Best Subset Regression - R Code

Best subset regression

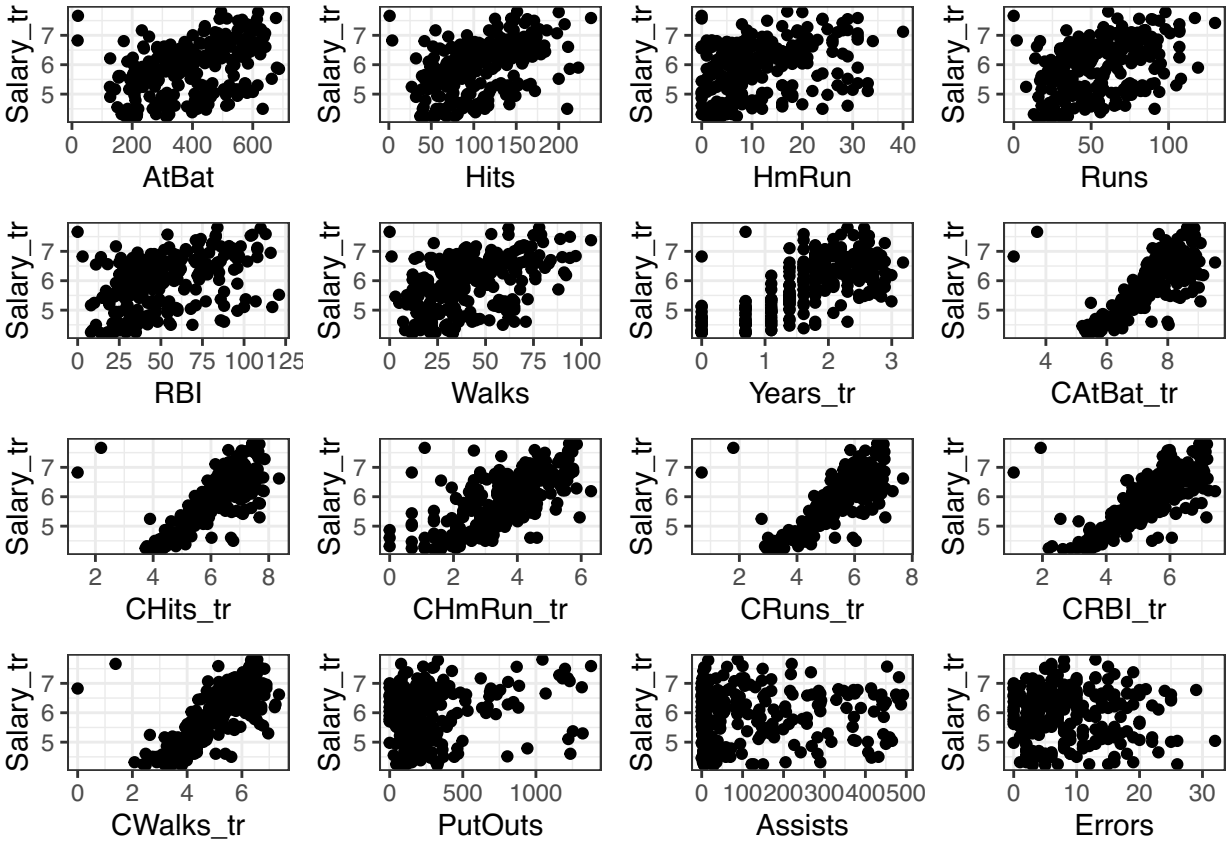
Example Dataset `Hitters` in ISLR2 package. Major League Baseball Data from the 1986 and 1987 seasons.

References James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013) An Introduction to Statistical Learning with applications in R, https://www.statlearning.com, Springer-Verlag, New York

