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. Although it may be bit harder to learn than other software packages, it is the 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 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.

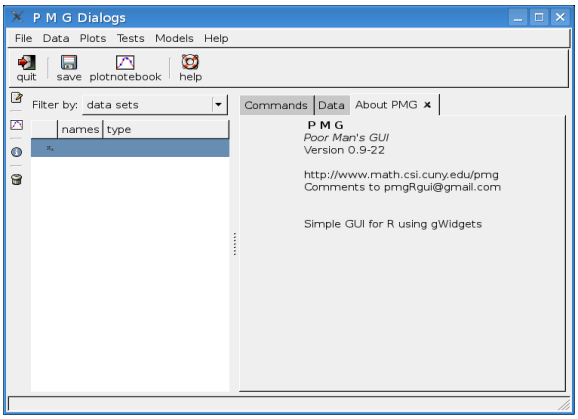

Figure 1: The PMG GUI on startup

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. 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. The key features are from left to right: a area where variables are shown, a command line under the Commands tab, a spreadsheet-like area for entering data under the Data tab, and a blurb about PMG. 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 into R in two ways. Without a GUI, one would enter the following command into R

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

This combines the data into a data set and then assigns the value to a variable `mpg`. We could do this in the "Commands" tab as in Figure 2. Clicking the "evaluate"

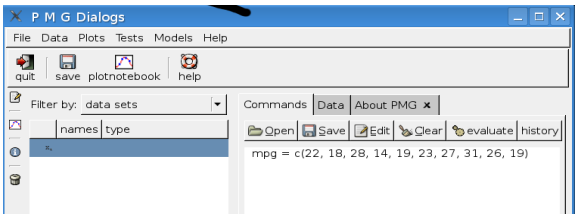

Figure 2: Entering data into the PMG command area

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 3. (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.

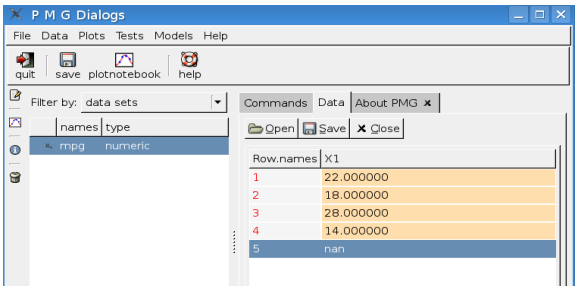

Figure 3: Entering data into the Data area

3 Using the data

Lets see how to make a stem and leaf plot of the data. The easiest way is to use the command line. Switch to the command tab, and enter the command

```
> mpg

 [1] 22 18 28 14 19 23 27 31 26 19
```

(Don't enter the ">", that is a prompt that appears if you do this at the R command line.) 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. We want to make a stem-and-leaf diagram. The function `stem()` will do so. We simply need to enter

```
> stem(mpg)

  The decimal point is 1 digit(s) to the right of the |
  1 | 4
  1 | 899
  2 | 23
  2 | 678
  3 | 1
```

into the command area.

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

Creating histograms can be straightforward. We show three ways.

First, to use the PMG graphic device, open up a plot notebook by clicking the "plotnotebook" toolbar item.

As with a stem-and-leaf plot, a histogram can be created by a command, in this case `hist(mpg)`. Evaluate this at the command line to see. There are ways to modify the command to adjust the size of the bins, etc. but we will describe how to do this in a simple way.

Under the Plots menu item is "Univariate" and then "histogram". Select this. It is supposed to open a tab on the right side, but may popup a separate window with a dialog for creating histograms (Figure 4)

Figure 4: Histogram dialog found under Plots::univariate::histogram

Now drag (or simple type the name) the `mpg` variable to the "x=" text area and click "OK". A histogram should appear in the graphic device. The dialog allows a few attributes to be changed. Play around with them to see what they do.

Finally, we show how to use the "Lattice Explorer" to make a histogram. Close the plot window, and then under the "Plots" menu item select the "Lattice Explorer". A new window pops. Change the drop list from Dotplot to histogram, and the window looks like Figure 5.

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.

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

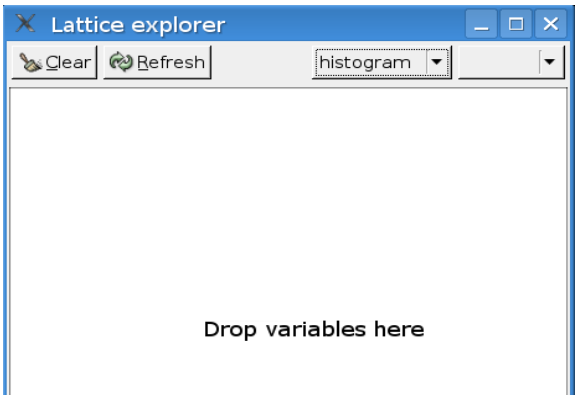

Figure 5: Lattice explorer after changing droplist to "histogram."

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 literally a 1,000 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?

10. Load some other data set and explore the variables. Describe their shape.