Paper 290-2007

An Introduction to SAS® Stat Studio: A Programmable Successor to SAS/INSIGHT®

Rick Wicklin, SAS Institute Inc.
Peter Rowe, CART Statistics & Modeling Team, Wachovia Corporation

OVERVIEW

SAS Stat Studio is a new statistical software product: a programmable successor to SAS/INSIGHT®. Stat Studio runs on a PC in the Microsoft Windows operating environment and connects with one or more SAS servers. Stat Studio is available in SAS 9.2.

Like SAS/INSIGHT, Stat Studio provides dynamically linked graphics for exploring multivariate data and a point-and-click interface for standard statistical analyses. However, Stat Studio extends the functionality of SAS/INSIGHT by providing a programming environment in which you can run SAS/STAT® or SAS/IML® analyses, and display the results in dynamically linked graphics and data tables.

Stat Studio is intended for high-end data analysts. These are innovative problem solvers who routinely analyze complex data by writing programs, and who also need dynamically linked graphics.

This paper uses financial data to illustrate both point-and-click and programming features of Stat Studio. The paper consists of four main sections. Section 1 discusses the intended audience for Stat Studio and introduces the main features of Stat Studio. Section 2 presents the financial data, and uses exploratory data analysis to discover features of the data and relationships between variables. Section 3 applies a standard logistic regression model. Section 4 presents two examples of writing a program to extend Stat Studio functionality. An appendix contains frequently asked questions about Stat Studio.

1. WHO SHOULD USE STAT STUDIO? WHAT DOES IT DO?

Stat Studio is designed for high-end data analysts with complex (and often messy!) data. High-end data analysts are proficient programmers who require dynamically linked graphics in order to fully understand and model intricate relationships between variables.

The high-end data analyst often begins an analysis by graphically exploring the data and by applying standard statistical methods. As characteristics of the data become clear, the analyst progresses to customized analyses. Dynamically linked graphics are necessary throughout the process and enable the analyst to move from simple to more complex models.

Stat Studio provides tools to facilitate each step in this process: dynamically linked graphics, a point-and-click interface to standard analyses, and a programming environment for customized analyses.

- Dynamically linked graphics are one of the primary tools of modern exploratory data analysis. The concept is simple: you can select observations in any tabular or graphical view of the data, and see those same observations highlighted in all other views of the data. You can use dynamically linked graphics to explore marginal distributions, identify outliers, and discover relationships between variables.
- You can use the Stat Studio graphical user interface (GUI) to run standard statistical analyses that analyze univariate distributions, fit explanatory models, and analyze multivariate relationships. These analyses are performed by using SAS/STAT procedures.
- You can use Stat Studio's integrated development environment to write, debug, and execute programs that combine the following:
  - the high-level matrix-oriented programming statements of the SAS/IML matrix language
  - the analytical power of SAS/STAT procedures

- the data manipulation capabilities of the SAS DATA step
- dynamically linked graphics

2. EXPLORATORY ANALYSIS USING THE STAT STUDIO GUI

The data examined in this paper were provided by the Wachovia Corporation. There are 50,000 observations and 30 variables in the data set. Each observation contains information about a customer who had a Prime Equity Line (PEL) of credit. This is a secured loan that uses home equity as collateral. The account either was still open in March 2005 or was closed in the previous 13 months. To preserve customer anonymity, no identifying variables are present.

This paper explores a small number of the variables:

- Closed_Flag: a variable that indicates whether the account is closed ('Y') or open ('N')
- Close_Reason: the reason the account was closed. The values for this variable are the following:

  missing (.): the account is open

  1: the customer moved or sold the home

  2: the customer refinanced with Wachovia

  3: the customer refinanced with a competitor, or gave no reason

- NbrSvcs: the number of different services a customer has with Wachovia
- Num_Accts: the number of accounts a customer has with Wachovia
- Tenure: the number of months the account has been open with Wachovia
- Age_Open: the customer's age when the account was opened
- High_Util_Rate: the ratio of the account variables High_Bal_Life_Amt to Credit_Line. High_Bal_Life_Amt is the highest historical balance on the account. Credit_Line is the maximum amount the customer can borrow on this account.

Wachovia analysts examine similar data to understand customer behavior. In particular, this paper focuses on identifying customers who are likely to close their PEL accounts.

Exploring Data Distributions: Binning, Zooming, and Identifying Outliers

When you open a data set in Stat Studio, you are initially presented with a tabular view of the data. The Stat Studio GUI provides menus from which you can choose to create plots or run statistical analyses for variables in the data table.

When you begin to analyze data, it is often useful to visualize the data distributions of the variables. In Stat Studio, it is easy to create a panel of histograms and bar charts: select a set of variables in a data table, and select **Graph ► Histogram** from the main menu, as shown in Figure 1.

This produces an array of bar charts and histograms, as shown in Figure 2. A few features of the data are apparent. The distribution of the Tenure variable (lower-left plot) has a long tail. The median tenure is about 24 months, and 96% of the customers have tenure less than 144 months.

The histogram of Age_Open (lower-center plot) shows some negative values, and a few large values, too. The Age_Open variable is created as the difference between the date that the PEL account was opened and the customer's birth date. It is easy to select and query dubious values of a variable to see if a value has been misrecorded. You can select the values of Age_Open that are less than 18 by dragging out a selection rectangle in the histogram and viewing those selected records in a data table. The birth dates of these customers are likely misrecorded. For example, about 50% of these customers have the same birthday: January 1.

