Lab Assignment #8

Nick Noel and Liz Villa

Due April 5, 2023

Instructions

The purpose of this lab is to introduce model selection in regression.

```
library(tidyverse)
library(ISLR2)
library(ggplot2)
library(dplyr)
library(leaps)

madden17_QB <- readr::read_csv("madden17_QB.csv")
```

This lab assignment is worth a total of **15 points**.

Problem 1: Model Selection Using Cross-Validation

Part a (Code: 1 pt)

Run the code in ISLR Labs 5.3.2 and 5.3.3. Put each chunk from the textbook in its own chunk.

```
glm.fit <- glm(mpg ~ horsepower, data = Auto)
coef(glm.fit)
```

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

```
lm.fit <- lm(mpg ~ horsepower, data = Auto)
coef(lm.fit)
```

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

```
library(boot)
glm.fit <- glm(mpg ~ horsepower, data = Auto)
cv.err <- cv.glm(Auto, glm.fit)
cv.err$delta
```

```
## [1] 24.23151 24.23114
```

```
cv.error <- rep(0, 10)
for(i in 1:10){
  glm.fit <- glm(mpg ~ poly(horsepower, i), data = Auto)
  cv.error[i] <- cv.glm(Auto, glm.fit)$delta[1]
}
cv.error
```

```
##  [1] 24.23151 19.24821 19.33498 19.42443 19.03321 18.97864 18.83305 18.96115
##  [9] 19.06863 19.49093
```

```
set.seed(17)
cv.error.10 <- rep(0, 10)
for(i in 1:10){
  glm.fit <- glm(mpg ~ poly(horsepower, i), data = Auto)
  cv.error.10[i] <- cv.glm(Auto, glm.fit, K = 10)$delta[1]
}

cv.error.10
```

```
##  [1] 24.27207 19.26909 19.34805 19.29496 19.03198 18.89781 19.12061 19.14666
##  [9] 18.87013 20.95520
```

Part b (Code: 3 pts; Explanation: 1 pt)

Using the Auto dataset and 5-fold cross-validation, determine which of these sets of predictors produces the best linear model for predicting mpg, and explain your reasoning:

- horsepower and displacement
- acceleration and displacement
- horsepower and acceleration
- horsepower, acceleration, and displacement

Do not use the `cv.glm` function. Instead, modify the code in the "Automated Model Selection" class activity to do the cross-validation. Either the "Base R" or "tidymodels" example is fine to follow.

```
set.seed(17)

k <- 5

reorder_rows <- sample(nrow(Auto))

fold_numbers <- (reorder_rows %% k) + 1
```

```
model_MSE <- function(model, df, response){
  # model: a model object
  # df: a data frame on which we want to predict
  # response: a character vector giving the name of the response variable

  predictions <- predict(model, newdata = df)
  MSE <- mean((predictions - df[[response]])^2)
  return(MSE)
}
```

```
models <- vector("list", length = 4)
models[[1]] <- lm(mpg ~ horsepower + displacement, data = Auto)
models[[2]] <- lm(mpg ~ acceleration + displacement, data = Auto)
models[[3]] <- lm(mpg ~ horsepower + acceleration, data = Auto)
models[[4]] <- lm(mpg ~ horsepower + acceleration + displacement, data = Auto)
```

```
nmodels <- length(models)
cv_error <- matrix(0, nrow = k, ncol = nmodels)
# each row of cv_error represents a fold
# each column of cv_error represents a model
```

```
for (i in 1:k){
  fold_validation_rows <- which(fold_numbers == i)
  train_set <- Auto[-fold_validation_rows,]
  validation_set <- Auto[fold_validation_rows,]

  for(j in 1:nmodels){
    models[[j]] <- update(models[[j]], data = train_set)
    cv_error[i, j] <- model_MSE(models[[j]], df = validation_set, response = "mpg")
  }
}
```

```
cv_rmse <- sqrt(cv_error)
apply(cv_rmse, 2, mean)
```

```
## [1] 4.545213 4.632073 4.784674 4.496172
```

The best model based on lowest rmse is the model with all three predictors.

Part c (Code: 2 pts)

For the model you selected in part (b), re-fit the model on the entire `Auto` dataset. Then, write a couple of lines of code to compute C_p and BIC for this model (as given in the book) without relying on the `AIC`/`BIC` functions or any functions in the `olsrr` package. Some hints:

- You can obtain RSS by creating an `aov` object and running the code `summary(aov_object)[[1]]`, then finding the appropriate way to subset the resulting matrix.
- You can obtain $\hat{\sigma}$ for a model by running `summary(full_model)$sigma`. You may assume that the full model is the one with all three predictors.

```
lm4 <- lm(mpg ~ horsepower + acceleration + displacement, data = Auto)

aov_object <- aov(lm4)

summary(aov_object)[[1]]
```

```
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## horsepower      1 14433.1 14433.1 723.121 < 2.2e-16 ***
## acceleration    1   581.0   581.0  29.107 1.194e-07 ***
## displacement    1  1060.7  1060.7  53.143 1.754e-12 ***
## Residuals     388  7744.3    20.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RSS <- summary(aov_object)[[1]][4, 2]
```

```
sig_hat <- summary(lm4)$sigma
```

```
(Cp <- 1/nrow(Auto)*(RSS + 2*3*sig_hat^2))
```

```
## [1] 20.06126
```

```
(BIC <- 1/nrow(Auto)*(RSS + log(nrow(Auto))*3*sig_hat^2))
```

```
## [1] 20.66788
```

