

Evaluating the Use of Enterprise Guide in Introductory Statistics Classes

Sandra B. Donaghy and Joy M. Smith, North Carolina State University, Raleigh, NC

ABSTRACT

Enterprise Guide (EG), a Windows thin client, provides a point and click interface to the SAS® System. EG allows users to access all data types supported by the SAS System and utilize the computing power of any Version 8 SAS Server to generate professional reports and graphics from the PC client interface.

In addition to the tasks that are available in the point and click environment, EG provides a program editor window where the full power of the SAS System is available through traditional SAS programming. The code generated using the point and click environment is viewable and can be useful for learning SAS programming.

This paper includes a handout that we plan to use in a pilot study this summer and our personal evaluation of EG for use in beginning statistics classes. The presentation will include an online demo and discussion of the pilot study.

INTRODUCTION

At NCSU, graduate students use SAS for their research data analysis. Many students report that prospective employers prefer candidates who are skilled SAS programmers. For these reasons, it would be very helpful if students were exposed to SAS in their introductory statistics classes. The complexity of the SAS System has made it difficult to introduce in beginning statistics classes where the focus is on teaching statistical concepts. We are currently evaluating EG to determine if it will make using SAS less complex for both students and faculty.

We intend to conduct a pilot study this summer to evaluate the use of EG in statistics classes. We believe students will be able to quickly perform homework assignments without having to learn SAS syntax. In addition, students who want to learn SAS programming could review the generated code. We are optimistic that EG will satisfy our need for a point and click interface to SAS.

In order for EG to be valuable, it must provide easy data access, analysis, and reporting with minimal need for support. This pilot study should help identify EG features that students find confusing. We are hopeful that the EG interface will provide a gentle introduction to SAS. As the students progress in their studies, they will benefit from learning SAS code and the code window should be useful at that point. The code

window will be used in classes where instructors provide sample SAS programs to their students. The results of this pilot study will be shared at the paper presentation.

In addition to teaching, we provide computer and statistical consulting to researchers on campus. EG may be useful to these faculty, staff, and graduate students when they analyze research data. EG will affect this consulting, but we can only speculate on what the effects will be. We are glad that the option to write our own code exists in case a task does not exist or we do not know how to accomplish a task using the EG menus and tasks.

This paper is structured as a student handout so that large parts of it can be used in the pilot study. The handout introduces basic EG concepts and provides one practice exercise. The concepts introduced were carefully chosen to meet the anticipated needs of beginning statistics students. EG has many features that are not covered. We hope to encourage the use of EG by keeping this introduction simple for the instructors and students. We hope that students will benefit from the exposure to SAS and be better prepared to learn more advanced SAS features in the future.

TESTING ENVIRONMENT

Our tests were run on Microsoft Windows NT 4.0 and Microsoft Windows 2000 Professional Edition using EG version 1.2.0.242 and SAS version 8.2 loaded locally. No effort has been made to evaluate processing on remote servers since this arrangement will not be necessary in the classroom.

Currently at NCSU, there are two PC environments in which we anticipate students and researchers will use EG. In the computer labs, licensed software packages, including SAS, are distributed using Novell Client V4.60 for Windows NT and Zenworks Application Launcher. In this environment, student files are stored in their personal network file space. This storage space is accessed using Andrew File System (AFS). The SASUSER library is located in this network space. This setup provides automatic backups and private storage for student files. Processing and workspace are on the local PC. Statistics classes will be taught in these labs, and students who do not have access to a personal computer may use these labs for

their homework. Systems administrators will be responsible for EG Administrator settings.

Alternatively, licensed SAS users may load SAS and EG on personal computers. Many of these SAS users will have networked drives from which data can be accessed, but we expect they will process their data locally. Individuals using this setup will be responsible for the EG Administrator settings. In our experience no modifications are required, so this should not present a problem

We have noticed WORK libraries left behind by abnormally terminated SAS sessions in both of these environments. Any user can delete these libraries from the PC they are logged on to, but we have not found an efficient mechanism for removing these inactive libraries from the lab machines en masse. This is not a problem unless disk space is limited and these folders are allowed to accumulate.

STUDENT HANDOUT

WHY USE SAS AND ENTERPRISE GUIDE?

SAS is a very rich system of procedures that is used extensively at research universities, government agencies, and companies for data analysis and reporting. Many of our graduating students report that they are asked if they know SAS when they interview for jobs. According to the SAS web site, www.sas.com, "Ninety-eight percent of Fortune 100 and ninety percent of Fortune 500 companies are SAS customers." Also, "SAS serves more than 35,000 business, government and university sites in 110 countries." Clearly, experience with SAS is an asset to students who intend to pursue advanced degrees, perform research, or seek high paying jobs in information technology.

