Lab 5

Erin Canada

November 16, 2018

This dataset records the salary of n= 263 Major League Baseball players during the 1987 season as well as q= 19 statistics associated with the performance of each player during the previous season. Specifically, the dataset contains observations from the following variables:

. AtBat: Number of times at bat in 1986 . Hits: Number of hits in 1986 . HmRun: Number of home runs in 1986 . Runs: Number of runs in 1986 . RBI: Number of runs batted in in 1986 . Walks: Number of walks in 1986 . Years: Number of years in the major leagues . CAtBat: Number of times at bat during his career . CHits: Number of hits during his career . CHmRun: Number of home runs during his career . CRuns: Number of runs during his career . CRBI: Number of runs batted in during his career . CWalks: Number of walks during his career . League: A categorical variable with levels A (for American) and N (for National) indicating the player's league at the end of 1986 . Division: A factor with levels E (for East) and W (for West) indicating the player's division at the end of 1986 . PutOuts: Number of put outs in 1986 . Assists: Number of assists in 1986 . Errors: Number of errors in 1986 . Salary: 1987 annual salary on opening day in thousands of dollars . NewLeague: A factor with levels A and N indicating the player's league at the beginning of 1987

Interest lies in developing a model that relates a player's annual salary to their previous performance. Your job in this Lab is to investigate several such models. Where computation is required, you must perform the calculations in both R and Python (unless otherwise indicated).

```
getwd()
```

```
## [1] "C:/Users/Erin Canada/Documents/USF/Fall 2018/Regression/Lab"
```

```
library(car)
```

```
## Loading required package: carData
```

```
hitter <- read.csv("hitters.csv")

head(hitter)
```

```
##   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
## 1   315   81     7   24  38    39    14   3449   835     69   321  414
## 2   479  130    18   66  72    76     3   1624   457     63   224  266
## 3   496  141    20   65  78    37    11   5628  1575    225   828  838
## 4   321   87    10   39  42    30     2    396   101     12    48   46
## 5   594  169     4   74  51    35    11   4408  1133     19   501  336
## 6   185   37     1   23   8    21     2    214    42      1    30    9
##   CWalks League Division PutOuts Assists Errors Salary NewLeague
## 1    375      N        W     632      43     10  475.0         N
## 2    263      A        W     880      82     14  480.0         A
## 3    354      N        E     200      11      3  500.0         N
## 4     33      N        E     805      40      4   91.5         N
## 5    194      A        W     282     421     25  750.0         A
## 6     24      N        E      76     127      7   70.0         A
```

(a) Calculate the variance inflation factor (VIF) for each of the explanatory variables.Comment on whether multicollinearity appears to be an issue. If it is, identify the three explanatory variables that are most seriously affected by the issue.

```
model <- lm(Salary ~ .,data = hitter)

vif(model)
```

```
##      AtBat       Hits      HmRun       Runs        RBI      Walks
##  22.944366  30.281255   7.758668  15.246418  11.921715   4.148712
##      Years     CAtBat      CHits     CHmRun      CRuns       CRBI
##   9.313280 251.561160 502.954289  46.488462 162.520810 131.965858
##     CWalks     League   Division    PutOuts    Assists     Errors
##  19.744105   4.134115   1.075398   1.236317   2.709341   2.214543
##  NewLeague
##   4.099063
```

```
collinearity <- c(vif(model))

multi <- sort(collinearity)
multi
```

```
##   Division    PutOuts     Errors    Assists  NewLeague     League
##   1.075398   1.236317   2.214543   2.709341   4.099063   4.134115
##      Walks      HmRun      Years        RBI       Runs     CWalks
##   4.148712   7.758668   9.313280  11.921715  15.246418  19.744105
##      AtBat       Hits     CHmRun       CRBI      CRuns     CAtBat
##  22.944366  30.281255  46.488462 131.965858 162.520810 251.561160
##      CHits
## 502.954289
```

```
tail(multi,3)
```

```
##    CRuns   CAtBat    CHits
## 162.5208 251.5612 502.9543
```

It seems as though multicollinearity appears to be an issue. Three explanatory variables seriously affected by this issue are CHits, CAtBat, and CRuns. Each of these explanatory variables have a variance inflation factor of well above a range of 5 or 10, which indicates multicollinearity.