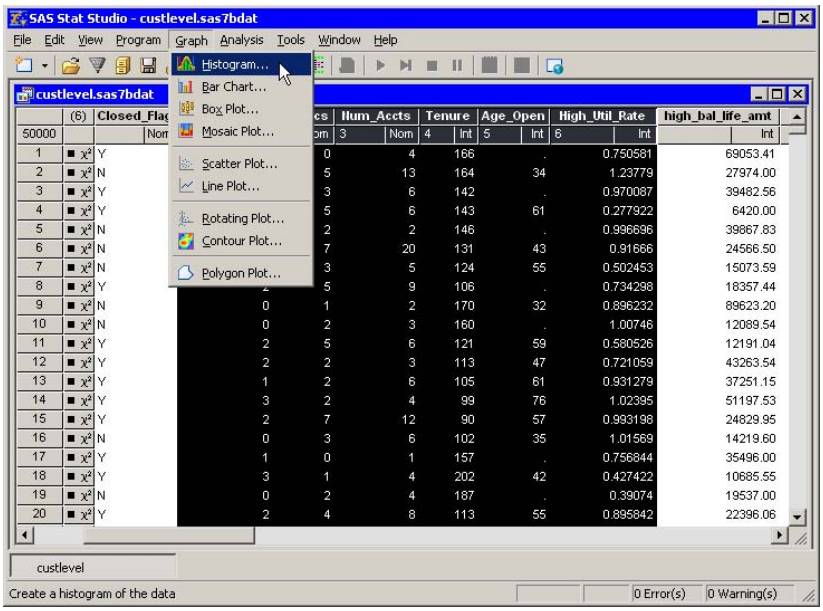

Figure 1. Creating Plots

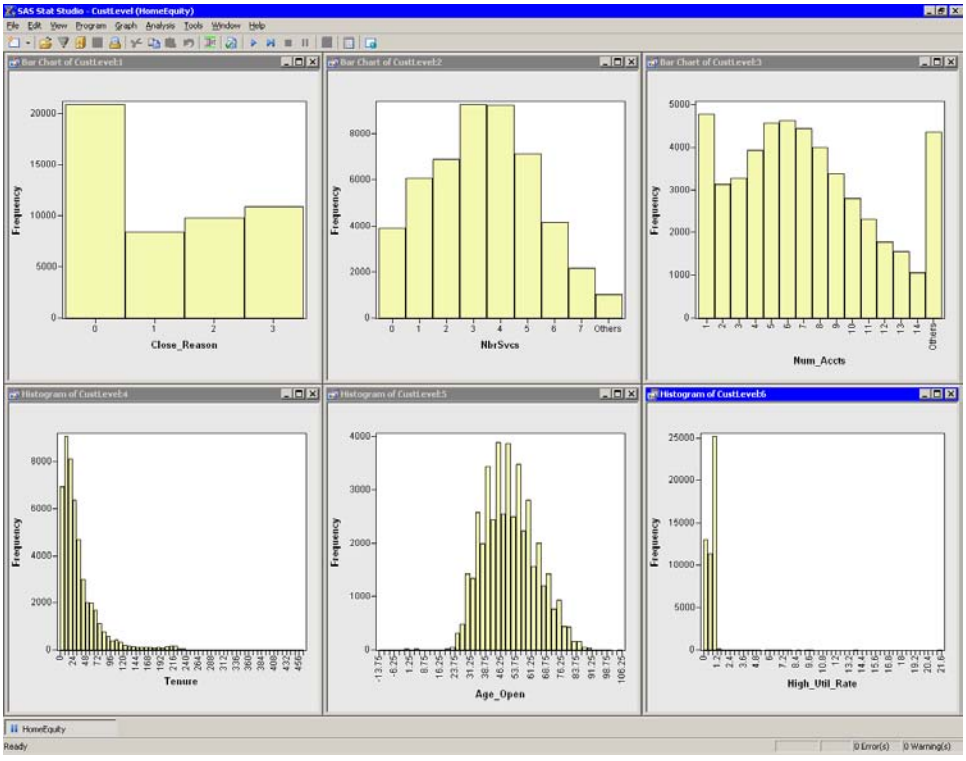

Figure 2. A Matrix of Histograms and Bar Charts

The histogram of High_Util_Rate has many observations greater than 1. Because customers pay interest on the loan, the balance of an account can be greater than the credit line. For example, there are about 7,000 customers with High_Util_Rate in the interval (1,1.022]. Values greater than 1.2 probably indicate that the credit line for this account is lower now than it was in the past.

Identifying data quality issues is important in predictive modeling. Misrecorded data can be replaced with a missing value, or the observations can be excluded. For example, you can replace values of Age_Open that are less than 18 with missing values. If you want to exclude observations from your analyses, you can use the **Exclude from Plots** and **Exclude from Analysis** features of Stat Studio. Excluded observations do not appear in any graph and are not used in statistical analyses. For example, you can exclude accounts whose value for High_Util_Rate is greater than or equal to 1.2, as shown in Figure 3.

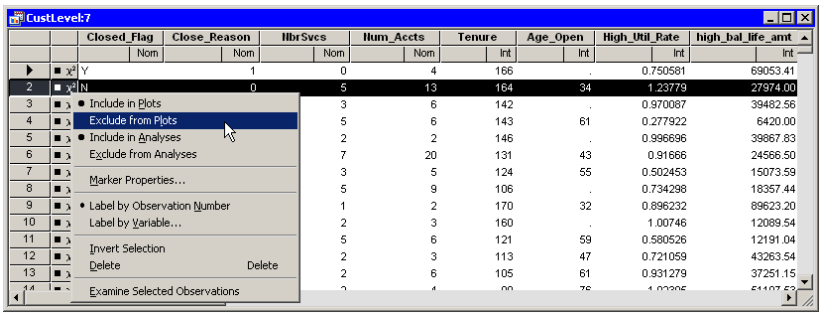

Figure 3. Excluding Observations

When exploring data, it is convenient to rebin histograms and interactively zoom in on regions of plots that are of interest. Stat Studio provides both of these features. Figure 4 show an example of interactively rebinning a histogram. If you drag the pointer around in the plot area, then the histogram is rebinned. Dragging the pointer horizontally changes the anchor position. Dragging the pointer vertically changes the bin width.

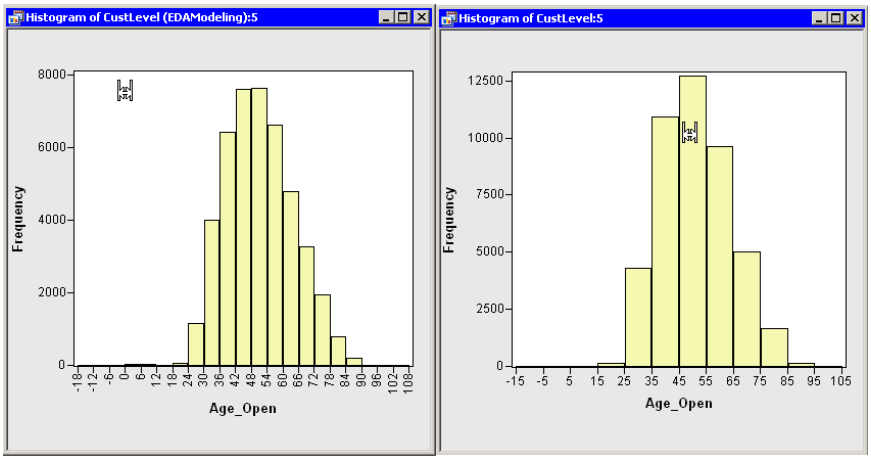

Figure 4. Two Snapshots While Interactively Rebinning a Histogram

Similarly, you can interactively zoom in on any feature in a plot. For example, if you decide to focus on customer accounts that have been open at most 12 years, you can zoom to the interval [0,144] on the histogram for Tenure.

Through these manipulations you can identify univariate outliers and miscoded data, exclude customers whose credit limit has been decreased, rebin histograms, and zoom to the bulk of the data.

Exploring Reasons for Closing a PEL Account

When you understand the marginal distribution of each variable, you can begin to explore relationships between variables. You can select observations by clicking with a mouse. Clicking on a marker in a scatter plot selects the corresponding observation. Clicking on a bar in a histogram or bar chart selects all observations represented by that bar. Dragging a rectangle selects all observations within that rectangle.

You can use exploratory data analysis to discover relationships between variables. Of particular interest for these data are relationships between the response variable Close_Reason and variables that might explain the reasons why customers close their PEL accounts.

The Close_Reason variable is categorical with four values. A simple exploratory technique for a categorical variable is to select each category of the response variable and to examine the marginal distributions of the selected observations in plots of other variables.

