

# ViSta: A Visual Statistics System<sup>1</sup>

**Forrest W. Young & Carla M. Bann**  
**L.L. Thurstone Psychometrics Laboratory**  
**University of North Carolina at Chapel Hill**

## 1.0 Introduction

ViSta – the Visual Statistics System – is an open, extensible and freely distributed system for teaching introductory and multivariate statistics and for research and development in visual and computational statistics. ViSta features state-of-the-art statistical visualization techniques for visually guiding novice data analysts; for visualizing the overall structure of the data analysis session; for visualizing the results of analyses; and for visually exploring the effects of re-parameterizing models. In this paper we discuss visual statistical analysis using ViSta.

ViSta is designed for an audience of users having a very wide range of data analysis sophistication, ranging from novice to expert. ViSta provides seamlessly integrated data analysis environments specifically tailored to the user's level of expertise. Visual guidance is available for novices (such as students), and visual authoring tools are available for experts (such as teachers) to create guidance for these novices. A structured graphical user interface is available for competent users, and a command line interface is available for sophisticated users. The complete Lisp-Stat (Tierney, 1990) programming environment is available to researchers and graduate students who wish to extend ViSta's capabilities.

The fundamental hypothesis underlying ViSta's design is that data analyses performed in an environment that visually guides and structures the analyses will be more productive, accurate, accessible and satisfying than data analyses performed in an environment without such visual aids. We believe that this should be true for all data analysts, but more so for novices. ViSta's design understands that visualization techniques are not useful for everyone all of the time, regardless of their sophistication. Thus, all visualization techniques are optional, and can be dispensed with or reinstated at any time. In addition, standard nonvisual data analysis methods are available, including printed reports, a command-line interface and support for scripts for automated analysis. This combination means that ViSta provides a visual environment for data analysis without sacrificing the strengths of standard statistical system features that have proven useful over the years.

ViSta provides four state-of-the-art visualization techniques. These techniques, which are overviewed in Young (1994) include 1) GuideMaps that visually guide users with little or no knowledge about data analysis (Young & Lubinsky, 1995); 2) WorkMaps that visualize the structure of an on-going data analysis session and that let the analyst return to earlier steps to pursue new analysis ideas (Young & Smith, 1991); (3) Dynamic statistical visualization techniques designed to accurately and efficiently communicate data structure and modeling results (Young, Faldowski & McFarlane, 1993); and (4) Statistical Re-Vision techniques that let the analyst visually explore the effects of revising model parameter estimates (McFarlane & Young, 1994; Faldowski, 1995; and Lee, 1994). We present examples of the first three of these techniques in this paper.

ViSta is based on the Lisp-Stat system (Tierney, 1990). The complete Lisp-Stat data analysis and programming environment is available to those who wish to use it. However, ViSta hides this programming environment from those who are less expert, so that they can perform analyses entirely in a point-and-click manner. ViSta runs under Microsoft Windows, on Macintosh microcomputers, and under Unix. For more information about ViSta, see Young (1994). The

---

1. This paper was published in: Stine, R.A. & Fox, J. (Eds.) *Statistical Computing Environments for Social Research.*, Sage Publications, Inc. (1997), pp. 207-235.

complete ViSta software system, documentation and relevant papers may be freely downloaded from the World-Wide-Web site at <http://forrest.psych.unc.edu>, or by anonymous ftp from [www.psych.unc.edu](http://www.psych.unc.edu).

In this paper we show how ViSta can be used to perform the analyses of the Duncan (1961) data that Tierney (1995) performed using LispStat. In addition, we show how Bann (1996) used LispStat to enhance ViSta so that it can perform certain additional analyses which cannot be performed with Lisp-Stat.

## 2.0 Basic Use of ViSta

ViSta provides five different data analysis environments (guidemaps, workmaps, menus, command lines and scripts). Each of these environments provide identical data analysis capabilities, but the human-computer interaction style of each environment is tailored for users with a specific type of data analysis competence. In addition, ViSta provides a sixth computing environment (authoring tools) for expert statisticians to create the computing environment that is designed for novices, and a seventh computing environment (Lisp, itself) for programmers who wish to extend ViSta's capabilities. We review these seven environments here.

**GuideMaps.** ViSta has *guidemaps*, visual diagrams that guide novice data analysts through the steps of a data analysis. Figure 1 is a guidemap for exploring data. The steps are indicated by buttons — highlighted (dark) buttons are suggested steps. The sequence of steps is indicated by arrows pointing from one button to the next. The structure of the guidemap doesn't change as the analysis proceeds, although the highlighting does.

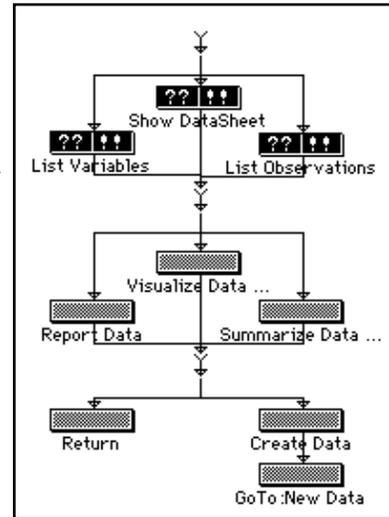


Figure 1: A GuideMap

The novice user makes choices by pointing and clicking with a mouse on the !! side of a highlighted button. Help about the step may be obtained by clicking on the ?? side. After a suggested step is taken the selection of active buttons changes to show the user which actions can be taken next. In the guidemap shown in Figure 1, the button highlighting indicates that the analyst has the choice of three actions: show the datasheet, list variable names or list observation labels. When the user chooses any one of these three actions, the action takes place and the chosen button turns gray, since it is no longer a recommended action. The other two buttons remain highlighted. These two buttons have to be used before the next three buttons are activated. In this way the user is guided to use all three active buttons before doing anything else. They can be used in any order. Once they are all used, the next group of three buttons is activated, and the analyst must use them (in any order) before going on. For more detail see Young & Lubinsky, 1995.

**Workmaps.** ViSta has *WorkMaps*, visual diagrams of the steps taken in a data analysis session, which can serve as a memory aid for novices as well as those who are more competent. An example is shown in Figure 2. Unlike a guidemap, whose structure doesn't change, a workmap automatically expands as the steps of the analysis take place. It serves as a history of the

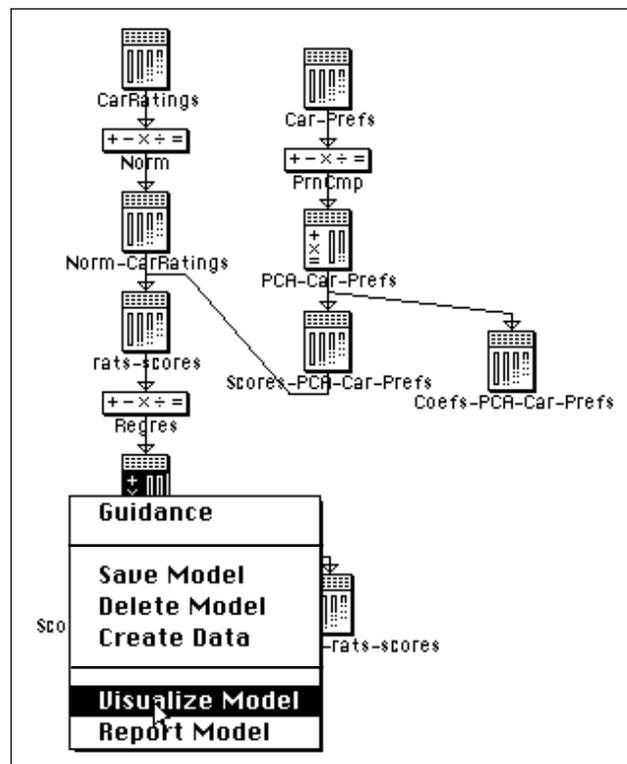


Figure 2: A WorkMap with a Popup Model Menu

analysis, and can be used to return to previous steps. On the workmap, the statistical objects that form the grist of the analysis are represented by icons. The flow of the analysis is indicated by the arrows connecting the icons.

