

Tools for Teaching Regression Concepts Using Dynamic Graphics

Mervyn G. Marasinghe and
William M. Duckworth
Department of Statistics
Iowa State University
Ames, IA50011

Tae-Sung Shin
StatSoft, Inc.
2300 E 14th St.
Tulsa, OK 74104

September 26, 2003

Abstract

This paper extends work on the construction of instructional modules that use graphical and simulation techniques for teaching statistical concepts (Marasinghe et al., 1996, Iversen and Marasinghe, 2001). These modules consist of two components: a software part and a lesson part. A computer program written in LISP-STAT with a highly interactive user interface that the instructor and the students can use for exploring various ideas and concepts comprises the software part. The lesson part is a prototype document providing guidance for instructors for creating their own lessons using the software module. This includes a description of concepts to be covered, instructions on how to use the module and some exercises. The regression modules described here are designed to illustrate various concepts associated with regression model fitting such as the use of residuals and other case diagnostics to check for model adequacy, the assessment of the effects of transforming the response variable on the regression fit using well-known diagnostic plots and the use of statistics to measure effects of collinearity on model selection.

Keywords: Education, statistics instruction, active learning, simulation, regression diagnostics, Lisp-Stat

1 Introduction

1.1 Instructional Modules

A primary goal of a new NSF-supported project targeted at improving the effectiveness of undergraduate statistics courses has been to develop instructional tools

that are easily adaptable for general use independent of specific courses. These tools will provide statistics instructors with the capability of supplementing their teaching methods by presenting and illustrating statistical concepts more effectively than is possible using conventional instructional methods. They allow exploration of important concepts using graphical displays that are easy to use and provide instantaneous visual feedback, thus encouraging *active* learning.

Under this project, we have developed a number of instructional modules designed to illustrate various statistical concepts and provide important insights into the application of these concepts. Use of these modules in teaching will enable students to get more meaningful learning experiences than are otherwise possible from traditional instructional methods alone. Exercises designed to accompany these modules will further reinforce these learning experiences. The *software component* of these modules employs a combination of state-of-the-art computing hardware, a statistical programming environment, high resolution color graphics, computer simulation, and a highly interactive user interface. The modules can be used both as classroom demonstration tools and as self-paced exercises for individual students or small groups. The *instructional component* is a prototype lesson that includes a description of concepts to be covered, instructions on how to use the module, and some exercises with solutions. Course coordinators and instructors can modify or augment these lesson plans to create their own class assignments.

1.2 Implementation of the Software Components

The software components of the modules described here have been programmed in the Lisp-Stat language. The system provides a computing environment with the following important features:

- It is a powerful object-oriented programming language that allows rapid prototyping and development.
- It allows development of modules using true dynamic graphics.
- It allows links to routines written in Fortran or C.
- It can be installed on the different instructional computing platforms in use at many educational institutions (Unix workstations, Macintoshes, and Desktop PC's running Windows software).
- It has a large core of statistical functions available.

Another important advantage of using Lisp-Stat for developing software is that, for noncommercial applications, it is available without cost. For a good introduction to Lisp-Stat and other Lisp-Stat based software, see de Leeuw (1994).

We have developed user interfaces that are consistent across modules. Students need only execute one command to start a module. All further interaction is through

a mouse-driven interface. One may click on a push button to initiate action, click and drag on a menu button to select items from a pull-down menu or click and move slide-bars to set values of various parameters. Some of the dynamic graphical techniques used in the modules are discussed in detail in Cook and Weisberg (1994).

1.3 Lesson Prototypes for Instructional Modules

For each instructional module, we have developed, in \LaTeX format, a sample instructional document. These documents consist of

- An introduction to and a short description of the important statistical concept(s) to be covered.
- Objectives for the instructional module.
- Instructions on how to execute the software component of the module.
- Warm-up exercises to familiarize the student with the software module.
- Formal exercises and questions requiring execution of the software (usually following a reasonably precise set of instructions), careful thought, interpretation of results, and explanation of conclusions.
- Notes for the instructor, including comments on what is expected to be understood from doing the formal exercises.

Instructors can either use these documents or customize them for their own purposes and classes. The documents (perhaps without the notes for the instructor) could be made available to the student as a classroom handout or online.

1.4 Related Work

Becker, Cleveland, and Wilks (1987) review many of the dynamic graphic techniques such as linking and selecting. Many of these techniques have been extensively used in the exploration and the analysis of multivariate data (see Cleveland and McGill (1988)). Uses for rotation and animation techniques in regression diagnostics were first developed by Cook and Weisberg (1989), and later a more elaborate presentation appeared in Cook and Weisberg (1994).

However, until recently, little use of these techniques has been made for instructional purposes though the possibilities are many. Saunders (1986) describes the use of dynamic graphics to produce “moving visualizations designed to introduce difficult concepts, reinforce mathematical ideas, and explore the techniques of probability modeling.” The visualizations were used in a distance-learning television program produced by the BBC. The paper, for example, describes visualizations to illustrate

the effect that parameter changes have on binomial, Poisson and bivariate normal distributions.

Another example is the STEPS project, a UK consortium based in Glasgow involving nine departments in seven universities, which was conceived for the purpose of developing problem-based teaching and learning materials for statistics. Modules were developed around specific problems in several subject areas and incorporated computational and graphical tools to assist in the exploration of the statistical ideas encountered in solving these problems. The problem-based approach was used to motivate interest in the students by selecting problems in their own areas of study and also because it allowed integration with other more standard laboratory materials. Trumbo (1994) uses graphics and simulation to illustrate elementary probability concepts, using programs written in QuickBasic and giving rough equivalents in Minitab.

Even fewer examples appear in the area of regression where the techniques and concepts are ideally suited for illustration using dynamic graphics. Apart from the introductory examples in Tierney (1991) and the work of Cook and Weisberg (1994), a paper that describes software constructed for this purpose is Nurhonen and Puntanen (1992).

Anderson and Dayton (1995) present program code written to demonstrate various features available in the Lisp-Stat language. Rather than providing a complete set of educational modules, they illustrate how this language can be adapted for building instructional tools to enhance the teaching of various concepts in regression. That approach may not be suitable for instructors averse to learning the Lisp-Stat language at the required level for developing modules. On the other hand, no programming ability is required to use the modules presented in this paper. In addition, the lesson plans provided serve as templates for instructors for creating their own lessons. Users of these modules may find different ways to use them other than those suggested in the paper or the lessons. The present modules are also extensible in that anyone may add a module to the system without affecting current users of the modules.

A commercially available multimedia package `ActivStats` written by Paul Velleman contains several interactive modules that illustrate various concepts in regression. However, these modules are built around an introductory statistics course, and instructors may find integrating `ActivStats` with a regression course difficult. Also, `ActivStats` is not free.

There are several archives of JAVA applets available for regression. An example is the VESTAC system at www.kuleuven.ac.be/ucs/java/index.htm, described in Darius, Michiels, Raeymaekers, Ottoy, and Thas (2002). While some of these JAVA applets may be useful for demonstrating the same regression concepts covered in this paper, no lessons are provided with the applets. Using the JAVA applets requires a network connection to access the applets. This can result in slower execution and a less responsive interface than locally installed modules. Modules described in this paper are based on Lisp-Stat and can be customized using the Lisp-Stat language; however, the JAVA applets cannot be customized.

A more recent addition to the literature is the Cook and Weisberg (1999) regression text accompanied by the software package *Arc*, also written in Lisp-Stat. Although designed specifically for performing analyses described in the book, *Arc* could conceivably be used independently to demonstrate selected regression concepts. A more useful suggestion for instructors using the above text and *Arc*, is to use the modules described in this paper as a supplement to their course.

2 The Regression Modules

In this article we describe a set of instructional modules that have been specifically designed to aid instructors in teaching introductory and advanced regression methodology. For simplicity of use and ease of construction, the software component consists of modules each covering a different but interrelated set of regression concepts. The five regression modules are described in detail in Sections 2.2 through 2.6.

Section 3 describes how to obtain the software described in this section. Readers may find it helpful to install and use the software while following the descriptions in Sections 2.2 through 2.6.

2.1 Regression Concepts

Statistics students from different disciplines often have various levels of ability in mastering some of the more sophisticated modern regression techniques they learn from textbooks and lectures. Interpretation of results produced by standard software packages such as case deletion statistics or residual plots is a complex task and can lead to confusion and improper use of such statistics. An adequate understanding of the concepts behind these techniques and some experience in using them can help alleviate this problem. However, students enrolled in regression courses cannot acquire the necessary experience through involvement in real statistical data analysis projects alone because these projects are often too time consuming to incorporate more than a few into a course.

Some examples of the kinds of concepts that students need to understand for developing an ability to interpret regression computations are:

- Different graphical displays highlight different relationships among variables. To explain the relationship between a Y and an X variable in a multiple regression model, both Y and X must be adjusted for effects of other explanatory variables in the model.
- Interpretation of case statistics from a regression analysis is often far from straightforward. Not only does the presence of more than one extreme observation tend to complicate their interpretation, different case statistics provide fundamentally different types of diagnostic information.

