Tutorial 3

Q1) refer to data uploaded in <u>table2_7</u> that measured from types of insects :

(a) Ch . Concinna “A” and

(b) Ch . Heikertlingeri . “B”

It shows two variables :

• X_1: first the width of the first joint

• X_2: the width for the second joint .

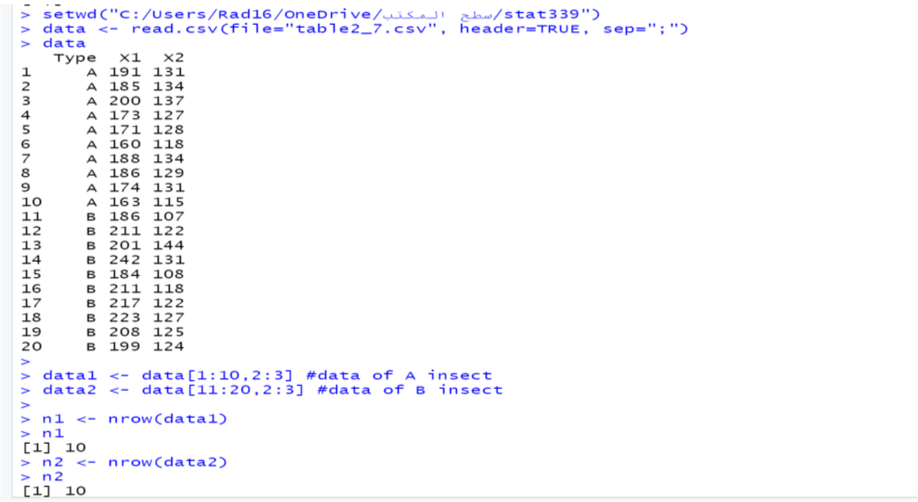
الجدول التالي يبين عملية تشريح حشرات الشاتوكتيميا لعشرين من ذكور الخنافس الصغيرة حيث أن المتغيرات هي)
(عرض المفصل (الكاحل) الأول والثاني

Find

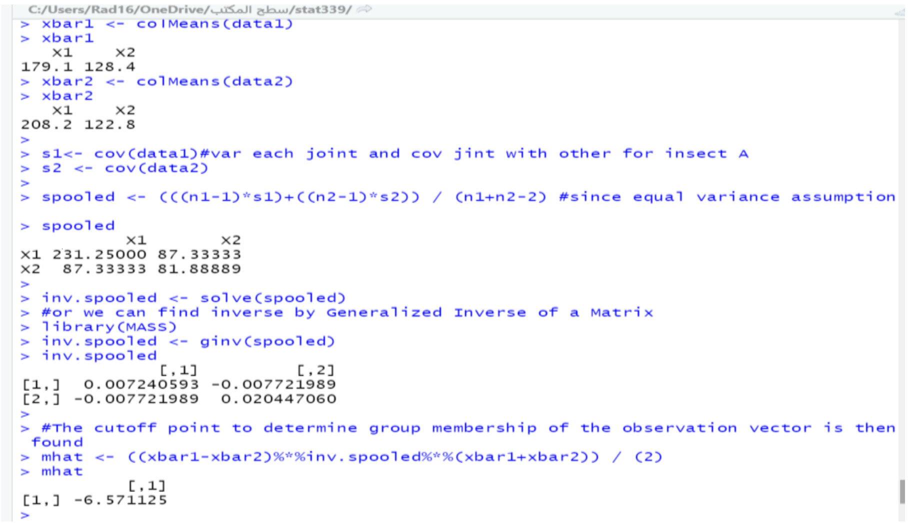
1- Find the estimated Fisher’s linear discriminant function.
2- Classify the new insect with observation (194 , 124)

Solution:

1-



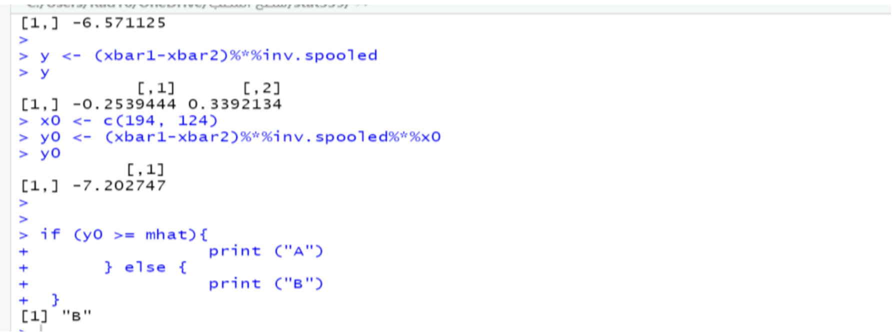
```
> setwd("C:/Users/Rad16/OneDrive/سطح المكتب/stat339")
> data <- read.csv(file="table2_7.csv", header=TRUE, sep=";")
> data
   Type  X1  X2
1     A 191 131
2     A 185 134
3     A 200 137
4     A 173 127
5     A 171 128
6     A 160 118
7     A 188 134
8     A 186 129
9     A 174 131
10    A 163 115
11    B 186 107
12    B 211 122
13    B 201 144
14    B 242 131
15    B 184 108
16    B 211 118
17    B 217 122
18    B 223 127
19    B 208 125
20    B 199 124
> 
> data1 <- data[1:10,2:3] #data of A insect
> data2 <- data[11:20,2:3] #data of B insect
> 
> n1 <- nrow(data1)
> n1
[1] 10
> n2 <- nrow(data2)
> n2
[1] 10
```



C:/Users/Rad16/OneDrive/سطح المكتب/stat339/

```
> xbar1 <- colMeans(data1)
> xbar1
   x1    x2 
179.1 128.4 
> xbar2 <- colMeans(data2)
> xbar2
   x1    x2 
208.2 122.8 
> 
> s1<- cov(data1)#var each joint and cov jint with other for insect A
> s2 <- cov(data2)
> 
> spooled <- (((n1-1)*s1)+((n2-1)*s2)) / (n1+n2-2) #since equal variance assumption

> spooled
          x1       x2
x1 231.25000 87.33333
x2  87.33333 81.88889
> 
> inv.spooled <- solve(spooled)
> #or we can find inverse by Generalized Inverse of a Matrix
> library(MASS)
> inv.spooled <- ginv(spooled)
> inv.spooled
             [,1]         [,2]
[1,]  0.007240593 -0.007721989
[2,] -0.007721989  0.020447060
> 
> #The cutoff point to determine group membership of the observation vector is then
 found
> mhat <- ((xbar1-xbar2)%*%inv.spooled%*%(xbar1+xbar2)) / (2)
> mhat
          [,1]
[1,] -6.571125
> 
```

2-



```
[1,] -6.571125
> 
> y <- (xbar1-xbar2)%*%inv.spooled
> y
           [,1]      [,2]
[1,] -0.2539444 0.3392134
> x0 <- c(194, 124)
> y0 <- (xbar1-xbar2)%*%inv.spooled%*%x0
> y0
          [,1]
[1,] -7.202747
> 
> 
> if (y0 >= mhat){
+                  print ("A")
+        } else {
+                  print ("B")
+   }
[1] "B"
```

we classify it to insect B since Y<m

Q2) upload "iris" data frame with 150 cases (rows) and 5 variables (columns) where

It shows two variables :

