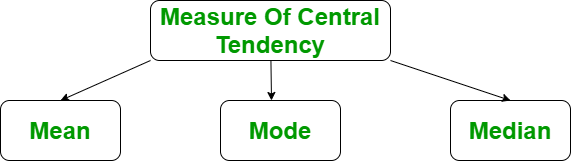
# Probabilities using R

# Mean, Median and Mode in R Programming

Last Updated : 08 Jul, 2024

Statistical measures like mean, median, and mode are essential for summarizing and understanding the central tendency of a dataset. In R, these measures can be calculated easily using built-in functions. This article will provide a comprehensive guide on how to calculate mean, median, and mode in [R Programming Language](https://www.geeksforgeeks.org/r-programming-language-introduction/).



*Mean, Median and Mode in R Programming*

### Dataset used for Calculating the Mean, Median, and Mode in R Programming

Before doing any computation, first of all, we need to prepare our data and save our data in external .txt or .csv files and it’s a best practice to save the file in the current directory. After that import, your data into R as follow:

**

## Mean in R Programming Language

It is the sum of observations divided by the total number of observations. It is also defined as average which is the sum divided by count.

[Mean(μ)=1N∑i=1Nxi][Mean(*μ*)=*N*1​∑*i*=1*N*​*xi*​]

**

## Median in R Programming Language

It is the middle value of the data set. It splits the data into two halves. If the number of elements in the data set is odd then the center element is median and if it is even then the median would be the average of two central elements.

