**nit: III**

Statistics and Probability, basic data visualization, probability, common probability distributions: common probability mass functions, bernoulli, binomial, poisson distributions, common probability density functions, uniform, normal, student's t- distribution.

**basic data visualization:**

- ✓ Data visualization is the graphical representation of data, making it easier to understand, interpret, and communicate information. It's a powerful tool that can transform raw data into meaningful insights.
- ✓ By using the data visualization technique, we can work with large datasets to efficiently obtain key insights about it.
- ✓ Graphics play an important role in carrying out the important features of the data.

**Common Types of Data Visualization**

- ✓ **Bar Charts:** Suitable for comparing discrete values.
- ✓ **Line Charts:** Ideal for showing trends over time.
- ✓ **Pie Charts:** Useful for representing parts of a whole.
- ✓ **Scatter Plots:** Effective for visualizing relationships between two variables.
- ✓ **Histograms:** Used to show the distribution of a single numerical variable.
- ✓ **Box plot:** The box plot is a data visualization tool that provides a concise overview of data distribution, from central tendencies to potential outliers

**Bar Charts:**

- ✓ A bar chart uses rectangular bars to visualize data.
- ✓ The height or length of the bars are proportional to the values they represent.
- ✓ A bar chart is used for summarizing a set of categorical data.

**Types of Bar Charts**

- ✓ **Vertical Bar Charts:** The most common type, where the bars are oriented vertically.
- ✓ **Horizontal Bar Charts:** The bars are oriented horizontally, often used when categories have long names.

- ✓ **Stacked Bar Charts:** Multiple categories are stacked on top of each other within each bar, showing the total and individual contributions.
- ✓ **Grouped Bar Charts:** Categories are grouped together, and bars within each group are compared.

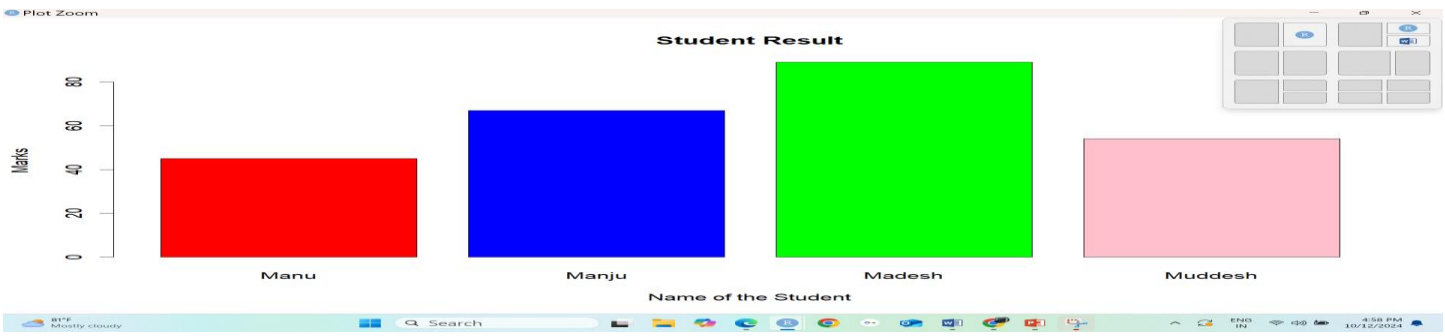
# syntax:

barplot (h, x, y, main, names.arg, col)

| S.No | Parameter | Description |
|---|---|---|
| 1. | H | A vector or matrix which contains numeric values used in the bar chart. |
| 2. | xlab | A label for the x-axis. |
| 3. | ylab | A label for the y-axis. |
| 4. | main | A title of the barchart. |
| 5. | names.arg | A vector of names that appear under each bar. |
| 6. | col | It is used to give colors to the bars in the graph. |

Example:

```
a <- c(45,67,89,54)
name<-c("Manu","Manju","Madesh","Muddesh")
r<-c("red","blue","green","pink")
barplot(A,names.arg = name, xlab = "Name of the Student", ylab = "Marks", main ="Student Result",col=r)
```

