Ratnagiri Education Society's

R. P. GOGATE COLLEGE OF ARTS AND SCIENCE AND

R. V. JOGALEKAR COLLEGE

OF COMMERCE.

Department Of Information Technology

Practical 7

Roll No: TTA01	**Class: TY BSc IT**
Subject: Business Intelligence	**Semester: 6**
Date:	**Sign:**

Aim: Prediction Using Linear Regression

In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

y = ax + b is an equation for linear regression. Where, y is the response variable, x is the predictor variable and a and b are constants which are called the coefficients.

A simple example of regression is predicting weight of a person when his height is known.

To do this we need to have the relationship between height and weight of a person.

The steps to create the relationship is –

- Carry out the experiment of gathering a sample of observed values of height and corresponding weight.
- Create a relationship model using the lm() functions in R.
- Find the coefficients from the model created and create the mathematical equation using these
- Get a summary of the relationship model to know the average error in prediction. Also called residuals.
- To predict the weight of new persons, use the predict() function in R.

➢ **Input Data**

Below is the sample data representing the observations −

# Values of height

151, 174, 138, 186, 128, 136, 179, 163, 152, 131

# Values of weight.

63, 81, 56, 91, 47, 57, 76, 72, 62, 48

lm() Function: This function creates the relationship model between the predictor and the response variable.

- Syntax

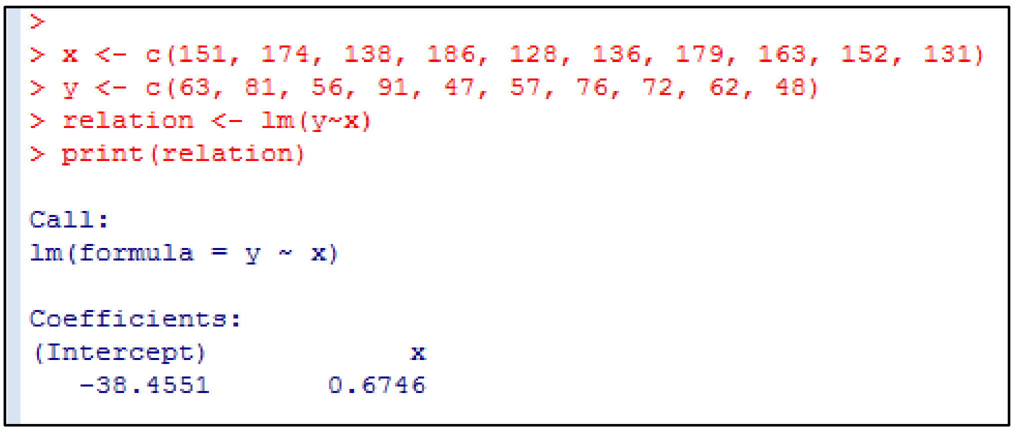
lm(formula,data)

Following is the description of the parameters used −

• formula is a symbol presenting the relation between x and y.

• data is the vector on which the formula will be applied.

