A short list of some useful R commands

help() # give help regarding a command, e.g. help(hist)
c() # concatenate objects, e.g. x = c(3,5,8,9) or y = c("Jack", "Queen", "King")
1:19 # create a sequence of integers from 1 to 19
(...) # give arguments to a function, e.g. sum(x), or help(hist)
[...] # select elements from a vector or list, e.g. x[2] gives 5, x[c(2,4)] gives 9 for x as above
matrix() # fill in (by row) the values from y in a matrix of 4 rows and 3 columns by giving
m = matrix(y,4,3,byrow=T) # fill in (by row) the values from y in a matrix of 4 rows and 3 columns by giving
dim() # gives the number of rows and the number of columns of a matrix, or a data frame
head() # gives the first 6 rows of a large matrix, or data frame
tail() # gives the last 6 rows of a large matrix, or data frame
m[,3] # gives the 3rd column of the matrix m
m[2,] # gives the 2nd row of the matrix m
= or <- # assign something to a variable, e.g. x = c("a","b","c")
== # ask whether two things are equal, e.g. x = c(3,5,6,3) and then x == 3 gives T F F T
Then y[x == 3] gives those entries of y where x equals 3, i.e. the 1st and 4th entry of y
< # ask whether x is smaller than y, e.g. x < 6 in the example above gives True True False True
> # ask whether x is larger than y
& # logical 'and'
| # logical 'or'
sum() # get the sum of the values in x by sum(x)
mean() # get the mean of the values in x by mean(x)
median() # get the median of the values in x by median(x)
sd() # get the standard deviation of the values in x
var() # get the variance of the values in x
IQR() # get the IQR of the values in x
summary() # get the summary statistics of a single variable, or of all variables in a data frame
round() # round values in x to 3 decimal places by round(x,3)
sort() # sort the values in x by giving sort(x)
unique() # get the non-duplicate values from a list, e.g. x = c(3,5,7,2,3,5,9,3) and then
unique(x) # gives 3 5 7 2 9
length() # gives the length of the vector x, which is 8
hist() # create a histogram of the values in x by hist(x)
stem() # create a stem and leaf plot of the values in x by stem(x)
boxplot() # create a boxplot of the values in x by boxplot(x)
plot() # scatterplot of x vs. y by plot(x,y); for more parameters see help(plot.default)
cor() # gives the linear correlation coefficient
lm() # fit a least squares regression of y (response) on x (predictor) by fit = lm(y~x)
names() # get or set the names of elements in a R object. E.g. names(fit) will give the names of the R
object named "fit", or
# get or set the names of variables in a data frame.
fit$scoeff # gives the least squares coefficients from the fit above, i.e. intercept and slope
fit$fitted # gives the fitted values for the regression fitted above
fit$residuals # gives the residuals for the regression fitted above
lines() # add a (regression) line to a plot by lines(x,fit$fitted)
abline() # add a straight line to a scatterplot
points() # add additional points (different plotting character) to a plot by points(x,y2,pch=5)
scan() # read data for one variable from a text file, e.g. y = scan("ping.dat")
# Don't forget to change to the appropriate directory first
read.table() # read spreadsheet data (i.e. more than one variable) from a text file
table() # frequency counts of entries, ideally the entries are factors (although
# it works with integers or even reals)
write() # write the values of a variable y in a file data.txt by write(y, file="data.txt")
log() # natural logarithm (i.e. base e)
log10() # logarithm to base 10
seq() # create a sequence of integers from 2 to 11 by increment 3 with seq(2,11,by=3)
rep() # repeat n times the value x, e.g. rep(2,5) gives 2 2 2 2 2
getwd() # get the current working directory.
setwd() # change the directory to. E.g. setwd("c:/RESEARCH/GENE.project/Chunks")
dir() # list files in the current working directory
search() # searching through reachable datasets and packages
library() # link to a downloaded R package to the current R session. E.g. library(Biostrings) link to the
# R package called "Biostrings" which you had downloaded earlier onto your laptop

Input and Display

load("c:/RData/pennstatet1.RData") # load a R data frame
read.csv(file=as.character("C:\stat251\ui\csv"),header=T) # read .csv file with labels in first row
x=c(1,2,4,8,16) # create a data vector with specified elements
y=c(1:10) # create a data vector with elements 1-10
vect=c(x,y) # combine them into one vector of length 2n
mat=cbind(x,y) # combine them into a n x 2 matrix
mat[4,2] # display the 4th row and the 2nd column
mat[3,] # display the 3rd row
mat[,2] # display the 2nd column
subset(dataset,logical)  #those objects meeting a logical criterion
subset(data.df,select=variables,logical)  #get those objects from a data frame that meet a logical criterion

data.df[data.df=logical]  #yet another way to get a subset
x[order(x$B),]  #sort a dataframe by the order of the elements in B
x[rev(order(x$B)),]  #sort the dataframe in reverse order

Moving Around

ls()  #list the R objects in the current workspace
rm(x)  #remove x from the workspace
rm(list=ls())  #remove all the variables from the workspace
attach(mat)  #make the names of the variables in the matrix or data frame #available in the workspace
detach(mat)  #releases the names
new-old[,n]  #drop the nth column
new-old[-n],]  #drop the nth row

complete = subset(data.df,complete.cases(data.df))  #find those cases with no missing values
new=old[n1:n2,n3:n4]  #select the n1 through n2 rows of variables n3 through n4)

Data Manipulation

x.df=data.frame(x1,x2,x3 ...)  #combine different kinds of data into a data frame
scale()  #converts a data frame to standardized scores
round(x,n)  #rounds the values of x to n decimal places
ceiling(x)  #vector x of smallest integers > x
floor(x)  #vector x of largest integer < x
as.integer(x)  #truncates real x to integers (compare to round(x,0)
as.integer(x < cutpoint)  #vector x of 0 if less than cutpoint, 1 if greater than cutpoint)
factor(ifelse(a < cutpoint, "Neg", "Pos"))  #is another way to dichotomize and to make a factor for analysis
transform(data.df,variable names = some operation)  #can be part of a set up for a data set

Statistical Tests

binom.test()  #perform test with proportion(s)
prop.test()  #perform t test
t.test()  #perform Chi-square test
chisq.test()  
pairwise.t.test()
power.anova.test()
power.t.test()
aov()  
anova()  
TukeyHSD()  
kruskal.test()

Distributions

sample(x, size, replace = FALSE, prob = NULL)  # take a simple random sample of size n from the population x with or without replacement
rbinom(n,size,p)  
pbinom()  
qbinom()  
dbinom()  

rnorm(n,mean,sd)  #randomly generate n numbers from a Normal distribution with the specific mean and sd
pnorm()  #find probability (area under curve) of a Normal(10,3^2) distribution to the left
#of 8, i.e. P(X <= 8), by pnorm(8,mean=10,sd=3)
qnorm()  #find quantity or value x such that area under Normal(10,3^2) curve and to the left
#of x equals 0.25 by qnorm(0.25,mean=10,sd=3)
rt()  
pt()  
qt()  
runif(n,lower,upper)  
punif()  
quainf()