Load and look at the data

```
## [1] 263  17
```

```
##                   AtBat Hits HmRun Runs RBI Walks PutOuts Assists Errors
## -Alan Ashby         315   81     7   24  38    39     632      43     10
## -Alvin Davis        479  130    18   66  72    76     880      82     14
## -Andre Dawson       496  141    20   65  78    37     200      11      3
## -Andres Galarraga   321   87    10   39  42    30     805      40      4
## -Alfredo Griffin    594  169     4   74  51    35     282     421     25
## -Al Newman          185   37     1   23   8    21      76     127      7
##                    Years_tr CAtBat_tr CHits_tr CHmRun_tr CRuns_tr  CRBI_tr
## -Alan Ashby       2.6390573  8.145840 6.727432 4.2484952 5.771441 6.025866
## -Alvin Davis      1.0986123  7.392648 6.124683 4.1588831 5.411646 5.583496
## -Andre Dawson     2.3978953  8.635509 7.362011 5.4205350 6.719013 6.731018
## -Andres Galarraga 0.6931472  5.981414 4.615121 2.5649494 3.871201 3.828641
## -Alfredo Griffin  2.3978953  8.391176 7.032624 2.9957323 6.216606 5.817111
## -Al Newman        0.6931472  5.365976 3.737670 0.6931472 3.401197 2.197225
##                   CWalks_tr Salary_tr
## -Alan Ashby        5.926926  6.163315
## -Alvin Davis       5.572154  6.173786
## -Andre Dawson      5.869297  6.214608
## -Andres Galarraga  3.496508  4.516339
## -Alfredo Griffin   5.267858  6.620073
## -Al Newman         3.178054  4.248495
```



Forward stepwise selection

Here we use the `regsubsets` function but specify the `method="forward"` option.

```
# Identify the models to compare with forward stepwise selection method
regfit.fwd <- regsubsets(Salary_tr ~., data = Hitters_tr,nvmax = 16, method = "forward")
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(Salary_tr ~ ., data = Hitters_tr, nvmax = 16,
##     method = "forward")
## 16 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
## PutOuts       FALSE      FALSE
## Assists       FALSE      FALSE
## Errors        FALSE      FALSE
## Years_tr      FALSE      FALSE
## CAtBat_tr     FALSE      FALSE
## CHits_tr      FALSE      FALSE
## CHmRun_tr     FALSE      FALSE
## CRuns_tr      FALSE      FALSE
```

```
## CRBI_tr       FALSE      FALSE
## CWalks_tr     FALSE      FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: forward
##           AtBat Hits HmRun Runs RBI Walks PutOuts Assists Errors Years_tr
## 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 ) "*"   "*"  "*"   "*"  "*" "*" "*"     "*"     "*"    "*"    "*"
##           CAtBat_tr CHits_tr CHmRun_tr CRuns_tr CRBI_tr CWalks_tr
## 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 ) "*"       "*"      "*"       "*"      "*"     "*"
```

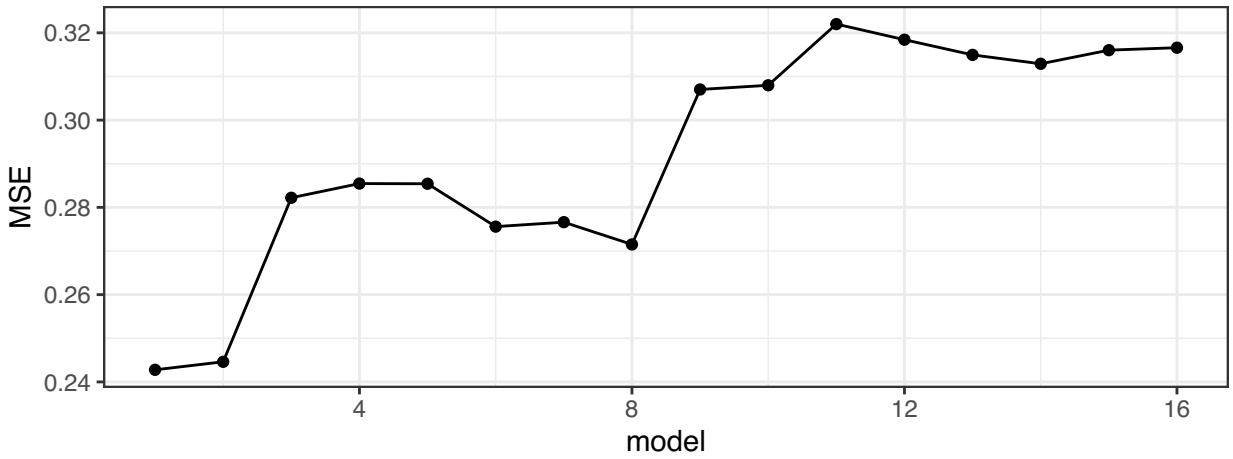
Model Selection Using a Validation Set

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

```
# Split the data train/test
set.seed(598384)
train_inds <- caret::createDataPartition(
  y = Hitters_tr$Salary_tr, # response variable as a vector
  p = 2/3                   # approx. proportion of data used for training
)
# Create the training and test data sets
Hitters_train <- Hitters_tr %>% slice(train_inds[[1]])
Hitters_test  <- Hitters_tr %>% slice(-train_inds[[1]])
```

