Assignment 1: R Programming

Sagar Kumar | 194161013

```
knitr::opts_chunk$set(echo = TRUE)
```

Importing Data from csvfile

```
df<-read.csv("xy.csv") #read data in csv file.
```

x "independent variable"
y "dependent variable"

Data Set

```
df  #dataframe
```

```
##        x     y
## 1  46.75 92.64
## 2  42.18 88.81
## 3  41.86 86.44
## 4  43.29 88.80
## 5  42.12 86.38
## 6  41.78 89.87
## 7  41.47 88.53
## 8  42.21 91.11
## 9  41.03 81.22
## 10 39.84 83.72
## 11 39.15 84.54
## 12 39.20 85.66
## 13 39.52 85.87
## 14 38.05 85.23
## 15 39.16 87.75
## 16 38.59 92.62
## 17 36.54 91.56
## 18 37.03 84.12
## 19 36.60 81.22
## 20 37.58 83.35
## 21 36.48 82.29
## 22 38.25 80.92
## 23 37.26 76.92
## 24 38.59 78.35
## 25 40.89 74.57
## 26 37.66 71.60
## 27 38.79 65.64
## 28 38.78 62.09
## 29 36.70 61.66
## 30 35.10 77.14
## 31 33.75 75.47
## 32 34.29 70.37
```

```
## 33 32.26 66.71
## 34 30.97 64.37
## 35 28.20 56.09
## 36 24.58 50.25
## 37 20.25 43.65
## 38 17.09 38.01
## 39 14.35 31.40
## 40 13.11 29.45
## 41  9.50 29.02
## 42  9.74 19.05
## 43  9.34 20.36
## 44  7.51 17.68
## 45  8.35 19.23
## 46  6.25 14.92
## 47  5.45 11.44
## 48  3.79 12.69
```

48 Observation Two Variables y is response variable and x is input variable.

Question 1

A simple least squares model y = B0+B1*x from first principles and obtain B0 and B1.

```
Y<-df$y #dependent Variable
intercept<-rep(1,length(Y)) #replicates the values.
#intercept
```

Design Matrix

```
X<-as.matrix(cbind(intercept,df$x))
X  #Design Matrix
```

```
##       intercept
##  [1,]         1 46.75
##  [2,]         1 42.18
##  [3,]         1 41.86
##  [4,]         1 43.29
##  [5,]         1 42.12
##  [6,]         1 41.78
##  [7,]         1 41.47
##  [8,]         1 42.21
##  [9,]         1 41.03
## [10,]         1 39.84
## [11,]         1 39.15
## [12,]         1 39.20
## [13,]         1 39.52
## [14,]         1 38.05
## [15,]         1 39.16
## [16,]         1 38.59
## [17,]         1 36.54
## [18,]         1 37.03
## [19,]         1 36.60
## [20,]         1 37.58
## [21,]         1 36.48
## [22,]         1 38.25
## [23,]         1 37.26
## [24,]         1 38.59
## [25,]         1 40.89
## [26,]         1 37.66
## [27,]         1 38.79
## [28,]         1 38.78
## [29,]         1 36.70
## [30,]         1 35.10
## [31,]         1 33.75
## [32,]         1 34.29
## [33,]         1 32.26
## [34,]         1 30.97
## [35,]         1 28.20
## [36,]         1 24.58
## [37,]         1 20.25
## [38,]         1 17.09
## [39,]         1 14.35
## [40,]         1 13.11
## [41,]         1  9.50
## [42,]         1  9.74
```

```
## [43,]         1  9.34
## [44,]         1  7.51
## [45,]         1  8.35
## [46,]         1  6.25
## [47,]         1  5.45
## [48,]         1  3.79
```

Estmiating Coefficient

```
XtX_matrix<-solve(crossprod(X)) #crossprod ->X'X (inverse of X matrix)
#d_matrix
b=XtX_matrix  %*% crossprod(X,Y)  #Matrix Multiplication(%*%)
b
```

```
##               [,1]
## intercept 3.128201
##           2.005476
```

So **intercept** is: 3.128201
and **slope** is : 2.005476

Find B0 and B1 using the linear model package lm.

```
lm.fit<-lm(df$y~df$x,data=df) #lm function
coef(lm.fit)
```

```
## (Intercept)        df$x
##    3.128201    2.005476
```

So result is same .

Generate 100 values of x between [3:00; 50:00] and find the value of y

```
random_no<-runif(100,3,50)  #Generate random no
random_no
```

```
##   [1]  4.841876 42.412438  6.656206  9.402746 32.351082 27.232139  7.709168
##   [8] 10.768057 44.070033 45.577271 37.475751 39.052250 26.616560 41.458435
##  [15] 13.124296 47.789346 37.461831 28.500409 40.102999 11.019767 40.088667
##  [22] 41.926313 37.906489 17.026728 35.837940 21.398779 30.381435 31.427410
##  [29] 17.650299 38.851025 24.368319 19.516642 12.711852 14.328293 20.733200
##  [36] 40.277468 22.480454 44.160983 22.193824  7.533848 11.817460 19.394242
##  [43] 15.644676 43.331827 16.609495  6.334897 38.334793 26.645779 33.898589
##  [50] 16.181738  4.387496 11.785874 24.162450 14.286239 33.642186 19.859327
##  [57] 28.969128 37.897100 20.034477  4.994606 30.675928  8.502890 19.608900
##  [64] 35.383830 38.234121 49.339877 36.564337 22.937353 28.120064 39.027861
##  [71]  7.182412  8.862171 26.680459 46.962561 33.131555  3.748842 38.196961
##  [78] 26.192993 42.357243 15.915256 39.642358 38.859784 48.242362 38.138200
##  [85] 15.125787 13.492409 46.530629  4.946465  5.726662 42.176809 36.268605
##  [92] 35.965282 17.881280  7.389928 28.702087 26.889636 27.660395 21.218335
##  [99] 23.148775 31.572927
```

```
y=3.128+2.005*random_no
y #yvalues
```

```
##   [1]  12.83596  88.16494  16.47369  21.98051  67.99192  57.72844  18.58488
##   [8]  24.71795  91.48842  94.51043  78.26688  81.42776  56.49420  86.25216
##  [15]  29.44221  98.94564  78.23897  60.27132  83.53451  25.22263  83.50578
##  [22]  87.19026  79.13051  37.26659  74.98307  46.03255  64.04278  66.13996
##  [29]  38.51685  81.02430  51.98648  42.25887  28.61526  31.85623  44.69807
##  [36]  83.88432  48.20131  91.67077  47.62662  18.23337  26.82201  42.01346
##  [43]  34.49558  90.00831  36.43004  15.82947  79.98926  56.55279  71.09467
##  [50]  35.57238  11.92493  26.75868  51.57371  31.77191  70.58058  42.94595
##  [57]  61.21110  79.11169  43.29713  13.14218  64.63323  20.17629  42.44385
##  [64]  74.07258  79.78741 102.05445  76.43950  49.11739  59.50873  81.37886
##  [71]  17.52874  20.89665  56.62232  97.28793  69.55677  10.64443  79.71291
##  [78]  55.64495  88.05427  35.03809  82.61093  81.04187  99.85394  79.59509
##  [85]  33.45520  30.18028  96.42191  13.04566  14.60996  87.69250  75.84655
##  [92]  75.23839  38.97997  17.94481  60.67568  57.04172  58.58709  45.67076
##  [99]  49.54129  66.43172
```