- If diagnostic plots indicate departures from assumptions (e.g. nonnormality of residuals, nonhomogeneous errors, etc.), a transformation of either the Y or the X variable (or both) may result in variables that more closely match the model assumptions.

While standard homework assignments, lab exercises, and class projects provide mechanical practice, illustrate a few ideas, and help to introduce these concepts to students, true *understanding* comes only after extensive experience. Repeated interaction with graphical displays of simulation results allows students to be exposed to pseudo-experiences that will facilitate their understanding of important concepts. For example, by plotting straight lines fitted to regression data generated from a known fixed model repeatedly, the student can understand the difference between fitted models and the *true* model. In another example, students can dynamically change the value of a data point to study the effect of that point on the fitted regression line. By controlling the way that a change is effected (e.g., by holding X fixed and changing Y in simple linear regression), the program can highlight a single property of a case statistic (such as the simple fact that leverages depend only on the predictors). Examples like these illustrate the potential benefit of incorporating modules like ours into a traditional regression course.

2.2 Exploring Least Squares Fitting

The `regteach1` module is useful for illustrating some of the fundamental concepts related to simple linear regression. For example:

- A single summary statistic like a correlation coefficient or R^2 , by itself, cannot be used to interpret the strength of a relationship. A scatterplot is an essential component of examining the relationship between two variables.
- It is important to understand the idea of least squares fitting. It can be demonstrated that one may not always be minimizing the sum of squared deviations when “fitting a line by eye”.
- Magnitudes of the residuals from a regression depend on the fitted line. Thus a simple residual plot can reveal a lot about the goodness of the fit.

The frame on the left-hand side of Figure 1 shows the initial view of the `regteach1` module window. Changing the slope and intercept values in the slide-bars will dynamically change the slope and intercept of the plotted line and update the numerical coefficients in the box above the plot. Pushing the **Select Data** menu button and using the resulting pull-down menu allows the user to select a data set from a list of simple regression data sets. The frame on the right-hand side of Figure 1 shows the `regteach1` module after the `OAK-SEEDLING` data set has been selected.

——— Figure 1 appears here ———

Pushing the **Residuals** button will produce an additional window containing a plot of residuals from the current fitted line. The signed deviations are displayed as line segments drawn from a zero baseline, plotted at the corresponding x -values. The residuals will change as the slope and intercept are changed in the main window. This can be used to demonstrate dynamically the dependence of the magnitudes of the residuals on the fitted line (such as the fact that as the fit improves the residuals get smaller), as well as identifying various patterns in the residuals (such as curvature) for diagnosing the fit. Figure 2 shows the residual plots corresponding to two different lines fitted to a data set.

——— Figure 2 appears here ———

The user can attempt to find the best fitting line to the selected data “by eye” (graphically) using the sliders to change the slope and intercept; the plotted line will be updated dynamically. When satisfied with the line fitted “by eye”, the user can push the **Fit Least Squares** button to display the least squares line fitted exactly to the data and the resulting summary statistics (including regression coefficients, sum of squared residuals, and the residual standard deviation) as shown in Figure 3. Comparing the least squares regression line to several different “by eye” fits provides the user the opportunity to learn the meaning of the least squares criterion for fitting a line to data. The **Fit Least Squares** button also creates a scatterplot of the residuals plotted against the X variable for making a visual assessment of the fit (e.g., curvature exhibited by the residuals may indicate nonlinearity in the data). The **Correlation** button will display the between X and Y . In addition to a graphical fit, a user may also attempt to fit the best line numerically, by attempting to minimize the value of the RSS, or better, by using the RSS “thermometer” displayed on the right margin of the primary window, both activated by pressing the **Residual SS** button.

——— Figure 3 appears here ———

2.3 Properties of Regression Coefficient Estimates

The `regteach2` module is constructed to illustrate some of the concepts related to statistical properties of parameter estimates in the simple linear regression model. In particular, students will have an opportunity to examine the sampling distribution of the slope estimate dynamically. Some of the ideas that can be explored with this module are:

- The fitted line passes through the “center” of the data, i.e., through the point (\bar{x}, \bar{y}) .
- Variability in the data affects the accuracy of estimation of the regression coefficients.

- The estimate of the regression slope is symmetrically distributed around the true value of the slope.
- Parameter estimates depend on the data only through a few summary statistics.

If it is assumed that the errors are a random sample from $N(0, \sigma^2)$, the slope estimate b is distributed normally with mean β , the true value of the slope, and variance σ^2/s_{xx}^2 , where $s_{xx}^2 = \sum_{i=1}^n (x_i - \bar{x})^2$. Thus it is easy to see that for fixed values of the regressor variables, the distribution of the estimate depends only on σ^2 .

The initial window of the `regteach2` module is shown in the top frame of Figure 4. A value for the sample variance σ^2 is first selected for generating data from the model (displayed in the box near the top margin of the plot) by moving the slide-bar. For each push of the `Simulate Lines` button, seven new y values corresponding to the seven x values specified are generated, and the straight line fitted to the data is plotted. By repeating this procedure one can observe dynamically the variation in the lines fitted to different data sets generated from the same model. The slopes of fitted regression lines constitute a draw from the sampling distribution of the slope estimate b . By pushing `Dist. of b` prior to starting the simulation process, the user can observe these values being accumulated in a dynamically updated histogram. As the histogram is updated with more and more samples of the estimated slope, the user is able to visualize that the distribution begins to take the appearance of a Normal distribution centered at 2, the true value of the slope.

————— Figure 4 appears here —————

The `Save 15 Lines` button, when pushed, will plot 15 such lines on the same graph for a fixed value of σ^2 and save a copy of the graph in a separate, smaller window. This is useful for comparing sets of lines obtained for different choices of σ^2 , as exhibited in Figure 5. These graphs clearly demonstrate that the variation in the slope of the lines fitted to data is dependent on the actual variability in the data.

————— Figure 5 appears here —————

2.4 Diagnostic Use of Case Statistics

Many regression programs in standard statistical packages produce a large number of the regression diagnostics proposed in the literature. However, not much attention is paid to describing or indicating how such statistics should be used or interpreted. Rather than providing students with additional tools for judging the adequacy of a fitted model, this vast array of statistics has added to the confusion of many students.

The `regteach3` module has been designed to illustrate important concepts related to the use of some of the more popular case statistics in regression analysis:

- The leverages depend only on the explanatory variables. Cases away from the centroid of the data have large leverages compared to those near the middle.
- The predicted response for cases with high leverages is largely determined by the observed response. Hence the fitted line is constrained to pass as close to the corresponding data point as possible.
- Deletion of a single point can have a large effect on the fit of a model. A case is determined to be influential if its deletion substantially affects the parameter estimates.
- Cook's D is a combination of an outlier measure and a leverage measure. An influential observation (i.e., one with a large D value) that has small leverage is an "important outlier."

When `regteach3` is started-up, the main window (top left frame in Figure 6) shows a scatterplot of a working data set along with an overlaid regression line (displayed in magenta; the regression statistics corresponding to the fitted line are also displayed in the same color). The three secondary windows display, clockwise from the main window, indexed plots of the studentized residuals, Cook's D statistics, and the leverages, respectively, computed with respect to the fitted regression line. Initially, all four windows of the module are in *selecting mode*; i.e., a point may be selected by clicking the mouse pointer near it. By pushing the **Point Moving Mode** button, the main window can be changed to the *point-moving mode*. In addition, all four plots are *linked* with each other, so that if a point is *selected* in one of the secondary windows, the serial index corresponding to the point will identify the point in *all* four windows.

————— Figure 6 appears here —————

In the point-moving mode, points in the scatterplot can be dragged into other positions; however, in `regteach3` they are constrained to move only in the vertical direction, so that the value of the X coordinate of the point moved remains fixed. To demonstrate a case statistic like *leverage*, we will first move to a secondary window and *select* one of the points indexed, say, 18 or 19. The selected point will be highlighted and labelled. These points are furthest from the "centroid" of the X values (the centroid here is the mean \bar{X}), correspond to the two largest leverages, and lie above the $2p/n$ reference line shown on the leverage plot. Now, moving to the main window, the selected point is dragged to a new location (i.e., the point will have a new Y coordinate while the corresponding X coordinate remains the same). The regression line will be recomputed and plotted in a new position. The original line will still appear on the graph (in a light green color) for comparing with the refitted line. Notice that this action leaves the leverage plot unchanged, while the other two plots, as well as the estimates of the parameters, are all updated dynamically to correspond to the line fitted to the modified data. The **Restore All** button is now used to restore

the plots to their original appearance. Then, a data point closer to the centroid of the X values, say, point 9, is selected similarly to demonstrate that such cases have comparatively smaller leverage values.

