Practical - 6 Shivam Tawari A-58

- Aim : Write a program in R for implementing Linear Regression.

- Theory :

A linear regression is one of the easiest statistical models in machine learning. Ir is used to show the linear relationship between a dependent variable and one or more independent variable.

Regression :

Regression Analysis is a form of predictive modelling technique which investigates the relationship between a dependent and an independent variable.

There are 3 types of regression :-

1) Linear Regression
2) Logistic Regression
3) Polynoomial Regression

Least Squares Method :

In this case, linear regression assumes that there exists a linear relationship between the response variable and the explanatory variables.

Least squares is a statistical method used to determine the best fit line or the regression line by minimizing the sum of squares created by a mathematical fn.

The "square" here refers to squaring the distance between a data point and the regression line.

The line with the maximum value of the sum of square is the best-fit regression line.

The line with the minimum value of the sum of square is the best fit regression line. Equation of regression line in simplest form:

$$y = mx + c$$ where,

$y$ = Dependent variable
$x$ = Independent variable
$c$ = $y$- intercept

In general,

$$h_{\theta}(x) = \sum_{i=0}^{n} \theta_i x_i$$

where $h_{\theta}(x)$ is the hypothesis fn.
$\theta_i$ are the parameters
$x_i$ are the feature variables with
$x_0 = 1$.

- lm () function :

This function creates the relationship model between the predictor and the response Variable .

Syntax :

The basic syntax for lm () function in linear regression :

- formula is a symbol presenting the relation between x and y.
- data is the vector on which the formula applied .

- predict () function :

predict (object , newdata)

- object is the formula which is already created using lm() function .
- newdata is the vector containing the new value for predictor variable .

Code :

```
## Example 1 : Predicting weight of a person
## when height is known
x ← c (151, 174, 138, 186, 128, 136, 179, 163, 152, 13
y ← c (63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation ← lm (y ~ x)
print (relation)
print (summary (relation, a))
```

⇒ Output : 076.22869

```
## Example 2 : Calculate height of a child
## if age is given .

install.packages ("readxl")
library (readxl)

ageandht ← read_excel ('ageandheight2.xls',
                        sheet = 'Hoja2')
lmheight ← lm (height ~ age , data = ageandht)

str (ageandht)
names (ageandht)
summary (lmheight)
```

⇒ Output :

```
              Estimate Std.   Error
(Intercept)     4.9283        0.5084
  Age           0.6350        0.0214
```

## Example 3 : Income and happiness dataset

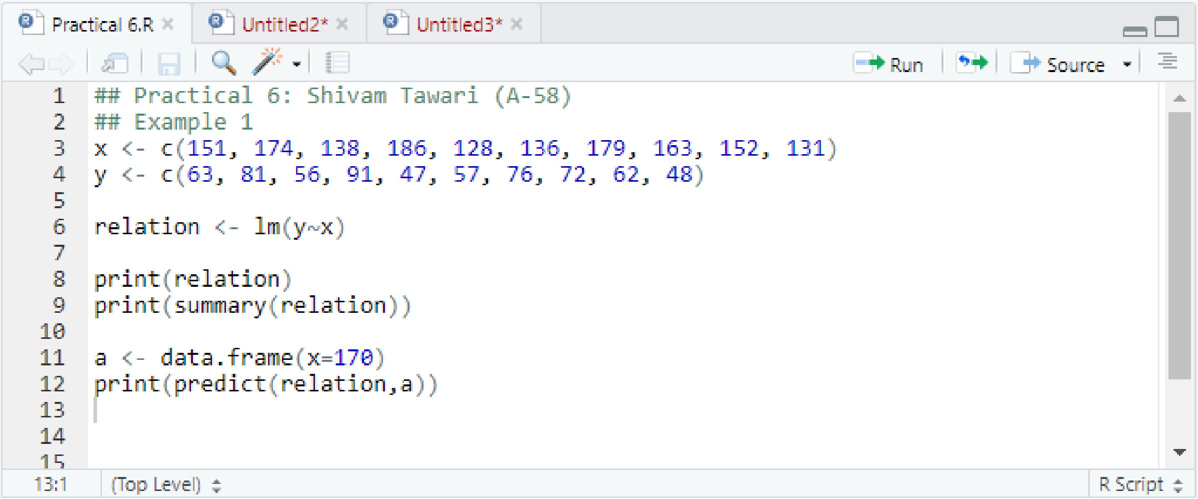
```
install.packages ("ggplot2")
install.packages ("dplyr")
install.packages ("broom")
install.packages ("ggpubr")
install.packages ("readxl")

library (ggplot2)
library (dplyr)
library (broom)
library (ggpubr)
library (readxl)

income <- read_excel('income2.xls', sheet = 'income')
summary (income)

hist (income $ happiness)
plot ( happiness ~ income, data = income )

income.happiness.lm <- lm (happiness ~ income,
                              data = income )
```

```
summary (income . happiness . lm)

predict (income . happiness . lm, data.frame (income = 7.95))
```

