Journal of Statistics Education

Using R in Introductory Statistics Courses with the pmg Graphical User Interface

John Verzani
CUNY / College of Staten Island

Journal of Statistics Education Volume 16, Number 1 (2008),
www.amstat.org/publications/jse/v16n1/verzani.html

Key Words: statistics software, statistical computing, R, introductory statistics, EDA, exploratory data analysis

Abstract

The pmg add-on package for the open source statistics software R is described. This package provides a simple to use graphical user interface (GUI) that allows introductory statistics students, without advanced computing skills, to quickly create the graphical and numeric summaries expected of them.

1. Introduction

As much as there is such a thing as a consensus among educators, there is a consensus that introductory statistics courses should be accompanied by some sort of technology component to ease the burden of performing calculations and allow students access to real-world data sets (MAA CUPM, recommendation 5), (GAISE report, recommendations 2 and 5), (Chance, et al, 2007). What the appropriate technology is, however, seems to have no consensus. Coming across my desk have been books for using all of the following: the TI-83, Excel, ActivStats, SPSS, SAS, MINITAB, JMP, and R, still leaving room for books on Stata, Fathom, DataDesk and others. What the proper choice is depends on many factors. This article promotes the use of R (R Development Core Team, 2006) in these introductory courses.

Why R? There are many good reasons. R is an open-source project, hence free for the students and the institution to use. R is multi-platform (Windows, Mac OS X, and Linux). It has excellent graphics. It has an extensive collection of add-on packages, such as pmg. It is used by many statisticians, and so can grow with the student as they progress. Finally, at its heart, R is a modern, consistent programming language. This last point, however, means that R is relatively hard to learn, as students need to master a series of commands in order to be productive. In this author's opinion, this is R's primary drawback for usage at the introductory level. Still R has its advocates, as illustrated by Hodgess (2004) in this journal.

A graphical user interface (GUI) can help ease the learning of sometimes obscure commands. R

currently has a few GUIs. For Windows and Mac OS X systems, the default installation has a GUI that can assist with administrative tasks. The JGR project (Helbig and Urbanek) provides a JAVA-based alternative to these. For help with statistical tasks, the Rcmdr (Fox) add-on package, which is available for all the major platforms, provides a series of dialogs giving access to R's statistical functions. (In this article, the wide sense of a "dialog" as a window or page that allows a user to control the options for a particular feature of the GUI is used. This is in contrast to a "dialog box" which is often the name used for simple popup windows for showing messages or requesting confirmations.)

The pmg project has a design goal similar to Rcmdr. It too provides several dialogs for collecting arguments to function calls and displaying these function calls as illustrations of R commands. However, pmg also provides some "dynamic" dialogs which allow the student to use drag and drop to quickly and easily explore a data set or perform a statistical analysis. These dialogs, which were inspired by the interesting commercial software package Fathom (Key Press), are the focus of the examples in this article.

In the author's experience of using R in computer labs accompanying introductory statistics courses, the use of pmg greatly simplifies the initial learning curve required of the students. He has had good success with the software with non-calculus based introductory courses which target a population consisting primarily of nursing students.

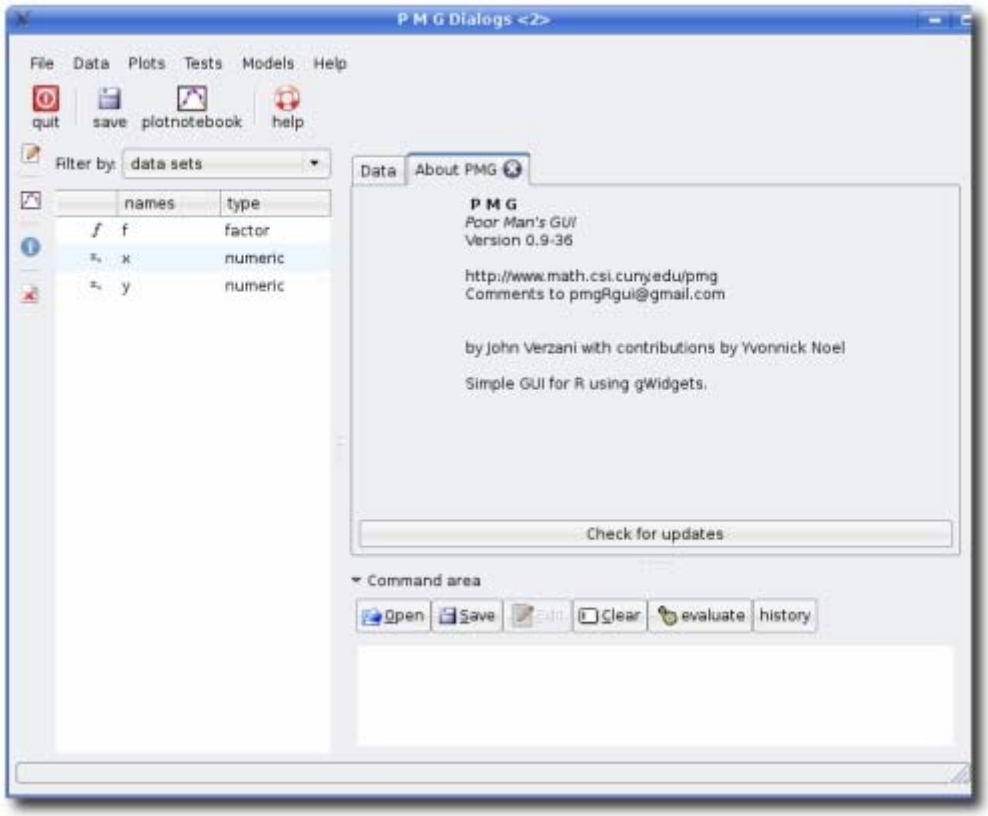


Figure 1: The basic pmg GUI at start up. The menu bar and tool bar are found, as usual, at the top of the GUI. On the left is a variable browser indicating that in this R session, three variables have been defined (f, x, and y). The right-hand side contains a notebook for displaying additional dialogs. At startup, the Data tab is present and an About PMG tab. The Data tab offers a primitive spread sheet interface to rectangular data sets (data frames). Beneath this area is the Command area where R commands may be typed and evaluated.