(b) Using the all-possible-subsets approach, find the model that best fits the observed data. This procedure may be automated using the regsubsets() function in R, but you must explain in your own words how this algorithm identifies the 'best'model. Note that you do not need to perform this task in Python.

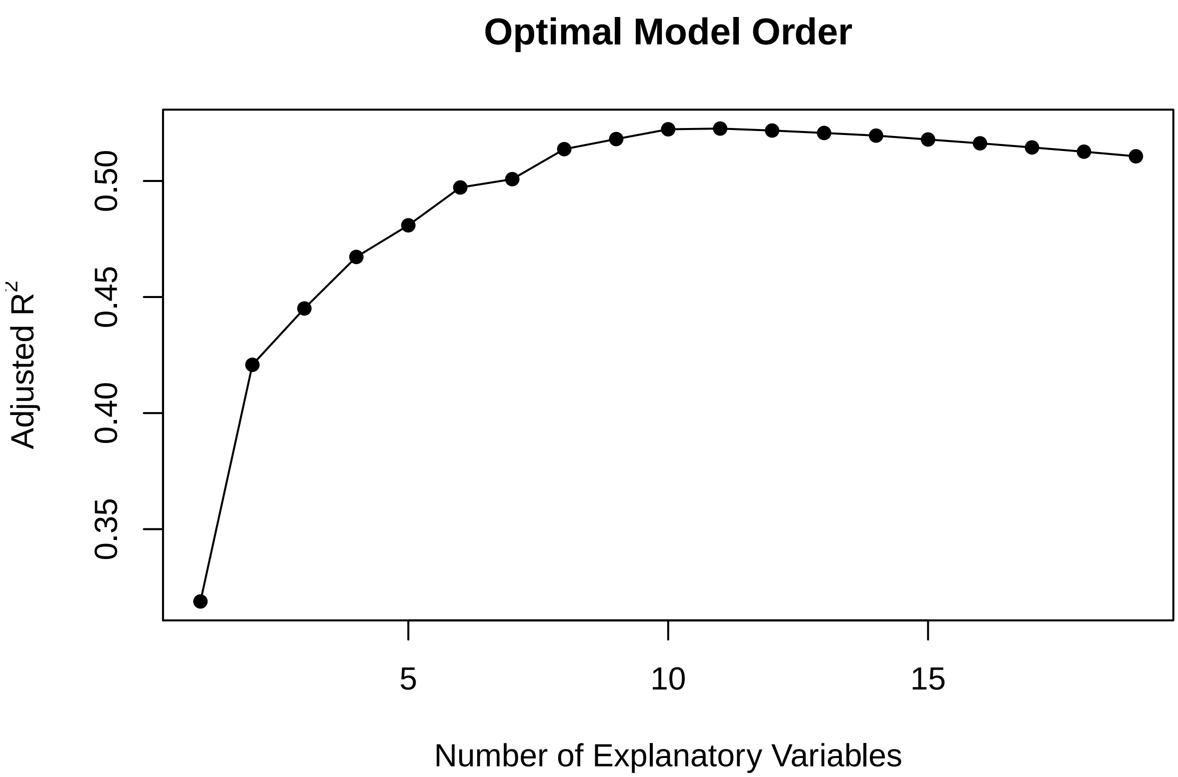
```
# Fit all possible models and for a given number of explanatory variables, find
# the best model (in terms of R^2)
library(leaps)
all_poss <- regsubsets(Salary ~ ., data = hitter, nvmax = 19)
all_poss_summ <- summary(all_poss)
all_poss_summ
```