There are three kinds of icons on the workmap: First, *data icons*, the icons in the figure which are taller than they are wide with tall and very narrow boxes inside. This icon represents a dataset with variables inside. Second, *analysis icons*, the icons which are short and wide with mathematical symbols inside them. This icon represents an analysis (such as multiple regression) as a mathematical operation. Finally, *model icons*, the icons which are shaped like a data icon, but with both vertical boxes and mathematical symbols inside. This icon shows data *and* mathematical operations since a model is a mathematical abstraction of data.

The data and model icons in the workmap can be opened to visualize or report data or results. A double-click on a data icon opens it to show its datasheet. A double-click on a model icon opens it to show its report. An option-click on a data or model icon produces the icon's popup menu. In the figure the model icon's popup menu is shown with the user about to open the model's visualization. The analyst can use popup menus to carry out an entire data analysis.

The workmap shown in Figure 2 shows the steps that have already been taken in an on-going analysis. We see that the analyst began with the "CarRatings" data. These data were normalized, creating new data named "Norm-CarRatings". The analyst then loaded "Car-Prefs" data, creating a third data icon. These data were analyzed by principal component analysis, producing an analysis procedure icon named "PrnCmp", and a model icon named "PCA-Car-Prefs". The analyst then requested that the model create two datasets, one for the scores and the other for the coefficients. After the normalized ratings were merged with the scores, a multiple regression analysis was performed. The popup menu hides the remainder of the workmap (a partially hidden regression model icon and icons for two datasets output from the regression). For more details see Young & Smith (1991) and Young (1994).

**Menus.** ViSta also has an ordinary menubar-type menu system, which is designed to be used by those familiar with statistics and statistical systems. The menu shown in Figure 3 is the **Data** menu; it contains menu items oriented towards exploring data. Menubar menus can be used whether or not the workmap is open, since they are pulled down from the menubar rather than being popped up from the workmap. The two menu systems have identical menu items. More importantly, the menu items are identical to buttons on the guidemap, as can be seen by comparing Figure 3 with Figure 1. Note that guidemaps are a *structured* menu system (and menus are an *unstructured* guidemap system).

**Command Lines.** ViSta includes a command line interface for sophisticated data analysts. The commands are entered through the keyboard in standard Lisp syntax. The commands that the user has access to are all of those in Lisp-Stat, plus the additional commands provided by ViDal – ViSta's data analysis language. These commands cause the analysis to take place and they create the workmap. For example, the ViDAL statements in Figure 4 are equivalent to the workmap shown in Figure 2, plus they include code for browsing the data and for creating a model report and visualization.

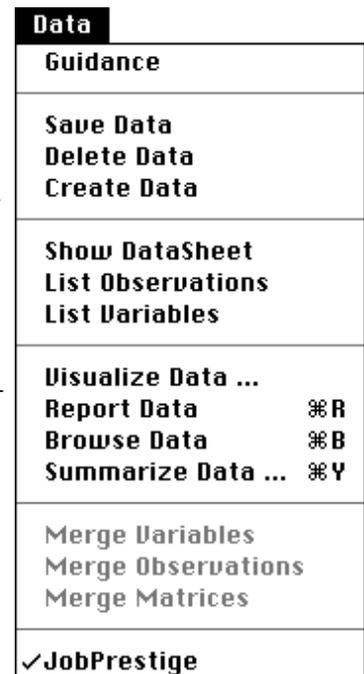


Figure 3: The Data Menu

```
(load-data "carrats")
(normalize-data)
(browse-data)
(visualize-data)
(load-data "carprf14")
(principal-components)
(create-data)
(setcd scores-pca-car-prefs)
(setcd norm-carratings)
(merge-variables "rats-scores")
(multiple-regression
 :data rats-scores
 :responses
   (send carratings :variables)
 :predictors
   ("PC0" "PC1" "PC2" "PC3" "PC4"))
(visualize-model)
(report-model)
(create-data)
```

Figure 4: ViDAL - ViSta's Data Analysis Language

**Scripts.** The environments discussed above are all *highly interactive*. This means that ViSta responds as soon as a button or icon is clicked, a menu item is chosen, or a command is typed. This is desirable in many situations, especially when the analysis is a one-shot or exploratory one. However, in other situations, such as when an analysis will be repeated again in the future on a new wave of data, it is better to be able to collect all commands together into a file and run them all at once without user interaction. This can be done with ViSta: When commands like those shown in Figure 4 are saved in a file, they become a script. The script can be loaded into ViSta to create an analysis that requires no user interaction. Note that the script creates the workmap shown in Figure 2, if desired.

**GuideMap Authoring Tools.** ViSta provides graphical tools so that experts can author the guidemaps just discussed. These tools are described by Young & Lubinsky (1995).

**Lisp.** The entire Lisp programming language is available to the programmer who wishes to extend the capabilities of ViSta. A brief programming example of how ViSta was extended to include the robust regression analysis capabilities reported in this paper is given in section 6.0.

**Seamless Integration.** The data analysis environments outlined above are all seamlessly integrated. Guidemap buttons correspond to menu items, and generate commands that are identical to those typed at the command line. In fact, the titles of the guidemap buttons and the names of the menu items are both identical to the commands that can be typed. These commands, in turn, generate the structured analysis diagram and perform the data analysis. Scripts, as we mentioned, contain the same commands. Finally, the guidemap authoring tools are based on the same underlying commands. Thus, all environments are seamlessly integrated via the underlying data analysis commands.

Because of this seamless integration, it is possible to switch between the several data analysis environments at any time. When the analyst moves into an unfamiliar type of data analysis or loses track of the overall structure of the analysis, s/he can switch from the command line interface to the menu-based interface. If the workmap is hidden, it can be shown at any time, with the entire structured history of the analysis session being presented. Similarly, guidemaps can be switched on or off as desired, without loss of continuity. Also, once a script-based analysis has been completed, the analysis can continue interactively in any of the above ways.

Note that in the remainder of this paper we say that the user performs a step of the analysis by “using an item” that carries out that step. Due to the seamless integration of the various environments, the user could be clicking on a guidemap button, choosing an item from a menubar menu or from a popup menu, or by typing a command. They are all equivalent.