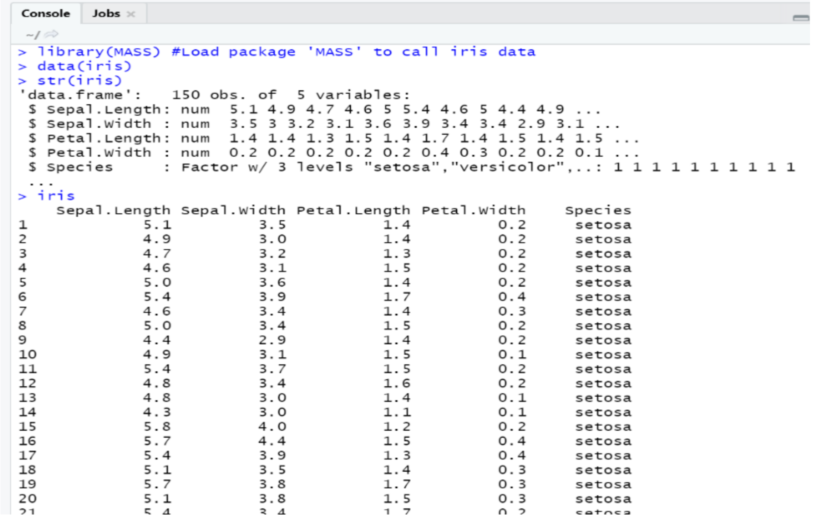
- X_1: iris where its Species :setosa
- X_2: iris where its Species : versicolor

Find

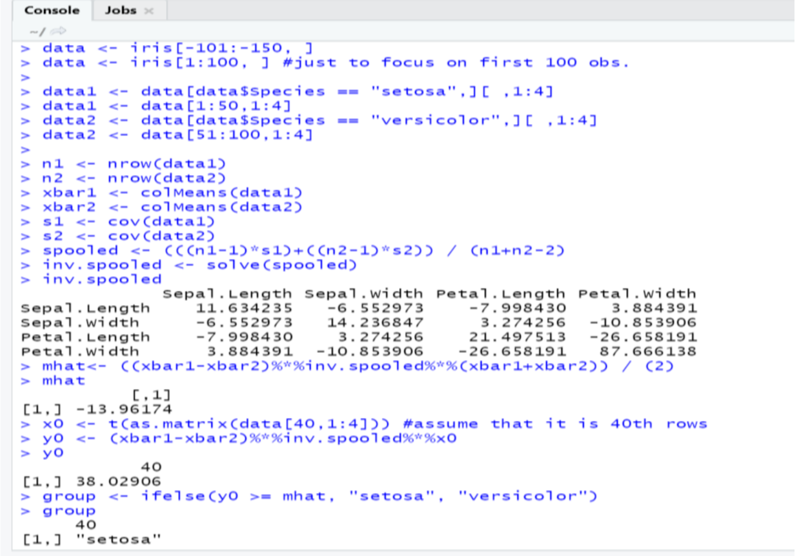
1- the estimated Fisher's linear discriminant function.
2- Classify the new observation of row #40

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
40	5.1	3.4	1.5	0.2

Solution:



```
> library(MASS) #Load package 'MASS' to call iris data
> data(iris)
> str(iris)
'data.frame':	150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
11           5.4         3.7          1.5         0.2     setosa
12           4.8         3.4          1.6         0.2     setosa
13           4.8         3.0          1.4         0.1     setosa
14           4.3         3.0          1.1         0.1     setosa
15           5.8         4.0          1.2         0.2     setosa
16           5.7         4.4          1.5         0.4     setosa
17           5.4         3.9          1.3         0.4     setosa
18           5.1         3.5          1.4         0.3     setosa
19           5.7         3.8          1.7         0.3     setosa
20           5.1         3.8          1.5         0.3     setosa
21           5.4         3.4          1.7         0.2     setosa
```



```
> data <- iris[-101:-150, ]
> data <- iris[1:100, ] #just to focus on first 100 obs.
>
> data1 <- data[data$Species == "setosa",][ ,1:4]
> data1 <- data[1:50,1:4]
> data2 <- data[data$Species == "versicolor",][ ,1:4]
> data2 <- data[51:100,1:4]
>
> n1 <- nrow(data1)
> n2 <- nrow(data2)
> xbar1 <- colMeans(data1)
> xbar2 <- colMeans(data2)
> s1 <- cov(data1)
> s2 <- cov(data2)
> spooled <- (((n1-1)*s1)+((n2-1)*s2)) / (n1+n2-2)
> inv.spooled <- solve(spooled)
> inv.spooled
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    11.634235   -6.552973    -7.998430    3.884391
Sepal.Width     -6.552973   14.236847     3.274256  -10.853906
Petal.Length    -7.998430    3.274256    21.497513  -26.658191
Petal.Width      3.884391  -10.853906   -26.658191   87.666138
> mhat<- ((xbar1-xbar2)%*%inv.spooled%*%(xbar1+xbar2)) / (2)
> mhat
          [,1]
[1,] -13.96174
> x0 <- t(as.matrix(data[40,1:4])) #assume that it is 40th rows
> y0 <- (xbar1-xbar2)%*%inv.spooled%*%x0
> y0
           40
[1,] 38.02906
> group <- ifelse(y0 >= mhat, "setosa", "versicolor")
> group
       40
[1,] "setosa"
```

