# Practical - 7

**Aim:-**

Demonstrate & calculate Binomial Distribution, Poisson Distribution and Normal Distribution for given values.

**Theory:-**

**Binomial Distribution in R**

Binomial distribution in R is a probability distribution used in statistics. The binomial distribution is a discrete distribution and has only two outcomes i.e. success or failure. All its trials are independent, the probability of success remains the same and the previous outcome does not affect the next outcome. The outcomes from different trials are independent. Binomial distribution helps us to find the individual probabilities as well as cumulative probabilities over a certain range.

**Formula:-**

$P(X=k) = {}^{n}C_r \, p^r q^{n-r}$, where $r = 0, 1, 2, 3, \ldots n$

$p$ is the probability of success

$q$ is the probability of failure

$p + q = 1$

**Functions for Binomial Distribution**

**dbinom() Function**

This function is used to find probability at a particular value for a data that follows binomial distribution i.e. it finds:

$P(X = k)$

**Syntax:-**

```
dbinom(k, n, p)
```

Page No.
Date

- **pbinom() Function**

The function pbinom() is used to find the cumulative probability of a data following binomial distribution till a given value i.e it finds $P(X <= K)$

Syntax:—

pbinom(K, n, p)

where n is total number of trials, p is probability of success, K is the value at which the probability has to be found out.

- **qbinom() Function**

This function is used to find the nth quantile, that is if $P(X <= K)$ is given, it finds K.

Syntax:—

qbinom(P, n, p)

where P is the probability, n is the total number of trials and p is the probability of success.

- **rbinom() Function**

This function generates n random variables of a particular probability.

Syntax:—

rbinom(n, N, p)

where n is numbers of observations, N is the total number of trials, p is the probability of success.

**Poisson Distribution in R**

It denote a poisson process as a random experiment that consist on observe the occurrence of specific events over a continuous support (generally the space or the time), such that the process is stable (the number of occurrences, $\lambda$ is constant

in the long run) and the events occur randomly and independently. The Poisson distribution is used to model the number of events that occur in a poisson process. Let $X \sim P(\lambda)$, this is, a random variable with Poisson distribution where the mean number of events that occur at a given intervals is $\lambda$.

- The probability mass function (PMF) is $P(X=x) = \frac{e^{-\lambda}\lambda^x}{x!}$ for $x = 0, 1, 2, \ldots$
- The cumulative distribution function (CDF) is $F(x) = \sum_{i=0}^{x} \frac{e^{-\lambda}\lambda^i}{i!}$
- The quantile function is $Q(p) = F^{-1}(p)$.
- The expected mean and variance of X are $E(X) = Var(X) = \lambda$

## Functions for poisson distribution

### The dpois function

The poisson probability function with mean $\lambda$ can be calculated with the R dpois function for any value of x. The following block of code summarizes the arguments of the function:-

Syntax:-

```
dpois( x,          # x-axis values (x = 0, 1, 2, ...)
       lambda,     # Mean number of events that occur on the interval.
       log= FALSE) # If TRUE, probabilities are given as log
```

### The ppois function

The probability of a variable X following a Poisson distribution taking values equal or lower than x can be calculated with the ppois function, which arguments are described below:-

Syntax:-

```
ppois( q,          # Quantile or vector of quantities
       lambda,     # Mean or vector of means
       lower.tail= TRUE # If TRUE, probabilities are P(X<=x), or P(X>x)
       log.p = FALSE) # If TRUE, probabilities are given as log.
```

- **The qpois function**

The R qpois function allows obtaining the corresponding Poisson quantiles for a set of probabilities.

Syntax:-

```
qpois(p,          # probability or vector of probabilities
      lambda,     # Mean or vector of means
      lower.tail=TRUE, # If TRUE, probabilities are P(X<=x) or P(X>x)
      log.p=FALSE) # If TRUE, probabilities are given as log.
```

- **The rpois function**

If you want to draw n observations from a poisson distribution you can make use of the rpois function.

Syntax:-

```
rpois(n,      # Number of random observations to be generated
      lambda) # Mean or vector of means
```

**Normal Distribution in R**

Normal Distribution is a probability function used in statistics that tells about how the data values are distributed. It is the most important probability distribution function used in statistics because of its advantages in real case scenarios.

