

Installing R and Rstudio

To install R and RStudio on your system, you can follow these steps general instructions. Please note that the steps might vary slightly depending on your operating system.

Installing R:-

For windows:-

Download the latest version of R for windows from the official CRAN website: CRAN-R for windows. Run the installer and follow the installation instructions.

For Mac OS:-

Download the latest version of R for macos from the official CRAN-R for macos. Run the installer package (.pkg) and follow the installation instructions.

For Linux:-

The method of installing R on Linux can vary depending on your distribution. For Debian-based systems (like Ubuntu), you can use the following commands in the terminal:

```
sudo apt update
```

```
sudo apt install r-base
```

for Red Hat-based systems (like fedora), you can use:

```
sudo yum install R
```

Installing R studio:-

Download the latest version of RStudio Desktop from the official RStudio website: RStudio - Download. Choose the appropriate installer for your operating system (Windows, macOS, or Linux) and follow the installation instructions.

Verifying the installation:-

After installation, you can check if R and RStudio are installed correctly.

For R:

Open a terminal or command prompt

Type R and press Enter. This should launch the R console.

For R Studio:

2. Basic functionality of R, variable, data types in R

R is a programming language and environment designed for statistical computing and graphics. Here are some basic functionalities, variable usage, and common data types in R:

Basic functionality:-

Arithmetic Operations:-

R supports standard Arithmetic Operations like addition (+), subtraction (-), multiplication (*), division (/), and exponentiation (^).

```
a <- 5  
b <- 3  
result <- a+b
```

functions:-

R has a wide range of built-in functions and allows users to define their own functions.

```
Square <- function(x) {  
  return (x^2)  
}  
result <- square(4)
```

Vectors:-

Vectors are one-dimensional arrays in R.

```
numbers <- c(1, 2, 3, 4, 5)
```

Data frames:-

Data frames are two-dimensional tables where data can be stored in rows and columns.

```
df <- data.frame{  
  Name = c("Alice", "Bob", "Charlie"),  
  Age = c(25, 30, 22)  
}
```

Variables and Data Types:-

Assigning Variables:

The assignment operator `<-` is commonly used to assign values to variables.

`x <- 10`

Data types:-

R has several basic data types, including:

Numeric : for real numbers.

Integer : for whole numbers.

Character : for text

Logical : for boolean values (TRUE or FALSE).

```
num-var <- 10.5      # Numeric  
int-var <- 7L       # Integer  
char-var <- "Hello" # Character  
bool-var <- TRUE    # Logical
```

Checking Data types:-

The `class()` function is used to check the data type of a variable

```
class(num-var) # Returns "numeric"
```

```
class(char-var) # returns "character"
```

Conversion:-

R can automatically convert between certain data types

```
num-var <- as.integer(10.5) # converts 10.5 to integer(10)
```

factors:-

factors are used to represent categorical data.

```
gender <- c("Male", "female", "Male", "female")
```

```
gender-factor <- as.factor(gender)
```

These are some fundamental concepts in R. As you explore R further, you'll encounter more advanced data structures, statistical functions, and data visualization tools that make R a powerful language for data analysis and statistics.

Week 9
• Implement R-Script to show the usage of various operators available in the R language

Program:-

Arithmetic Operators

a <- 10

b <- 5

Addition <- a+b

Subtraction <- a-b

Multiplication <- a*b

Division <- a/b

Remainder <- a%..% b

Exponentiation <- a^b

cat ("Arithmetic operators : \n")

cat ("Addition : ", addition, "\n")

cat ("Subtraction : ", Subtraction, "\n")

cat ("Multiplication : ", Multiplication, "\n")

cat ("Division : ", Division, "\n")

cat ("Remainder : ", Remainder, "\n")

cat ("Exponentiation : ", Exponentiation, "\n\n")

Relational Operators

x <- 1

y <- 12

cat ("Relational Operator : \n")

cat ("x is equal to y : ", x==y, "\n")

cat ("x is not equal to y : ", x!=y, "\n")

cat ("x is greater than y : ", x>y, "\n")

cat ("x is less than y : ", x<y, "\n")

cat ("x is greater than or equal to : ", x>=y, "\n")

cat ("x is less than or equal to : ", x<=y, "\n\n")

Logical Operators

p <- TRUE

q <- FALSE

cat ("Logical Operations : \n")

cat ("Logical AND (P AND q) : ", P&q, "\n")

cat ("Logical OR (P or q) : ", P|q, "\n")

cat ("Logical NOT (NOT P) : ", !p, "\n\n")

Assignment Operators

cat ("Assignment Operations:\n")

z <- 15

cat ("z = ", z, "\n")

z <- z + 5

cat ("z+=5, z=, "z, "\n")

z <- z * z

cat ("z*=2, z=", z, "\n\n")

Miscellaneous Operators

cat ("Miscellaneous Operators:\n")

cat ("concatenation:", paste ("Hello", "World"), "\n")

cat ("sequence:", 1:5, "\n")

cat ("Membership:", 3 %in% c (1, 2, 3, 4, 5), "\n")

Output

Arithmetic Operators:

Addition: 15

Subtraction: 5

multiplication: 50

division: 2

remainder: 0

Exponentiation: 1e+05

Relational Operators

x is equal to 4 : false

x is not equal to 4 : true

x is greater than 4 : false

x is less than 4 : true

x is greater than or equal to 4 : false

x is less than or equal to 4 : false

x is less than or equal to 4 : true

Logical Operators:-

Logical AND (P AND Q) : false

Logical OR (P OR Q) : true

Logical NOT (NOT P) : false

Assignment Operators

$z = 15$

$z+ = 15 \quad z = 20$

$z^+ = 8 \quad z = 110$

Miscellaneous Operator

Concatenation : Hello world

Sequence : 12345

membership : True.

9. Implement R-Script to read persons age from keyboard and display whether he is eligible for vote.

Certainly! you can use "readline" function to take input from user and then use a simple 'if' statement to check person is eligible for voting.

```
age <- as.numeric(readline("Enter your age:"))
if (age >= 18)
{
  cat("you are eligible for voting\n")
} else {
  cat("you are not eligible for voting\n")
```

NOTE:- The 'as.numeric' function is used to convert the input from 'readline' to an numeric type, as 'readline' returns a character string.

Output

Enter your age:19

You are eligible for voting

3. Implement R-script to find biggest number between two numbers.

```
num1 <- numeric (readline ("Enter the first number:"))
num2 <- numeric (readline ("Enter the second number:"))

if (num1 > num2) {
  cat ("the biggest number is:", num1, "\n")
} else if (num2 > num1) {
  cat ("the biggest number is:", num2, "\n")
} else {
  cat ("Both numbers are equal, \n")
}
```

Output Enter first number: 79
Enter second number: 98

Greatest is 98

4. Implement R-script to check the given year is leap year or not.

Certainly! you can check whether a given year is a leap year or not by applying certain rules.

```
year <- as.numeric (readline ("Enter the year:"))
if ((year %% 4 == 0 & year %% 100 != 0) ||
    (year %% 400 == 0)) {
  cat (year, " is a leap year.\n")
} else {
  cat (year, " is not a leap year.\n")
}
```

Output

Enter the year : 2024

2024 is leap year.