R FUNCTIONS (IN-BUILT)

ACTS-DBDA

Sep 2021, CDAC Bangalore

R FUNCTIONS

- Almost everything in **R** is done through functions
- Functions in R can be classified as
  I. Numeric functions
  II. Character functions
  III. Statistical Functions
- To know all inbuilt R functions, we use the command `builtins()`
- For looking at help pages, we use `?command_name`

R GENERAL FUNCTIONS

- `abbreviate(x,2)` # Abbreviate the names in the vector x
- `abs(x)` # The absolute value of "x"
- `append()` # Add elements to a vector
- `Trigonometric functions` # Trigonometric function
- `addNA()` # Adds NA level
- `all()` # Checks all of the values true in a logical vector

R GENERAL FUNCTIONS

- `is.na()` # 'Not Available' / Missing Values
- `apply()` #Apply Functions Over Array Margins
- `array()` # Creation of Array
- `as.character()` # Creates the character vector
- `is.character()` # Checks if it is character vector
- `as.character.Date()` # Date Conversion Functions to and from Characters.

```
> dates <- c("02/27/92", "02/27/92", "01/14/92", "02/28/92", "02/01/92")
> as.Date(dates, "%m/%d/%y")
[1] "1992-02-27" "1992-02-27" "1992-01-14" "1992-02-28" "1992-02-01"
> |
```

R GENERAL FUNCTIONS

- `as.data.frame()` # Coerce to a Data Frame
- `is.data.frame()` # Checking if it is dataframe
- `as.factor()` # Coerce to a Factor
- `is.factor()` # Checking if it is factor
- `as.integer()` # Coerce to a Integer
- `is.integer()` # Checking if it is Number
- `as.matrix()` # Coerce to a Integer
- `is.matrix()` # Checking if it is Matrix

R GENERAL FUNCTIONS

- `which()` # Which indices are TRUE?
  - `which(LETTERS == "R")`
  - `which(ll <- c(TRUE, FALSE, TRUE, NA, FALSE, FALSE, TRUE))`
  - `which((1:12)%%2 == 0)`

- `as.vector()` # Coerce to a vector
- `is.vector()` # Checking if it is Matrix
- **Quotes**
  - \n newline
  - \t tab
  - \b backspace
  - \\ backslash

R FUNCTIONS

- `cbind()` # Column bind
- `grep()` # Pattern matching
  - `grep("a+", c("abc", "def", "cba a", "aa))`
- `identical()` # Test if 2 objects are *exactly* equal
- `jitter()` #Add a small amount of noise to a numeric vector
- `length(x)` # Return no. of elements in vector x
- `ls()` # List objects in current environment
- `paste(x)` # Concatenate vectors after converting to character
- `rep(1,5)` # Repeat the number 1 five times
- `rev(x)` # List the elements of "x" in reverse order

R FUNCTIONS

- `seq(1,10,0.4)` # Generate a sequence (1 -> 10, spaced by 0.4)
- `sequence()` # Create a vector of sequences
- `sort(x)` # Sort the vector x
- `tolower(),toupper()` # Convert string to lower/upper case letters
- `unique(x)` # Remove duplicate entries from vector system
- **Rounding Functions**
  - `floor(x), ceiling(x), round(x)`
- `Sys.time()` # Return system time
- `Sys.Date()` # Return system date
- `getwd()` # Return working directory
- `setwd()` # Change the working directory

R FUNCTIONS

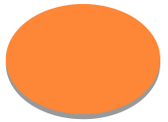
- `list.files()` # List files in a give directory
- **Built-in constants**:
  - pi, letters, LETTERS
- **Trignometric Functions**
  - cos(x), sin(x), tan(x), acos(x), asin(x), atan(x), atan2(y, x)
- **Arithmetic Operators**
  - x + y, x – y, x * y, x / y, x ^ y, x %% y
- `log(x),logb(),log10(),log2(),exp(),expm1(),log1p(),sqrt()`
- **Comparison operators**
  - `<, >, <=, >=, ==, !=`