2. Starting pmg

R is attractive since it supports multiple platforms. Unfortunately, this makes it difficult to shortly detail the steps involved for the installation of pmg. As such, we postpone that disussion for Appendix A.

Assuming pmg is installed properly, the main GUI is created by loading the package with the following command:

```
library(pmg)
```

Startup can take a few moments. Within Windows, R, by default, runs in a multiple document interface (MDI) mode, meaning all windows reside in a single parent window. Unfortunately, the windows created by pmg do not show up in this parent window, so they can easily be covered by the parent window. To avoid this, it is recommended that the main R window be minimized after starting, or R be used in the single document interface (SDI) mode. Once started, the pmg GUI looks something like Figure 1.

Most of the main features of the GUI are familiar. At the top is a menu bar giving access to several dialogs, below this a toolbar with a few prominent buttons. On the left, one finds set of small icons, which can be used as drop targets for values dragged from the variable browser. This is to the immediate right of these drop targets. The variable browser shows the name and the type of objects that are easily accessible to the user. In Figure 1 there are just three variables f, x and y defined. To the right of the variable browser is a "notebook" with tabs to choose between its pages. A permanent tab (no close icon) is labled Data. This page provides a somewhat primitive spreadsheet interface for viewing and editing data frames (R's terminology for rectangular data with each column holding a variable, and each row an experimental unit). The About PMG tab shows version information and can be removed by clicking on the close icon. Most of the dialogs called from the menu bar are added as pages to this notebook to keep down on desktop clutter. Finally, underneath the dialog notebook is a command area that allows a user to interact directly with R.

We illustrate the use of the pmg GUI by showing how some of the data analysis covered in two previous *Journal of Statistics Education* articles could have been done.

3. Example: Fourth grade foot measurements

A recent *Journal of Statistics Education* article by Mary C. Meyer looked at a data set on fourth-grade foot sizes. We follow her analysis, illustrating how pmg could be used to provide her results.

First we must load the data. The *Journal of Statistics Education* provides a url for the data: http://www.amstat.org/publications/jse/datasets/kidsfeet.dat. There is an Import data set... dialog under the Data menu (Figure 2) that allows the user to specify a url to download. The figure shows the above url copied-and-pasted into the appropriate text entry area. Clicking the OK button will cause R to download the file. By default, R will guess how to read the file, in this case choosing a text file. The second graphic in Figure 2 shows how the data will be read in after the value for header (indicating whether a header line containing variable names is present in the file) was adjusted to FALSE.

Figure 2: Import data set... dialog showing how to import a data set from the *Journal of Statistics Education* archive. The left figure, shows where to specify the url of the data set. After the OK button is clicked, the file is downloaded and read into R. Afterwards adjustments can be made as to how the file is parsed. In this case, the lack of a header line to specify variable names was specified by changing the value for header to FALSE.

As there is no header line suggesting names for the variables, they are assigned the default values of V1, V2, ... If desired, we can change these with the dialog opened by Data::Manipulate::Edit data frame properties. This dialog first asks for a selection of a data frame. After choosing the newly created feet data frame, one can rename the variables contained within. We chose the names month, year, length, width, gender, measured, and handedness (to record the childs's dominant hand). The left graphic of Figure 3 shows the process half way through. (The factor type is how R stores categorical variables.)