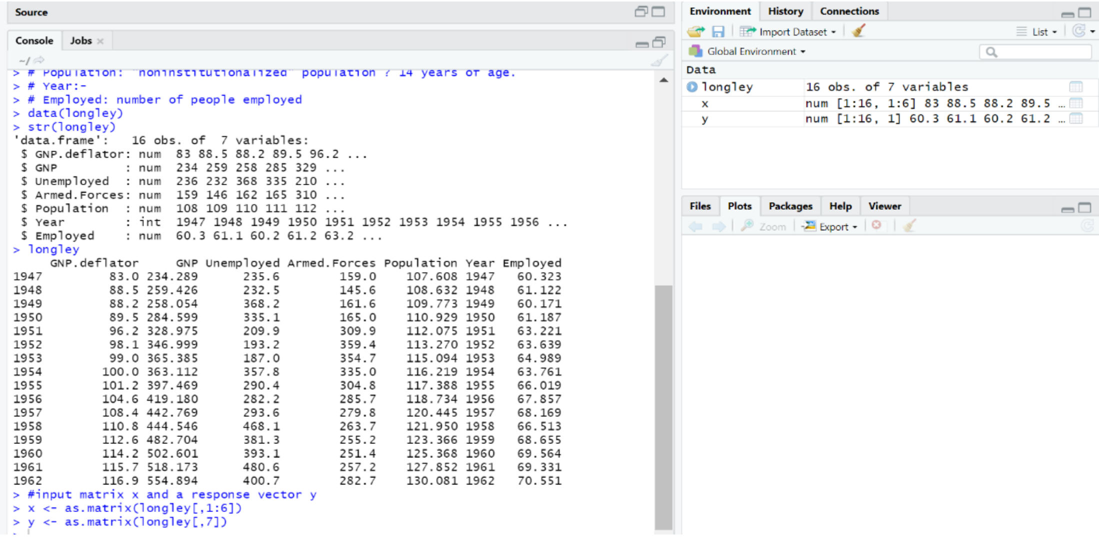
we classify it to "setosa" species since y>m as it is actual be.

Q3)According to slide 87 about “Longely” dataset in R that describes 7 economic variables observed from 1947 to 1962 used to predict the number of people employed yearly (n=16).

a. Fit classical multiple linear regression and ridge regression
b. Obtain the estimated regression coefficient. Which technique provide the smallest coefficients?
c. Perform LOOCV to get best lambda?