## 3.0 Looking at the Data

In this section we begin to explore the data, but first we must get the data into ViSta.

### 3.1 Bringing data into ViSta

Data can be imported into ViSta by using the **Import Data** item. Imported data must come from a text file that has one line for each row of data, where each line has one number, symbol or string for each variable, the numbers, symbols or strings being separated by spaces. The first column of data will become the observation labels (this can be switched later). The Duncan (1961) data begin:

Accountant	PROF	62	86	82
"Airline Pilot"	PROF	72	76	83
Architect	PROF	75	92	90



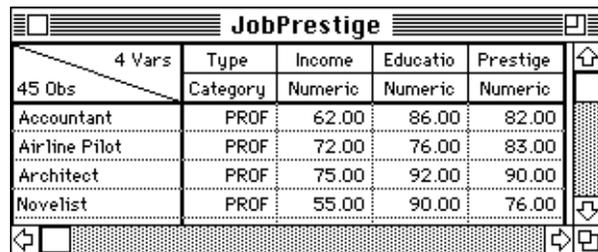
**Figure 5: WorkMap after the Data are Loaded into ViSta**

This defines observation labels and four variables. The first variable is understood to be a category variable and the last three are understood to be numeric. Note that `Accountant` and `PROF` are symbols, whereas `"Airline Pilot"` is a string. Strings must be used to input blanks as part of a value.

Rather than importing the data from a text file, **New Data** can be used to create a datasheet into which the data can be directly typed. **Save Data** can be used to save the data as a ViSta datafile. This file can be used again later with **Open Data**. Once the data are into ViSta, either by importing text, typing it into a datasheet, or by opening a datafile, the data are represented in the workmap by a data icon, as shown in Figure 5. Since this is the first step in the analysis, the workmap only has one icon.

## 4.0 Looking at the data

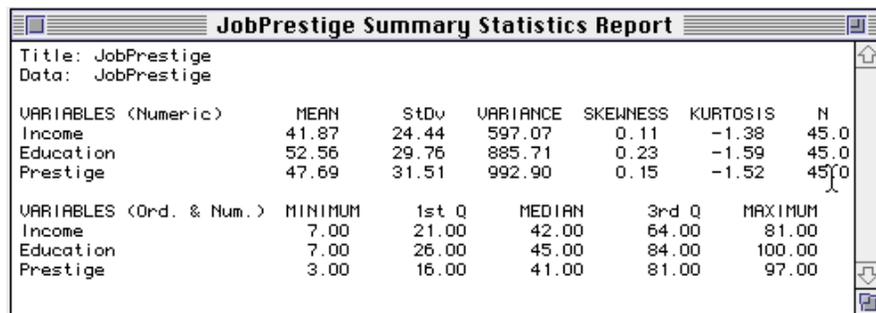
We can now begin our exploration of the data by double-clicking on the data icon in Figure 5, or by using **Show Datasheet**. This produces Figure 6. Note that the datasheet has been used to edit the default variable names (which are simply Var1, Var2, etc.) and the observation labels. Looking at the datasheet reveals no noteworthy features, so we proceed to look at summary statistics and at a visualization of the data.



4 Vars		Type	Income	Educatio	Prestige
45 Obs	Category	Numeric	Numeric	Numeric	
Accountant	PROF	62.00	86.00	82.00	
Airline Pilot	PROF	72.00	76.00	83.00	
Architect	PROF	75.00	92.00	90.00	
Novelist	PROF	55.00	90.00	76.00	

Figure 6: Datasheet showing first 4 Observations

The **Summarize Data** menu item can be used to see the univariate summary statistics shown in Figure 7. An options dialog box lets the user see ranges, correlations and covariances. Note that when the novice user uses the **Summarize Data** guidemap button, no options dialog is presented, under the assumption that such choices confuse novices.



JobPrestige Summary Statistics Report						
Title: JobPrestige						
Data: JobPrestige						
VARIABLES (Numeric)	MEAN	Stdv	VARIANCE	SKENNESS	KURTOSIS	N
Income	41.87	24.44	597.07	0.11	-1.38	45.0
Education	52.56	29.76	885.71	0.23	-1.59	45.0
Prestige	47.69	31.51	992.90	0.15	-1.52	45.0
VARIABLES (Ord. & Num.)	MINIMUM	1st Q	MEDIAN	3rd Q	MAXIMUM	
Income	7.00	21.00	42.00	64.00	81.00	
Education	7.00	26.00	45.00	84.00	100.00	
Prestige	3.00	16.00	41.00	81.00	97.00	

Figure 7 Summary Statistics for the Duncan Data

Finally, the more sophisticated user can type (`summarize-data :correlations t :ranges t`) which will cause only correlations and ranges to be displayed. Other options are available to print other summary statistics. As with the datasheet, the summary statistics reveal nothing noteworthy.

The **Visualize Data** menu item can be used to obtain the five-window visualization of the data shown in Figure 8. This visualization includes a scatterplot-matrix in the upper-left, a spin-plot at the upper-middle, a scatterplot at the lower-left, a histogram at the lower-middle, and a observation-label window on the right. These five windows are linked together in two ways. First, they are linked by variables: It is possible to click on cells of the scatterplot-matrix to determine which variables are shown in the scatterplot, spin-plot and histogram. Second, they are linked by observations: Clicking on one or more points in the spin-plot or scatterplot, dragging across a portion of the histogram, or clicking on observation labels will cause all four of these windows to highlight information showing where the corresponding observations are in each window. In addition, the spin-plot and scatterplot can display labels, as is shown. Of course, it is possible to spin the spin-plot into an interesting position, as has been done here.

This figure demonstrates that simultaneously displayed linked dynamic plots can provide a very powerful way to explore data. Spinning the spin-plot reveals that the data's point-cloud is elliptical, but that there seem to be three outlying jobs (Minister, on one side of the main swarm of points, and the two Railroad jobs on the other). Looking at the raw data, or at the histograms, reveals that ministers have a high prestige, high education job which pays poorly, whereas railroad jobs pay well but are associated with low education (and conductors also have low prestige). As pointed out by Tierney (1995), an OLS regression analysis of prestige on income and education is likely to be determined by the first principal component of variation in income and education, but the fit might be sensitive to the three outliers. We turn to such an analysis now.