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = hitter, nvmax = 19)
```

```
## 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: 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 )  "*"   "*"  " "   " "  " " "*"   " "   " "    " "   "*"    "*"  
## 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 ) "*"  "*"    "*"     "*"       "*"     "*"     "*"    "*"
```

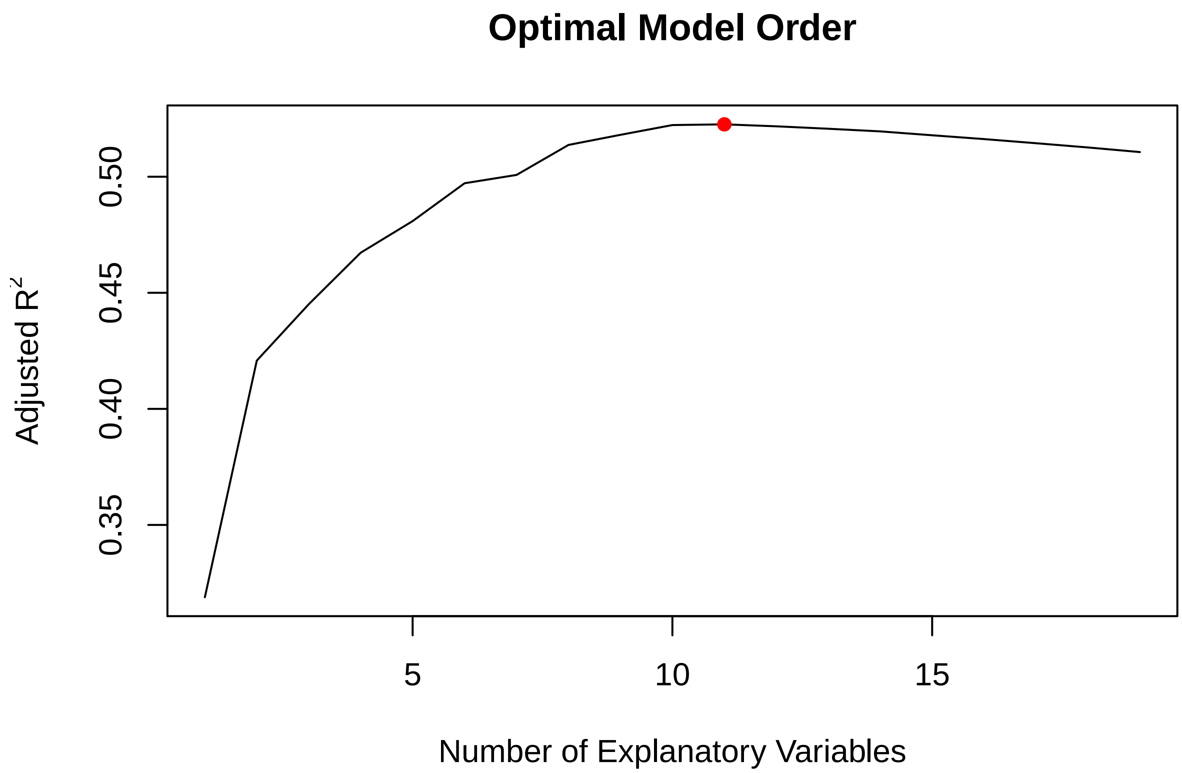
```
# Plot the Adjusted R^2 for each of these
plot(all_poss_summ$adjr2, type = "l", xlab = "Number of Explanatory Variables",
     ylab = bquote("Adjusted R"^2), main = "Optimal Model Order")
points(all_poss_summ$adjr2, pch = 16)
```




```
# Find the optimal number of explanatory variables
max_idx <- which.max(all_poss_summ$adjr2)
max_idx
```

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

```
plot(all_poss_summ$adjr2, type = "l", xlab = "Number of Explanatory Variables",
     ylab = bquote("Adjusted R"^2), main = "Optimal Model Order")
points(x = max_idx, y = all_poss_summ$adjr2[max_idx], pch = 16, col = "red")
```