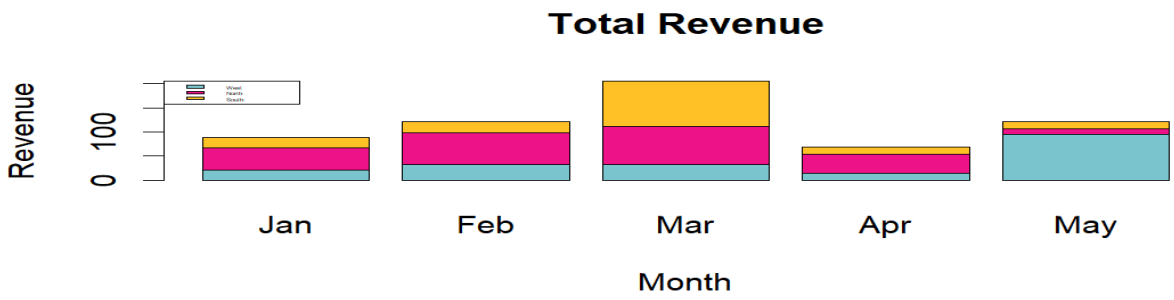


**Group Bar Chart & Stacked Bar Chart**

- ✓ We can create bar charts with groups of bars and stacks using matrices as input values in each bar.
- ✓ One or more variables are represented as a matrix that is used to construct group bar charts and stacked bar charts.

Example

```
months<- c("Jan","Feb","Mar","Apr","May")
regions <- c("West","North","South")
Values <- matrix(c(21,32,33,14,95,46,67,78,39,11,22,23,94,15,16), nrow = 3,ncol = 5, byrow = TRUE)
barplot(Values, main = "Total Revenue", names.arg = months, xlab = "Month", ylab = "Revenue", col =c("cadetblue3","deeppink2","goldenrod1"))
legend("topleft",regions,cex =0.2,fill=c("cadetblue3","deeppink2","goldenrod1"))
```




**R Pie Charts**

- ✓ A pie chart is a circular graphical view of data.
- ✓ A pie-chart is a representation of values in the form of slices of a circle with different colors.
- ✓ Slices are labeled with a description, and the numbers corresponding to each slice also shown in the chart.
- ✓ The Pie charts are created with the help **of pie () function**, which takes positive numbers as vector input
- ✓ By default, the plotting of the first pie starts from the x-axis and move counterclockwise
- ✓ Note: The size of each pie is determined by comparing the value with all the other values, by using this formula: The value divided by the sum of all values: x/sum(x)

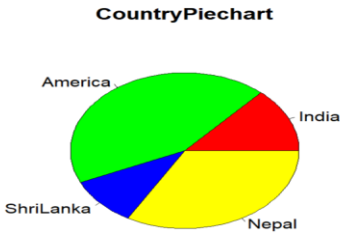
**Syntax:**

pie(X, Labels, Radius, Main, Col, Clockwise)

| Parameter | Description |
|---|---|
| X | is a vector that contains the numeric values used in thepie chart. |
| Labels | are used to give the description to the slices. |
| Radius | describes the radius of the pie chart. |
| Main | describes the title of the chart. |
| Col | defines the colour palette. |
| Clockwise | is a logical value that indicates the clockwise or anti-clockwise direction in which slices are drawn. |

Example

```
x <- c(20, 65, 15, 50)
labels<-c("India","America","ShriLanka","Nepal")
pie(x,labels,main="CountryPiechart",col=c("red","green","blue","yellow"))
```




**Slice Percentage & Chart Legend**

- ✓ There are two additional properties of the pie chart, i.e., slice percentage and chart legend. We can show the data in the form of percentage as well as we can add legends to plots in R by using the legend() function.

Example

```
x <- c(10,20,30,40)
mylabel <- c("Apples", "Bananas", "Cherries", "Dates")
colors <- c("blue", "yellow", "green", "black")
pie(x, label = mylabel, main = "Pie Chart", col = colors)
legend("bottomright", mylabel, fill = colors)
```


## R Histogram

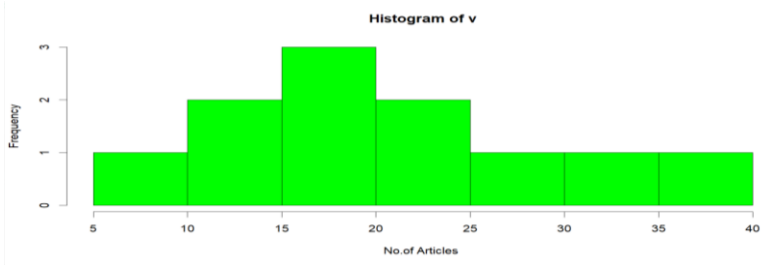
- ✓ A histogram is a type of bar chart which shows the frequency of the number of values which are compared with a set of values ranges.
- ✓ The histogram is used for the distribution, where as a bar chart is used for comparing different entities.
- ✓ In the histogram, each bar represents the height of the number of values present in the given range.
- ✓ For creating a histogram, R provides hist() function, which takes a vector as an input.

