R commands

We give a brief introduction to some important R commands in this document. it is recommended that you consult at least one book on R, such as 'An introduction to R' listed in the course document.

For a lot more information on R commands type
help(pnorm)
if you want to know about the command pnorm, or
help.search(binomial)
to find out about commands involving the binomial distribution or
help.start()
to go to the main help menu. R does not have the most user-friendly help system for beginners, but it is very useful with a bit of practice.
When finishing a session, type
q()
Note that if you have done a lot of calculations with R, using various columns etc. of data (see below), then you may wish to remove these to save confusion before quitting.
ls()
will list all of these.
rm(x,y,z)
will remove the three data sets x, y and z.

To read a string of data into the programme e.g. 1,2,3,5,7 recorded as data set with label x, type
x<-c(1, 2, 3, 5, 7)
Simple transformations can be obtained, for example, as follows;
lx=log(x)
sx=sqrt(x)
R can also be used to perform numerical calculations just like a calculator, for instance
exp(-2)
log(2)

To work out the probability that a standard normal random variable takes value less that a value (e.g. 1.5), type
pnorm(1.5)
For a normal distribution with mean 2 and variance 9, this would become
pnorm(1.5,mean=2,sd=3).
The probability that a binomial random variable, parameters 4 and 0.3, is less than or equal to 2 is given by
pbinom(2,size=4,prob=.3)

The probability that it is equal to 2 is
dbinom(2,size=4,prob=.3)
To generate a random sample of 10 observations from a standard normal distribution, type
rnorm(10)

If you have a set of data labelled x with 7 values with known variance 4 you can find a 95% confidence interval with
mean(x)-1.96*2/sqrt(7)
mean(x)+1.96*2/sqrt(7)

A similar confidence interval for a data set with 7 values and unknown variance is given by mean(x)-2.447*sd(x)/sqrt(7)
mean(x)+2.447*sd(x)/sqrt(7)

To find a 95% confidence interval of the difference of two data samples labelled x (5 elements) and y (9 elements), where the variance of the two data sets can be assumed to be the same can be found by
pool<-(var(x)*4+var(y)*8)/12
mean(x)-mean(y)-2.178*sqrt(pool)*sqrt(1/5+1/9)
mean(x)-mean(y)+2.178*sqrt(pool)*sqrt(1/5+1/9)

to test whether the underlying mean of a distribution which produced data set x is equal to 6, type
t.test(x1,mu=6)
if the alternative hypothesis is that the mean is not equal to 6, and
t.test(x1,mu=6,alternative=”less”)
if the alternative is that it is less than 6 (similarly ”greater” for greater than 6).

To perform a two-sample t-test on x and y, we first need to merge them into a single column
z<-c(x,y)
and label an indicator column
ic<-c(1,1,1,1,1,2,2,2,2,2,2,2,2,2)
if there are 5 x elements and 9 y elements.
To test for equality of means when the variances can be assumed to be equal and the alternative hypothesis is that they are different type
t.test(z2~ic,var.equal=T)
To perform Welch's test where the assumption of inequality does not hold type
t.test(z2~ic)
It is possible to test for a different alternative hypothesis as before. For paired data, x and y can be converted to a single set of data by
w<-x-y

To produce a figure on screen, just type the appropriate command, such as;
boxplot(x)
to produce a boxplot of x
hist(x)
to produce a histogram
qqnorm(x)
to produce a normal QQ plot.
boxplot(z∼ic)
would produce back to back boxplots of the two data sets merged into a single column as before.

To produce a plot that can be printed, save it to a postcript file by typing
postscript(”graph1.ps”, horizontal=FALSE, height=5,pointsize=10)
and then the required figure to produce this figure and call in graph1.
The commands that have been given above (hist, boxplot) produce the default histogram and boxplot with no figure labels. To produce the kind of plot that you would put into a report requires more work (and also sometimes the default plot is not appropriate to explain the data). You should investigate how to do this.
To perform a Wilcoxon test that x has mean 6, type
wilcox.test(x,mu=6)

A simple plot of the data pairs x,y can be produced by
plot(x,y)
To carry out a linear regression of y on x, type
z<-lm(y∼x)
and to plot the regression line from the model
lines(x,fitted(z))
Other information can be found from this fitting, for instance the residuals can be obtained by typing
zr<-resid(z)
The correlation of x and y can be found using
cor(x,y)
To find confidence and prediction intervals for the above regression, type
predict(z,interval=”c”)
predict(z,interval=”p”)

To carry out an analysis of variance, firstly we need to set up the categories correctly. if we have 9 observations, split into three groups we could use
group<-c(”A”, ”A”, ”A”, ”B”, ”B”, ”B”, ”C”, ”C”, ”C”)
group1<-factor(group)
Then type in the observations, e.g.
x<-c(1, 4, 2, 4, 3, 4, 2, 1, 2)

Typing
anova(lm(x∼group1))
performs the analysis of variance. For more details, type
summary(lm(x∼group1))
To test whether a vector (e.g. (3, 17, 14) comes from a given multinomial distribution (probabilities (0.1, 0.4, 0.5)), type
x<-c(3, 17, 14)
cp<-c(0.1, 0.4, 0.5)
library(stats)
chisq.test(x,p=cp)

To test for the equality of several multinomials, type the matrix as follows (which would gives the numbers 1 to 9 in three rows, and so also three columns)
m<-matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), nrow=3, byrow=T)
chisq.test(m)
carries out the test. To display the expected scores for each cell, type
E<-chisq.test(m)$expected
The observed entries can be re-generated similarly, by
O<-chisq.test(m)$observed
and the value contributing to the chi-square statistic by each cell can then be shown using
(O-E)ˆ2/E

To fit a polynomial regression of y on x, type (for a quadratic fit)
summary(lm(y∼x+I(xˆ2)))
if we want to perform a mutliple regression using the data set ”multiple”, we type data(multiple)
To observe the data, type
multiple
To carry out the regression, type
lm(multiple)
You may also want to perform the regression on some of the variables from ”multiple” only. We will investigate how to do this at the appropriate time.

For principal component analysis, you must first type library(pcurve)
To carry out a principal component analysis on a correlation matrix m, type
pca(m, cent=FALSE, scle=FALSE)
If you have a file containing data, called ”pcdata”, you can obtain a correlation matrix using
data(pcdata)
m<-cor(pcdata)
and then carry out your analysis.