

Ch 5 1 ModelSelect

Analyze the ISLR (Introduction to Statistical Learning with R) data package's baseball 'Hitters' data frame:

```
library(ISLR)
summary(Hitters)
```

```
##           AtBat           Hits           HmRun           Runs
## Min.      : 16.0   Min.       :  1   Min.       : 0.00   Min.       :  0.00
## 1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25
## Median :379.5   Median : 96   Median : 8.00   Median : 48.00
## Mean     :380.9   Mean    :101   Mean     :10.77   Mean     : 50.91
## 3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00
## Max.     :687.0   Max.     :238   Max.     :40.00   Max.     :130.00
##
##           RBI           Walks           Years           CAtBat
## Min.      :  0.00   Min.      :  0.00   Min.      : 1.000   Min.      :  19.0
## 1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.: 816.8
## Median : 44.00   Median : 35.00   Median : 6.000   Median :1928.0
## Mean     : 48.03   Mean     : 38.74   Mean     : 7.444   Mean     :2648.7
## 3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.:3924.2
## Max.     :121.00   Max.      :105.00   Max.      :24.000   Max.      :14053.0
##
##           CHits           CHmRun           CRuns           CRBI
## Min.      :  4.0   Min.      :  0.00   Min.      :  1.0   Min.      :  0.00
## 1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.: 88.75
## Median : 508.0   Median : 37.50   Median : 247.0   Median :220.50
## Mean     : 717.6   Mean     : 69.49   Mean     : 358.8   Mean     :330.12
## 3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.:426.25
## Max.     :4256.0   Max.      :548.00   Max.      :2165.0   Max.      :1659.00
##
##           CWalks           League Division           PutOuts           Assists
## Min.      :  0.00   A:175   E:157   Min.      :  0.0   Min.      :  0.0
## 1st Qu.: 67.25   N:147   W:165   1st Qu.: 109.2   1st Qu.:  7.0
## Median : 170.50                               Median : 212.0   Median : 39.5
## Mean     : 260.24                               Mean     : 288.9   Mean     :106.9
## 3rd Qu.: 339.25                               3rd Qu.: 325.0   3rd Qu.:166.0
## Max.     :1566.00                               Max.      :1378.0   Max.      :492.0
##
##           Errors           Salary           NewLeague
## Min.      :  0.00   Min.      : 67.5   A:176
## 1st Qu.:  3.00   1st Qu.: 190.0   N:146
## Median :  6.00   Median : 425.0
## Mean     :  8.04   Mean     :535.9
## 3rd Qu.: 11.00   3rd Qu.: 750.0
## Max.     : 32.00   Max.     :2460.0
##                      NA's      :59
```

There are missing values, before we proceed we will remove them:

```
with(Hitters, sum(is.na(Salary)))
```

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

```
Hitters=na.omit(Hitters)
with(Hitters, sum(is.na(Salary)))
```

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

Best Subset regression

We will now use the package `leaps` to evaluate all the best-subset models.

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

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = 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
## 1 ( 1 ) " " " " " " " " " " " " " " " " " "
## 2 ( 1 ) " " "*" " " " " " " " " " " " " " "
## 3 ( 1 ) " " "*" " " " " " " " " " " " " " "
## 4 ( 1 ) " " "*" " " " " " " " " " " " " " "
## 5 ( 1 ) "*" "*" " " " " " " " " " " " " " "
## 6 ( 1 ) "*" "*" " " " " " " "*" " " " " " " "
## 7 ( 1 ) " " "*" " " " " " " "*" " " "*" "*" " "
## 8 ( 1 ) "*" "*" " " " " " " "*" " " "*" "*" *
```

