Pau Blasco Roca
1600959

Homework 2

Choose a dataset for the analysis. You can use the *data*() function to find a dataset from library *datasets* in R.

1. Provide details of the chosen dataset. Design models to be analysed for this dataset.

I will be using the dataset "longley", originally from the paper "An appraisal of least-squares programs from the point of view of the user. *Journal of the American Statistical Association* **62**, 819–841", published in 1967 by J. W. Longley. "longley" is a macroeconomic data set which provides a well-known example for a highly collinear regression.

It has 16 different readings from 7 variables, observed between 1947 to 1962. These are the following:

- GNP: Gross National Product
- GNP.deflator: GNP implicit price deflator (*1954=100*)
- Employed: number of people employed.
- Unemployed: number of unemployed.
- Armed.Forces: number of people in the armed forces
- Population: 'noninstitutionalized' population ≥ 14 years of age.
- Year: the current year (time).

I will be having the variable "Employed" test against all the other six variables, and see if they correlate to each other in a meaningful way. My linear model will be defined, then, as follows:

```
library(datasets)
myData = longley
FitAll = lm(Employed ~ . , data=myData)
summary(FitAll)
```

Which prints the following output to the terminal:

```
Call:
lm(formula = Employed ~ ., data = myData)

Residuals:
     Min       1Q   Median       3Q      Max
-0.41011 -0.15767 -0.02816  0.10155  0.45539

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 **
GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
GNP          -3.582e-02  3.349e-02  -1.070 0.312681
Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
Population   -5.110e-02  2.261e-01  -0.226 0.826212
Year          1.829e+00  4.555e-01   4.016 0.003037 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3049 on 9 degrees of freedom
Multiple R-squared:  0.9955,    Adjusted R-squared:  0.9925
F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

We can see the p-value is small, but some of the variables don't have a high significance level. Let's fix that.

2. Apply backward selection to find the best fit model using p-value and AIC criteria. Compare the results found by both methods. Do the same for forward selection. Comment on the results.

We will start with backward selection. We will be first using the manual p-value criterion. The R code is the following:

```
model_full<-lm(Employed~., data=myData)
summary(model_full) # INITIAL p-value: 4.984e-10

#we remove GNP.deflator, as it reduces our p-value the most
model<-update(model_full,.~.-GNP.deflator)
summary(model) #p-value: 2.242e-11

#we remove Population, as it reduces our p-value the most
model2<-update(model,.~.-Population)
summary(model2) #FINAL p-value: 9.5e-13

#From this point every variable we remove increases the p-value.
```

Using the manual AIC method, the results are very similar:

```
model_full<-lm(Employed~., data=myData)
summary(model_full) # INITIAL AIC: -33.21933

#we remove GNP.deflator, as it reduces our AIC the most
model<-update(model_full,.~.-GNP.deflator)
summary(model) # AIC: -35.1635

#we remove Population, as it reduces our AIC the most
model2<-update(model,.~.-Population)
summary(model2) #FINAL AIC: -36.79916

#From this point every variable we remove increases the AIC.
```

Using the automatic *step()* function, we get the same result.

```
step(model_full, trace=TRUE, direction="backward")
```

The result is the following: both **GNP.deflator** and **Population** are removed from the model, as they are non significant following both the p-value and the AIC methods. We are left with this four variables: **GNP**, **Unemployed**, **Armed.Forces**, and **Year**.

We may now continue with the same procedure for the forward selection algorithm.

We will be first using the manual p-value criterion. The R code is the following:

```
model_null<-lm(Employed~1, data=myData)
summary(model_null) # INITIAL p-value: -

#we add GNP, as it reduces our p-value the most
model<-update(model_null,.~.+GNP)
summary(model) # p-value: 8.363479e-12

#we add Unemployed, as it reduces our p-value the most
model2<-update(model,.~.-Unemployed)
summary(model2) #p-value: 7.2904e-12

#From this point every variable we add increases the p-value.
```

Using the manual AIC method, the results appear similar to the ones using backward selection:

```
model_null<-lm(Employed~1, data=myData)
summary(model_null) # INITIAL AIC: 41.17

#we add GNP, as it reduces our AIC the most
model<-update(model_null,.~.+GNP)
summary(model) # AIC: -11.6

#we add Unemployed, as it reduces our AIC the most
model2<-update(model,.~.-Unemployed)
summary(model2) #AIC: -17.66

#we add Armed.Forces, as it reduces our AIC the most
Model3<-update(model2,.~.-Armed.Forces)
summary(model3) #AIC: -20.14

#we add Year, as it reduces our AIC the most
Model4<-update(model3,.~.-Year)
summary(model4) #AIC: -36.8

#From this point every variable we add increases the AIC.
```

Using the automatic *step()* function, we get the same result.

```
step(model_null, trace=TRUE, direction="forward", scope=
formula(model_full))
```

To retrieve a final model, we will use the automatic *step()* function with *direction* set to “both”.

```
step(model_null, trace=TRUE, direction="both", scope=
formula(model_full))
```

This returns the following:

```
lm(formula = Employed ~ GNP + Unemployed + Armed.Forces + Year,
    data = myData)

Coefficients:
 (Intercept)           GNP    Unemployed  Armed.Forces          Year
  -3.599e+03    -4.019e-02    -2.088e-02    -1.015e-02     1.887e+00
```

Pau Blasco Roca
1600959

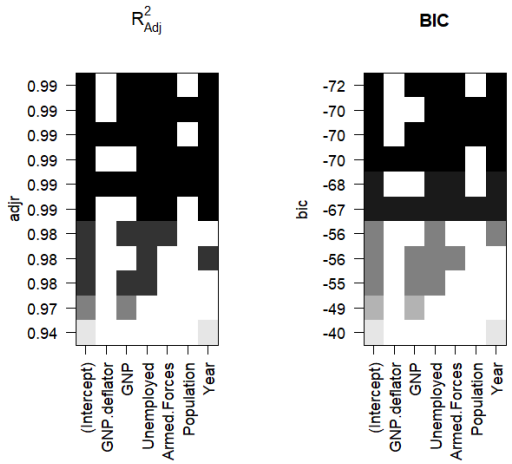
3. Find the best possible subset of variables to select the best fit model. Compare the results with the final models obtained in the previous point.

For this, we can make good use of the library “leaps”.

```
library(leaps)
subs<-regsubsets(Employed ~ . , data=myData, nbest=2)
summary(subs)$which

par(mfrow=c(1,2))
plot(subs,scale="adjr", main=expression(R[Adj]^2))
plot(subs,scale="bic", main="BIC")
```

Which outputs the following image:



Then, we can choose the best model of the subsets:

```
summary(subs)$adjr2
summary(subs)$bic

#retrieve best indexes for each comparison
kval = which(summary(subs)$adjr2==max(summary(subs)$adjr2))
summary(subs)$which[kval,]

kval = which(summary(subs)$bic==min(summary(subs)$bic))
summary(subs)$which[kval,]
```

The indexes “kval” both turn out to be equal to seven, retrieving the same model. This is the summary of the best subset of variables we just found:

```
Intcp | GNP.deflator | GNP | Unemployed | Armed.Forces | Population |Year
 TRUE          FALSE   TRUE         TRUE           TRUE         FALSE  TRUE
```

The results are the same as the ones we got in exercise two. We have also found that both **Population** and **GNP.deflator** are not significant for our model, while the other four variables remaining are necessary.