**

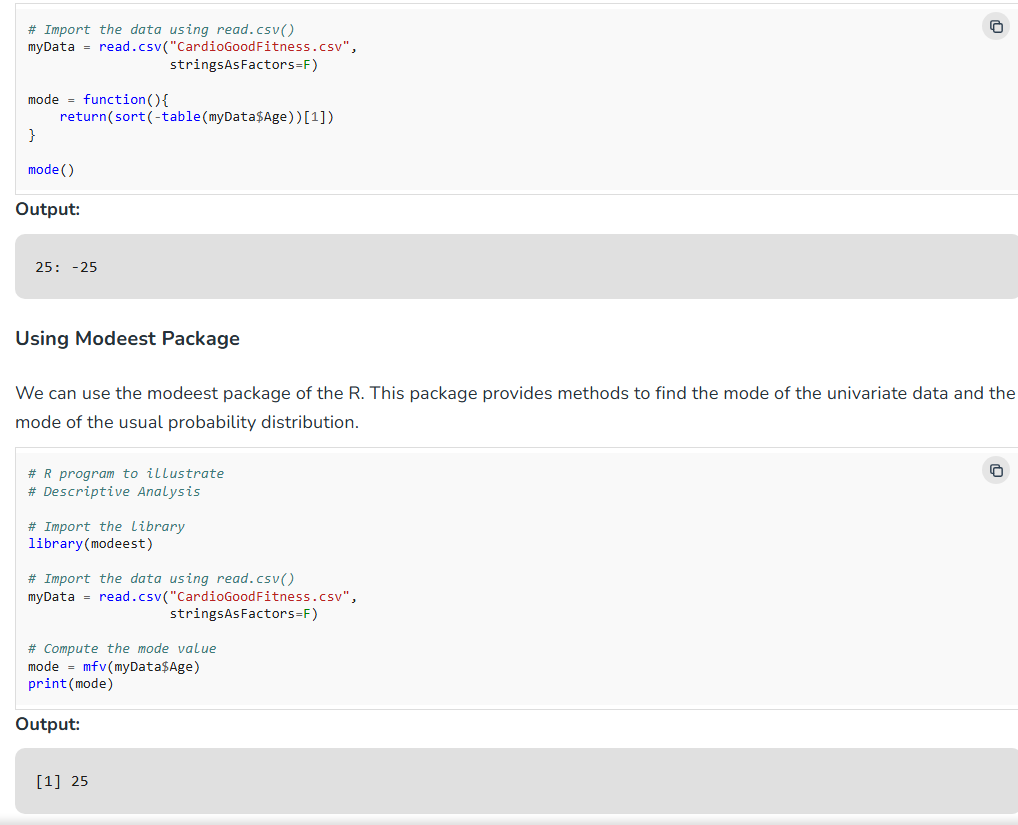
## Mode in R Programming Language

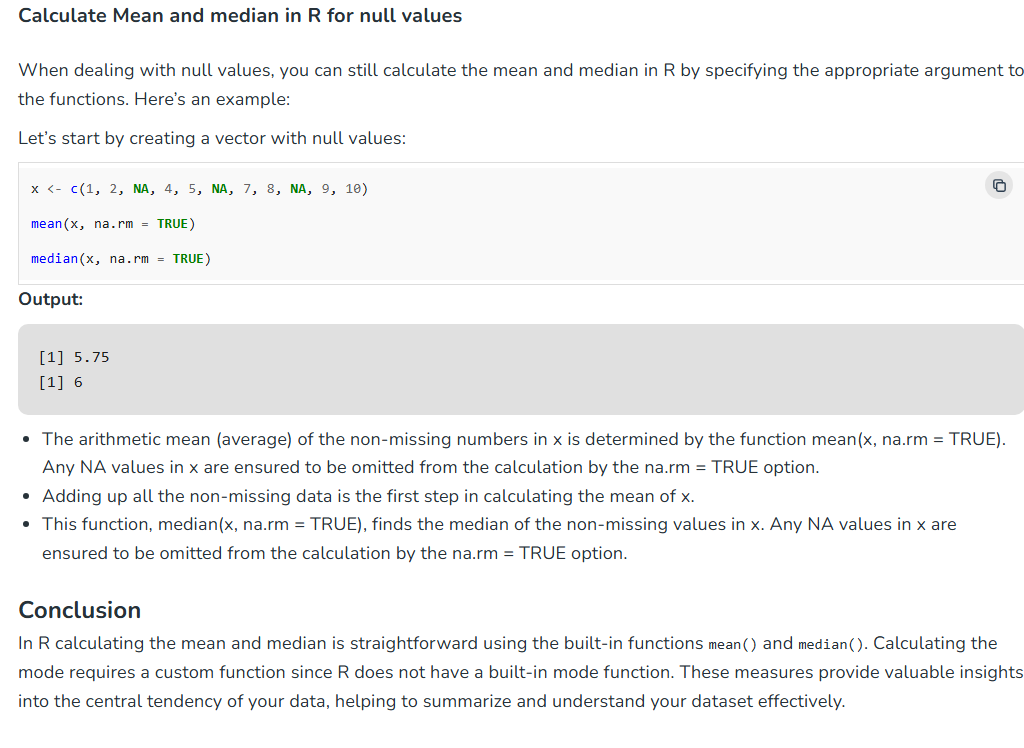
It is the value that has the highest frequency in the given data set. The data set may have no mode if the frequency of all data points is the same. Also, we can have more than one mode if we encounter two or more data points having the same frequency. There is no inbuilt function for finding mode in R, so we can create our own function for finding the mode or we can use the package called modest.

[Mode=The value that appears most frequently in the dataset][Mode=The value that appears most frequently in the dataset]

### Creating a user-defined function for finding Mode

There is no in-built function for finding mode in R. So let’s create a user-defined function that will return the mode of the data passed. We will be using the [table()](https://www.geeksforgeeks.org/create-a-tabular-representation-of-data-in-r-programming-table-function/) method for this as it creates a categorical representation of data with the variable names and the frequency in the form of a table. We will sort the column Age column in descending order and will return the 1 value from the sorted values.

**

**

# Exploring Statistical Measures in R: Average, Variance, and Standard Deviation Explained

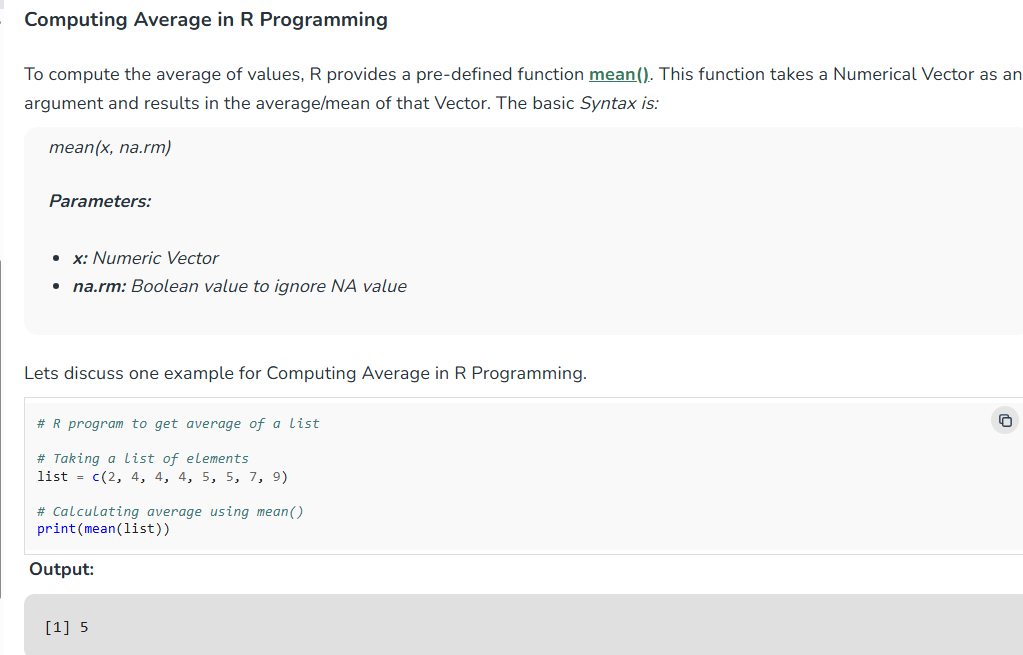
Statistical measures like average, variance, and standard deviation are crucial in data analysis. These metrics help summarize data and understand its distribution. In this article, we’ll explore how to calculate these measures in R. The R language provides very easy methods to calculate the average, variance, and standard deviation.