Problem 2: Subset Selection

Part a (Code: 1 pt)

Run the code in ISLR Lab 6.5.1, "Best Subset Selection" and "Forward and Backward Stepwise Selection" subsections. (Do not run the "Choosing Among Models Using the Validation-Set Approach and Cross-Validation" section.)

```
library(ISLR2)
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
```

```
Hitters <- na.omit(Hitters)
dim(Hitters)
```

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

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

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

```
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 ) "*"    " "     "*"       "*"     " "     " "    " "
```

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

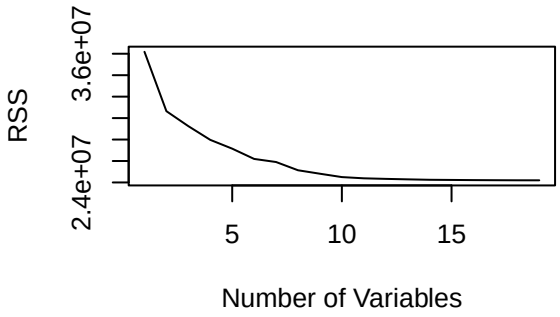
```
names(reg.summary)
```

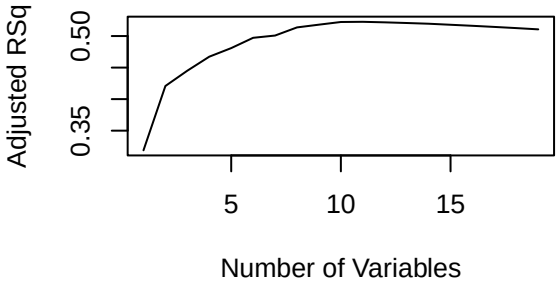
```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
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
```

```
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")
```

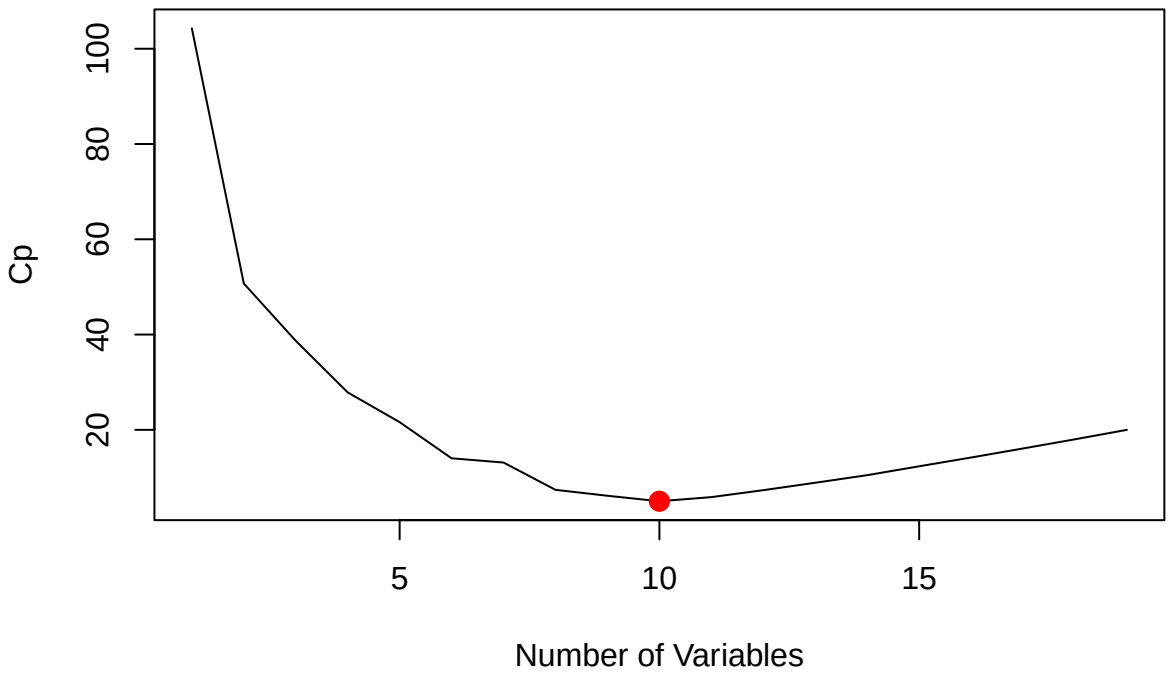





```
plot(reg.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
which.min(reg.summary$cp)
```