```
##          CRBI CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1 ( 1 ) "*" " " " " " " " " " " " "
## 2 ( 1 ) "*" " " " " " " " " " " " "
## 3 ( 1 ) "*" " " " " " " "*" " " " " "
## 4 ( 1 ) "*" " " " " "*" "*" " " " " " "
## 5 ( 1 ) "*" " " " " "*" "*" " " " " " "
## 6 ( 1 ) "*" " " " " "*" "*" " " " " " "
## 7 ( 1 ) " " " " " " "*" "*" " " " " " "
## 8 ( 1 ) " " "*" " " "*" "*" " " " " " "
```

By default, it gives the first 8 variables best-subset models. Let's do it again for all the variables:

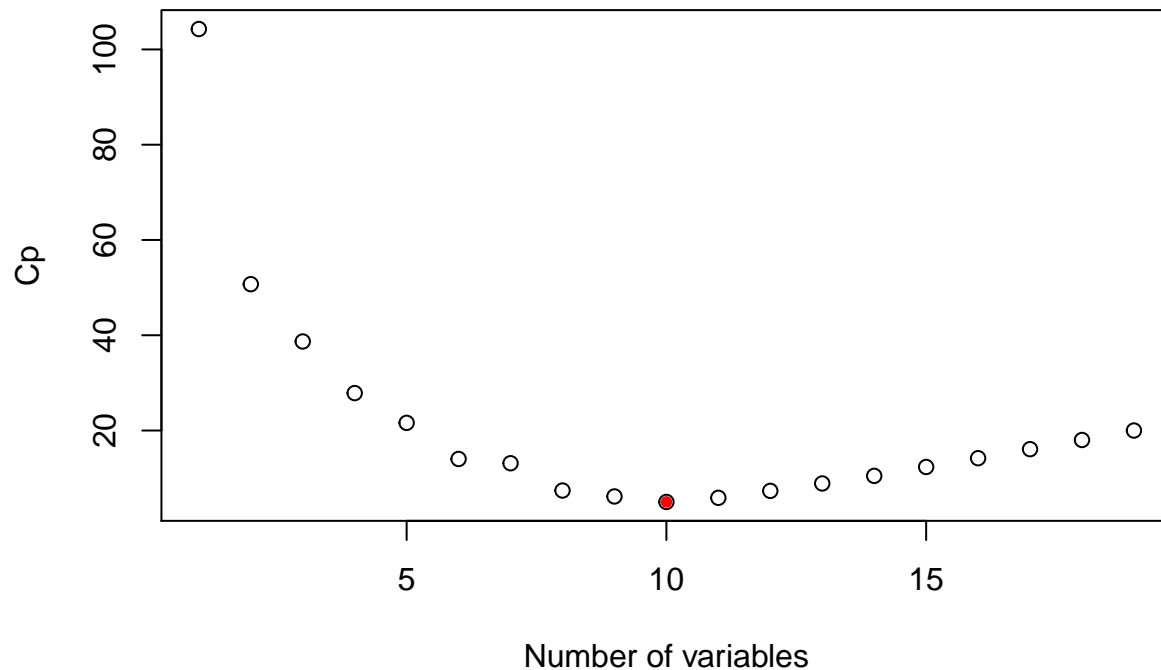
```
regfit.full = regsubsets(Salary~., data=Hitters, nvmax=19)
reg.summary = summary(regfit.full)
names(reg.summary)
```

```
## [1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"
```

```
plot(reg.summary$cp, xlab="Number of variables", ylab="Cp")
which.min(reg.summary$cp)
```

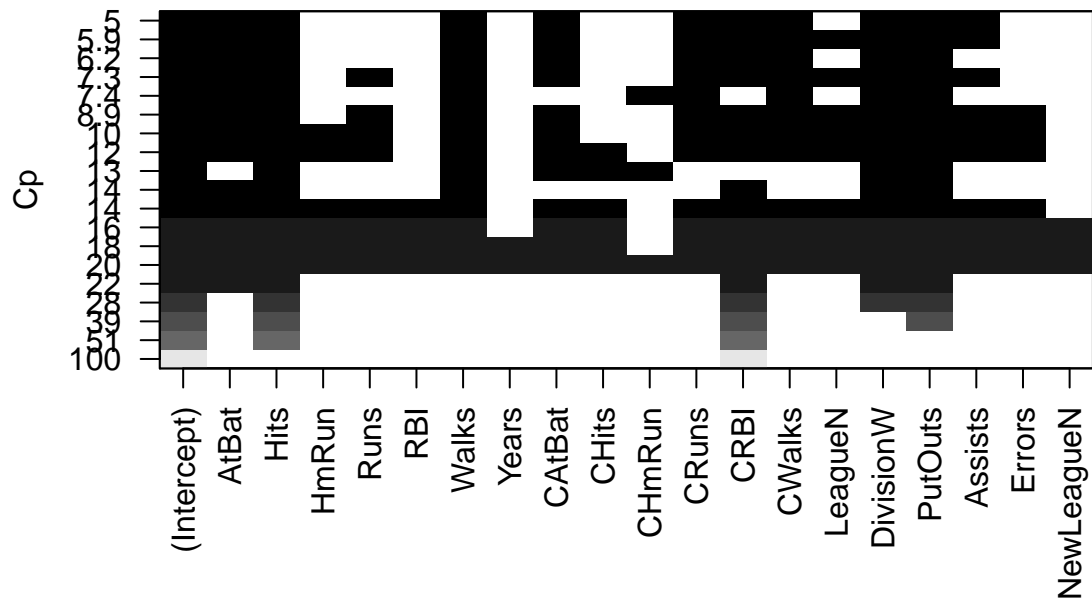
```
## [1] 10
```

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



There is a method for the `regsubset` object:

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