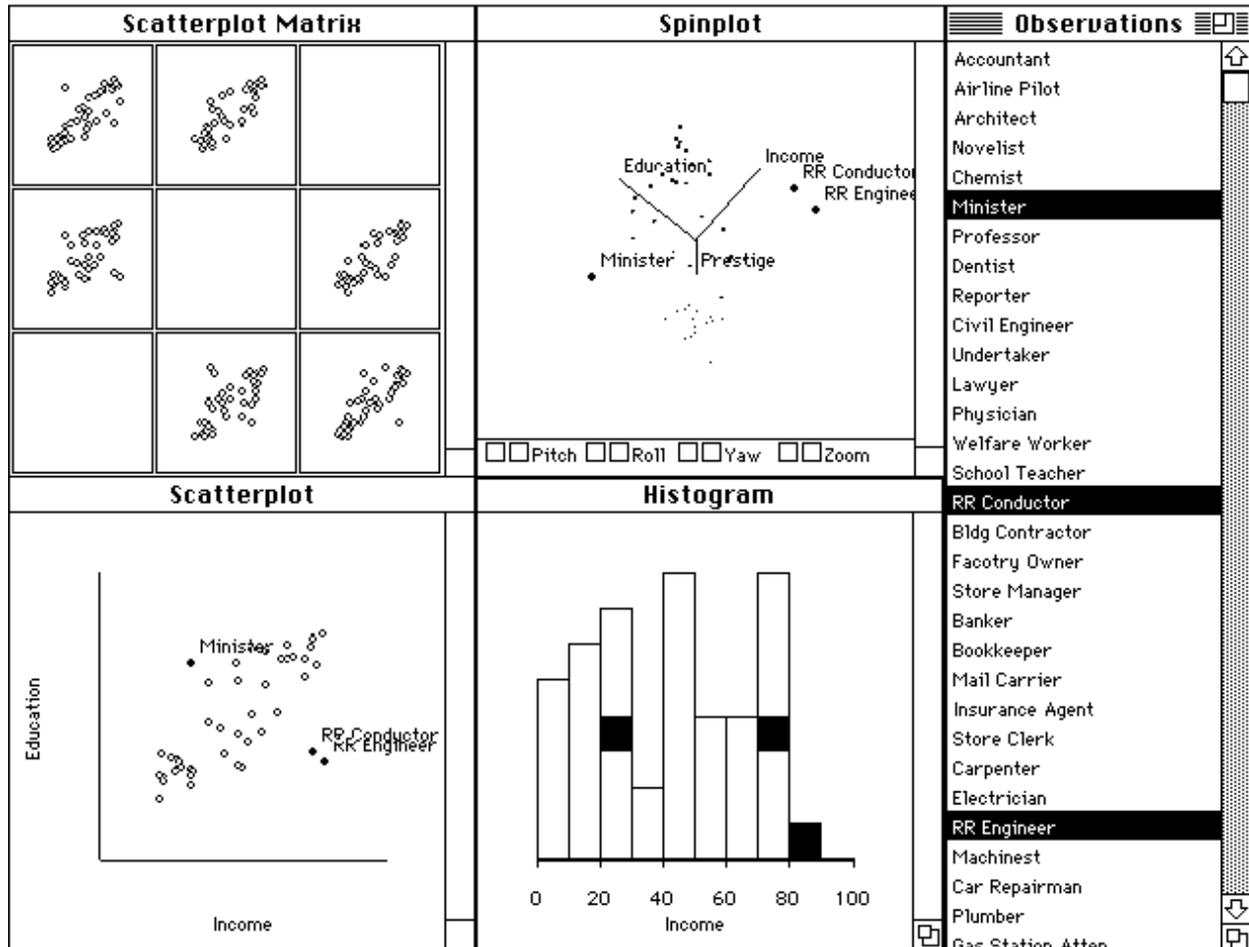


Figure 8: Five Window SpreadPlot Visualization for the Duncan Data

## 5.0 Multiple Regression with ViSta

ViSta can perform four kinds of multiple regression: Univariate or multivariate ordinary least squares (OLS) multiple regression; and univariate robust or monotonic multiple regression. For more detail see Bann (1996a; 1996b). We analyze the Duncan data with each of the three kinds of univariate regression.

### 5.1 Linear Regression

The **Regression Analysis** menu item produces a dialog box for selecting response predictor variables. Once variables have been selected, a regression analysis object and a regression model object are defined and represented on the workmap, which now looks like Figure 9. The regression analysis object is represented by the (small and wide) analysis icon in the middle. The regression model object resulting from the analysis of these data is represented by the lower (tall and narrow) model icon with both vertical boxes and mathematical symbols inside it. Instead of using the **Regression Analysis** menu item, the analysis can also be done by typing:

```
(regression-analysis
 :response "prestige"
 :predictors `("Income" "Education"))
```

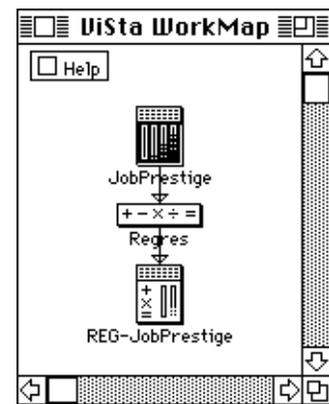


Figure 9: WorkMap after Regression Analysis

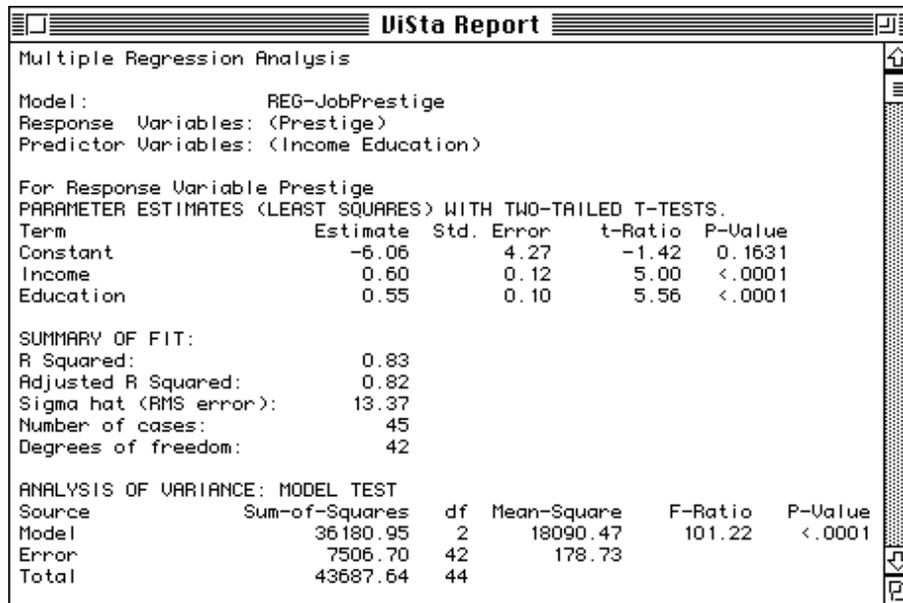


Figure 10: Report of the Regression Analysis

Double-clicking on the model icon (or using **Report Model**) produces the report of the analysis shown in Figure 10. We see that this is like the report provided by Lisp-Stat (Tierney, 1995) but that it is enhanced to include t-tests for each predictor’s contribution to the regression, as well as an analysis of variance of the overall model fit (and, optionally, response values, fit values, residuals, leverages and Cook’s distances). We see that both predictor variables, and the model as a whole, fit the response very significantly.

