

R handout for the newbies.

I. What is it?

A. R, is a statistical programming language based on the [S language](#). S has been around since the 70s, and eventually became the S-Plus statistical package. R spun off as an open source alternative in the mid 90s with some of the original S people involved, e.g. S language creator John Chambers. R is also referred to as Gnu-S.

II. How it works

A. The first few things to know going into using R are objects, functions, and packages

i. Object: Symbolic representation of something else. Objects can be *anything*: variables, datasets, functions, output from an analysis, a list of names, a graph... *anything*. For instance, if you type:

```
MyObj = rnorm(100)
```

You have just created an object called MyObj that is 100 random samples from a standard normal distribution (i.e. 100 z-scores with a mean of zero sd of 1).

ii. Function: functions have arguments or inputs which the function will then use to produce some output. In the above example rnorm is the function, and while we just did the defaults we could have specified a specific mean and standard deviation as additional arguments. Here is an example of the lm function (linear model) used to do regression:

```
MyModel = lm(Y ~ X, data = mydata)  
summary(MyModel)
```

The above created an object called MyModel which was a regression analysis performed by the lm function of Y predicted by X, which were variables from a dataset called mydata. Afterwards we used the summary function to give more detailed output such as an ANOVA table, R-squared etc.

For applied use that little bit can actually take you along way. As an example of modern techniques, with the right packages installed you could have put:

```
lm.boot(MyModel, R=500)
```

 A bootstrapped regression after creating the model as above

```
MyModel= rlm(Y ~ X, data=mydata)
```

 A robust regression resistant to outliers and more accurate with violations of the homoscedasticity assumption

iii. Package: collection of functions, datasets and other things. Packages are required to be loaded before before the functions it contains are able to be used. For example:

```
library(Rcmdr)
```

Will load the R-commander Graphical User Interface

B. GUI

- i. We usually use [R-commander](#) for easy-to-use menu's when applicable, which is probably most of the time. In some ways it doesn't look like much, but as slim and trim as it is, along with providing the basics, it even surpasses SPSS on things like testing Regression assumptions, Factor analysis in general, and graphical options. It always spits out the code so can be helpful in learning more ways to use R.

- C. R is open source and able to be adapted and added to by anyone interested in doing so. This allows statisticians and applied researchers from around the world to contribute packages that perform thousands of functions (there are about 1600 user submitted packages). The benefits of this for research production and quality are unmatched by any other statistical package.

III. Links

- A. Main website: www.r-project.org
- B. Different packages grouped under topical headings can be found here: <http://cran.r-project.org/web/views/> Note the Psychometrics and Social Sciences views in particular.
- C. [Quick-R](#) I find it very useful for initially learning as it knows where you are coming from

IV. Installation

- A. You can go from CRAN link on the R website but here is a direct [link](#) to the executable via Berkeley's mirror.
- B. You may choose all the default settings during install. Note that this is not like common programs; it is very easy to delete. It is intimidating to start up, all you see is a blank white page with a prompt, but remember, you're not flying solo.
- C. Installing packages. For your home install I always recommend installing all packages in one sitting so you don't have to worry with it again. Because there are so many packages however, it will take awhile, but it won't keep you from browsing the web and doing other things.
 - i. From the menus: Packages/Install new packages. USA (CA 1) is the Berkeley mirror.
 - ii. The list of packages is huge. To install all of them simply select all of them and click ok. This will put your installation at about 2 gigs or more. The alternative is to install packages as you come to need them, but that will mean added time to every assignment/project.

V. Use

- A. When using R, do not work from the command line (red prompt)- it's just not very forgiving. File/Open Script will provide a text editor and anything you do there is easily saved for later use. As we typically have Rcmdr running, many students type or paste the lab code in its script editor, though I personally find the limited view space of both it and the output a drawback. There is also [Tinn-R](#) which you might use further down the line.
- B. Searching R: We try to keep R in as much an applied fashion as possible but with enough experience to feel confident to try your own stuff later if you should choose. But you may still help yourself learn more once you start using it.
 - i. Each time we use a new function type `?functionname` at the command line to bring up the helpfile associated with that particular function.
 - a) These helpfiles vary in their worth to the novice user. Some are written assuming quite a bit of statistical knowledge, some of the examples may include code or data

not required by the specific function etc. However at the very least you will learn more about what inputs and outputs are available, as in the lab we often just stick with defaults.

- ii. `??functionname` will do a basic R helpfile search and list any function that might be related
- iii. [Rseek](#) is a page devoted filtering out web searches to give only R stats results, and has Firefox search engine plugin.

VI. Summary

- A. R takes time to get used to, but if you're interested in doing more with your research, avoiding the problems and issues methodologists have been talking about for decades, adhering to the now decade old suggestions by APA etc. it actually is more efficient