```
coef(regfit.full, 10)
```

```
## (Intercept)      AtBat      Hits      Walks      CAtBat
## 162.5354420    -2.1686501    6.9180175    5.7732246   -0.1300798
##      CRuns      CRBI      CWalks    DivisionW    PutOuts
##   1.4082490    0.7743122   -0.8308264  -112.3800575    0.2973726
##      Assists
##   0.2831680
```

Forward Stepwise Selection

We use `regsubset` again, but specify the `method="forward"` option.

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

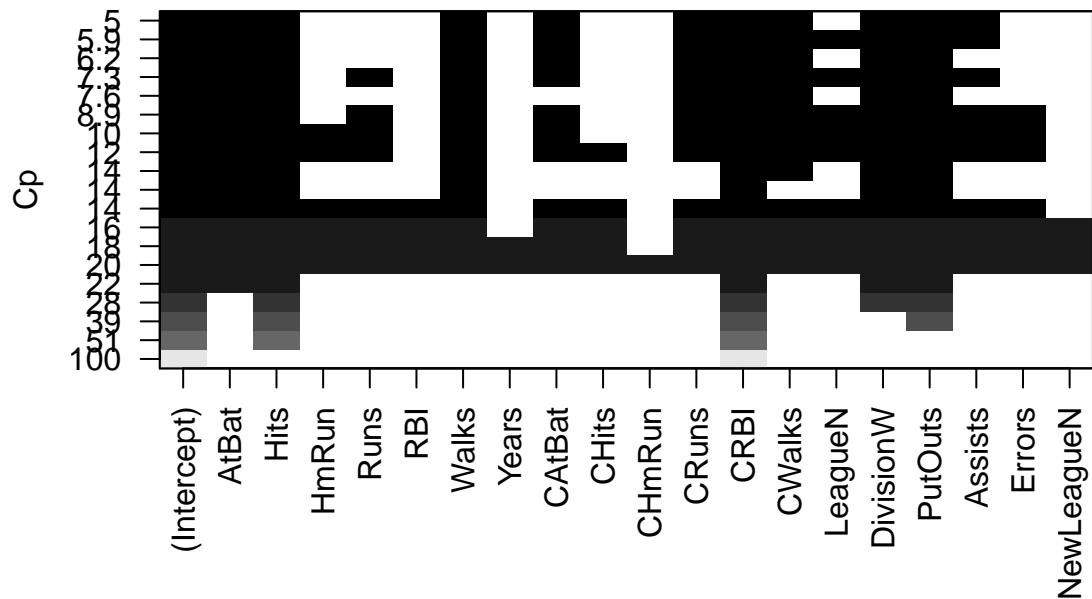
```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19, method = "forward")
## 19 Variables (and intercept)
##           Forced in Forced out
## AtBat      FALSE      FALSE
## Hits       FALSE      FALSE
## HmRun       FALSE      FALSE
## Runs       FALSE  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

```

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



Model Selection Using a Validation Set

Let's make a training and validation set, so that we can choose a good subset model.

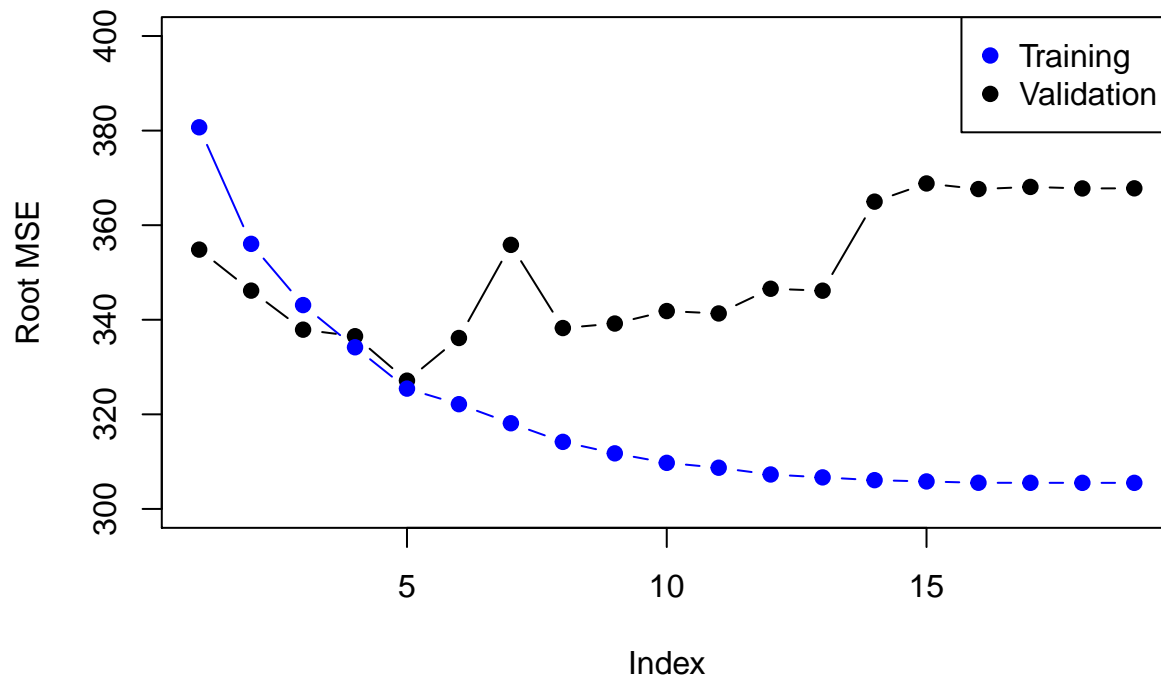
```
dim(Hitters)
```