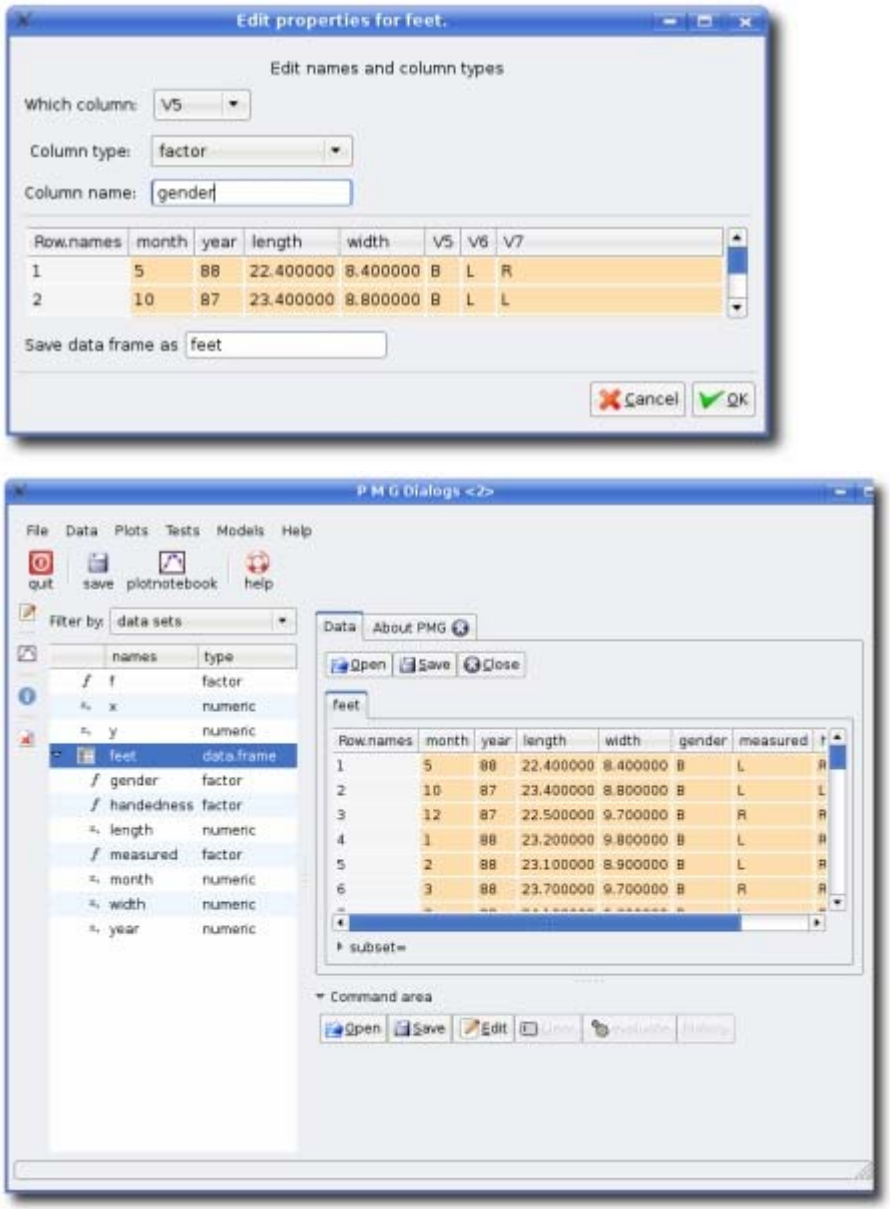
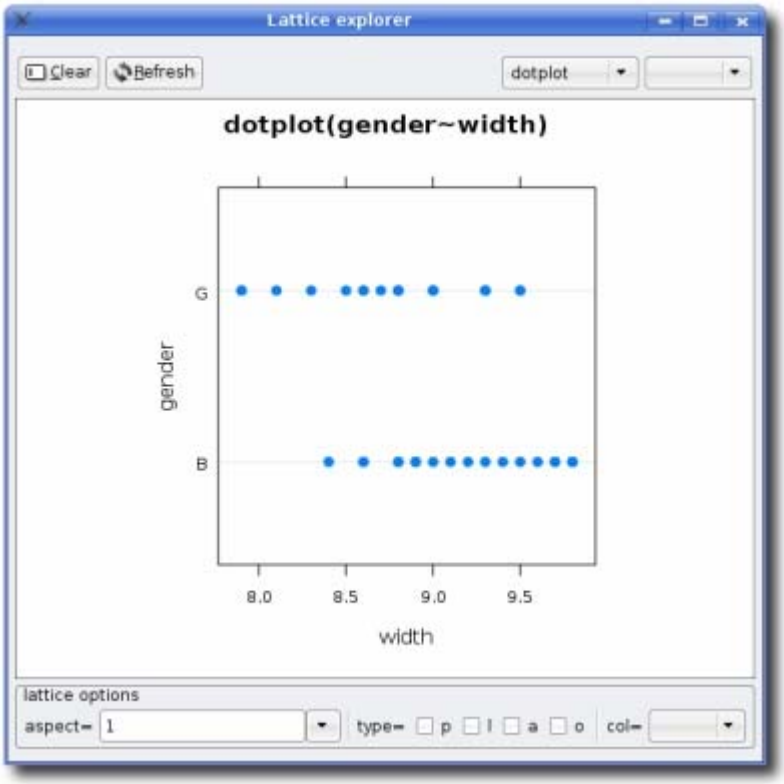


Figure 3: Two dialogs for working with data frames. The left graphic shows a dialog to edit the name and variable type properties of a data frame. In this case, the names of the feet data frame are being changed from the default names prefaced with a "V" to more descriptive names. The right graphic shows the Data tab with the feet data frame after its variables were renamed. The names appear as column headers. The same names are also shown in the variable browser after expanding the entry for feet. These names may be dragged and dropped into other dialogs.

If desired, we can now look at the data. Clicking on the Data tab opens an area that allows the viewing and editing of data frames. Either drag the feet variable from the variable browser and drop on the open button, or click open and select the feet data frame. The data area should show the new column names, as in Figure 3.

Typically, the first task in exploratory data analysis is to graph the data. The Lattice explorer dialog under the Plots menu opens up a window where variables can be dragged and subsequently plotted using R's lattice graphics. The variables may be dragged from the Data area (drag the column names) or from

the variable browser. The first variable dragged is the primary variable plotted, the second is used to group the data. For some graphs a third variable may be used to group even further. If we drag first width, and then gender we will see two density plots for width -- one for each level of gender. Selecting the dotplot graph in the upper right, produces the dot plot of the data shown in the leftmost graphic of Figure 4.



