Linear Model Selection and Regularization

Rebecca C. Steorts, Duke University

STA 325, Chapter 6 ISL

Agenda

Goals

Our focus is to discuss some ways in which the simple linear model can be improved, by replacing plain least squares fitting with some alternative fitting procedures.

As we will see, alternative fitting procedures can yield better prediction accuracy and model interpretability.

Looking Ahead

In this chapter, we discuss three important classes of methods.

1. Subset Selection: This approach involves identifying a subset of the p predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.
2. Shrinkage: This approach involves fitting a model involving all p predictors. However, the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage has the effect of reducing variance.
3. Dimension Reduction: This approach involves projecting the p predictors into a $M-$dimensional subspace, where $M < p$. This is achieved by computing M different linear combinations, or projections, of the variables. Then these M projections are used as predictors to fit a linear regression model by least squares.

Subset Selection

We consider some methods for selecting subsets of predictors. These include **best subset** and **stepwise model** selection procedures.

Best Subset Selection

To perform best subset selection, we fit a separate least squares regression for each possible combination of the predictors.

We first all models that contain one predictor, all $\binom{p}{2} = p(p-1)/2$ models that contain exactly two predictors, and so forth.

We then look at all of the resulting models, with the goal of identifying the one that is "best."

The problem of selecting the best model from among the 2^p possibilities considered by **best subset selection** is not trivial.

This is usually broken up into two stages (see Algorithm).

Best Subset Selection Algorithm

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, \ldots, p$:

   - Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   - Pick the best among these $\binom{p}{k}$ models and call it M_k.
   - Here, best is defined as having the smallest RSS (or largest R^2).

3. Select a single best model from among $M_0, \ldots M_p$ using CV, C_p, AIC, BIC, or adjusted R^2.

Caution: Best Subset Selection

- Step 2 identifies the best model (on the training data) for each subset size, in order to reduce the problem from one of 2^p possible models to one of $p + 1$ possible models.
- To select a single best model, we must simply choose among these $p + 1$ options.
- This task must be performed with care, because the RSS of these $p + 1$ models decreases monotonically, and the R^2 increases monotonically, as the number of features included in the models increases.
- If we use these statistics to select the best model, then we will always end up with a model involving all of the variables.
- The problem is that a low RSS or a high R^2 indicates a model with a low training error, whereas we wish to choose a model that has a **low test error**.

Solution: Best Subset Selection

- Thus, in Step 3, we use cross-validated prediction error (CV, C_p, AIC, BIC, or adjusted R^2) to select from among $M_0, \ldots M_p$.

Computational Limitations

- While best subset selection is a simple and conceptually appealing approach, it suffers from computational limitations.
- In general, there are 2^p models that involve subsets of p predictors.
- If p $= 10$, then there are approximately 1,000 possible models to be considered.
- If p $= 20$, then there are over one million possibilities!
- Consequently, best subset selection becomes computationally infeasible for values of $p > 40$.
- Due to this, we present computationally efficient alternatives to best subset selection.

Stepwise Selection

- Due to computational limitations and suffering from statistical limitations when p is large, we consider **stepwise selection methods**.

Forward Stepwise Selection

Forward stepwise selection is a computationally efficient alternative to best subset selection.

- While the **best subset selection** procedure considers all 2^p possible models containing subsets of the p predictors, **forward stepwise** considers a much smaller set of models.

Forward Stepwise Selection

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.
- We present the algorithm on the next slide.

Forward Stepwise Selection Algorithm

1. Let M_0 denote the null model which contains no predictors.
2. For $k = 0, \ldots p - 1$:

- Consider all $p - k$ models that augment the predictors in M_k with one additional predictor.
- Choose the best among these $p - k$ models, and call it M_{k+1}. Here, best is defined as having smallest RSS or highest R^2

3. Select a single best model from among $M_0, \ldots M_p$ using cross validated prediction error: C_p, AIC, BIC, or adjusted R^2.

Computational Advantages

- Unlike best subset selection, which involved fitting 2^p models, forward stepwise selection involves fitting one null model, along with $p - k$ models in iteration k, for $k = 0, \ldots, p - 1$.
- This amounts to a total of

$$1 + p - 1(p - k) = 1 + p(p + 1)/2$$

  models.
- This is a substantial difference: when $k = 0, p = 20$, best subset selection requires fitting 1,048,576 models, whereas forward stepwise selection requires fitting only 211 models.

Computation versus Best Model

- Though forward stepwise tends to do well in practice, it is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.
- For instance, suppose that in a given data set with $p = 3$ predictors, the best possible one-variable model contains X_1, and the best possible two-variable model instead contains X_2 and X_3.
- Then forward stepwise selection will fail to select the **best possible two-variable model**.
- This is because M_1 will contain X_1, so M_2 must also contain X_1 together with one additional variable.
- Thus, a model with X_2 and X_3 is impossible!

Backward Stepwise Selection

- Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection.
- Unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.
- We give the details of the algorithm on the next slide.

Backward Stepwise Selection Algorithm

1. Let M_p denote the full model, which contains all p predictors.
2. For $k = p, p-1, \ldots 1$:

- Consider all k models that contain all but one of the predictors in M_k, for a total of $k-1$ predictors.
- Choose the best among these k models, and call it M_{k-1}.
- Here, best is defined as having smallest RSS or highest R^2.

3. Select a single model from among $M_0, \ldots M_p$ using cross validated prediction error: C_p, AIC, BIC, or adjusted R^2.

Computational Advantages

- Like forward stepwise selection, the backward selection approach searches through only $1 + p(p + 1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection
- Also like forward stepwise selection, backward stepwise selection is not guaranteed to yield the best model containing a subset of the p predictors.

Backward versus Forward Selection

- Backward selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit).
- In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

Choosing the Optimal Model

- Best subset selection, forward selection, and backward selection result in the creation of a set of models, each of which contains a subset of the p predictors.
- In order to implement these methods, we need a way to determine which of these models is best.
- We wish to choose a model with a low test error and there are two common approaches:

1. We can indirectly estimate test error by making an adjustment to the training error to account for the bias due to overfitting.
2. We can directly estimate the test error, using either a validation set approach or a cross-validation approach, as discussed in Chapter 5.