SAS Enterprise Guide (EG) is a Windows thin client. It provides a point and click interface to many parts of the SAS System and access to all SAS features using traditional programming methods. Students in this class will use EG to perform their assignments. EG will be used because it allows students to concentrate on the statistical concepts instead of the SAS code. Students interested in learning SAS syntax can view the generated SAS code in the code window. Usually the generated code contains optional statements in addition to the required statements. This makes the code interesting to more advanced SAS users, but probably more overwhelming to new SAS users.

EG provides very thorough online help and a tutorial. Students are encouraged to use them to answer their questions. To learn more select

Help → Enterprise Guide Help or Help → Getting Started Tutorial.

PROJECTS

EG uses a project-based interface. You can have only one project open at a time. For class use, we recommend that you make each homework assignment a separate project and name them something like HW1, HW2, etc. The project window (top left panel) displays the active project and its associated data, code, notes, and results links. These links are referred to in the SAS documentation as nodes.

CREATING PROJECTS

To open a new project select: File → Open and select the **New** tab.

To open an existing project select: File → Open and select the **Existing** tab. Double click on the project file. The project will also open if you highlight it and select **OK**.

To save a new project select: File → Save project-name As

This method allows you to choose a name and storage location. It can also be used to change the name or location of your project. EG projects are binary files that have the extension .seg.

To save an existing project select: File → Save project-name

This method will save the project with its original name in its original location.

To close a project select: File → Close project-name

If you have created any temporary data sets, you will be asked to save them before you close the project.

Important Note: Be aware that your data is not stored in your project file. Your project will contain links to your data. It is important to know where your data is stored so you can back it up or manage it as needed. To learn more about data storage, see the **SAS Data Libraries** section of this paper.

PROJECT NODES

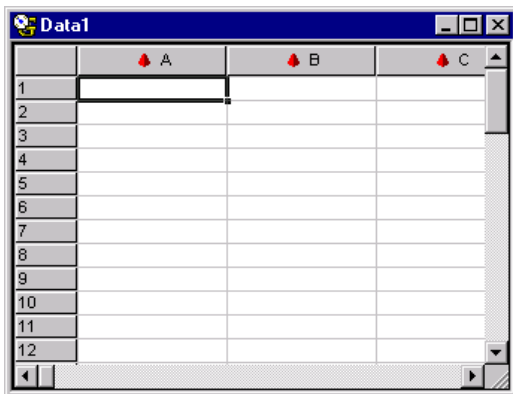
Nodes will be added to your project tree as you work. There will be nodes for data tables, query data views, analysis tasks, SAS code, notes, and results. You can open these nodes by double clicking on them. You can drag and drop these nodes to rearrange them. You can delete nodes you do not want.

PROJECT DATA

The first step in working with an EG project is to create a link to your data. To do this select **Insert → Data**, or **Tools → Import Data**. You

can also drag and drop data from Windows Explorer. EG can use any SAS data set that has one of the following extensions: sd2, sd7, or sd7bdat. One advantage of EG is you can access various types of data files, i.e. Excel and Oracle, from any platform where SAS runs. The necessary conversions are transparent to the user. However, it is essential that you check your data carefully. Import problems have been observed. These problems occur when the data does not match the assumptions that SAS makes in order to import data transparently. In the vast majority of cases the assumptions are correct, but they may not be correct for your data. Class data sets will be fine, but if you use other data sets it is up to you to verify the data is imported correctly.

Within your project, data will appear in a data table that resembles a spreadsheet. Variable type, character or numeric, is indicated by the symbol beside the variable name. A red pyramid indicates character data and a blue ball indicates numeric data.



	A	B	C
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Entering data into the data table: Data can be entered directly into the data table. To save this data, select **File → Save** or **File → Save As**. Choose a folder in the **Save in** drop down list and enter a filename in the **File name** box. Data sets are stored as SAS Data Files (*.sas7bdat, V7 Long Names) unless a different file type is selected in the **Save as type** drop down list. If you forget to save this data, you will be asked to save it when you close the data set or project.

Adding an existing SAS data set to your project: **Insert → Data** and click the **Existing** tab. Select a folder from the **Look in** drop-down list. Select the data set and click **OK**, or double-click the filename. To drag and drop a SAS data set into your project, click and hold the left mouse button, drag the file from an Explorer window into the Project window and release the left mouse button.

