Some useful stuff to know about R:

1. "Workspace" (where your stuff is) is a file with the name of `.Rdata`
   It typically will get deposited onto your "desktop." The first time you click R it will open an empty workspace. If you save it when you exit, then next time you can just click on the icon for .Rdata. If you always use this desktop workspace it will get full of stuff and get unruly, so you may want to copy into a particular project directory. You can see what is in your workspace using: `ls()`
2. Parentheses in R can take parameters. In the case of ls() you were just asking for a list of everything in your current working directory, so there were no parameters. Had you typed just `ls` you were not following directions (tsk-tsk!) and you would get a listing of the `ls` script.
3. The "up arrow" key will take you through your history. After you've looked around in the past, you can come back to the present with the "down arrow" key.
4. Given #1 above, it is useful to copy that initial R workspace to specific project directories, and then:
5. Go to "\misc\remove all objects" in the drop-down menu to clear out the space.
6. Script editing: `fix(whatever.you.want.to.call.it)`
   If that object (i.e., the script) did not already exist then the following will be created:

   ```
   function ()
   {
   }
   ```

   which is a function that will do nothing. But you could edit it in order to do something, like:

   ```
   fix(Hello.World)

   function ()
   {
   cat(paste('\n\n Hello World! on:',date()),'\n\n')
   }
   ```

7. Speaking of editing, R will use your default editor. But you can change to a different editor if you want. First let's see what our options are:

   `options()` – gives us a long list, but
   `options()$ed` – gives us just the editor, which starts off as "internal," and
   `options(editor='c://Program files (x86)/Notepad++/notepad++.exe')`

would switch your editor to "Notepad++" provided you got the path correct and that you have the program installed there.

8. But hey, maybe I want that all of the time. Then place into a script called: `.First`

   The script `.First` runs automatically when you open and close the R workspace.

9. GUI (graphical user interface) options – I've never been a fan of the R default colors. You can change a lot of stuff using the drop-down menu "\edit\GUI preferences." If you save the preferences move this file to the desktop if you want it to apply across the board.
10. The good news is that for the workshop you will be copying and pasting text so that you won't have to edit a bunch of stuff. We'll only really be using the editor to look through the scripts once they have been pasted.
11. Was that = or <-? They work equivalently, with the equals sign being the more modern style. Realize that if you want to test whether something is equal to something else, you need double equals signs. So `1==1` will return TRUE but `1==2` returns FALSE, while 5!=4 returns TRUE (the exclamation means "not," so != is "not equal). If you really wanted 5! (factorial, not like you gotta' have a five and you gotta' have it right now!), that's `factorial(5)`.
12. For simulation, if you want everyone to get the exact same results as you, then set the seed: `set.seed(123456)`
13. For graphics, you can copy and paste a "metafile" into Powerpoint, Word, etc., but for publication: `dev.copy2eps(file='fig01.eps')`
14. RTFM (read the fine manual) – If you feel so inclined, go to the drop down menu "\help\Manuals (in PDF)."
15. Not feeling like reading the manual? Try the drop-down menu "\Help\Html help" then "Packages." Most functions are in "base," "stats," or "graphics." If you know what you're looking for a quick way to get help is by preceding the function with a question mark, such as: `?dnorm`
16. If you are on the grail quest ("What is your favorite colour?") and quite lost, "googling" with the preface "cran r" might get you where you want to be. For example, I wanted a quick way to get empirical highest posterior densities in R for the workshop. I knew that the package "coda" has this, but wanted to avoid using this package. A search on "cran r highest posterior density" got me to another package (TeachingDemos) from which I pulled a tee-toncey (that's southern Appalachian for a little bit) of code that we will use in the workshop.