Syntax: hist(v,main,xlab,ylab,xlim,ylim,breaks,col,border)

| Parameter | Description |
|---|---|
| v | It is a vector that contains numeric values. |
| main | It indicates the title of the chart. |
| col | It is used to set the color of the bars. |
| border | It is used to set the border color of each bar. |
| xlab | It is used to describe the x-axis. |
| ylab | It is used to describe the y-axis. |
| xlim | It is used to specify the range of values on the x-axis. |
| ylim | It is used to specify the range of values on the y-axis. |
| breaks | It is used to mention the width of each bar. |

Example

```
v <- c(19, 23, 11, 5, 16, 21, 32, 14, 19, 27, 39)
hist(v, xlab = "No.of Articles ",col = "green", border = "black")
```



**R Box Plot**

- ✓ Box plots are a measure of how well data is distributed across a data set. This divides the data set into three quartiles.
- ✓ This graph represents the minimum, maximum, average.
- ✓ Box plot is also useful in comparing the distribution of data in a data set by drawing a box plot for each of them.
- ✓ R provides a boxplot() function to create a boxplot.

**Syntax: boxplot(x, data, notch, varwidth, names, main)**

| **Parameter** | **Description** |
|---|---|
| x | It is a vector or a formula. |
| data | It is the data frame. |
| notch | It is a logical value set as true to draw a notch. |
| varwidth | It is also a logical value set as true to draw the widthof the box same as the sample size. |
| names | It is the group of labels that will be printed under eachboxplot. |
| main | It is used to give a title to the graph. |

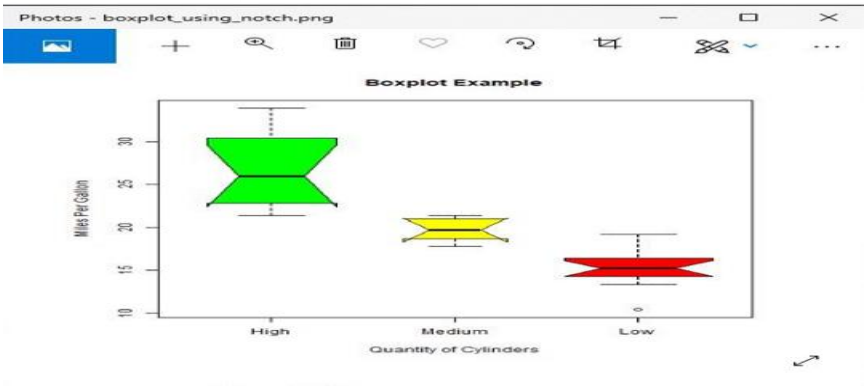
Example

```
Teaching<-c(3,3,7,8,8,10,11,12,15,18)
boxplot (Teaching)
```


**Box plot using notch In R, we can draw a box plot using a notch.**

- ✓ In R, we can draw a box plot using a notch.

```
boxplot(mpg ~ cyl, data = mtcars, xlab="QuantityofCylinders", ylab = "Miles Per Gallon", main = "Boxplot Example",
notch = TRUE, varwidth=TRUE, ccol= c("green","yellow","red"), names=c("High","Medium","Low"))
```




**R Scatter plots**

- A "scatter plot" is a type of plot used to display the relationship between two numerical variables, and plots one dot for each observation.
- It needs two vectors of same length, one for the x-axis (horizontal) and one for the y-axis (vertical):

**Syntax:**

plot(x, y, main, xlab, ylab, xlim, ylim, axes)

| **Parameters** | **Description** |
|---|---|
| x | It is the dataset whose values are the horizontalcoordinates. |
| y | It is the dataset whose values are the verticalcoordinates. |
| main | It is the title of the graph. |
| xlab | It is the label on the horizontal axis. |
| ylab | It is the label on the vertical axis. |
| xlim | It is the limits of the x values which is used for plotting. |
| ylim | It is the limits of the values of y, which is used for plotting. |
| axes | It indicates whether both axes should be drawn on the plot. |

