Demography 213
Week 1: Introduction to Linux, The Demography Lab, and R

Carl Mason
carlm@demog.berkeley.edu

August 24, 2016

Abstract

This week's exercise consists of logging into the Linux system; becoming familiar with the ***terminal window*** application and with the ***MATE desktop*** in general; using a few of the "12" most important Linux commands to set up a few directories; and experiencing R as a command line application. Then spending at **at least five hours** at Data Camp learning the basics of R – the statistical computing environment used by the most successful and good looking demographers.

Contents

- The class meets Mondays 1-2 and Wednesdays 1-3 in the basement lab. Occasionally there will be exceptions when Monday holidays screw things up. We will hold office hours in the basement lab on Fridays from 1-?.
- Demography 213 is a **required** pass/not pass course. To get a degree in demography you must pass this class; to pass this class you must complete **all** assignments.
- During the first 5 weeks, the class will consists of weekly assignments, which are introduced on Monday and due on the following Friday. The first 2 of these 5 weeks we will make use of an online R training program called Data Camp `http://www.datacamp.com`.
- The final 9 weeks of class will be taken up with longer projects which will sometimes be done in groups of two or three and will often include a short class presentation.
- During the first 5 weeks of the course we will roughly follow the *optional but recommended* book *R in Action* by Robert Kabacoff `http://www.manning.com/kabacoff`.
- The real "text book" for this course is *Introduction to R* by Venebles and Smith `http://www.demog.berkeley.edu/Refs/R-intro.pdf`. *Introduction to R* is a excellent, short and free. It contains just about everything that you need, but many students find it frustratingly terse.
- We will also use *Getting Started with Rstudio* by John Verzani. `http://proquest.safaribooksonline.com/9781449314798`
- The course website has links to a few other R resources, some of which are licensed by the campus and can only be accessed from campus **or** through the library proxy server. You'll want to set this up as soon as class is over. `http://www.lib.berkeley.edu/using-the-libraries/proxy-server`

1 Linux/MATE basics

1.1 Logging in and arranging the desktop

If you have not already done so, now would be a good time to login to a Linux workstation. Trying out the commands described below is a much more effective way to learn than *thinking about trying out the commands described below*

Upon logging in, you should be presented with a graphic user interface – also called a "desktop". The Linux desktop "looks and feels" like any other modern computer and should take very little getting used to. (There are some very significant differences between the Linux windowing system and those of other operating systems – but you'll catch on to those differences as they become important to you) One difference, that will become evident shortly is a greater reliance on typed commands as opposed to pointing and clicking. As you will soon discover, the Linux terminal is much more sophisticated than the command line under windows – If it weren't, we would not use it.

1.2 Typographical conventions

Some typographical conventions will make things easier to understand. In what follows in this document and all others that *I* give you for this class:

- " **@:>** " will be used to represent the ***Linux prompt***. The Linux prompt appears in a terminal window to indicate that the command interpreter is ready. When you are instructed to type:

  @:> `ls`

  you do not type @:> –it is there to indicate that what follows is to be typed at the Linux prompt–generally, inside a ***terminal window***. In real life the Linux prompt does not look exactly like **@:>** . If you were me, typing this handout, it would look like this:

  ```
  carlm@immigrant ~213/Week1\$ .
  ```

- When describing Linux commands **optional *arguments*** will be shown between brackets like <`this`>. For example, If we were learning about the command, `cd`, an example like this:

  @:> `cd` <`213/Week9`>

  means that you *may* type something after `cd` and thereby change it's behavior. In the above case, typing `cd 213/Week9` will cause the shell to make `213/Week9` the new ***current working directory***; typing `cd` alone will cause the shell to make your ***home directory*** the new current working directory.

1.3 Terminal window ?

Terminal Windows also known as ***xterm windows***, ***Mate Terminal Windows***, ***ROXTerm*** and several other variants, are windows in which the Linux ***shell*** is running. The shell, or command interpreter is the part of Linux that presents the command prompt, **@:>** , reads a command you type and then does what *it thinks you wanted.*

Opening terminal windows Since you will want to open terminal windows fairly frequently, it would probably make sense to create a ***launcher*** on your ***panel***. Here's how to do it:
Applications →Accessories →(drag to panel) Terminal
which is a typographical convention meaning:

1. click on the Applications menu
2. click on the Accessories sub menu
3. click **and hold** the Terminal item and drag it to the ***panel*** (the gray bar at the top of the screen that holds things like this)– then release the mouse button.

Now you can launch a terminal window process by simply clicking on the launcher in the panel. When the mouse pauses over the Terminal icon it will shout: Terminal: Use the Command Line.
Of course you will still be able launch terminal windows from the menu.
If you do not find a suitable Terminal application under Accessories, try System Tools.

In your copious free time, you will probably want to modify some of the features of you ***terminal window*** application.

When it first comes up there may be a menu bar from which you can select Edit→Current Profile. If not, hold down the RIGHT BUTTON , somewhere over the main part of the terminal windows (not on the frame) and a dialog box will appear – at the bottom is a "show Menu bar" check box. After you change things to suit your preferences, check that box to make the menu bar go away – you can use that space to do science.

- Pick a better color scheme: A reasonable color choice is Black on Light Yellow but suit yourself.
- change the font size: The default font is kind of small for people over 30–or people who read alot as children. To increase the font size hit `ctrl` +`+` to increase the font size in the current terminal window. NOTE `+` implies `shift` . To shrink the font, hit `ctrl` +`-` (no `Shift` this time)

It is from within a terminal window that you generally perform the 12 most important Linux commands. These commands are described in the *The "twelve" most important Linux commands* `lab.demog.berkeley.edu/12important` (hereafter, 12UC) . You will need to use a few of them to complete this weeks assignment.

2 Organizing your life with directories

Now that we have mastered the terminal window, let's use if for something moderately useful. Each week in this course you may be asked to download something (like a demonstration program) and then to create perhaps other files. In order to keep straight what you are working on, wouldn't it be nice to have a set of directories with clever names already in place so you won't have to think about it?

1. Create a subdirectory of your home directory called ***213***

   @:> cd

   @:> mkdir 213

2. Create 12 subdirectories of your new 213 directory – one to hold each week's work.

   @:> cd ~/213

   @:> mkdir Week{1,2,3,4,5,6,7,8,9,10,11,12}

3. Copy a file called `demonstration.r` from carlm's `213/Week1` directory into your new Week1 directory. Use ***TAB completion*** to save valuable nanoseconds.

   @:> cd ~/213/W TAB 1

   @:> cp ~carlm/213/We TAB 1/demon TAB

4. Use the ***gedit*** editor to (yes) edit your new copy of the `demonstration.r` file.

   launch a new terminal window and in it type:

   @:> gedit ~/213/Week1/demonstration.r

5. Launch an R session into which you can drop R commands.

   @:> cd ~/213/Week1

   @:> R

6. Read ponder the `demonstration.r` file as and practice the fine art of dragging and dropping from the gedit window to the R session window.

   The trick is to select with the LEFT BUTTON and drop with the MIDDLE BUTTON. There is no "copy" step as there is in other desktops, just select and drop.

3 For NEXT WEEK

By Monday, visit `http://datacamp.com` and work through the entire ***Introduction to R*** course: the Basics, Vectors, Matrices, DataFrames and Lists. It *should* take no more than five hours.

Send email to carlm@demog.berkeley.edu when you are finished or your head hurts too much to go on.