Solution:

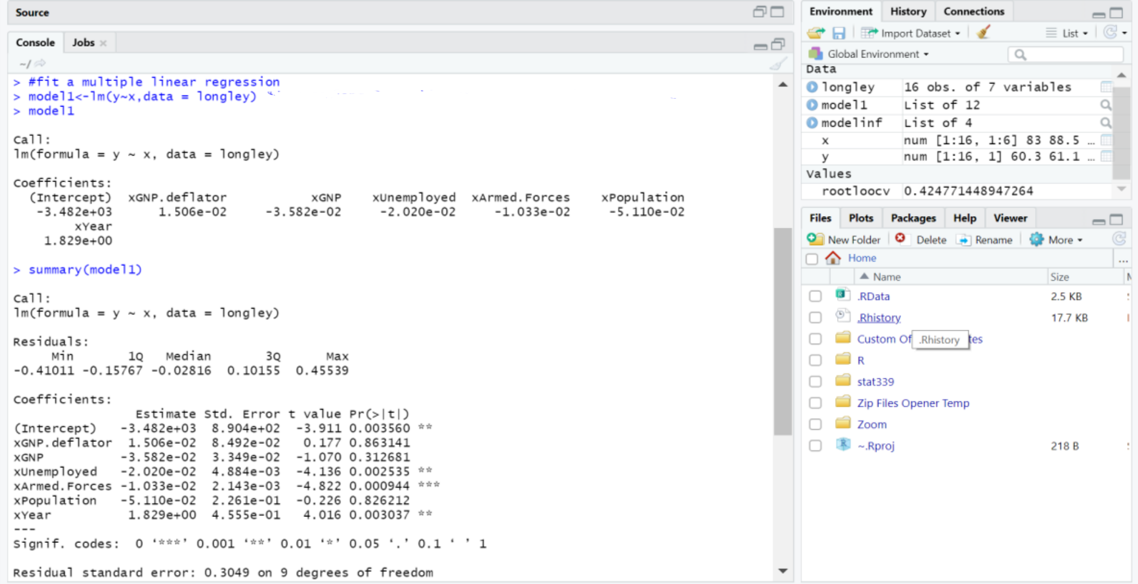
First we call the data and split it into independent and dependent variable:



```
> # Population: "noninstitutionalized" population ? 14 years of age.
> # Year:-
> # Employed: number of people employed
> data(longley)
> str(longley)
'data.frame':	16 obs. of  7 variables:
 $ GNP.deflator: num  83 88.5 88.2 89.5 96.2 ...
 $ GNP         : num  234 259 258 285 329 ...
 $ Unemployed  : num  236 232 368 335 210 ...
 $ Armed.Forces: num  159 146 162 165 310 ...
 $ Population  : num  108 109 110 111 112 ...
 $ Year        : int  1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 ...
 $ Employed    : num  60.3 61.1 60.2 61.2 63.2 ...
> longley
     GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
1947         83.0 234.289      235.6        159.0    107.608 1947   60.323
1948         88.5 259.426      232.5        145.6    108.632 1948   61.122
1949         88.2 258.054      368.2        161.6    109.773 1949   60.171
1950         89.5 284.599      335.1        165.0    110.929 1950   61.187
1951         96.2 328.975      209.9        309.9    112.075 1951   63.221
1952         98.1 346.999      193.2        359.4    113.270 1952   63.639
1953         99.0 365.385      187.0        354.7    115.094 1953   64.989
1954        100.0 363.112      357.8        335.0    116.219 1954   63.761
1955        101.2 397.469      290.4        304.8    117.388 1955   66.019
1956        104.6 419.180      282.2        285.7    118.734 1956   67.857
1957        108.4 442.769      293.6        279.8    120.445 1957   68.169
1958        110.8 444.546      468.1        263.7    121.950 1958   66.513
1959        112.6 482.704      381.3        255.2    123.366 1959   68.655
1960        114.2 502.601      393.1        251.4    125.368 1960   69.564
1961        115.7 518.173      480.6        257.2    127.852 1961   69.331
1962        116.9 554.894      400.7        282.7    130.081 1962   70.551
> #input matrix x and a response vector y
> x <- as.matrix(longley[,1:6])
> y <- as.matrix(longley[,7])
```

```
Data
longley   16 obs. of 7 variables
  x       num [1:16, 1:6] 83 88.5 88.2 89.5 ...
  y       num [1:16, 1] 60.3 61.1 60.2 61.2 ...
```