Going back to point 18 or 19, notice that when one of these is moved, the fitted line attempts to “track” its movement more closely than, say, when point 9 is moved. This can be observed either directly, by following the movement of the line while the point is being dragged, or indirectly, by noting the changes in the plot of residuals or in the parameter estimates. This effectively demonstrates that leverage measures the *weight* each point carries in the prediction calculations. Thus, when the leverage of an observed point is large, the corresponding prediction for that point attempts to move closer to or “track” the observed point. This forces the fitted line to pass closer to points with higher leverage.

Point 18 is deleted by first selecting it and then pushing the **Delete Selection** button. This will remove point 18, and straight line model is refitted to the resulting data. The change in the fit statistics is noted and the point replaced by pushing the **Restore All** button. Figure 6 and Figure 7 display the results of these operations. By repeating the same procedure with point 19, it will be revealed that fitted values are more sensitive to the deletion of case 19 than of case 18. Case 19 is said to be a more *influential* observation than case 18. This fact is reflected in the Cook’s D statistics computed for the original data and shown in Figure 6; the largest value corresponds to case 19. Cook’s D statistic is a measure of the influence an individual case has on the regression fit.

———— Figure 7 appears here ————

2.5 Examining Relationships among Regression Variables

Plotting Y against each of the explanatory variables and computing correlations among them are acceptable as simple tools for studying the relationships among these variables. For example, these may be useful in detecting collinearities involving a pair of variables. However, if incorrectly interpreted, these correlations alone may lead to misleading conclusions regarding the contribution of an explanatory variable to a regression model, particularly in the presence of other variables in the model. Moreover, in multiple regression, collinearities can involve three or more variables. To detect these relationships, simultaneous use of several diagnostic plots may be necessary.

Some of the important concepts related to understanding and interpreting relationships among regression variables that can be illustrated using the `regteach4` module are:

- A plot of Y against an X variable, called the *partial response plot*, cannot be used alone to explain the contribution of the X variable to the multiple regression model.

- A useful way to display the strength of the relationship between Y and an X variable in a model is to plot these against each other after removing the linear effects of the other explanatory variables from each. This plot is called the *added-variable plot* or the *partial regression plot*.
- Added-variable plots are useful in directly determining how individual cases affect the estimation of the corresponding regression parameters.
- Finding the best linear fit is difficult when the predictors are highly correlated. This is reflected in high *variance inflation factors* (VIFs) for some parameter estimates in the fitted model.

Figure 8 shows a $(Y, X1, X2)$ spin-plot with buttons labelled **Pitch**, **Roll**, and **Yaw** which can be used to rotate the 3-dimensional point cloud around any one of three fixed axes. Using the **Yaw** button, the point cloud can be rotated around the Y axis until the strongest fit to a straight line is observed on the plane of the computer screen. This corresponds to the fitting of a least squares plane by eye, and the line is the 2-dimensional view of that plane.

————— Figure 8 appears here —————

In **regteach4** a 2- or a 3-variable regression model with predetermined values for the X variables may be selected by pressing the corresponding button to illustrate concepts described above. The user has some control over how the data for the X variables are generated; the correlation between the $X1$ and $X2$ variables can be specified using the *Corr* slider-bar. For example, in Figure 8 the correlation has been set to .40 as can be observed in the scatterplot of $X1$ vs. $X2$. The symbols V , H , and O identify the variables plotted on each axis. When the **3-Var** button is pushed, the user is given the choice of selecting the pair of independent variables to be plotted on the H and O axes by pushing one of the buttons $X1\&X2$, $X2\&X3$, or $X1\&X3$.

The spin-plot gives the user the ability to observe various 2-dimensional projections of the data, as demonstrated in Figure 8. In particular, the more interesting projections are those that will coincide with the partial response plots. For instance, the projection shown on the spin-plot in Figure 9 is identical to the plot of Y vs. $X1$ shown on the the upper left corner of the *scatterplot matrix*. The scatterplot matrix is used in the **regteach4** module to display the relevant partial response plots simultaneously. Using the **Yaw** button the spin-plot can be rotated to obtain another projection that is identical to the plot of Y vs. $X2$, i.e., the middle plot in the top row of the scatterplot matrix.

————— Figure 9 appears here —————

In addition to the partial response plots, the scatterplot matrix also displays all pairwise plots among the X -variables. Pressing the **Scatterplot Matrix** button produces this plot, displayed here in Figures 9 through 12. Pressing the **Added Variable**

button results in the appearance of the added-variable plots shown in Figures 10, 11 and 12.

————— Figure 10 appears here —————

Figures 10 and 11 show examples where the bivariate plots described above have been constructed for samples in which predictors X_1 and X_2 were generated with correlations of 0.04 (inducing low multicollinearity) and 0.96 (inducing high multicollinearity), respectively. The partial response plots in Figure 10 indicate a strong linear relationship between Y and X_2 and a weak linear relationship between Y and X_1 . However, the added-variable plots show that each of these relationships are strongly linear when the linear effect of the other variable is removed from the regression. Conversely, as evident from Figure 11, lack of a linear relationship in the added-variable plot does not imply that the corresponding (Y, X) variables are independent; rather it could be that in the presence of the other variables, the X variable plotted does not provide any additional explanatory power to the fitted regression model.

————— Figure 11 appears here —————

Figure 12 displays an example of a 3-variable model fitted to data where the variables X_1 and X_2 are generated to be highly correlated with each other. By examining the added-variable plots it becomes evident that X_1 does not contribute additional predictive information to the regression in the presence of X_2 and X_3 in the model. No linear trend is apparent in the first added-variable plot in Figure 12 showing very clearly that fitting a model with both X_1 and X_2 in the model will cause either or both estimated coefficients to possess a large sampling variance. The VIFs for the coefficients b_1 and b_2 are relatively large indicating the usefulness of the VIF as a direct measure of the effect of multicollinearity in model estimation. The VIFs are displayed when the **Regression Stat** button is selected.

————— Figure 12 appears here —————

2.6 Transforming Nonnormality and Nonconstant Variance

Two assumptions in regression are that the mean response is a linear function of unknown regression coefficients and that the errors are additive and are a random sample from a normal distribution with constant standard deviation. Graphical tools that help to check these assumptions using the fitted model are generally known as *residual plots*. To check the normality assumption one would use a normal probability plot of either the raw residuals or studentized residuals. For checking homogeneity of error variances, plots of residuals against the predicted values (\hat{Y} 's) and residuals against each explanatory variable (X_i) are useful. When patterns in these plots indicate devi-

ations from model assumptions, one remedy often advocated is to attempt transforming (or re-expressing) the variables to better satisfy these assumptions. Transforming the response variable using the Box-Cox power family of transformations is often recommended to restore normality. More traditional transformations that are designed to achieve constant error variances also promote normality in some instances.

The `regteach5` module has been constructed to illustrate some of the important concepts related to the use of residual plots and transformations in regression analysis:

- A normal probability plot of the residuals can be used to identify features of the shape of their distribution such as skewness and whether it is long-tailed or short-tailed.
- Certain patterns in the plot of residuals against the predicted values (i.e. the \hat{Y} 's) can be used to identify the form of dependence of the error variance on the mean of Y . This plot will be called the *residual plot* below.
- Variance stabilizing transformations achieve more nearly constant error variances. These may also help restore normality to the data in some instances.
- The Box-Cox power transformation can also be used to transform Y so that the transformed data may be adequately described by a normal distribution.

Figure 13 shows the start-up window of the `regteach5` module. In the example displayed, the **Select Data Set** button has been used to select a data set and the **Normal Plot** button employed to obtain the corresponding normal probability plot.

————— Figure 13 appears here —————

The start-up window also contains a **Simulation** button that can be used to generate *data simulation windows*: a window to generate simulated data and a window to display the corresponding normal probability plot (see Figure 14). While in this set-up, the module is said to be in the *simulation mode*. In this mode, random samples may be drawn from distributions with various shapes. By observing normal probability plots of these data sets, students learn to associate types of deviations from a straight-line pattern with the shapes of the underlying distributions.

Selecting an item from the **Distributions** pulldown menu initiates the creation of a normal probability plot of a random sample of 30 data points drawn from one of three distributions: $N(0,10)$, Chi-squared (3 d.f.) and Student's t (3 d.f.). A plot of the density of the selected distribution overlaid with a dot plot of the actual sample data drawn is displayed in the first simulation window. New samples can be drawn by pushing the **New Sample** button repeatedly or by holding it down using the mouse. Figure 14 shows the examples of plots for samples drawn from each of the above 3 distributions.

————— Figure 14 appears here —————

In the *transformation mode*, the `regteach5` module allows the user to perform data transformations on a selected data set. After choosing a data set by pressing `Select Data Set` button, the user can select a variance stabilizing scheme from a pulldown menu by pressing the `Var Stbl Trans` button. By observing the changes made dynamically in the *residual plot* and the *normal probability plot* in response to each transformation attempted, the user will be able to select a transformation that adequately stabilizes the variance and brings the data closer to normality. Also the `Power` and `Shift` slider-bars allow the user to try Box-Cox transformations of the original data by choosing various power and shift values. The residual plot and the normal probability plot can be saved for comparison among satisfactory transformations by pushing on the `Save Plot` button. Figure 15 and Figure 16 show an example of each of the above plots saved for such a comparison.