⇒ Output : 5.88

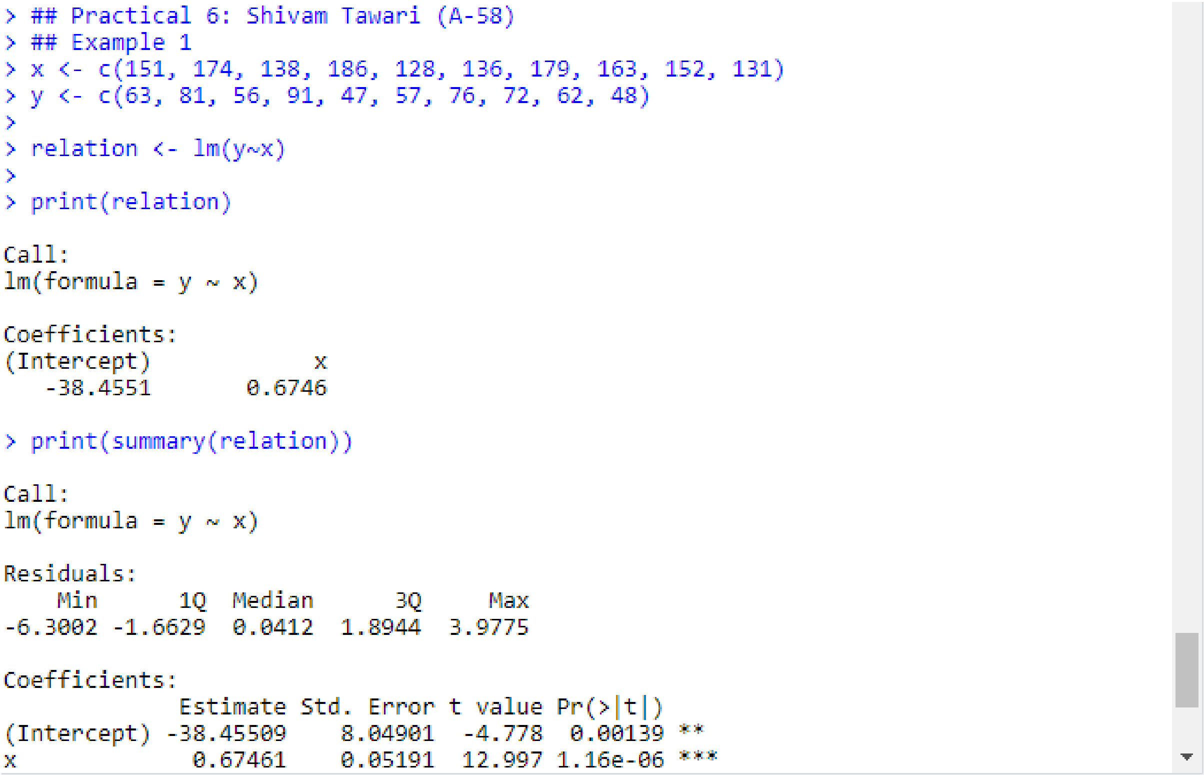
Conclusion : Hence, implemented Linear regression in R.

# Code:

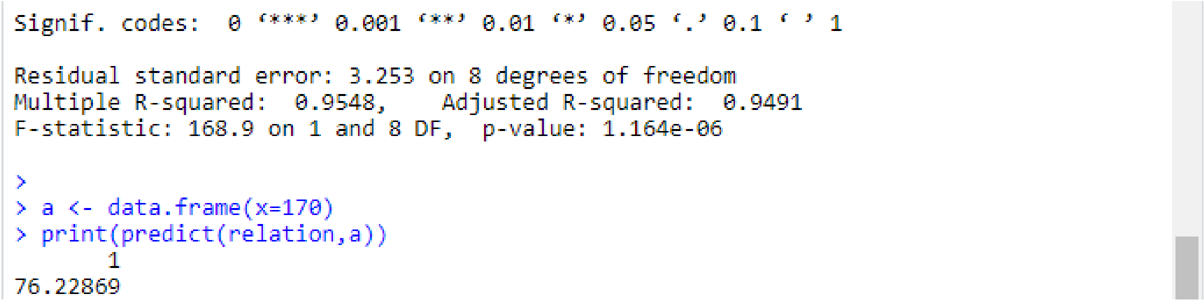


```
## Practical 6: Shivam Tawari (A-58)
## Example 1
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

relation <- lm(y~x)

print(relation)
print(summary(relation))

a <- data.frame(x=170)
print(predict(relation,a))
```