```r
all_poss_summ$which[max_idx,]
```

```
## (Intercept)       AtBat        Hits       HmRun        Runs         RBI
##        TRUE        TRUE        TRUE       FALSE       FALSE       FALSE
##       Walks       Years      CAtBat       CHits      CHmRun       CRuns
##        TRUE       FALSE        TRUE       FALSE       FALSE        TRUE
##        CRBI      CWalks     LeagueN   DivisionW     PutOuts     Assists
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##      Errors  NewLeagueN
##       FALSE       FALSE
```

```r
m_all <- lm(Salary ~ AtBat + Hits + Walks+ CAtBat + CRuns+ CRBI
            + CWalks + League + Division + PutOuts + Assists, data = hitter)
```

The all possible subsets compares each of the possible models' adjusted R square values. This approach is done in two stages. For a given number of explanatory variables, we choose the best model using R^2 which yields q + 1 optimal models. Next, these models are compared to one another using the adjusted R square which incorporates a penalty for including too many explanatory variables. In this case, we compare each of the possible models to one another and chose the best one. The best model using the all possible subsets includes the variables AtBat,Hits,Walks, CAtBat,CRuns,CRBIC, Walks,League,Division,PutOuts, and Assists. According to this approach, these variables will create the best model to use in predicting future salaries.

(c) Using the forward-stepwise-selection approach, find the model that best fits the observed data. This procedure may be automated using the stepAIC() function in R, but you must explain in your own words how this algorithm identifies the 'best' model. Note that you do not need to perform this task in Python.

```
library(MASS)
sml <- lm(Salary ~ 1, data = hitter)
lrg <- lm(Salary ~ ., data = hitter)

# Forward
 stepAIC(object = sml, scope = list(upper = lrg, lower = sml), direction = "forward", trace = 0)
```

```
##
## Call:
## lm(formula = Salary ~ CRBI + Hits + PutOuts + Division + AtBat +
##     Walks + CWalks + CRuns + CAtBat + Assists, data = hitter)
##
## Coefficients:
## (Intercept)         CRBI         Hits      PutOuts    DivisionW
##    162.5354       0.7743       6.9180       0.2974    -112.3801
##       AtBat        Walks       CWalks        CRuns       CAtBat
##     -2.1687       5.7732      -0.8308       1.4082      -0.1301
##     Assists
##      0.2832
```

```
m_f <- stepAIC(object = sml, scope = list(upper = lrg, lower = sml), direction = "forward", trace = 0)
```

In the forward model selection using AIC, each important variable will be added until there are no more important variables remaining that would imrove the model, however once a variable has been added, it cannot be removed. The best model using this approach would be CRBI, Hits, PutOUts, DIvision,AtBat,Walks,CWalks,CRuns,CAtBat,and Assists.

(d) Using the backward-stepwise-selection approach, find the model that best fits the observed data. This procedure may be automated using the stepAIC() function in R, but you must explain in your own words how this algorithm identifies the 'best' model. Note that you do not need to perform this task in Python.

```
# Backward
stepAIC(object = lrg, scope = list(upper = lrg, lower = sml), direction = "backward", trace = 0)
```

```
##
## Call:
## lm(formula = Salary ~ AtBat + Hits + Walks + CAtBat + CRuns +
##     CRBI + CWalks + Division + PutOuts + Assists, data = hitter)
##
## Coefficients:
## (Intercept)        AtBat         Hits        Walks       CAtBat
##    162.5354      -2.1687       6.9180       5.7732      -0.1301
##       CRuns         CRBI       CWalks    DivisionW      PutOuts
##      1.4082       0.7743      -0.8308    -112.3801       0.2974
##     Assists
##      0.2832
```

```r
m_b <- stepAIC(object = lrg, scope = list(upper = lrg, lower = sml), direction = "backward", trace = 0)
```

In a sense, backward selection is similar to forward selection except, you start with the full model and remove variables that are unimportant one at a time. However, once these variables are removed, they cannot be added back. The best model using this approach is AtBats, Hits,Walks, CAtBat, CRuns, CRBI, CWalks, Division, PutOuts, and Assists.

(e) Using the hybrid-stepwise-selection approach, find the model that best fits the observed data. This procedure may be automated using the stepAIC() function in R, but you must explain in your own words how this algorithm identifies the 'best' model. Note that you do not need to perform this task in Python.