In R, there a 4 built-in functions to generate normal distribution:

- **dnorm()**

```
dnorm(x, mean, sd)
```

- **pnorm()**

```
pnorm(x, mean, sd)
```

- **qnorm()**

```
qnorm(p, mean, sd)
```

Page No.
Date

## rnorm()

rnorm(n, mean, sd)

where,

- x represents the data set of values
- mean(x) represents the mean of data set x. It's default value is 0.

$$= \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

- sd(x) represents the standard deviation of data set x. It's default value is 1.

$$= \sqrt{\frac{\sum_{i=1}^{n} (x_i - mean)^2}{n}}$$

- n is the number of observation
- p is vector of probabilities.

## Functions to Generate Normal Distribution in R

### dnorm()

dnorm() function in R programming measures density function of distribution. In statistics, it is measured by below formula -

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2}$$

where, $\mu$ is mean and $\sigma$ is standard deviation.

Syntax:-

dnorm(x, mean, sd)

### pnorm()

pnorm() function is the cumulative distribution function which measures the probability that a random number X takes a value less than or equal to x i.e, in statistics it is given by -

$$F_X(x) = P_r[X \leq x] = \alpha$$

Page No.
Date

## qnorm()

qnorm() function is the inverse of pnorm() function. It take the probability value and gives output which corresponds to the probability value. It is useful in finding the percentiles of a normal distribution.

Syntax:-

```
qnorm(p, mean, sd)
```

## rnorm()

rnorm() function in R programming is used to generate a vector of random numbers which are normally distributed.

Syntax:-

```
rnorm(x, mean, sd)
```

Page No.
Date

# Practical-8

## Aim:-

Write R script to build Linear Regression Model using given dataset.

## Theory:-

### Linear Regression in R

Linear regression is a regression model that uses a straight line to describe the relationship between variables. It finds the line of best fit through your data by searching for the value of the regression coefficient (s) that minimizes the total error of the model.

There are two main types of linear regression:-

- Simple linear regression uses only one independent variable
- Multiple linear regression uses two or more independent variables.

### Getting started in R

Start by downloading R and RStudio. Then open RStudio and click on File > New File > R script.

To install the packages you need for the analysis, run this code

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("broom")
install.packages("ggubr")
```

Next, load the packages into your R environment by running this code

```
library(ggplot2)
library(dplyr)
library(broom)
library(ggubr)
```

Page No.
Date

**Step 1:- Load the data into R**

Follow these four steps for each dataset:-

1. In R studion, go to File > Import dataset > From Text (base).
2. Choose the data file you have downloaded (income.data or heart.data) and an import Dataset window pops up.
3. In the Data Frame window, you should see an X (index) column and columns listing the data for each of the variables (income and happiness or biking, and heart.disease
4. Click on the import button and the file should appear in your Environment tab on the upper right side of the RStudio screen.

**Step 2:- Make sure your data meet the assumptions**

We can use R to check that our data meet the four main assumptions for linear regression.

Simple regression

1. Independence of observations

Because we only have one independent variable and one dependent variable, we don't need to test for any hidden relationship among variables.

2. Normality

To check whether the dependent variable follows a normal distribution, use the hist() function

```
hist(income.data$happiness)
```

3. Linearity

The relationship between the independent and dependent variable must be linear. We can test this visually with a scatter plot to see if the distribution of data points could be described with a straight line.

```
plot(happiness ~ income, data = income.data)
```

4. Homoscedasticity

This means that the prediction error doesn't change significantly over the range of prediction of the model.

Multiple regression

1. Independence of observations

Use the cor() function to test the relationship between your independent variables and make sure they aren't too highly correlated.

```
cor(heart.data$biking, heart.data$smoking)
```

2. Normality

Use the hist() function to test whether your dependent variable follows a normal distribution.

```
hist(heart.data$heart.disease)
```

3. Linearity

We can check this using two scatterplots one for biking and heart disease, and one for smoking and heart disease.