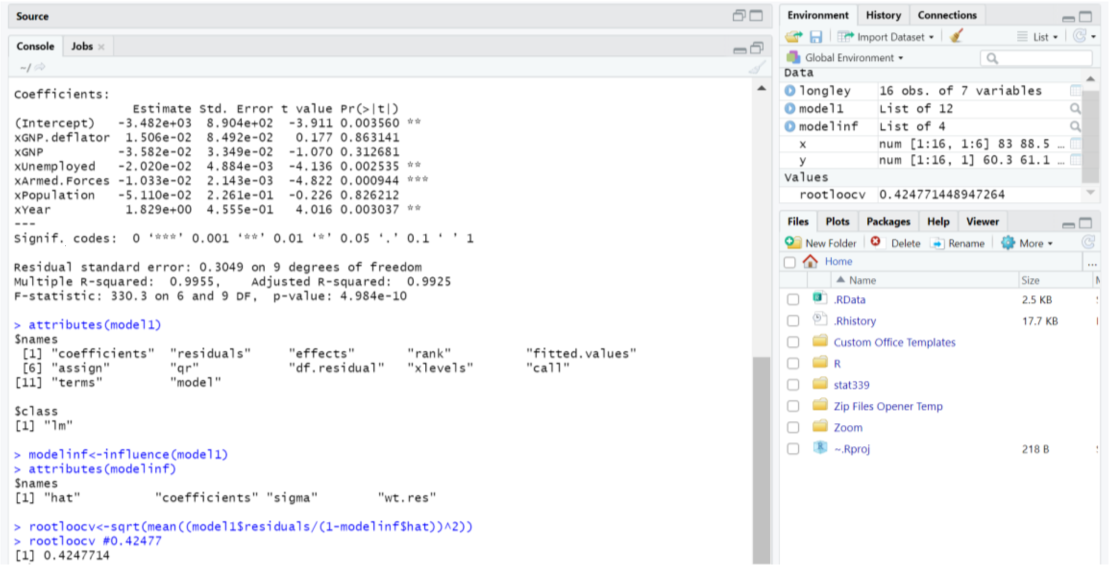
a. Fit classical multiple linear regression to find its Coefficients.



```
> #fit a multiple linear regression
> model1<-lm(y~x,data = longley)
> model1

Call:
lm(formula = y ~ x, data = longley)

Coefficients:
  (Intercept)  xGNP.deflator           xGNP    xUnemployed  xArmed.Forces    xPopulation  
   -3.482e+03      1.506e-02     -3.582e-02     -2.020e-02     -1.033e-02     -5.110e-02  
        xYear  
    1.829e+00  

> summary(model1)

Call:
lm(formula = y ~ x, data = longley)

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 ** 
xGNP.deflator  1.506e-02  8.492e-02   0.177 0.863141    
xGNP          -3.582e-02  3.349e-02  -1.070 0.312681    
xUnemployed   -2.020e-02  4.884e-03  -4.136 0.002535 ** 
xArmed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
xPopulation   -5.110e-02  2.261e-01  -0.226 0.826212    
xYear          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
```

```
Data
longley     16 obs. of 7 variables
modell      List of 12
modelinf    List of 4
  x         num [1:16, 1:6] 83 88.5 ...
  y         num [1:16, 1] 60.3 61.1 ...
Values
  rootloocv 0.424771448947264
```

Name	Size
.RData	2.5 KB
.Rhistory	17.7 KB
Custom Of[illegible]tes	
R	
stat339	
Zip Files Opener Temp	
Zoom	
~.Rproj	218 B