Adding an Excel spreadsheet to your project:

To create a SAS data set from an Excel spreadsheet select **Tools → Import Data** and double click on the Excel icon. Select a folder in the **Look in** drop down list and double click on the filename. If there are multiple sheets in the file, you will be able to choose the sheet you want. Follow the instructions provided for the rest of the steps. You will be able to select variables, rename variables, and add labels and formats. You will need to choose a data set name and storage location.

Excel spreadsheets can also be dropped into the project window. Using this method, a link to the spreadsheet will be added to your project tree instead of a SAS data set. You will not be able to edit this data.

Adding data using SAS code: A data set node will be added to your project tree for data sets created using SAS code. If you use a one level name, the data set will be stored in the WORK data library until you end your project. When you close your project you will be asked if you want to store the data set permanently. It is not required because the data set can be recreated by rerunning the SAS code. However, if you do not save the data set, the associated tasks will not be saved either.

Some instructors provide SAS programs on their Web sites that include both data and code. SAS Institute provides sample programs in the OnlineDoc and on their Web site. These programs can be copied and pasted into the code window. More information is provided in the section **Using the SAS Code Window**.

Note: Other methods for adding data to a project are available but will not be covered because they are more complex. Data problems should be rare if students are provided SAS data sets and they store them in either their SASUSER or EGTASK data library.

Editing data tables: Data tables are opened in Read-Only mode. To change the protection level to Update select: **Data → Protected**. You will receive a message that asks if you want to change this data to Update mode and indicates that changes made will be applied directly to the data. You will also receive this prompt if you try to edit a data set that is in Read-Only mode.

Some changes made in the data grid can be undone using **Edit → Undo**. However, not all changes to a data grid can be undone. We recommend that you make a backup copy of your data before editing it.

Class data sets are protected so that they cannot be modified. If you need to modify a class data set, you must create a copy of the data set and modify your copy. You can use Windows Explorer to place a copy of the data set into your own file space, but do not change the data set name. To save a copy of a data set from within EG, select **Insert → Data** and highlight the desired data set. Right click on the data set node and select **Save data-set-name As**. Choose a folder in the **Save in** drop down list and click **Save**. Now you need to change the data node so it points to your data instead of the class data. To do this right click on the data node and select **Properties**. Click **Change**, choose a folder in the **Look in** drop down list, double click on the filename, and click **OK**. We were not able to use this method if the data set was stored in the WORK data library.

Exporting Project Data: SAS data sets can be exported and saved in many file formats. To export your data, highlight the data node, right click the mouse button, and select **Save data-set-name As/Export**. Choose a folder in the **Save in** drop down list, choose a file type from the **Save as type** drop down list, and click **Save**.

CREATING A COMPUTED COLUMN

A new column or variable can be created either directly in the data grid or using the Query Builder. Using the data grid method, the new variable is added to the existing SAS data set, but no record of how it was created is stored unless you copy the expression into the **Label** box. With the Query Builder method, a query data view is created that contains all the original data and the new column. The funnel icon in the project tree indicates a query data view. It may be confusing to have data nodes for both the original data set and the query data view. The advantage of the Query Builder is that the SQL code used to create the new column is stored with the query data view. The SQL code can be viewed by checking the **Preview the query code** box in the Query Builder window or by dragging and dropping the query data view into a code window.

Creating a computed column in the Data Grid:

Open the data file and place the cursor on the column heading to the left of where you want to add the new column. Select **Insert → Columns** and click the **General** tab. Type the new column name in the **Name** box and the expression in the **Expression** box. We recommend that you copy the expression into the **Label** box so it is stored with the data. Click **OK**.

For example, if Y is an existing column, you could create a new column or variable, LY, using the expression LOG(Y).

Creating a computed column in the Query Builder: Select **Tools → Query → Create from Active Data**, click the **Select and Sort** tab, and click **New**. In the **General** tab, type the name of the new column in the **Alias** box. Now, click the **Expression** tab. You can use the Expression Builder to create the expression or type the expression directly into the Expression box. Click **OK** to add the new column. The new computed column appears in the list of computed columns on the left side of the Query Builder.

FILTERING DATA USING THE QUERY BUILDER

When performing statistical analysis, you may need to delete observations or subset the data. This can be done using the Query Builder. The Query Builder generates SQL code to perform these tasks. Check boxes at the bottom of the Query Builder window allow you to view the query result, code, and log. You can also open a code window and drag and drop the query data view from the project tree into the code window. The SQL code will appear in the code window.

