# Write R program to find the following Regressions with the sample data and visualize the regressions graphically

1. Linear Regression
2. Multiple Regression
3. Logistic Regression
4. Poisson Regression

# R - Linear Regression

Regression analysis is a very widely used statistical tool to establish a relationship model between two variables. One of these variable is called predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

The general mathematical equation for a linear regression is −

**Y = aX+b**

Following is the description of the parameters used −

**y** is the response variable.

**x** is the predictor variable.

**a** and **b** are constants which are called the coefficients.

## 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 is −

**lm(formula,data)**

### Create Relationship Model & get the Coefficients

1. x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
2. y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
3. # Apply the lm() function.
4. relation <- lm(y~x)
5. print(relation)

### 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)

# Apply the lm() function.

relation <- lm(y~x)

print(summary(relation))

### Predict the weight of new persons

# The predictor vector.

x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# The resposne vector.

y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.

relation <- lm(y~x)

# Find weight of a person with height 170.

a <- data.frame(x = 170)

result <- predict(relation,a)

print(result)

### Visualize the Regression Graphically

# Create the predictor and response variable.

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)

# Give the chart file a name.

png(file = "linearregression.png")

# Plot the chart.

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")

# Save the file.

dev.off()

# R - Multiple Regression

Multiple regressions is an extension of linear regression into relationship between more than two variables. In simple linear relation we have one predictor and one response variable, but in multiple regression we have more than one predictor variable and one response variable.

The general mathematical equation for multiple regressions is −

**y = a + b1x1 + b2x2 +...bnxn**

### Syntax

The basic syntax for lm() function in multiple regression is −

**lm(y ~ x1+x2+x3...,data)**

Following is the description of the parameters used −

* formula is a symbol presenting the relation between the response variable and predictor variables.
* data is the vector on which the formula will be applied.

## Example

### Input Data

input <- mtcars[,c("mpg","disp","hp","wt")]

print(head(input))

### Create Relationship Model & get the Coefficients

input <- mtcars[,c("mpg","disp","hp","wt")]

# Create the relationship model.

model <- lm(mpg~disp+hp+wt, data = input)

# Show the model.

print(model)

# Get the Intercept and coefficients as vector elements.

cat("# # # # The Coefficient Values # # # ","\n")

a <- coef(model)[1]

print(a)

Xdisp <- coef(model)[2]

Xhp <- coef(model)[3]

Xwt <- coef(model)[4]

print(Xdisp)

print(Xhp)

print(Xwt)

### Create Equation for Regression Model

Based on the above intercept and coefficient values, we create the mathematical equation.

Y = a+Xdisp.x1+Xhp.x2+Xwt.x3

or

Y = 37.15+(-0.000937)\*x1+(-0.0311)\*x2+(-3.8008)\*x3

Y = 37.15+(-0.000937)\*221+(-0.0311)\*102+(-3.8008)\*2.91 = 22.7104

# R - Logistic Regression

The Logistic Regression is a regression model in which the response variable (dependent variable) has categorical values such as True/False or 0/1. It actually measures the probability of a binary response as the value of response variable based on the mathematical equation relating it with the predictor variables.

The general mathematical equation for logistic regression is −

**y = 1/(1+e^-(a+b1x1+b2x2+b3x3+...))**

Following is the description of the parameters used −

* y is the response variable.
* x is the predictor variable.
* a and b are the coefficients which are numeric constants.

The function used to create the regression model is the glm() function.

### Syntax

The basic syntax for glm () function in logistic regression is −

**glm(formula,data,family)**

### Example

The in-built data set "mtcars" describes different models of a car with their various engine specifications. In "mtcars" data set, the transmission mode (automatic or manual) is described by the column am which is a binary value (0 or 1). We can create a logistic regression model between the columns "am" and 3 other columns - hp, wt and cyl.

**# Select some columns form mtcars.**

input <- mtcars[,c("am","cyl","hp","wt")]

print(head(input))

## Create Regression Model

input <- mtcars[,c("am","cyl","hp","wt")]

am.data = glm(formula = am ~ cyl + hp + wt, data = input, family = binomial)

print(summary(am.data))

# R - Poisson Regression

Poisson Regression involves regression models in which the response variable is in the form of counts and not fractional numbers. For example, the count of number of births or number of wins in a football match series. Also the values of the response variables follow a Poisson distribution.

The general mathematical equation for Poisson regression is −

log(y) = a + b1x1 + b2x2 + bnxn.....

Following is the description of the parameters used −

* y is the response variable.
* a and b are the numeric coefficients.
* x is the predictor variable.

The function used to create the Poisson regression model is the glm() function.

### Syntax

The basic syntax for glm() function in Poisson regression is −

**glm(formula,data,family)**

### Example

We have the in-built data set "warpbreaks" which describes the effect of wool type (A or B) and tension (low, medium or high) on the number of warp breaks per loom. Let's consider "breaks" as the response variable which is a count of number of breaks. The wool "type" and "tension" are taken as predictor variables.

**Input Data**

input <- warpbreaks

print(head(input))

## Create Regression Model

output <-glm(formula = breaks ~ wool+tension, data = warpbreaks,

family = poisson)

print(summary(output))