Base R Cheat Sheet

by Todin via cheatography.com/174736/cs/36706/

Language itself

R is a free software environment for statistical computing and graphics.
It's a compiled language.
R is a dynamically-typed,that means the type of a variable is determined at runtime, and that variables do not have to be explicitly declared with a specific type.
R is weakly-typed, which means that it will automatically convert between types when necessary, without the need for explicit type casting.
R is a garbage-collected language. This means the R runtime system automatically manages the memory used by the program.
R is both a functional an object oriented language but it doesn't support inheritance, polymorphism, and encapsulation.
In R, functions call by value. This means that when a function is called, the values of the arguments passed to the function are passed to the function's local scope, and any changes made to the arguments within the function do not affect the original values of the arguments outside the function.

Language environment

To code in R you can install the extension in your IDE like VSC or install its own IDE *R Studio* :
R for Windows
R for Linux
Then to run it you can run it with the IDE if you don't have one you can use R Studio server, which is a web-based version of R Studio that can be accessed through a web browser. The command for R Studio is `Rscript myscript.R` .
When installed R Studio it would have a minimal impact impact on the local machine but if you runs large R scripts or using R for data analysis and machine learning tasks, which require a lot of memory,it may have a bigger impact on the local machine. But in the end it does not modify something in your machine that would block you in it's use.

Library

Code

`instal l.p ack age s(' dplyr')` : Download and install a package from CRAN.
`librar y(d plyr')` : Load the package into the file.
`dplyr: :select` : Use a function from the package

Commonly used library

stringr for string manipulation
dplyr for data frame manipulation
ggplot2 for plotting graphs
lubridate for date

To go further

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes:
-An effective data handling and storage facility,
-A suite of operators for calculations on arrays, in particular matrices,
-A large, coherent, integrated collection of intermediate tools for data analysis,
-Graphical facilities for data analysis and display either on-screen or on hardcopy, and
-A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.
The people who mainly use R are data scientist / engineer

Syntax and data structures

Variable Assignment

`a <- 'bonjour'` : *string*
`a <- 6` : *integer*

Loop

For loop :
`for (variable in sequence){ Do something }`
While loop :
`while (condi tion){ Do something }`

Condition

`if (condi tion){ Do something } else { Do something`
`a == b` : Are equal
`a > b` : Greater than
`a >= b` : Greater than or equal to
`is.na(a)` : Is missing
`a != b` : Not equal
`a < b` : Less than
`a <= b` : Less than or equal to
`is.null(a)` : Is null

Functions

`functi on_name <- function (var){ Do something retu`
`) }`

Maths Functions

`log(x)` : Natural log / `sum(x)` : Sum
`exp(x)` : Exponential / `mean(x)` : Mean
`max(x)` : Largest element / `median(x)` : Median
`min(x)` : Smallest element / `quanti le(x)` : Percentage quantiles
`round(x, n)` : Round to n decimal places
`rank(x)` : Rank of elements
`signif(x, n)` : Round to n significant figures
`var(x)` : The variance / `cor(x, y)` : Correlation
`sd(x)` : The standard deviation

Syntax and data structures (cont)

Lists

`l <- list(x = 1:5, y = c('a', 'b'))` : A list is collection of elements which can be of different types.

`l[[2]]` : second element of l

`l[1]` : new list with only the first element

`l$x` : element named x

`l['y']` : new list with only element named y

String

`paste(x, y, sep = ' ')` : Join multiple vectors together.

`paste(x, collapse = ' ')` : Join elements of a vector together.

`grep(p attern, x)` : Find regular expression matches in x.

`gsub(p attern, replace, x)` : Replace matches in x with a string.

`toupper(x)` : Convert to uppercase.

`tolower(x)` : Convert to lowercase.

`nchar(x)` : Number of characters in a string.

Matrixes

`m <- matrix(x, nrow = 3, ncol = 3)` : Create a matrix from x.

`df[ , 2]` : Select a column .

`df[2, ]` : Select a row.

`df[2, 2]` : Select an element.

Data Frames

`df <- data.f rame(x = 1:3, y = c('a', 'b', 'c'))` : A special case of a list where all elements are the same length.

`View(df)` : See the full data frame.

`head(df)` :See the first 6 rows.

`nrow(df)` : Number of rows.

`ncol(df)` : Number of columns.

`dim(df)` : Number of columns and rows.

As the same sub settings as matrixes and lists

Ploting

`plot(x)` : plot values of x in order

`plot(x, y)` : plot values of x against y

Reading and writing data

`df <- read.t abl e(' fil e.txt')` / `df <- read.c sv( - 'fi le.c sv')` / Read a file text or a csv

`write.t ab le(df, 'file.t xt')` / `write.c sv(df, 'fil e.c sv')` / Write a file text or a csv