R FUNCTIONS – GRAPHICAL FUNCTIONS

- `plot()` # Generic function for plotting of R objects
- `arrows()` # Draw arrows
- `abline()` # Adds a straight line to an existing graph
- `lines()` # Join specified points with line segments
- `segments()` # Draw line segments between pairs of points
- `hist(x)` # Plot a histogram of x
- `pairs()` # Plot matrix of scatter plots
- `matplot()` # Plot columns of matrices
- `pdf()` # Plot to pdf file
- `png()` # Plot to PNG file
- `jpeg()` # Plot to JPEG

R FUNCTIONS – STATISTICAL FUNCTIONS

- `cor.test()` # Perform correlation test
- Cumulative Functions for Vectors
  - `cumsum(); cumprod(); cummin(); cummax()`
- `mean(x), median(x), min(x), max(x), quantile(x), sd(), var()`
- `rnorm()` # Generate random data with Gaussian/uniform distribution
- `sample()` # Random samples

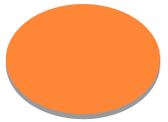


R FUNCTIONS – CHARACTER FUNCTIONS

- `substr()` #Extract or replace substrings in a character vector.
- `grep(pattern, x)` # Search for pattern in character vector.
- `gsub(pattern, replacement, x)` #Finds all pattern in x and replace with replacement text.
- `sub(pattern, replacement, x)` #Find first pattern in x and replace with replacement text.
- `toupper(x), tolower(x)` #Converts into upper and lower cases respectively

R FUNCTIONS – OTHER USEFUL FUNCTIONS

- `seq(from , to, by)` # Sequence Generation
- `rep(x, ntimes)` # Replicate Elements of Vectors and Lists
- `cut(x, n)` # Convert Numeric to Factor



R REGULAR EXPRESSIONS

- The package used for regular expressions in R is `regex` (in-built)
  - `\\d` Digit, 0,1,2 ... 9
  - `\\D` Not Digit
  - `\\s` Space
  - `\\S` Not Space
  - `\\w` Word
  - `\\W` Not Word
  - `\\t` Tab
  - `\\n` New line
  - `^` Beginning of the string
  - `$` End of the string
  - `|` Alternation match. e.g. /(e|d)n/ matches "en" and "dn"

R REGULAR EXPRESSIONS

- `.` Any character, except \n or line terminator
- `[ab]` a or b
- `[0-9]` All Digit
- `[A-Z]` All uppercase A to Z letters
- `[a-z]` All lowercase a to z letters
- `[A-za-z]` All Uppercase and lowercase letters
- `i+` i at least one time
- `i*` i zero or more times
- `i?` i zero or 1 time
- `i{n}` i occurs n times in sequence
- `i{n1,n2}` i occurs n1 to n2 times in sequence

R REGULAR EXPRESSIONS

- `i{n,}` i occurs >= n times
- `[:alnum:]` Alphanumeric characters: [:alpha:] and [:digit:]
- `[:alpha:]` Alphabetic characters: [:lower:] and [:upper:]
- `[:blank:]` Blank characters: e.g. space, tab
- `[:space:]` Space characters: tab, newline, space
- `[:upper:]` Upper-case letters in the current locale
- `[:lower:]` Lower-case letters in the current locale

R REGULAR EXPRESSIONS : EXAMPLES

Examples

- grep("a+", c("abc", "def", "cba a", "aa"), perl=TRUE, value=FALSE)
- grepl("a+", c("abc", "def", "cba a", "aa"), perl=TRUE, value=TRUE)

- x <- c("16_24cat", "25_34cat", "35_44catch", "45_54Cat", "55_104fat")
- grep(pattern = "cat", x = x)
- grep("cat$", x, ignore.case = T)
- grepl("cat$", x, ignore.case = T)
- strsplit(x, split = "_")

R REGULAR EXPRESSIONS : EXAMPLES

Examples

- strings <- c("a", "ab", "acb", "accb", "acccb", "accccb")
  - grep("ac*b", strings, value = TRUE)
  - grep("ac+b", strings, value = TRUE)
  - grep("ac?b", strings, value = TRUE)
  - grep("ac{2}b", strings, value = TRUE)
  - grep("ac{2,}b", strings, value = TRUE)
  - grep("ac{2,3}b", strings, value = TRUE)

- strings <- c("abcd", "cdab", "cabd", "c abd")
  - grep("^ab", strings, value = TRUE)

REFERENCES

- R Tutorials
- Beginning R – Dr. Mark Gardener
- Relevant information from public domain

THANK YOU !!!!