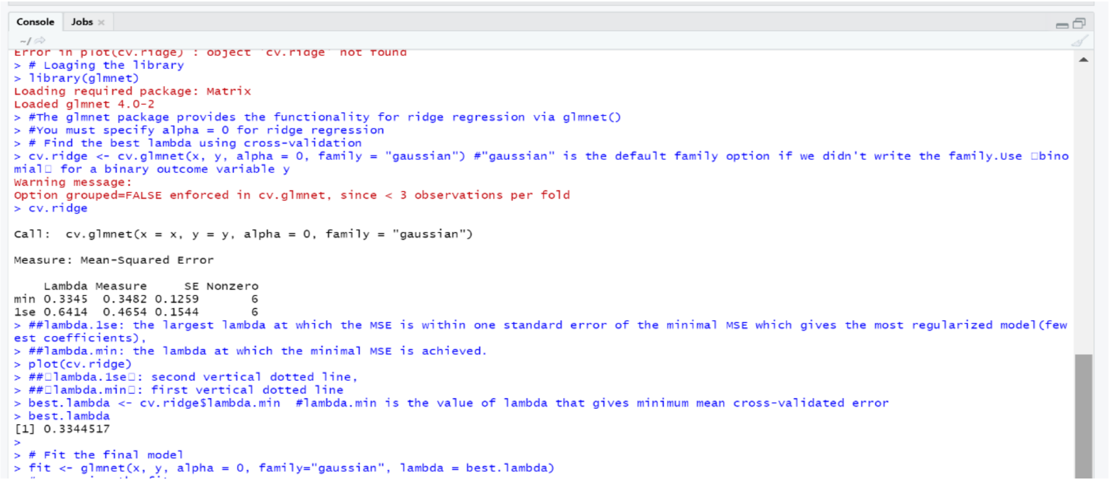


```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.482e+03  8.904e+02  -3.911 0.003560 **
xGNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
xGNP          -3.582e-02  3.349e-02  -1.070 0.312681
xUnemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
xArmed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
xPopulation   -5.110e-02  2.261e-01  -0.226 0.826212
xYear          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

> attributes(model1)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values"
 [6] "assign"        "qr"            "df.residual"   "xlevels"       "call"
[11] "terms"         "model"

$class
[1] "lm"

> modelinf<-influence(model1)
> attributes(modelinf)
$names
[1] "hat"          "coefficients" "sigma"        "wt.res"

> rootloocv<-sqrt(mean((model1$residuals/(1-modelinf$hat))^2))
> rootloocv #0.42477
[1] 0.4247714
```

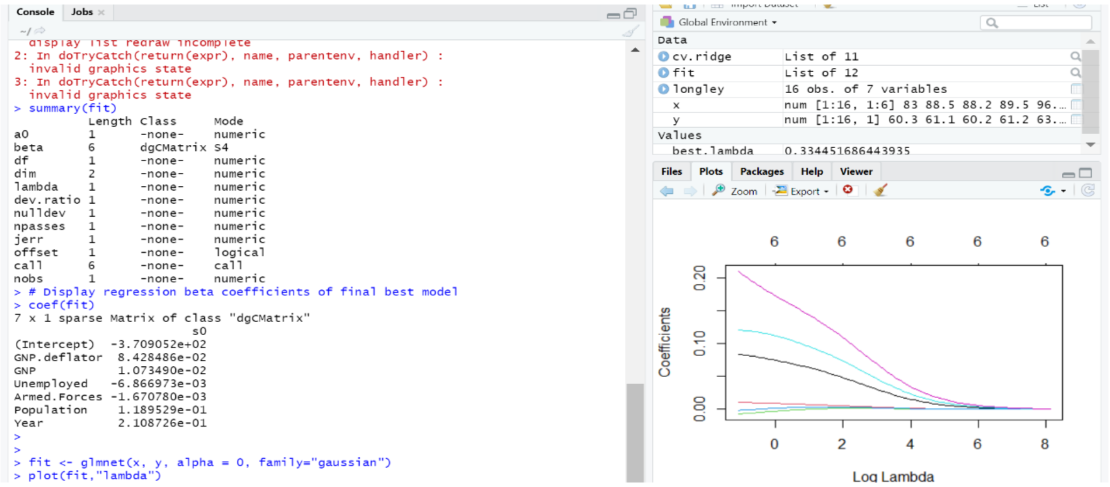
```
Data
  longley     16 obs. of 7 variables
  model1      List of 12
  modelinf    List of 4
  x           num [1:16, 1:6] 83 88.5 ...
  y           num [1:16, 1] 60.3 61.1 ...
Values
  rootloocv   0.424771448947264
```