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

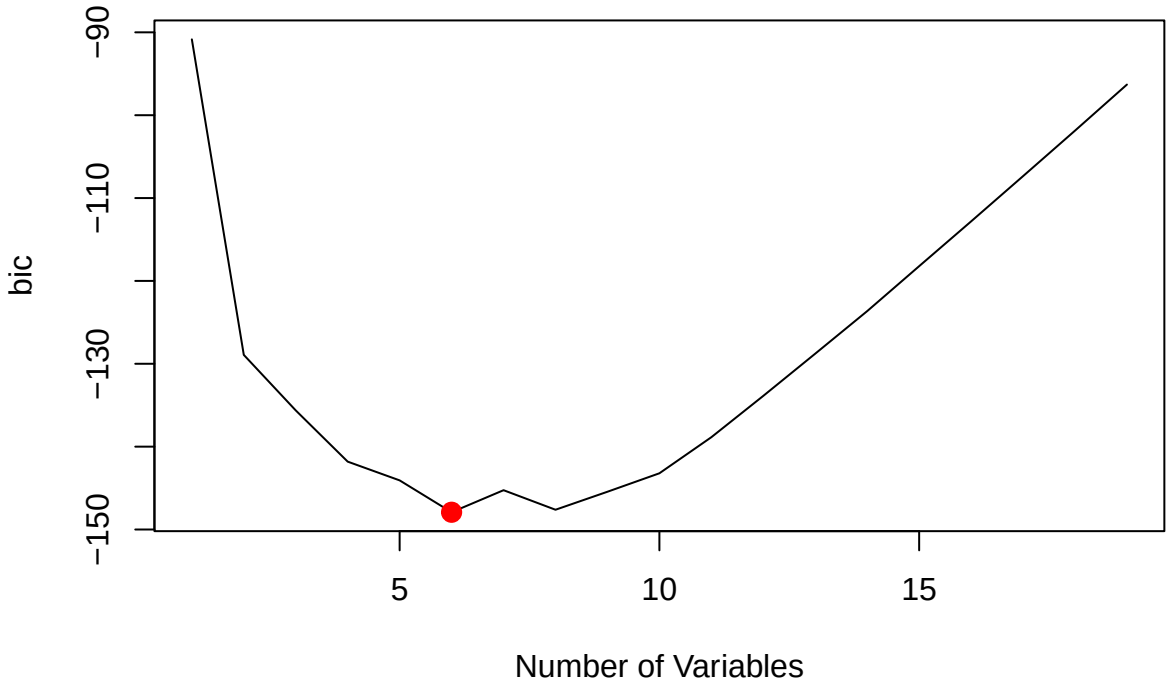
```
points(10, reg.summary$cp[10], col="red", cex = 2, pch = 20)
```