```
plot(heart.disease ~ biking, data = heart.data)
plot(heart.disease ~ smoking, data = heart.data)
```

Step3:- Perform the linear regression analysis

Step4:- Check for homoscedasticity

Step5:- Visualize the results with a graph.

# Practical-9

**Aim:-**
Perform Apriori Analysis using arules package.

**Theory:-**

**Apriori Algorithm in R**

Apriori Algorithm is used for finding frequent itemsets in a dataset for association rule mining. It is called Apriori because it uses prior knowledge of frequent itemset properties. we apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.
To improve the efficiency of the level-wise generation of frequent itemsets an important property is called Apriori property which helps by reducing the search space.

**Important Terminologies**

- **Support:-** Support is an indication of how frequently the itemset appears in the dataset. It is the count of records containing an item 'x' divided by the total number of records in the database.

- **Confidence:-** Confidence is a measure of times such that if an item 'x' is bought, then item 'y' is also bought together. It is the support count $(x \cup y)$ divided by the support count of 'x'.

- **Lift:-** Lift is the ratio of the observed support to that which is expected if 'x' and 'y' were independent. It is the support count of $(x \cup y)$ divided by the product of individual counts of 'x' and 'y'.

Page No.
Date

## Algorithm

1. Read each item in the transaction.
2. Calculate the support of every item.
3. If support is less than minimum sport, discard the item. Else, insert it into frequent itemset.
4. Calculate confidence for each non-empty subset.
5. If confidence is less than minimum confidence, discard the subset. Else, it into strong rules.

## Apriori Algorithm Implementation in R

Rstudio provides popular open source and enterprise-ready professional software for the R statistical computing environment. R is a language that is developed to support statistical calculations and graphical computing / visualizations. It has an in-built library function called arules which implements the Apriori algorithm for Market Basket Analysis and computes the strong rules through association rule mining, once we specify the minimum support and minimum confidence, according to our needs.

**Step 1:-** Load required library

→ 'arules' package provides the infrastructure for representing, manipulating, and analyzing transaction data and patterns

```
library(arules)
```

→ 'arulesViz' package is used for visualizing Association Rules and frequent itemsets. It extends the package 'arules' with various visualization techniques for association rules and itemsets. The package also includes several interactive visualizations for rule exploration.

```
library(arulesViz)
```

→ 'RColorBrewer' is a ColorBrewer palette which provides color schemes for maps and other graphics.

```
library(RColorBrewer)
```

Step 2:- Import the dataset

'Groceries' dataset is predefined in the R package. It is a set of 9835 records / transactions, each having 'n' number of items, which were bought together from the grocery store data ("Groceries")

Step 3:- Applying the apriori() function

'apriori()' function is in-built in R to mine frequent itemsets and association rules using the Apriori algorithm. Here, 'Groceries' is the transaction data. 'parameter' is a named list that specifies the minimum support and confidence for finding the association rules. The default behavior is to mine the rules with minimum support of 0.1 and 0.8 as the minimum confidence. Here, we have specified the minimum support to be 0.01 and the minimum confidence to be 0.2.

```
rules <- apriori(Groceries, parameter = list(supp=0.01, conf=0.2))
```

Step 4:- Applying inspect() function

inspect() function prints the internal representation of an R object or the result of an expression.

Here, it displays the first 10 strong association rules.

```
inspect(rules[1:10])
```

Step 5:- Applying itemFrequencyPlot() function

ItemFrequencyPlot() creates a bar plot for item frequencies / support. It creates an item frequency bar plot for inspecting the distribution of objects based on the transactions. The items are plotted ordered by descending support. Here, 'topN = 20' means that 20 items with the highest item frequency / left will be plotted.

```
arules:: itemFrequencyPlot(Groceries, topN = 20,
```

```
col = brewer.pal (8, 'Pastel2'),
main = 'Relative Item Frequency Plot',
type = "relative",
ylab = "Item Frequency (Relative)")
```

# Practical-10

**Aim:-**

Case Study: Implement K-Means Algorithm on Worlddata dataset and visualize the clusters.

**Theory:-**

**K-Means Clustering in R**

K Means clustering in R programming is an Unsupervised Non-Linear algorithm that cluster data based on similarity or similar groups. It seeks to partition the observations into a pre-specified number of clusters. Segmentation of data takes place to assign each training example to a segment called a cluster.

In the unsupervised algorithm, high reliance on raw data is given with large expenditure or manual review for review of relevance is given. It is used in a variety of fields like Banking, healthcare, retail, Media, etc.

**Theory:-**

K-Means clustering groups the data on similar groups. The algorithm is as follows:-

1. Choose the number K clusters
2. Select at random K points, the centroids (Not necessarily from the given data).
3. Assign each data point to closest centroid that forms K clusters.
4. Compute and place the new centroid of each centroid.
5. Reassign each data point to new cluster.

After final reassignment, name the cluster as final cluster.

Page No.
Date

## The Dataset

Iris dataset consists of 50 samples from each of 3 species of Iris (Iris setosa, Iris virginica, Iris versicolor) and a multivariate dataset introduced by British statistician and biologist Ronald fisher in his 1936 paper The use of multiple measurements in taxonomic problems. Four features were measured from each sample i.e. length and width of the sepals and petals and based on the combination of these four features, fisher developed a linear discriminant model to distinguish the species from each other.

```
# Loading data
data(iris)
# structure
str(iris)
```