Statistical Data Analysis and Visualization With R Programming: From Basics to Advanced Analytics

Resource Person

Sujit Kumar Roy

MSc. in Water resources Development

Bangladesh University of Engineering and Technology (**BUET**)

Phone: +8801748985407

Email: sujitroy.bejoy@gmail.com

Organizer

Advances in Geographical Research

Website: https://www.aigr.co.in/

WhatsApp: +91-94751 72399

aigeo.research@gmail.com

ADVANCES IN GEOGRAPHICAL RESEARCH

Making the Complex Simple

2nd Lecture

Outline

- R Variable
- Concatenate Elements
- R multiple Variable
- Variable Naming
- Basic Data Type
- R Operators
  - Arithmetic Operators
  - Relational Operators
  - Logical Operators
  - Assignment Operators
  - Miscellaneous Operators

R Variables

- Variables are containers for storing data values.
- R does not have a command for declaring a variable. A variable is created the moment you first assign a value to it. To assign a value to a variable, use the **<-** sign. To output (or print) the variable value, just type the variable name:
- **Example**
- name <- "John"
  age <- 40

  name # output "John"
  age # output 40

Concatenate Elements

- You can also concatenate, or join, two or more elements, by using the **paste()** function.
- To combine both text and a variable, R uses comma (,):
- **Example**
- text <- "awesome"

  paste("R is", text)
- Or,
- text1 <- "R is"
  text2 <- "awesome"

  paste(text1, text2)

R Multiple Variables

- R allows you to assign the same value to multiple variables in one line:
- # Assign the same value to multiple variables in one line
- **var1 <- var2 <- var3 <- "Orange"**

- # Print variable values
- var1
- var2
- var3

R Variable Names (Identifiers)

- A variable can have a short name (**like x and y**) or a more descriptive name (**age, carname, total_volume**).
- Rules for R variables are:
- A variable name must start with a letter and can be a combination of **letters, digits, period(.)** and **underscore(_)**. If it starts with period(.), it cannot be followed by a digit.
- A variable name cannot start with a number or underscore (_)
- Variable names are case-sensitive (**age, Age and AGE are three different variables**)
- **Reserved words cannot be used as variables (TRUE, FALSE, NULL, if...)**

# Legal variable names:	# Illegal variable names:
myvar <- "John"	2myvar <- "John"
my_var <- "John"	my-var <- "John"
myVar <- "John"	my var <- "John"
MYVAR <- "John"	_my_var <- "John"
myvar2 <- "John"	my_v@ar <- "John"
.myvar <- "John"	TRUE <- "John"

Finding Variables

To know all the variables currently available in the workspace we use the **ls()** function.

Deleting Variables

Variables can be deleted by using the **rm()** function.

Delete all the variable

All the variables can be deleted by using the **rm()** and **ls()** function together.

Basic Data Types

Basic data types in R can be divided into the following types:

- ✓ numeric - (**10.5, 55, 787**)
- ✓ integer - (**1L, 55L, 100L, where the letter "L" declares this as an integer**)
- ✓ complex - (**9 + 3i, where "i" is the imaginary part**)
- ✓ character (a.k.a. string) - **("k", "R is exciting", "FALSE", "11.5")**
- ✓ logical (a.k.a. boolean) - (**TRUE or FALSE**)

We can use the **class()** function to check the data type of a variable

Type Conversion

We can convert from one type to another with the following functions:

- **as.numeric()**
- **as.integer()**
- **as.complex()**

Print a New Line in String

- storing strings in variables

string1 <- "GEEKS"

