Cheatography

R Cheat Sheet

by Sanjana via cheatography.com/67198/cs/16836/

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, ...) {
      Fun ction body
}
```

Functions are followed by paranthesis

String functions

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

String functions (cont)

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

Factor functions

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

Factor functions (cont)

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

Date Time functions

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

Flow control functions

if(condition){ //execute when condition is true}

if(condition){//execute when condition is true} else(){//execute when condition is false}

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}

ifelse(condition, x, y)

switch(expression, case1, case2, case3....)

Loop functions

while (condition){ Do something }

for (variable in sequence){ Do something }

apply(), lapply(), sapply()

A loop statement allows us to execute a statement or group of statements multiple times based on the condition

File format functions

read.csv()	To read the data
read.table()	To read the table contents
read.xlsx2()	To read data from excel sheet

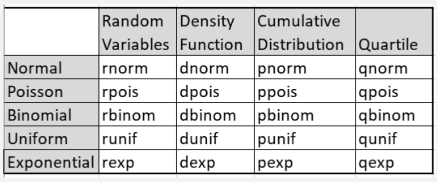
Data summary functions

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

Visualization functions

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

Probability Distributions



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

Central tendency and Dispersion

mean()	find mean
median()	find median
range()	find range
sd()	find standard deviation
var()	find variance
cor()	find correlation

Hypothesis Testing

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

Algorithms - statistics

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