Filtering Data: To filter data, highlight the data node and select **Data → Filter**. Drag a variable from the left panel to the **Filter Data** window. The **Edit Filter Condition** window will appear to help you build an expression. The data set created will be a query data view. A data view does not actually contain data; it contains instructions on how to get data from the original data set. The funnel icon is used in the project tree to indicate a query data view.

Saving the query data view as a SAS data set:

To save the query data view as a SAS data set, check the **Save as Data** box at the top of the Query Builder window. If you do this, two nodes will be added to your project tree; one for the SAS data set and one for the query data view. You will be able to tell them apart by their icon. The output SAS data set will have the name you entered in the **Name** box and it will be stored in the SASUSER library by default. To change the default storage location you can define the EGTASK libref. For instructions, see the **SAS Data Library** section of this paper.

JOINING DATA USING THE QUERY BUILDER

Up to sixteen tables can be joined using the Query Builder. To open the Query Builder window, select **Tools → Query → Create from Active Data**. Select the **Tables** tab and add data tables using the **Add Data** button at the bottom. EG will draw join lines between matching variables in the data sets, indicating how it will match the data sets. The diamond on the line will contain the join operator, usually an equal sign

(=). To modify the join, right click on the operator in the diamond and select **Modify Join**.

The default join is an inner join and the resulting data set will contain matching rows only. Other join types can be selected in the **Modify Join** box, which is opened by right clicking on the operator in the diamond. If you are joining two tables, your output choices are matching rows only; all rows from data1; all rows from data2; all rows from data1 and data2. These are referred to as inner, left outer, right outer, and full outer joins respectively. This is explained in detail in the EG online help.

CONCATENATING SAS DATA SETS

To concatenate SAS data sets, select **Data→Append Data**. If you need to rename variables before you append the data sets, this can be done in the data grid.

ADDING NOTES TO YOUR PROJECT

Documentation and comments can be added to your project using notes.

To insert a note in your project select: Insert → Note.

PROJECT TASKS

EG contains about 60 stored tasks. These tasks provide a point and click interface to many SAS procedures. Tasks can be accessed using the **Tools, Data, Analysis, and Graph** menus. They are also available in the Task window. In this window the task list can be viewed and sorted using the **Task by Category** tab or **Tasks by Name** tab.

Tasks always use the current data set. The current data set is the data set highlighted in your project tree. Many tasks provide an easy way to select analysis and classification variables; procedure options; additional graphic output; and output data sets. Tasks generate SAS code and produce results. Code and results nodes are added to the project tree beneath the task. You may need to expand the project tree in order to see all the added nodes.

THE CODE WINDOW

The EG code window is color-coded and provides syntax checking. To submit code, use one of these methods. To use the pull down menu, select **Code → Run on Local**. To use the toolbar, click the **Run on Local** icon. To use the mouse, click the right mouse button in the code window and select **Run on Local**.

When you run your analysis from a code window, a code node and all the usual nodes will be added to your project tree.

Viewing EG generated SAS code: When you perform a task, EG generates the SAS code. This code can be viewed by opening the code node. At the beginning of the code, EG reports the task; the date and time of execution; the server (such as local PC); and the data source. Reviewing the code is a good way to learn SAS syntax. However, for a beginning user, the SAS code may be overwhelming. EG code contains many options and statements that are not required, especially when graphs are requested. Although this makes the code more complicated it is a good learning tool.

Rerunning EG generated code: The code for any task can be modified and rerun in a code window. To do this, open the code node you wish to repeat. When you attempt to make changes to the code you will receive a prompt that states: "This code has read-only attributes. Do you want to add this as a modifiable code window?" If you choose yes, a new modifiable code window will be opened and added to your project tree. This window will contain the code you want to modify.

To rerun code with a different data set, replace the data set name that follows the DATA= option on the procedure statement. This can be done globally using **Edit → Replace**. If the program creates a SAS data set, you may want to replace this data set name as well. Many procedures use the OUT= option to name output data sets. If you do not replace these names, the new data sets will replace the existing data sets when you rerun the code.

To identify a SAS data set, you must know its full name. One-way to determine the data set name is to highlight the data node in the Project window, click the right mouse button and select **Properties**. The **File name** box will display the full data set name. A second method is to drag and drop the data set node from the Project window to an open code window. A comment that reports the data set name will appear, along with a LIBNAME statement, if one was automatically created when the data was generated. For more information on LIBNAME statements see the **SAS Data Libraries** section or look in **Help**.

