An introduction to R and pmg

Statistics involves a fair number of computations that can be made much more convenient using either a calculator or a computer. Although the basic TI-83 or 84 series of calculators can do most of the statistical computations expected of you this semester, we spend a good deal of time learning a new software package to do statistical analysis. We do this for several reasons: a calculator is not practical for large data sets that would be encountered in real life, a statistical software package can offer many more directions for analysis than a calculator, and in any job situation a computer is the tool you would be expected to use to solve the problem.

For this class we use the R software (`www.r-project.org`) which is free and runs on all the major operating systems.

To download a copy of R visit `cran.r-project.org`.

R is used by typing command into a command line, unlike most other software you have used recently. (It is similar to MATLAB, if you know what that means.) Although it may be bit harder to learn than other software packages, it is this professor's opinion that it is worthwhile learning. To make life a bit easier we will use the PMG (`www.math.csi.cuny.edu/pmg`) graphical interface for R.

This is installed in the lab, but if you are at home and want to install it on a windows machine, you can type the following command into an R session:

```
source("http://www.math.csi.cuny.edu/pmg/installpmg.R")
```

The pmg GUI is software being developed that although *a bit buggy*, can make most of the computations we do quite accessible.

This project is meant to introduce you to use R to make some plots that summarize a data set. We will refer to the PMG GUI to do so.

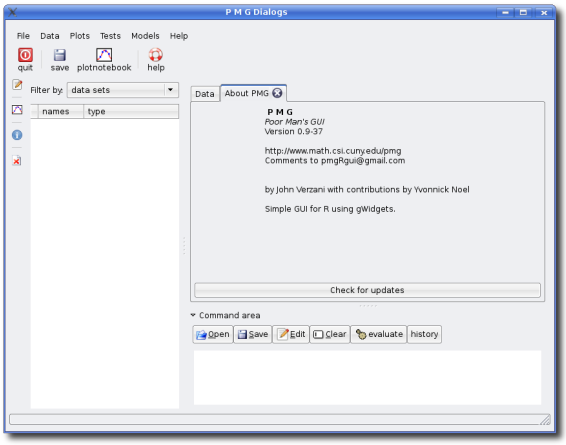

Figure 1: The PMG GUI on startup

An introduction to R and pmg

1 Starting R and the pmg GUI

R is started by clicking on the desktop icon.

R starts up in a mode that uses a base window to hold other windows. These other windows include a command line (where commands are typed), and possible windows to hold graphs or help pages. In this project, we will only use this command line to start the PMG GUI. Go the command line and type

```
library(pmg)
```

Now minimize the R window. The pmg GUI should popup looking something like Figure 1. If you don't minimize the main R window, the pmg GUI windows will appear under the big R window.

The key features are from left to right: a area where variables are shown, a notebook to hold some dialogs like a spreadsheet-like area for entering data under the Data tab, and a blurb about PMG and a command area under this notebook. On the top is a menubar giving access to many functions, and a toolbar with access to some basic features.

2 Entering a data set

Let's see two ways to enter a data set. Suppose we find we surveyed 10 students in the class and found that their cars got the following miles per gallon

```
22 18 28 14 19 23 27 31 26 19
```

We can enter this data into R a few ways. The command line usage would be to type

```
mpg = c(22, 18, 28, 14, 19, 23, 27, 31, 26, 19)
```

into the command area and then click the `evaluate` button (Figure 2).

This combines the data into a data set and then assigns the value to a variable `mpg`.

Clicking the `evaluate` button will cause the command to be evaluated, and a `mpg` variable to appear on the left.

Alternatively, the data can be entered into the Data area. Click on the Data tab and you see a primitive spreadsheet-like area. Enter in values by double clicking in a cell, and then using the down arrow key to create new cells at the bottom, as in Figure 2. (The ENTER key does not create a new cell if it is needed.)

The funny "nan" is actually funnier under windows. Not much to do about that. It indicates that the entry is not a number. Once the data is entered in, there are a few ways to access it.

Luckily, as we'll see, we don't need to enter in data too often, as R has many built-in data sets, and means to download data off internet sites.