Step 1: Create Relationship Model & get the Coefficients

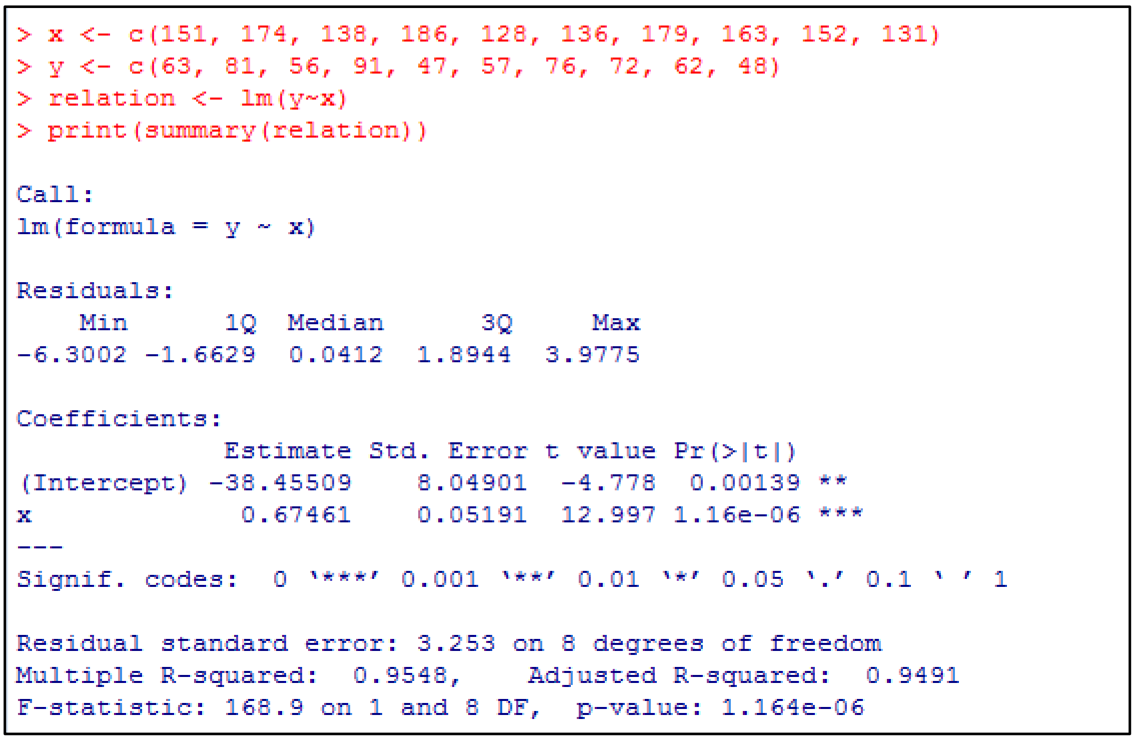


```
>
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> relation <- lm(y~x)
> print(relation)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x  
   -38.4551       0.6746  
```

Step 2: Get the Summary of the Relationship



```
> 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(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
```

- **predict() Function**
- **Syntax**

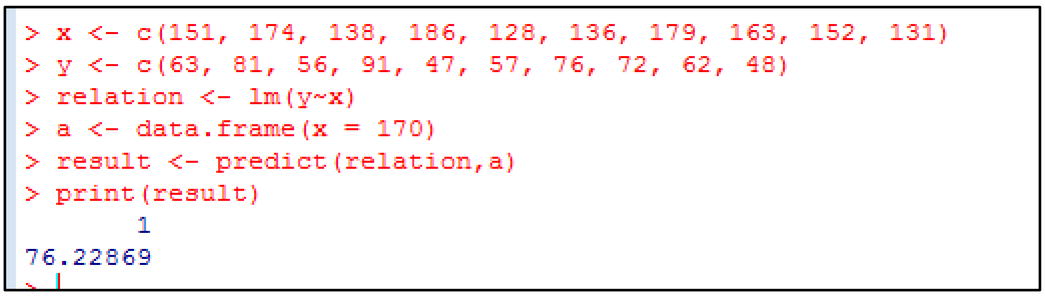
The basic syntax for predict() in linear regression is −

predict(object, newdata)

Following is the description of the parameters used −

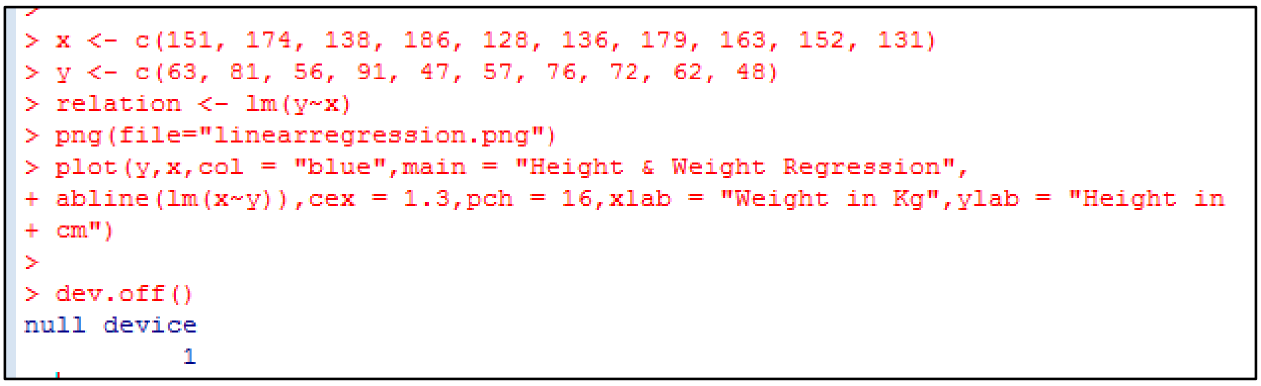
- object is the formula which is already created using the lm() function.
- newdata is the vector containing the new value for predictor variable.

