Model Selection Criteria and Automated Search Procedures Tutorial

In this tutorial, we will learn how to use model selection criteria and automated search procedures in setting up our MLR model. We will be using the `regsubsets()` function from the `leaps` package to automate the process of assessing models using model selection criteria, so install and load the `leaps` package:

```
library(leaps)
```

We will use the `mtcars` data set that comes built in with R. The data come from 32 classic automobiles.

```
Data<-mtcars
```

Type `?mtcars` to read the description of the data. Notice that two variables, `vs` and `am` are actually categorical and are coded using 0-1 indicators. Since they are correctly coded as 0-1 indicators, we do not need to use the `factor()` function to convert these variables to be viewed as categorical.

When using `lm()`, R will perform the 0-1 coding associated with categorical variables.

In the examples below, we consider `mpg` to the response variable and all the other variables are potential predictors.

1 Model Selection Criteria

The `regsubsets()` function from the `leaps` package will fit all possible regression models based on the supplied dataframe and specified response variable, and then calculate the values of R^2, adjusted R^2, SS_{res}, Mallows C_p, and BIC of each model. It does not calculate the PRESS statistic and the AIC. To do so:

```
allreg <- leaps::regsubsets(mpg ~., data=Data, nbest=1)
summary(allreg)
```

```
## Subset selection object
## Call: regsubsets.formula(mpg ~ ., data = Data, nbest = 1)
## 10 Variables  (and intercept)
##      Forced in Forced out
## cyl      FALSE      FALSE
## disp     FALSE      FALSE
## hp       FALSE      FALSE
## drat     FALSE      FALSE
## wt       FALSE      FALSE
## qsec     FALSE      FALSE
## vs       FALSE      FALSE
## am       FALSE      FALSE
## gear     FALSE      FALSE
## carb     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          cyl disp hp  drat wt  qsec vs  am  gear carb
## 1  ( 1 ) " " " " " " " " "*" " " " " " " " " " "
## 2  ( 1 ) "*" " " " " " " "*" " " " " " " " " " "
## 3  ( 1 ) " " " " " " " " "*" "*" " " "*" " " " "
## 4  ( 1 ) " " " " "*" " " "*" "*" " " "*" " " " "
```

```
## 5  ( 1 ) " " "*"  "*" " "  "*" "*"  " " "*" " "  " " 
## 6  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" " "  " " 
## 7  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" "*"  " " 
## 8  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" "*"  "*" 
```

The default value for `nbest` is 1. This means that the algorithm will return the one best set of predictors (based on R^2) for each number of possible predictors.

So based on R^2, among all possible 1-predictor models, the model that is best has `wt` as the one predictor. Among all possible 2-predictor models, the model that is best has `cyl` and `wt` as the two predictors.

Changing `nbest` to be 2 gives:

```r
allreg2 <- leaps::regsubsets(mpg ~., data=Data, nbest=2)
summary(allreg2)
```

```
## Subset selection object
## Call: regsubsets.formula(mpg ~ ., data = Data, nbest = 2)
## 10 Variables  (and intercept)
##      Forced in Forced out
## cyl      FALSE      FALSE
## disp     FALSE      FALSE
## hp       FALSE      FALSE
## drat     FALSE      FALSE
## wt       FALSE      FALSE
## qsec     FALSE      FALSE
## vs       FALSE      FALSE
## am       FALSE      FALSE
## gear     FALSE      FALSE
## carb     FALSE      FALSE
## 2 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          cyl disp hp  drat wt  qsec vs  am  gear carb
## 1  ( 1 ) " " " "  " " " "  "*" " "  " " " " " "  " " 
## 1  ( 2 ) "*" " "  " " " "  " " " "  " " " " " "  " " 
## 2  ( 1 ) "*" " "  " " " "  "*" " "  " " " " " "  " " 
## 2  ( 2 ) " " " "  "*" " "  "*" " "  " " " " " "  " " 
## 3  ( 1 ) " " " "  " " " "  "*" "*"  " " "*" " "  " " 
## 3  ( 2 ) "*" " "  "*" " "  "*" " "  " " " " " "  " " 
## 4  ( 1 ) " " " "  "*" " "  "*" "*"  " " "*" " "  " " 
## 4  ( 2 ) " " " "  " " " "  "*" "*"  " " "*" " "  "*" 
## 5  ( 1 ) " " "*"  "*" " "  "*" "*"  " " "*" " "  " " 
## 5  ( 2 ) " " " "  " " "*"  "*" "*"  " " "*" " "  "*" 
## 6  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" " "  " " 
## 6  ( 2 ) " " "*"  "*" " "  "*" "*"  " " "*" "*"  " " 
## 7  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" "*"  " " 
## 7  ( 2 ) "*" "*"  "*" "*"  "*" "*"  " " "*" " "  " " 
## 8  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" "*"  "*" 
## 8  ( 2 ) " " "*"  "*" "*"  "*" "*"  "*" "*" "*"  " " 
```

So based on R^2, among all possible 1-predictor models, the model that is best has `wt` as the one predictor. The second best 1-predictor model has `cyl` as the one predictor.