C_p, AIC, BIC, and Adjusted R^2

- Recall that the training set MSE is generally an under-estimate of the test MSE.
- Recall that MSE $=$ RSS/n.
- A number of techniques for adjusting the training error for the model size are available.
- These approaches can be used to select among a set of models with different numbers of variables.

C_p, AIC, BIC, and Adjusted R^2

We now consider four such approaches:

1. C_p,
2. Akaike information criterion (AIC),
3. Bayesian information criterion (BIC), and
4. adjusted R^2

For simplicity, we consider these approaches for least squares.

C_p

For a fitted least squares model containing d predictors, the C_p estimate of test MSE is computed using the equation

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2), \tag{1}$$

where $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ associated with each response measurement in the standard linear model:

$$Y = \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p + \epsilon \tag{2}$$

Mallow's C_p

Mallow's C_p is sometimes defined as

$$C_p\prime = RSS/\hat{\sigma}^2 + 2d - n$$

- This is equivalent to definition given above in the sense that

$$C_p = \hat{\sigma}^2(C_p\prime + n)$$

  so the model with the smallest C_p has the smallest $C_p\prime$.
- The C_p statistic adds a penalty of $2d\hat{\sigma}^2$ to the training RSS to adjust for the fact that the training error tends to underestimate the test error.
- Clearly, the penalty increases as the number of predictors in the model increases; this is intended to adjust for the corresponding decrease in training RSS.

AIC criterion

In the case of equation 2 with Gaussian errors, maximum likelihood and least squares are the same thing.

AIC is given by:

$$\text{AIC} = \frac{1}{n\hat{\sigma^2}}(RSS + 2d\hat{\sigma^2}), \tag{3}$$

where, for simplicity, we have omitted an additive constant.

For least squares models, C_p and AIC are proportional to each other.

BIC

BIC is derived from a Bayesian point of view, but ends up looking similar to C_p (and AIC) as well.

For the least squares model with d predictors, the BIC is, up to irrelevant constants, given by

$$\text{BIC} = \frac{1}{n}(RSS + \log(n) d \hat{\sigma}^2).$$

Like C_p, the BIC will tend to take on a small value for a model with a low test error, and so generally we select the model that has the **lowest BIC value**.

BIC and C_p

Recall

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2),$$

$$\text{BIC} = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2).$$

- BIC replaces the $2d\hat{\sigma}^2$ used by C_p with a $\log(n)d\hat{\sigma}^2$ term, where n is the number of observations.

Since $\log(n) > 2$ for any $n > 7$, the BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than C_p.

adjusted R^2

Recall that

$$R^2 = 1 - RSS/TSS$$

, where

$$\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \qquad \text{(total sum of squares)}.$$

Since RSS always increases as more variables are added, the R^2 always increases as well.

For a least squares model with d variables, adjusted R^2 statistic is calculated as

$$\text{adjusted } R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}.$$

adjusted R^2

Unlike C_p, AIC, and BIC, for which a small value indicates a model with a low test error, a large value of adjusted R^2 indicates a model with a small test error.

Maximizing the adjusted R^2 is equivalent to minimizing

$$RSS/(n-d-1).$$

While RSS always decreases as the number of variables in the model increases, RSS/(n-d-1) may **increase or decrease** due to the presence of d in the demoninator.

adjusted R^2

The intuition behind the adjusted R^2 is that once all of the correct variables have been included in the model, adding additional noise variables will lead to only a very small decrease in RSS.

Since adding noise variables leads to an increase in d, such variables will lead to an increase in RSS/(n-d-1) and consequently a decrease in the adjusted R^2.

Unlike the R^2 statistic, the adjusted R^2 statistic pays a price for the inclusion of unnecessary variables in the model.

Justifications

- Cp, AIC, and BIC all have rigorous theoretical justifications that are beyond the scope of this course.
- These justifications rely on asymptotic arguments (scenarios where the sample size n is very large).

Validation and Cross-Validation

As an alternative to the approaches just discussed, we can directly estimate the test error using the validation set and cross-validation methods.

We can compute the validation set error or the cross-validation error for each model under consideration, and then select the model for which the resulting estimated test error is smallest.

This procedure has an advantage relative to AIC, BIC, C_p, and adjusted R^2, in that it provides a **direct estimate** of the test error, and makes **fewer assumptions** about the true underlying model.

It can also be used in a wider range of model selection tasks, even in cases where it is hard to pinpoint the model degrees of freedom (e.g. the number of predictors in the model) or hard to estimate the error variance σ^2.

Best Subset Regression to Hitters Data Set

We apply the best subset selection approach to the Hitters data. We wish to predict a baseball player's Salary on the basis of various statistics associated with performance in the previous year.

Note that the Salary variable is missing for some of the players. The is.na() function can be used to identify the missing observations. The sum() function can then be used to count all of the missing elements.

Best Subset Regression to Hitters Data Set

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

```
##  [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RB
##  [6] "Walks"     "Years"     "CAtBat"    "CHits"     "CH
## [11] "CRuns"     "CRBI"      "CWalks"    "League"    "Di
## [16] "PutOuts"   "Assists"   "Errors"    "Salary"    "Ne
```

```
dim(Hitters)
```

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

```
# count number of missing values
sum(is.na(Hitters$Salary))
```

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

Best Subset Regression to Hitters Data Set

We see that Salary is missing for 59 players. The na.omit() function removes all of the rows that have missing values in any variable.

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

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

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

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

Best Subset Regression to Hitters Data Set

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.

The syntax is the same as for lm().

The summary() command outputs the best set of variables for each model size.

Best Subset Regression to Hitters Data Set

```
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
## 1  ( 1 ) " "   " "  " "   " "  " " " "   " "   " "    " "   " "    " "
## 2  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "
## 3  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "
## 4  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "
```

Best Subset Regression to Hitters Data Set

An asterisk indicates that a given variable is included in the corresponding model.

For instance, this output indicates that the best two-variable model contains only 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.

Here we fit up to a 19-variable model.

