## What is R Logistic Regression?

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

General mathematical equation for logistic regression is

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

Below 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.

glm() function is used for creating the regression model.

## Syntax

Basic syntax for glm() function in logistic regression is

## glm(formula,data,family)

Below is the description of the parameters used

* formula is the symbol which represents the relationship between the variables.
* data is the data set which gives values of these variables.
* family is R object used for specifying the details of the model. Its value is binomial for logistic regression.

## Example

In-built data set "mtcars" will describe various models of a car with various engine specifications. In "mtcars" data set, transmission mode (automatic or manual) will be described by the column “am” which is a binary value (0 or 1). Logistic regression model can be created 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))

## When above code is executed, it produces following result

## am cyl hp wt

## Mazda RX4 1 6 110 2.620

## Mazda RX4 Wag 1 6 110 2.875

## Datsun 710 1 4 93 2.320

## Hornet 4 Drive 0 6 110 3.215

## Hornet Sportabout 0 8 175 3.440

## Valiant 0 6 105 3.460

## Create Regression Model

glm() function is used for creating the regression model and get its summary for analysis.

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

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

## print(summary(am.data))

## When above code is executed, it produces following result

## Call:

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

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -2.17272 -0.14907 -0.01464 0.14116 1.27641

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) 19.70288 8.11637 2.428 0.0152 \*

## cyl 0.48760 1.07162 0.455 0.6491

## hp 0.03259 0.01886 1.728 0.0840 .

## wt -9.14947 4.15332 -2.203 0.0276 \*

## ---

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

## (Dispersion parameter for binomial family taken to be 1)

## Null deviance: 43.2297 on 31 degrees of freedom

## Residual deviance: 9.8415 on 28 degrees of freedom

## AIC: 17.841

## Number of Fisher Scoring iterations: 8

## Conclusion

As p-value in the last column is more than 0.05 for the variables "cyl" and "hp", these will be considered to be insignificant in contributing to the value of the variable "am". Only weight (wt) impacts the "am" value in this regression model.

## What is normal distribution in R?

Distribution of data is normal in a random collection of data. This means, on plotting a graph with the value of the variable in the horizontal axis and the count of the values in the vertical axis, we get a bell shape curve. Centre of the curve represents mean of the data set and in the graph, fifty percent of values lie to the left of the mean and the other fifty percent lie to the right of the graph. This is referred as normal distribution in statistics.

R has four in built functions for generating normal distribution. They are described as below.

dnorm(x, mean, sd)

pnorm(x, mean, sd)

qnorm(p, mean, sd)

rnorm(n, mean, sd)

Below is the description of the parameters used in above functions

* x is a vector of numbers.
* p is a vector of probabilities.
* n is number of observations(sample size).
* mean is the mean value of the sample data and its default value is zero.
* sd is the standard deviation and its default value is 1.

## dnorm()

dnorm() function gives height of the probability distribution at each point for a given mean and standard deviation.

# Create a sequence of numbers between -10 and 10 incrementing by 0.1.

x <- seq(-10, 10, by = .1)

# Choose the mean as 2.5 and standard deviation as 0.5.

y <- dnorm(x, mean = 2.5, sd = 0.5)

# Give the chart file a name.

png(file = "dnorm.png")

plot(x,y)

# Save the file.

dev.off()

## pnorm()

pnorm() function will give you the probability of a normally distributed random number to be less that the value of a given number. It is also called "Cumulative Distribution Function".

# Create a sequence of numbers between -10 and 10 incrementing by 0.2.

x <- seq(-10,10,by = .2)

# Choose the mean as 2.5 and standard deviation as 2.

y <- pnorm(x, mean = 2.5, sd = 2)

# Give the chart file a name.

png(file = "pnorm.png")

# Plot the graph.

plot(x,y)

# Save the file.

dev.off()

## qnorm()

qnorm() function takes the probability value and gives a number whose cumulative value matches the probability value.

# Create a sequence of probability values incrementing by 0.02.

x <- seq(0, 1, by = 0.02)

# Choose the mean as 2 and standard deviation as 3.

y <- qnorm(x, mean = 2, sd = 1)

# Give the chart file a name.

png(file = "qnorm.png")

# Plot the graph.

plot(x,y)

# Save the file.

dev.off()

## rnorm()

rnorm() function is used for generating random numbers whose distribution is normal. It will take a sample size as an input and will generate those many random numbers. A histogram is drawn to show the distribution of the generated numbers.

# Create a sample of 50 numbers which are normally distributed.

y <- rnorm(50)

# Give the chart file a name.

png(file = "rnorm.png")

# Plot the histogram for this sample.

hist(y, main = "Normal DIstribution")

# Save the file.

dev.off()

What is R Binomial distribution?

Binomial distribution model in R deals with discovering the possibility of success of an event which has only two possible outcomes in a series of experiments. For example, tossing a coin gives has only 2 possibilities either head or a tail. Probability of getting exactly 3 heads while tossing a coin repeatedly for 10 times can be expected during binomial distribution.

R has four in-built functions to generate binomial distribution and they are described below.

dbinom(x, size, prob)

pbinom(x, size, prob)

qbinom(p, size, prob)

rbinom(n, size, prob)

Below is the description of the parameters used

* x is a vector of numbers.
* p is a vector of probabilities.
* n is number of observations.
* size is the number of trials.
* prob is the probability of success of each trial.

## dbinom()

dbinom() function gives the probability density distribution at each point.

# Create a sample of 50 numbers which are incremented by 1.

x <- seq(0,50,by = 1)

# Create the binomial distribution.

y <- dbinom(x,50,0.5)

# Give the chart file a name.

png(file = "dbinom.png")

# Plot the graph for this sample.

plot(x,y)

# Save the file.

dev.off()

## pbinom()

pbinom() function gives the cumulative probability of an event and is a single value representing the probability.

# Probability of getting 26 or less heads from a 51 tosses of a coin.

x <- pbinom(26,51,0.5)

print(x)

## qbinom()

qbinom() function takes the probability value and provides a number whose cumulative value matches with the probability value.

# How many heads will have a probability of 0.25 will come out when a coin is tossed 51 times.

x <- qbinom(0.25,51,1/2)

print(x)

## rbinom()

rbinom() function produces necessary number of random values of given probability from a given sample.

# Find 8 random values from a sample of 150 with probability of 0.4.

x <- rbinom(8,150,.4)

print(x)