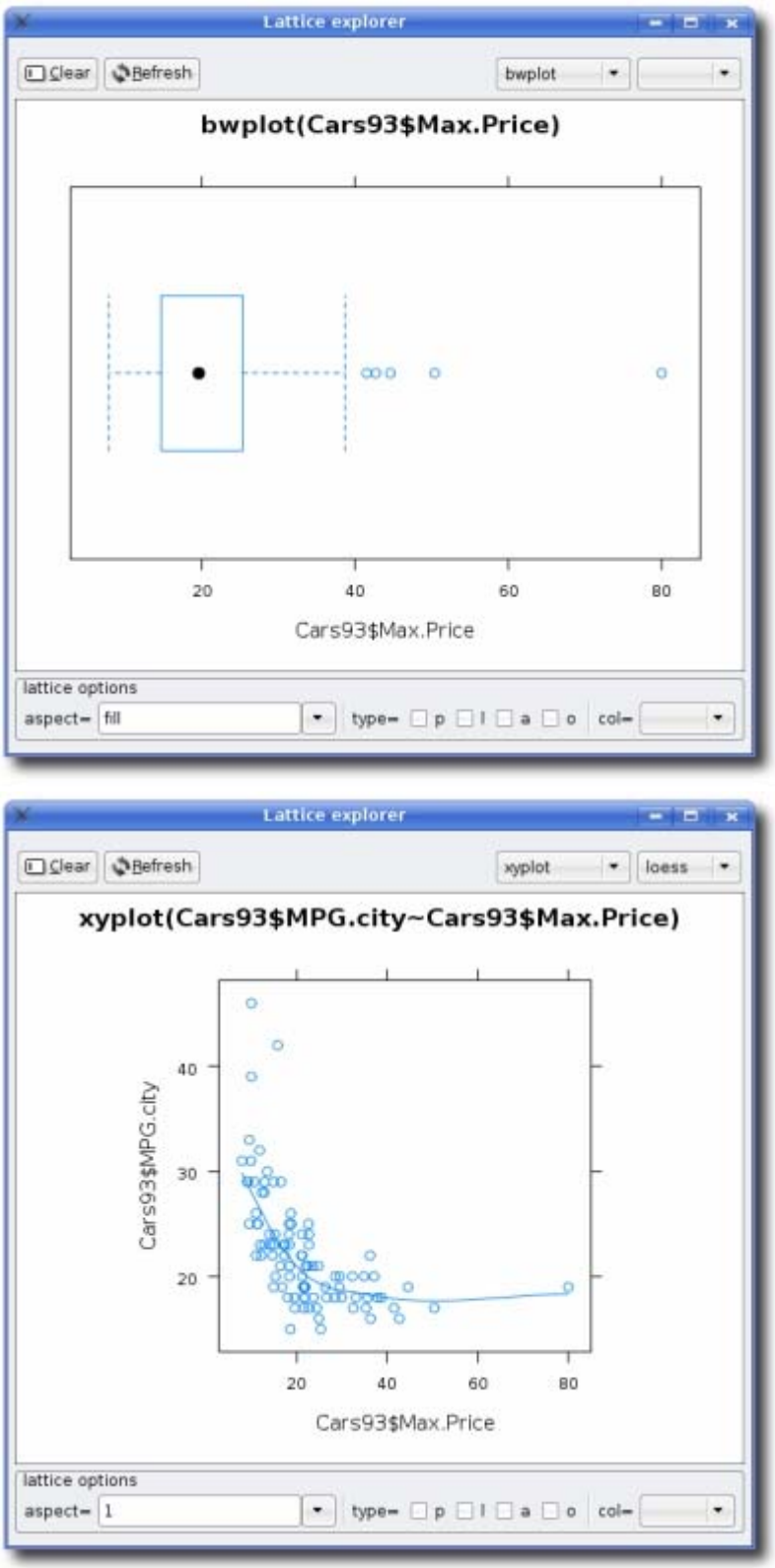


Figure 4: The Lattice explorer showing several types of plots created by dragging and dropping variables onto the main graphic window. The left-most plot is a dotplot, using the feet data frame, of the width data for both levels of the gender variable. In this instance, the variables were dragged from the column headers of the data viewer (as indicated by the lack of a "$"). The first variable dragged (width) specifies the data, the second (gender) is used to group the data. If desired, a second grouping can be

achieved by dragging over one more factor. The middle Box and whisker plot shows the right skew of the Max.price variable in the Cars93 data set. Outliers are marked individually using the 1.5 IQR formula. The right-most scatterplot (xyplot) of miles per gallon by maximum price for the Cars93 data set. The loess trend line is added by adjusting the panel popup.

Other graphs can be drawn. Changing the plot to bwplot produces box-and-whisker plots. If a new data set is to be used, the clear button resets the graphic so that the next dropped variable is the new primary variable. The R code to produce the graph is the title for the graphic.

There are other dialogs to produce the familiar graphical displays discussed in an introductory statistics course found under the Plots menu. These use the base R graphics. The dialogs are not interactive, but have the advantage that the code to produce the graphics can be easily edited, customized, and saved as it appears in the command area.

If the iplots package is installed, a dialog is provided for calling its functions. This package provides a Java based way to interactively explore a data set using various graphs. Since it requires a user to have an implementation of Java, which may be quite a large download, it is not required by pmg.

Following the author of the article, we next perform a *t*-test of width for each level of gender. We use the Dynamic tests dialog under the Tests menu. This dialog allows significance tests to be performed by dragging and dropping data to the dialog. We want to perform a two sample *t*-test. As the data is stored with a grouping variable recording gender, and not two variables each with numeric values, we use the 2-sample t-test, formula value for the popup. We then drag and drop the variables width and gender to the data: line. The basic result looks like Figure 5. The author carried out a one-sided test of H_A:$\mu > 0$, which is done within this dialog by changing the popup value to greater than. The dialog also prints out the appropriate confidence interval, in this case a one-sided one. The default confidence level is 95%, but this can be adjusted by clicking on the bold-faced 95 label and entering the desired value.

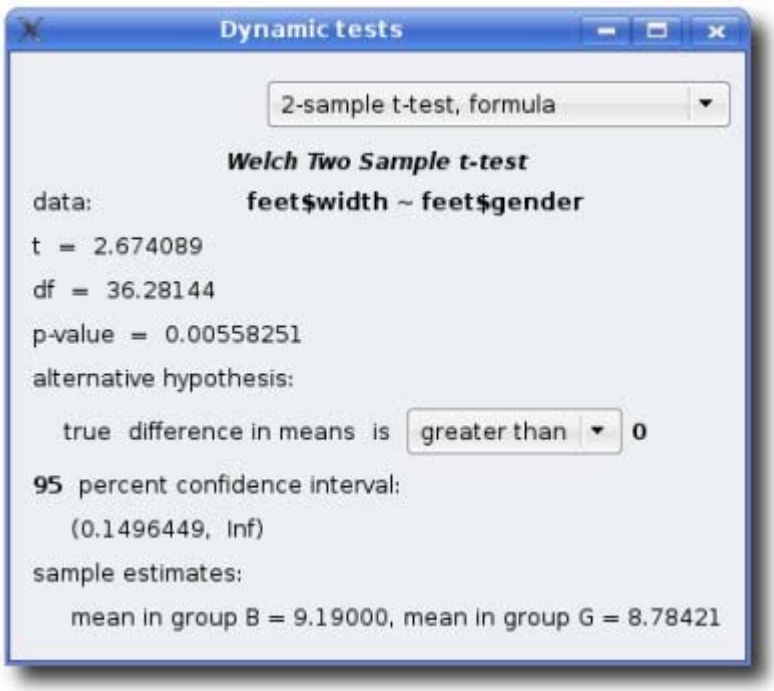


Figure 5: The Dynamic tests dialog showing a two-sample *t*-test for the width data with the gender

variable providing the grouping. A one-sided test is performed by changing the popup to greater than. The computed *p*-value, after rounding, is 0.0056. For this usage, the variable names were dragged from the variable browser on the left of the main GUI and dropped on the bold-faced data: area of the dialog. They could also be typed in by clicking in the same bold face area and editing. This can be useful if transformations of the data (e.g. sqrt) are desired.