Let's see what can be extracted from `summary(allreg2)`:

```r
names(summary(allreg2))
```

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

We can extract information regarding adjusted R^2, Mallow's C_p, and BIC, so we can find the best models based on these criteria:

```
which.max(summary(allreg2)$adjr2)
```

```
## [1] 9
```

```
which.min(summary(allreg2)$cp)
```

```
## [1] 5
```

```
which.min(summary(allreg2)$bic)
```

```
## [1] 5
```

From `allreg2`, model 9 has the best adjusted R^2, while model 5 has the best Mallow's C_p and BIC. To get the corresponding coefficients and predictors of these models:

```
coef(allreg2, which.max(summary(allreg2)$adjr2))
```

```
## (Intercept)        disp          hp          wt        qsec          am
## 14.36190396  0.01123765 -0.02117055 -4.08433206  1.00689683  3.47045340
```

```
coef(allreg2, which.min(summary(allreg2)$cp))
```

```
## (Intercept)          wt        qsec          am
##    9.617781   -3.916504    1.225886    2.935837
```

```
coef(allreg2, which.min(summary(allreg2)$bic))
```

```
## (Intercept)          wt        qsec          am
##    9.617781   -3.916504    1.225886    2.935837
```

It turns out we have 2 candidate models. They all have `wt`, `qsec`, and `am`. The model with the best adjusted R^2 has two additional predictors: `disp` and `hp`.

2 Forward selection, backward elimination, and stepwise regression

Fitting all possible regression models can be slow to run if there are many predictors and many observations. To cut down the number of potential models considered, we can use forward selection, backward elimination, and stepwise regression. These procedures will require declaring the smallest possible model, and the largest possible model first. The algorithm will consider models within this range of models.

To do so, we start by declaring an intercept-only model and a full model (with all predictors). These two models contain the scope of all possible models to consider:

```
##intercept only model
regnull <- lm(mpg~1, data=Data)
##model with all predictors
regfull <- lm(mpg~., data=Data)
```

To carry out forward selection:

```
step(regnull, scope=list(lower=regnull, upper=regfull), direction="forward")
```

```
## Start:  AIC=115.94
## mpg ~ 1
##
##        Df Sum of Sq     RSS     AIC
## + wt    1    847.73  278.32  73.217
## + cyl   1    817.71  308.33  76.494
```

```
## + disp  1    808.89  317.16  77.397
## + hp    1    678.37  447.67  88.427
## + drat  1    522.48  603.57  97.988
## + vs    1    496.53  629.52  99.335
## + am    1    405.15  720.90 103.672
## + carb  1    341.78  784.27 106.369
## + gear  1    259.75  866.30 109.552
## + qsec  1    197.39  928.66 111.776
## <none>              1126.05 115.943
##
## Step:  AIC=73.22
## mpg ~ wt
##
##        Df Sum of Sq    RSS    AIC
## + cyl   1    87.150 191.17 63.198
## + hp    1    83.274 195.05 63.840
## + qsec  1    82.858 195.46 63.908
## + vs    1    54.228 224.09 68.283
## + carb  1    44.602 233.72 69.628
## + disp  1    31.639 246.68 71.356
## <none>              278.32 73.217
## + drat  1     9.081 269.24 74.156
## + gear  1     1.137 277.19 75.086
## + am    1     0.002 278.32 75.217
##
## Step:  AIC=63.2
## mpg ~ wt + cyl
##
##        Df Sum of Sq    RSS    AIC
## + hp    1   14.5514 176.62 62.665
## + carb  1   13.7724 177.40 62.805
## <none>              191.17 63.198
## + qsec  1   10.5674 180.60 63.378
## + gear  1    3.0281 188.14 64.687
## + disp  1    2.6796 188.49 64.746
## + vs    1    0.7059 190.47 65.080
## + am    1    0.1249 191.05 65.177
## + drat  1    0.0010 191.17 65.198
##
## Step:  AIC=62.66
## mpg ~ wt + cyl + hp
##
##        Df Sum of Sq    RSS    AIC
## <none>              176.62 62.665
## + am    1    6.6228 170.00 63.442
## + disp  1    6.1762 170.44 63.526
## + carb  1    2.5187 174.10 64.205
## + drat  1    2.2453 174.38 64.255
## + qsec  1    1.4010 175.22 64.410
## + gear  1    0.8558 175.76 64.509
## + vs    1    0.0599 176.56 64.654

##
## Call:
```

```
## lm(formula = mpg ~ wt + cyl + hp, data = Data)
##
## Coefficients:
## (Intercept)           wt          cyl           hp
##    38.75179     -3.16697     -0.94162     -0.01804
```

At the start of the algorithm, the AIC of the intercept-only model is calculated to be 115.94. The algorithm then considers adding each predictor to the intercept-only model. For each 1-predictor model, the AIC is calculated, and the 1-predictor models are arranged from smallest to largest in terms of AIC. In the output, all 1-predictor models are superior to the intercept-only model. The predictor `wt` is chosen to be used since it results in the model with the smallest AIC.