If you click on the bar corresponding to the '2' category in the upper-left plot of Figure 5, you select customers who closed their accounts because they refinanced with Wachovia. The highlighted portions shown in the other plots represent those same accounts. Refinancing a loan is a favorable event for Wachovia, since it leads to continued revenue and preserves an existing customer relationship.

In contrast, the accounts highlighted in Figure 6 represent customers who either refinanced their loan with a competitor or else closed their account for an unspecified reason. Analysts at Wachovia want to predict which customers might make this choice in the future so that Wachovia can take steps to retain the account.

The highlighted portion of the bar chart for the NbrSvcs variable (upper-center plot) shows the distribution of the number of services in each case. The customers who refinanced their loans with Wachovia (Figure 5) tend to have a larger number of services and accounts with Wachovia than customers who refinanced with a competitor or failed to give a reason (Figure 6).

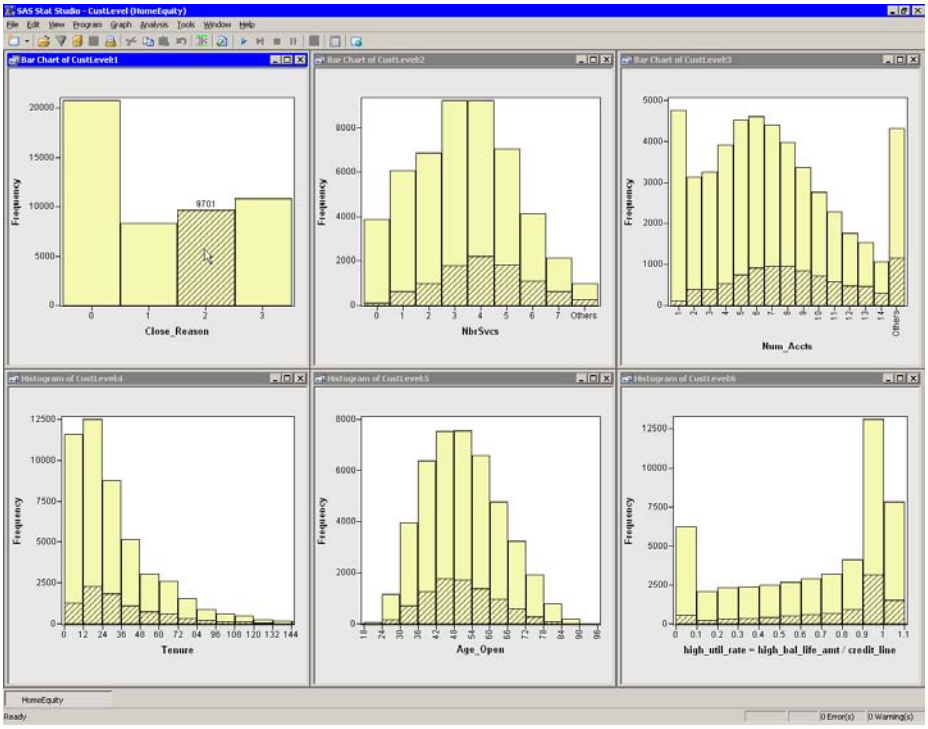

Figure 5. Accounts Refinanced with Wachovia (Close_Reason=2)

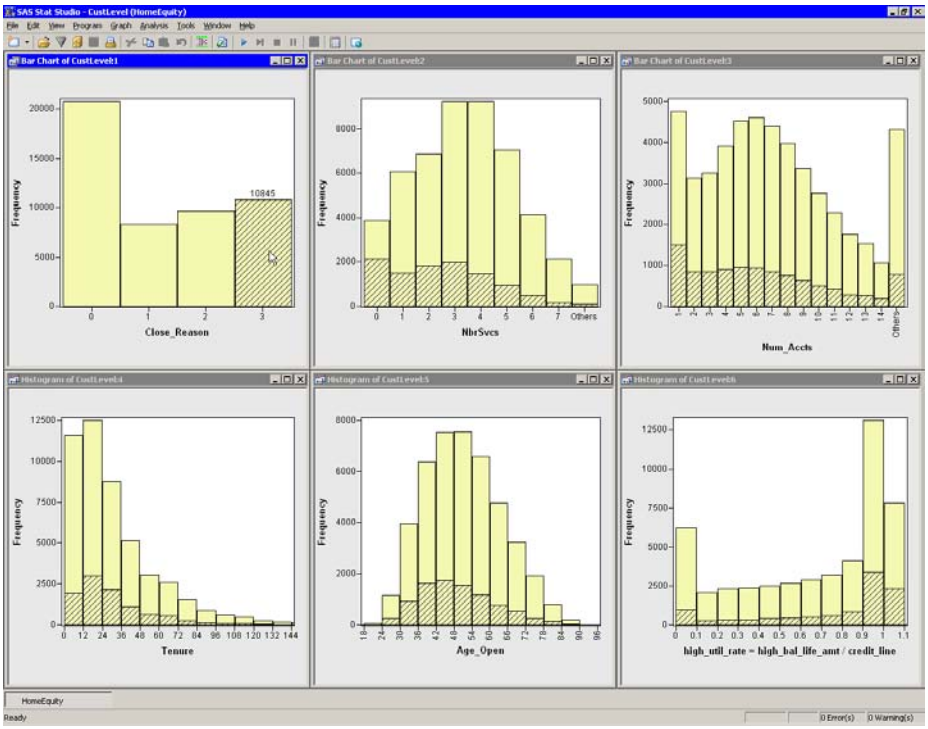

Figure 6. Accounts Refinanced with Competitor, or Unspecified Reason (Close_Reason=3)

Thus, NbrSvcs might help identify customers who are willing to refinance with Wachovia. Similar exploratory analyses can identify other variables that discriminate between categories of Close_Reason. Although it is not presented in this paper, you can use Stat Studio to construct a generalized linear model that predicts categories of Close_Reason.

3. STANDARD STATISTICAL ANALYSES USING THE STAT STUDIO GUI

Analysts at Wachovia want to understand the reasons why customers close accounts, but they also want to predict whether a customer will close a PEL account, for any reason. If it is possible to identify customers with a high probability of closing their account, then Wachovia can proactively contact these customers through promotional mailings and special offerings.

This section formulates a logistic model to predict whether a customer is likely to close a PEL account.

Identifying Explanatory Variables: Which Customers Are Likely to Close a PEL Account?

The binary variable Closed_Flag indicates whether a PEL account was closed in the past 13 months. In formulating a model to predict Closed_Flag, you can use dynamically linked graphics to identify variables that are related to Closed_Flag.

Figure 7 shows a histogram of the variable High_Util_Rate. The plot on the right is called a *spine plot* of the Closed_Flag variable. In a spine plot, the width of a bar corresponds to the frequency of a category; the height of the shaded region corresponds to the proportion of selected observations in each category.

The accounts selected in the histogram have had (at some point in the past) a balance that is large, relative to their line of credit. In other words, these customers used most or all of their line of credit. A relatively large proportion of these accounts are closed. In contrast, relatively few accounts with a low usage rate (not shown) are closed. Therefore, High_Util_Rate might be useful as an explanatory variable for predicting whether a customer is likely to close a PEL account.