Best Subset Regression to Hitters Data Set

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

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

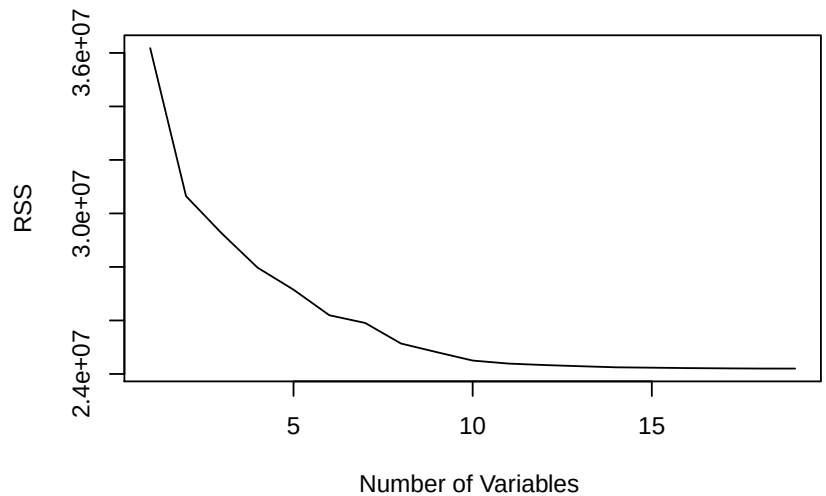
The summary() function also returns R^2, RSS, adjusted R^2, C_p, and BIC.

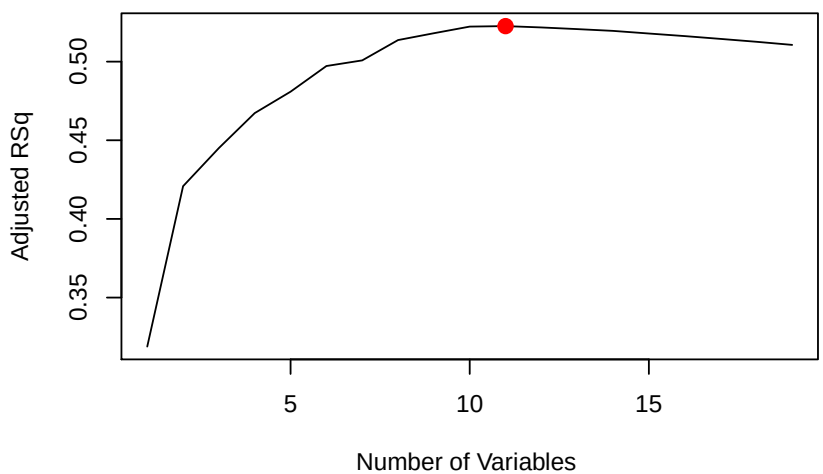
We can examine these to try to select the best overall model.

Best Subset Regression to Hitters Data Set

```
##  [1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0
##  [8] 0.5285569 0.5346124 0.5404950 0.5426153 0.5436302 0
## [15] 0.5454692 0.5457656 0.5459518 0.5460945 0.5461159
```