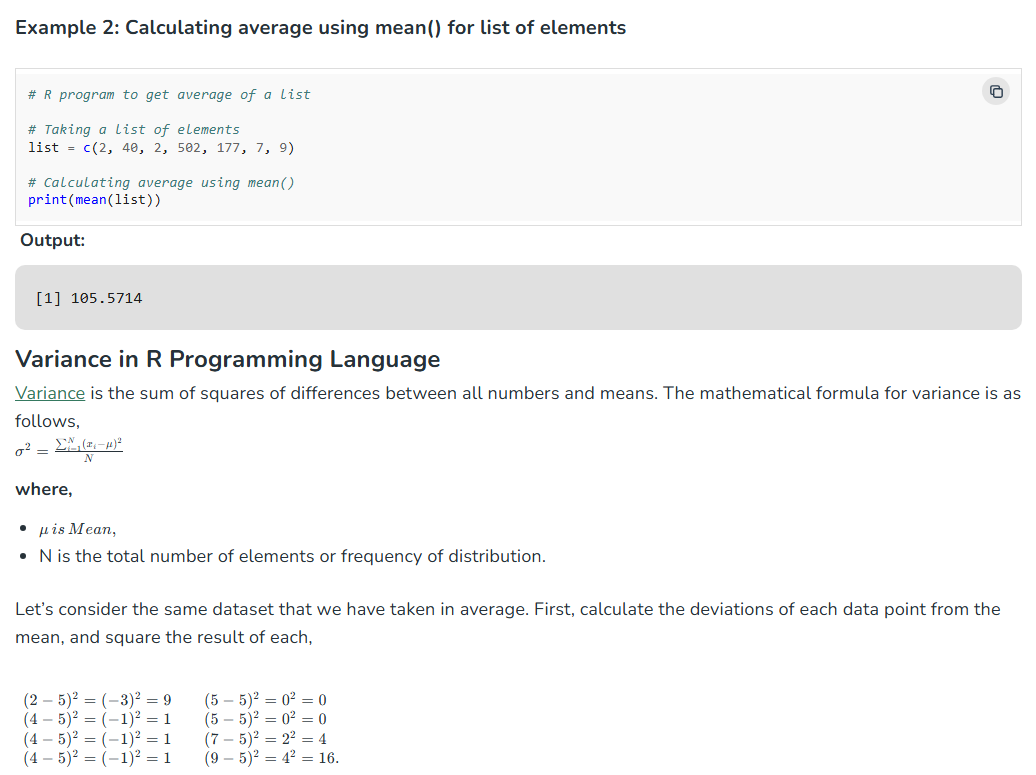
**Table of Content**

* [Average in R Programming](https://www.geeksforgeeks.org/calculate-the-average-variance-and-standard-deviation-in-r-programming/?ref=lbp#average-in-r-programming)
* [Variance in R Programming Language](https://www.geeksforgeeks.org/calculate-the-average-variance-and-standard-deviation-in-r-programming/?ref=lbp#variance-in-r-programming-language)
* [Standard Deviation in R Programming Language](https://www.geeksforgeeks.org/calculate-the-average-variance-and-standard-deviation-in-r-programming/?ref=lbp#standard-deviation-in-r-programming-language)

## Average in R Programming

An [average](https://www.geeksforgeeks.org/average/) is a number expressing the central or typical value in a set of data, in particular the mode, median, or (most commonly) the mean, which is calculated by dividing the sum of the values in the set by their number.

**

**

**

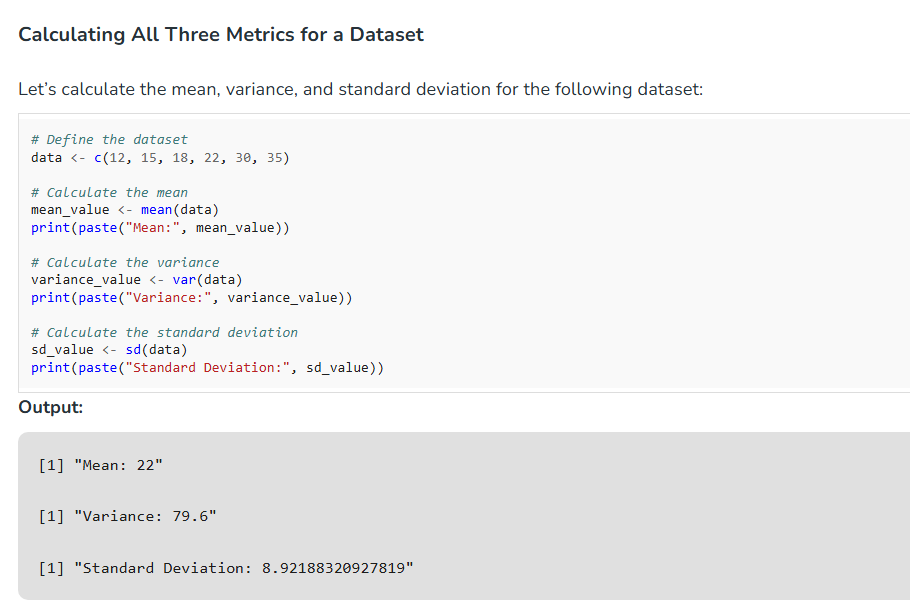
## Standard Deviation in R Programming Language

[Standard Deviation](https://www.geeksforgeeks.org/standard-deviation-formula/) is the square root of variance. It is a measure of the extent to which data varies from the mean. The mathematical formula for calculating standard deviation is as follows,    
StandardDeviation=variance    *StandardDeviation*=*variance*​

Standard Deviation for the above data,  
StandardDeviation=4=2*StandardDeviation*=4​=2

### Computing Standard Deviation in R

**

**

## Conclusion

In R, calculating the average, variance, and standard deviation is simple and efficient using the mean(), var(), and sd() functions. These metrics provide valuable insights into the central tendency and dispersion of your data, helping to summarize and understand the distribution.

[Probability](https://www.geeksforgeeks.org/mathematics-probability/)theory is a fundamental concept in mathematics and statistics that plays a crucial role in various fields such as finance, engineering, medicine, and more. Understanding probabilities allows us to make informed decisions in uncertain situations. In this comprehensive guide, we'll delve into the basics of probabilities using [R Programming Language](https://www.geeksforgeeks.org/r-programming-language-introduction/).