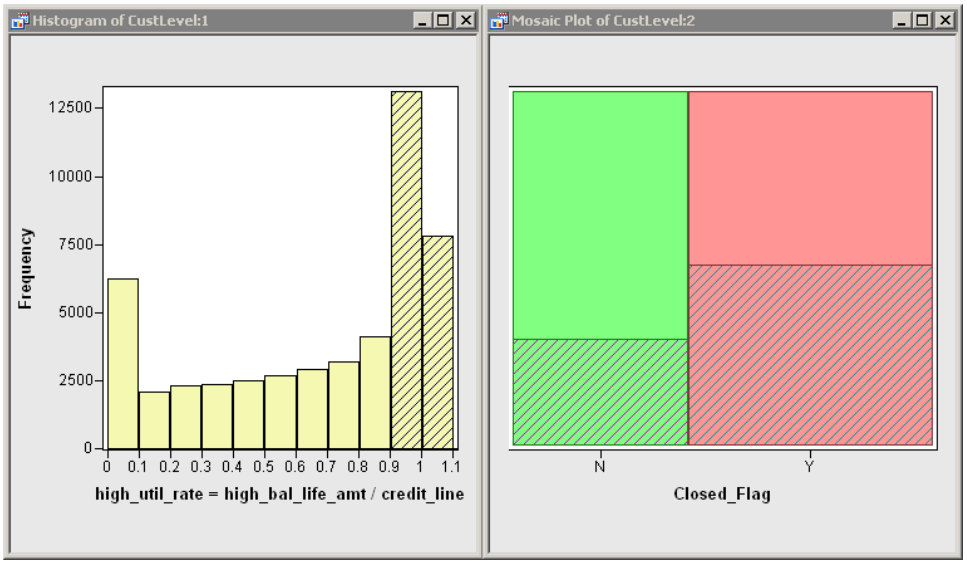

Figure 7. Selecting Accounts with High Historical Usage

Similarly, Figure 8 displays a box plot of the variable Tenure, truncated at 144 months. The lower quartile of Tenure corresponds to customers who have been with Wachovia for less than 12 months. These accounts are selected in Figure 8. Unsurprisingly, the figure shows that customers who have been with Wachovia for a short time are less likely to have closed their account.

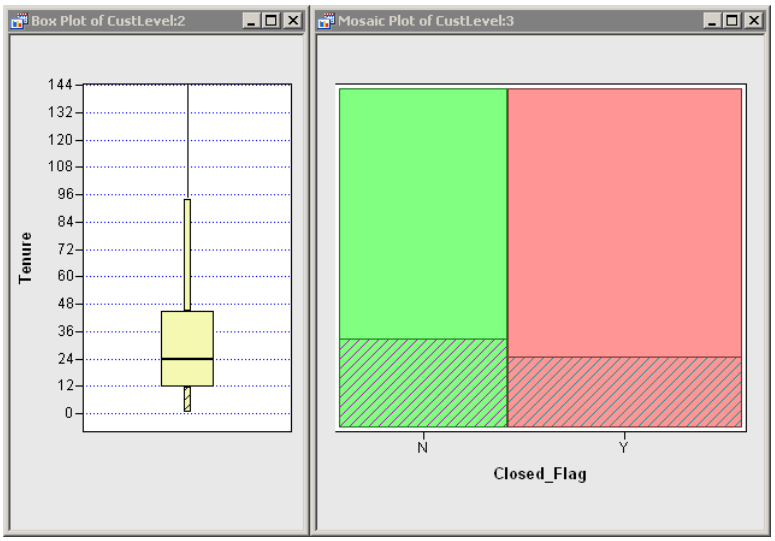

Figure 8. Selecting Accounts with Low Tenure

A similar figure (not shown) indicates that customers with longer tenure are more likely to have closed their PEL account in the previous 13 months. Therefore Tenure might be useful as an explanatory variable.

Although Tenure is a continuous variable, it is sometimes useful to group customers into categories such as “new,” “recent,” “loyal,” and so on. The next section describes how to bin the Tenure variable into its quartiles.

Variable Transformations and Creating New Variables

Data analysts frequently create new variables from existing variables. Stat Studio provides a Variable Transformation Wizard that enables you to quickly apply standard transformations to your data. These include normalizing transformations (such as logarithmic and power transformations), logit and probit transformations, affine transformations (including centering and standardizing), and rank transformations.

You can also create your own transformations within the Variable Transformation Wizard by using SAS DATA step syntax and functions.

For example, you can create a classification variable with a small number of levels by binning the Tenure variable into quartiles. You can create a new variable, TenureQ, with values Q1–Q4. The quantiles can be read from the box plot shown in Figure 8, or you can select **Analysis ► Distribution Analysis ► Descriptive** from the main Stat Studio menu to compute descriptive statistics and quantiles for the Tenure variable.

To create this binned variable, select **Analysis ► Variable Transformation** from the main menu. The Variable Transformation Wizard appears, as shown in Figure 9.

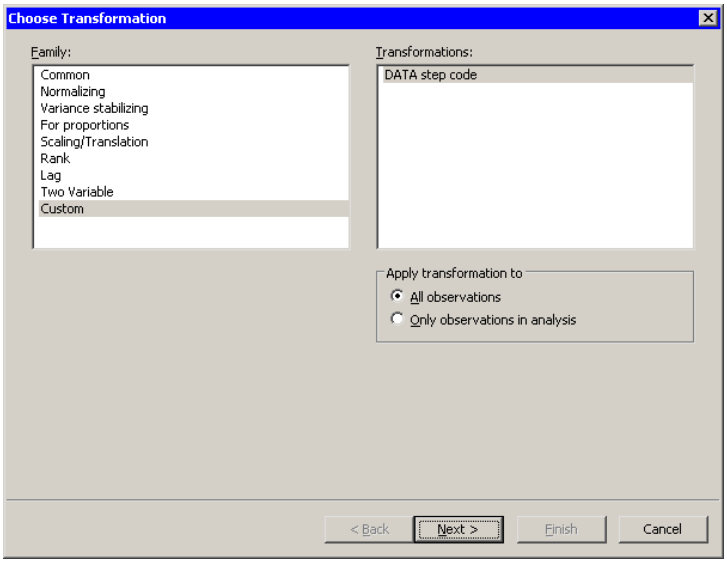

Figure 9. Selecting a Transformation

When you select **Custom** from the **Family** list and click **Next**, the wizard displays a window in which you can enter a DATA step that defines a new variable, as shown in Figure 10. When you click **Finish**, the newly created variable, TenureQ, is copied to the Stat Studio data table. This variable is used in the next section to predict the probability that a customer will close a PEL account.

