

CHEAT SHEET FOR R

By
Nanditha T (F17095)
Sanjana S (F17109)
Vivin Pearl Kishore (F17119)

Util functions

getwd()	gets the working directory
setwd("c://file/path")	sets the working directory
ls()	list all the variables
rm(var_name)	removes variable name
str(variable name)	displays the structure
help.start()	opens help
install.packages("package_name")	installs packages
library("package_name")	makes the content available to use
detach("package_name")	detaches the package
history()	displays history

Data Structures

Vectors	d=c(3,4,5)
Arrays	2D = array(1:24, dim = c(6,4))
Matrices	mat = matrix(1:12, nrow=4, ncol=3)
Lists	list_data <- list("Red", "Green", c(21,32,11), TRUE, 5, 3)
Dataframe	df = data.frame(subjectID=1:5,gender=c("M","F","M","M","F"),score=c(8,3,6,5,5))

Vector

num = c(1,2,3,4,5,6)	numeric vector
chr = c("aaa","bbb")	character vector
log = c(TRUE,TRUE,FALSE)	logical vector
which.min(vec)/which.max(vec)	position of the min/max value
rep(1:5,times=3)	Replicate elements of vector

Arrays

1D = array(1:24)	1-D array
2D=array(1:24,dim=c(6,4))	2-D array
3D=array(1:24,dim=c(4,3,2))	3-D array

Matrix Functions

t(m)	transpose
m %*% n	matrix multiplication
solve(m,n)	find x in m*x = n
det(m)	determinant
m*n	dot product
rbind/cbind(mat1,mat2)	row/column bind

Data Frames

df = data.frame(subjectID=1:5,gender=c("M","F","M","M","F"),score=c(8,3,6,5,5))	Created data frames in R
---	--------------------------

Data Frames (cont)

fw = read.csv(file.choose())	Importing data by choosing a file
grass = read.csv('C:/path/sample.csv')	Importing data by specifying paths
view(df)	opens editor
rbind(a_data_frame, another_data_frame)	Bind rows/columns of frames
merge(frame1, frame2, by = "x")	Merge 2 data frames

Descriptive Statistics

rowMeans(data[])	row mean
rowSums(data[])	row sum
colMeans(data[])	column mean
colSums(data[])	column sum

Data type Conversion

Use is.foo to test for data type foo. Returns TRUE or FALSE

Use as.foo to explicitly convert it

is.numeric(), is.character(), is.vector(), is.matrix(), is.data.frame()
as.numeric(), as.character(), as.vector(), as.matrix(), as.data.frame()

Creating a Function

```
function_name <- function(arg_1, arg_2, ...) {  
  Function body  
}
```

Functions are followed by paranthesis

String functions

toString(x)	produce a character string
noquote(x)	print character strings without quotes



By Sanjana
cheatography.com/sanjana/

Not published yet.
Last updated 30th August, 2018.
Page 1 of 3.

Sponsored by [CrosswordCheats.com](https://crosswordcheats.com)
Learn to solve cryptic crosswords!
[http://crosswordcheats.com](https://crosswordcheats.com)

String functions (cont)

<code>sprintf()</code>	returns a character vector containing a formatted combination of text and variable values
<code>cat()</code>	converts into strings and concatenates
<code>toupper()</code> / <code>tolower()</code>	converts text to uppercase/lowercase
<code>substr(x,first,-last)</code>	extracts parts of a string
<code>strsplit(x, split, fixed = FALSE, perl = FALSE, useBytes = FALSE)</code>	split elements of a string into substrings
<code>paste(..., sep = " ", collapse = NULL)</code>	concatenate strings

Factor functions

<code>factor()</code>	it is used to encode a vector as a factor (the terms 'category' and 'enumerated type' are also used for factors)
<code>levels()</code>	it provides access to the levels attribute of a variable
<code>nlevels()</code>	Return the number of levels which its argument has.
<code>relevel()</code>	The levels of a factor are re-ordered so that the level specified by ref is first and the others are moved down
<code>unique()</code>	it returns a vector, data frame or array like x but with duplicate elements/rows removed.
<code>droplevels()</code>	The function <code>droplevels</code> is used to drop unused levels from a factor or, more commonly, from factors in a data frame

Factor functions (cont)

<code>cut()</code>	<code>cut</code> divides the range of x into intervals and codes the values in x according to which interval they fall
--------------------	--