Example

x1 <- c(5,7,8,7,2,2,9,4,11,12,9,6)

y1 <- c(99,86,87,88,111,103,87,94,78,77,85,86)

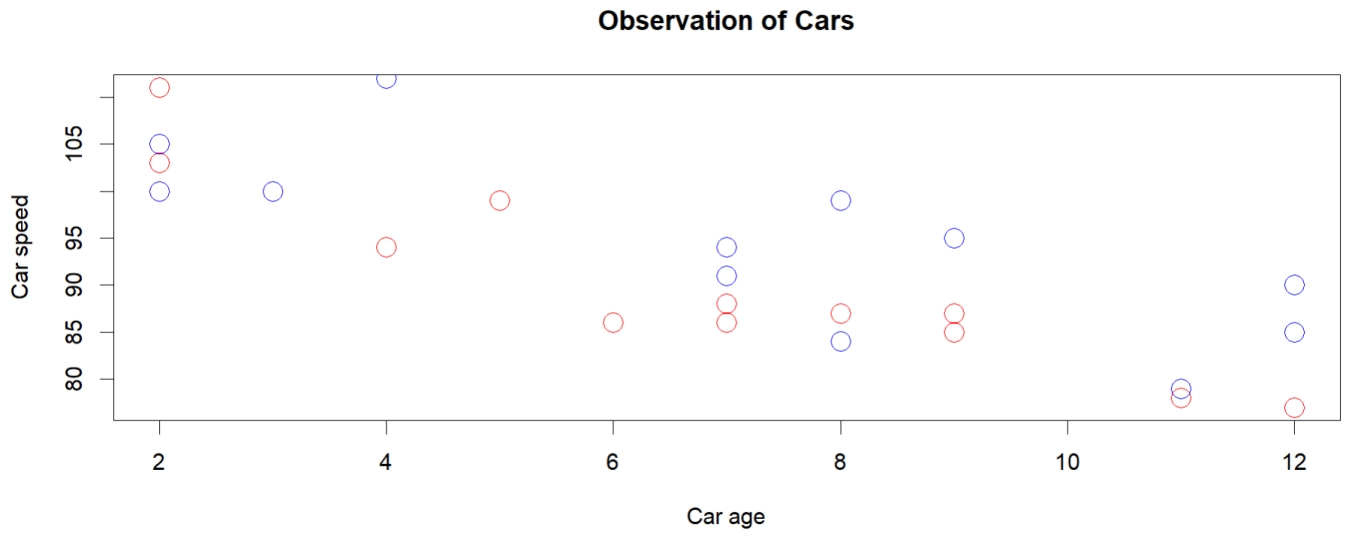
# day two, the age and speed of 15 cars:

x2 <- c(2,2,8,1,15,8,12,9,7,3,11,4,7,14,12)

y2 <- c(100,105,84,105,90,99,90,95,94,100,79,112,91,80,85)

plot(x1, y1, main="Observation of Cars", xlab="Car age", ylab="Car speed", col="red", cex=2)

points(x2, y2, col="blue", cex=2)




**Statistics**

Statistics is a form of mathematical analysis that concerns the collection, organization, analysis, interpretation, and presentation of data.

**R– Statistics**

R is a programming language and is used for environment statistical computing and graphics.

**Average in R Programming**:

Average is calculated by dividing the sum of the values in the set by their number.

$$A = (x_1 + x_2 ..... + x_n)/n$$

**Mean**

- ✓ The average of a set of numbers, calculated by adding all the numbers together and dividing by the total number of numbers. It's also known as the arithmetic mean.

$$\text{Mean } (\bar{x}) = \frac{\text{Sum of Values}}{\text{Number of Values}}$$

Example

The dataset `study_hours` contains the values: 2, 4, 5, 3, and 6.

The **mean** is the sum of these values divided by the number of values. In this case:

$$\text{Mean} = \frac{(2+4+5+3+6)}{5} = \frac{20}{5} = 4$$

## Median

- ✓ A Median is a middle value for sorted data. The sorting of the data can be done either in ascending order or descending order.
- ✓ A median divide the data into two equal halves.

Example

**Case with an Odd Number of Elements**

The dataset `ages` contains the values: 21, 34, 19, 42, 27, 29, and 36.

