# 3. Multiple Regession

Thu Trang Luu

11/2/2023

\subsection Exercise 3.2

```
if(!require(Ecdat)) install.packages("Ecdat", repos="http://R-Forge.R-project.org")
```

```
## Loading required package: Ecdat
```

```
## Warning: package 'Ecdat' was built under R version 4.1.3
```

```
## Loading required package: Ecfun
```

```
## Warning: package 'Ecfun' was built under R version 4.1.3
```

```
##
## Attaching package: 'Ecfun'
```

```
## The following object is masked from 'package:base':
##
##     sign
```

```
##
## Attaching package: 'Ecdat'
```

```
## The following object is masked from 'package:datasets':
##
##     Orange
```

```
library("Ecdat")
data("Airq")
# df to matrix
air_data = data.matrix(Airq)
# scaling data
scaled_air_data = cbind(air_data[,1],scale(air_data[,-1]))
# make matrix X with a column of 1 for intercept
X=cbind(1,scaled_air_data[,-1])
```

\subsection Exercise 3.4

```
set.seed(123)
RSS <- function (beta,y,X){
  # calculate RSS of a certain beta vector
  e = y-X%*%beta
  return(t(e)%*%e)
}

MM_34 <- function (y,X,verbose=FALSE){
  eps = 0.0000001 # set epsilon
  p=ncol(X) # find number of columns

  beta_0 = matrix(rnorm(p),p,1) # random beta_0
  RSS_0 = RSS(beta_0,y,X) # RSS of beta_0
  RSS_k = 0 # set RSS of beta_k
  XtX = t(X)%*%X
  lambda = max(eigen(XtX)$values) # find lambda as maximum eigenvalue

  k = 1 # iteration
  while (k == 1 || (RSS_0 - RSS_k)/RSS_0 > eps){

    # Calculate new beta_k
    beta_k = beta_0 - lambda**-1*XtX%*%beta_0 + lambda**-1*t(X)%*%y
    # Find RSS(beta_k)
    RSS_k = RSS(beta_k,y,X)

    RSS_0 = RSS(beta_0,y,X)

    # Print progress
    if (verbose){
      cat("Iter =",k,
          "RSS(beta_k) =",RSS_k,
          "RSS(beta_k-1) - RSS(beta_k) =",RSS_0-RSS_k,
          "RSS(beta_k-1) =",RSS_0,"\n")
    }
    # Update beta
    beta_0 = beta_k

    # Next iter
    k=k+1
  }
  return(beta_0)
}
```

```
MM_34(scaled_air_data[,1],X)
```

```
##             [,1]
##       104.700000
## vala    4.138040
## rain    3.382994
## coas  -15.558365
## dens   -3.033145
## medi    6.881840
```

```
model34 = lm(Airq$airq~scaled_air_data[,-1])
summary(model34)
```

```
##
## Call:
## lm(formula = Airq$airq ~ scaled_air_data[, -1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.958  -9.891  -6.173  13.714  69.430
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 104.700      4.419  23.694   <2e-16 ***
## scaled_air_data[, -1]vala     4.090     10.447   0.392   0.6989
## scaled_air_data[, -1]rain     3.382      4.634   0.730   0.4726
## scaled_air_data[, -1]coas   -15.567      4.874  -3.194   0.0039 **
## scaled_air_data[, -1]dens    -3.035      4.590  -0.661   0.5148
## scaled_air_data[, -1]medi     6.931     10.627   0.652   0.5205
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 24.2 on 24 degrees of freedom
## Multiple R-squared:  0.3829, Adjusted R-squared:  0.2544
## F-statistic: 2.979 on 5 and 24 DF,  p-value: 0.03133
```

```
round(model34$coefficients,4)
```

```
##               (Intercept) scaled_air_data[, -1]vala scaled_air_data[, -1]rain
##                  104.7000                    4.0903                    3.3815
## scaled_air_data[, -1]coas scaled_air_data[, -1]dens scaled_air_data[, -1]medi
##                  -15.5667                   -3.0352                    6.9306
```

\subsection Exercise 3.5

```
MM_better_subset <- function (y,X,m=0,verbose=FALSE){
  eps = 0.0000001 # set epsilon
  p=ncol(X) # find number of columns

  beta_0 = matrix(rbind(rnorm(m),rep(0,p-m)),p,1)
  RSS_0 = RSS(beta_0,y,X) # RSS of beta_0
  RSS_k = 0 # set RSS of beta_k
  XtX = t(X)%*%X
  lambda = max(eigen(XtX)$values) # find lambda as maximum eigenvalue

  k = 1
  while (k == 1 || (RSS_0 - RSS_k)/RSS_0 > eps){
    u = beta_0 - lambda**-1*t(X)%*%X%*%beta_0 + lambda**-1*t(X)%*%y
    beta_k = matrix(0,p,1)
    beta_k[1] = u[1]
    phi = sort.int(abs(u), decreasing = TRUE,index.return = TRUE)
    for (i in 1:m){
      beta_k[phi$ix[i]] = u[phi$ix[i]]
    }

    RSS_k = RSS(beta_k,y,X)
    RSS_0 = RSS(beta_0,y,X)

    if (verbose){
      cat("Iter =",k,
          "RSS(beta_k) =",RSS_k,
          "RSS(beta_k-1) - RSS(beta_k) =",RSS_0-RSS_k,
          "RSS(beta_k-1) =",RSS_0,"\n")
    }

    beta_0 = beta_k
    k=k+1
  }
  return(beta_0)
}
```

```
MM_better_subset(scaled_air_data[,1],X,m=3)
```

```
##            [,1]
## [1,] 104.696387
## [2,]   9.257619
## [3,]   0.000000
## [4,] -13.831263
## [5,]   0.000000
## [6,]   0.000000
```