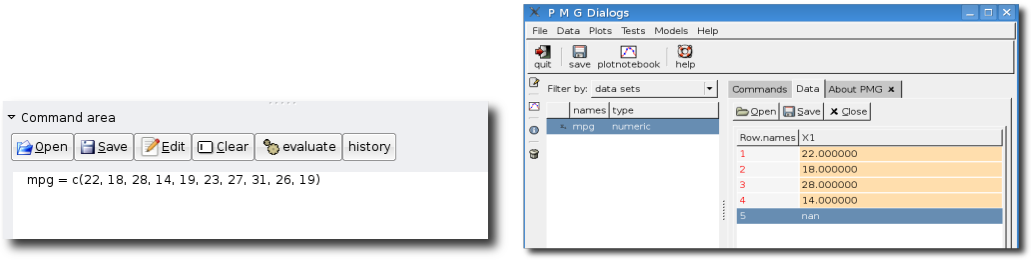

Figure 2: The left shows how to enter data in the command area, the right shows how data is entered into the spreadsheet area.

3 Using the data

Lets see how to make a table from the data. The easiest way is to use the command line. Switch to the command tab, and enter the command

```
table(mpg)
```

```
mpg
14 18 19 22 23 26 27 28 31
 1  1  2  1  1  1  1  1  1
```

Again, entering involves typing it in, and clicking `evaluate`. You may need to clear out the old command. If you previously typed in the `mpg` variable the values print out. Data sets are stored in variable names so that they can be referred to again and again.

If you want to use the data from the spreadsheet, you must first save it to make it accessible. To do so, right click on its tab and follow the save dialog. If you save it as "mydata" the values can be referred to as `mydata$X1`. (A little clunky, but we won't typically type this.)

4 Histograms

A histogram is a graphical summary of a data set. If you haven't seen this already, we learn that they show more area where there is more data. (Area is proportional to frequency within a "cell.") Hence we can see where the data is concentrated.

Creating histograms can be straightforward, but there are few different ways.

From the command area we could type

```
hist(mpg)
```

There are many arguments for the histogram function `hist`. The `Plots::Univariate::histogram` menu item opens a dialog for adjusting several of the possible arguments. We'll reserve this for more advanced uses.

Finally, we show how to use the `Lattice Explorer` to quickly make a histogram or other plots. Close the open plot window, and then under the `Plots` menu item select the `Lattice Explorer`. A new window pops up. Change the combo box from `densityplot` to `histogram`, and the window looks like Figure 3.

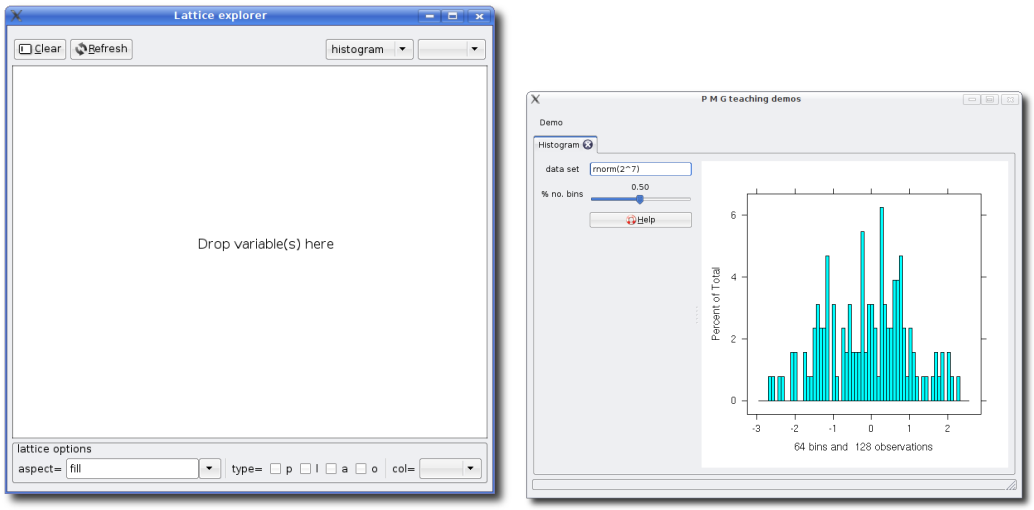

Figure 3: Left graphic is `Lattice explorer` after changing the combo box to `histogram`. The left graphic shows the histogram bin selection demo. The data is randomly produced through the use of `rnorm(100)`. Adjust the number of bins through their proportion by moving the slider. What is a good percent? In this case, 50% seems too large.