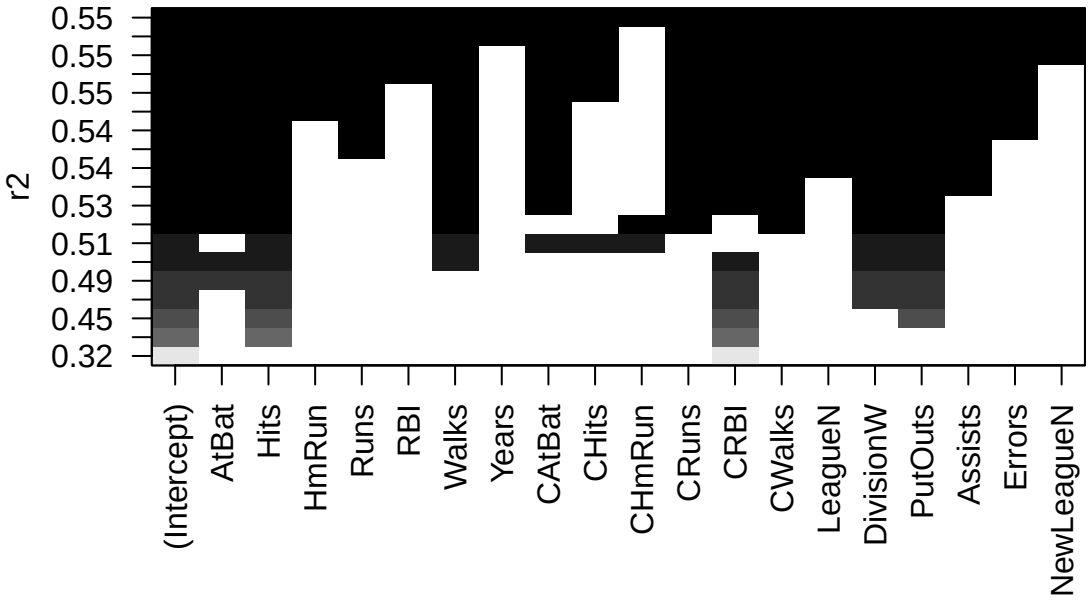

```
which.min(reg.summary$bic)
```

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

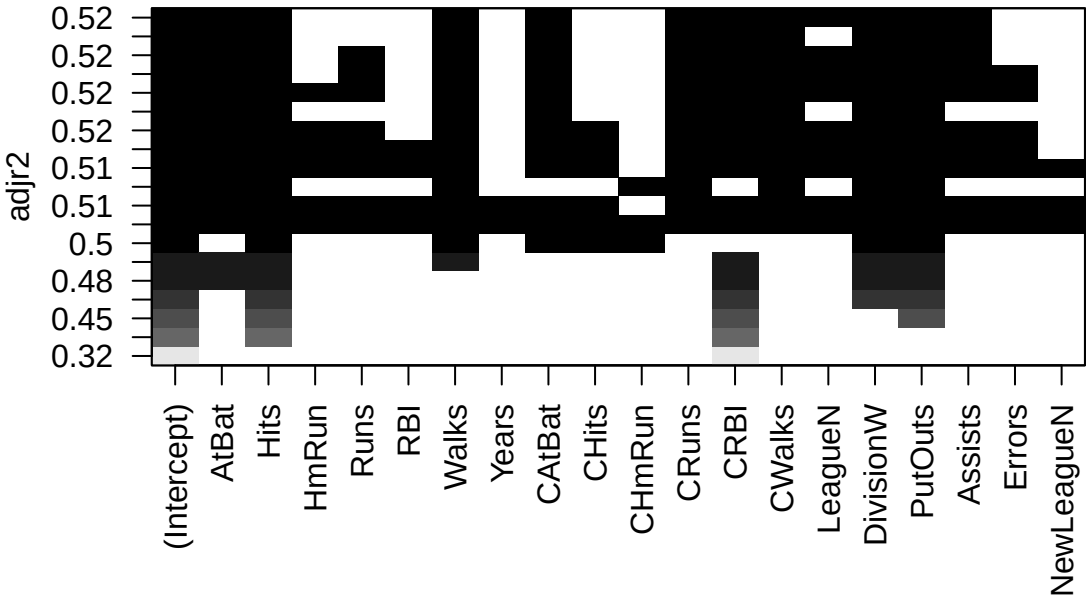
```
plot(reg.summary$bic, xlab = "Number of Variables", ylab = "bic", type = "l" )
points(6, reg.summary$bic[6],col="red", cex = 2, pch = 20 )
```



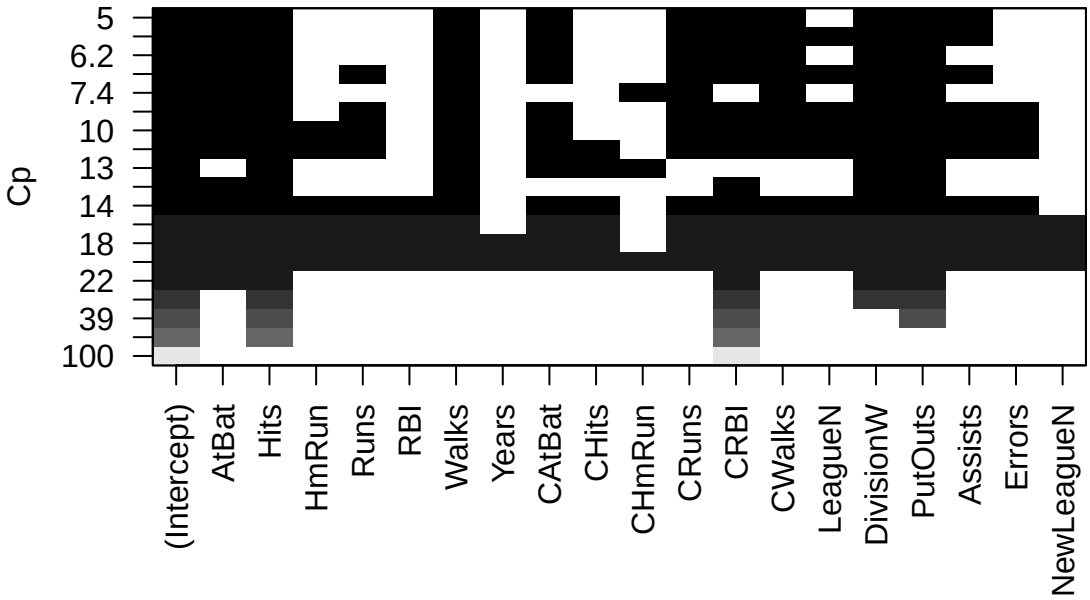

```
plot(regfit.full, scale = "r2")
```



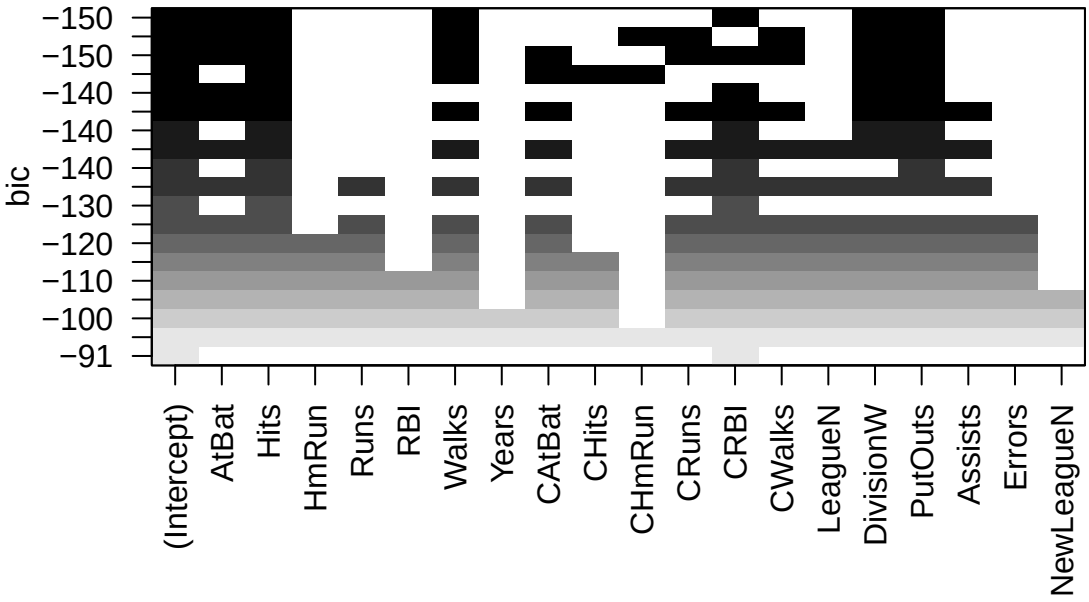

```
plot(regfit.full, scale = "adjr2")
```




