R Codes for Chapter-9
Model Selection in R

In our discussion of regression so far, we have assumed that all the explanatory variables included in the model are chosen in advance. However, in many situations the set of explanatory variables to be included is not predetermined and selecting them becomes part of the analysis.

There are two main approaches towards variable selection in R: **The best subset algorithms** and **Stepwise regression methods.**

The Best Subset Algorithms

The best subset algorithms consider all possible subsets of the pool of explanatory variables and finds the model that best fits the data according to some criteria (e.g. Adjusted R2 , AIC and BIC). These criteria assign scores to each model and allow us to choose the model with the best score.

The function **regsubsets()** in the library **"leaps"** can be used for regression subset selection. Thereafter, one can view the ranked models according to different scoring criteria by plotting the results of **regsubsets().**

Before using the function for the first time you will need to install the library using the R GUI. Alternatively, you can use the command **install.packages("leaps")** to install it.

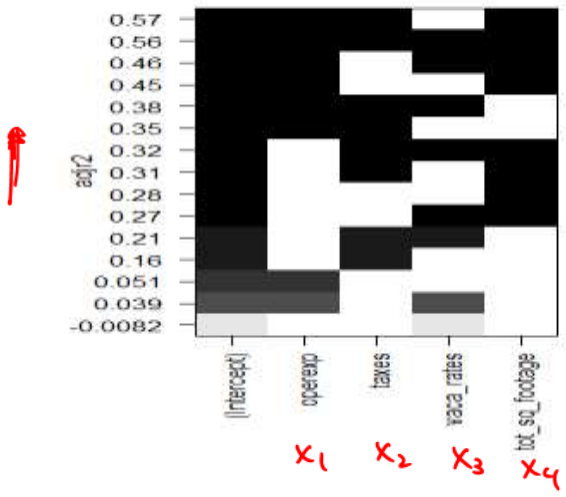
Ex. (Ex. 6.18: Commercial Properties Example) : Y- rental rates, X1-operating expenses, X2-taxes, X3- vacancy rates, x4- total square footage.

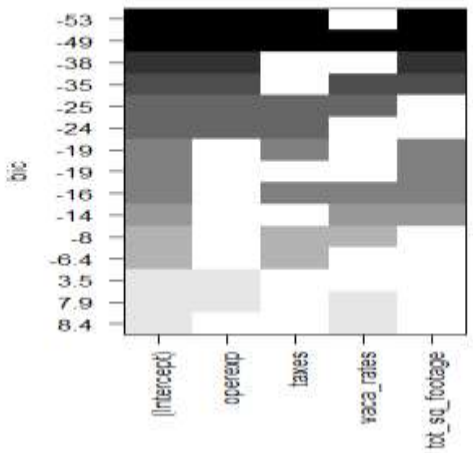
Use rental rates as the response variable and determine which of the four explanatory variables should be included in the regression model using the all possible regressions approach.

```
> data=read.table("R:\\Teaching\\2016\\MA 542\\Class preparation\\Ex.6.18.csv
",header = FALSE)
> library(leaps)
> leaps=regsubsets(rentalrates~operexp+taxes+vaca_rates+tot_sq_footage,data=d
ata,  nbest=15)
```

To view the ranked models according to the adjusted R-squared criteria and BIC, respectively, type:

```
> plot(leaps, scale="adjr2")
> plot(leaps, scale="bic")
```





Here black indicates that a variable is included in the model, while white indicates that they are not. The model containing operating expenses, taxes, total square footage minimizes both the adjusted Rsquare criteria (left) and the BIC (right). Looking at the values on the y-axis of the plot indicates that the top two models have roughly the same adjusted R-square and BIC values, thus possibly explaining the discrepancy in the results.

Stepwise Regression Methods.

Stepwise regression methods are useful when the number of explanatory variables is large and it is not feasible to fit all possible models. In this case, it is more efficient to use a search algorithm (e.g., Forward selection, Backward elimination and Stepwise regression) to find the best model.

The R function **step()** can be used to perform variable selection. To perform forward selection we need to begin by specifying a starting model and the range of models which we want to examine in the search.

To fit the null model (The model with only the intercept):

```
> null=lm(rentalrates~1, data=data)
> null

Call:

lm(formula = rentalrates ~ 1, data = data)

Coefficients:
(Intercept)
      15.14
```

To fit the Full model (the model with all the possible predictors):

```
> full=lm(rentalrates~operexp+taxes+vaca_rates+tot_sq_footage,data=data)
> full

Call:

lm(formula = rentalrates ~ operexp + taxes + vaca_rates + tot_sq_footage,
    data = data)

Coefficients:
   (Intercept)         operexp           taxes      vaca_rates  tot_sq_footage
     1.220e+01      -1.420e-01       2.820e-01       6.193e-01       7.924e-06
```