# Output:

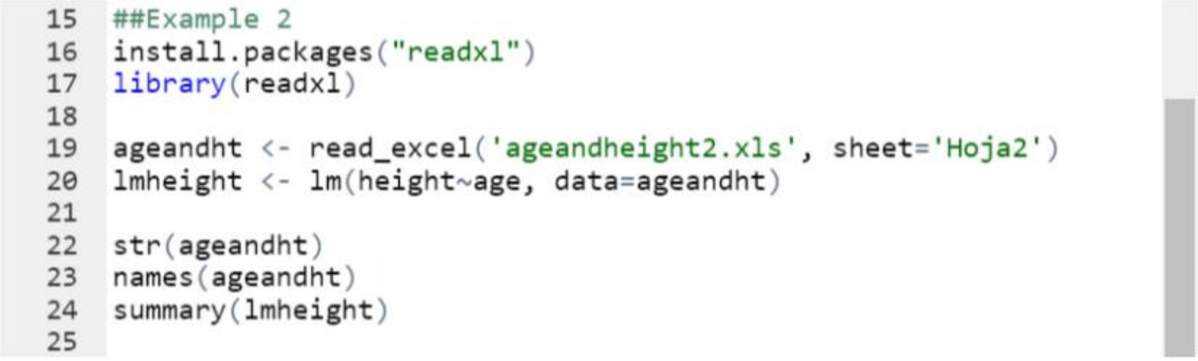


```
> ## Practical 6: Shivam Tawari (A-58)
> ## Example 1
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
>
> relation <- lm(y~x)
>
> print(relation)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551       0.6746

> print(summary(relation))

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002 -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
```



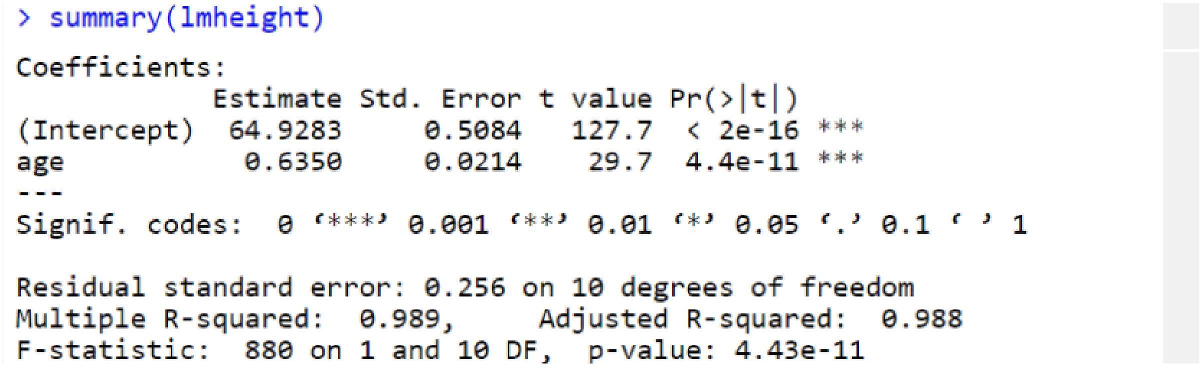
```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06

>
> a <- data.frame(x=170)
> print(predict(relation,a))
       1
76.22869
```