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




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




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

```
##  (Intercept)        AtBat         Hits        Walks         CRBI    DivisionW
##   91.5117981   -1.8685892    7.6043976    3.6976468    0.6430169 -122.9515338
##      PutOuts
##    0.2643076
```

Forward and 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 ) "*"    "*"     "*"       "*"     "*"     "*"    "*"
```

```
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
```

Part b (Explanation: 1 pt)

Briefly explain how to interpret the plots created by `plot(regfit.full, scale = "some metric")` at the end of the Best Subset Selection section.

To interpret the plots you are looking for either the highest (R^2 and $adjusted R^2$) or lowest (Cp and BIC) values. These will be colored black on the plot which tells us what predictors are the most useful in terms of prediction. Thus for BIC the best model will be the one with the 6 variables that are at the top of the plot and colored black.

Part c (Code: 1 pt; Explanation: 1 pt)

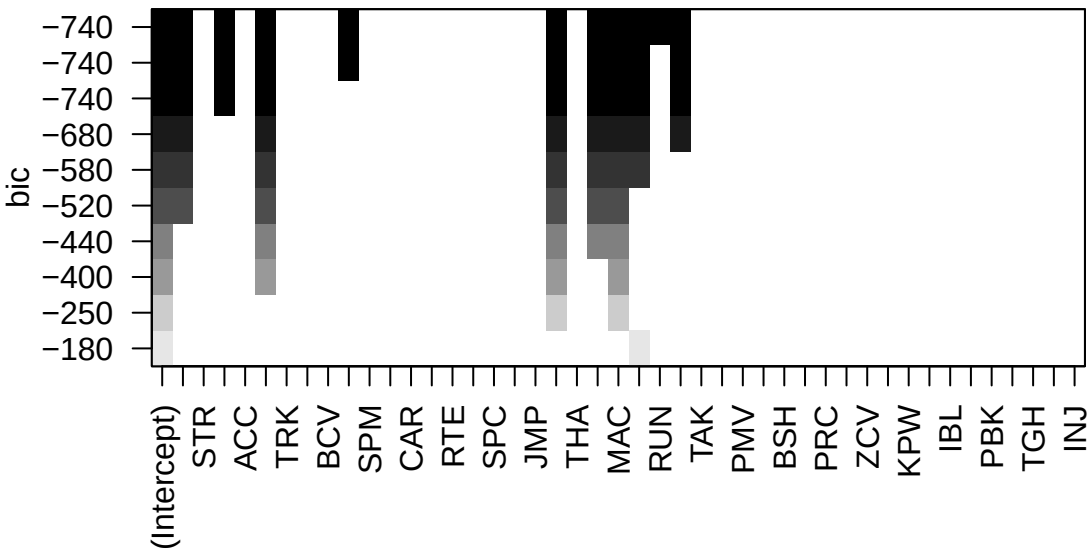
In the rest of this problem, we will explore a situation in which the true model is *known* (more-or-less). In this true model, however, the error term is due to rounding and is *not* normally distributed, and there are some major collinearity issues. Let's see whether these violations of least-squares assumptions affect subset selection.

The madden17_QB dataset contains the overall rating (`OVR`) and individual skill ratings for 112 quarterbacks in the Madden NFL 2017 video game. According to an article on fivethirtyeight.com, the overall rating for quarterbacks is a linear combination of the following skill ratings: `AWR`, `THP`, `SAC`, `MAC`, `DAC`, `PAC`, `SPD`, `AGI`, `RUN`, and `ACC`. The other 34 skill ratings are not relevant.

Perform best subset selection on this dataset, using `nvmax = 10`. You may have to remove the categorical variables (`Name` and `Team`) in the formula or the dataset used to fit the model.

