A Quick Introduction to R

The point of these few pages is to give you a quick introduction to the possible uses of the free software R in statistical analysis. I will only expect you to know how to perform very basic tests using R, but will give you some additional information here that you might find useful (either now for the purpose of this course or later in your professional life if you deal with randomness).

1 Obtaining R

The size of the basic R package is about 30Mb, so it should take only a few minutes to download (possibly less if using a fast connection). You can get it at

http://www.r-project.org/

First click on "Download R", then click on the name of your operating system (Linux, Mac OS X, or Windows), after which you can pick the package to install (the link to follow depends on your operating system). The rest should be fairly self-explanatory.

2 Distributions

The following is a(n incomplete) list of distributions available in R. Many things can be done with these distribution, but two important features are the possibility to

- generate random numbers according to the given distribution
- draw the p.d.f. or c.d.f. of the given distribution

Distribution	R name	parameters
$binomial$	$binom$	$size, prob$
$Cauchy$	$cauchy$	$location, scale$
$chi-squared$	$chisq$	df, ncp
$exponential$	exp	$rate$
F	f	$df1, df2, ncp$
$gamma$	$gamma$	$shape, scale$
$geometric$	$geom$	$prob$
$hypergeometric$	$hyper$	m, n, k
$negative\ binomial$	$nbinom$	$size, prob$
$normal$	$norm$	$mean, sd$
$Poisson$	$pois$	$lambda$
$Student's\ t$	t	df, ncp
$uniform$	$unif$	min, max

Note that these commands for distributions and a few others can be found on p.39 of one of the many user's manuals for R at

cran.r-project.org/doc/manuals/R-intro.pdf

2.1 random number generation

- To generate random data from one of the distributions above, stick an "r" (for "random number") in front of the chosen distribution. For instance, "U=runif(1000,1,3)" produces 1000 independent $\mathcal{U}(1,3)$ (that is, uniform with parameters 1 and 3) random variables. (Note: "U" is the name I chose to give to this vector of random variables. Pretty much any other name you wish to choose should be fine as well)

- To see a histogram of data , write "hist(U)". This will automatically generate a certain number of bars for your histogram. If you would like to change the number of bars, you can do that. For instance, if you'd like to generate 100 bars, write "hist(U,breaks=100)".

It's certainly worth playing with this for a while and trying a number of distributions.

2.2 drawing p.d.f.'s

To obtain the probability density function of a given random variable, use the same suffix as in the table above, but stick a "d" (for density) in front of it.

This is only very slightly involved, as one has to define the domain as well. Of course, infinity exists in mathematics, but should be avoided in computer science (infinity is a nice concept, but you don't want your computer to do anything for infinite time). Therefore, you should pick a part of the domain you're interested in. The key to defining the domain is to choose the left and right bounds and say how often you want the p.d.f. to be evaluated in between (this is the number of points your graph will consist of). Let's look at an example (try this at home).

Let's draw the p.d.f. of an exponential with mean 1. Since exponential random variables are positive, the interesting part of the domain is on the positive real axis. We'll do this in three steps:

1. Let's see what we get if we choose $x \in [0,4]$.

   - Type "x=seq(0,4,0.5)" (this will create points for x-values between 0 and 4 with increments of 0.5; that is, it will create the set of points 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4).
   - Now type "y=dexp(x)" (this will define the value of the p.d.f. at the points we just defined above).
   - Now type "plot(x,y)". You now have a rough picture of the p.d.f. of an $Exp(1)$ random variable.

2. You'll probably agree that the picture you just got could look better. Let's try taking a bigger domain (say from 0 to 8).

   - Type "x=seq(0,8,0.5)".
   - Now type "y=dexp(x)".
   - Now type "plot(x,y)". You now have a better picture of the p.d.f. of an $Exp(1)$ random variable.

3. It's still not a great picture. Let's try again with the same range but more x values.

   - Type "x=seq(0,8,0.01)".
   - Now type "y=dexp(x)".

- Now type "plot(x,y)". You now have a much better picture of the p.d.f. of an $Exp(1)$ random variable.
- We still shouldn't be completely satisfied, since we the graph is composed of lots of big circles. We can fix the way the points are linked with the command "plot(x,y,type="l")", which says the points should be linked with a line. Now the picture looks pretty good.

Now it's your turn. With the few basic commands from above, you can draw the pdf of any random variable you've ever encountered. You can also generate large random set drawn (independently) from those distributions.

2.3 the maximum of a sequence of random variables

It usually takes a while to grasp why the maximum of a sequence of random variables has a different distribution from the random variables themselves. For instance if you ask someone what should the distribution of the maximum of 10 independent standard normal random variables, most people will at first think that it is also a standard normal. We saw in class that this is not the case. The following example will provide further empirical evidence of this fact.

We will create a histogram designed to mimic the shape of the p.d.f. of the maximum of a certain number of random variables. The idea is to put the random variables in a matrix as opposed to a vector.

Suppose you have a vector of length $n \cdot m$. R allows you to transform this vector into an $n \times m$ matrix. For instance, you can transform the vector

$$[a, b, c, d, e, f, g, h, i, j, k]$$

into the matrix

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix}.$$

We will now see how to do this with a vector of independent realizations of normal random variables. When you generate a large number of random samples form a certain distribution, the histogram of these values looks roughly like the p.d.f. (or p.m.f. in the discrete case). So if we generate, for instance, 1000 times 4 independent standard normal random variables and pick every one of the 1000 times the largest of the 4 values (that is, we pick the maximum), then a histogram of these 1000 maxima will look roughly like the p.d.f. of the maximum.