To find the **median**, you first need to order the data from smallest to largest:

$$\text{Ordered Data} = 19, 21, 27, 29, 34, 36, 42$$

The **median** is the middle value in the ordered list. In this case, the middle value is 29, which is the fourth value.

the **median** age is **29**.

**Case with an Even Number of Elements**

**Example**

The ordered dataset would be: 19, 21, 27, 29, 34, 36, 40, 42.

Since there is no single middle value (the dataset has an even number of elements), the median is the average of the two middle values:

$$\text{Median} = \frac{(29+34)}{2} = 31.5$$

## Mode

- ✓ The **mode** is a statistical term that refers to the **value that occurs most frequently** in a dataset. It represents the data point that appears with the highest frequency.

## Example

The dataset is: `3, 5, 2, 5, 4, 5, 6`.

The number **5** occurs **3 times**, which is more frequent than any other number in the list.

Therefore, the **mode** is **5**.

## Standard Deviation

- ✓ The **standard deviation** is a measure of how spread out the values in a dataset are around the mean.
- ✓ A low standard deviation means that the values are close to the mean, while a high standard deviation means the values are more spread out.

### Formula for Standard Deviation:

For a sample, the standard deviation is calculated as:

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

Where:

- $x_i$ is each individual value,
- $\bar{x}$ is the mean of the dataset,
- $n$ is the number of values in the dataset.

For a **population**, the formula is similar but divided by $n$ instead of $n - 1$.

## Variance

- ✓ **Variance** is a measure of how much the values in a dataset differ from the mean. It tells you how far the values in the dataset are spread out.
- ✓ A high variance means that the numbers are more spread out from the mean, while a low variance means that the numbers are closer to the mean.

**Formula for Variance:**

- **Sample Variance** (when you are working with a sample of the population):

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Where:

- $x_i$ is each individual value,
- $\bar{x}$ is the mean of the dataset,
- $n$ is the number of values in the dataset.

- **Population Variance** (when you are working with the entire population, you divide by $n$ instead of $n-1$):

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

Where $\mu$ is the population mean.

# PROBABILITY:

- ✓ **Probability**: is a measure of how likely an event is to occur. It is expressed as a number between 0 and 1, where:**0** means the event is impossible. **1** means the event is certain.

**For example:**

- ✓ The probability of flipping a coin and getting heads is 0.5 (or 50%), as there are two equally likely outcomes (heads or tails).
- ✓ The probability of drawing a red card from a standard deck of 52 cards is 26/52, or 0.5 (or 50%), as there are 26 red cards and 52 cards total.

**• Basic Definitions:**

- ✓ **Experiment**: A procedure that yields one of a possible set of outcomes (e.g., rolling a die).
- ✓ **Sample Space (S)**: The set of all possible outcomes of an experiment (e.g., rolling a die: {1, 2, 3, 4, 5, 6}).
- ✓ **Event (E)**: A subset of the sample space. An event can include one outcome or multiple outcomes (e.g., rolling an even number: {2, 4, 6}).

- The probability of an event $E$ is given by:

$$P(E) = \frac{\text{Number of favorable outcomes}}{\text{Total number of outcomes in the sample space}}$$

- The probability value ranges from 0 to 1, where 0 indicates impossibility and 1 indicates certainty.

**Random Variable**

- ✓ A random variable is a numerical quantity whose value depends on the outcome of a random experiment. In simpler terms, it's a variable that can take on different values based on chance.

**OR**

- ✓ A random variable assigns a numerical value to each possible outcome of a random experiment

**OR**

- ✓ A **random variable** is a variable that takes on different values based on the outcome of a random event or process. The values that a random variable can take are determined by chance, and the probability of each possible value is often known or can be computed.
- ✓ For example, let us consider an experiment for tossing a coin two times.
- ✓ Hence, the sample space for this experiment is S = {HH, HT, TH, TT}
- ✓ If X is a random variable and it denotes the number of heads obtained, then the values are represented as follows: Getting 2 heads x ={0,1,2}

X(HH) = 2, X(HT) = 1, X(TH) = 1, X(TT) = 0.

| X | 0 | 1 | 2 |
|---|---|---|---|
| P(X=x) | 1/4 | 2/4 | 1/4 |

Similarly, we can define the number of tails obtained using another variable, say Y.

(i.e) Y(HH) = 0, Y(HT) = 1, Y(TH) = 1, Y(TT)= 2.