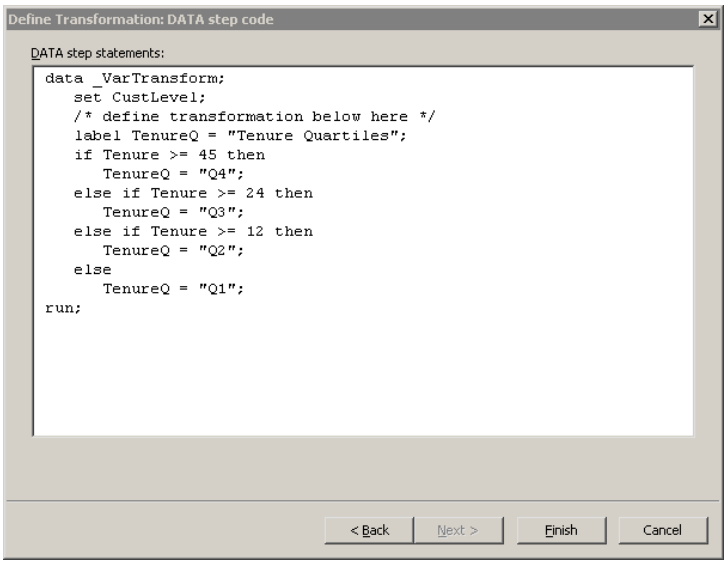

Figure 10. Entering a Custom Transformation

Creating a Logistic Analysis

The Stat Studio GUI provides standard statistical analyses. You can analyze a univariate distribution by computing descriptive statistics, creating quantile-quantile plots, and fitting parametric and kernel density estimates for distributions.

You can fit parametric and nonparametric regression models such as loess models, robust regression models, logistic models, and generalized linear models. You can analyze multivariate relationships with correlation analysis, reduce dimensionality with principal component analysis, and examine relationships between nominal and continuous variables with discriminant analysis. Stat Studio uses SAS/STAT procedures to produce the analyses.

As an example, this section describes how to construct a logistic model to predict the probability that a customer will close a PEL account. Wachovia analysts routinely form predictive models involving dozens of explanatory variables, but this simple model uses only one classification variable (TenureQ) and a single continuous variable (High_Util_Rate).

Figure 11 shows how to initiate a logistic analysis from the Stat Studio main menu.

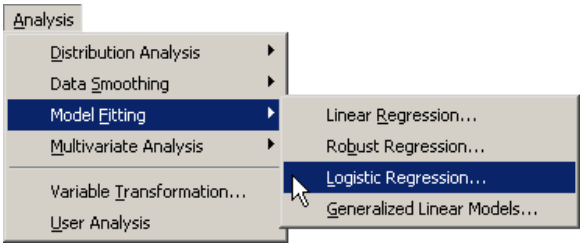

Figure 11. Selecting a Logistic Analysis

When you select an analysis, a dialog box appears that enables you to specify a model. You can also specify options to create certain plots, tables, and output variables such as predicted probabilities, residuals, and influence diagnostics.

The left side of Figure 12 shows a tab that you can use to specify general linear effects and interactions for variables in the model. This example specifies a model that contains two main effects and a crossed effect. The right side of the figure shows a tab used to specify plots.

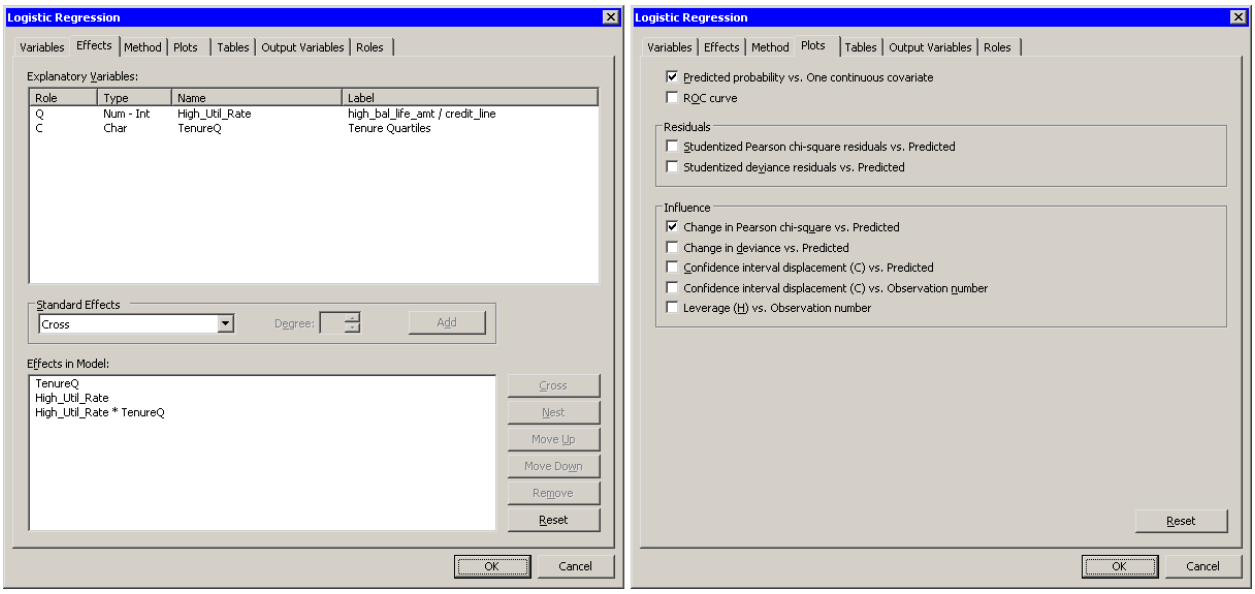

Figure 12. Two Tabs in a Logistic Analysis: Effects and Plots

When you click **OK** in the dialog box, the options you specified are used to call the LOGISTIC procedure. Output variables created by the procedure are added to the Stat Studio data table. All plots created by the analysis are dynamically linked to all other plots and data tables.

Figure 13 shows results for the logistic analysis. The output from the LOGISTIC procedure appears in the lower-left corner of the figure. The line plot in the upper-right corner shows the predicted probability of a customer closing a PEL account, plotted versus High_Util_Rate and grouped according to the levels of TenureQ. The model predicts that the probability of closing the account is relatively small for customers who have used very little of their line of credit and is smallest for customers in the first quartile of Tenure.

The model predicts the highest probability of closure for customers who have used most of their line of credit. For these customers, the quantiles of Tenure make little difference in the model's prediction. It is interesting to note how the effect of Tenure varies with High_Util_Rate.

The observations selected in the lower-right scatter plot of Figure 13 correspond to customers whose observed behavior was different from that predicted by the model. These customers did not close their PEL accounts, although the model assigns them a high probability of doing so.

Identifying customers that do not fit the model (in general, identifying outliers) is an important step in constructing more complex—and more realistic—models. You can select customers who do not fit a model, and look at plots of other potential explanatory variables. If a variable not currently in the model helps to predict the outliers, you can create a new model that incorporates this variable. You can then examine outliers for the new model, and continue this process until you are satisfied with your model.

But what do you do when your data are not amenable to the standard analyses built into the Stat Studio GUI? High-end data analysts are often confronted with this situation. They spend a lot of time writing programs to implement customized analyses. The next section discusses writing programs in Stat Studio.