Here's how to do this:

- Create 4000 independent realizations of standard normal r. v.'s by typing "N=rnorm(4000)".
- We now want to split these random variables into 1000 groups of 4, more precisely put them in a matrix of 1000 rows and 4 columns. To do this, type "dim(N)=c(1000,4)".
- Now we want to create a vector that contains the 1000 maxima of the groups of 4, that is, the maximum of every row in the matrix. To do this, we first have to define a vector that will contain these maxima. Let's call it M. Initially, we have to define the spaces in which to put the maxima. One way of doing this is to initially let M be a vector containing nothing but zeros. To do this, type "M=array(0,1000)". Once that is done, we can ask R to put the largest value of all the rows of N into the vector M. To do this, type "for (i in 1:1000)M[i]=max(N[i,])".
- The vector M now contains the 1000 maxima obtained from the 4000 realizations of independent normals. To see how they are distributed, just plot a histogram by typing "hist(M)".

3 Importing data

Data files come in many shapes and colors. We will deal only with .txt files. In this example, we will look at the following data set, which gives heights of men and women. The question to answer will be: Might they be the same?

http://userhome.brooklyn.cuny.edu/cbenes/Height.txt

1. First we need to save the file on our computer, for instance in our Desktop directory. This is something I'm sure you've done many times, just perhaps not with .txt files. The procedure is of course the same. If you're not sure how to do this, let me know.

2. Now we would like to make sure that R is able to find this file. One way to ensure this is to put R in the same directory as the file. First, we have to find where R is sitting on our computer. To do this, type "getwd()" (this means: "get working directory"). Everyone will get something different (call it "X"), but in my case it was "/Users/christian". Now, we can move the location of the working directory, using the command "setwd". In my case, I had to type

   setwd("/Users/Christian/Desktop")

   Once we're working in the same directory where R is, we can load the file by typing

   HEIGHT=read.table("height.txt",header=TRUE)

   To now see what the file "HEIGHT" is, type "HEIGHT". Note that the command "header=TRUE" considers the first line of the file height.txt as being a header of the file, but not part of the data.

3. The last preparatory step requires separating the data set into "woman" data and "man" data. To create the "man" data set, we want to take all the elements from the first column and rows 1 to 25 of the data set. To do this, type

   man = HEIGHT[1:25,1]

   and for the "woman" data set, type

   woman = HEIGHT[1:23,2]

4. Now the goal is to test equality of means for the two data sets "man" and "woman". First we test that the variances are equal by typing "var.test(man,woman)". Since there is no reason to reject the hypothesis that variances are equal, we can perform a two-sample t-test by typing "t.test(man, woman, var.equal=TRUE)" and we see that we should reject the hypothesis that they are equal, since the p-value is $\approx 2.251 \cdot 10^{-9}$

Note: An alternative to steps 1. and 2. is to load the file directly into R using the following two commands:

> www="http://userhome.brooklyn.cuny.edu/cbenes/Height.txt"
> HEIGHT=read.table(www,fill=TRUE)

4 Bivariate data

This section will be useful at the end of the semester when we discuss the linear model. R can perform in a few seconds various tests which could take a huge amount of time when done by hand. If you have a set of bivariate date $(X, Y)_i$, in the form of two vectors X and Y of the same length, then you can

- compute the Pearson sample correlation using the command "cor(X,Y)" and test for zero correlation using the command "cor.test(X,Y)".
- Find the least squares line $y = a + bx$ by using the command "fit = lm(Y $\sim$ X)" and then "fit", which will give you the two coefficients a and b in that order. Typing "resid(fit)" will then give you the residuals (which you can graph by typing "plot(resid(fit))"). To see the data together with the least squares line, first type "plot(X,Y,type="l")" and then "abline(fit)", which will draw the least squares line on top of the data.

5 Assignment

This assignment is due in class on December 3 (this means you have plenty of time to do it; the later you start, the less likely you are to complete it in time and benefit from it). It is worth a total of 3 points. The number of points you obtain on this assignment will be added to your highest quiz grade of the semester. Explain clearly what you did when using R (or another software). In particular, explain which commands or buttons you used. Write down or print what answers the software gives, and interpret them in full sentences the same way as when you perform tests by hand. DO NOT hand in any pages and pages of data (save trees!). I will have an idea from your graphs of what data you produced.

1. (a) Generate 10000 independent samples of $\mathcal{U}(-1, 1)$ random variables. Plot a histogram and hand it in with your solutions to this assignment.
   (b) Generate 10000 independent samples of the maximum of 5 independent $\mathcal{U}(-1, 1)$ random variables. Plot a histogram and hand it in with your solutions to this assignment.
   (c) Generate 10000 independent samples of the maximum of 10 independent $\mathcal{U}(-1, 1)$ random variables. Plot a histogram and hand it in with your solutions to this assignment.
   (d) Generate 10000 independent samples of the maximum of 100 independent $\mathcal{U}(-1, 1)$ random variables. Plot a histogram and hand it in with your solutions to this assignment.
   (e) What do you notice happens to the maximum of a set of $\mathcal{U}(-1, 1)$ random variables when the size of that set increases? Explain why this happens.
2. Results of an experiment to test whether directed reading activities in the classroom help elementary school students improve aspects of their reading ability. A treatment class of 21 third-grade students participated in these activities for eight weeks, and a control class of 23 third-graders followed the same curriculum without the activities. After the eight-week period, students in both classes took a Degree of Reading Power (DRP) test which measures the aspects of reading ability that the treatment is designed to improve. The scores of the DRP test (called "response") were the following:

Test this data set for

(a) equality of variances; state what assumptions you are making, what your conclusion is and why it is what it is.

(b) equality of means; state what assumptions you are making, what your conclusion is and why it is what it is.

Note that you will not need to include the command "header=TRUE" when loading the file, since the first line of the file is already part of the data set and not the header.