Introduction to R: Part I

Course : Introduction to Probability and Statistics, Math 113 Section 3234
Instructor: Abhijit Champanerkar
Date: Sept 5th 2012

R is a powerful freeware (i.e. you can download it for free!) computer package that makes doing statistics easier.

In this introductory session we will learn:

1. How to log on to the computers at CSI ... Passwords and ID numbers and all that.
2. Basic operations in **R** : how to start **R** , insert data into **R** and different ways of looking at data using **R** graphics.
3. How to access one of the many data sets already inside **R** .

---

1. Logging onto Computers in the Lab

To use the computers in the CSI computer labs, you will need an login name and a password. For your initial logon, the following should work:

- Login Name: firstname.lastname
- Initial Password: Your Birthday in six numbers: mmddyyyy+ last 4 digits ofyour SSN
- Log on to: SLAS

For example, John Smith, born on July 4, 1990, SSN: 123-45-6789, would have:

- Login Name: john.smith
- Initial Password: 070419906789
- If this password does not work try using mmddyy i.e. 070490.

If you need more help, call your instructor or visit

`http://www.csi.cuny.edu/studenthelpdesk/ServicesInstructions/HD_SERVICES_LAB_LOGIN.html`

---

2. Basic operations in R

2.1 Starting R: Once you've successfully logged in, starting **R** is as simple as *clicking* on the **R** icon on the desktop. Several windows will open up and you are ready to do some statistics!

1. You communicate with **R** through the **R** *Console Window*. All commands are entered here. This window will have a prompt given by a red > sign.

2. **R** is good at arithmetic. Go ahead and try some basic aritmetic. For example:

```
> 265+343
[1] 608
> 123*45
[1] 5535
> 375/12
[1] 31.25
> 73^(1/3)
[1] 4.179339
```

3. If you ask **R** for help, the results will appear in the **R** *Information Window*. Help and more information about functions in **R** is obtained by typing "?" followed by the function name. For example type:

```
> ?mean
```

4. When you construct graphs and plots, these will appear in a seperate **R** *Graphics Window*. Control of the different windows is available by *clicking* on them or selecting them from the 'Windows' pull down menu at the top of the screen.

2.2 Quitting R: What? You want to leave so soon? Well its always good to know how to get out of a program once you are in. Most things in R are *functions* which mean they take *arguements* inside parentheses. To exit **R** , type in the function `quit()`:

```
> quit()
```

That's it, now you are back to the safety of Windows.

2.3 Entering data in R: Renter **R** if you have quit. As we know, Statistics is all about *data* so we need to know how to enter some data we've collected into **R** so we can manipulate, reaarange and graphically decribe it. One easy way to insert data in **R** is using the `c()` function. For example:

```
> c(1,2,3,4,5)
[1] 1 2 3 4 5
> c(1:5)
[1] 1 2 3 4 5
> c(1:5,10:15)
 [1]  1  2  3  4  5 10 11 12 13 14 15
```

Suppose we had collected 'favorite number' data 3, 2, 1, 4, 3 from five people, we could store this data in a variable named `Numbers` by typing:

```
> Numbers = c(3, 2, 1, 4, 3)
```

Now **R** has a copy of our data in `Numbers`. If we want to look at what is contained in `Numbers`, just type:

```
> Numbers
[1] 3 2 1 4 3
```

and the results appear. Suppose you wanted to add 22 to all the favorite numbers. This is easy:

```
> Numbers+22
[1] 25 24 23 26 25
```

Or, perhaps something more complicated like adding 22 and then taking the square root of the results:

```
> sqrt(Numbers+22)
[1] 5.000000 4.898979 4.795832 5.099020 5.000000
```

2.4 Plotting data in R: Looking at raw numbers on a screen is never very exciting. To remedy this **R** has a huge number of different plotting routines for graphically representing data.
The simplest graphing routine is a function called `plot()`. For example, to show our favorite number data, we could try:

```
> plot(Numbers)
```

This command produces no output in the Console Window, but it does give us a nice plot of our numbers in the Graphics Window. Try it.
How about histograms? **R** is good at that too. The function for producing histograms is `hist()`. Try it on `Numbers`.

```
> hist(Numbers)
```

What do you get? Is it 'right'? Think about it!
Maybe this makes more sense:

```
> hist(Numbers,breaks=c(0.5:4.5))
```

Better? Why? What's this `breaks=` stuff? Think about it and lets discuss.

---

3. Existing Data Sets in R

R has a huge number of strange (and interesting) data sets already installed and ready to look at statistically. To see a complete listing of **R** 's data collection type the command:

```
> data()
```

This should open a new window with a long list of data sets. To get more information about any data set, type:

```
> help( name of the dataset here )
```

For example, lets look at the data containing `airquality` information in New York before you were born (1973). To load this data into **R** ,

```
> data(airquality)
> attach(airquality)
```

The last command `attaches` the names of different data in `airquality` to the actual data. To see what information is in the dataset, and what the names of the different pieces of data are:

```
> names(airquality)
```

We can now look at the data. For example, make a histogram of the OZONE measurements during 1973 - Ozone is bad for you, especially if you have lung problems to begin with. Try:

```
> hist(Ozone)
```

---

4. Classwork

1. Make a histogram of Ozone, Temperature and Wind Speed.
2. **Print** the Ozone histogram.
3. What was the AVERAGE temperature during the measurement period? **Print** the Temperature histogram and write this on the Temperature histogram.
4. **To hand in:** Write your name and answers on the 2 printouts and submit it to the instuctor.