While the fit is very significant, it may be due to the outliers we saw in the data visualization. Thus, we use **Visualize Model** menu item to see if this is the case, producing the visualization shown in Figure 11. This spreadplot has

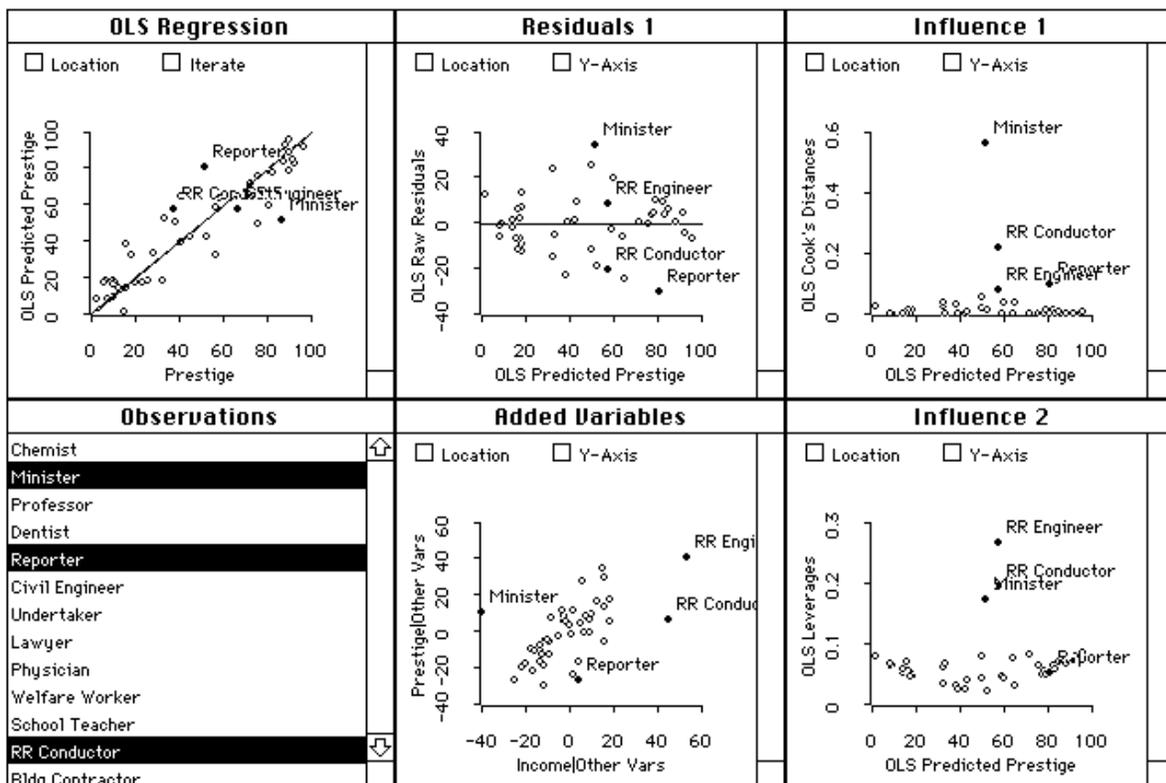


Figure 11: Linear Regression Spreadplot

eight windows, six of which can be see at any one time. Of these eight windows, six are specialized scatterplots and two are lists of variable names and observation labels. The **OLS Regression** scatterplot plots the observed response values against fit values, along with the superimposed regression line. It shows the overall regression, as well as residuals from regression (shown indirectly as deviations perpendicular to the line). The residuals are directly shown in the **Residuals** plot, which plots fit values against residuals. Ideally, the regression and residuals point-clouds should be narrow (showing strong regression), linear and without outliers.

The two **Influence** plots are designed to reveal outliers. They will appear as points that are separated from the main point-cloud. These two plots reveal four points that may be outliers. They have been selected and labeled. Since the plots are linked, the selected points are shown in all windows, with labels. We see that these four points include the three points that looked like outliers in the raw data (Figure 8) plus the “Reporter” job. Looking back at the raw data we see that reporters have a rather high education level, but only average income and prestige. Spinning the data’s spinplot also reveals that the reporter job is on the edge of the main swarm of points.

The **Added Variables** plot shows the relationship between the response and a single predictor, controlling for the effects of all of the other predictors. If the plot is linear, then there is a linear relationship between the response and the plotted predictor, an assumption underlying the OLS regression analysis. The plot is also useful for determining the contribution of a particular predictor variable. If the plot is linear the variable contributes to the relationship, but if the plot shows no pattern the variable does not contribute.

### 5.2 Robust Regression

We now turn to robust multiple regression. First, we added the code presented by Tierney (1995) to ViSta’s regression module, as discussed below in section 6.0. Then we performed the analysis by clicking on the **Iterate** button located in the **OLS Regression** plot. Clicking there presents a choice of two iterative regression methods, one for robust regression, the other for monotone regression. Choosing the robust option, and specifying 10 iterations, produces the visualization shown in Figure 12. This visualization has the same plots as for the linear regression (the