In Figure 15, the residual plot indicates a dependence of the residual variance on the magnitude of the response, and the probability plot indicates some deviation from normality. The *square root* transformation appears to restore normality and stabilize the variance to some extent, as evinced from the bottom set of frames.

————— Figure 15 appears here —————

In Figure 16, the residual plot shows both curvature and nonconstant variance, although the residuals do not exhibit significant deviation from normality. As shown in the bottom set of frames, a power transformation of $\lambda = 0.3$ appears to stabilize the variance while retaining normality.

————— Figure 16 appears here —————

3 Availability

The instructional modules and Lisp-Stat source code for the software components of our modules are available via anonymous ftp from Iowa State University. To obtain these, use the command `ftp isua.iastate.edu` with “anonymous.stat” as the username and “yourusername@your.email.host” as the password. This should get you into the statistics directory named “anonymous”. The subdirectories “Teach”, “RegTeach”, and “DsnTeach” contain Readme files describing three sets of software. If you are accessing files from a Unix host, ftp the compressed archive “regteach.tar.gz” to obtain a version that can be installed on the Unix platform. Otherwise obtain the file appropriate for your platform (e.g., `regteach.exe` for PC/Windows and `regteach.sea.hqx` for older Macs). These files are binary archives that unbundle when executed. Other Readme/Install files will be found in each package after unbundling.

Pdf formatted files of the current version of this paper and instructional module lessons designed for use with the software modules are available in the subdirectory `/Docs`. The subdirectory `/Lessons` and `/Figures` contain the original latex files and

corresponding figures used in the lesson documents.

`Lisp-Stat` is freely available from `umnstat.stat.umn.edu` or from `statlib`. It is recommended that Mac OS X users adapt the Unix versions of the software; instructions to do this are available in a help file in the shell archive. Our software currently runs under Version 2.1 Release 3.52 of `Lisp-Stat`. We plan to make this software available from other servers on the internet such as `statlib` and the UCLA statistics archive.

4 Concluding Remarks

In this article we have extended the collection of instructional modules described in Marasinghe et al. (1996). As with the modules described in Iversen and Marasinghe (2001), the ones described here are specific to regression analysis and are much more advanced, elaborate, and complex than the ones describing elementary statistical concepts. One tenet we have attempted to follow in developing these modules is to obtain ideas and feedback from those instructors involved in teaching these topics. We hope that the research presented here will lead to exploration, refinement, and dissemination of other such modules for teaching statistics interactively. We welcome comments from potential users of these modules.

We are grateful to the statistics instructors who used the earlier sets of modules in their teaching and sent us comments and words of encouragement. The current set of modules was developed for use in courses where the primary audience is undergraduate and graduate students from disciplines other than statistics. We hope that they will enable these students to obtain an improved understanding of the underlying statistical concepts.

References

- Anderson, J. E., and Dayton, J. D. (1995), "Instructional Regression Modules using XLISP-STAT," *Journal of Statistical Education*, 3, No. 1.
- Becker, R. A., Cleveland, W., and Wilks, A. (1987), "Dynamic Graphics for Data Analysis," *Statistical Science*, 4, 355–395.
- Cleveland, W. S., and McGill, R. (eds.) (1988), *Dynamics Graphics*, New York: Chapman and Hall.
- Cook, R. D., and Weisberg, S. (1989), "Regression Diagnostics with Dynamic Graphics (with discussion)," *Technometrics*, 31, 277–311.
- Cook, R. D., and Weisberg, S. (1994), *An Introduction to Regression Graphics*, New York: Wiley.

Cook, R. D., and Weisberg, S. (1999), *Applied Regression Including Computing Graphics*, New York: Wiley.

Darius, P., Michiels, S., Raeymaekers, B., Ottoy, J-P., and Thas, O. (2002), "Applets for Experimenting with Statistical Concepts," in *Proceedings of the Sixth International Conference on Teaching Statistics*, Cape Town, South Africa.

de Leeuw, J. (1994), "The Lisp-Stat Statistical Environment," *Statistical Computing and Graphics Newsletter*, 5, No. 3, 13–17.

Iversen, P., and Marasinghe, M. G. (2001), "Dynamic Graphical Tools for Teaching Experimental Design and Analysis Concepts," *The American Statistician*, 55, No. 4, 345–351.

Marasinghe, M. G., Meeker, W. Q., Cook, D., and Shin, T. (1996), "Using Graphics and Simulation to Teach Statistical Concepts," *The American Statistician*, 50, No. 4, 342–351.

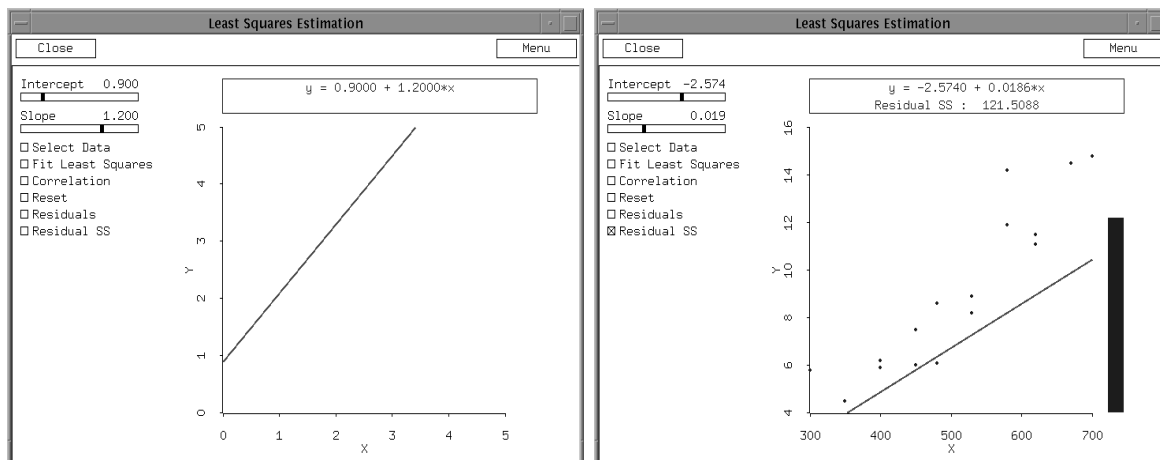
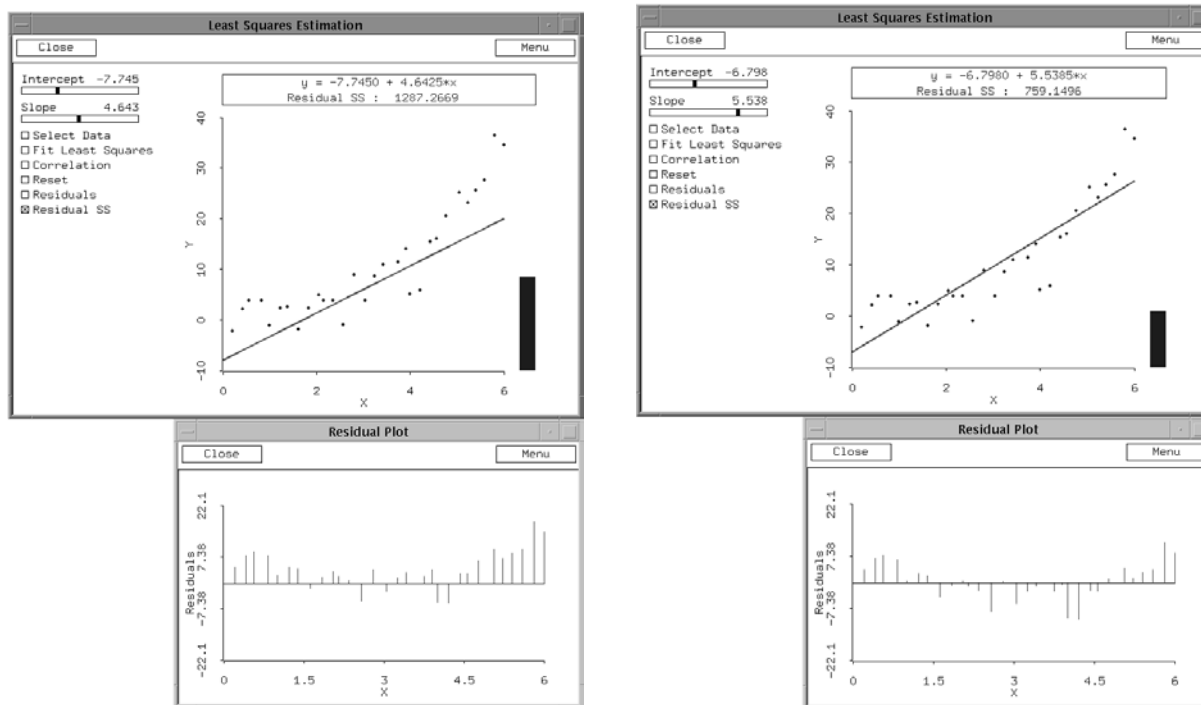
Nurhonen, M., and Puntanen, S. (1992), "Illustrating Regression Concepts," *Teaching Statistics*, 14, No. 1, 20–23.