The author then carries out an analysis of covariance on the data, using length as a covariate and gender as a predictor variable. This model can be fit using the Dynamic models dialog under the Models menu. Loading this dialog and selecting linear regression from the popup, allows us to drag the variables to the proper place. Dragging width to the response area and both length and gender to the predictor variable will fit the model

$$\text{width}_i = \beta_0 + \beta_1 \text{length}_i + \beta_2 1_{\text{gender}} = G + \varepsilon_i$$

.

The default contrasts for unordered factors in an ANOVA are treatment contrasts. The value of "B" for gender is chosen as the baseline by R here, leaving the dummy variable different from that of the original article. The fitted model is seen to be

$$\hat{\mu_{\text{width}}} = 4.0832 + (0.2210)\text{length} + (-0.2325)1_{\text{gender=G}}.$$

(Hence, β_0 is different from the articles which is $\beta_0 + \beta_2$ in our case. If we replace the dummy variable with the one in the article, we can get the same estimates, of course. This requires using relevel which isn't straightforward to do with this dialog, but not difficult from the command line.) This dialog only fits regression models without interactions. R has a model formula syntax for specifying more complicated linear models. A separate dialog allows these to be specified and fit.

p-value for the two-sided test $H_0 : \beta_2 = 0$ is \$0.08055 (Figure 6). The one-sided *p*-value from the article, \$0.040275, showing the borderline significance of the gender variable, is found by mentally halving that found from the two-sided test.

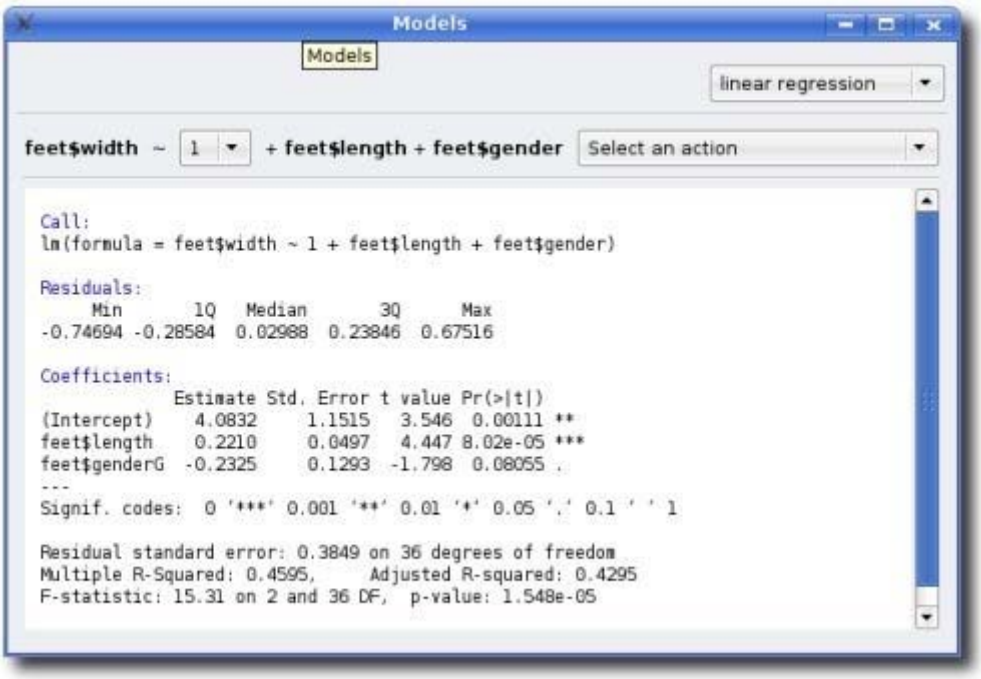


Figure 6: The Models dialog showing the output of an ANCOVA model for width modeled by length and gender. This dialog was used by dragging variable names from the variable browser and dropping them where the formula is specified on the top left. Only main effects can be added by dragging and dropping (without editing the formula by hand). More complicated models may be fit using a dialog found under Models::Regression::lm} menu item.

4. Example: Cars93

Our next example uses one of R's built-in data sets. R has many such data sets which are useful for illustrating different statistical features. We use one of this author's favorite, a data set provided by Robin Lock in the first *Journal of Statistics Education*. This data set, Cars93, is conveniently provided by the MASS package.

R has numerous add-on packages. A few come bundled with R, others, like pmg itself, are available through a set of web sites known as CRAN, and may be downloaded from within an R session. The MASS package is such a useful and important package it comes bundled with R, and is loaded when pmg is. (If it weren't, there is a dialog to load an external package under the File menu.) To load a data set, we can use the Load data set... dialog under the Data menu (Figure 7). We scroll through the output until we find the data set we desire and then double click on that line to load the data set.

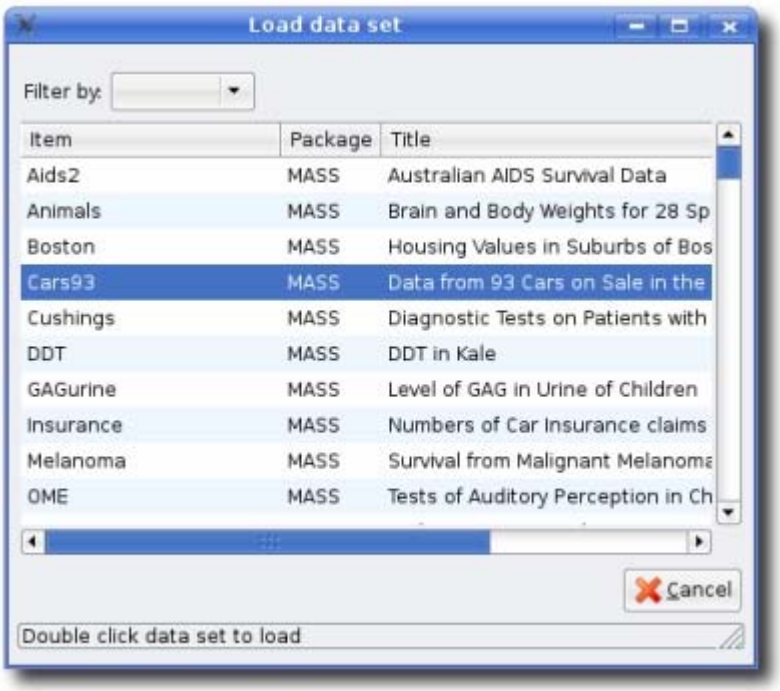


Figure 7: Dialog to load a data set from an installed package. Many of R's add-on packages have accompanying data sets including the MASS package which contains the Cars93 data set. This data frame will appear in the variable browser after one double clicks on the entry.