**Types of Random Variables:**

1. **Discrete Random Variables:**

   - ✓ These can only take on a countable number of values.
   - ✓ For example, the number of heads obtained after flipping a coin three times is a discrete random variable. The possible values of this variable are 0, 1, 2, or 3

2. **Continuous Random Variables:**
   - ✓ These can take on any value within a specified range.
- ✓ Takes an infinite number of possible values within a given range. For example, the height of a person or the time taken to complete a task.

- ✓ Example
  - ✓ For instance, the height of a person is a continuous random variable.
  - ✓ The speed of a vehicle on a highway.

**Difference between Continuous Random Variable and Discrete Random Variable**

| Continuous Random Variable | Discrete Random Variable |
|---|---|
| The value of a continuous random variable falls between a range of values. | The value of a discrete random variable is an exact value. |
| The probability density function is associated with a continuous random variable. | The probability mass function is used to describe a discrete random variable |
| A continuous random variable can take on an infinite number of values. | Such a variable can take on a finite number of distinct values. |
| Mean of a continuous random variable is $E[X] = \int_{-\infty}^{\infty} x f(x) dx$ | The mean of a discrete random variable is $E[X] = \sum x P(X = x)$, where $P(X = x)$ is the probability mass function. |
| The variance of a continuous random variable is $Var(X) = \int_{-\infty}^{\infty} (x-\mu)^2 f(x) dx$ | The variance of a discrete random variable is $Var[X] = \sum (x - \mu)^2 P(X = x)$ |
| The examples of a continuous random variable are uniform random variable, exponential random variable, normal random variable, and standard normal random variable. | The examples of a discrete random variable are binomial random variable, geometric random variable, Bernoulli random variable, and Poisson random variable. |

**Probability Distributions:**

To describe the behavior of a random variable, we use probability distributions. These functions assign probabilities to each possible value or range of values the variable can take.

- **Probability Mass Function (PMF):** Used for discrete random variables. It gives the probability that the variable takes a specific value.
- **Probability Density Function (PDF):** Used for continuous random variables. It gives the probability that the variable falls within a certain range of values.

## Common Probability Distribution

### Common Probability Mass Functions

### Bernoulli Distribution

- ✓ The **Bernoulli distribution** is a discrete probability distribution that describes the outcome of a **single experiment (or trial)** that has exactly two possible outcomes: "**success" (often coded as 1) and "failure" (coded as 0)**.
- ✓ For example, In R it can be represented as a coin toss where the probability of getting the head is 0.5 and getting a tail is 0.5. It is a probability distribution of a random variable that takes value 1 with probability p and the value 0 with probability q=1-p.

The Bernoulli distribution is characterized by a single parameter $p$, which represents the probability of success (i.e., $P(X = 1) = p$).

- ✓ Consequently, the probability of failure is $P(X = 0) = 1 - p$.

**The probability mass function f of this distribution, over possible outcomes k, is given by:**

$$f(k;p) = \begin{cases} p & \text{if } k = 1, \\ q = 1 - p & \text{if } k = 0. \end{cases}$$

The mean and variance are defined as follows, respectively:

$$\mu_X = p \qquad \text{and} \qquad \sigma_X^2 = p(1-p)$$

In R Programming Language, there are 4 built-in functions to for Bernoulli distribution. They are:

**dbern()**

- ✓ dbern() function in R programming measures the mass function of the Bernoulli distribution.
- ✓ Syntax: dbern(x, prob, log=FALSE)
- ✓ Parameter: x: vector of quantiles prob: probability of success on each trial log: logical; if TRUE, probabilities p is given also g(p)

**pbern()**

- ✓ this function is used to find cumulative distribution function (CDF) or cumulative frequency function, describes the probability that a variate X takes on a value less than or equal to a number x

Syntax: pbern(q, prob, lower.tail = TRUE, log.p = FALSE)
Parameter:

- ✓ q: vector of quantiles
- ✓ prob: probability of success on each trial
- ✓ lowe.tail: logical
- ✓ log.p: logical; if TRUE, probabilities p are given as log(p).

**qbern()**

- ✓ this function is used to find A quantile function in statistical terms specifies the value of the random variable such that the probability of the variable being less than or equal to that value equals the given probability.
- ✓ Syntax: qbern(p, prob, lower.tail = TRUE, log.p = FALSE)

Parameter:

- p: vector of probabilities.
- prob: probability of success on each trial.
- lower.tail: logical
- log.p: logical; if TRUE, probabilities p are given as log(p).