We can perform forward selection using the following command: This tells R to start with the null model and search through models lying in the range between the null and full model using the forward selection algorithm. It gives rise to the following output:

```
> step(null, scope=list(lower=null, upper=full), direction="forward")


Start:  AIC=88.81
rentalrates ~ 1

                 Df Sum of Sq    RSS    AIC
+ tot_sq_footage  1    67.775 168.78 63.467
+ taxes           1    40.503 196.05 75.599
+ operexp         1    14.819 221.74 85.571
<none>                        236.56 88.811
+ vaca_rates      1     1.047 235.51 90.452
```

```
Step:  AIC=63.47
rentalrates ~ tot_sq_footage

             Df Sum of Sq    RSS    AIC
+ operexp     1    42.275 126.51 42.114
+ taxes       1     9.291 159.49 60.881
<none>                    168.78 63.467
+ vaca_rates  1     0.130 168.65 65.405

Step:  AIC=42.11
rentalrates ~ tot_sq_footage + operexp

             Df Sum of Sq    RSS    AIC
+ taxes       1   27.8575  98.65 23.968
<none>                    126.51 42.114
+ vaca_rates  1    2.5183 123.99 42.486

Step:  AIC=23.97
rentalrates ~ tot_sq_footage + operexp + taxes

             Df Sum of Sq    RSS    AIC
<none>                    98.650 23.968
+ vaca_rates  1   0.41975 98.231 25.622

Call:
lm(formula = rentalrates ~ tot_sq_footage + operexp + taxes,
    data = data)

Coefficients:
  (Intercept)  tot_sq_footage         operexp           taxes
    1.237e+01       8.178e-06      -1.442e-01       2.672e-01
```

According to this procedure, the best model is the one that includes the variables `tot_sq_footage`, `operexp` and `taxes`.

We can perform backward elimination on the same data set using the command:

```
> step(full, data=Housing, direction="backward")

Start:  AIC=25.62
rentalrates ~ operexp + taxes + vaca_rates + tot_sq_footage

                Df Sum of Sq     RSS    AIC
- vaca_rates     1     0.420  98.650 23.968
<none>                       98.231 25.622
- taxes          1    25.759 123.990 42.486
- tot_sq_footage 1    42.325 140.556 52.643
- operexp        1    57.243 155.473 60.814

Step:  AIC=23.97
rentalrates ~ operexp + taxes + tot_sq_footage

                Df Sum of Sq    RSS    AIC
<none>                       98.65 23.968
- taxes          1    27.857 126.51 42.114
- tot_sq_footage 1    50.287 148.94 55.335
- operexp        1    60.841 159.49 60.881

Call:
lm(formula = rentalrates ~ operexp + taxes + tot_sq_footage,
    data = data)
```

```
Coefficients:
   (Intercept)         operexp           taxes  tot_sq_footage
     1.237e+01      -1.442e-01       2.672e-01       8.178e-06
```

and stepwise regression using the command:

```
step(null, scope = list(upper=full), data=Housing, direction="both")
```

```
Start:  AIC=88.81
rentalrates ~ 1

                 Df Sum of Sq    RSS    AIC
+ tot_sq_footage  1    67.775 168.78 63.467
+ taxes           1    40.503 196.05 75.599
+ operexp         1    14.819 221.74 85.571
<none>                        236.56 88.811
+ vaca_rates      1     1.047 235.51 90.452

Step:  AIC=63.47
rentalrates ~ tot_sq_footage

                 Df Sum of Sq    RSS    AIC
+ operexp         1    42.275 126.51 42.114
+ taxes           1     9.291 159.49 60.881
<none>                        168.78 63.467
+ vaca_rates      1     0.130 168.65 65.405
- tot_sq_footage  1    67.775 236.56 88.811

Step:  AIC=42.11
rentalrates ~ tot_sq_footage + operexp

                 Df Sum of Sq    RSS    AIC
+ taxes           1    27.857  98.65 23.968
<none>                        126.51 42.114
+ vaca_rates      1     2.518 123.99 42.486
- operexp         1    42.275 168.78 63.467
- tot_sq_footage  1    95.231 221.74 85.571

Step:  AIC=23.97
rentalrates ~ tot_sq_footage + operexp + taxes

                 Df Sum of Sq     RSS    AIC
<none>                        98.650 23.968
+ vaca_rates      1     0.420  98.231 25.622
- taxes           1    27.857 126.508 42.114
- tot_sq_footage  1    50.287 148.937 55.335
- operexp         1    60.841 159.491 60.881

Call:
lm(formula = rentalrates ~ tot_sq_footage + operexp + taxes,
    data = data)
```

$$\frac{\sum (y_i - y_{i(i)})^2}{n}$$

```
Coefficients:
   (Intercept)  tot_sq_footage         operexp           taxes
     1.237e+01       8.178e-06      -1.442e-01       2.672e-01
```

Both algorithms give rise to results that are equivalent to the forward selection procedure in the Commercial Properties example.