```
the_big_game <- madden17_QB %>%
  select(-"Name", -"Team")

big_game_full_model <- regsubsets(OVR ~ ., data = the_big_game, nvmax = 10)
big_game_summary <- summary(big_game_full_model)
plot(big_game_full_model, scale = "bic")
```




```
coef(big_game_full_model,10)
```

```
##   (Intercept)           SPD           AGI           AWR           SFA
## -70.239820431   0.101469635   0.043056144   0.220717268  -0.007153152
##           THP           SAC           MAC           DAC           RUN
##   0.460989280   0.296144707   0.347966687   0.221341059   0.022022327
##           PAC
##   0.071344772
```

Did the algorithm correctly identify the 10 important variables in the model? If not, which variables were incorrectly left out, and which were incorrectly included?

SFA from literature was not significant but was chosen in our model, the algorithm also removed the variable we expected to be significant, ACC.

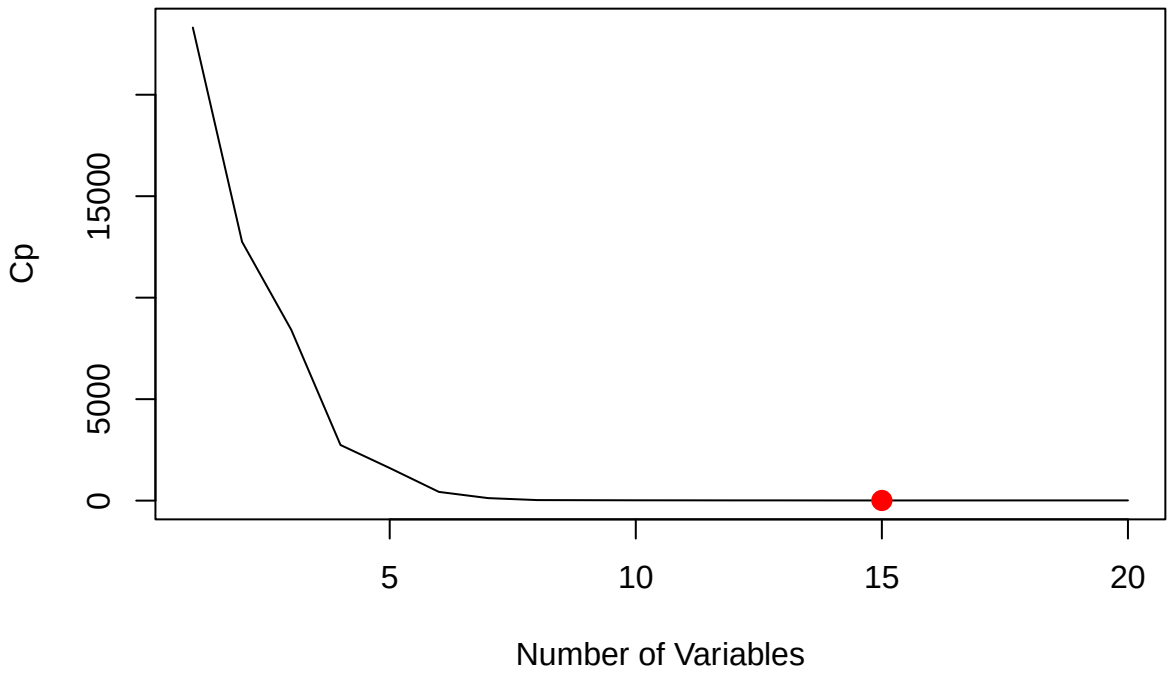
Part d (Code: 1 pt; Explanation: 1 pt)

Perform forward selection on this dataset, using `nvmax = 20`. How many variables are in the "best" model using BIC as a selection criterion? What about Cp? For the "best" model (using one of the criteria), which variables were incorrectly left out, and which were incorrectly included?

```
the_big_game.fwd <- regsubsets(OVR~., data = the_big_game, nvmax = 20, method = "forward")
big_summary_fwd <- summary(the_big_game.fwd)
plot(big_summary_fwd$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
which.min(big_summary_fwd$cp)
```

```
## [1] 15
```

```
points(15, big_summary_fwd$cp[15], col="red", cex = 2, pch = 20)
```



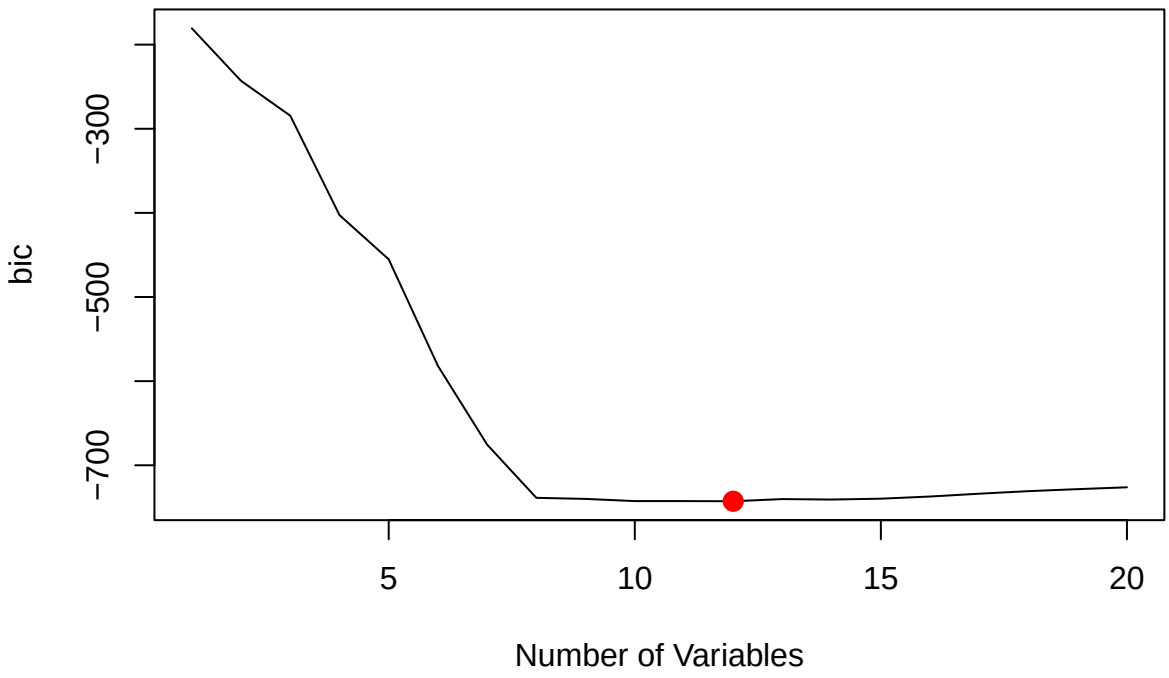