Saunders, D. J. (1986), "Computer graphics and animations for teaching probability and statistics," *International Journal of Mathematical Education in Science and Technology*, 17, 561–568.

Tierney, L. (1991), *Lisp-Stat*, New York: Wiley.

Trumbo, B. E. (1994), "Some Demonstration Programs for Use in Teaching Elementary Probability: Part 1 and 2," *Journal of Statistical Education*, 2, No. 2.

Velleman, P. (2004), *ActivStats 2003-2004 Release*, Addison-Wesley.

Figure 1: Two Frames of the *regteach1* Module WindowFigure 2: *regteach1* Two Fitted Lines and Corresponding Residuals

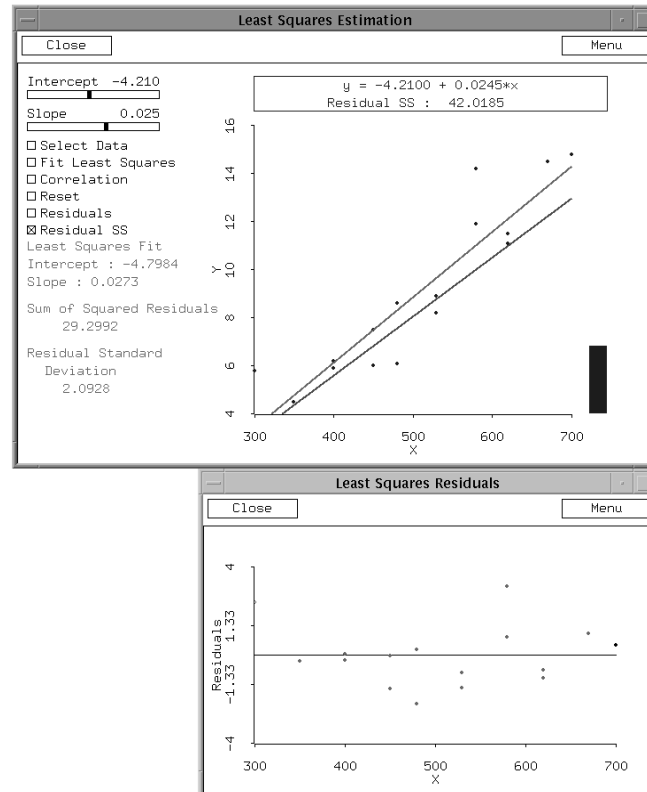


Figure 3: regteach1 Module With Fitted Regression Line

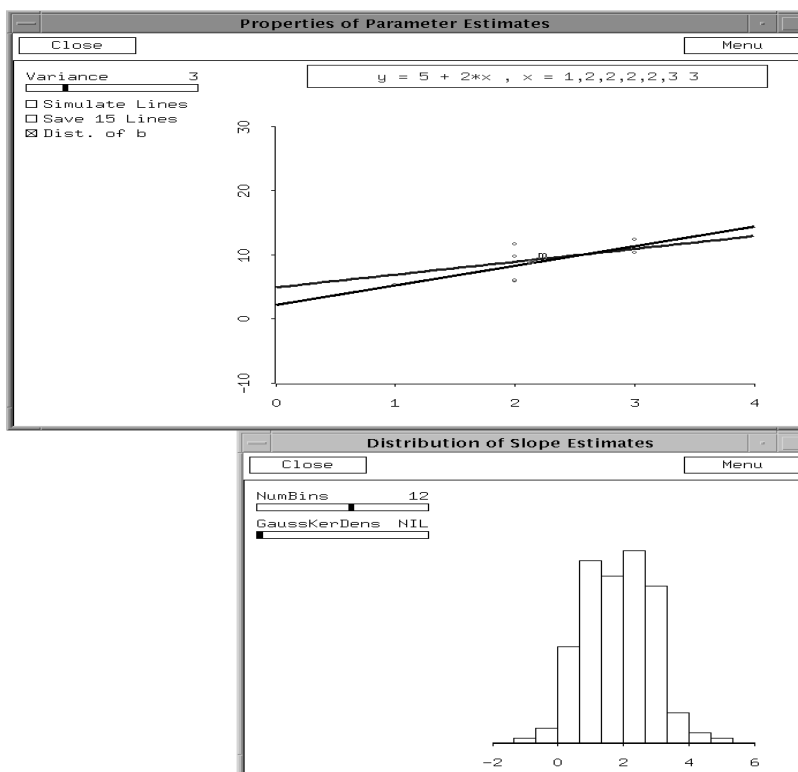


Figure 4: regteach2 A Fitted Regression Line with the Dynamic Histogram of Slopes

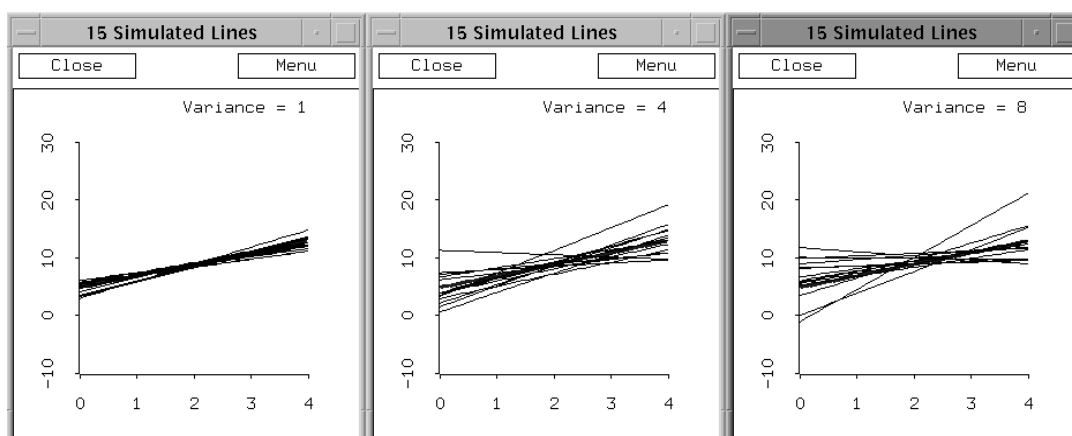


Figure 5: regteach2 Simulations of Straightline Regressions for 3 Error Variances

