R Essentail Functions

1 Mathematical Functions

Function	Syntax	Description	Example Execution	Example Output
sum	`sum(x)`	Calculates the sum of values in `x`.	`sum(c(1, 2, 3, 4))`	`10`
mean	`mean(x)`	Calculates the mean (average) of values in `x`.	`mean(c(1, 2, 3, 4))`	`2.5`
median	`median(x)`	Calculates the median of values in `x`.	`median(c(1, 2, 3, 4, 5))`	`3`
sd	`sd(x)`	Calculates the standard deviation of values in `x`.	`sd(c(1, 2, 3, 4))`	`1.118034`
abs	`abs(x)`	Calculates the absolute value of `x`.	`abs(-3)`	`3`
sqrt	`sqrt(x)`	Calculates the square root of `x`.	`sqrt(16)`	`4`
log	`log(x, base)`	Calculates the logarithm of `x` with the given base.	`log(100, base = 10)`	`2`
exp	`exp(x)`	Calculates the exponential of `x`.	`exp(2)`	`7.389056`
round	`round(x, digits)`	Rounds `x` to the specified number of `digits`.	`round(3.14159, digits = 2)`	`3.14`
ceiling	`ceiling(x)`	Rounds `x` up to the nearest integer.	`ceiling(3.14)`	`4`
floor	`floor(x)`	Rounds `x` down to the nearest integer.	`floor(3.14)`	`3`
min	`min(x)`	Finds the minimum value in `x`.	`min(c(2, 4, 1, 6))`	`1`
max	`max(x)`	Finds the maximum value in `x`.	`max(c(2, 4, 1, 6))`	`6`
range	`range(x)`	Finds the range (min and max) of `x`.	`range(c(2, 4, 1, 6))`	`1, 6`
IQR	`IQR(x)`	Calculates the interquartile range of `x`.	`IQR(c(1, 2, 3, 4, 5))`	`1.5`
summary	`summary(object)`	Provides a concise summary of object `x`	`summary(c(1, 2, 3, 4, 5))`	Min. 1st Qu. Median Mean 3rd Qu. Max.

2 String Functions

Function	Syntax	Description	Example Execution	Example Output
paste	`paste(..., sep = " ")`	Combines multiple strings or values.	`paste("Hello", "World")`	`"Hello World"`
paste0	`paste0(...)`	Combines multiple strings or values without space.	`paste0("Hello", "World")`	`"HelloWorld"`
toupper	`toupper(x)`	Converts characters in `x` to uppercase.	`toupper("hello")`	`"HELLO"`
tolower	`tolower(x)`	Converts characters in `x` to lowercase.	`tolower("Hello")`	`"hello"`
startsWith	`startsWith(x, prefix)`	Checks if the string `x` starts with the `prefix`.	`startsWith("Hello World", "Hello")`	`TRUE`

3 Data Structures

Function	Syntax	Description	Example Execution	Example Output
c	`c(...)`	Combines values into a vector or list.	`c(1, 2, 3)`	`1, 2, 3`
data.frame	`data.frame(...)`	Creates a data frame from vectors or lists.	`data.frame(Name=c("Alice", "Bob"), Age=c(25, 30))`	A data frame object
matrix	`matrix(data, nrow, ncol)`	Creates a matrix from data with specified rows and columns.	`matrix(1:9, nrow = 3, ncol = 3)`	A matrix object
str	`str(object)`	Displays the internal structure of an R object.	`str(data.frame(x = 1:3, y = c("A", "B", "C")))`	‘data.frame’: 3 obs. of 2 variables.

4 Data Manipulation Functions

Function	Syntax	Description	Example Execution	Example Output
length	`length(x)`	Returns the length of vector `x`.	`length(c(1, 2, 3, 4))`	`4`
sort	`sort(x)`	Sorts the elements of vector `x` in ascending order.	`sort(c(3, 1, 4, 1, 5))`	`1, 1, 3, 4, 5`
order	`order(x)`	Returns the permutation needed to sort `x`.	`order(c(3, 1, 4, 1, 5))`	`2, 4, 1, 3, 5`
rank	`rank(x)`	Computes the ranks of elements in `x`.	`rank(c(3, 1, 4, 1, 5))`	`4, 1, 5, 2, 3`