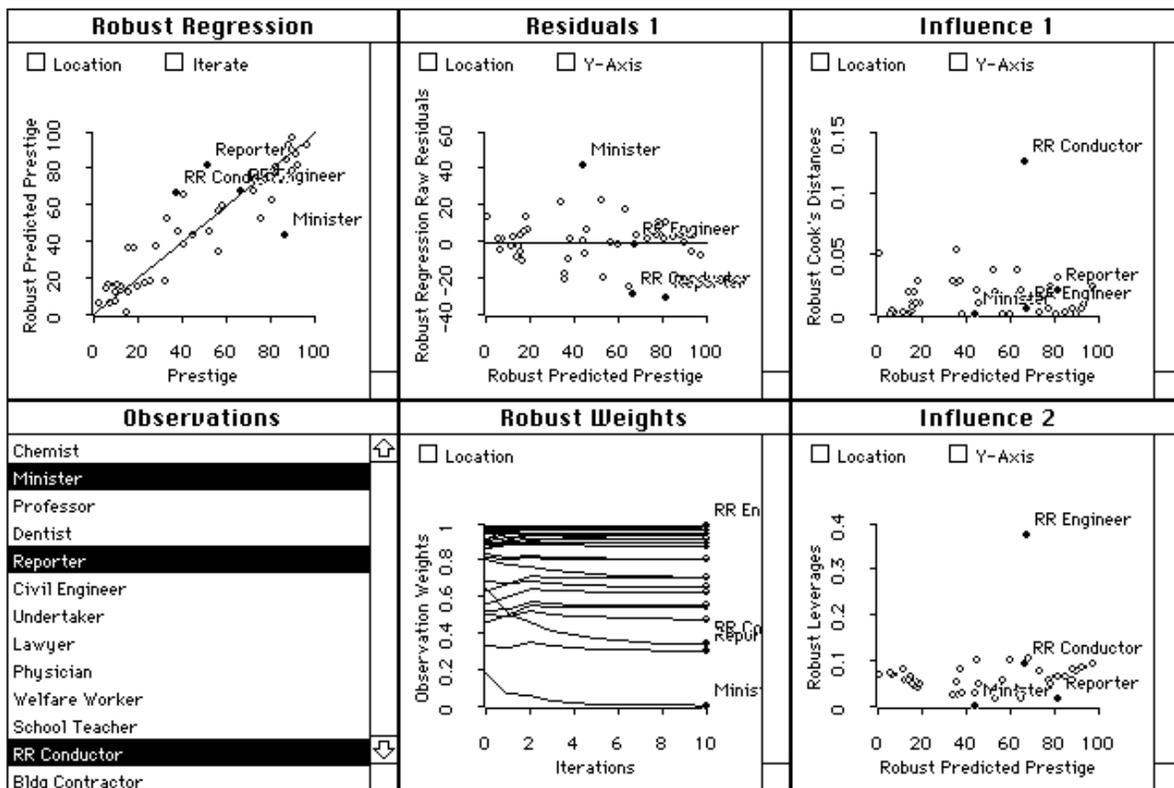


Figure 12: Robust Regression SpreadPlot

regression plot is now named **Robust Regression**), except that a **Robust Weights** plot replaces the **Added Variables** plot. This weights plot shows the iterative history of the weights assigned to each observation. First, we see that the estimated weight values have stabilized. Second, we see that three of the outliers we have identified (Minister, Reporter and RR Conductor) have been given low weight. In addition, the robust Cook's Distances in the **Influence 1** plot and the robust leverages in the **Influence 2** plot show that RR-Conductor has the only extreme Cook's Distance, and RR-Engineer the only extreme leverage. Noting, however, that the Cook's distance scale in the **Influence 1** plot is much smaller than it was before the robust regression iterations, we conclude that the only outlier remaining after the robust iterations is RR-Engineer, which still has high leverage. It also still has a high weight in the weights plot, so it is having an effect (probably detrimental) on the analysis. There is slight, but unimportant, change in the residuals and regression plots.

### 5.3 Linear Regression with Outliers Removed

On the basis of the robust regression, we removed the four apparent outliers from the dataset. This was done by displaying the observation labels window and removing the names of the outliers. We then used the **Create Data** menu item to create a new dataset with these outliers removed. The data icon "JobSubset" is added to the workmap, as is shown in Figure 13.

We analyzed the subsetted data by using the **Regression Analysis** menu item. The report of this analysis (which we do not present) shows that the squared correlation has *increased* to .90, and that all fit tests remain significant. This means that the fit in the original analysis was not due to outliers. The visualization of the OLS analysis of this subset of the Duncan data is shown in Figure 14. First, we note that there are no unusual residuals or leverage values. Second, we see that "Tram Motorman" has an unusual Cook's distance, but in comparison to the linear regression analysis of the

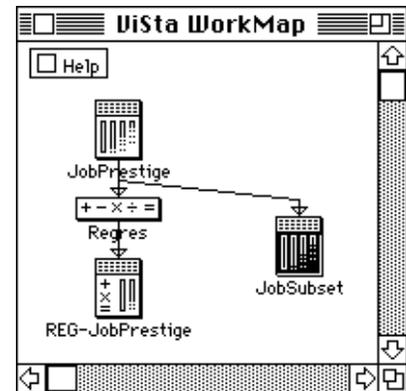


Figure 13: WorkMap after subsetting the Data

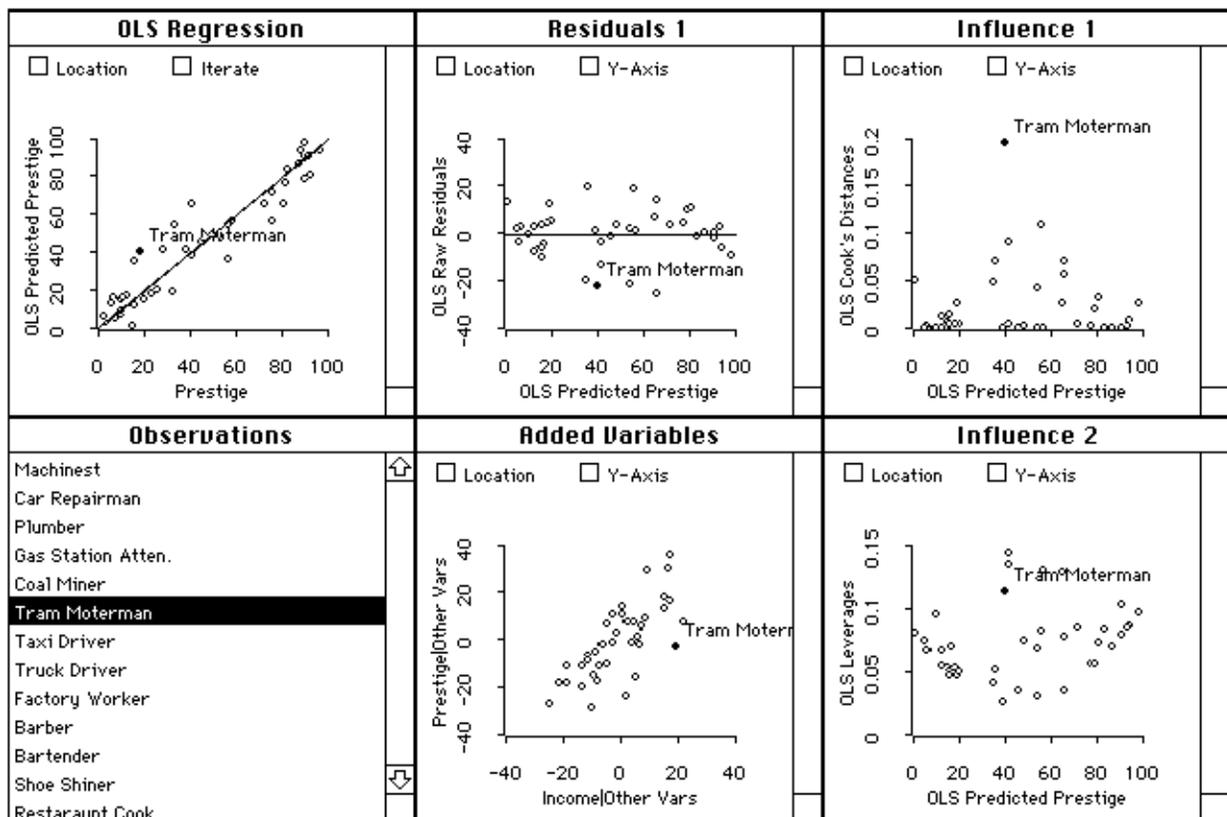


Figure 14: OLS Regression SpreadPlot for Subsetted Data

entire data (Figure 11) the Cook's distance value is relatively small (note the change in scale of the y-axis). Thus, we conclude that the subset of data probably does not contain any outlying observations.

The regression analysis, which has added the analysis and model icons shown in Figure 15, could have been done by typing:

```
(remove-selection (select-observations
  `("Minister" "Reporter"
    "RR Engineer" "RR Conductor")))
(create-data "JobSubset")
(regression-analysis
  :response "Prestige"
  :predictors `("Education" "Income"))
```

### 5.4 Monotonic Regression

Having satisfied ourselves that the subsetted data no longer contain outliers, we turn our investigation to the linearity of the relationship between the response and the predictors. This is done with ViSta-Regres via monotonic regression, a technique based on the MORALS (multiple optimal regression using alternating least squares) algorithm proposed by Young, de Leeuw & Takane (1976).