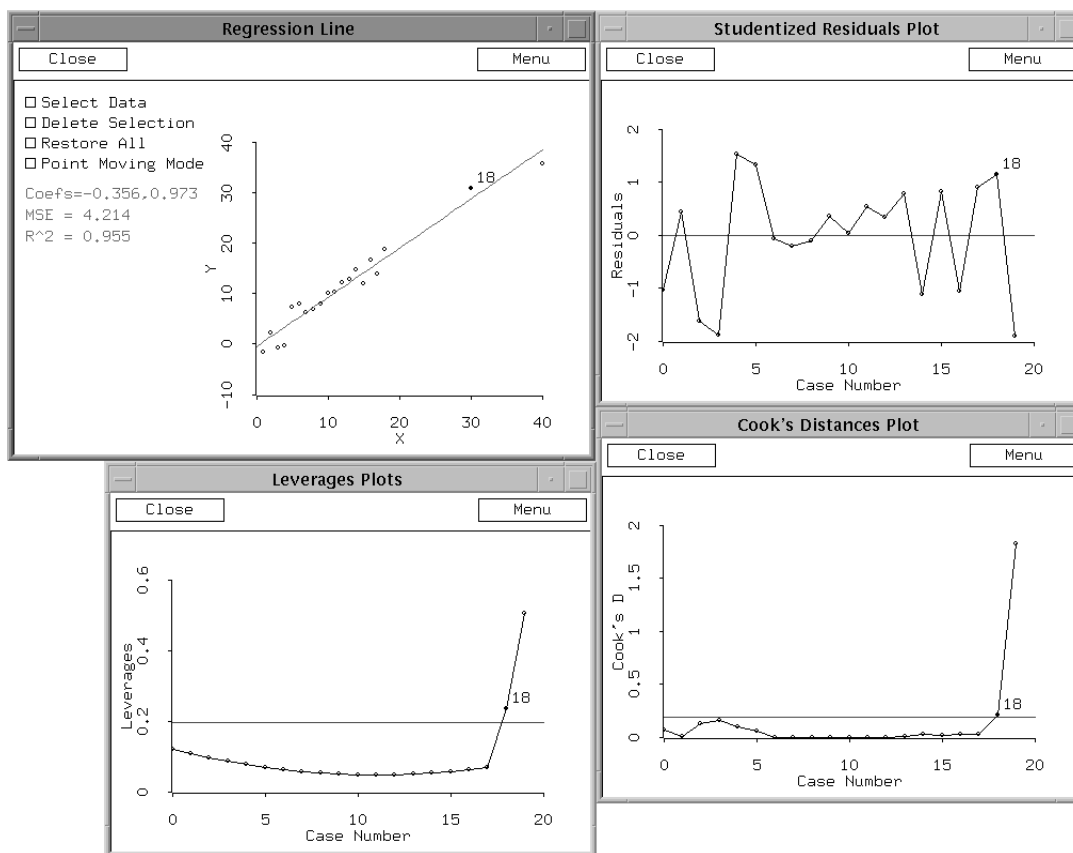


Figure 6: regteach3 Module Windows with Highlighted Observation

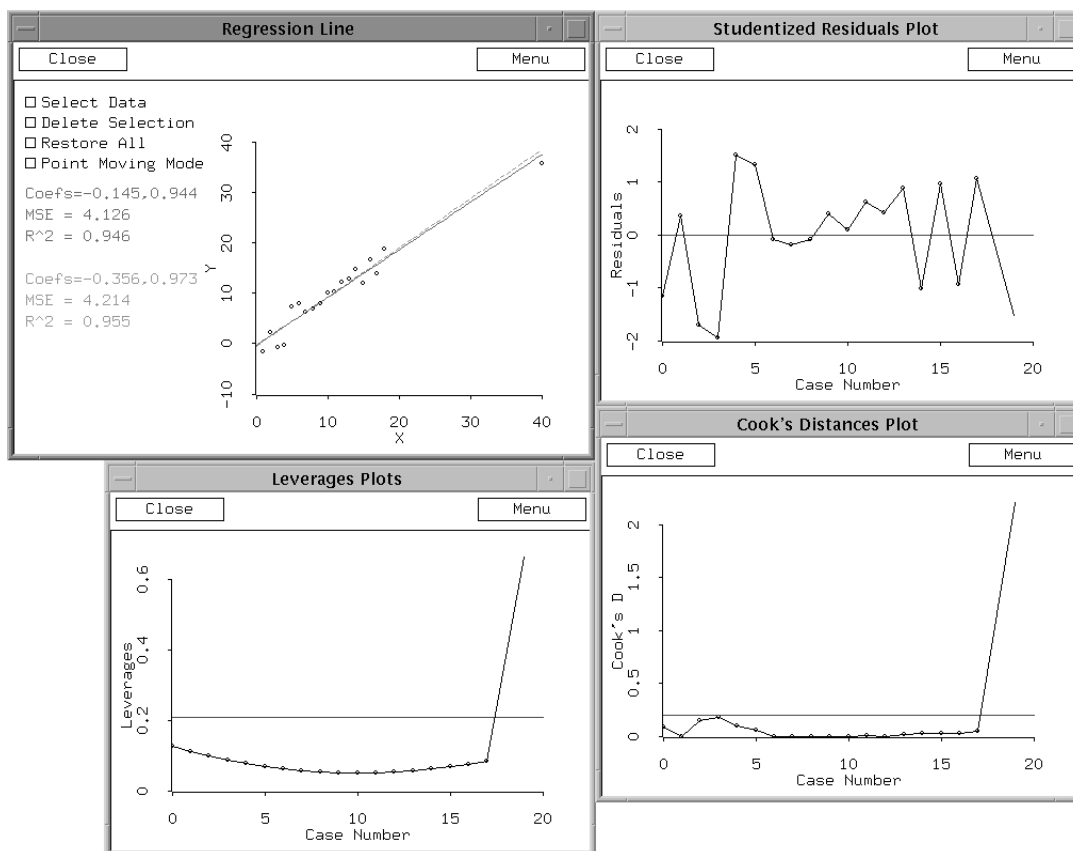


Figure 7: regteach3 Module Windows with Observation 18 Deleted and Regression Refitted

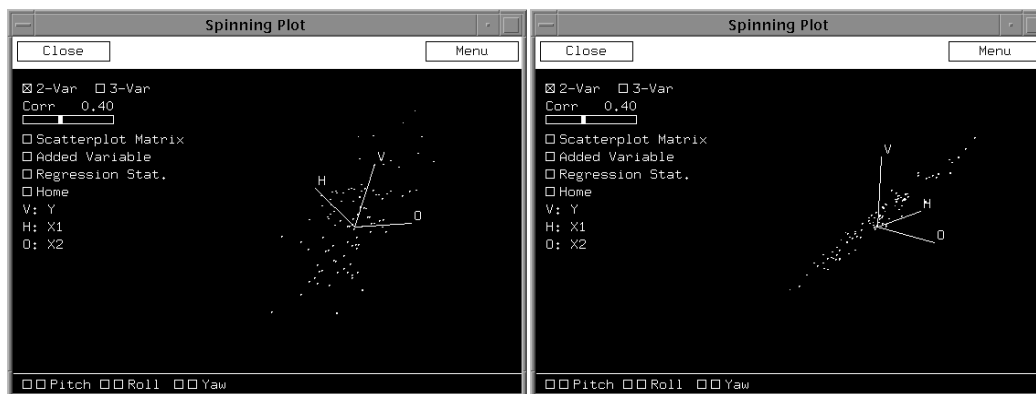


Figure 8: Two Frames of the regteach4 Module Window

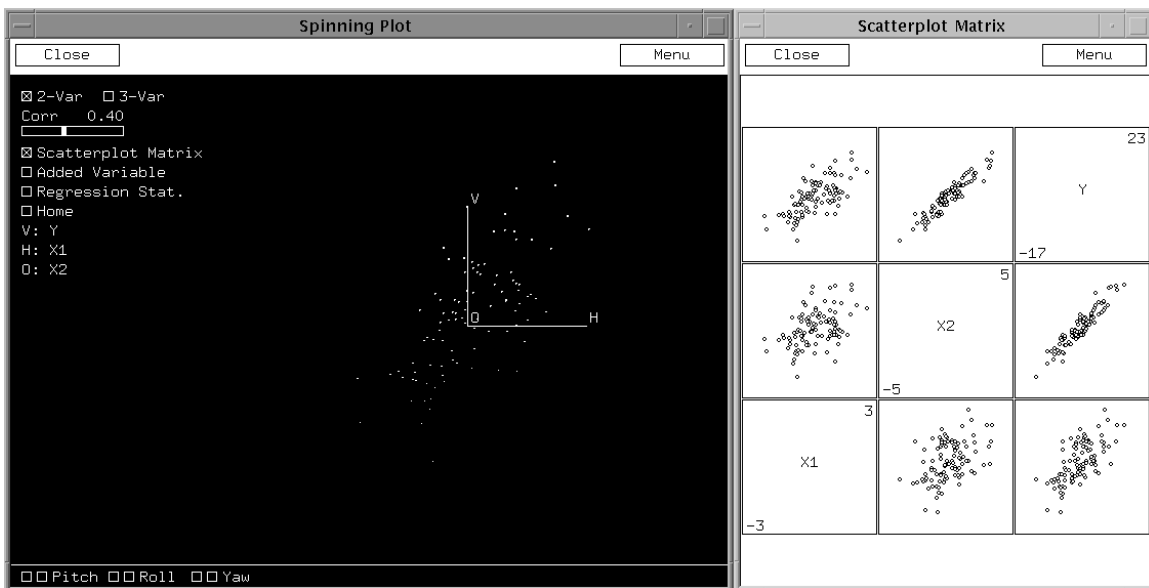


Figure 9: Two Frames of the regteach4: Spin-plot with Scatterplot Matrix

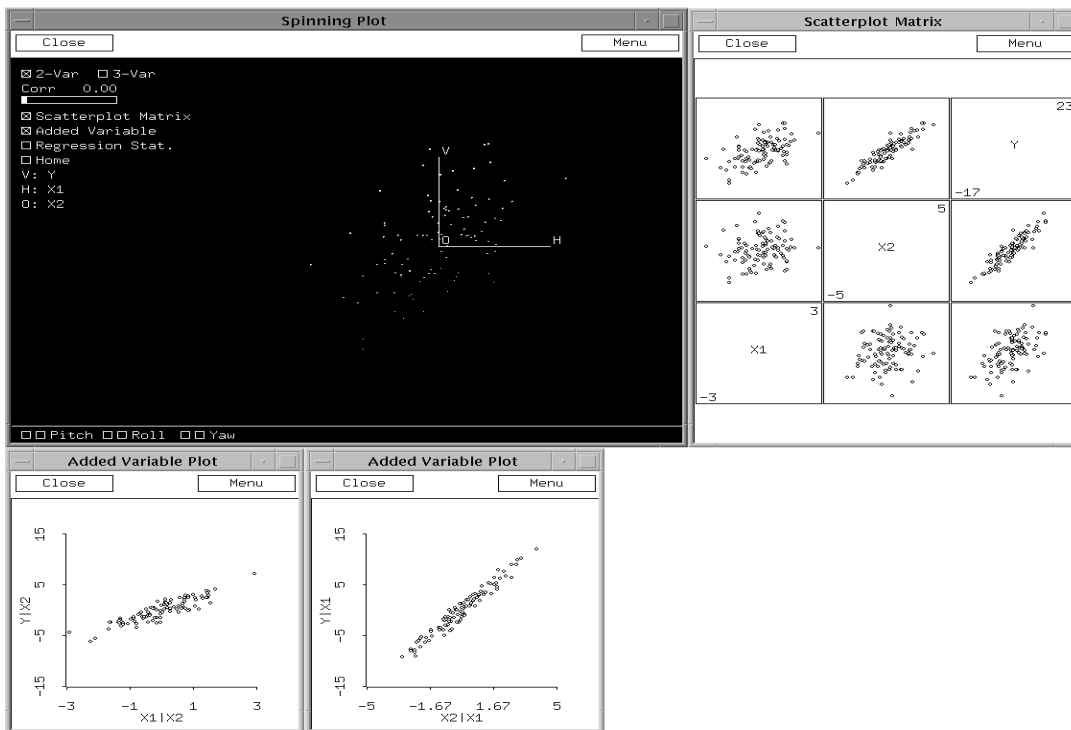


Figure 10: regteach4 Module Windows with Bivariate Plots of Data with Low-Correlated Predictors ($\rho = .04$)