The Lock article suggests several explorations of interest. We indicate how they would be done using pmg.

Number 13 suggests: "Box-whisker plot: PRICE or MPG variables give good examples of somewhat skewed data with potential outliers among the upper fences." Simply dragging the variable to a cleared lattice explorer window will make a density plot of the variable. Changing the popup to bwplot will produce the desired plot. As is customary, potential outliers are marked individually. The middle graphic in Figure 4 shows the boxplot. Once the large values are understood, they can be identified within the data set by other means.

Number 14 suggests: "Small sample confidence interval for a mean: Look at HPW or RPM within one TYPE of car. Different students may be assigned different TYPEs." Subsetting can be done using R's [notation or the subset function. However, these powerful commands can be difficult for students to learn, although pmg provides a limited interface to subset. Alternatively, this task can be done with the Dynamic tests dialog taking advantage of the subset= feature of the data viewer. The "Dynamic" dialogs, when used with variables dragged from the data viewer, update immediately to reflect changes in the data viewer. Drag the Cars93 data set to the Data tab and drop it on the Open button. Open up the one-sample *t*-test dialog and drag the column variable for {Horsepower to the data: line. A 95% confidence interval for the entire data set is shown. Now in the subset= area of the data viewer, click the expander and you have a way to adjust the data considered in the confidence interval calculation. Selecting the variable Type for subset= and adjusting which level of Type is considered by clicking the arrows, will cause different confidence intervals to be shown in the dialog. Figure 8 shows the use of subset= to show only data for compact cars.

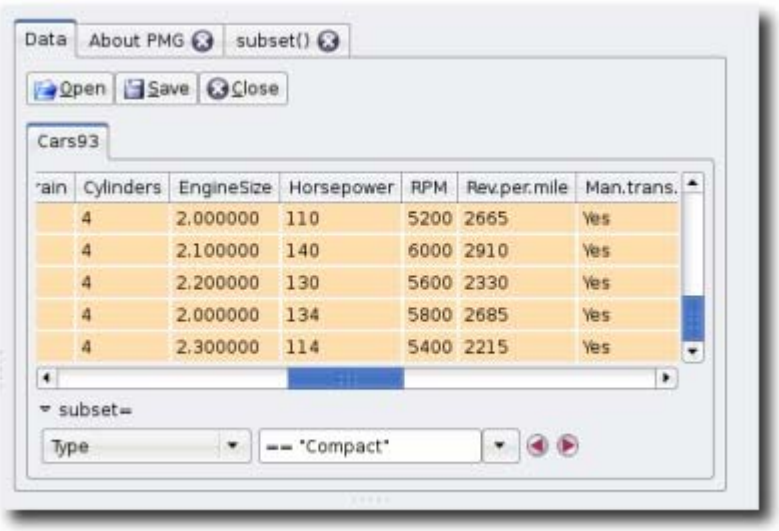


Figure 8: Close up of the Data tab showing the Cars93 data frame using the subset= feature. The Type column header was chosen using the popup. By clicking the arrows, just the data for different levels of the Type variable will be shown. If a column header is dragged to one of the dynamic dialogs, such as Dynamic tests, then just the filtered data will be used in the computations. In this case, only the data for compact cars will be used.

Number 15 suggests: "Difference in means between two independent samples: Compare PRICE levels between DOMESTIC and FOREIGN cars. Also watch out for significant differences in the variances between these two groups." For this task, an exploratory plot is in order. As the domestic versus foreign information is stored in the more non-US-centric titled Origin variable as a factor, we can use the lattice explorer to make graphs comparing the shapes and spread of the data. The bwplot suggests that the means may be similar, but the spreads are not. If a formal *t*-test is desired, the Dynamic tests dialog can be used as before. Although in this case, since the data is stored in two variables and is not suited for a formula, the 2-sample t-test item is used.

Number 16 suggests: "One-way ANOVA for difference in means: Check out city MPG ratings between the three DRIVETRAIN categories." This can be done with the Oneway ANOVA popup in the Dynamic tests dialog. Simply drag the two variables to the data: line. Alternatively, the linear regression popup in the Dynamic models dialog may be used with City.MPG as the response, and factor DriveTrain the predictor.

Number 17 suggests: "Contingency table: Construct and analyze a two-way table of AIRBAGS by FOREIGN/DOMESTIC." Contingency tables can be constructed with the table dialog under the Data::Univariate summaries menu, despite this being a bivariate problem. Simply set one variable for x and the other for y. For more complicated cases the Cross tabulation dialog under the Data::Bivariate summaries menu can be used. The left graphic in Figure 9 shows how to use this dialog for this task. This dialog calls R's xtab function which uses a formula to specify the variables. The exact specification of the formula depends on how the data is stored. In this case, the left side of the formula is left blank, indicating to R that the frequencies should be tabulated. Were the data tabulated, the counts would be specified. The right side specifies the cross-classifying variables.