string2 <- "FOR“

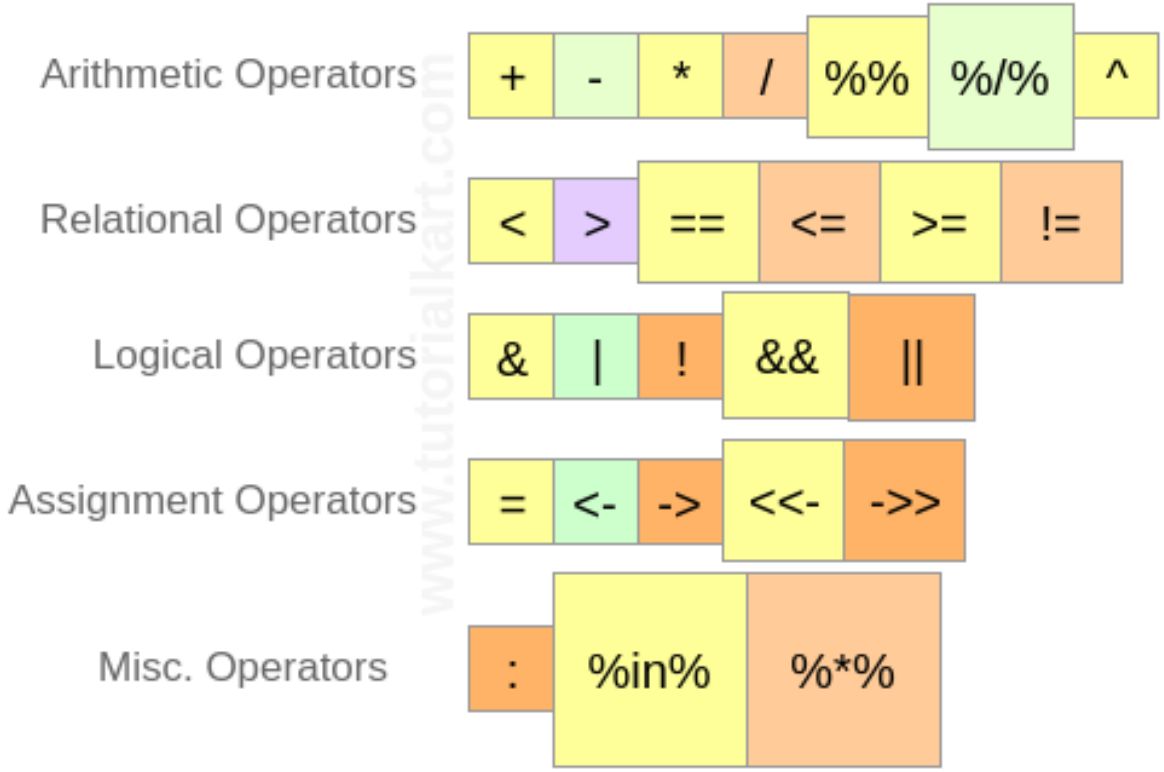
string3 <- "GEEKS"

- # passing variable in **cat()** without new
- # line serperator
- cat(string1,string2,string3)
- # passing a string using **\n** to split
- cat("GEEKS \nFOR \nGEEKS")
- # passing variables using \n
- cat(string1,"\n",string2,"\n",string3)

R Operators

We have the following types of operators in R programming

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Miscellaneous Operators



R Arithmetic Operators

```
a <- 5
b <- 16
```

Operator	Name	Description	Usage
+	Addition	Addition of two operands	a + b
–	Subtraction	Subtraction of second operand from first	a – b
*	Multiplication	Multiplication of two operands	a * b
/	Division	Division of first operand with second	a / b
%%	Modulus (Remainder from division)	Remainder from division of first operand with second	a %% b
%/%	Integer Division	Quotient from division of first operand with second	a %/% b
^	Exponent	First operand raised to the power of second operand	a^b

Relational Operators

```
a <- 5
b <- 16
```

Operator	Name	Description	Usage
<	Less than	Is first operand less than second operand	a < b
>	Greater than	Is first operand greater than second operand	a > b
==	Equal	Is first operand equal to second operand	a == b
<=	Less than or equal to	Is first operand less than or equal to second operand	a <= b
>=	Greater than or equal to	Is first operand greater than or equal to second operand	a > = b
!=	Not Equal	Is first operand not equal to second operand	a!=b

R Logical Operators

```
a <- 5
b <- 16
```

Operator	Description	Usage
&	Element-wise Logical AND operator. It returns TRUE if both elements are TRUE	a & b
&&	Logical AND operator - Returns TRUE if both statements are TRUE	a && b
|	Elementwise- Logical OR operator. It returns TRUE if one of the statement is TRUE	a | b
||	Logical OR operator. It returns TRUE if one of the statement is TRUE.	a || b
!	Logical NOT - returns FALSE if statement is TRUE	!a

Assignment Operators

Assignment Operators are those that help in assigning a value to the variable.

Operator	Description	Usage
=	Assigns right side value to left side operand	a = 3
<-	Assigns right side value to left side operand	a <- 5
->	Assigns left side value to right side operand	4 -> a
<<-	Assigns right side value to left side operand	a <<- 3.4
->>	Assigns left side value to right side operand	c(1,2) ->> a

Miscellaneous Operators

These operators does not fall into any of the categories mentioned above, but are significantly important during R programming for manipulating data.

Operator	Description	Usage
:	Creates series of numbers from left operand to right operand	a:b
%in%	Identifies if an element(a) belongs to a vector(b)	a %in% b
%*%	Performs multiplication of a vector with its transpose	A %*% t(A)

Thank You

We Hope You Enjoy The Presentation