## Basic Concepts of Probability in R

Probability in R is the measure of the likelihood that an event will occur. The probability of an event A, denoted as P(A), lies between 0 and 1, where 0 indicates impossibility and 1 indicates certainty. Some key concepts include:

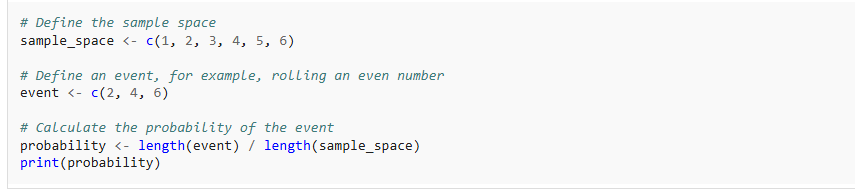
* [**Sample Space (S):**](https://www.geeksforgeeks.org/sample-space-probability/)The set of all possible outcomes of a random experiment.
* [**Event:**](https://www.geeksforgeeks.org/types-of-events-in-probability/) Any subset of the sample space.
* [**Probability of an Event:**](https://www.geeksforgeeks.org/events-in-probability/)The likelihood of occurrence of an event, calculated as the ratio of favorable outcomes to the total number of outcomes.

### Calculating Probabilities in R

R offers various functions and packages for calculating Probability in R and performing statistical analyses. Some commonly used functions include:

* [dbinom():](https://www.geeksforgeeks.org/a-guide-to-dbinom-pbinom-qbinom-and-rbinom-in-r/) Computes the probability mass function (PMF) for the binomial distribution.
* [pnorm():](https://www.geeksforgeeks.org/a-guide-to-dnorm-pnorm-rnorm-and-qnorm-in-r/)Calculates the cumulative distribution function (CDF) for the normal distribution.
* [dpois():](https://www.geeksforgeeks.org/how-to-draw-poisson-density-curve-in-r/) Computes the PMF for the Poisson distribution.
* [punif():](https://www.geeksforgeeks.org/compute-the-value-of-cdf-on-uniform-distribution-in-r-programming-punif-function/)Calculates the CDF for the uniform distribution.

Here is the basic example of calculating Probability in R:



**Output:**

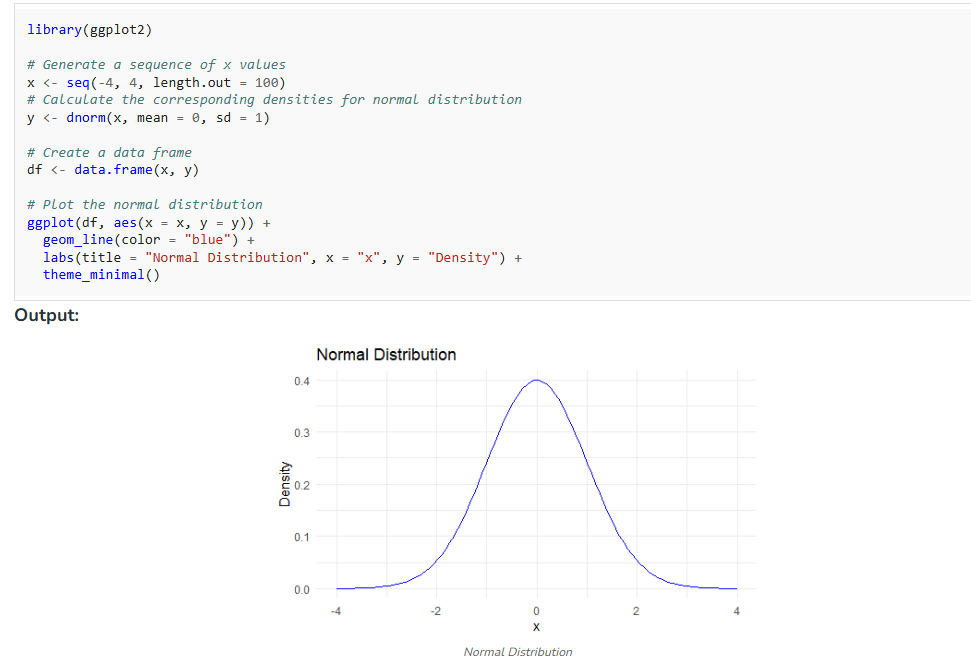
[1] 0.5

**Probability Distributions in R**

R provides extensive support for probability distributions, which are mathematical functions that describe the likelihood of different outcomes in a random experiment. Common probability distributions include:

* [**Uniform Distribution:**](https://www.geeksforgeeks.org/continuous-uniform-distribution-in-r/)All outcomes are equally likely.
* [**Normal Distribution:**](https://www.geeksforgeeks.org/normal-distribution-in-r/) Symmetric bell-shaped curve, characterized by mean (μ) and standard deviation (σ).
* [**Binomial Distribution:**](https://www.geeksforgeeks.org/binomial-distribution-in-r-programming/) Describes the number of successes in a fixed number of independent Bernoulli trials.
* [**Poisson Distribution:**](https://www.geeksforgeeks.org/poisson-distribution-in-r/)Models the number of events occurring in a fixed interval of time or space.

Let’s visualize the normal distribution with a mean of 0 and standard deviation of 1.



### Visualizing Probabilities in R

Visualization is essential for gaining insights from Probability in R and it offers numerous packages such as ggplot2, lattice, and base graphics for creating visualizations. Common plots include histograms, density plots, boxplots, and scatter plots, which help in understanding the shape and characteristics of probability distributions.