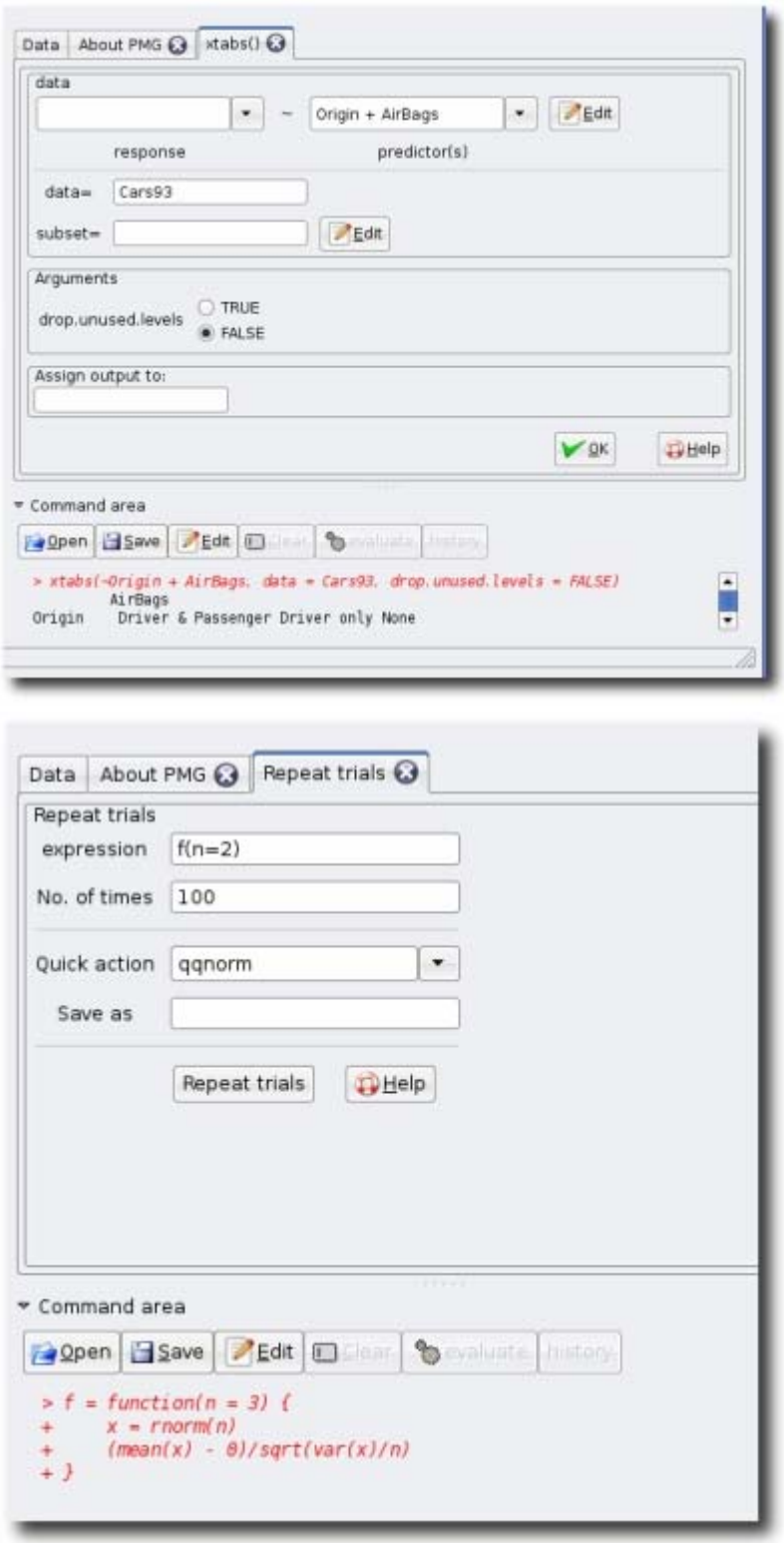


Figure 9: Two dialogs that appear in the notebook. The left graphic shows how cross tabulations using R's model formula syntax can be performed. For the xtabs function both AirBags and Origin are used as "predictors" while the response variable is left empty, as the data has not been previously tabulated. The Repeat trials dialog calls a function to repeat an R expression a specified number of times. In this example, a user-defined function f, defined in the command area, is called 100 times, producing a simulation of the sampling distribution of the *t*-statistic for 1 degree of freedom. The data produced can

be stored, or, as shown, graphed using a Quick action. In this case, the qqnorm function (quantile-normal plot) was entered in.

Number 18 suggests: "Scatterplot: Plot PRICE by MPG. Identify any unusual points." A scatterplot can be produced using with the Lattice explorer and the xyplot selection for the plot popup. Adding a regression line is done using the panel popup. The right-most graphic in Figure 4 shows the graph with a loess fit overlayed. Identifying points with the mouse, while possible in R, is not implemented in pmg.

5. Teaching with R and pmg

The pmg GUI provides a few useful dialogs for teaching with R.

The Plots::Teaching demos menu item launches a window allowing the selection of several standard teaching demonstrations interactively illustrating concepts such as the central limit theorem, confidence intervals and statistical power. Many of these were contributed by Yvonnick Noel. The pmg package is written using the gWidgets package, which allows such demos to be written in a relatively easy manner and integrating additional demos within pmg is a simple matter.

Figure 9 shows the dialog called by Data::Simulation::Repeat trials. This dialog will repeat an R expression a specified number of times, providing an alternative to for loops or R's replicate function.

In the figure, the dialog is used to repeat a function call 100 times. The function is previously defined in the command area. R's strength is its ease in allowing users to define new functions. The definition of the function f is:

```
f = function(n=3) {
x = rnorm(n)
(mean(x) - 0)/sqrt( var(x)/n )
}
```

The keyword function is used to indicate a function is being defined. The last value evaluated is the return value, in this case the familiar *t*-statistic for a random sample from a normal population of size n. By default n will be 3, but in the usage n was specified to be 2.

Further, although R has several useful data sets, and as illustrated, data sets in the proper format can be downloaded into R easily, one may also want to provide functions or other R code for students to use. This can be facilitated using R's source function combined with its url function to reference a file on a website, as is shown in the Appendix with a script to install pmg for Windows users. For example, if a more complicated function were to be defined for a simulation, then this function could be written to a file and a student could read its contents into their R session with a command typed into the command area such as

```
source("http://www.yoursite.edu/rexamples/ex1.R")
```

(Although, the author has found that asking students to type--as opposed to copy-and-pasting--in a url can be asking for trouble.)