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

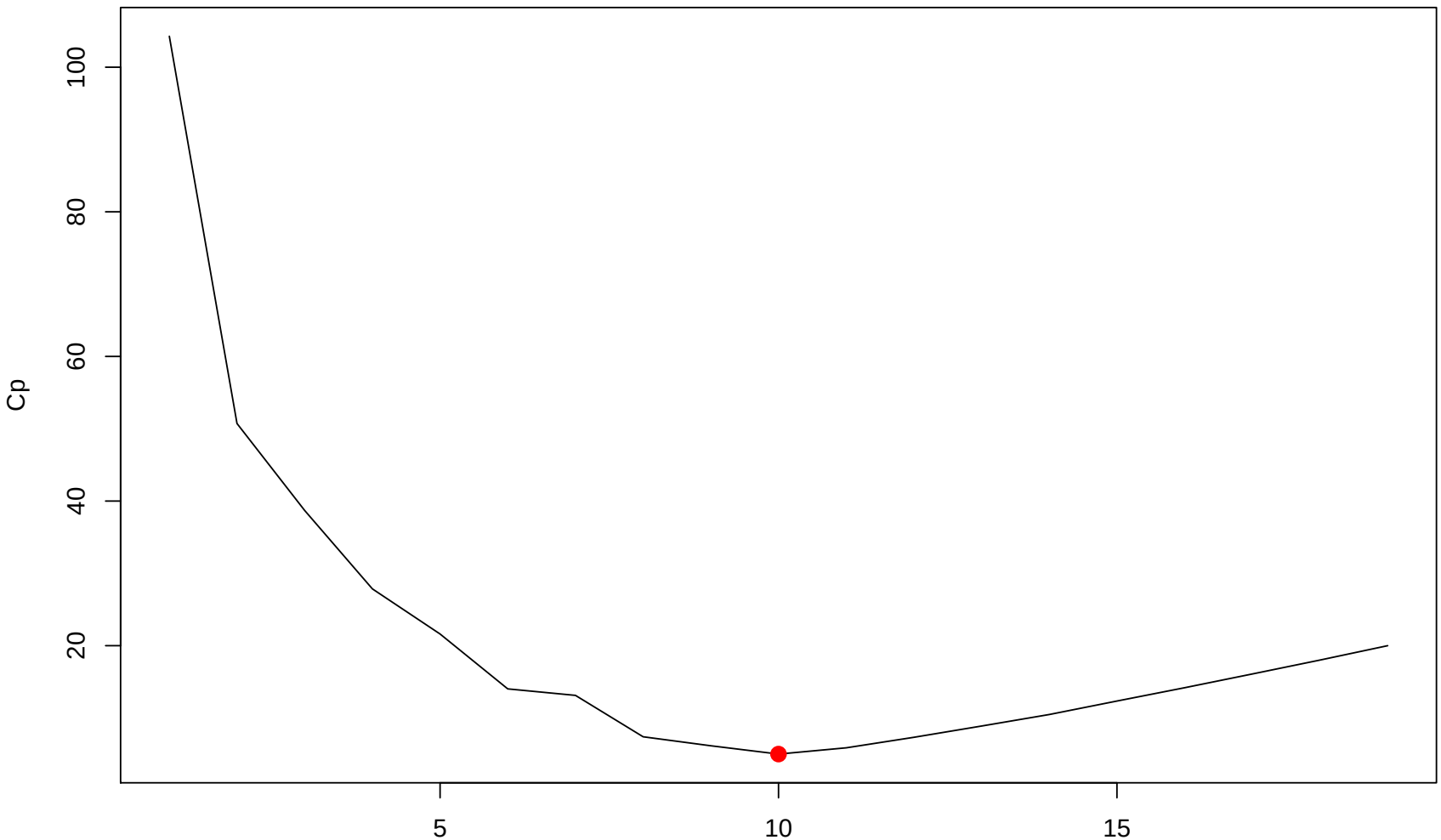




C_p and BIC

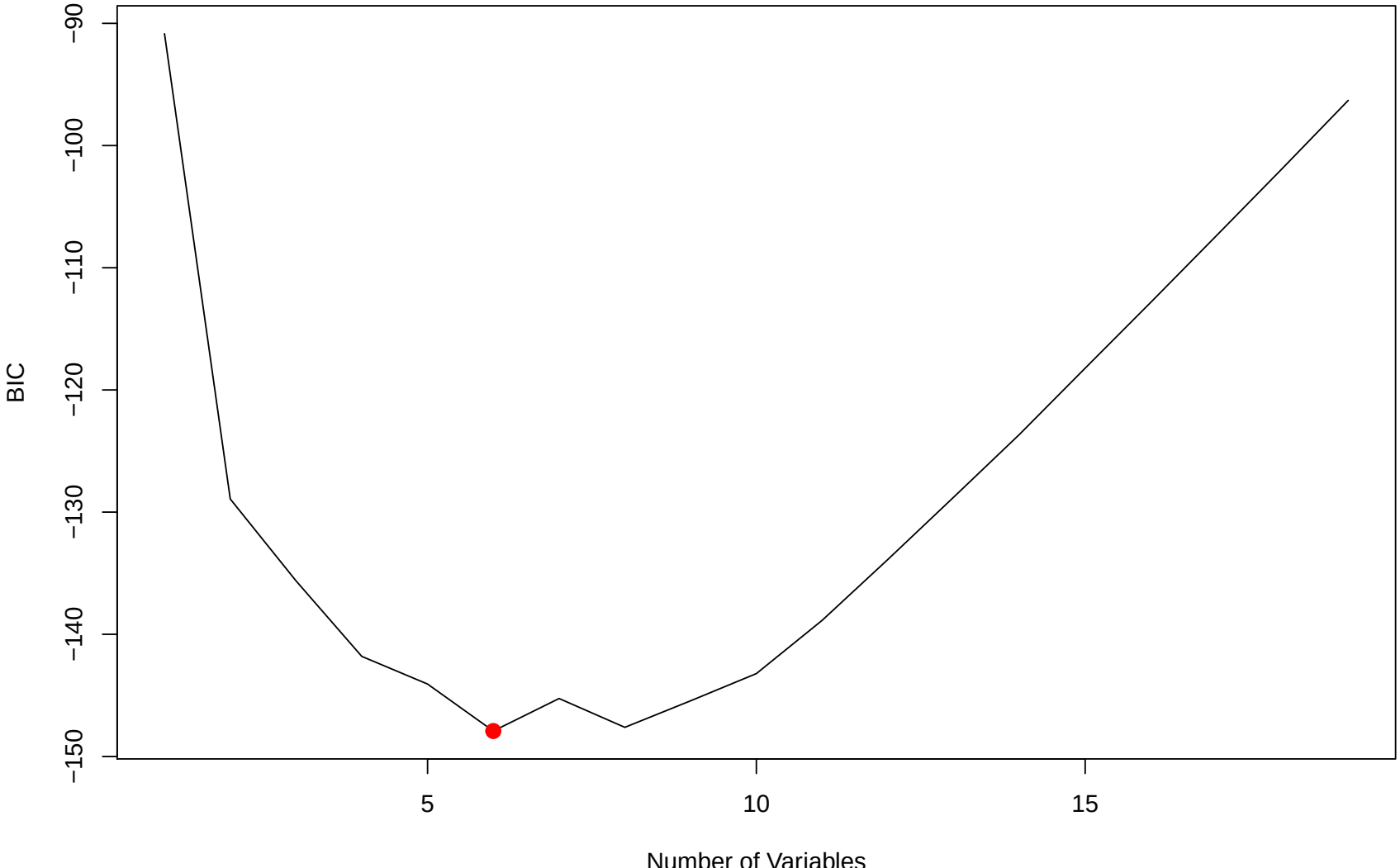
In a similar fashion we can plot the C_p and BIC statistics, and indicate the models with the smallest statistic using which.min().