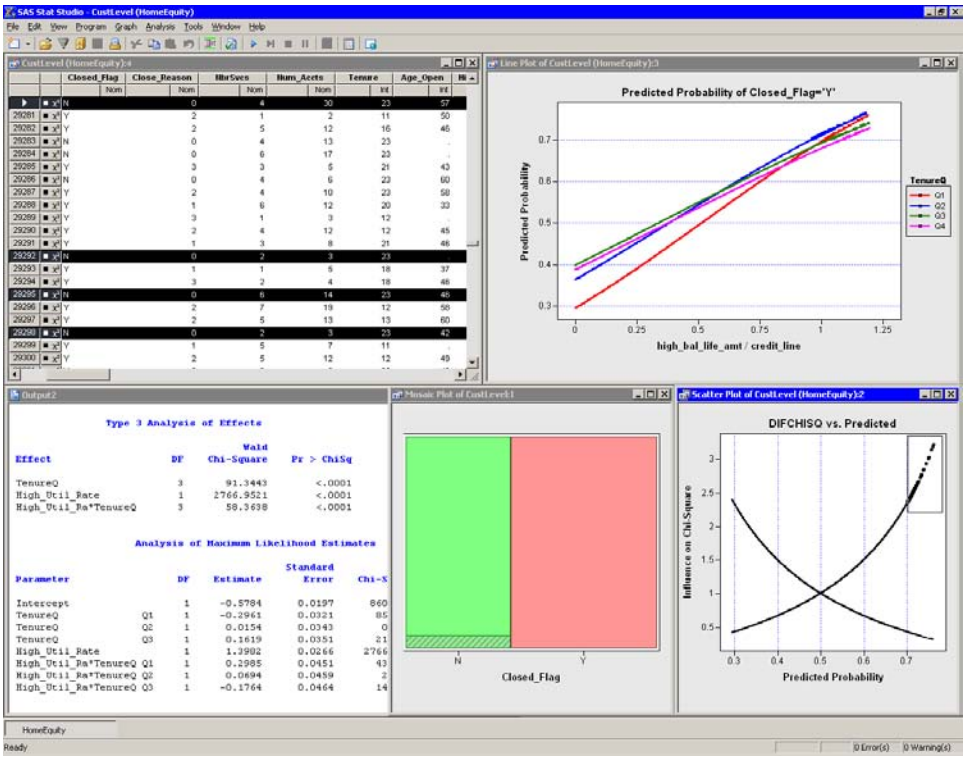

Figure 13. Examining Influence Diagnostics for a Logistic Model

4. WRITING PROGRAMS IN STAT STUDIO

The Stat Studio programming language, called *IMLPlus*, is an enhanced version of the SAS/IML programming language. SAS/IML provides features such as matrix-vector operations and a rich library of numerical and statistical functions. IMLPlus extends SAS/IML to provide the capability to create and manipulate statistical graphics, and to call SAS procedures.

The first example in this section writes a program to create plots, position them on the screen, and change their default characteristics. The second example is a custom analysis that calls SAS procedures.

Example 1: Creating Graphics

The financial data at Wachovia are updated regularly. Analysts often need to rerun models and examine trends in the updated data. Manually creating the same graphs every week with the Stat Studio GUI would become tedious. Instead, you can write a program that creates the graphs and also zooms or rebins them.

To illustrate, the following IMLPlus statements create some of the graphs in Figure 5. Each group of statements is briefly described in the text. All the methods used in this program are documented in the Stat Studio online Help.

The program first opens the data and loads the contents into memory on the client, as shown in the following statements:

```
/* open SAS data set into Stat Studio */
declare DataObject dobj;
dobj = DataObject.CreateFromFile("Wachovia/CustomerData");
```

The in-memory data are managed by a class, called the DataObject class, that provides methods to query, retrieve, and manipulate the data. It manages graphical information about observations such as the shape and color of markers, the selected state of observations, and whether observations are displayed in plots or excluded.

The IMLPlus programming language borrows ideas from object-oriented programming, particularly Java. In SAS/IML, all variables are matrices. In IMLPlus, a variable is implicitly assumed to be an IML matrix unless the variable is declared to refer to an object. You can specify that an IMLPlus variable refers to an object by using the declare keyword. Thus dobj is a variable that refers to an object of the DataObject class.

To call methods in IMLPlus, you use a "dot notation" syntax in which the method name is appended to the name of the object. In the following statements, the SelectObsWhere method in the DataObject class selects all observations that satisfy High_Util_Rate ≥ 1.2. These observations are then excluded from all plots and analyses.

```
/* exclude certain observations */
dobj.SelectObsWhere("High_Util_Rate", WHERE_GE, 1.2);
dobj.IncludeInPlots(false);    /* exclude selected obs */
dobj.IncludeInAnalysis(false);
```

The following statements demonstrate that you can use standard IML syntax, operators, indices, and functions:

```
/* create numeric or character matrices in SAS/IML */
width = 100 / 3; /* 33% of the width of Stat Studio's main window */
height = 50;     /* 50% of the height of Stat Studio's main window */
varName = {"Close_Reason" "NbrSvcs" "num_accts"};
```

The width and height variables are IML scalars, and the varName variable is a 1×3 character matrix.

The following statements use a SAS/IML DO loop to create and position three bar charts across the top of the screen. Thus, you can use SAS/IML to create, manage, and manipulate dynamically linked graphics.

```
/* create and position BarCharts in top row */
declare BarChart[] bar = new BarChart[3]; /* array of BarCharts */
do i = 0 to 2;
   bar[i] = BarChart.Create( dobj, varName[i+1] );
   bar[i].SetWindowPosition(i*width, 0, width, height);
end;
```

The next set of statements create and set properties for a scatter plot of two variables:

```
/* create ScatterPlot, zoom, and set size of markers */
declare ScatterPlot plot;
plot = ScatterPlot.Create( dobj, "Tenure", "High_Util_Rate" );
plot.SetAxisTickUnit(XAXIS, 12);      /* ticks every 12 months */
plot.SetAxisViewRange(XAXIS, 0, 144); /* view 12 years */
plot.SetMarkerSize(2);                /* use small markers */
plot.SetWindowPosition(0, 50, width, height);
```

You can also define and call IML modules. Stat Studio comes with a library of modules that you can call to perform common tasks in data analysis. For example, the following module colors observations according to the four values of the Close_Reason variable:

```
run ColorCodeObs( dobj, "Close_Reason", BLUE//CYAN//YELLOW//RED, 4 );
```

Figure 14 displays the Stat Studio program editor (which color-codes keywords) and the graphs created by running the program. After the graphs are created, you can interact with the graphics as described in previous sections. For example, in Figure 14, the selected observations represent customers with more than two years of tenure who have used less than 50% of their credit line.