**rbern()**

- ✓ rbern( ) function in R programming is used to generate a vector of random numbers which are Bernoulli distributed.
- ✓ Syntax: rbern(n, prob)
- ✓ Parameter:
- ✓ n: number of observations.
- ✓ prob: number of observations.

**Binomial distribution**

- ✓ 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.
- ✓ is a discrete probability distribution that models the number of successes in a fixed number of independent Bernoulli trials, each with the same probability of success.
- ✓ 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

**The probability Mass Function (PMF) is**

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}; \quad x = \{0, 1, \ldots, n\}$$

where

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

The mean and variance are defined as follows:

$$\mu_X = np \quad \text{and} \quad \sigma_X^2 = np(1-p)$$

In R Programming Language, there are 3 built-in functions to for Binomial distribution. They are:

**dbinom () Function**

- ✓ This function is used to find probability mass function at a particular value for a data that follows binomial distribution i.e. it finds: if P (X = k)

**Syntax: dbinom(k, n, p)**

**pbinom() Function**

- ✓ The function **pbinom()** is used to find the cumulative probability of a data following binomial distribution till a given value ie it finds (X <= k)
- ✓ **Syntax: pbinom(k, n, p)**

**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)**

**rbinom() Function**

- ✓ This function generates n random variables of a particular probability.
- ✓ Syntax: rbinom(n, 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.

## poisson distributions

- ✓ The Poisson distribution represents the probability of a provided number of cases happening in a set period of space or time if these cases happen with an identified constant mean rate.

- ✓ In mathematical terms, for a discrete random variable and a realization X=x, the Poisson mass function f is given as follows, where λp is a parameter of the distribution.

$$f(x) = \frac{\lambda_p^x \exp(-\lambda_p)}{x!}; \quad x = \{0, 1, \ldots\}$$

The notation

$$X \sim \text{POIS}(\lambda_p)$$

Where

- x = 0, 1, 2, 3...
- e is the Euler's number (e = 2.718)
- λ is an average rate of the expected value and λ = variance, also λ>0

**There are four Poisson functions available in R:**

**dpois()**

- ✓ This function is used for the illustration of Poisson mass function
- ✓ Syntax: dpois(k,λ,log) dpois(k,λ,log)

where,

- K: number of successful events happened in an interval
- lambda: mean per interval
- log: If TRUE then the function returns probability in form of log

**ppois()**

- ✓ This function is used to find the cumulative probability function.
- ✓ The function ppois() calculates the probability of a random variable that will be equal to or less than a number.
- ✓ **Syntax:** ppois(q,λ,lower.tail,log) ppois(q,λ,lower.tail,log)

**where**

- **K:** number of successful events happened in an interval
- **lambda**: mean per interval
- **lower.tail:** If TRUE then left tail is considered otherwise if the FALSE right tail is considered
- **log:** If TRUE then the function returns probability in form of log

**rpois()**

- ✓ The function rpois() is used for generating random numbers from a given Poisson distribution.
- ✓ **Syntax:** rpois(q,λ) rpois(q,λ)
- ✓ **where, q:** number of random numbers needed **lambda**: mean per interval

**qpois()**

- ✓ The function qpois() is used for generating the quantile of a given Poisson's distribution.

- ✓ **Syntax:** qpois(q,λ,lower.tail,log) qpois(q,λ,lower.tail,log)

**where,**

- ✓ **K:** number of successful events happened in an interval
- ✓ **lambda**: mean per interval
- ✓ **lower.tail:** If TRUE then left tail is considered otherwise if the FALSE right tail is considered
- ✓ **log:** If TRUE then the function returns probability in form of log

**Common Probability Density Functions**

## Uniform Distribution

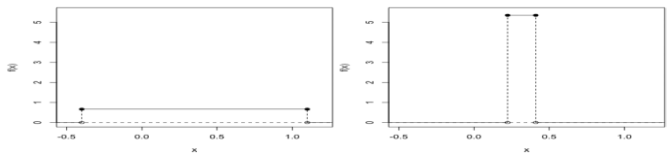
- ✓ The **uniform distribution** is a type of probability distribution in which all outcomes are equally likely within a defined range. It is characterized by its simplicity and is commonly used in statistical modeling and simulations.
- ✓ A uniform distribution holds the same probability for the entire interval. Thus, its plot is a rectangle, and therefore it is often referred to as rectangular distribution.