*# Visualizing the binomial distribution of coin flips*

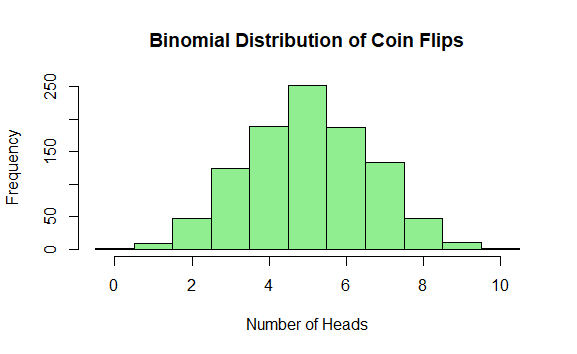
flips <- rbinom(1000, size = 10, prob = 0.5)

hist(flips, breaks = seq(-0.5, 10.5, by = 1), col = "lightgreen",

main = "Binomial Distribution of Coin Flips", xlab = "Number of Heads",

ylab = "Frequency")

**Output:**



## Conclusion

Calculating Probability in R is the part of statistics and data analysis, enabling us to quantify uncertainty and make informed decisions. By mastering probabilities using R, you gain powerful tools for analyzing data, conducting simulations, and drawing meaningful insights. With the knowledge and skills gained from this guide, you'll be well-equipped to tackle real-world problems involving uncertainty and randomness.

# Normal Distribution in R

In this article, we will discuss in detail the normal distribution and different types of built-in functions to generate normal distribution using [R Programming Language](https://www.geeksforgeeks.org/r-programming-language-introduction/).

## What is Normal Distribution in R?

[Normal Distribution](https://www.geeksforgeeks.org/normal-distribution/)in R 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. For example, the height of the population, shoe size, IQ level, rolling a dice, and many more. It is generally observed that data distribution is normal when there is a random collection of data from independent sources. In R, there are 4 built-in functions to generate normal distribution:

1. [dnorm()](https://www.geeksforgeeks.org/a-guide-to-dnorm-pnorm-rnorm-and-qnorm-in-r/)
2. [pnorm()](https://www.geeksforgeeks.org/a-guide-to-dnorm-pnorm-rnorm-and-qnorm-in-r/)
3. [qnorm()](https://www.geeksforgeeks.org/a-guide-to-dnorm-pnorm-rnorm-and-qnorm-in-r/)
4. [rnorm()](https://www.geeksforgeeks.org/a-guide-to-dnorm-pnorm-rnorm-and-qnorm-in-r/)

Now we will discuss all 4 built-in functions to generate normal distribution in detail using R Programming Language.

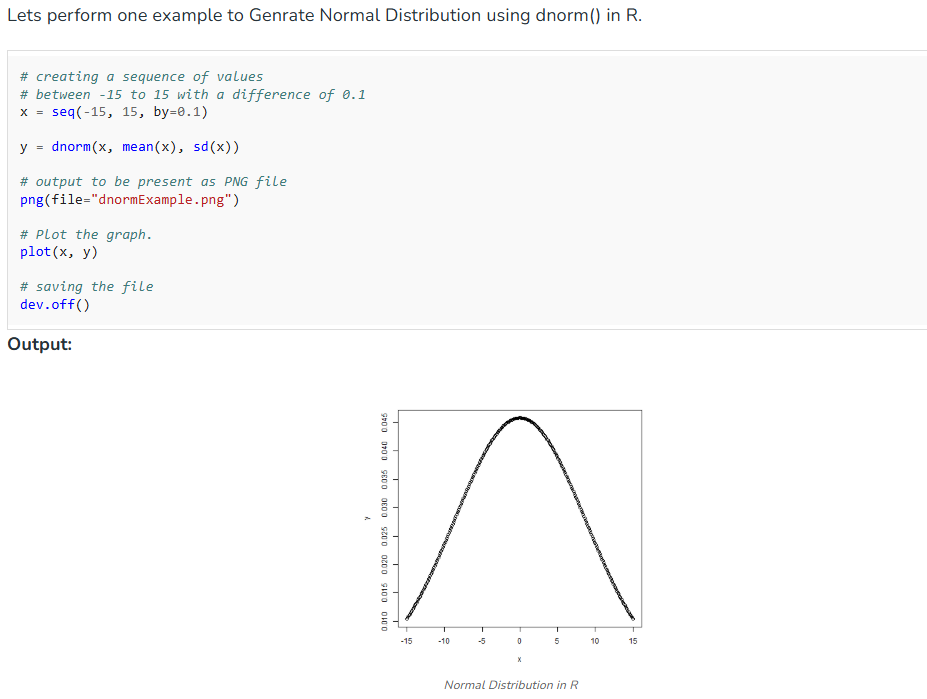
### 1: Genrate Normal Distribution using dnorm()

The dnorm() function computes the density (height of the probability density function) at a given point for a normal distribution.

*norm(x, mean = 0, sd = 1, log = FALSE)*

*Where,*

* ***x:*** *A vector of quantiles.*
* ***mean:*** *The mean of the distribution.*
* ***sd:*** *The standard deviation of the distribution.*
* ***log:*** *If TRUE, probabilities p are returned as log(p).*



### 2: Genrate Normal Distribution using pnorm()

The pnorm() function calculates the cumulative probability up to a given quantile.

*pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)*

*Where,*

* ***q:*** *A vector of quantiles.*
* ***mean:*** *The mean of the distribution.*
* ***sd:*** *The standard deviation of the distribution.*
* ***lower.tail:*** *If TRUE, probabilities are P[X ≤ x]; otherwise, P[X > x].*
* ***log.p:*** *If TRUE, probabilities p are returned as log(p).*

### 4: Genrate Normal Distribution using rnorm()

The rnorm() function generates random numbers following a normal distribution.

*rnorm(n, mean = 0, sd = 1)*

* ***n:*** *Number of observations to generate.*
* ***mean:*** *The mean of the distribution.*
* ***sd:*** *The standard deviation of the distribution.*

## ****Conclusion****

The normal distribution is a crucial concept in statistics, with applications ranging from data analysis to inferential statistics. R provides a variety of functions (dnorm(), pnorm(), qnorm(), rnorm()) to work with normal distributions, making it easy to perform statistical analyses and generate visualizations. Understanding how to use these tools is essential for effective data analysis in R.