Step 3: Predict the weight of new persons



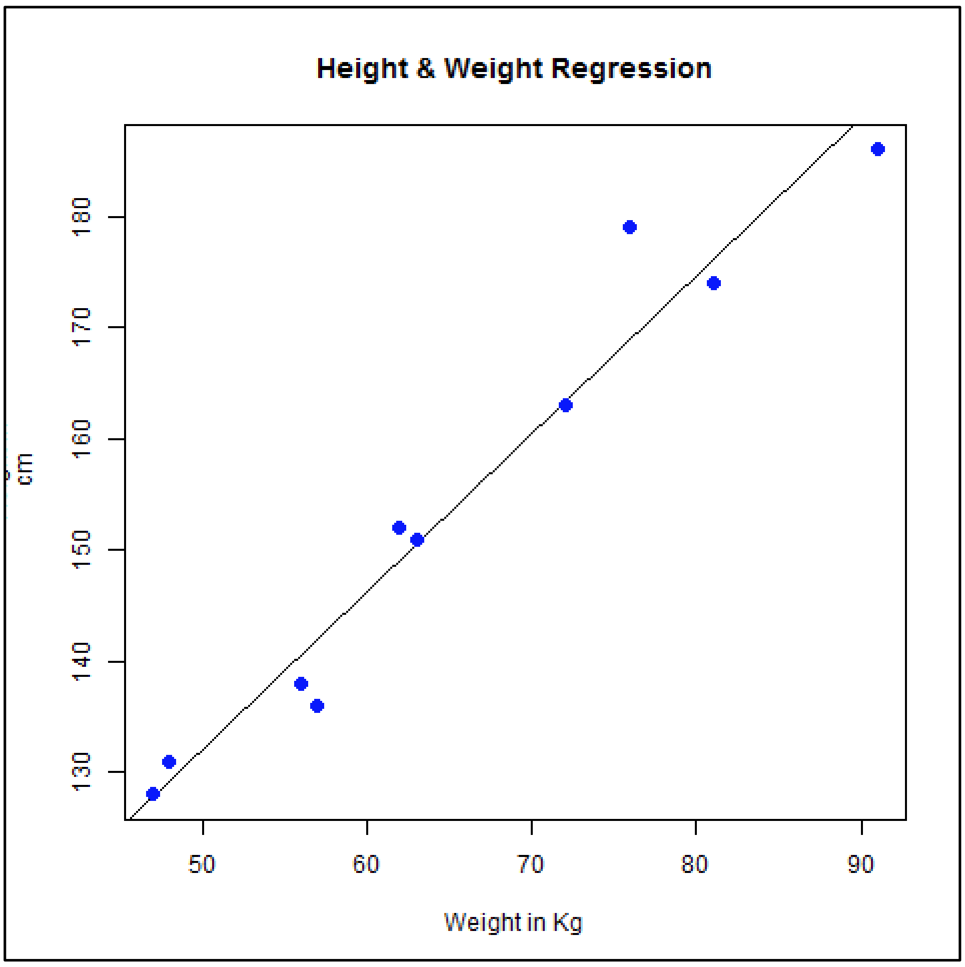
```
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> relation <- lm(y~x)
> a <- data.frame(x = 170)
> result <- predict(relation,a)
> print(result)
       1
76.22869
> 
```

Step 4: Visualize the Regression Graphically



```
> 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)
> png(file="linearregression.png")
> plot(y,x,col = "blue",main = "Height & Weight Regression",
+ abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in
+ cm")
>
> dev.off()
null device
          1
```

Output:


Height & Weight Regression
cm
Weight in Kg