Now we will train the models found with the forward selection with the train data and make predictions on the observations not used for training. We know there are 16 models, so we set up some vectors to record the errors.

```
val.errors <- rep(NA, 16)

for (i in 1:16){
  # Fit the model
  coefs <- coef(regfit.fwd, i)
  nams <- names(coefs)
  nams <- nams[!nams %in% "(Intercept)"] # exclude the intercept

  # want to do model fit -> get the names of the variables that will be fitted
  form <- as.formula(paste("Salary_tr", paste(nams, collapse = " + "), sep = " ~ "))
  red_model <- lm(form, data = Hitters_train)

  # Get estimated test MSE
  pred = predict(red_model, Hitters_test) # prediction using the test data
  val.errors[i] = mean((Hitters_test$Salary_tr - pred)^2)
}
res_MSE <- data.frame(model = seq(1:16), MSE = val.errors)
ggplot(res_MSE, aes(x= model, y  = MSE)) +
  geom_point() + geom_line() + theme_bw()
```



The first model is the model we are going to use, since the MSE of the test data is the lowest for the first model.

As we expected, the testing error does not go down monotonically as the model gets bigger.

```r
# The best model based on MSE test is the one with 1 variables
coef(regfit.fwd, 1)
```

Get coefficients of "Best models"

```
## (Intercept)     CRBI_tr
##   2.9197778   0.5704067
```

Model Selection Using Cross Validation

Repeat the exercise above replacing the Validation set approach with a CV 5-fold.

Step 1: Split into training and test sets, obtain validation folds

```r
# Set seed for reproducibility
set.seed(7304)

# Generate partition of the 5 folds
# The result is a list of length 5 with indices of observations to include in each fold.
num_crossval_folds <- 5
cross_fold_inds <- caret::createFolds(
  y = Hitters_tr$Salary_tr,    # response variable as a vector
  k = num_crossval_folds # number of folds for CV
)
```

Step 2: Get performance for each fold, using the other folds put together as a training set.

```r
# Object to store the results
results_mse <- expand.grid(
  coef_num = seq_len(16),
  fold_num    = seq_len(num_crossval_folds),
  test_mse    = NA
)
# For loops:
#    16 models from the forward stepwise selection (outside loop)
#    5 model fits for the 5 folds (inside loop)

for(coef_num in seq_len(16)) { # models
  for(fold_num in seq_len(num_crossval_folds)) { # folds

    # Index where to store results
    results_index <- which(
      results_mse$coef_num == coef_num &
      results_mse$fold_num == fold_num
    )

    # Training and testing sets (depends on the fold)
    Hitters_train <- Hitters_tr %>% slice(-cross_fold_inds[[fold_num]])
    Hitters_test  <- Hitters_tr %>% slice(cross_fold_inds[[fold_num]])

    # Fit the model
    coefs <- coef(regfit.fwd, coef_num)
    nams <- names(coefs)
```