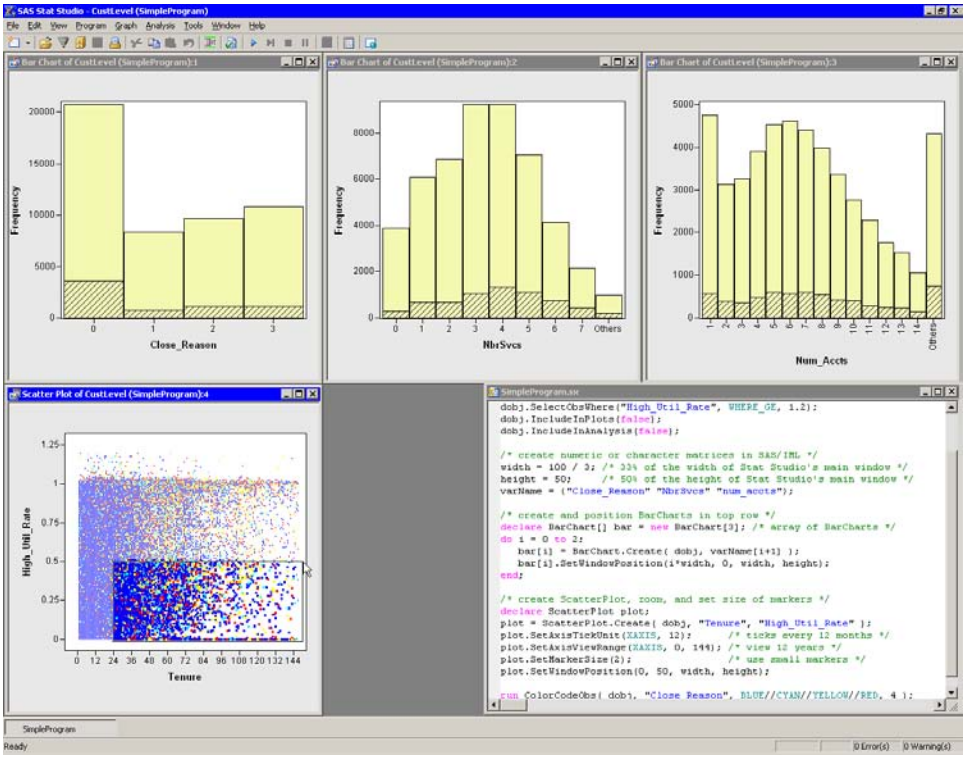

Figure 14. A Simple IMLPlus Program

Example 2: Writing a Custom Analysis

Stat Studio provides standard statistical analyses and graphics, but perhaps the most powerful feature of Stat Studio is that it enables you to create customized analyses. You can write programs that call any SAS procedure, and you can visualize the results of that procedure in dynamically linked graphics and data tables. In fact, all built-in analyses in Stat Studio are written in IMLPlus.

The example in this section uses only the 29,134 accounts closed during a 13-month time period. The program enables you to explore trends in the mean number of accounts closed each month, and to examine relationships between the closed accounts and the means of other variables.

The main steps of the program are as follows:

1. Write the data to the SAS server, if it is not already there.
2. Run the SORT procedure to sort the data by DateClosed and Close_Reason.
3. Run the MEANS procedure to create an output data set containing the number of observations in each BY group, and the mean of the Tenure variable for each BY group. (You could also ask for statistics on other variables.)
4. Read the output data set into Stat Studio.
5. Create a line plot of the counts for each month, grouped by the reason the account was closed.

6. Create a histogram of the mean of Tenure for each BY group.

It is important to remember that the plots created in steps 5 and 6 are dynamically linked; it is the linking that makes this analysis successful. Appendix B lists the IMLPlus program that implements these steps.

Figure 15 shows the Stat Studio windows created by running the program. The program editor is in the upper-left window; the output from the MEANS procedure is in the lower-left window. The line plot is dynamically linked to the histogram.

You can examine the line plot to determine whether there are any trends. There appears to be seasonality in the curve labeled "Moved," which corresponds to customers who sold the house used as collateral for the PEL loan. Of the customers who closed their accounts in January and February, relatively few cited selling their home ("Moved") as the reason. In contrast, many of the customers who closed their accounts in summer months cited moving as the reason. This seasonality corresponds to peaks and valleys in real estate activity.

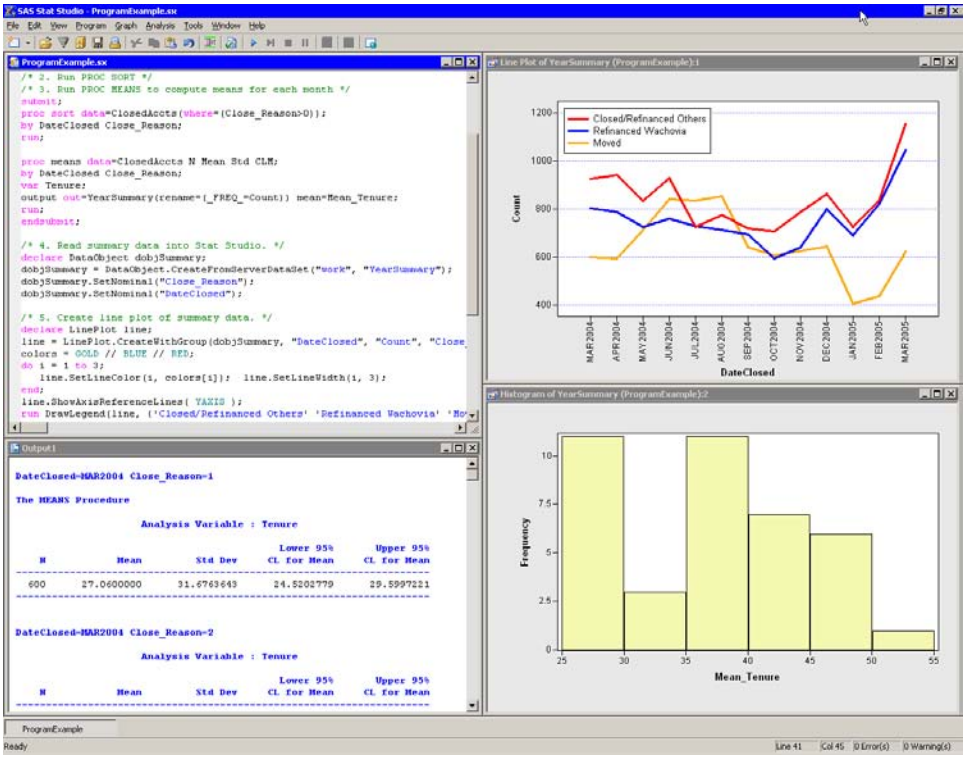

Figure 15. A Custom Analysis That Calls SAS Procedures

Notice the gap between the number of customers who refinanced with competitors and the number who refinanced with Wachovia. This gap was relatively small after December 2004; the difference was larger in the previous spring. The smaller gap might indicate that Wachovia became increasingly successful in competing for refinancing during the year.

After the program runs, you can select relatively large values of the Mean_Tenure variable, as shown in Figure 16. These same observations are shown selected in the line plot. It turns out that these values correspond primarily to customers who chose to refinance their account with Wachovia! Said differently, the mean tenure of customers who refinanced with Wachovia (during a given month) is higher than that of customers who did not.

