Cheatsheet R – I

Basics

`x <- y`	Asign to variable
`c(x,y,z)`	concatenate vector
`x^2,sqrt(x)`	x squared, square root x
`abs(x)`	absolute value of x
`?`*f*	call help for function *f*
`args(`*f*`)`	show arguments for function *f*

Workspaces and Files

`getwd()`	get current working directory
`setwd()`	set current working directory
`ls()`	list objects in workingspace
`list.files(),dir()`	list all files
`dir.create("name")`	create dir "name" in pwd
`file.create("file.R")`	create file.R
`file.exist("file.R")`	check existence of file.R
`file.info("file.R")`	information about file.R
`file.info("file.R")$size`	specific info (size)
`file.rename("file.R","file2.R")`	rename to file2.R
`file.copy("file.R","file2.R")`	copy
`file.path("file.R")`	relative path to file.R
`file.path("dir1",dir2)`	construct path dir1/dir2
`dir.create("dir1")`	create dir1
`dir.create(file.path("d1","d2",recursive=T))`	created nested d1/d2

Sequences

`1:20`	1,2,3,...20
`pi:20`	3.14,4.14,5.14,...19.14
`15:1`	15,14,13,...1
`seq(1,10,by=0.5)`	1,1.5,2,...10
`seq(2,10,length=5)`	2,4,6,8,10
`length(x)`	length N of vector x
`1:length(x)`	1,2,3,...N
`seq(along.with=x)`	1,2,3,...N
`seq_along(x)`	1,2,3,...N
`rep(0,times=5)`	(0 0 0 0 0)
`rep(c(0,1),times=2)`	(0 1 0 1)
`rep(c(0,1),each=2)`	(0 0 1 1)

Vectors: logical and text

```
> x<-c(1,2,3)        > str<-c("Hello","world")
```

`<;>;<=;>=;==;!=`	logical operators
`|;&;!`	OR,AND,NOT
`||;&&`	evaluates just the 1st element
`x>1`	FALSE TRUE TRUE
`5>8||6!=8&&4>3.9`	TRUE (AND before OR)
`isTRUE(6>4)`	TRUE
`xor(TRUE,FALSE)`	TRUE (exclusive or)
`which(x>2)`	2 3
`any(x>2)`	TRUE
`all(x>2)`	FALSE
`paste(str,collapse=" ")`	"Hello world"
`str<-c(str,"!")`	"Hello","world","!"
`paste("Hello","!")`	"Hello !"
`paste(x,str,sep="")`	"1Hello" "2world","3!"
`paste(str,1:3,sep="-")`	"Hello-1" "world-2" "!-3"

Missing valuesss

```
> x<-c(1,NA,3)
```

`x*2`	2 NA 6
`is.na(x)`	FALSE TRUE FALSE
`sum(is.na(x))`	1
`rnorm(n,mean-0,sd=1)`	normal distribution
`sample(x,10,replace=T)`	sample 10 items
`0/0`	NaN (Not a Number)
`Inf-Inf`	NaN (Not a Number)

Subsets

```
> x<-c(7,NA,13)                > y<-c(17,19,23)
> z<-c("foo","bar","qq")
```

`x[2:3]`	NA 13
`x[is.na(x)]`	NA
`x[!is.na(x)]`	7 13
`x[y>18]`	NA 13
`x[!is.na(x) & y>18]`	13
`x[c(1,3)]`	7 13
`x[c(-2,-3)]`	7
`x[10]`	NA
`names(y)<-z`	foo|17 bar|19 qq|23
`y["bar"]`	bar|19
`a<-c(foo=17,bar=19,qq=23)`	foo|17 bar|19 qq|23
`identical(a,y)`	TRUE
`M[,11:17]`	subset matrix columns 11:17
`range(y)`	17 23
`unique(c(1,1,2,2,3))`	1 2 3

Matrices

```
> x<-1:20             > y<-c("A","B","C","D")
```

`dim(x)<-c(4,5)`	create 4x5 matrix
`dim(x)`	4 5
`attributes(x)`	$dim 4 5
`class(x)`	"matrix"
`str(M)`	display internal structure
`matrix(1:20,nrow=4,ncol=5)`	identical to x
`cbind(x,y)`	add column (all characters)
`M<-data.frame(x,y)`	add col (mixed int&char)
`colnames(M)<-`	name columns
`rownames(M)<-`	name rows
`colMeans(M[1:4])`	column 1-4 means

Functions

`function(arg1,arg2)expr`	basic syntax
`function(x,y)x+y`	sum var x and y
`function(x,y=1)x+y`	default value for y
`args(function)`	show function arguments
`f<-function(fun,x)fun(x)`	pass function as arg
`f(function(x)x+1,6)`	anonymous function
`(...)`	one or more R objects
`function(...){paste(">",...,"<")}`	>...<
`paste(...,sep=" ")`	args after ... must be default
`list1[["arg"]]`	extract named argument
`"%t%"<-function(x,y)x*y`	binary operator
`4 %t% 6`	24
`lapply(list,fun)`	applies function to list
`sapply(list,fun)`	returns vector/matrix
`lapply(x,function(arg)expr)`	apply anon fun
`vapply(x,f,numeric(1))`	expect elements num of length 1
`table(x,y)`	counts at each combi
`tapply()`	split data into groups
`tapply(X,INDEX,fun)`	fun(X) for each INDEX

Simulation

`sample(1:6,4,replace=T)`	roll 4 dices
`sample(0:1,10,rep=T,pro=c(0.4,0.6))`	false coin
`rbinom(n,size,prob)`	binomial distribution
`rnorm(n,mean=0,sd=1)`	normal distribution
`rpois(n,lambda)`	discrete poisson
`replicate(n,expr)`	perform expr n times
`hist(x)`	plot histogram

Created by Raoul Grouls, 2017
verbatim of swirl package `http://swirlstats.com/`