In the next step, `wt` is in the model and cannot be removed. The AIC is 73.22. The algorithm then considers adding each predictor in addition to `wt`. For each two-predictor model, the AIC is calculated. The two-predictor models are then ordered from smallest to largest. Adding `cyl` leads to the smallest AIC so it is chosen to be added to `wt`. Note that adding `drat`, `gear`, or `am` to `wt` actually increases the AIC.

The algorithm continues until the last step. At this stage, `wt`, `cyl`, and `hp` are added to the model and have an AIC of 62.66. The algorithm considers adding one of the remaining predictors, but adding any of them results in a higher AIC. Thus the algorithm stops.

For backward elimination, the code is similar. The `direction` is changed from `forward` to `backward`:

```
step(regfull, scope=list(lower=regnull, upper=regfull), direction="backward")
```

And for stepwise regression, `direction` is set to `both`:

```
step(regnull, scope=list(lower=regnull, upper=regfull), direction="both")
```

```
## Start:  AIC=115.94
## mpg ~ 1
##
##        Df Sum of Sq     RSS     AIC
## + wt    1    847.73  278.32  73.217
## + cyl   1    817.71  308.33  76.494
## + disp  1    808.89  317.16  77.397
## + hp    1    678.37  447.67  88.427
## + drat  1    522.48  603.57  97.988
## + vs    1    496.53  629.52  99.335
## + am    1    405.15  720.90 103.672
## + carb  1    341.78  784.27 106.369
## + gear  1    259.75  866.30 109.552
## + qsec  1    197.39  928.66 111.776
## <none>              1126.05 115.943
##
## Step:  AIC=73.22
## mpg ~ wt
##
##        Df Sum of Sq     RSS     AIC
## + cyl   1     87.15  191.17  63.198
## + hp    1     83.27  195.05  63.840
## + qsec  1     82.86  195.46  63.908
## + vs    1     54.23  224.09  68.283
## + carb  1     44.60  233.72  69.628
## + disp  1     31.64  246.68  71.356
## <none>                278.32  73.217
## + drat  1      9.08  269.24  74.156
```

```
## + gear  1      1.14  277.19  75.086
## + am    1      0.00  278.32  75.217
## - wt    1    847.73 1126.05 115.943
##
## Step:  AIC=63.2
## mpg ~ wt + cyl
##
##        Df Sum of Sq    RSS    AIC
## + hp    1    14.551 176.62 62.665
## + carb  1    13.772 177.40 62.805
## <none>              191.17 63.198
## + qsec  1    10.567 180.60 63.378
## + gear  1     3.028 188.14 64.687
## + disp  1     2.680 188.49 64.746
## + vs    1     0.706 190.47 65.080
## + am    1     0.125 191.05 65.177
## + drat  1     0.001 191.17 65.198
## - cyl   1    87.150 278.32 73.217
## - wt    1   117.162 308.33 76.494
##
## Step:  AIC=62.66
## mpg ~ wt + cyl + hp
##
##        Df Sum of Sq    RSS    AIC
## <none>              176.62 62.665
## - hp    1    14.551 191.17 63.198
## + am    1     6.623 170.00 63.442
## + disp  1     6.176 170.44 63.526
## - cyl   1    18.427 195.05 63.840
## + carb  1     2.519 174.10 64.205
## + drat  1     2.245 174.38 64.255
## + qsec  1     1.401 175.22 64.410
## + gear  1     0.856 175.76 64.509
## + vs    1     0.060 176.56 64.654
## - wt    1   115.354 291.98 76.750

##
## Call:
## lm(formula = mpg ~ wt + cyl + hp, data = Data)
##
## Coefficients:
## (Intercept)           wt          cyl           hp
##    38.75179     -3.16697     -0.94162     -0.01804
```

Notice with stepwise regression, we consider removing any predictor as well as adding any predictor in each step.

It turns out that for this dataset, forward selection, backward elimination, and stepwise regression propose the same model with `wt`, `cyl`, and `hp` as predictors.

3 Some Comments

1. `regsubsets()` and `step()` functions only consider 1st order models (no interactions or higher order terms).

2. `regsubsets()` and `step()` functions do not check if the regression assumptions are met. You still need to check the residual plot.
3. `regsubsets()` and `step()` functions do not guarantee the best model will be identified for your specific goal.
4. Notice that the various model selection criteria used in `regsubsets()` can lead to different models.
5. The various procedures in the `step()` function can lead to different models.
6. `step()` function can lead to different models if you have a different starting point.
7. For the `step()` function, R uses AIC to decide when to stop the search. The textbook describes the procedure using the F statistic. The choice of criteria could impact the end result.

4 Practice Question

For the candidate models found above, create residual plots and Box Cox plots to see if the response variable should be transformed. If needed, transform the response variable, and re run these the `regsubsets()` and `step()` functions to see what candidate models are suggested.