As a starting off point, I wanted to have a bit of info on R up front here just so folks would have it for reference as some have indicated they have R experience and some have not. I am not going to trace the history of R, you can look that up on your own. But, there are lots of good places to look for information. Your first stops should be:

**R Project website:** [http://www.r-project.org](http://www.r-project.org)

**R FAQ:** [http://cran.r-project.org/faqs.html](http://cran.r-project.org/faqs.html) *(general/OS specific FAQs on here)*

**R Manuals:** [http://cran.r-project.org/manuals.html](http://cran.r-project.org/manuals.html)

**CRAN (Comprehensive R Archive Network):** [http://cran.r-project.org/](http://cran.r-project.org/)

**R Search:** [http://www.r-project.org/search.html](http://www.r-project.org/search.html)

**Task View:** [http://cran.r-project.org/web/views/](http://cran.r-project.org/web/views/)

**Texas A&M University that has worked up some R video’s that are interesting:** [http://dist.stat.tamu.edu/pub/rvideos/](http://dist.stat.tamu.edu/pub/rvideos/).

Each of these above links to a wide variety of R-related information and should be in your internet browser bookmark’s collection.

Interacting with R can be accomplished in a variety of ways, most folks tend to use a script editor to write their programming code and then send the code to R (see list @: [http://www.sciviews.org/_rgui/projects/Editors.html](http://www.sciviews.org/_rgui/projects/Editors.html)). I use Eclipse ([http://www.walware.de/goto/statet](http://www.walware.de/goto/statet)) and that is what I will likely use today (or I will use the base GUI to keep it simple). Assuming most of you are Windows users and new to R, RStudio ([http://www.rstudio.com/](http://www.rstudio.com/)) is popular, as is R Commander ([http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/](http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/)). If you are a Mac user, yes you are cool and trendy, but you are on your own as I don’t mess with the trendy stuff so stick with the GUI. If your a Linux guy as I am you should be using ESS-Emacs (which for the sake of most peoples sanity I will not use in class; [http://ess.r-project.org/](http://ess.r-project.org/)) or alternatively Eclipse ([http://www.walware.de/goto/statet](http://www.walware.de/goto/statet)), but given you use Linux you already know this and you are awesome. Note: when you are using one of these editors, you can save your code as “filename.R” just as you would with any other file type. Note the ‘.R’ extension, I tend to use .R as a matter of habit, but you can use ’.txt’ as well but its sometimes not as seamless (FYI, notepad++ is a great free editor).

**Getting R Installed**

You all are competent, so I am assuming you managed to get the latest version of R (its 3.0.2 Frisbie Sailing) installed on your computer (its on the CRAN: [http://cran.r-project.org/](http://cran.r-project.org/)) either by downloading the executable or compiling from source (if not, holler at me before the workshop via email: bret@tamu.edu). If you do not have administrative privileges on your computer that might be a problem as you will need to install packages on your system, so be sure to check that you can 1) download and install R and 2) can download/install R packages (see below) before we have the course meeting. The basic R environment consists of a command window (Console) or graphical user interface (GUI). When you get it installed, click the executable on your desktop and it should look roughly like the following image:
Once you see this, try typing the below in the R console where the cursor is flashing, hitting enter, and see if it says 4. If so, its a party.

\[
2 + 2
\]

## [1] 4

### R Package Background

Ok, now that R is installed there are a couple of other things we will need to worry about. First, while base R does lots of things that we need to do, R also allows you the ability to use code written by a wide variety of people to conduct your own analysis. Since R is open-source freeware, lots of folks contribute to R. The primary individual contributions (outside of the R Development Core Team) are called packages and are the foundation of R (see: http://cran.r-project.org/web/packages/ or http://www.bioconductor.org/ for a wide array of packages). R has more than a thousand packages written by a wide variety of users from around the world ranging from topics such as capture-mark-recapture and habitat analysis to time series and genetic analysis. Basically, packages provide a mechanism for loading optional code and attached documentation as needed for specific analyses. The R distribution provides several base packages that are installed with R. For example, one package, called base, is installed with R, and includes everything that the common user would use or need, like definitions of mathematical or logical operators. But, for the most part, R does not install anything you don’t want or request, which is nice (see ?install.packages for approaches to loading packages). However, to access all this cool extra code you have to load a ‘package’. Straight from the R website, “Packages provide a mechanism for loading optional code, data and documentation as needed.” Thus, packages provide you, the user, with the ability to use code developed and used by other folks within R. Now, there are a few packages that you will need to have installed for the workshop, but, in an effort to front run the workshop so we don’t have to spend time on it I have also downloaded all the package binaries and I think all their dependencies that we might need to http://irnr.tamu.edu/bret/offloads/TxTWS_rworkshop.zip and I have provided a list at the end of this startup helpfile.

### R Package Installation

Installing a package is actually very easy, so I am going to step through it here in a bulleted format so you can see how it works. Lets say you want to install the R package RMark. There are 2 options:

1. If you are online, you can install the package via the Packages toolbar via Packages->Install Packages. Then, you will be prompted (a window will open) to pick CRAN mirror (choose the USA Tx1 one), then another window will open with packages listed in it you can scroll down through the alphabetic list and highlight RMark as this is the package you want to install on your system and hit OK.

2. **Best Option**: In the R command line you can do: `install.packages("packagename", dependencies=TRUE)` (being sure you are online) and by typing in the package name you are interested in installing R will find the most
recent version of the package and install it along with everything else that the package requires. Note that today we will be using a package I created called moveud but you will not be able to find moveud on CRAN as it is not on CRAN yet so you have to install it following the instructions below in 2.

3. If you have the package binaries downloaded as individual .zip files (they all are found at http://irnr.tamu.edu/bret/offloads/TxTWS_rworkshop.zip), or from the CRAN repositories, you can install the package via the Packages toolbar via Packages->Install Package(s) from local zip file(s). Then, a window will open and you can navigate to where you put the R binary .zip packages from the above workshop link and just install them one at a time.

Now, very important to note, whatever package you just 'installed' is still not available to R for use, you have to use R to load the package into your workspace using library("packagename") in the R command line as below. Below is the example one would use to load the RMark package for use in R.

Once you see this, the package is installed and loaded and you are good to use the functions in it. Finally, it is important to note, that lots of packages have 'dependencies' that require R to install other packages on the fly. I think I have all the dependencies included in the .zip file I made available, but you will need to install all the packages in the .zip file before you load them (load meaning use the library("packagename") call as shown above). If you get an error that says ‘Warning message: package ‘packagename’ is not available (for R version 3.0.2)’ don’t worry about it, it just means R wants to see the source package as well, and you just have the binaries (you will probably see this for moveud cause I have not got it on CRAN yet so don’t sweat it).

Packages you will probably need (note I am running R version 3.0.2).

Included in .zip file found at http://irnr.tamu.edu/bret/offloads/TxTWS_rworkshop.zip. Here is a list of extra packages that we will primarily need for class, I think I put all of these (as binary’s) into the class folder as well as the packages they depend on, but in case you were wanting to get them yourself here is the list that I think covers everything.

- RMark
- adehabitatHR
- mrds
- sp
- BBMM
- PBSmapping
- move
- maptools
- rgdal
- ggplot2
- geosphere
- raster
- gridExtra
- moveud-note you have to use the binary I included in the .zip for this one, its not on CRAN