TASK 5:

Range , summary,mean, variance ,median, standard deviation

The range can be defined as the difference between the maximum and minimum elements in the given data, the data can be a vector or a dataframe. So we can define the range as the difference between maximum\_value – minimum\_value

## Find range in a vector using min and max functions

We can find the range by performing the difference between the minimum value in the vector and the maximum value in the given vector. We can find maximum value using max() function and minimum value by using min() function.

The **range() function** in R is used to return a vector with two elements:

* The first element represents the minimum value of the input vector.
* The second element is the maximum value of the input vector.

**Syntax**

The range() function takes the syntax below:

range(…, na.rm = FALSE)

**Parameter value**

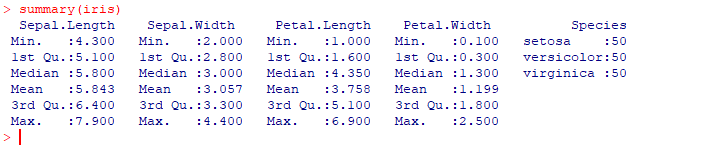
The range() function takes the following parameter values:

* **...**: This represents any numeric or character objects or vectors.
* **na.rm**: This takes a Boolean value (TRUE or FALSE) indicating if the NaN (Not a Number) values should be omitted or not.
* max(vector)-min(vector)
* If a vector contains NA values then we should use na.rm function to exclude NA values
* **Example:**

## R

|  |
| --- |
| # create vector  data <-c(12, 45, NA, NA, 67, 23, 45, 78, NA, 89)  # display  print(data)  # find range  print(max(data, na.rm=TRUE)-min(data, na.rm=TRUE)) |

* **Output:**
* [1] 12 45 NA NA 67 23 45 78 NA 89
* [1] 77

2) 

## Mean

It is calculated by taking the sum of the values and dividing with the number of values in a data series.

The function **mean()** is used to calculate this in R.

### Syntax

The basic syntax for calculating mean in R is −

mean(x, trim = 0, na.rm = FALSE, ...)

Following is the description of the parameters used −

* **x** is the input vector.
* **trim** is used to drop some observations from both end of the sorted vector.
* **na.rm** is used to remove the missing values from the input vector.

### Example

[Live Demo](http://tpcg.io/6bbZx4)

# Create a vector.

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

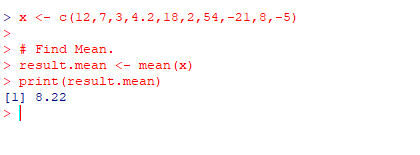
# Find Mean.

result.mean <- mean(x)

print(result.mean)

When we execute the above code, it produces the following result −

[1] 8.22



## Median

The middle most value in a data series is called the median. The **median()** function is used in R to calculate this value.

### Syntax

The basic syntax for calculating median in R is −

median(x, na.rm = FALSE)

Following is the description of the parameters used −

* **x** is the input vector.
* **na.rm** is used to remove the missing values from the input vector.

### Example

[Live Demo](http://tpcg.io/Hs6UZ9)

# Create the vector.

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

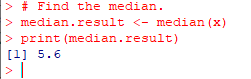
# Find the median.

median.result <- median(x)

print(median.result)

When we execute the above code, it produces the following result −

[1] 5.6



 Standard deviation is the measure of the dispersion of the values. It can also be defined as the square root of variance.

Formula of sample standard deviation:

**where,**

* s = sample standard deviation
* N = Number of entities
* = Mean of entities

Basically, there are two different ways to calculate standard Deviation in R Programming language, both of them are discussed below.

**Method 1: Naive approach**

In this method of calculating the standard deviation, we will be using the above standard formula of the sample standard deviation in R language.

**Example 1:**

* R

|  |
| --- |
| v <- c(12,24,74,32,14,29,84,56,67,41)  s<-sqrt(sum((v-mean(v))^2/(length(v)-1)))  print(s) |

**Output:**

*[1] 25.53886*

### Method 2: Using sd()

The sd() function is used to return the standard deviation.

***Syntax:****sd(x, na.rm = FALSE)*

***Parameters:***

* ***x:****a numeric vector, matrix or data frame.*
* ***na.rm:****missing values be removed?*

***Return:****The sample standard deviation of x.*

**Example 1:**

* R

|  |
| --- |
| v <- c(12,24,74,32,14,29,84,56,67,41)    s<-sd(v)    print(s) |

**Output:**

*[1] 25.53886*

**Example 2:**

* R

|  |
| --- |
| v <- c(71,48,98,65,45,27,39,61,50,24,17)    s1<-sqrt(sum((v-mean(v))^2/(length(v)-1)))  print(s1)  s2<-sd(v)  print(s2) |

**Output:**

*[1] 23.52175*

# Compute Variance

**var()** function in [R Language](https://www.geeksforgeeks.org/introduction-to-r-programming-language/) computes the sample variance of a vector. It is the measure of how much value is away from the mean value.

***Syntax:****var(x)*

***Parameters:******x :****numeric vector*

|  |
| --- |
| # R program to illustrate  # variance of vector  # Create example vector  x <- c(1, 2, 3, 4, 5, 6, 7)  # Apply var function in R  var(x)  print(x) |
|  |

**Output:**

4.667

**library**(datasets)

data(iris)

summary(iris)

## ****Boxplot using ggplot****

*# Typical boxplot*

boxplot(Petal.Length~Species, data = iris, xlab = c('Species'), ylab = c('Petal Length'))