Name	Size
.RData	2.5 KB
.Rhistory	17.7 KB
Custom Office Templates	
R	
stat339	
Zip Files Opener Temp	
Zoom	
~.Rproj	218 B

Fit ridge regression to find its Coefficients.



```
Error in plot(cv.ridge) : object 'cv.ridge' not found
> # Loaging the library
> library(glmnet)
Loading required package: Matrix
Loaded glmnet 4.0-2
> #The glmnet package provides the functionality for ridge regression via glmnet()
> #You must specify alpha = 0 for ridge regression
> # Find the best lambda using cross-validation
> cv.ridge <- cv.glmnet(x, y, alpha = 0, family = "gaussian") #"gaussian" is the default family option if we didn't write the family.Use □bino
mial□ for a binary outcome variable y
Warning message:
Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per fold
> cv.ridge

Call:  cv.glmnet(x = x, y = y, alpha = 0, family = "gaussian")

Measure: Mean-Squared Error

    Lambda Measure     SE Nonzero
min 0.3345  0.3482 0.1259       6
1se 0.6414  0.4654 0.1544       6
> ##lambda.1se: the largest lambda at which the MSE is within one standard error of the minimal MSE which gives the most regularized model(few
est coefficients),
> ##lambda.min: the lambda at which the minimal MSE is achieved.
> plot(cv.ridge)
> ##□lambda.1se□: second vertical dotted line,
> ##□lambda.min□: first vertical dotted line
> best.lambda <- cv.ridge$lambda.min  #lambda.min is the value of lambda that gives minimum mean cross-validated error
> best.lambda
[1] 0.3344517
>
> # Fit the final model
> fit <- glmnet(x, y, alpha = 0, family="gaussian", lambda = best.lambda)
```



```
  display list redraw incomplete
2: In doTryCatch(return(expr), name, parentenv, handler) :
  invalid graphics state
3: In doTryCatch(return(expr), name, parentenv, handler) :
  invalid graphics state
> summary(fit)
          Length Class     Mode
a0        1      -none-    numeric
beta      6      dgCMatrix S4
df        1      -none-    numeric
dim       2      -none-    numeric
lambda    1      -none-    numeric
dev.ratio 1      -none-    numeric
nulldev   1      -none-    numeric
npasses   1      -none-    numeric
jerr      1      -none-    numeric
offset    1      -none-    logical
call      6      -none-    call
nobs      1      -none-    numeric
> # Display regression beta coefficients of final best model
> coef(fit)
7 x 1 sparse Matrix of class "dgCMatrix"
                         s0
(Intercept)   -3.709052e+02
GNP.deflator   8.428486e-02
GNP            1.073490e-02
Unemployed    -6.866973e-03
Armed.Forces  -1.670780e-03
Population     1.189529e-01
Year           2.108726e-01
>
>
> fit <- glmnet(x, y, alpha = 0, family="gaussian")
> plot(fit,"lambda")
```

```
Data
  cv.ridge      List of 11
  fit           List of 12
  longley       16 obs. of 7 variables
  x             num [1:16, 1:6] 83 88.5 88.2 89.5 96....
  y             num [1:16, 1] 60.3 61.1 60.2 61.2 63....
Values
  best.lambda   0.334451686443935
```

b.

- Increasing lambda increases the shrinking of the coefficients.
- If the sample size is very small compared to the number of covariates, estimation is not efficient and therefore we might not get the desirable shrinking.