```r
# Hybrid
stepAIC(object = sml, scope = list(upper = lrg, lower = sml), direction = "both", trace = 0)
```

```
##
## Call:
## lm(formula = Salary ~ CRBI + Hits + PutOuts + Division + AtBat +
##     Walks + CWalks + CRuns + CAtBat + Assists, data = hitter)
##
## Coefficients:
## (Intercept)         CRBI         Hits      PutOuts    DivisionW
##    162.5354       0.7743       6.9180       0.2974    -112.3801
##       AtBat        Walks       CWalks        CRuns       CAtBat
##     -2.1687       5.7732      -0.8308       1.4082      -0.1301
##     Assists
##      0.2832
```

```r
m_h <- stepAIC(object = sml, scope = list(upper = lrg, lower = sml), direction = "both", trace = 0)
```

The Hybrid-stepwise selection approach does a combination of both forward and backward model selection. It starts by fitting the intercept only model and then considers adding the most influential variable. If a variable is added, it is then considered to add another variable or remove the least influential variable, repeating for each stage to improve the model. This is continued until there are no variables that can be added or removed to improve the model. However, variales are never stuck in or out of the model. Using this approach the best model uses the variables CRBI, Hits, PutOuts, Division, AtBat, Walks, CWalks, CRuns, CAtBat, and Assists.

(f) In this part you will compare the predictive performance of four models:

i. The full model with all 19 explanatory variables.

ii. The optimal model identified in part (b).

iii. The best model from parts (c)-(e) (i.e., the best stepwise-selection model).

iv. The model that is considered optimal with respect to the Bayesian Information Criterion (BIC) which contains the variables AtBat, Hits, Walks, CRBI, Division and PutOuts.

##Randomly split the observed data into a training set (containing roughly 80% of all of the data) and a held-out test set (containing roughly 20% of all of the data).Calculate the predictive root-mean-square error (RMSE) for each of the four models. Which model appears to be most appropriate? Justify why this model is most appropriate.

```r
rmse <- function(data,model){
  n <- dim(data)[1]
  trn <- sample(x = c(rep(TRUE, round(0.8*n)), rep(FALSE, n-round(0.8*n))), size = n, replace = FALSE)
  train <- data[trn,]
  tst <- !trn
  test <- data[tst,]

  pred <- predict(object = model , newdata = test)
  result<- sqrt(mean((test$Salary - pred)^2))
  return(result)
}

l <- c()

#i The full model with all 19 explanatory variables.
full <- lm(Salary ~ ., data = hitter)
rmse_full <- rmse(hitter,full)



## ii. The optimal model identified in part (b).
rmse_all <- rmse(hitter,m_all)
l[1] <- rmse_all

## iii. The best model from parts (c)-(e) (i.e., the best stepwise-selection model).
rmse_f <- rmse(hitter,m_f)
rmse_b <- rmse(hitter,m_b)
rmse_h <- rmse(hitter,m_h)

## iv. The model that is considered optimal with respect to the Bayesian
#Information Criterion (BIC) which contains the variables AtBat, Hits, Walks, CRBI, Division and PutOut
m_bic <- lm(Salary ~ AtBat + Hits + Walks + CRBI + Division + PutOuts, data = hitter)
BIC(m_bic)
```

```
## [1] 3817.785
```

```r
rmse_bic <- rmse(hitter,m_bic)

compare.rmse <- data.frame("Full" = rmse_full,"All"=rmse_all,
                           "Forward"=rmse_f,"Backward"=rmse_b,"Hybrid"=rmse_h,"Bic"=rmse_bic)
rownames(compare.rmse) <- "Cross-Fold"
```

(g) As in part (f), you must compare the predictive performance of the same four models, but here you must determine the predictive accuracy (predictive RMSE) by using 10-Fold Cross Validation. Which model appears to be most appropriate? Justify why this model is most appropriate.