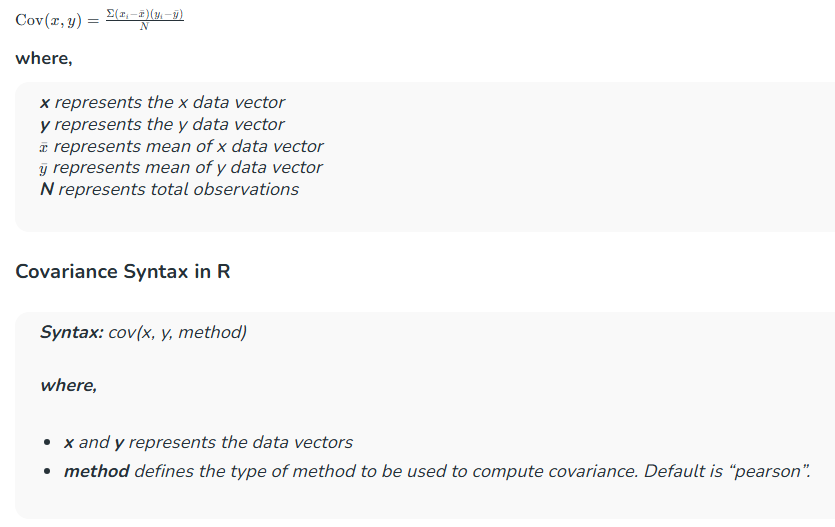
# Covariance and Correlation in R Programming

[**Covariance**](https://www.geeksforgeeks.org/mathematics-covariance-and-correlation/) and [**Correlation**](https://www.geeksforgeeks.org/mathematics-covariance-and-correlation/) are terms used in statistics to measure relationships between two random variables. Both of these terms measure linear dependency between a pair of random variables or bivariate data. They both capture a different component of the relationship, despite the fact that they both provide information about the link between variables. Let’s investigate the theory underlying correlation and covariance:

We can discuss some of the main difference between them as below:In this article, we are going to discuss **cov()**, **cor()** and **cov2cor()** functions in [R](https://www.geeksforgeeks.org/introduction-to-r-programming-language/) which use covariance and correlation methods of statistics and probability theory.

## Covariance in R Programming Language

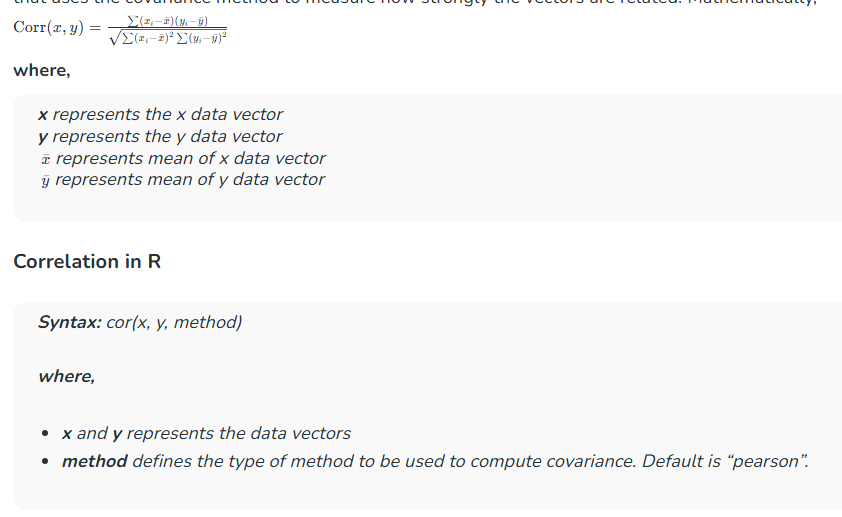
In R programming, covariance can be measured using the **cov()** function. Covariance is a statistical term used to measure the direction of the linear relationship between the data vectors. Mathematically,

