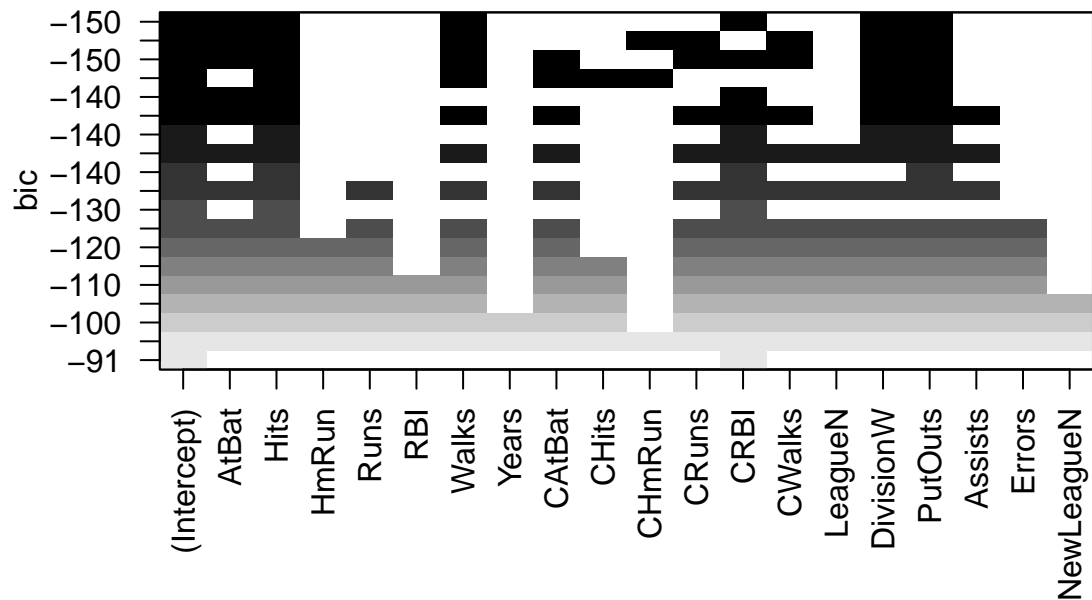
```
plot(regfit.full,scale="adjr2")
```



```
plot(regfit.full,scale="Cp")
```



```
plot(regfit.full,scale="bic")
```



6.5.2 Forward and Backward Stepwise Selection

We can also use `regsubset()` to do a forward or backward stepwise selection.

```
regfit.fwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="forward")
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19, method = "forward")
## 19 Variables (and intercept)
##      Forced in Forced out
```

```

## AtBat          FALSE      FALSE
## Hits           FALSE      FALSE
## HmRun          FALSE      FALSE
## Runs           FALSE      FALSE
## RBI            FALSE      FALSE
## Walks          FALSE      FALSE
## Years          FALSE      FALSE
## CAtBat         FALSE      FALSE
## CHits          FALSE      FALSE
## CHmRun         FALSE      FALSE
## CRuns          FALSE      FALSE
## CRBI           FALSE      FALSE
## CWalks         FALSE      FALSE
## LeagueN        FALSE      FALSE
## DivisionW      FALSE      FALSE
## PutOuts        FALSE      FALSE
## Assists        FALSE      FALSE
## Errors         FALSE      FALSE
## NewLeagueN     FALSE      FALSE
## 1 subsets of each size up to 19
## Selection Algorithm: forward
##      AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
## 1 ( 1 ) " " " " " " " " " " " " " " " " " "
## 2 ( 1 ) " " "*" " " " " " " " " " " " " " "
## 3 ( 1 ) " " "*" " " " " " " " " " " " " " "
## 4 ( 1 ) " " "*" " " " " " " " " " " " " " "
## 5 ( 1 ) "*" "*" " " " " " " " " " " " " " "
## 6 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " "
## 7 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " "
## 8 ( 1 ) "*" "*" " " " " " " "*" " " " " "*" " " "
## 9 ( 1 ) "*" "*" " " " " " " "*" " " "*" " " " " "
## 10 ( 1 ) "*" "*" " " " " " " "*" " " "*" " " " " "
## 11 ( 1 ) "*" "*" " " " " " " "*" " " "*" " " " " "
## 12 ( 1 ) "*" "*" " " " " "*" " " "*" " " " " " "
## 13 ( 1 ) "*" "*" " " " " "*" " " "*" " " " " " "
## 14 ( 1 ) "*" "*" "*" " " " " " "*" " " " " " " " "
## 15 ( 1 ) "*" "*" "*" " " " " " "*" "*" " " " " " "
## 16 ( 1 ) "*" "*" "*" " " " "*" " " "*" "*" " " " " " "
## 17 ( 1 ) "*" "*" "*" " " " "*" " " "*" "*" " " " " " "
## 18 ( 1 ) "*" "*" "*" " " " "*" " " "*" "*" " " " " " "
## 19 ( 1 ) "*" "*" "*" " " " "*" " " "*" "*" "*" " " " " "
##      CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1 ( 1 ) " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " "*" " " " " "
## 4 ( 1 ) " " " " "*" " "*" " " " " "
## 5 ( 1 ) " " " " "*" " "*" " " " " "
## 6 ( 1 ) " " " " "*" " "*" " " " " "
## 7 ( 1 ) "*" " " " " "*" " "*" " " " " "
## 8 ( 1 ) "*" " " " " "*" " "*" " " " " "
## 9 ( 1 ) "*" " " " " "*" " "*" " " " " "
## 10 ( 1 ) "*" " " " " "*" " "*" " " " " "
## 11 ( 1 ) "*" "*" " " "*" " "*" " " " " "
## 12 ( 1 ) "*" "*" " " "*" " "*" " " " " "