```
coef(the_big_game.fwd, 15)
```

```
##   (Intercept)           SPD           AGI           ACC           AWR
## -70.353411999   0.088560782   0.038007682   0.019369509   0.222045670
##           SFA           CTH           SPC           RLS           THP
##  -0.008792997   0.003763186   0.007214520  -0.011167750   0.457919896
##           SAC           MAC           DAC           RUN           PAC
##   0.297640916   0.338856202   0.224475043   0.021510792   0.074114194
##           PBK
##   0.014974778
```

```
plot(big_summary_fwd$bic, xlab = "Number of Variables", ylab = "bic", type = "l")
which.min(big_summary_fwd$bic)
```

```
## [1] 12
```

```
points(12, big_summary_fwd$bic[12], col="red", cex = 2, pch = 20)
```




```
coef(the_big_game.fwd, 12)
```

```
##   (Intercept)           SPD           AGI           ACC           AWR
## -70.905427945   0.089091453   0.040224551   0.017672577   0.221273932
##           SFA           THP           SAC           MAC           DAC
##  -0.008137894   0.458856321   0.299879394   0.344153038   0.222862467
##           RUN           PAC           PBK
##   0.023763655   0.072649942   0.012927661
```

BIC had 12 significant variables and cp had 15 variables. Both BIC and cp had all 10 variables that previous literature mentioned but BIC also incorrectly included SFA and PBK and cp also incorrectly included SFA, CTH, SPC, RLS, PBK.

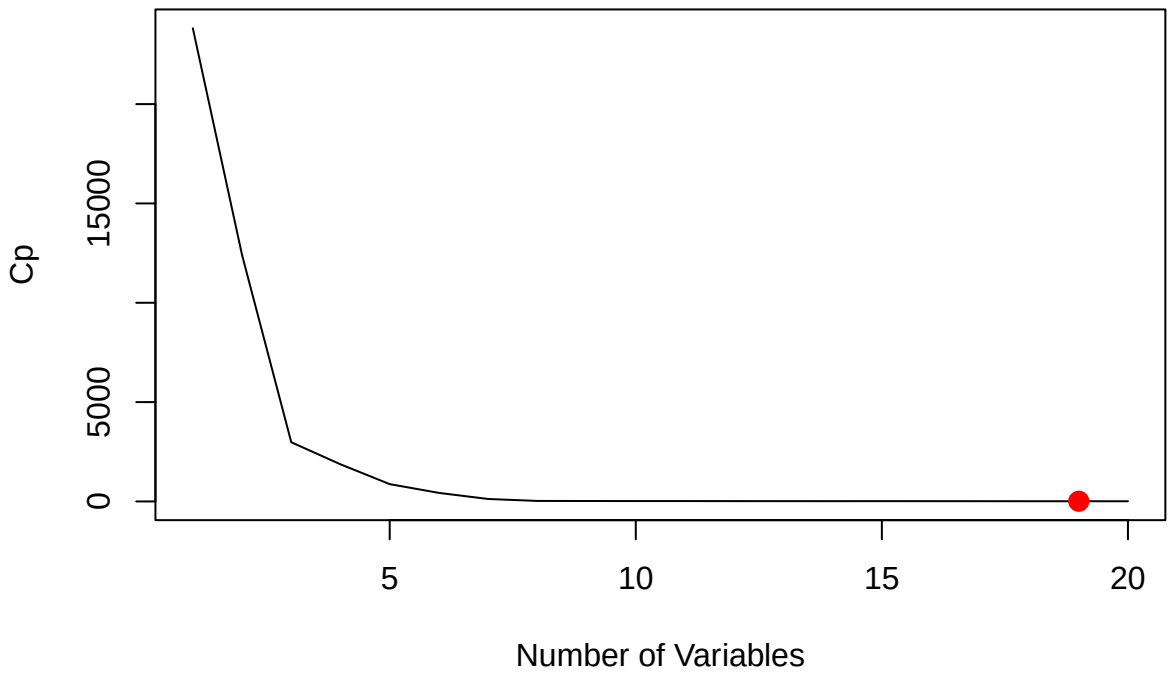
Part e (Code: 1 pt; Explanation: 1 pt)

Perform backward selection on this dataset, using `nvmax = 20`. How many variables are in the "best" model using BIC as a selection criterion? What about Cp? For the "best" model (using one of the criteria), which variables were incorrectly left out, and which were incorrectly included?