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



C_p and BIC

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

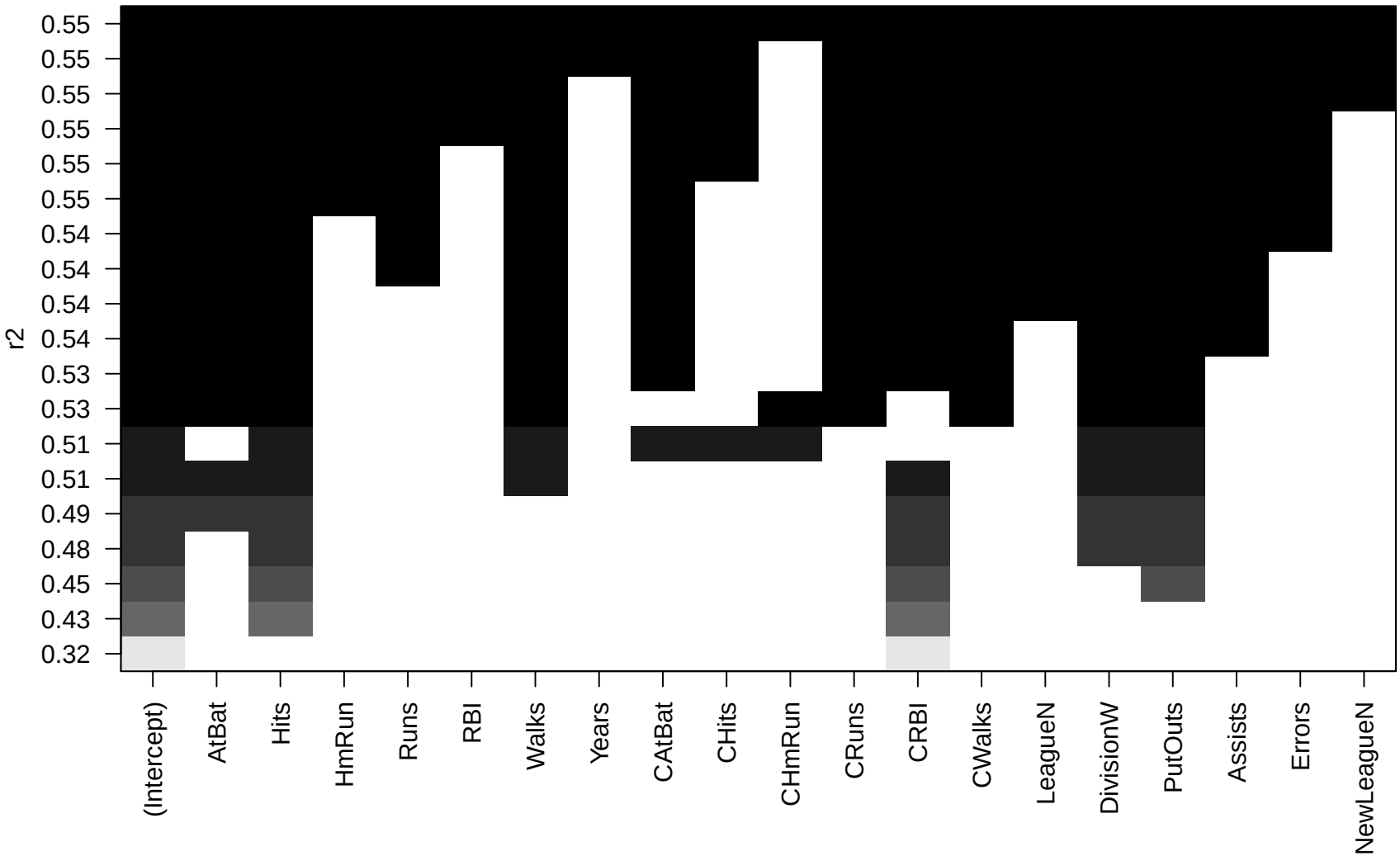


BIC, C_p, adjusted R^2, or AIC

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, ranked according to the BIC, C_p, adjusted R^2, or AIC. To find out more about this function, type ?plot.regsubsets

