R Statistical computing

CONCEPTS

Almost everything is a vector; a single number is a vector of length 1. Operations work on whole vectors at once.

Vector one type
↓ `names + dims`
List mixed types
↓ `equal-length columns`
Data frame table of columns
↓ `model / plot`
Result print or write

RUNNING R

`R`
Start an interactive session

`Rscript <file.R> [args]`
Run a script non-interactively

`Rscript -e '<expr>'`
Evaluate an expression and exit

`R --vanilla`
Ignore profile, site and saved state

`commandArgs(trailingOnly = TRUE)`
Script arguments as a character vector

`q()`
Quit the session

`source("<file.R>")`
Run a script inside a session

HELP AND PACKAGES

`?<topic>`
Open the help page for a function

`??<term>`
Search all help pages

`install.packages("<pkg>")`
Install from CRAN

`library(<pkg>)`
Attach an installed package

`<pkg>::<fn>`
Call a function without attaching

VECTORS

`c(1, 2, 3)`
Combine values into a vector

VECTORS (cont.)

`1:10`
Integer sequence

`seq(0, 1, by = 0.25)`
Sequence with a step

`length(<x>)`
Number of elements

`x[3]`
Third element

`x[c(1, 3)]`
Several elements by position

`x[x > 5]`
Elements matching a condition

`which(x > 5)`
Positions where TRUE

`%in%`
Membership test: x %in% y

`sort(x) / order(x)`
Sorted values / sorting positions

`table(x)`
Count each value

`is.na(x)`
Test for missing values

`ifelse(<test>, <yes>, <no>)`
Vectorised if/else

`paste("a", "b", sep = "-")`
Join strings

DATA STRUCTURES

`list(a = 1, b = "x")`
List of mixed types

`l$a / l[["a"]]`
Element of a list

`factor(x, levels = c("lo", "hi"))`
Categorical vector

`class(x) / typeof(x)`
Class / internal type

`str(x)`
Compact structure of any object

`as.numeric(x)`
Convert type (also as.character, as.integer)

DATA FRAMES

`data.frame(a = 1:3, b = c("x", "y", "z"))`
Build a data frame

`head(df, <n>)`
First rows

`nrow(df) / ncol(df)`
Row / column count

`names(df)`
Column names

`summary(df)`
Per-column statistics

`df$<col>`
One column as a vector

`df[<rows>, <cols>]`
Rows and columns by index or name

`df[df$<col> > 5, ]`
Filter rows

`df[order(df$<col>), ]`
Sort rows

`df$<new> ← df$a * 2`
Add a column

CONTROL AND FUNCTIONS

`f ← function(x, y = 2) { x + y }`
Define a function

APPLY FAMILY

`sapply(x, f)`
Apply, simplify to vector or matrix

`lapply(x, f)`
Apply, always return a list

IMPORT AND EXPORT

`read.csv("<file>")`
Read a CSV into a data frame

`write.csv(df, "<file>", row.names = FALSE)`
Write a data frame as CSV

PLOTTING

`plot(x, y)`
Scatter plot