Function	Syntax	Description	Example Execution	Example Output
unique	`unique(x)`	Returns the unique values in `x`.	`unique(c(1, 2, 2, 3, 3))`	`1, 2, 3`
cbind	`cbind(...)`	Combines vectors or data frames by column binding.	`cbind(dataframe1, dataframe2)`	Combined data frame
rbind	`rbind(...)`	Combines vectors or data frames by row binding.	`rbind(dataframe1, dataframe2)`	Combined data frame
rownames	`rownames(x)`	Retrieves or sets the row names of a matrix or data frame `x`.	`rownames(dataframe)`	Character vector of row names
colnames	`colnames(x)`	Retrieves or sets the column names of a matrix or data frame `x`.	`colnames(dataframe)`	Character vector of column names

5 Set Functions

Function	Syntax	Description	Example Execution	Example Output
union	`union(x, y)`	Returns the union of sets `x` and `y`.	`union(c(1, 2, 3), c(3, 4, 5))`	`1, 2, 3, 4, 5`
intersect	`intersect(x, y)`	Returns the intersection of sets `x` and `y`.	`intersect(c(1, 2, 3), c(3, 4, 5))`	`3`
setdiff	`setdiff(x, y)`	Returns the set difference of sets `x` and `y`.	`setdiff(c(1, 2, 3), c(3, 4, 5))`	`1, 2`
setequal	`setequal(x, y)`	Checks if sets `x` and `y` are equal.	`setequal(c(1, 2, 3), c(3, 2, 1))`	`TRUE`

6 Apply and Sweep Functions

Function	Syntax	Description	Example Execution	Example Output
apply	`apply(X, MARGIN, FUN, ...)`	Applies a function `FUN` to rows or columns of a matrix `X`.	`apply(matrix(1:9, nrow = 3), 1, sum)`	`6, 15, 24`
lapply	`lapply(X, FUN, ...)`	Applies a function `FUN` to each element of a list `X`.	`lapply(list(1, 2, 3), function(x) x * 2)`	`2, 4, 6`
sapply	`sapply(X, FUN, ...)`	Applies a function `FUN` to each element of a list `X` and simplifies the result.	`sapply(list(1, 2, 3), function(x) x * 2)`	`2, 4, 6`
mapply	`mapply(FUN, ...)`	Applies a function `FUN` to multiple lists or vectors in parallel.	`mapply(function(x, y) x + y, c(1, 2, 3), c(10, 20, 30))`	`11, 22, 33`
tapply	`tapply(X, INDEX, FUN, ...)`	Applies a function `FUN` to subsets of `X` specified by `INDEX`.	`tapply(1:10, c(1, 2, 1, 2, 1, 2, 1, 2, 1, 2), sum)`	`15, 40`
sweep	`sweep(X, MARGIN, STATS, FUN)`	Sweeps through an array `X` and applies a function `FUN` to each element along the specified `MARGIN` while using `STATS` as the statistics array.		

7 Logical Functions

Function	Syntax	Description	Example Execution	Example Output
is.element	`is.element(x, y)`	Checks if elements of `x` are in set `y`.	`is.element(1, c(1, 2, 3))`	`TRUE`
ifelse	`ifelse(test, yes, no)`	Returns `yes` if `test` is TRUE, `no` otherwise.	`ifelse(2 > 1, "Yes", "No")`	`"Yes"`
is.numeric	`is.numeric(x)`	Checks if `x` is of numeric type.	`is.numeric(123)`	`TRUE`
is.integer	`is.integer(x)`	Checks if `x` is of integer type.	`is.integer(123)`	`TRUE`
is.character	`is.character(x)`	Checks if `x` is of character type.	`is.character("Hello")`	`TRUE`
is.logical	`is.logical(x)`	Checks if `x` is of logical type.	`is.logical(TRUE)`	`TRUE`
is.factor	`is.factor(x)`	Checks if `x` is a factor.	`is.factor(factor("A"))`	`TRUE`
is.matrix	`is.matrix(x)`	Checks if `x` is a matrix.	`is.matrix(matrix(1:6, nrow = 2))`	`TRUE`
is.data.frame	`is.data.frame(x)`	Checks if `x` is a data frame.	`is.data.frame(data.frame(a = 1:3))`	`TRUE`
is.list	`is.list(x)`	Checks if `x` is a list.	`is.list(list(1, 2, 3))`	`TRUE`
is.vector	`is.vector(x)`	Checks if `x` is a vector.	`is.vector(c(1, 2, 3))`	`TRUE`
is.null	`is.null(x)`	Checks if `x` is `NULL`.	`is.null(NULL)`	`TRUE`
is.na	`is.na(x)`	Checks for missing (NA) values in `x`.	`is.na(c(1, NA, 3))`	`TRUE, TRUE, FALSE`