R Cheatsheet

Notes:

1. This is by no means a comprehensive list, as a large number of useful functions have been left out, and not all options for the functions listed have been given. This list is purely intented to give a place to begin, as I remember how frustrating it was to not even know what to start looking for!
2. Typing ?*functionname* at the command line brings up a help window for the function name listed.
3. Assume in the examples that all vectors and matrices (v*i*'s and mat*i*'s) have been created.

Command		**Example**	**Result**
Operators			
General			
<-	Assignment operator (suggested)	ans1 <- 1	1
=	Assignment operator	ans2 = 1+1	2
#	Comment	#This is a comment	
Mathematical			
+	Addition	2.5+ans3	5.5
-	Subtraction	ans3-2.5	0.5
*	Scalar multiplication	2*3	6
/	Division operator	6/2	3
^	Exponentiation	2^3	8
Logical/Relational			
==	Equals	ans3==3	TRUE
!=	Not Equal	ans3!=3	FALSE
>	Greater Than	ans3>3	FALSE
>=	Greater Than or Equal To	ans3>=3	TRUE
<	Less Than	ans3<3	FALSE
<=	Less Than or Equal To	ans3<=3	TRUE
||	Or	ans1==2 || ans2==2	TRUE
|	Or (use with vectors and matrices)	v2[v1==3 | v1==4]	{3,5}
&&	And	ans1==2 && ans2==2	FALSE
&	And (use with vectors and matrices)	v2[v1==3 & v1==4]	{NA}
%*%	Matrix multiplication	mat1%*%mat1	
Functions			
sqrt	Square root	sqrt(16)	4
exp	Exponentiation	exp(1)	2.718282
log	Natural log	log(2.718282)	1
sum	Sum	sum(2,3,4)	9
prod	Product	prod(2,3,4)	24
ceiling	Smallest integer ≥number	ceiling(2.1)	2
floor	Integer part of a number	floor(2.1)	2
abs	Absolute value	abs(-0.2)	0.2
sin	Sine	sin(pi/2)	1
cos	Cosine	cos(pi)	-1
tan	Tangent	tan(pi/4)	1

table	Calculate frequency counts of a vector	table(v4)	1 3 5 [3 3 3]

Vector/Matrix Functions

Vector creation functions

c	Concatenate	v1 <- c(2,3,4)	2,3,4
		v2 <- c(1,3,5)	1,3,5
seq	Sequence	v3 <- seq(from=2, to=10, by=2)	2,4,6,8,10
		seq(from=2, to=4, length=5)	2.0,2.5,3.0,3.5,4.0
:	Integer sequence	2:10	2,3,4,5,6,7,8,9,10
rep	Repeat	v4 <- rep(v2, 3)	1,3,5,1,3,5,1,3,5

Combining vectors to create matrices

cbind	Column bind	mat1 <- cbind(v1,v2)	2 1 3 3 4 5
rbind	Row bind	mat2 <- rbind(v1,v2)	2 3 4 1 3 5
matrix	Create matrix	matrix(0, nrow=2, ncol=3)	0 0 0 0 0 0
as.data.frame	Create dataset from matrix	A<-as.data.frame(mat1)	2 1 3 3 4 5

Utility functions

[ ]	Subscript operator (Vectors)	answer <- v1[3]	4
[,]	Subscript operator (2D)	answer <- mat1[1,1]	2
		answer <- mat1[,1]	2,1
		answer <- mat1[1,]	2,3,4
		answer <- mat1[-1,]	3 3 4 5
[,,]	Subscript operator (3D)	answer <- arr1[2,4,3]	114
length	Length of vector	length(v4)	9
sort	Sort a vector	sort(v4)	1,1,1,3,3,3,5,5,5
order	Indices to sort a vector	order(v4)	1,4,7,2,5,8,3,6,9
	Useful for sorting matrices	v4[v4.order]	1,1,1,3,3,3,5,5,5
rev	Reverse order of vector	rev(v3)	10,8,6,4,2
unique	Lists unique objects in vector or matrix	unique(v4)	1,3,5

Statistics

max	Maximum of vector or matrix	max(v4)	5
min	Minimum of vector or matrix	min(mat1)	1
pmax	Parallel maximum of vectors/matrices	pmax(v1,v2)	2,3,5
pmin	Parallel minimum of vectors/matrices	pmin(v1,v2)	1,3,4
mean	Calculates mean of vector or matrix	mean(mat1)	3
median	Calculates median of vector or matrix	median(v3)	6
quantile	Calculate quantiles requested	quantile(1:5,probs=c(0,0.25,0.5,0.75,1))	1,2,3,4,5
var	Calculate variance of vector	var(v3)	10
cor	Calculates correlation of 2 vectors	cor(v4,1:9)	0.3162

Distributions

d<dist>(x,<parameters>) density at x	dunif(1.4,min=1,max=3)	0.5
p<dist>(x,<parameters>) CDF evaluated at x	pnorm(1.645,0,1)	0.95
q<dist>(x,<parameters>) inverse cdf	qnorm(0.95,0,1)	1.645
r<dist>(x,<parameters>) generates n random numbers	rbeta(3, shape1=0.5, shape2=1)	0.175083,0.668609,0.009384

<dist>	Distribution	Parameters	Defaults
beta	Beta	shape1, shape2	-,-
cauchy	Cauchy	location, scale	0,1
chisq	Chi-square	df	-
exp	Exponential	-	-
f	F	df1, df2	-,-
gamma	Gamma	shape	-
lnorm	Log-normal	mean, sd (of log)	0,1
Logis	Logistic	location, scale	0,1
norm	Normal	mean, sd	0,1
stab	Stable	index, skew	-,0
t	Student's t	df	-
unif	Uniform	min, max	0,1

For Loops

```
for(i in <vector>){ do stuff }
```

```
## calculate 5! using a for loop
ans <- 1
for(i in 1:5){  ans <- ans*i }
ans                                  120
```

if/else

```
if(<logical value>) { do stuff }
else { do other stuff }
```

```
## Threshold ans at 100
if(ans > 100){  ans2 <- 100}
else{  ans2 <- ans}
ans2                                 100
```

Functions

```
func.name <- function(arg1, arg2, ...){ do stuff; return(ans)}
```

```
## Function to do factorial
my.factorial <- function(x){
  if(!is.integer(x))
    stop("x must be an integer")
  ans <- 1
for(i in 1:x){  ans <- ans*i }
  return(ans)
}
my.factorial(5)                      120
```

Useful links:
http://cran.r-project.org/doc/contrib/usingR-2.pdf
http://www.isds.duke.edu/computing/S/Snotes/Splus.html
http://lib.stat.cmu.edu/S/cheatsheet