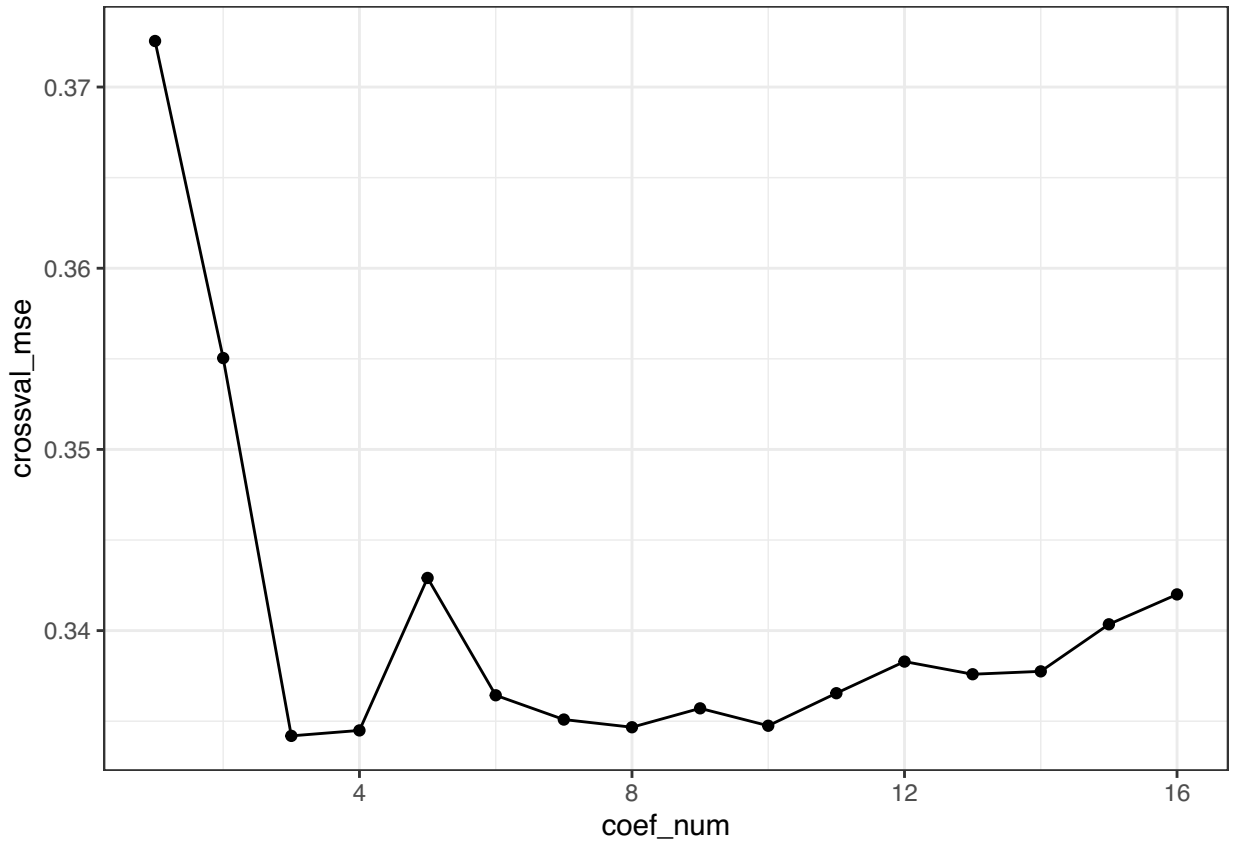
```
    nams <- nams[!nams %in% "(Intercept)"] # exclude the intercept
    form <- as.formula(paste("Salary_tr", paste(nams, collapse = " + "), sep = " ~ "))
    fit <- lm(form, data = Hitters_train)

    # Get estimated test MSE
    pred = predict(fit, Hitters_test)
    results_mse$test_mse[results_index] = mean((Hitters_test$Salary_tr - pred)^2)
  }
}
head(results_mse)
```

```
##   coef_num fold_num  test_mse
## 1        1        1 0.2474671
## 2        2        1 0.2200430
## 3        3        1 0.1858222
## 4        4        1 0.1684644
## 5        5        1 0.1817874
## 6        6        1 0.1708193
```

```
# summarize the results from cross validation
# need to take the average mse for the k folds
summarized_crossval_mse_results <- results_mse %>%
  group_by(coef_num) %>%
  summarize(
    crossval_mse = mean(test_mse)
  )
summarized_crossval_mse_results
```

```
## # A tibble: 16 x 2
##    coef_num crossval_mse
##       <int>        <dbl>
##  1        1        0.373
##  2        2        0.355
##  3        3        0.334
##  4        4        0.334
##  5        5        0.343
##  6        6        0.336
##  7        7        0.335
##  8        8        0.335
##  9        9        0.336
## 10       10        0.335
## 11       11        0.337
## 12       12        0.338
## 13       13        0.338
## 14       14        0.338
## 15       15        0.340
## 16       16        0.342
```

```
# plot the MSE test
ggplot(summarized_crossval_mse_results, aes(x= coef_num, y  = crossval_mse)) +
  geom_point() + geom_line() + theme_bw()
```



These results suggest that the model with 3 coefficient variables have the lowest MSE.

```
# The best model based on MSE test is the one with 3 variables
coef(regfit.fwd, 3)
```

Get coefficients of "Best models"

```
## (Intercept)        Runs    Years_tr     CRBI_tr
## 3.315070219 0.009864452 0.413207643 0.256611127
```