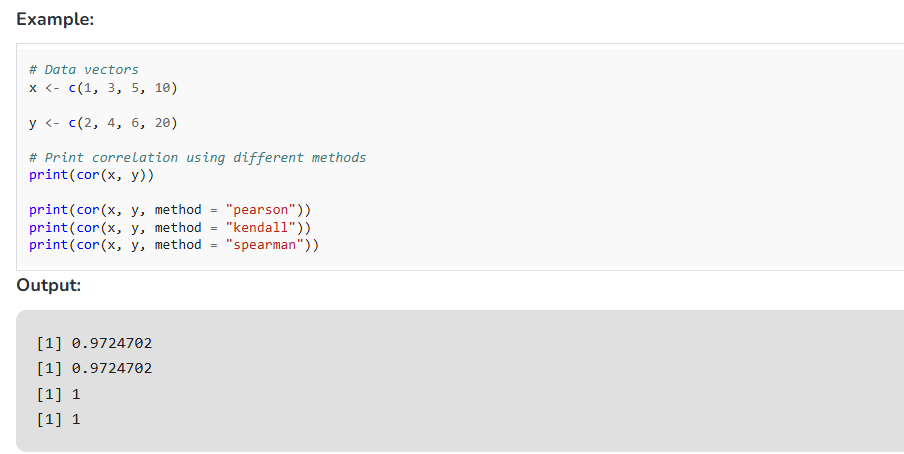


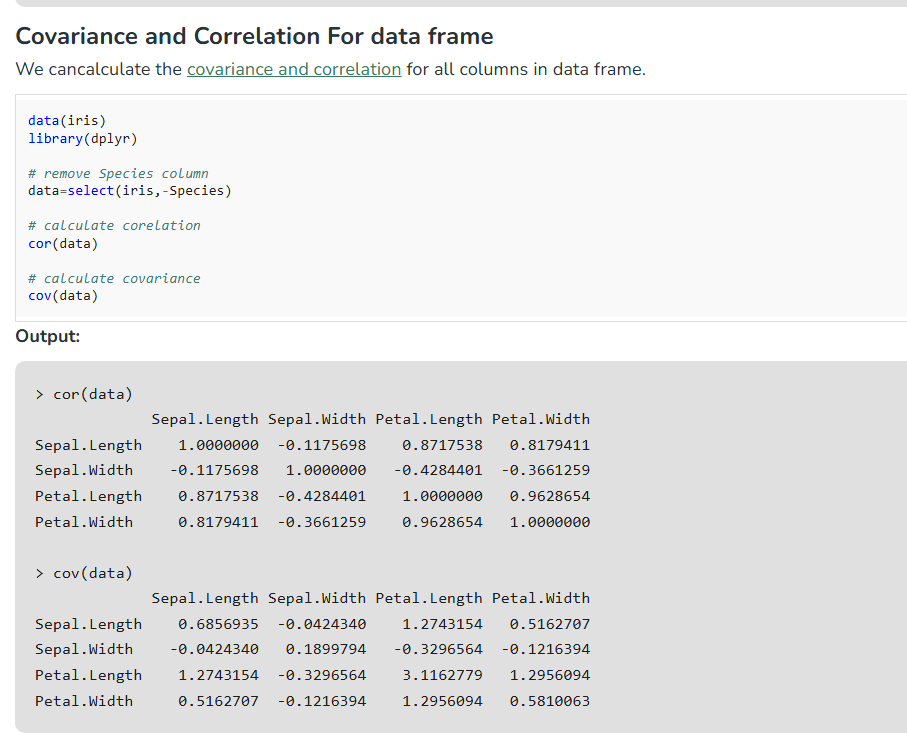


## Correlation in R Programming Language

**cor()** function in R programming measures the correlation coefficient value. Correlation is a relationship term in statistics that uses the covariance method to measure how strongly the vectors are related. Mathematically,







## Conversion of Covariance to Correlation in R

**cov2cor()** function in R programming converts a [covariance matrix](https://www.geeksforgeeks.org/covariance-matrix/) into a corresponding correlation matrix.

***Syntax:*** *cov2cor(X)*

***where,***

* ***X*** *and* ***y*** *represents the covariance square matrix*

## Conversion of Covariance to Correlation in R

**cov2cor()** function in R programming converts a [covariance matrix](https://www.geeksforgeeks.org/covariance-matrix/) into a corresponding correlation matrix.

***Syntax:*** *cov2cor(X)*

***where,***

* ***X*** *and* ***y*** *represents the covariance square matrix*



# Skewness in R Programming

In this article, we will discuss in detail Skewness in R Language. It is a numerical method to analyze the shape of a data set unlike, plotting graphs and histograms using [R Programming Language](https://www.geeksforgeeks.org/r-programming-language-introduction/).

## Introduction to Skewness in R

[Skewness](https://www.geeksforgeeks.org/skewness-measures-and-interpretation/)is a statistical numerical method to measure the asymmetry of the distribution or data set. It tells about the position of the majority of data values in the distribution around the mean value. A fundamental statistical notion called skewness quantifies the asymmetries in data distributions. It is essential to several disciplines, including data analysis, social sciences, economics, and finance.

* **Positive Skewness (Right-Skewed)**: The tail on the right side of the distribution is longer or fatter. The mean is greater than the median.
* **Negative Skewness (Left-Skewed)**: The tail on the left side is longer or fatter. The mean is less than the median.
* **Zero Skewness (Symmetrical)**: The distribution is symmetrical, and the mean and median are equal.

### Mathematical Definition of Skewness

Skewness can be mathematically defined as:

Skewness=(*n*−1)(*n*−2)*n*​∑*i*=1*n*​(*sxi*​–*x*ˉ​)3

**Where:**

* n is the number of observations, xi​ is the i-th observation.
* xˉ is the sample mean., s is the sample standard deviation.

**A skewness value:**

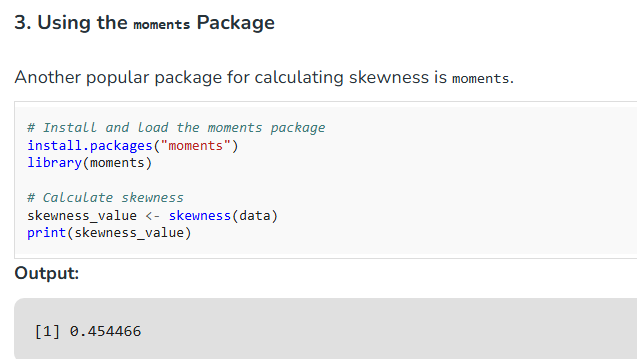
* Greater than 0 indicates positive skewness.
* Less than 0 indicates negative skewness.
* Equal to 0 indicates no skewness (symmetrical distribution).