For a continuous random variable $a \le X \le b$, the uniform density function $f$ is

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \le x \le b; \\ 0 & \text{otherwise} \end{cases} \qquad (16.5)$$

The mean and variance are as follows:

$$\mu_X = \frac{a+b}{2} \quad \text{and} \quad \sigma_X^2 = \frac{(b-a)^2}{12}$$



- ✓ The runif() function in R programming language is used to generate a sequence of random following the uniform distribution.

**Syntax:**runif(n, min = 0, max = 1)
**Parameter:**

- n= number of random samples
- min=minimum value(by default 0)
- max=maximum value(by default 1)

qunif()

- ✓ method is used to calculate quantile for any probability (p) for a given uniform distribution. To use this simply the function had to be called with the required parameters.
- ✓ Syntax:qunif(p, min = 0, max = 1)
- ✓ Parameter: p – The vector of probabilities, min , max – The limits for calculation of quantile function

## dunif()

- ✓ is used to generate density function. It calculates the uniform density function in R language in the specified interval (a, b).
- ✓ Syntax: dunif(x, min = 0, max = 1, log = FALSE)
- ✓ Parameter:
    - x: input sequence
    - min, max= range of values
    - log: indicator, of whether to display the output values as probabilities.

## punif()

- ✓ The punif() method in R is used to calculate the uniform cumulative distribution function, this is, the probability of a variable X taking a value lower than x (that is, x <= X). If we need to compute a value x > X, we can calculate 1 – punif(x).
- ✓ **Syntax:**punif(q, min = 0, max = 1, lower.tail = TRUE)

## Normal Distribution

- ✓ It is generally observed that data distribution is normal when there is a random collection of data from independent sources. The graph produced after plotting the value of the variable on x-axis and count of the value on y-axis is bell-shaped curve graph.
- ✓ The graph signifies that the peak point is the mean of the data set and half of the values of data set lie on the left side of the mean and other half lies on the right part of the mean.

For a continuous random variable $-\infty < X < \infty$, the normal density function $f$ is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \qquad (16.6)$$

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

**dnorm()**

- ✓ function in R programming measures density function of distribution.
- ✓ Syntax: dnorm(x, mean, sd)

**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.
- ✓ Syntax: pnorm(x, mean, sd)

**qnorm()**

- ✓ function is the inverse of pnorm() function.
- ✓ It is useful in finding the percentiles of a normal distribution.
- ✓ Syntax: qnorm(p, mean, sd)

**rnorm()**

- ✓ function in R programming is used to generate a vector of random numbers which are normally distributed.
- ✓ Syntax: rnorm(x, mean, sd)

## Student's t-distribution

- ✓ The Student's t-distribution is a continuous probability distribution generally used when dealing with statistics estimated from a sample of data.
- ✓ Any particular t-distribution looks a lot like the standard normal distribution— it's bell shaped, symmetric and it's centered on zero. T
- ✓ he difference is that while a normal distribution is typically used to deal with a population, the t-distribution deals with sample from a population.

**dt(): Density Function (PDF)**

- ✓ This function calculates the density (height of the curve) of the t-distribution at a given point $x$

  **Syntax: dt(x, df)**

  - x: The value (or vector of values) for which to evaluate the density.
  - df: The degrees of freedom for the t-distribution.

**pt(): Cumulative Distribution Function (CDF)**

- ✓ This function calculates the cumulative probability up to a given value $x$
- ✓ **Syntax: pt(q, df)**
  - `q`: The value (or vector of values) for which to compute the cumulative probability.
  - `df`: The degrees of freedom for the t-distribution.

**qt(): Quantile Function (Inverse CDF)**

- ✓ This function calculates the **quantile** corresponding to a given cumulative probability. It returns the value x for which the cumulative probability is equal to p.
- ✓ **Syntax: qt(p, df)**
  - ✓ `p`: The cumulative probability (a value between 0 and 1).
  - ✓ `df`: The degrees of freedom for the t-distribution.

**`rt()`: Random Number Generation**

- ✓ This function generates random numbers from a t-distribution with specified degrees of freedom.
- ✓ **Syntax: rt(n, df)**
  - `n`: The number of random samples to generate.
  - `df`: The degrees of freedom for the t-distribution