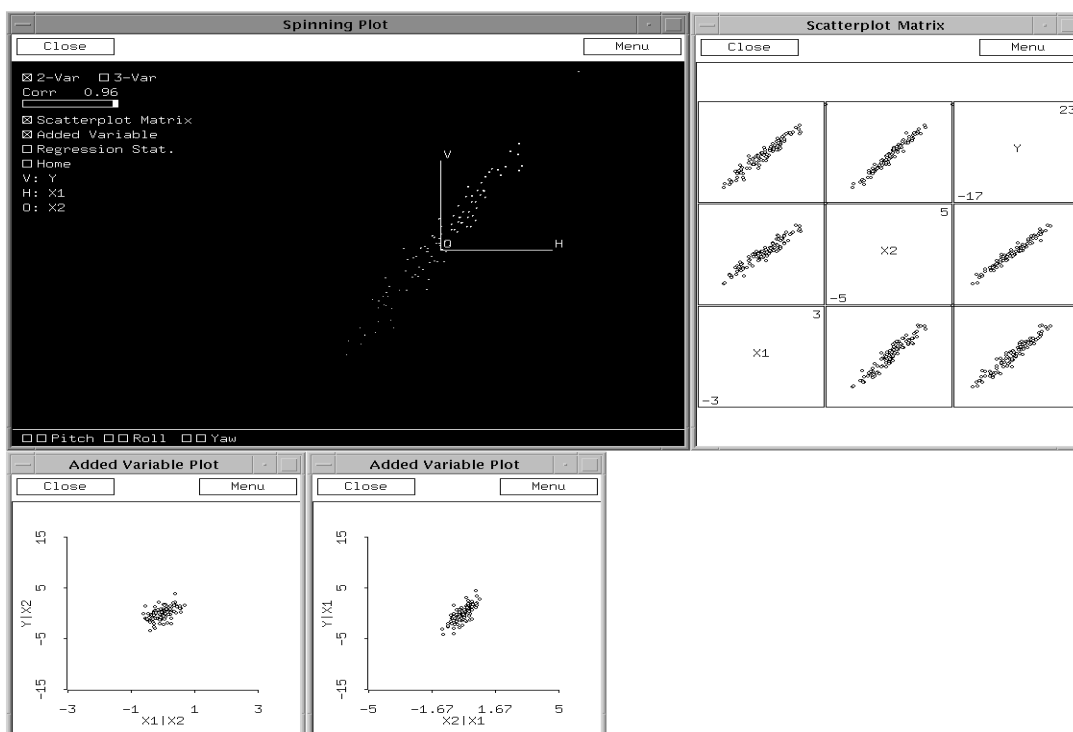


Figure 11: regteach4 Module Windows with Bivariate Plots of Data with High-Correlated Predictors ($\rho = .96$)