```
## [1] 263 20
```

```
set.seed(1)
train = sample(seq(263),180,replace=FALSE)
regfit.fwd = regsubsets(Salary~., data=Hitters[train,], nvmax=19, method="forward")
```

Now, we separate the data to two parts, one for training and another for test/validation to make prediction. There is no `prediction` method for `regsubsets`, so we need to write that part. We also create a vector to store the results of the 19 different models.

```
val.errors = rep(NA, 19)
x.test = model.matrix(Salary~., data=Hitters[-train,])
for(i in 1:19){
  coefi = coef(regfit.fwd, id=i)
  pred = x.test[,names(coefi)]%*%coefi
  val.errors[i] = mean((Hitters$Salary[-train]-pred)^2)
}
plot(sqrt(val.errors), ylab="Root MSE", ylim=c(300,400), pch=19, type="b")
points(sqrt(regfit.fwd$rss[-1]/180),col="blue",pch=19,type="b") # -1 excludes null model
legend("topright", legend=c("Training","Validation"),col=c("blue","black"),pch=19)
```



As expected, the model error goes down monotonically as the model gets bigger, but not so for the validation error.

This was a little tedious - not having a `predict` method `regsubsets`. So we will write one!

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

```
## Salary ~ .
```

Model Seletion by Cross-Validation

We will do a 10-fold cross-validation.

```
set.seed(11)
folds = sample(rep(1:10,length=nrow(Hitters)))
table(folds)
```

```
## folds
##  1  2  3  4  5  6  7  8  9 10
## 27 27 27 26 26 26 26 26 26 26
```

```
cv.errors = matrix(NA,10,19)
for(k in 1:10){ # Loop for folds
  best.fit = regsubsets(Salary~., data=Hitters[folds!=k,],nvmax=19,method="forward")
  for(i in 1:19){ # Loop for sizes of picked feature best subsets
```

```

pred = predict(best.fit,Hitters[folds==k,], id=i)
cv.errors[k,i] = mean((Hitters$Salary[folds==k]-pred)^2)
}
}
rmse.cv = sqrt(apply(cv.errors,2,mean))
plot(rmse.cv, pch=19, type="b")

```