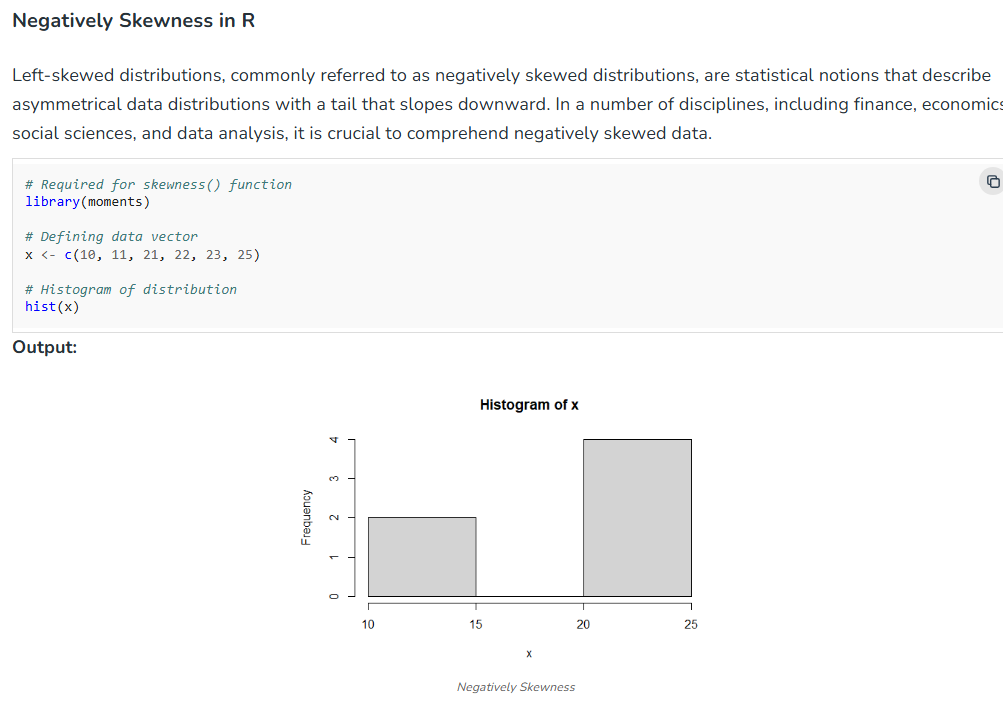
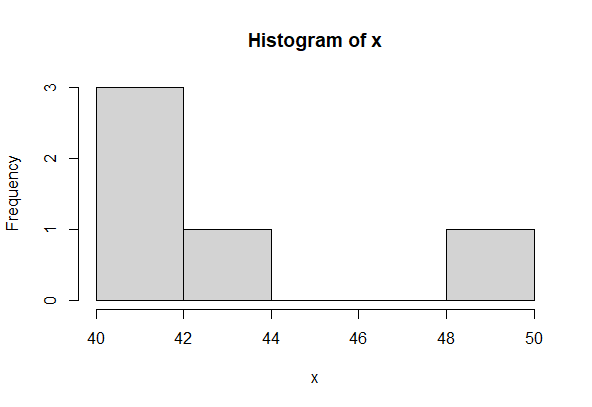
## How to Calculate Skewness in R

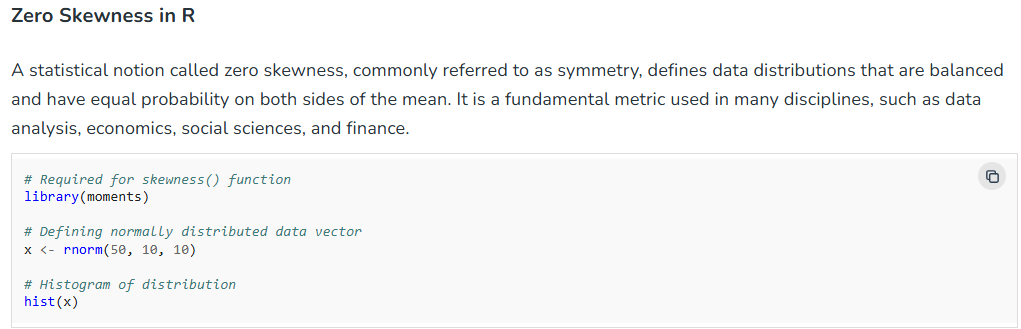
R provides multiple ways to calculate skewness, including base R functions, specialized packages, and custom implementations.

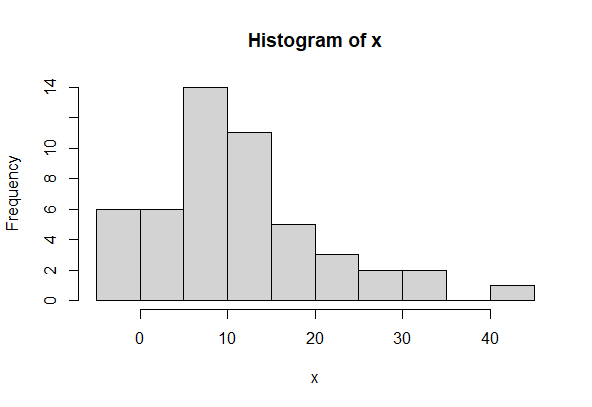
### 1. ****Base R Implementation****











### Importance of Skewness

Here are the some of the main Importance of Skewness.

* **Normality Assumption**: Many statistical methods, like t-tests and ANOVA, assume data is normally distributed. Skewness can indicate departures from normality.
* **Impact on Mean and Median**: In skewed distributions, the mean is pulled towards the tail, making the median a better measure of central tendency.
* **Interpretation in Data Analysis**: Understanding the skewness of your data can influence decisions in model selection, data transformation, and interpretation of results.

## Conclusion

Skewness is a critical statistical measure that provides insights into the shape of your data distribution. Understanding and addressing skewness is essential for accurate data analysis and modeling. In R, you can easily calculate and visualize skewness using various packages and techniques, allowing you to make informed decisions about data transformation and model selection.