**Code:**



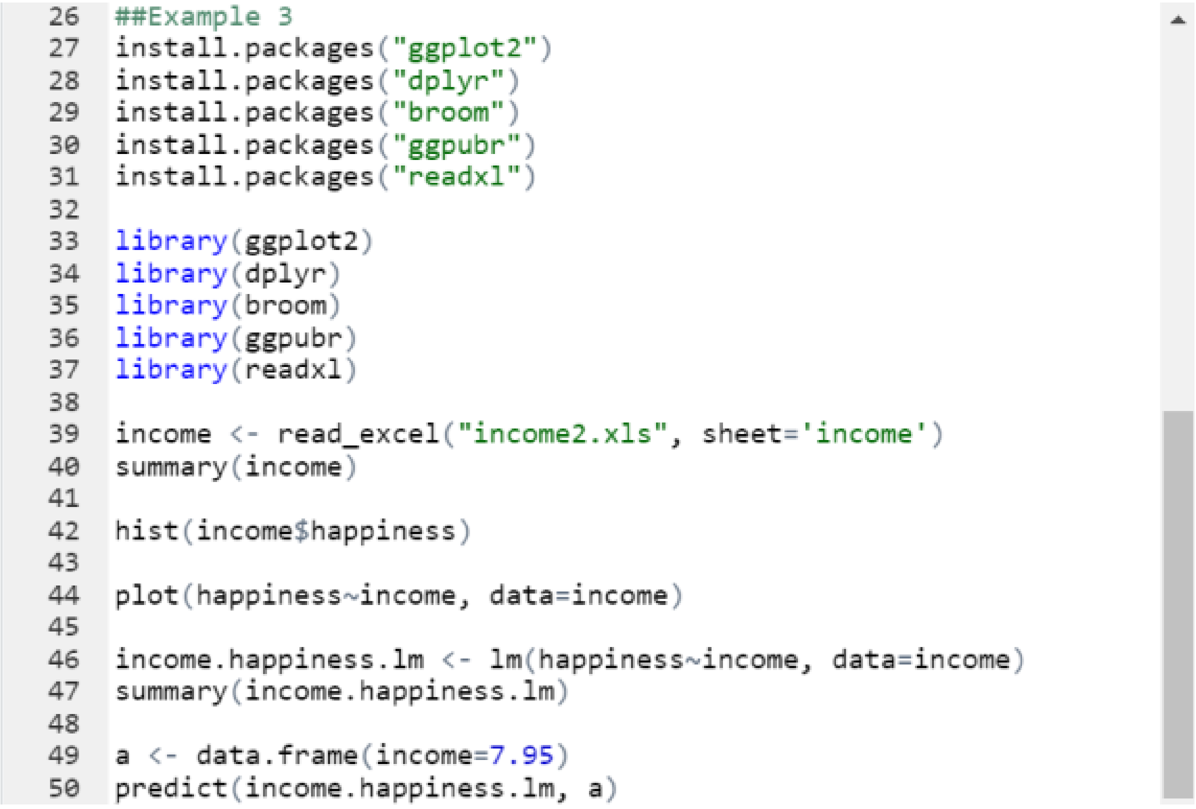
```
##Example 2
install.packages("readxl")
library(readxl)

ageandht <- read_excel('ageandheight2.xls', sheet='Hoja2')
lmheight <- lm(height~age, data=ageandht)

str(ageandht)
names(ageandht)
summary(lmheight)

```

**Output:**



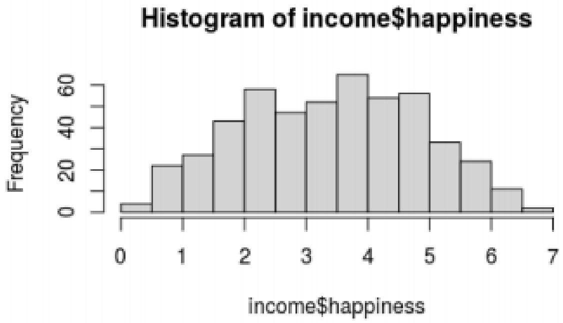
```
> summary(lmheight)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  64.9283     0.5084   127.7  < 2e-16 ***
age           0.6350     0.0214    29.7  4.4e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.256 on 10 degrees of freedom
Multiple R-squared:  0.989,	Adjusted R-squared:  0.988
F-statistic:  880 on 1 and 10 DF,  p-value: 4.43e-11
```

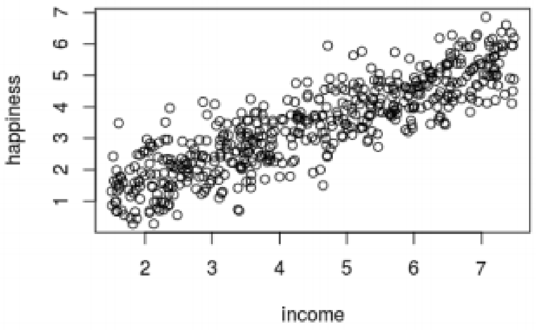
**Code:**

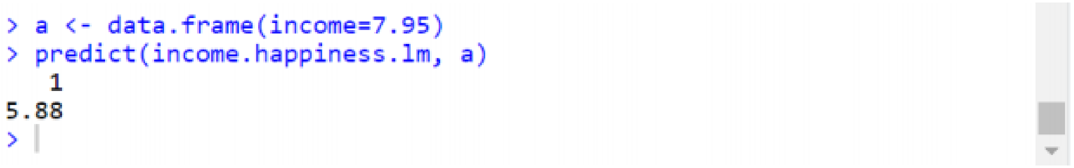


```
##Example 3
install.packages("ggplot2")
install.packages("dplyr")
install.packages("broom")
install.packages("ggpubr")
install.packages("readxl")

library(ggplot2)
library(dplyr)
library(broom)
library(ggpubr)
library(readxl)

income <- read_excel("income2.xls", sheet='income')
summary(income)

hist(income$happiness)

plot(happiness~income, data=income)

income.happiness.lm <- lm(happiness~income, data=income)
summary(income.happiness.lm)

a <- data.frame(income=7.95)
predict(income.happiness.lm, a)
```

## Output:







```
> a <- data.frame(income=7.95)
> predict(income.happiness.lm, a)
   1
5.88
> 
```