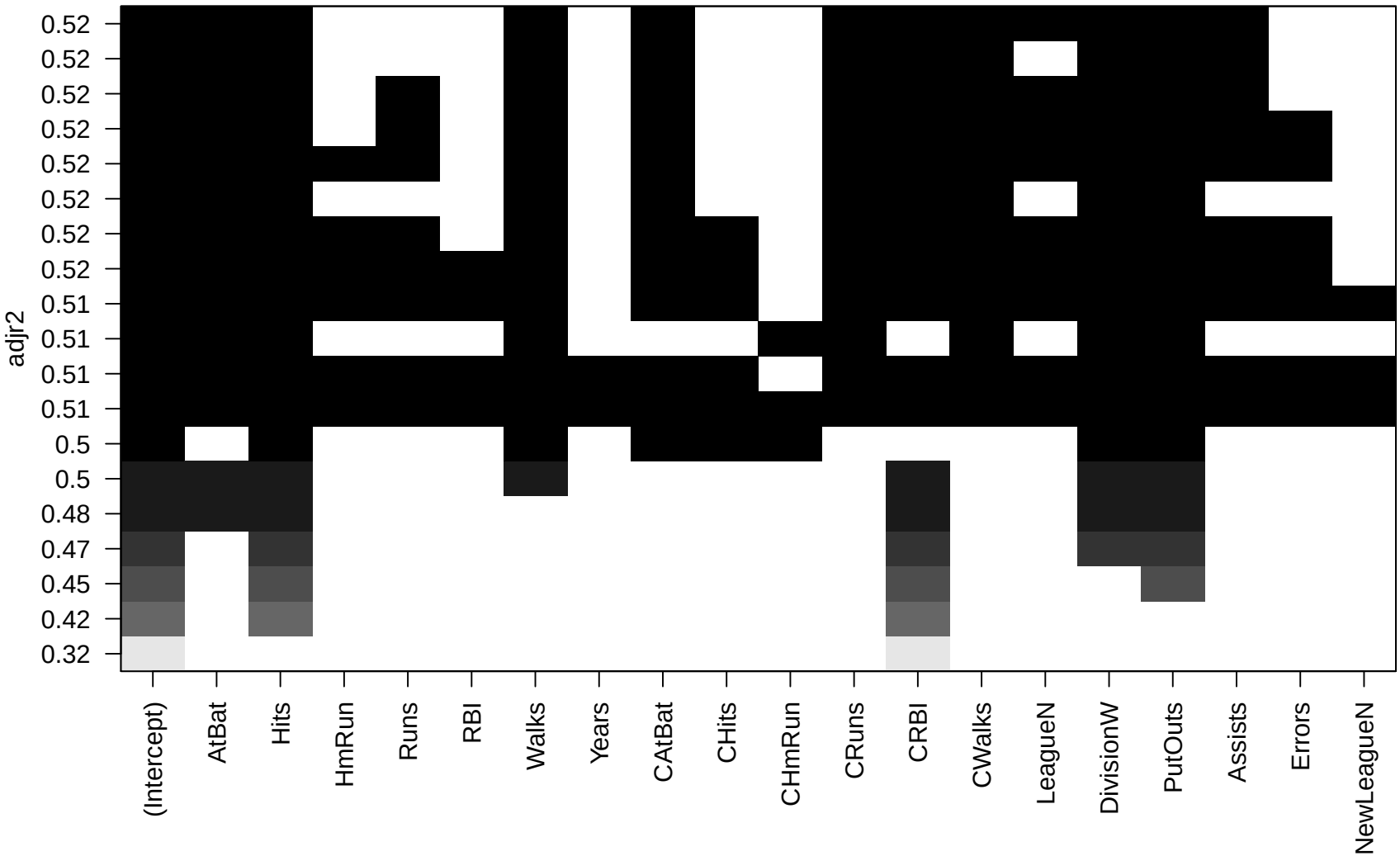
BIC, C_p, adjusted R^2, or AIC

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



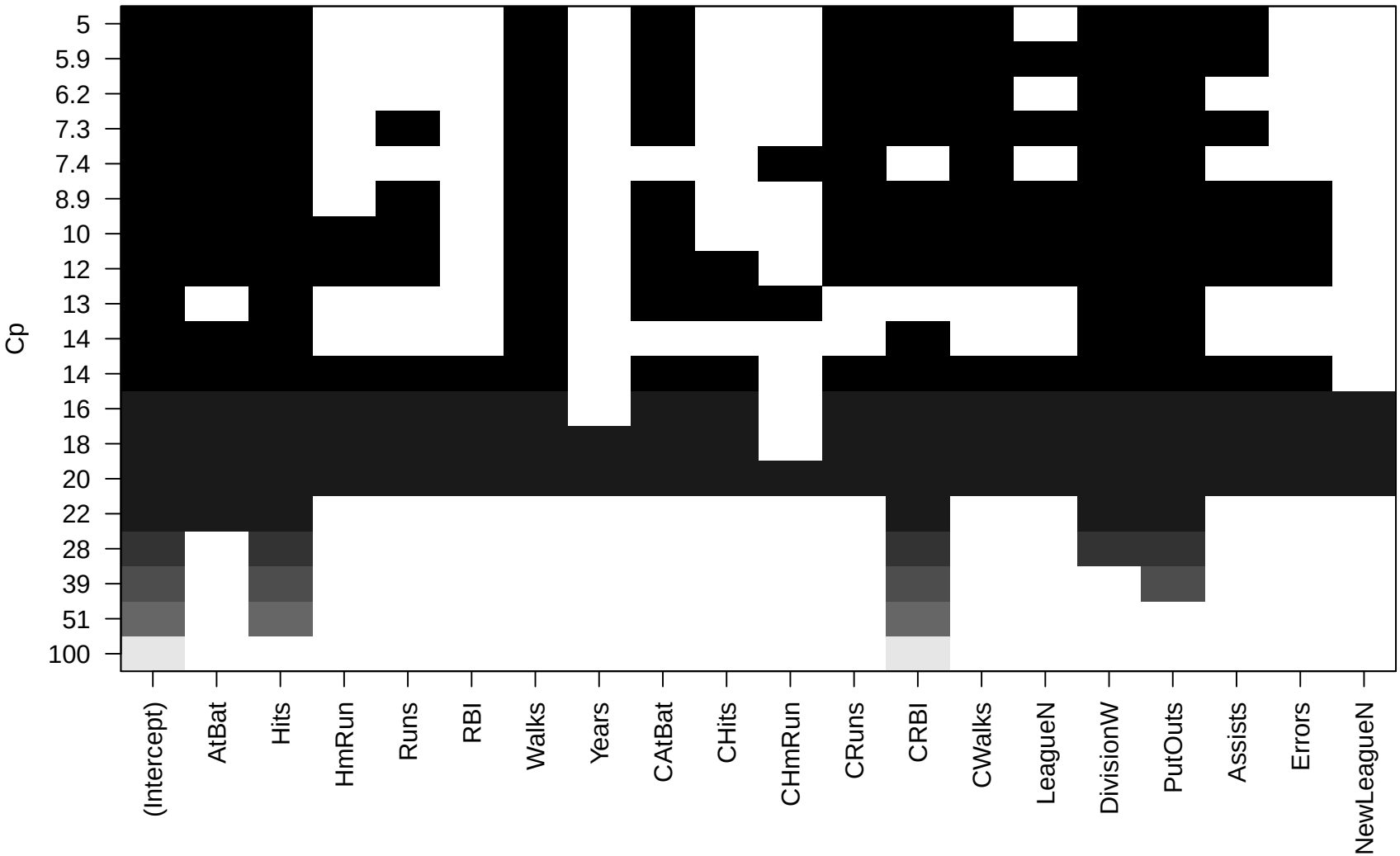
BIC, C_p, adjusted R^2, or AIC

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



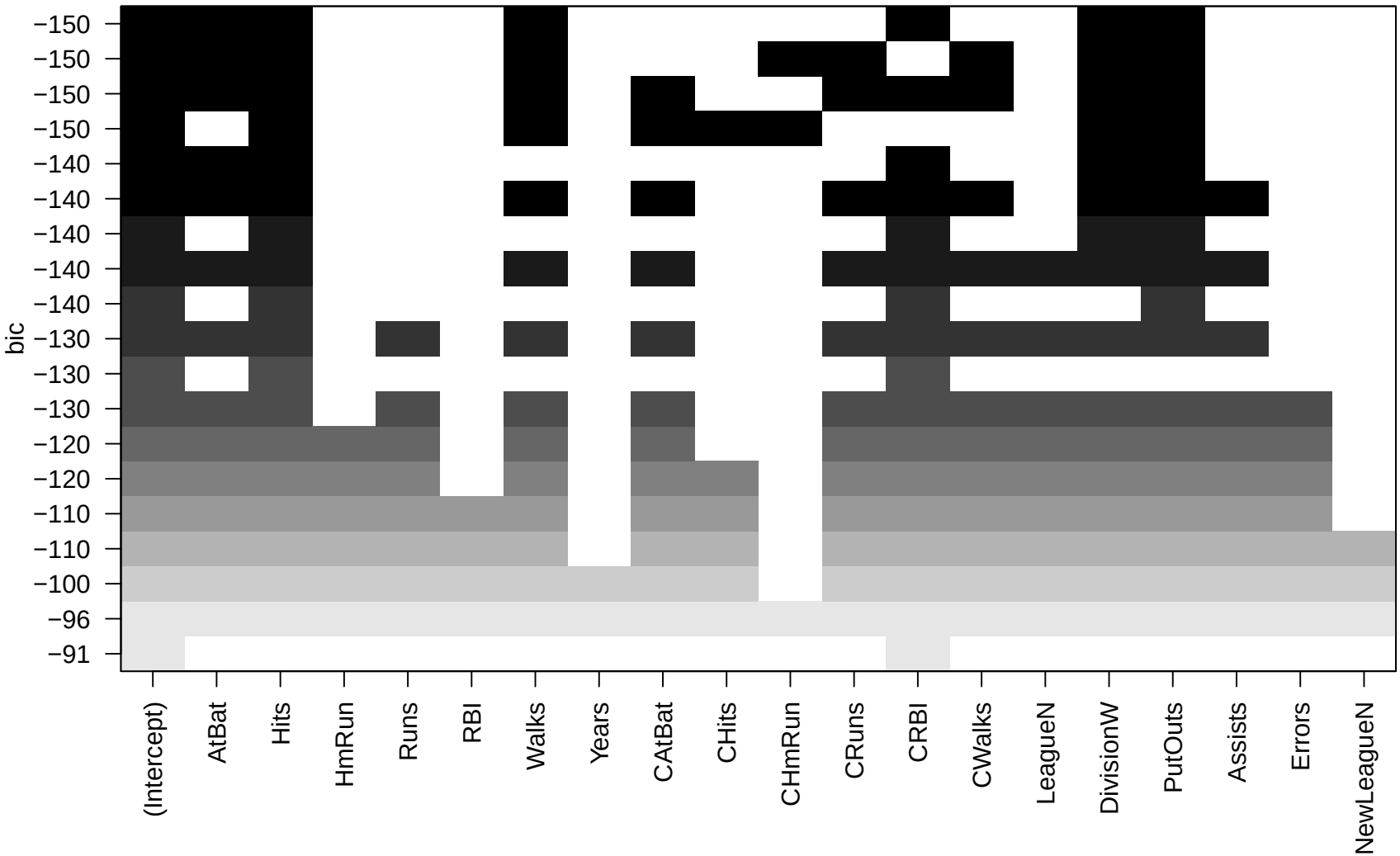
BIC, C_p, adjusted R^2, or AIC

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



BIC, C_p, adjusted R^2, or AIC

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



BIC, C_p, adjusted R^2, or AIC

The top row of each plot contains a black square for each variable selected according to the optimal model associated with that statistic.

- For instance, we see that several models share a BIC close to -150.
- However, the model with the lowest BIC is the six-variable model that contains only AtBat, Hits, Walks, CRBI, DivisionW, and PutOuts.
- We can use the coef() function to see the coefficient estimates associated with this model.

BIC, C_p, adjusted R^2, or AIC

```
coef(regfit.full, 6)
```

```
##  (Intercept)        AtBat         Hits        Walks
##   91.5117981   -1.8685892    7.6043976    3.6976468    (
##    DivisionW      PutOuts
## -122.9515338    0.2643076
```

Forward and Backward Stepwise Selection Applied to Hitters data set

We can also use the regsubsets() function to perform forward stepwise or backward stepwise selection, using the argument method =“forward” or method =“backward”

Forward and Backward Stepwise Selection Applied to Hitters data set

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

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvm
## 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
```

Forward and Backward Stepwise Selection Applied to Hitters data set

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

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvm
## 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
```

Forward and Backward Stepwise Selection Applied to Hitters data set

Forward Stepwise Selection:

- The best one-variable model contains only CRBI
- The best two-variable model additionally includes Hits.

Forward and Backwards Stepwise Selection: - For this data, the best one-variable through six-variable models are each identical for best subset and forward selection.

- However, the best seven-variable models identified by forward stepwise selection, backward stepwise selection, and best subset selection are different.
- Explore these more for practice.

Forward and Backward Stepwise Selection Applied to Hitters data set

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

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

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

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

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

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

We just saw that it is possible to choose among a set of models of different sizes using C_p, BIC, and adjusted R^2.

We will now consider how to do this using the validation set and cross-validation approaches.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

In order for these approaches to yield accurate estimates of the test error, we must use only the **training observations** to perform all aspects of model-fitting—including variable selection.

This point is subtle but important.

If the **full data set** is used to perform the best subset selection step, the validation set errors and cross-validation errors that we obtain **will not be accurate estimates** of the test error.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