```r
library(boot)
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
```

```
##     logit
```

```r
mg_full <- glm(Salary ~ ., data = hitter)

mg_ap <- glm(Salary ~ AtBat + Hits + Walks + CRuns+ CRBI + CWalks +
               League + Division + PutOuts + Assists, data = hitter)# 6 multicolinear
mg_f <- glm(Salary ~ CRBI + Hits + PutOuts + Division + AtBat + Walks
            + CWalks + CRuns + CAtBat + Assists, data = hitter)#6
mg_b<- glm(Salary ~ AtBat + Hits + Walks + CAtBat + CRuns + CRBI +
             CWalks + Division + PutOuts + Assists, data = hitter)#6
mg_h <- glm(Salary ~ CRBI + Hits + PutOuts + Division + AtBat + Walks
            + CWalks + CRuns + CAtBat + Assists, data = hitter)#6
mg_bic<- glm(Salary ~ AtBat + Hits + Walks + CRBI + Division + PutOuts, data = hitter)#3

rmse.k <- function(model){
  result <- sqrt((cv.glm(hitter, model, K = 10)$delta)[1])
  return(result)
}

## i. The full model with all 19 explanatory variables.
rmse.k_full <- rmse.k(mg_full)

## ii. The optimal model identified in part (b).
rmse.k_ap <- rmse.k(mg_ap)

## iii. The best model from parts (c)-(e) (i.e., the best stepwise-selection model).
rmse.k_f <- rmse.k(mg_f)
rmse.k_b <- rmse.k(mg_b)
rmse.k_h <- rmse.k(mg_h)

## iv. The model that is considered optimal with respect to the
#Bayesian Information Criterion (BIC) which contains the variables
#AtBat, Hits, Walks, CRBI, Division and PutOuts.
rmse.k_bic <- rmse.k(mg_bic)

k.rmse <- c(rmse.k_full,rmse.k_ap,rmse.k_f,rmse.k_b,rmse.k_h,rmse.k_bic)
compare.rmse <- rbind(compare.rmse,k.rmse)
rownames(compare.rmse) <- c("Cross-Fold",'K-Fold')
compare.rmse
```

```
##                Full      All  Forward Backward   Hybrid      Bic
## Cross-Fold 226.7866 285.3646 319.7999 308.5801 262.8926 264.6210
## K-Fold     340.3170 323.9113 326.4622 326.5075 329.9707 328.5902
```

##(h) Given the estimates of predictive accuracy from parts (f) and (g) indicate which estimates you believe to be more accurate. In other words, indicate which validation approach (i.e., cross validation vs. k-fold cross validation) you believe will most accurately estimate the predictive capability of a model. Briefly explain your rationale.

Given the parts from (f) and (g), it is obvious to see that the K-Fold cross validation is more accurate. This is because RMSE is used to measure how close a predicted value is to the response. With Cross-Fold validation, it is highly variable due to the specific observations when selecting our test set, overall changing the test error. The K-Fold cross validation which combats this with testing k estimates of the test set, stabalizing our test error and then we can get a more precise value with our model to overall better predict.

(i) Accounting for all of the analyses you've performed (i.e., multicollinearity, goodness-of-fit, and predictive accuracy), which model would you be most comfortable using? Briefly justify your choice. [Note: I'm not looking for a right or wrong answer here; I want to see that you can sensibly and eloquently justify your choice].

Accounting for all of the analyses I have performed, the model I would be most comfortable using would be the model using the BIC because it has the least amount of variables accounting for just as much as the other models and it has less multicollinear variables. It also does not contain any of the three worst multicollinear variables which could seriously effect the model,however, we should still be careful when including any variables that have multicollinearity.