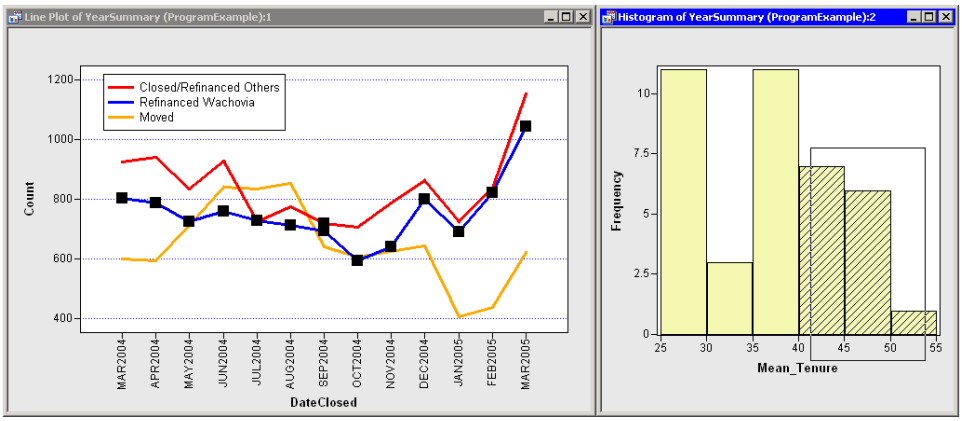

Figure 16. Linked Graphics Showing Relationship between Tenure and Closed Accounts

CONCLUSIONS

Stat Studio is a programmable successor to SAS/INSIGHT. Stat Studio provides tools for exploratory data analysis and predictive modeling. Stat Studio provides dynamically linked graphics for exploring data and for investigating the relationships between variables. You can use the graphics to identify outliers and miscoded data. You can rapidly rebin histograms and zoom in on interesting features in plots. You can create new variables by applying standard variable transformations or by writing SAS DATA step statements.

Stat Studio enables you to run standard statistical analyses, examine model fit, and identify influential observations. When standard analyses are inadequate, you can use the Stat Studio programming environment to write customized analyses of arbitrary complexity and visualize the results with linked graphics.

APPENDIX A: FREQUENTLY ASKED QUESTIONS ABOUT STAT STUDIO

Q: I currently use SAS/INSIGHT for exploratory data analysis. Can I use Stat Studio instead?

A: Yes. Like SAS/INSIGHT, Stat Studio provides dynamically linked graphics and a point-and-click interface to standard statistical analyses.

Q: Does Stat Studio do anything SAS/INSIGHT doesn't do?

A: Yes, quite a bit! You can extend Stat Studio's abilities by writing custom analyses. You can use IMLPlus statements to add legends, curves, text, or other graphics to a plot. Stat Studio provides five analyses and one graph type not found in SAS/INSIGHT. Stat Studio provides additional techniques for ordering categories of a nominal variable, selecting observations that satisfy complex criteria, managing graphs and multiple workspaces, and copying plots to the Windows clipboard.

Q: Can I paste graphs into Microsoft Word and PowerPoint?

A: Yes. You can copy a graph to the Windows clipboard. From there you can paste the graph as a bitmap or in Enhanced Metafile Format (EMF) into Word, Outlook, PowerPoint, etc.

Q: How much data can I explore using Stat Studio?

A: That depends on your computer's RAM capacity and on the types of analyses you intend to perform. By definition, dynamically linked graphics must respond quickly to the changes and manipulations of the

analyst. On a PC with a 1.8 GHz CPU and 512 MB of RAM (typical of a PC purchased in 2004–2005), Stat Studio can help you graphically analyze dozens of variables and tens of thousands of observations. Visualization of data with graphics such as histograms and box plots remains feasible for hundreds of thousands of observations. Scatter plots of this many observations suffer from overplotting.

Q: Can I read a SAS data set on my PC? What about in SASUSER?

A: You can read data sets on your local PC and on a networked drive. You can read data sets on the SAS server from any predefined libref, or from librefs you define with the LIBNAME statement.

Q: I am not a SAS programmer. Can I still use Stat Studio?

A: Yes. You can use the Stat Studio graphics and built-in analyses without any programming. Stat Studio provides a graphical user interface to commonly used statistical graphs and analyses.

Q: I am an experienced SAS/STAT programmer. What do I need to learn to write programs in Stat Studio?

A: There is documentation titled *Stat Studio for SAS/STAT Users*. This is a short book (about 65 pages and 40 figures) designed to teach you the basics of Stat Studio programming.

Q: Can I call any SAS procedure from within Stat Studio?

A: Yes. If you have a license for SAS/ETS®, SAS/QC®, etc., you can call procedures in those products.

Q: I've written a macro in SAS that implements a specialized analysis. Can I call it from Stat Studio?

A: You can call computational macros from within a SUBMIT/ENDSUBMIT block. Because the macro executes on the SAS server, you cannot run a macro that tries to create a window, such as for high-resolution graphics.

Q: I've written a Stat Studio program. Can I call it from the Stat Studio main menu?

A: Yes, provided that you save your program as an IMLPlus module. When you select **Analysis ► User Analysis** from the main menu, Stat Studio runs an IMLPlus module that you can modify to call any module you've written. You can also create your own menus (called *action menus*) and attach them to plots and data tables. Each menu item calls a module that you specify.

Q: Can Stat Studio record my session and play it back at a later time?

A: No, but you can write programs that create graphs, modify common graphical attributes, manipulate the data, and, in general, reproduce most graphical analyses.

APPENDIX B: PROGRAM LISTING

```
/* Open Wachovia data (on client PC). */
declare DataObject dobj;
dobj = DataObject.CreateFromFile("Wachovia/CustomerData");

/* 1. Write data to SAS server. */
dobj.WriteVarsToServerDataSet({"DateClosed" "Close_Reason" "Tenure"},
                              "work", "ClosedAccts", true );

/* 2. Run PROC SORT. */
/* 3. Run PROC MEANS to compute means for each month. */
submit;
proc sort data=ClosedAccts(where=(Close_Reason>0));
by DateClosed Close_Reason;
run;

proc means data=ClosedAccts N Mean Std CLM;
by DateClosed Close_Reason;
var Tenure;
output out=YearSummary(rename=(_FREQ_=Count)) mean=Mean_Tenure;
run;
endsubmit;

/* 4. Read monthly means into Stat Studio. */
declare DataObject dobjSummary;
dobjSummary = DataObject.CreateFromServerDataSet("work", "YearSummary");
dobjSummary.SetNominal("Close_Reason");
dobjSummary.SetNominal("DateClosed");

/* 5. Create line plot of monthly means. */
declare LinePlot line;
line = LinePlot.CreateWithGroup(dobjSummary, "DateClosed", "Count", "Close_Reason");
colors = GOLD // BLUE // RED;
do i = 1 to 3;
   line.SetLineColor(i, colors[i]);  line.SetLineWidth(i, 3);
end;
line.ShowAxisReferenceLines( YAXIS );
run DrawLegend(line, {'Closed/Refinanced Others' 'Refinanced Wachovia' 'Moved'},
    10, colors[3:1], GetLegendStyle(3, SOLID), -1, WHITE, 'ILT' );

/* 6. Create histogram of monthly means of Tenure. */
declare Histogram hist;
hist = Histogram.Create( dobjSummary, "Mean_Tenure");
hist.SetWindowPosition(50,50,50,50);
```

CONTACT INFORMATION

Rick Wicklin
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
www.sas.com/statistics