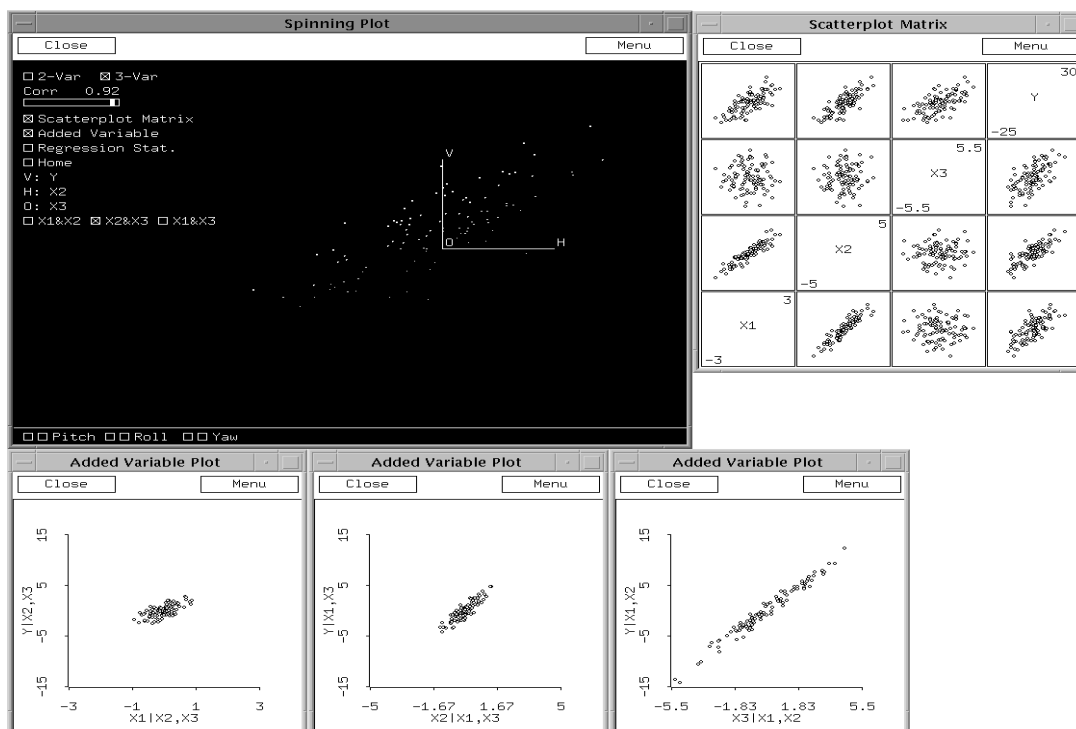


Figure 12: regteach4 Module Windows showing Effects of Multicollinearity

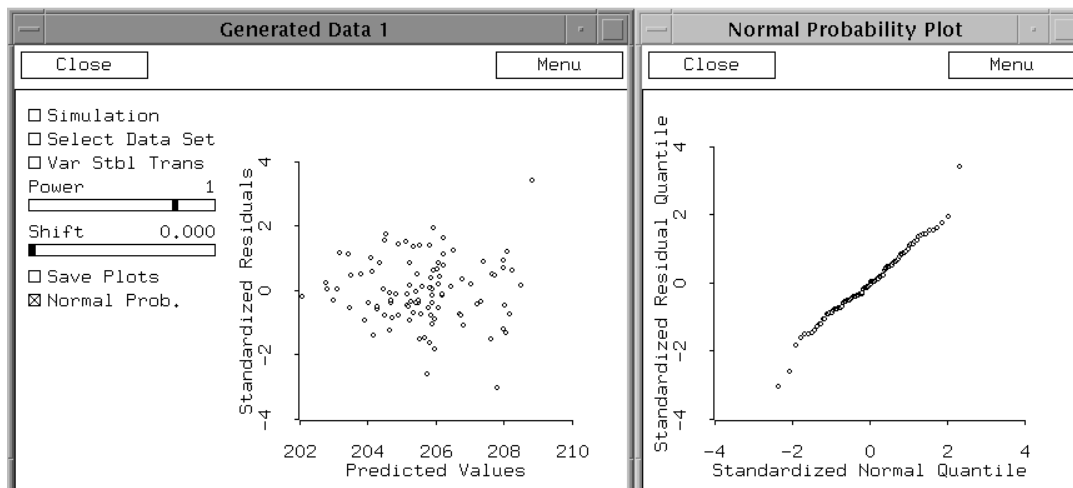


Figure 13: regteach5 Module Windows with Nonconstant Variance Data

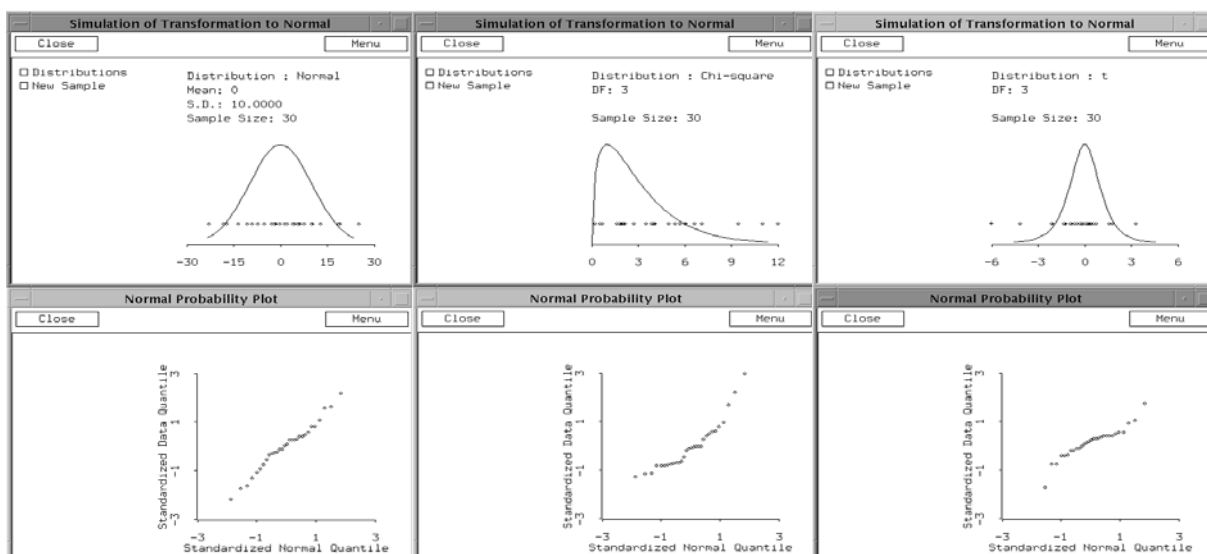


Figure 14: regteach5 Simulation Windows with Normal Probability Plots

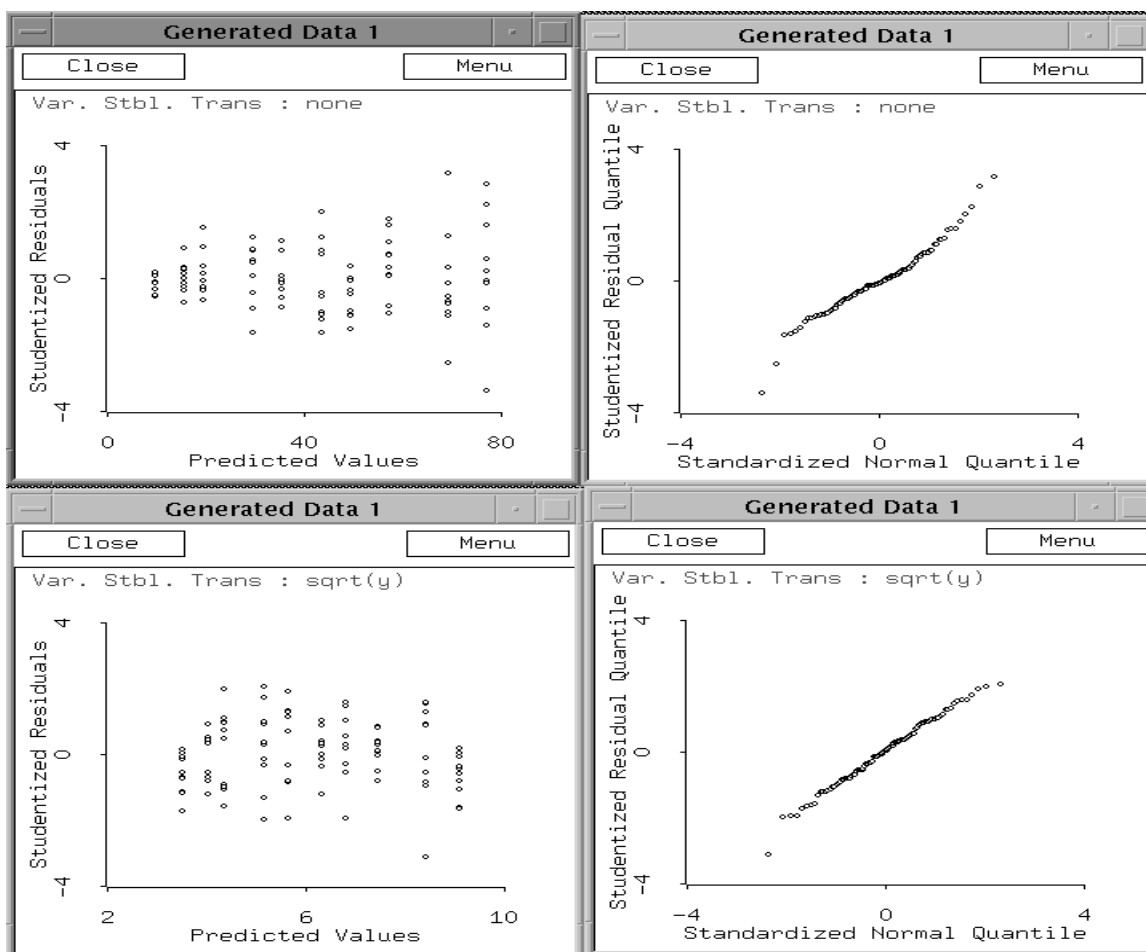


Figure 15: Saved Plot Windows: Variance Stabilizing Transformation of the Response

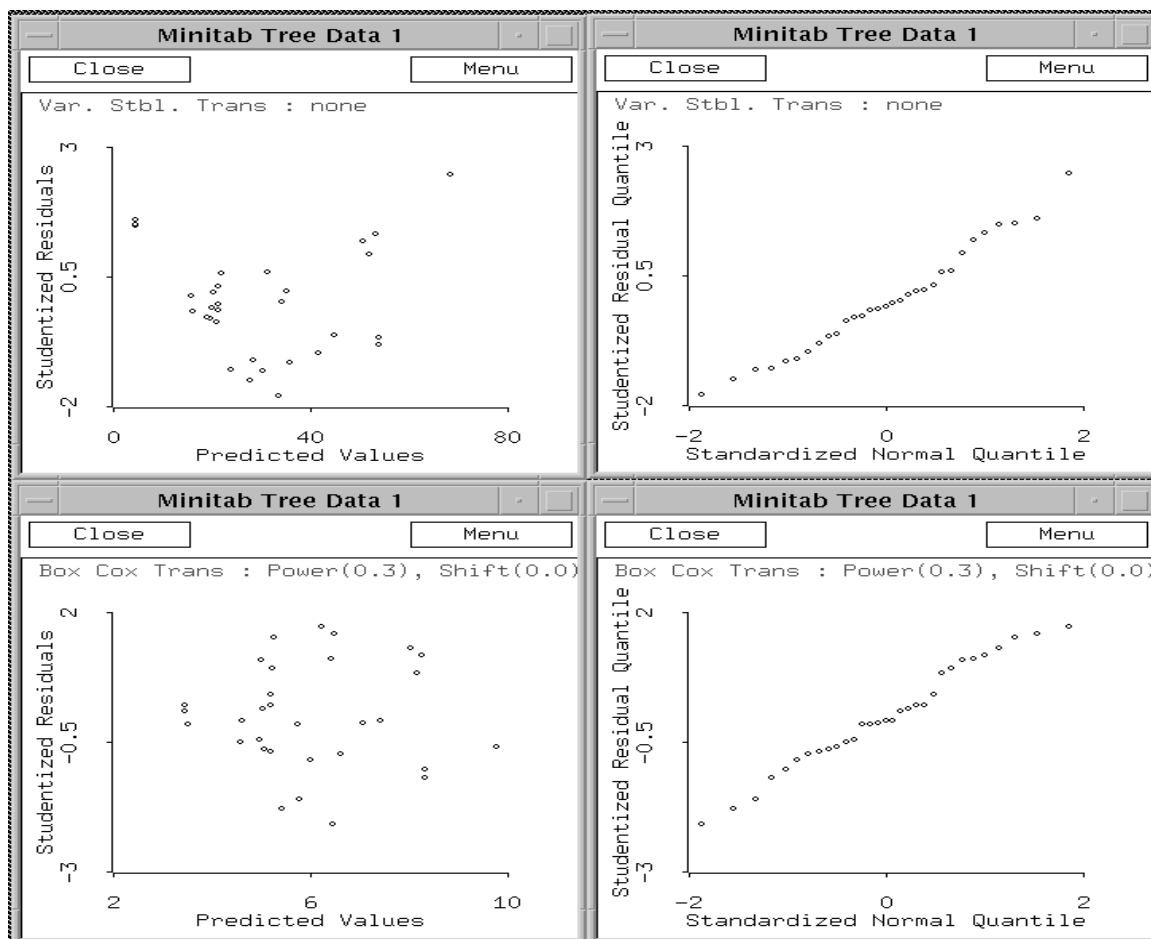


Figure 16: Saved Plot Windows: Box-Cox Power Transformation of the Response