Now drag the mpg variable and drop it in the main box of the lattice explorer. A histogram should be drawn. You can also drag a variable from the Data window. Simply grab the column header and drag. You may need to clear out variables before doing so.

4.1 Loading data sets

The lattice explorer makes it quick and easy to investigate data sets. We'll do this now.

First we load some different built-in data sets. The first one is built-into R's base data set collection. To load it, go to the Data menu and select `Load data set...`. When the dialog pops up, scroll down and select the `mtcars` data set by double clicking on it. In a second or two the data set appears in the pmg GUI. Other data sets are loaded the same way. For now close that dialog. The next data set we load is from an add-on package. R has over 1,700 add-on packages. A few are built in, most are downloaded off the internet.

The MASS package is built in and has many interesting data sets. To load that go to the `File` menu and select `Load package...`. When the dialog appears, double click on `MASS`. Now go back to the `Load data set...` dialog and notice that there are some more datasets to choose from. We want to load `Cars93`.

1. In the `mtcars` data set is a `mpg` variable. Make a histogram. Based on the histogram, what is the apparent range of the data?
2. In the `Cars93` data set are variables `MPG.city` and `MPG.highway`. Make histograms of both, and compare their shapes. Are they similar, different? How so?
3. Make a histogram of `MPG.highway` using the lattice explorer. Now change the droplist from `histogram` to `densityplot` What changed? How is a density plot similar to a histogram?
4. Now drag the `Cylinders` variable onto your lattice explorer plot of `MPG.highway`. The data is then broken up by the number of cylinders. Can you tell which cars get better mileage?
5. Repeat using `Origin` in place of `Cylinders` Which origin gets better gas mileage?
6. Look at histograms of all the variables in `mtcars`. Do any have a `bell-shaped` histogram? If so, which ones.
7. Look at densityplots of all the **numeric** variables in `Cars93`. Are any of then bell-shaped?
8. Close the lattice explorer, and open the plotnotebook. The icons of the left of the data sets in the PMG GUI are drop areas. Drag the variable `Cylinders` from the `Cars93` data set onto the one that looks like a plot. A barplot is made, as this data is categorical. (These are called "factors" in R.) Can you determine how many cars are in the Cars93 data set? Which cylinder-type is most represented?
9. Repeat with `DriveTrain`. Which is the most widely represented?

5 Cell (or bin) size in histograms

The shape of a histogram is determined by the selection of the bins. Each rectangle drawn has an area proportional to the number of points that are in the given bin. If there are too many bins, the histogram is too erratic, if there are too few bins, the main features are smoothed away. There is a need to find a balance between the rough and smooth.

The `Plots::Teaching demos` menu opens a window that allows us to explore the relationship (The right graphic in Figure 3). Do so, then select `Histogram bin selection` from the `Demo` menu.

We need to type in a data set and then enter. In the Figure, we use a *random* data set of 128 points given by the command `rnorm(2^7)`. Enter the command, then type ENTER.

Now move the slider to adjust the proportion of bins.

5.1 Problems

1. With 128 data points, the default proportion of 0.5 looks too rough – 64 bins is too many. Adjust the slider to the left to find a reasonable number. What do you get?

2. Repeat with just 16 data points (2^4). Is 8 bins too many? too few? Just right? Comment. If it doesn't look good, find a value that is good.

3. Now make a table. Find values that look good for 4,8,16,32,64,128,356,512 and 1024 data points. Write down the values and the proportion. What can you say in general. As the size goes up, the proportion does what?