The MORALS algorithm is similar to the ACE (alternating conditional expectations) algorithm developed more recently by Brieman and Friedman (1985), the difference being that MORALS employs a least squares monotone transformation, while ACE uses a smooth monotone transformation. As implemented in ViSta, the response variable is monotonically transformed to maximize the linearity of its relationship to the fitted model. Specifically, it solves for the monotone transformation of the response variable and the linear combination of the predictor variables which maximize the value of the multiple correlation between the linear combination and the monotonic transformation.

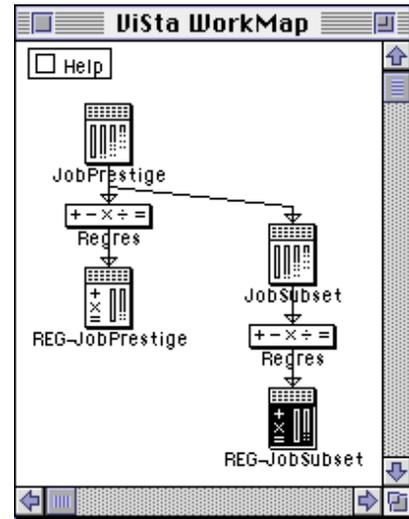


Figure 15: WorkMap after analyzing the subsetted Duncan Data

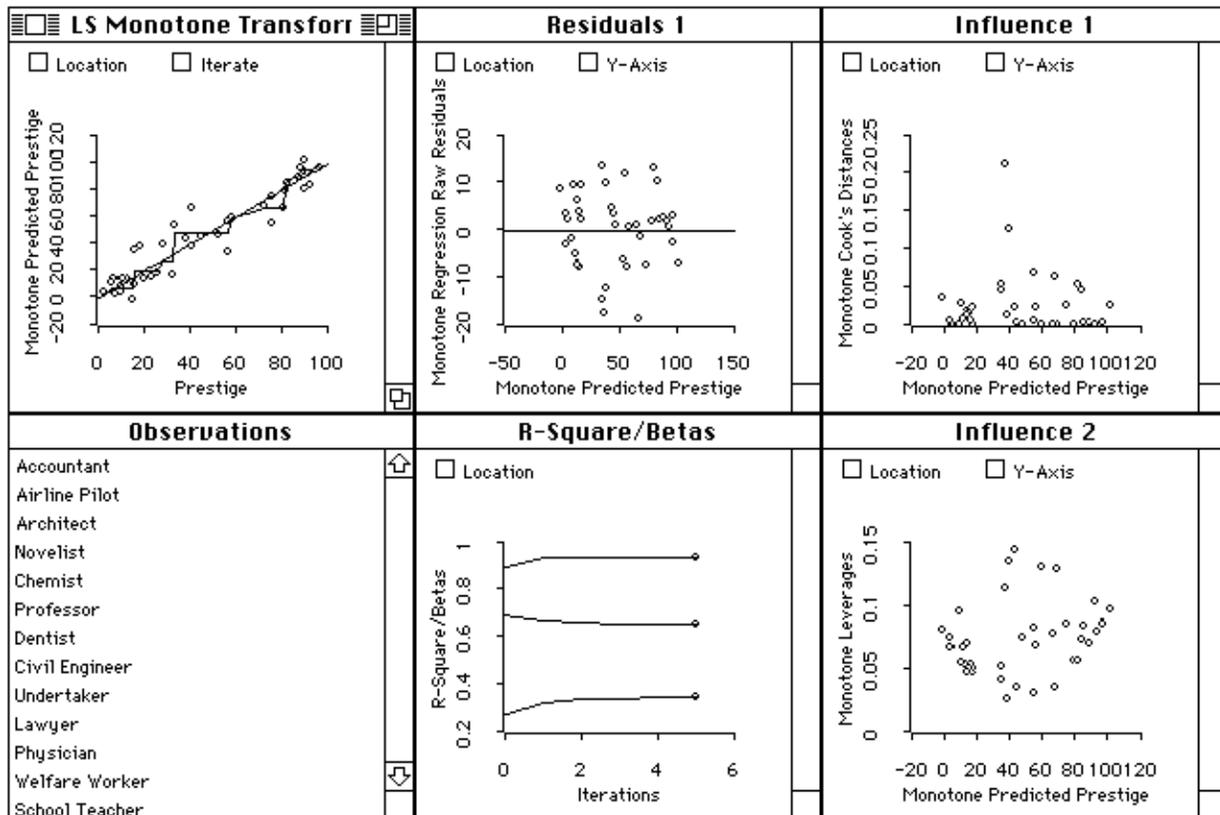


Figure 16: Monotonic Regression SpreadPlot for Subsetted Data

Monotonic regression is done by clicking on the OLS regression plot's **Iterate** button (see Figure 14), and specifying that we wish to perform 10 iterations of the monotonic regression method. When these iterations are completed the report (not shown) tells us that the squared correlation has increased (it cannot decrease) to .94, and that all fit tests remain significant. Thus, there was not much improvement in fit when we relaxed the linearity assumption.

The visualization is shown in Figure 16. We see that the regression plot, which is of primary interest, has a new, non-linear but monotonic line drawn on it. This is the least square monotonic regression line. Comparing this to the least squares linear regression line (the straight line) allows us to judge whether there is any systematic curvilinearity in the optimal monotonic regression. We judge that there does not seem to be systematic curvilinearity. The visualization also contains a new **RSquare/Betas** plot which shows the behavior, over iterations, of the value of the squared multiple correlation coefficient, and of the two "beta" (regression) coefficients. This plot shows that we have converged on stable estimates of the coefficients, and that the estimation process was well-behaved, both of which are desirable. Finally, we note that the two influence plots and the residuals plot have not changed notably from the linear results shown in Figure 15 (although there is some suggest that "Plumber" is a bit unusual). All of these results lead us to conclude, for the subset of data with outliers removed, that the relationship between the linear combination of the predictors and the response is linear, as we had hoped. If this analysis revealed curvilinearity, we would apply an appropriate transformation to the response variable, and repeat the above analyses with the transformed response.

As a final step, we may wish to output and save the results of the above analysis. This is done with the **Create Data** menu item. This item produces a dialog box that lets the user determine what information will be placed in new data objects. Figure 17 shows the workmap that results when the user chooses to create two data objects, one containing fitted values (scores) and the other containing regression coefficients. These data objects can serve as the focus of further analysis within ViSta, or their contents can be saved as datafiles for further processing by other software.

## 6.0 Adding Robust Regression

When we were asked to write this chapter, ViSta did not perform robust regression. It did, however, have a module for univariate regression which would perform both linear and monotonic regression. Thus, we had to modify our code to add robust regression.