```
the_big_game.bwd <- regsubsets(OVR~., data = the_big_game, nvmax = 20, method = "backward")
big_summary_bwd <- summary(the_big_game.bwd)
plot(big_summary_bwd$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
which.min(big_summary_bwd$cp)
```

```
## [1] 19
```

```
points(19, big_summary_bwd$cp[19], col="red", cex = 2, pch = 20)
```



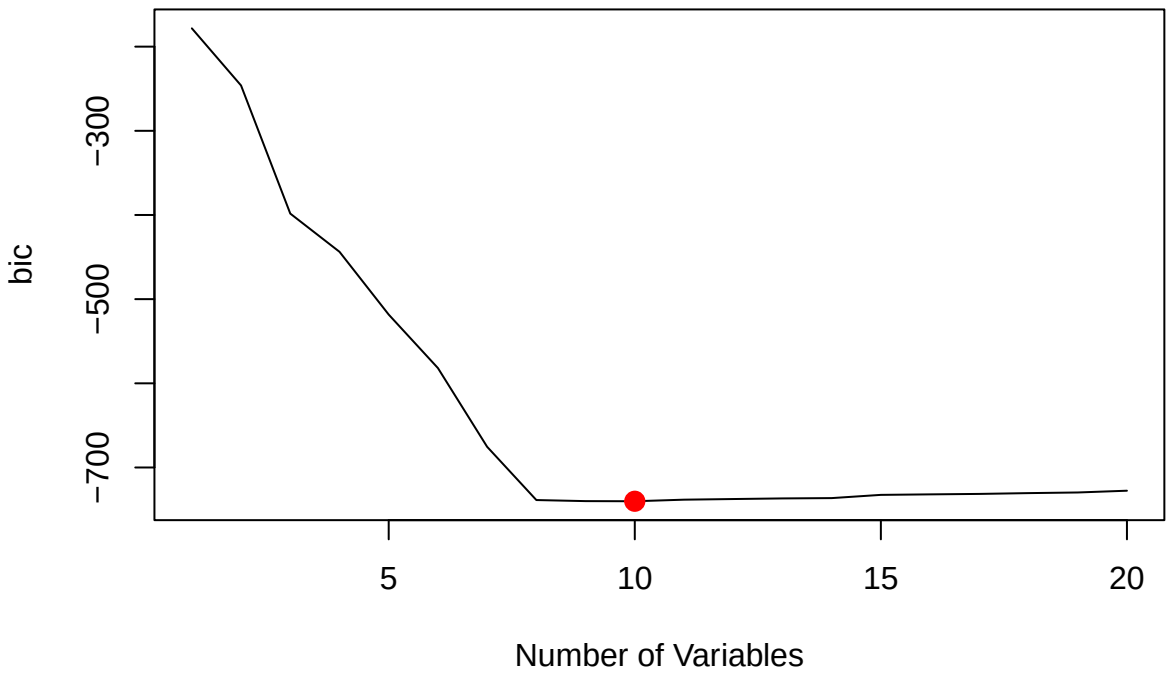

```
coef(the_big_game.bwd, 19)
```

```
##   (Intercept)           SPD           AGI           AWR           ELU
## -69.381429626   0.096921377   0.039415630   0.220213079   0.006968571
##           BCV           SFA           CTH           RLS           THP
##  -0.005789746  -0.007057549   0.007170439  -0.008422955   0.460327130
##           SAC           MAC           DAC           RUN           PAC
##   0.296560947   0.339900373   0.225978471   0.018277421   0.071283056
##           POW           KPW           KAC           RBK           PBK
##   0.014701498  -0.019810349   0.016501510  -0.008585389   0.023352179
```

```
plot(big_summary_bwd$bic, xlab = "Number of Variables", ylab = "bic", type = "l")
which.min(big_summary_bwd$bic)
```

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

```
points(10, big_summary_bwd$bic[10], col="red", cex = 2, pch = 20)
```




```
coef(the_big_game.fwd, 10)
```

```
##   (Intercept)           SPD           AGI           AWR           SFA
## -70.239820433   0.101469635   0.043056144   0.220717268  -0.007153152
##           THP           SAC           MAC           DAC           RUN
##   0.460989280   0.296144707   0.347966687   0.221341059   0.022022327
##           PAC
##   0.071344772
```

The model that used Cp as a measures had 19 variables and BIC had 10 variables. Both measurements failed to include ACC which we would expect to be in the model based on previous literature. Cp also had the additional variables ELU, BCV, SFA, CTH, RLS, POW, KPW, KAC, RBK, and PBK. For BIC, it had the predictor SFA incorrectly included and ACC incorrectly left out.