Inbuilt R functions

Mathematical Operations

R is a powerful programming language for performing mathematical operations and statistical calculations. Here are some common mathematical operations in R.

1. Arithmetic Operations: R can perform basic arithmetic operations such as addition (+), subtraction (-), multiplication (*), and division (/).

```
# Addition and Subtraction
5+9-3
```

```
[1] 11
```

```
# Multiplication and Division
(5 + 3) * 7 /2
```

```
[1] 28
```

2. Exponentiation and Logarithms: R can raise a number to a power using the ^ or ** operator or take logarithms.

```
# exponentiation
2^6
```

```
[1] 64
```

```
# Exponential of x=2 i.e. e^2
exp(2)
```

```
[1] 7.389056
```

```
# logarithms base 2 and base 10
log2(64) + log10(100)
```

```
[1] 8
```

3. Other mathematical functions: R has many additional useful mathematical functions.

- We can find the absolute value, square roots, remainder on division.

```
# absolute value of x=-5
abs(-9)
```

```
[1] 9
```

```
# square root of x=70
sqrt(70)
```

```
[1] 8.3666
```

```
# remainder of the division of 11/3
11 %% 3
```

```
[1] 2
```

- We can round numbers, find their floor, ceiling or up to a number of significant digits

```
# Value of pi to 10 decimal places
pi = 3.1415926536

# round(): This function rounds a number to the given number of decimal places
# For example, round(pi, 3) returns 3.142
round(pi,3)
```

```
[1] 3.142
```

```r
# ceiling(): This function rounds a number up to the nearest integer.
# For example, ceiling(pi) returns 4
ceiling(pi)
```

```
[1] 4
```

```r
# floor(): This function rounds a number down to the nearest integer.
# For example, floor(pi) returns 3.
floor(pi)
```

```
[1] 3
```

```r
# signif(): This function rounds a number to a specified number of significant digits.
# For example, signif(pi, 3) returns 3.14.
signif(pi,3)
```

```
[1] 3.14
```

4. Statistical calculations: R has many built-in functions for statistical calculations, such as mean, median, standard deviation, and correlation.

```r
x <- c(0, 1, 1, 2, 3, 5, 8)   # create a vector of 7 Fibonacci numbers
length(x) # count how many numbers do we have
```

```
[1] 7
```

```r
mean(x)   # calculate the mean
```

```
[1] 2.857143
```

```r
median(x) # calculate the median
```

```
[1] 2
```

```r
sd(x)     # calculate the standard deviation
```

```
[1] 2.794553
```

```r
y <- c(1, 2, 3, 4, 5, 6, 7) # create a new vector of positive integers
cor(x,y)  # calculate the correlation between x and y
```

```
[1] 0.938668
```

Assigning values to variables

1. A variable can be used to store a value. For example, the R code below will store the sales in a variable, say "sales":

```r
# use the assignment operator <-
sales <- 9
# alternately, use =
sales = 9
```

2. It is possible to use <- or = for variable assignments.
3. R is case-sensitive. This means that `Sales` is different from `sales`
4. It is possible to perform some operations with it.

```r
# multiply sales by 2
2 * sales
```

```
[1] 18
```

5. We can change the value stored in a variable

```r
# change the value
sales <- 15
# display the revised sales
sales
```

```
[1] 15
```

6. The following R code creates two variables holding the sales and the price of a product and we can use them to compute the revenue.

```r
# sales
sales <- 5

# price
price <- 7

# Calculate the revenue
revenue <- price*sales
revenue
```

```
[1] 35
```