Finally, here is a solution for how data can be shared between students and teacher. For instance, to aggregate simulations from several different students. One could set up a Google documents account for a class that all the students can share in. Students can save their data in CSV format using the Data::Write data as CSV file... dialog and then import this into their Google documents account. They

can then copy and paste their work into a shared document. If needed, they can add a column indicating who they are. Finally, Google documents allows spreadsheets to be "published" as CSV files publicly using complicated urls. These urls can be used with the \menuItem{Import data set...} dialog to import the data into R, as shown previously with the feet data set.

In the author's experience, pmg has greatly simplified the use of R with his least prepared student populations. The pmg website (\url{http://www.math.csi.cuny.edu/pmg}) has a few projects that have been used in a one-hour lab setting. The first exploratory lab asks the students to produce histograms and barplots of variables along with a grouping variable. Previously, the commands to produce these graphics were overwhelming to teach as the majority of this population of students have trouble grasping the concepts of variables, variables within data frames, and functions. Now the bulk of the class time can be spent discussing the graphics, and not how to produce the graphics.

6. Limitations and future plans

While the pmg package can be used effectively in a classroom or lab setting to facilitate the use of R, it is not professionally produced and has a few idiosyncrancies especially with the spreadsheet interface to the data frame. It is anticipated that with an increased user base these idiosyncrancies and bugs will be found and ironed out. In addition, there are some areas that should be addressed as time goes on. Some examples follow. For instance, the GUI needs to be localized to allow translations into other languges. Additions to the collection of teaching demos are also planned. Many R commands are based around the concept of a "formula," yet pmg does not have much facility to assist in specfying these formulas in a helpful, guided way. Finding a flexible and intituitive metaphor for entering formulas is a current goal. As graphics are an integral part of an introductory statistics course, providing interfaces to R's many graph-producing commands is of interest. Dialogs for the base graphics, lattice graphics and the add-on plots package are incorporated into pmg, but an interface for the newer, modular, ggplot2 package is in the process of being written. Additionally, as many students struggle to learn the powerful syntax of R for manipulating data sets, the current set of dialogs for data manipulations could be substantially improved. Finally, although pmg provides an alternative to the command line for many tasks, there is no escaping the command line for advanced usage. (Nor any desire, as this is a key reason to use R.) A redesigned command line with features to aid report writing would be an excellent addition to the GUI.

Development of pmg is primarily a one-person affair, but need not be. It is not so difficult to add new features to the GUI and the author is very open to doing so. Additionally, there is some limited ability to add menubar entries and new dialogs to the GUI within an R session. Details are in the vignette that accompany the package.

Appendix A: Installing pmg

Installation of pmg is straightforward, and explained below, provided R and the GTK+ toolkit are already installed. R may be downloaded from the CRAN link found on http://www.r-project.org and installed as other software packages are installed. Currently R is around 30 megabytes in size for the Windows version. GTK+ (GTK+ Team) is a multi-platform programming toolkit that relies on a collection of system libraries to draw graphical user interfaces. It is typically installed with most Linux distributions, but requires separate installation under Windows and Mac OS X platforms.

Under Windows, the GTK+ libraries can downloaded from http://gladewin32.sourceforge.net as a self installing file. Alternately, a script on the pmg website can be called from within an R session to install

GTK+ via the command
source("http://www.math.csi.cuny.edu/pmg/installpmg.R")

This will also install pmg.

For Mac OS X users, the GTK+ libraries can be found pre-compiled and packaged to install at http://r.research.att.com/gtk2-runtime.dmg. The GTK+ libraries require X11 to be installed. As of version 10.4, this is an add-on package that comes with the OS X system install disks.

If R and GTK+ are in place, then the installation of pmg is done similarly to all other CRAN packages. The install.packages command will do the work:
install.packages("pmg")

(Windows users can use the menu bar of the main R GUI to do this.) That command will install pmg and several other CRAN packages that pmg depends on.

If that doesn't work, more details are found on pmg's webpage and the web page for RGtk2 (http://www.ggobi.org/rgtk2).

References

B. Chance, D. Ben-Zvi, J. Garfield, and E. Medina. The role of technology in improving student learning of statistics. Technology Innovations in Statistics Education, 1(1), 2007. http://repositories.cdlib. org/uclastat/cts/tise/vol1/iss1/art2.

J. Fox. The R Commander: A Basic-Statistics GUI for R. http://socserv.mcmaster.ca/jfox/Misc/ Rcmdr/.

GAISE group. Guidelines for Assessment and Instruction in Statistics Education College Report, 2005. http://www.amstat.org/education/gaise/GAISECollege.htm.

GTK+ Team. The GIMP Toolkit. htp://www.gtk.org.

M. Helbig and S. Urbanek. Java GUI for R. http://stats.math.uni-augsburg.de/JGR/.

E. Hodgess. A Computer Evolution in Teaching Undergraduate Time Series. Journal of Statistics Education, 12(3), 2004. http://www.amstat.org/publications/jse/v12n3/hodgess.html.

Key Press. Fathom Dynamic Data Software. http://www.keypress.com.

R. Lock. 1993 New Car Data. Journal of Statistics Education, 1(1), 1993. http://www.amstat.org/ publications/jse/v1n1/datasets.lock.html.

MAA CUPM. Undergraduate programs and courses in the mathematical sciences: CUPM curriculum guide 2004. http://www.maa.org/cupm/curr_guide.html.

M. C. Meyer. Wider Shoes for Wider Feet? Journal of Statistics Education, 14(1), 2006. http: //www.amstat.org/publications/jse/v14n1/datasets.meyer.html.

R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2006. URL http://www.R-project.org. ISBN 3-900051- 07-0.

John Verzani
Department of Mathematics
CUNY / College of Staten Island
2800 Victory Boulevard 1S-215
Staten Island, NY 10314
verzani@math.csi.cuny.edu

Volume 16 (2008) | Archive | Index | Data Archive | Information Service | Editorial Board | Guidelines for Authors | Guidelines for Data Contributors | Home Page | Contact JSE | ASA Publications