In order to use the validation set approach, we begin by splitting the observations into a training set and a test set.

We do this by creating a random vector, train, of elements equal to TRUE if the corresponding observation is in the training set, and FALSE otherwise.

```
set.seed (1)
train <- sample(c(TRUE,FALSE), nrow(Hitters),rep=TRUE)
test <- (!train )
```

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

Now, we apply regsubsets() to the training set in order to perform best subset selection.

```
regfit.best=regsubsets(Salary~.,data=Hitters[train,], nvmax
```

Notice that we subset the Hitters data frame directly in the call in order to access only the training subset of the data, using the expression Hitters[train,]

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

We now compute the validation set error for the best model of each model size. We first make a model matrix from the test data.

```
test.mat=model.matrix(Salary~.,data=Hitters[test,])
```

The model.matrix() function is used in many regression packages for building an "X" matrix from data.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

Now we run a loop, and for each size i, we extract the coefficients from regfit.best for the best model of that size, multiply them into the appropriate columns of the test model matrix to form the predictions, and compute the test MSE.

```
val.errors=rep(NA,19)
for(i in 1:19){
  coefi=coef(regfit.best,id=i)
  pred=test.mat[,names(coefi)]%*%coefi
  val.errors[i]=mean((Hitters$Salary[test]-pred)^2)
}
```

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

We find that the best model is the one that contains ten variables.

```
val.errors
```

```
##  [1] 220968.0 169157.1 178518.2 163426.1 168418.1 17127(
##  [8] 157909.3 154055.7 148162.1 151156.4 151742.5 15221(
## [15] 158541.4 158743.3 159972.7 159859.8 160105.6
```

This was a little tedious, partly because there is no predict() method for regsubsets().

Since we will be using this function again, we can capture our steps above and write our own predict method.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

```
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
  }
```

Our function pretty much mimics what we did above. The only complex part is how we extracted the formula used in the call to regsubsets(). We demonstrate how we use this function below, when we do cross-validation.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

Finally, we perform best subset selection on the full data set, and select the best ten-variable model.

It is important that we make use of the full data set in order to obtain more accurate coefficient estimates.

Note that we perform best subset selection on the full data set and select the best ten-variable model, rather than simply using the variables that were obtained from the training set, because the best ten-variable model on the full data set may differ from the corresponding model on the training set.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

```
regfit.best=regsubsets(Salary~.,data=Hitters ,nvmax=19)
coef(regfit.best ,10)
```

```
##  (Intercept)        AtBat         Hits        Walks
##  162.5354420   -2.1686501    6.9180175    5.7732246   -0
##        CRuns         CRBI       CWalks    DivisionW
##    1.4082490    0.7743122   -0.8308264 -112.3800575    0
##      Assists
##    0.2831680
```

In fact, we see that the best ten-variable model on the full data set has a different set of variables than the best ten-variable model on the training set.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

We now try to choose among the models of different sizes using cross-validation.