```

```
## 13 ( 1 ) "*" "*" "*" "*" "*" "*" " "
## 14 ( 1 ) "*" "*" "*" "*" "*" "*" " "
## 15 ( 1 ) "*" "*" "*" "*" "*" "*" " "
## 16 ( 1 ) "*" "*" "*" "*" "*" "*" " "
## 17 ( 1 ) "*" "*" "*" "*" "*" "*" "*"
## 18 ( 1 ) "*" "*" "*" "*" "*" "*" "*"
## 19 ( 1 ) "*" "*" "*" "*" "*" "*" "*"

```

```
regfit.bwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="backward")
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19, method = "backward")
## 19 Variables (and intercept)
##           Forced in Forced out
## AtBat      FALSE      FALSE
## Hits       FALSE      FALSE
## HmRun      FALSE      FALSE
## Runs       FALSE      FALSE
## RBI        FALSE      FALSE
## Walks      FALSE      FALSE
## Years      FALSE      FALSE
## CAtBat     FALSE      FALSE
## CHits      FALSE      FALSE
## CHmRun     FALSE      FALSE
## CRuns      FALSE      FALSE
## CRBI       FALSE      FALSE
## CWalks     FALSE      FALSE
## LeagueN    FALSE      FALSE
## DivisionW  FALSE      FALSE
## PutOuts    FALSE      FALSE
## Assists    FALSE      FALSE
## Errors     FALSE      FALSE
## NewLeagueN FALSE      FALSE
## 1 subsets of each size up to 19
## Selection Algorithm: backward
##           AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
## 1 ( 1 ) " " " " " " " " " " " " " " " " "*" " "
## 2 ( 1 ) " " "*" " " " " " " " " " " " " " "*" " "
## 3 ( 1 ) " " "*" " " " " " " " " " " " " " "*" " "
## 4 ( 1 ) "*" "*" " " " " " " " " " " " " " "*" " "
## 5 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " "*" " "
## 6 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " "*" " "
## 7 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " "*" " "
## 8 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " "*" "*"
## 9 ( 1 ) "*" "*" " " " " " " "*" " " "*" " " " " " "*" "*"
## 10 ( 1 ) "*" "*" " " " " " " "*" " " "*" " " " " " "*" "*"
## 11 ( 1 ) "*" "*" " " " " " " "*" " " "*" " " " " " "*" "*"
## 12 ( 1 ) "*" "*" " " "*" " " "*" " " " "*" " " " " " "*" "*"
## 13 ( 1 ) "*" "*" " " "*" " " "*" " " " "*" " " " " " "*" "*"
## 14 ( 1 ) "*" "*" "*" "*" " " "*" " " " "*" " " " " " "*" "*"
## 15 ( 1 ) "*" "*" "*" "*" " " "*" " " " "*" "*" " " " " "*" "*"
## 16 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "*" "*" " " " " "*" "*"
## 17 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "*" "*" " " " " "*" "*"

```



```

## 18 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" " " "*" "*"
## 19 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
##           CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1 ( 1 ) " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " "*" " " " " "
## 4 ( 1 ) " " " " " " "*" " " " " "
## 5 ( 1 ) " " " " " " "*" " " " " "
## 6 ( 1 ) " " " " "*" "*" " " " " " "
## 7 ( 1 ) "*" " " "*" "*" " " " " " "
## 8 ( 1 ) "*" " " "*" "*" " " " " " "
## 9 ( 1 ) "*" " " "*" "*" " " " " " "
## 10 ( 1 ) "*" " " "*" "*" "*" " " " " "
## 11 ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 12 ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 13 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 14 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 15 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 16 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 17 ( 1 ) "*" "*" "*" "*" "*" "*" "*" " "
## 18 ( 1 ) "*" "*" "*" "*" "*" "*" "*" " "
## 19 ( 1 ) "*" "*" "*" "*" "*" "*" "*" " "

```

We can notice that the best seven-variable models identified by forward stepwise selection, backward stepwise selection, and best subset selection are different.

```
coef(regfit.full,7)
```

```

## (Intercept)      Hits      Walks      CAtBat      CHits      CHmRun
## 79.4509472    1.2833513    3.2274264   -0.3752350    1.4957073    1.4420538
## DivisionW      PutOuts
## -129.9866432    0.2366813

```

```
coef(regfit.fwd,7)
```

```

## (Intercept)      AtBat      Hits      Walks      CRBI      CWalks
## 109.7873062   -1.9588851    7.4498772    4.9131401    0.8537622   -0.3053070
## DivisionW      PutOuts
## -127.1223928    0.2533404

```

```
coef(regfit.bwd,7)
```

```

## (Intercept)      AtBat      Hits      Walks      CRuns      CWalks
## 105.6487488   -1.9762838    6.7574914    6.0558691    1.1293095   -0.7163346
## DivisionW      PutOuts
## -116.1692169    0.3028847

```

6.5.3 Choosing Among Models Using the Validation Set Approach and Cross-Validation