Roith

R Tutorial: Introduction

1. **What is R?**
   R is a high-level statistical programming language and computing environment.

2. **R is a programming language that is:**
   - interactive
   - interpretive
   - functional
   - object-oriented

3. **How to use R:** To use R effectively, you need to have some knowledge of:
   - statistical methods
   - numerical computing
   - programming techniques
   - statistical graphics

4. **Elements of R:** To learn R effectively, you will need to recognize and understand some of the many special features and concepts in R language, such as:
   - functions
   - objects
   - vectors
   - lists
   - data frames
   - methods

5. **Why use R in this class?** It's free! Many departments of statistics use R, along with lower level languages such as C and Fortran, as the programming language for supporting computations associated with research. Software developed in R by many statisticians is available to R users without the restrictions usually associated with commercial software. As R evolves and becomes more user friendly, with more graphical capabilities, many companies in industry and many disciplines outside of statistics are using it as the standard for data analysis.

6. **Manuals:** The following manuals are available for reference.
   a. Online PDF-formatted manuals: An Introduction to R, the R Reference Manual, and the R Language Manual are available through the Help menu on the standard menu bar in R.

7. **Texts:** The following books are useful for reference.

a. Data Analysis and Graphics using R – John Maindonald and John Braun
b. Modern Applied Statistics with S-PLUS – Venables and Ripley
c. Introduction to Statistics with R – Peter Dalgaard

8. **Web:** The following websites are useful.
   a. R site: http://cran.r-project.org/
   b. S-PLUS Vendor: http://www.insightful.com/
   c. Statlib database: http://lib.stat.cmu.edu/

9. **R Objects:**
   Structures that R creates to store or represent data are called *objects.* These may just be simple entities such as *vectors* or *matrices* or more complex ones such as *lists* or *functions.*

10. **Workspaces:**
    The collection of objects created and stored in memory during an R session is called the *workspace.* At the end of an R session, all objects created during the session may be permanently saved in a file called .RData. This file is identified with a large blue colored letter R (or as R Workspace) in your folder view. If the user caves the current workspace just by answering "yes" to the prompt "Save workspace image?" when the current R session is terminated, the .RData file will be saved in the *current working directory.* If the user had started up the current R session by clicking on the R icon on a public computer, the current working directory is usually a folder in a drive set by the administrator of the lab. However, when working on your own computers, you may save your workspace in any folder you choose by using the Save Workspace… option in the File menu. When R is restarted at a later time, a workspace thus saved may be restored using the Load Workspace… option on the File menu. This allows the user to access R objects saved in previous R sessions and also add newly created objects to the same workspace.

    If you use public labs, it is recommended that R objects created when doing computations in different courses and projects be saved in different .RData files and store them in directories (or folders) that can be identified by their names. You could then change to the appropriate directory (or folder) as your working directory at the start of each R session. Text files such as data files may also be stored in this folder allowing them to be easily accessed by R. Using different R workspaces also allows the user to avoid conflicts caused by identical names for objects with different contents.

11. **R Commands:**
    R commands are of the form of *expressions* entered at the ">" prompt. When a value is *assigned* to a name, an R *object* is created.

```
> x = 3.5
> x
[1] 3.5
```

Here, the R object x is created and the value 3.5 is assigned to it. The symbol = can be used in R for assignment, as an alternative, <- may be used for assignment of objects. It is your choice as to which you prefer to use, you may see me using both.

```
> 3*(11.5 + 2.3)
[1] 41.4
> g <- 3*(11.5 + 2.3)
> g
[1] 41.4
```

3*(11.5 + 2.3) is an example of an arithmetic expression. The result of *evaluating* this expression is displayed in lines immediately following the current command line and is said to be *printed*. When the name of an object is entered at the ">" prompt, the value of the object is printed, sometimes extending to several lines. These lines are labeled with numbers identifying elements of the R *object* being printed. Value of an expression may be *assigned* to a name, creating a new R *object.* R objects thus created may be used in other R expressions. Thus:

```
> g * 1000
[1] 41400
```

12. Some useful R functions:

The function ls() (for providing a list of objects in the workspace) and rm() (for removing objects in the workspace) may be used for checking and organizing R objects in your current workspace. You may also use ls() to check if you are in the right workspace by identifying names of objects that you have previously created.

```
> ls()
[1] "actors"   "blood"    "blood2"   "cars"     "insulin"  "insulin2"
[7] "x"        "y"        "z"
> rm(cars,x)
> ls()
[1] "actors"   "blood"    "blood2"   "insulin"  "insulin2" "y"
[7] "z"
```

The help() function produces the specifications to any R function in a pop-up window. For example, try:

```
>help(ls)
>help(c)
>?ls()
```

Closing the RGui Window by clicking the X or exiting via the File menu, or using the q() function, will prompt you with the question “Save workspace image?”. If you answer “yes”, the R objects you create in your session (your workspace) will be saved in the folder RData. You really do not want this to occur when you are working on a public computer. Thus, make sure that you use the Save Workspace… option on the File menu to save the workspace in the correct folder.

```
>q()
```

13. **Finding more R functions:**

    There are many other functions that are available in the base R software, and many more from packages that may be downloaded and installed. Use search engines such as Google to try and find a function or code that has already been written to perform a specific task. This is a good way to learn how to read the help documents and usually provides examples of short, efficient code.