The code was added by taking Tierney's (1995) robust regression code, which he had written to demonstrate how robust regression could be added to Lisp-Stat, and modifying it so that it would serve to add robust regression to ViSta. The fundamental conversion step was to change Tierney's robust regression *functions* to become robust regression *methods* for ViSta's regression object (whose prototype is `morals-proto`). Thus, Tierney's `biweight` and `robust-weights` functions become the following methods for our already existing `morals-proto` object:

```
(defmeth morals-proto :biweight (x)
  (^ (- 1 (^ (pmin (abs x) 1) 2)) 2))

(defmeth morals-proto :robust-weights (&optional (c 4.685))
  (let* ((rr (send self :raw-residuals))
        (s (/ (median (abs rr)) .6745)))
    (send self :biweight (/ rr (* c s)))))
```

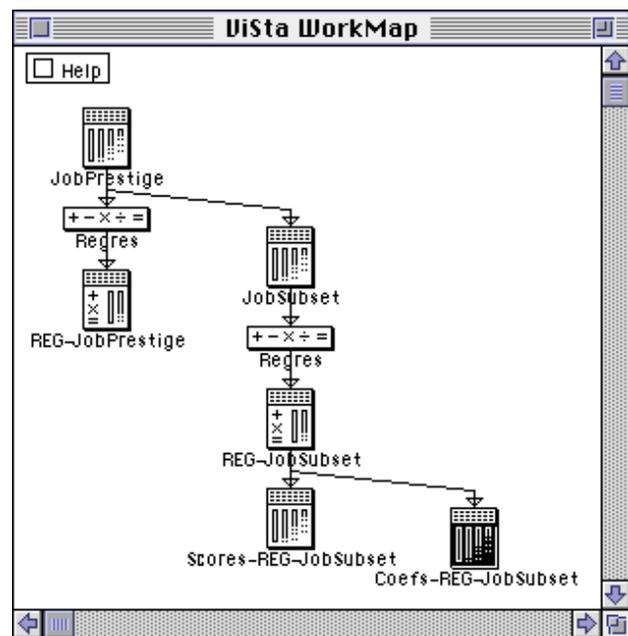


Figure 17: WorkMap after creating output datasets

Comparing these methods to Tierney's (1995) functions shows that we have only changed the first line of each piece of code. We made similar changes to the other functions presented by Tierney. Since ViSta is entirely based on object-oriented programming (see Young, 1994; Young & Lubinsky, 1995) this conversion added the new robust regression capability, while carrying along all other capabilities of the system (workmaps, guidemaps, menus, etc.).

Of course, once Tierney's code had been modified, we then had to modify ViSta's interface so that the user could use the new code. This involved modifying the action of the visualization's **Iterate** button so that it would give the user the choice of robust or monotonic regression (originally, it only allowed monotonic regression iterations). We also had to add plots to the visualization, and modify dialog boxes, axis labels, window titles, and other details appropriately. While these modifications took the major portion of the time, the basic point remains: We could take advantage of the fact that ViSta uses Lisp-Stat's object-oriented programming system to introduce a new analysis method by making it an option of an already existing analysis object, and we did not have to re-code major portions of our software. Furthermore, we did not have to jump outside of the ViSta/Lisp-Stat system to do the programming: No knowledge of another programming language was required. And finally, since ViSta is an open and extensible system, statistical programmers other than the developers can take advantage of its object oriented nature in the same way.

## 7.0 Conclusion

ViSta is a visual statistics system designed for an audience of data analysts ranging from novices and students, through those that are proficient data analysts, to expert data analysts and statistical programmers. It has several seamlessly integrated data analysis environments to meet the needs of this wide range of users, from guidemaps for novices, through workmaps and menu-systems for the more competent, on through command lines and scripts for the most proficient, to a guidemap-authoring system for statistical experts and a full-blown programming language for programmers.

As the capability of computers continues to increase and their price continues to decrease, the audience for complex software systems such as data-analysis systems will become wider and more naive. Thus, it is imperative that these systems be designed to guide users who need the guidance, while at the same time be able to provide full data-analysis and statistical programming power.

As we stated at the outset, our guiding principle is that data analyses performed in an environment that visually guides and structures the analysis will be more productive, accurate, accessible and satisfying than data analyses performed in an environment without such visual aids, especially for novices. However, we understand that visualization techniques are not useful for everyone all of the time, regardless of their sophistication. Thus, all visualization techniques are optional, and can be dispensed with or reinstated at any time. In addition, standard nonvisual data analysis methods are available. This combination means that ViSta provides a visual environment for data analysis without sacrificing the strengths of those standard statistical system features that have proven useful over the years. We recognize that it may be true that a single picture is worth a thousand numbers, but that this is not true for everyone all the time. And, in any case, pictures *and* numbers give the most complete understanding of data.

## 8.0 References

- Bann, C.M. (1996a) *Monotonic and Robust Multiple Regression*. MA Thesis, Psychometrics Laboratory, University of North Carolina, Chapel Hill, NC.
- Bann, C.M. (1996b) *ViSta Regress: Univariate Regression with ViSta, the Visual Statistics System*. L.L. Thurstone Psychometric Laboratory Research Memorandum (in preparation).
- Brieman, L. and Friedman, J.H. (1985) Estimating Optimal Transformations for Multiple Regression and Correlation. *J. Amer. Stat. Assoc.*, 77, 580-619
- Duncan, O.D. (1961) A Socioeconomic Index of All Occupations. In: Reiss, A.J., et al. (Eds.) *Occupations and Social Status*. Free Press. New York. pp. 109-38.
- Faldowski, R.A. (1995) *Visual Component Analysis*. Ph.D. Dissertation, Psychometrics Laboratory, University of North Carolina, Chapel Hill, NC.
- Lee, B-L. (1994) *ViSta Corresp: Correspondence Analysis with ViSta, the Visual Statistics System*. Research Memorandum 94-3, The L.L. Thurstone Psychometric Laboratory, University of North Carolina, Chapel Hill, NC

- McFarlane, M. & Young, F.W. (1994) Graphical Sensitivity Analysis for Multidimensional Scaling. *J. Computational and Graphical Statistics*, 3, 23-34.
- Tierney, L. (1990) *Lisp-Stat: An Object-Oriented Environment for Statistical Computing & Dynamic Graphics*. Addison-Wesley, Reading, Massachusetts.
- Tierney, L. (1995) Data Analysis Using Lisp-Stat. In: Fox, J. & Stine, R. *Sociological Methods & Research (Special Issue on Computing Environments)*, Vol 23 (3), pp. 329-351.
- Young, F.W. (1994) *ViSta – The Visual Statistics System: Chapter 1 – Overview; Chapter 2 – Tutorial*. May, 1994. Research Memorandum 94-1, The L.L. Thurstone Psychometric Laboratory, University of North Carolina, Chapel Hill, NC.
- Young, F.W., de Leeuw, J. & Takane, Y. (1976) Multiple and canonical regression with a mix of quantitative and qualitative variables: An alternating least squares method with optimal scaling features. *Psychometrika*, 41, 505-530.
- Young, F.W., Faldowski, R.A. & McFarlane (1993) M.M. Multivariate Statistical Visualization. In: Rao, C.R. (Ed.) *Handbook of Statistics*, 1993, 9, 959-998.
- Young, F.W. & Lubinsky, D.J. (1995) Guiding Data Analysis with Visual Statistical Strategies. *J. Computational and Graphical Statistics*, 4(4), (in press).
- Young, F.W. and Smith, J.B. (1991) Towards a Structured Data Analysis Environment: A Cognition-Based Design. In: Buja, A. & Tukey, P.A. (Eds.) *Computing and Graphics in Statistics*, 36, 253-279. New York: Springer-Verlag.

*PostScript error (--nostringval--, get)*