First, we create a vector that allocates each observation to one of k = 10 folds, and we create a matrix in which we will store the results.

```
k=10
set.seed (1)
folds=sample(1:k,nrow(Hitters),replace=TRUE)
cv.errors=matrix(NA,k,19, dimnames=list(NULL, paste(1:19)))
```

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

Now we write a for loop that performs cross-validation.

In the jth fold, the elements of folds that equal j are in the test set, and the remainder are in the training set.

We make our predictions for each model size (using our new predict() method), compute the test errors on the appropriate subset, and store them in the appropriate slot in the matrix cv.errors.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

```
for(j in 1:k){
best.fit= regsubsets(Salary~.,data=Hitters[folds!=j,], nvma
  for (i in 1:19){
  pred=predict(best.fit,Hitters[folds==j,],id=i)
  cv.errors[j,i] = mean((Hitters$Salary[folds==j]-pred)^2)
  }
}
```

This has given us a 10×19 matrix, of which the (i, j)th element corresponds to the test MSE for the ith cross-validation fold for the best j-variable model.

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

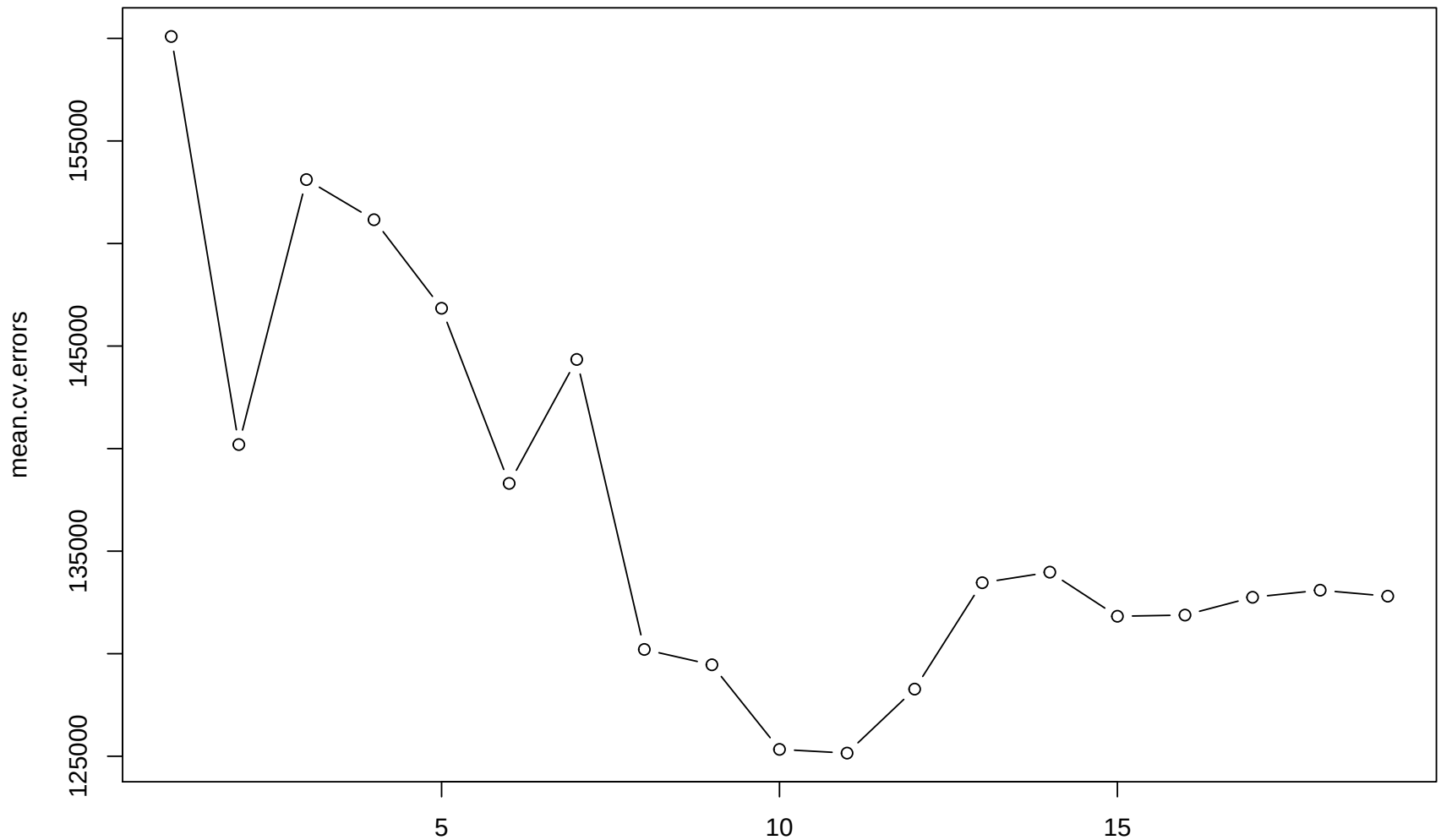
We use the apply() function to average over the columns of this matrix in order to obtain a vector for which the jth element is the cross-validation error for the j-variable model.

```
mean.cv.errors=apply(cv.errors ,2, mean)
mean.cv.errors
```

```
##        1        2        3        4        5        6
## 160093.5 140196.8 153117.0 151159.3 146841.3 138302.6 14
##        9       10       11       12       13       14
## 129459.6 125334.7 125153.8 128273.5 133461.0 133974.6 13
##       17       18       19
## 132750.9 133096.2 132804.7
```

Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

```
par(mfrow=c(1,1))
plot(mean.cv.errors ,type="b")
```



Choosing Among Models Using the Validation Set Approach and Cross-Validation for Hitters Data Set

We now perform best subset selection on the full data set in order to obtain the 11-variable model.

```
reg.best=regsubsets (Salary~.,data=Hitters , nvmax=19)
coef(reg.best ,11)
```

```
##  (Intercept)        AtBat         Hits        Walks
##  135.7512195   -2.1277482    6.9236994    5.6202755   -0
##        CRuns         CRBI       CWalks      LeagueN     D
##    1.4553310    0.7852528   -0.8228559   43.1116152 -11
##      PutOuts      Assists
##    0.2894087    0.2688277
```