Intro to R Cheat Sheet

Basics

`help(fn) or ?fn`	Get help on function “fn”
`sqrt(x)`	Takes the square root of x where x is a numeric.
`log(x)`	Takes the natural log of x. Optional argument base=n will take the log
`c(...)`	Combine values into a vector
`seq(from=2, to=10, by=2)`	Outputs a vector from 2 to 10 by 2’s. Same as c(2,4,6,8,10)
`length(x)`	Returns the number of elements in x (a vector)
`ls()`	Lists the current objects in the environment
`rm(x, y, z)`	Removes the objects x, y, and z from the environment
`class(x)`	See what kind of object “x” is
`toupper(x), tolower(x)`	Converts string x to all uppercase or lowercase
`sum(x), mean(x), sd(x)`	Computes the sum, mean, or standard deviation of all the elements in x
`data()`	Lists built-in datasets
`data(cars)`	Loads the built-in cars dataset
`head(d), tail(d)`	Prints the first or last few lines of a data.frame
`summary(d)`	Summarizes a data frame (or many other kinds of objects)
`dim(d)`	Returns a vector of two elements: the number of rows and the number of columns of a data.frame
`mydf$myvar`	Access the “myvar” variable in the “mydf” data.frame
`subset(d, somevar>1 & othervar<0, select=c(myvar1, myvar2)`	Returns a subset of data.frame “d” returning only “myvar1” and “myvar2” where “somevar” is greater than 1
`with(d, ...)`	Temporarily attach data.frame “d” and do “…” with the variables attached from d. E.g.: `with(d, mean(somevar))` is the same as mean(d$somevar).
`x[1:5]`	Prints elements 1 through 5 of vector x
`d[2:4, c(3,5,7)]`	Prints the 2nd through 4th row and 3rd, 5th, and 7th column of data.frame “d”
`d[ , 1:3]`	Prints *all* rows and the first three columns of data.frame “d”
`d[c(1:3,5), ]`	Prints rows 1, 2, 3, and 5, and *all* columns of data.frame “d”
`hist(x, breaks=10, col=“black")`	Plots a histogram of variable “x” with 10 bins colored black
`with(df, plot(x, y))`	Scatterplot of y versus x
`read.table("data.txt",header=TRUE)`	Reads “data.txt” from the current working directory, assuming that the file has a header row of variable names
`write.table(df, file="output.txt")`	Writes data.frame “df” to “output.txt” in the working directory
`read.csv(), write.csv()`	Reads and writes CSV files
`sessionInfo()`	Prints information about R session and versions of all attached packages

DESeq2

`library(DESeq2)`	Load the DESeq2 library (each time you use it; must install once, see below)
`vignette("DESeq2")`	Open the DESeq2 vignette (tutorial with data and examples).
`dds <- DESeqDataSetFromMatrix()`	Creates a DESeqDataSet object called “dds” from three objects: a count matrix, column data, and a design formula. See the DESeq2 vignette for more information.
`dds <- DESeq(dds)`	Runs the DESeq pipeline on a DESeqDataSet object named “dds” — estimates normalization size factors, estimates dispersion, and fits a negative binomial model. Reassigns the result to the same “dds” object.
`res(dds)`	Get differential expression results out of a DESeqDataSet object that has been run through the DESeq() pipeline. Returns a data.frame.
`plotMA(dds)`	Plots a MA-plot of an analyzed DESeqDataSet
`rld <- rlogTransformation(dds)`	Apply a regularized log transformation of a DESeqDataSet to create a dataset useful for clustering or heatmaps.
`assay(rld)`	Gets the regularized log data out of a transformed dataset.
`hclust(dist(t(assay(rld))))`	Runs hierarchical clustering (hclust) on the distance matrix (dist) of the transposed (t) expression data (assay) from a regularized log-transformed DESeqDataSet named “rld”

Installing Packages

Packages only need to be installed once, but must be loaded (with `library`) every session.

From CRAN (http://cran.r-project.org/web/packages/):

```
install.packages("mypackage")
```

From Bioconductor (http://www.bioconductor.org/):

```
source("http://bioconductor.org/biocLite.R")
biocLite("mypackage")
biocLite("myotherpackage")
```

Load an installed package:

```
library(mypackage)
```

RStudio

- Options / Pane layout: In the course my setup look like this (source top left, console top right, environment bottom left, files bottom right):

Source	Console
Environment/history	**Files/plots/help**

- Session → Set Working Directory → Choose…
- Always write commands and save them in a script file (file, new file, R Script). Save this script, and send commands from the editor (source) to the console with Ctrl-Enter (PC) or ⌘-Enter (Mac).
- If your console prompt starts with a + instead of a >, you likely forgot to close a quote or parenthesis. Focus on the console pane and hit the Escape key.