Date Time functions

<code>Sys.time()</code>	returns today's date
<code>date()</code>	returns current date and time
<code>as.POSIXlt()</code>	convert an object to one of the two classes used to represent date/times
<code>as.Date()</code>	convert character data to dates
<code>strptime()</code>	converts character vectors to class "POSIXlt": its input x is first converted by <code>as.character</code>
<code>strftime()</code>	a wrapper for <code>format.POSIXlt</code> , and it and <code>format.POSIXct</code> first convert to class "POSIXlt" by calling <code>as.POSIXlt</code>

Flow control functions

<code>if(condition){ //execute when condition is true }</code>	
<code>if(condition){ //execute when condition is true } else { //execute when condition is false }</code>	
<code>if(condition 1) { // Executes when the condition 1 is true } else if(condition 2) { // Executes when the condition 2 is true. } else if(condition 3) { // Executes when the condition 3 is true } else { // executes when none of the above condition is true }</code>	
<code>ifelse(condition, x, y)</code>	
<code>switch(expression, case1, case2, case3....)</code>	

Loop functions

<code>while (condition){ Do something }</code>	
<code>for (variable in sequence){ Do something }</code>	
<code>apply()</code> , <code>lapply()</code> , <code>sapply()</code>	
A loop statement allows us to execute a statement or group of statements multiple times based on the condition	

File format functions

<code>read.csv()</code>	To read the data
<code>read.table()</code>	To read the table contents
<code>read.xlsx2()</code>	To read data from excel sheet

Data summary functions

<code>summary()</code>	returns descriptive statistics of data
<code>str()</code>	structure of the variable
<code>describe()</code>	determines the type of a single variable and prints a concise statistical summary
<code>class()</code>	a simple generic function mechanism which can be used for an object-oriented style of programming
<code>dim()</code>	Dimension
<code>head()</code>	Returns the first or last parts of a vector, matrix, table, data frame or function.
<code>names()</code>	Functions to get or set the names of an object.
<code>View()</code>	Invoke a spreadsheet-style data viewer on a matrix-like R object.
<code>subset()</code>	Return subsets of vectors, matrices or data frames which meet conditions.

Visualization functions

<code>par(mfrow=c(2,2))</code>	create a matrix of n rows
<code>barplot()</code>	Relationship between a numerical and a categorical variable
<code>pie()</code>	piecharts
<code>mosaicplot()</code>	Plots a mosaic on the current graphics device
<code>hist()</code>	Histogram
<code>plot()</code>	simple scatter plots
<code>plot(density())</code>	Density plots. non-parametric way to estimate the probability density function of a random variable
<code>pairs()</code>	A matrix of scatterplots is produced
<code>matplot()</code>	Plot the columns of one matrix against the columns of another.
<code>boxplot()</code>	Distribution
<code>qqnorm()</code>	produces quantile-quantile plot
<code>qplot()</code>	produces quantile-quantile plot
<code>ggplot(mydata1, aes(x = 1, fill = subject)) + geom_bar()</code>	Initializes a ggplot object

Probability Distributions

	Random Variables	Density Function	Cumulative Distribution	Quartile
Normal	<code>rnorm</code>	<code>dnorm</code>	<code>pnorm</code>	<code>qnorm</code>
Poisson	<code>rpois</code>	<code>dpois</code>	<code>ppois</code>	<code>qpois</code>
Binomial	<code>rbinom</code>	<code>dbinom</code>	<code>pbinom</code>	<code>qbinom</code>
Uniform	<code>runif</code>	<code>dunif</code>	<code>punif</code>	<code>qunif</code>
Exponential	<code>rexp</code>	<code>dexp</code>	<code>pexp</code>	<code>qexp</code>

Central tendency and Dispersion

<code>mean()</code>	find mean
<code>median()</code>	find median
<code>range()</code>	find range
<code>sd()</code>	find standard deviation
<code>var()</code>	find variance
<code>cor()</code>	find correlation

Hypothesis Testing

<code>t.test(data)</code>	1 sample t-test
<code>t.test(data1,data2)</code>	2 sample t-test
<code>t.test(pre,post,paired=TRUE)</code>	paired sample t-test
<code>wilcox.test(data)</code>	Wilcox test
<code>cor.test(data1,data2)</code>	Correlation test
<code>chisq.test(data)</code>	Chi square test
<code>shapiro.test(data)</code>	Shapiro test
<code>aov()</code>	ANOVA

Algorithms - statistics

<code>summary(lm(y ~ x1 + x2 + x3, data=mydata))</code>	multiple regression
<code>summary(glm(y ~ x1 + x2 + x3, family="", data=mydata))</code>	classification
<code>cluster = kmeans(data)</code>	clustering

