

Demography 213

Graphics

Carl Mason
carlm@demog.berkeley.edu

October 19, 2011

Contents

1	Read Section 12 of <i>Introduction to R</i>	1
2	Saving printing and comparing your graphs	2
2.1	Comparing plots on screen	2
2.2	Quick and dirty printing	2
2.3	Multiple plots on one page	3
3	Postscript files	4
4	Assignment	4

Abstract

This week we will learn about R's graphics capabilities. Graphics are among the best features of R. The ability to add elements to graphs one at a time and to control just about every aspect of printing can lead to some pretty cool graphs. The assignment for this week builds on last week's IPUMS project. First, you should graph your result from last week in a different and hopefully better way. Second you will create an original and significant contribution to wall on the second floor.

1 Read Section 12 of *Introduction to R*

Read Chapter 12 (on Graphics) and Section 8.1, *R as a set of statistical tables* (page 33), in *Introduction to R*. The *Graphical Procedures* section contains a great deal of information, not all of which is perfectly intuitive. While you read it, be aware that you can use it as a reference as long as you understand what's in it. Don't try to memorize every graphic function and parameter, but understand the basic idea of how you create a plot and add lines, points and text to it.

2 Saving printing and comparing your graphs

There are a couple of things that could have been mentioned in Chapter 12 of the *Introduction to R* but were not. Here is far from exhaustive list:

2.1 Comparing plots on screen

To launch a graphics window in order to compare two different graphs on screen, type the command:

```
■ > X11()
```

This will launch an X11 *graphics device* and make it *current*. You can live a pretty full and rewarding life without ever using the `X11()` command because the first time you execute a *high level* plotting command, an X11 device will be created automatically and subsequent *high level* graphics commands will overwrite whatever is in it. It is handy to know about the `X11()` command, if you want to draw several graphs and compare them on the screen. In other words, if you have a graph on the screen and you want to create *another* graph in a separate window in order to compare the two – then you will want to use the `X11()` command. If you don't, then your second graph will overwrite your first and that will be fine with you.

2.2 Quick and dirty printing

The `X11()` driver does not have a `print` or a `save` button. In order to print (or save) the graph that you have produced on the screen, you can use these commands:

- To print the image to a printer
- `> dev.print(command='lpr -Pprintername')`
- To save the image as a postscript file
- `> dev.copy2eps(file="filename.eps")`

Note before committing `dev.print()` and `dev.copy2eps()` to memory, that these are **not** the best ways to print and save graphics. They are useful for quick and dirty work, but it is much easier to produce dissertation adviser quality output using the `postscript()` driver described in this weeks reading assignment

2.3 Multiple plots on one page

There are two ways to put multiple figures on a single graphics device (or “page”). The one described in the book is the older simpler technique. It involves using `par(mfrow=c(n,m))` to divide the plot area into $n*m$ equal sized boxes.

I might, reluctantly, show the newer way in class. The advantages of the newer way are

- It allows you to draw figures of different sizes on the same sheet. (which I never do)
- It allows you to redraw one of the figures on the page without redrawing the others (saves you a second or two now and then – probably less time than it took you to read this).

The disadvantage is that they are more confusing, and completely incompatible with the old way of doing things. So note that this technique exists, just in case you want to use it someday – but most folks around here still use the older but still generally adequate method described in Section 12.5.4 of *An Introduction to R*.

Here is the newer way of drawing multiple graphs on single page/screen.

- To split the screen into a 2X2 grid:
 - `> split.screen(figs=c(2,2))`
- To split the lower left box horizontally into 2 equal boxes:
 - `> split.screen(figs=c(2,1),screen=3)`
- To stick a boring histogram in the top half of the lower left quarter of the page:
 - `> screen(5); hist(rnorm(100))`
- To undo all the screen splitting stuff and return to normal behavior:
 - `> close.screen(all.screen=T)`

3 Postscript files

There are a lot of graphics file formats in popular use today. The right one for your work, depends on whether you plan to use your graphic creation on a web page, or in a publication, or in an informal document or in some sort of presentation software, or some other use.

Because postscript is *vector graphic* format, it can be edited and/or converted into just about any other format. Because the format is relatively old, the files tend not to take up too much space (and if they do, they can be converted). These two features make it an excellent default format to store graphics in.

For information on converting postscript to other formats, see the man pages on:

- `convert` for converting postscript files to raster formats like .jpg, .gif, or .png.
- `pstoedit` for converting postscript files to other vector formats such as .fig (for editing in xfig) or .svg.
- `ps2pdf` for converting postscript files to .pdf

4 Assignment

As you have done many times now... work; launch emacs from the new directory so as to:

- create a new directory for this week's work
- cd into the new (WeekN) directory and launch emacs so as to create a buffer whose name ends in `.r`;

Begin where you left off last week: Rather than importing a `demonstration.r` file from the course website, this time, you should start by importing the file that **you created last week** – the one that reads the IPUMS data that you extracted; calculates a rate of some sort; and creates a graph. You will probably need to rerun your stata code from last week in order to create the `.dta` file that R will read.

Improve last week's graph: Visit the following websites in search for inspiration and for a new and different way to illuminate the exciting contribution to human learning which you produced last week.

- <http://www.statmethods.net/graphs/boxplot.html>
- <http://addictedtor.free.fr/graphiques/>

After exploring these sites, apply what you have learned to dramatically improve last week's plot **and save it as a postscript file**. Please create an entirely different sort of plot from the one you used last week—don't just fiddle at the margins.

Create something new and beautiful: With IPUMS data as the starting point, create **at least one** graph that has artistic rather than (or in addition to) scientific merit. That is, create a graph that you will be proud to see every time you go to the bathroom. Again, the websites above can help give you ideas.