Entering and running code: You can enter and run SAS code in the code window. In some cases, this may be easier than using the menus. In some cases, it may be necessary because there is no programmed task in EG for what you need to do.

Copying SAS code from other sources, i.e. the Web: SAS programs that contain data and code can be copied from the other sources,

pasted into a code window, and submitted. This method allows you to run programs provided by your instructor, OnlineDoc, or the SAS Web site. http://www.sas.com/service/techsup/sample/sample_library.html.

To do this, select **Insert → Code** and click the **New** tab. Enter a name in the **Name** box and click **OK**. Copy and paste the SAS program and data into the code window. Right click in the code window and select **Run on Local**.

Saving your SAS code: Your code will automatically be saved in your project tree, but you can save it outside your project by selecting: **File → Save code As**.

Saving all project code: **File → Export All Code**. Exporting all the code is not the same as saving the project. The code cannot re-create the project. Before saving all the code, you may want to delete unwanted nodes. To delete task nodes, right click on the task in the project window and select **Delete**.

RESULTS

Selecting your results style: By default, results are stored in HTML format using the EG Default Style. You can select another style from the drop-down list on the toolbar. Results will print much better if you select **Printer** instead of **EG Default**. You do not have to rerun the task, because the HTML file is dynamically updated when you select a new style.

Selecting your results file type: In addition to HTML, results can be generated as text, PDF, or RTF files. RTF files copy very nicely into WORD documents. PDF files have a professional appearance and you can easily print selected pages from Adobe Acrobat. Multiple file types can be produced at the same time. To choose file types and other results options, select **Tools → Options** and click the **Results** tab.

Using EG Document Builder: You can use the EG Document Builder to combine the results of multiple tasks into a single document, but you cannot add text and ActiveX graphs may not be displayed. This limits the usefulness of the Document Builder. To build a document, select **Tools → Build Document**.

Copying your results into a Word document: You can cut and paste your results into a Word document. We recommend you choose the RTF file type if you plan to do this. Tables and graphs can be edited in Word.

Printing complete task results: To generate a complete listing of the task results, open the

results by double clicking on the results node and select **File → Print**.

Printing a single piece of task output: Fully expand the results node in the Project window, open the desired piece of output and select **File → Print**. Notice the entire contents of the window are printed.

Emailing your Results:

To email your project code or data, highlight the node, right click, and select **Send To**.

SAS DATA LIBRARIES

Folders that contain SAS files are referred to as SAS data libraries. Library references, or librefs, are assigned using LIBNAME statements.

SASUSER is a special libref that is automatically defined by SAS and output data sets are stored in this library by default.

WORK is a special libref that is automatically defined by SAS and temporary files are stored in this library.

EGTASK is a special libref, which you can define to identify the library where you prefer EG to store output data sets. If EGTASK is defined, task output data sets will be stored in the defined EGTASK library rather than in the SASUSER library.

To view a list of data sets within assigned libraries, select **File → Open**. Select, **Local → Libraries** in the drop down list. Data sets can be renamed, copied, and deleted from within this window. If other servers are defined they can be found by selecting **File → Open → Servers**.

To determine what librefs are defined and what library they refer to, submit one of the following statements:

```
LIBNAME _all_ list;  
LIBNAME sasuser list;  
LIBNAME egtask list;  
LIBNAME work list;
```

To define EGTASK, submit a LIBNAME statement like:
LIBNAME EGTASK "c:\st508";
Replace **c:\st508** with the name of an existing folder where you want output data sets stored.

Note: SAS data sets stored in the SASUSER or EGTASK libraries are considered permanent because they will not be erased by SAS at the end of your EG session. Data sets stored in the WORK library are considered temporary because they will be erased at the end of your EG session.

To facilitate transparent data access, EG defines temporary librefs when you insert data into your project from a library that has not been previously identified. These librefs begin with the prefix "EC"; for example, ECLIB000. You will notice these librefs when you look at the SAS code, or if you drag and drop a data node into a code window.

EXERCISE

Before beginning, please change the following EG settings. All these settings will remain in effect between EG sessions, except defining EGTASK. EGTASK has to be defined at the beginning of each EG session. EG may not use EGTASK if it is not defined at the beginning of the session.

Turn the agent sound off: By default, an agent, called Genie, will appear to assist you. The agent provides helpful hints, but the sounds can be distracting. To turn the agent sound off, right click on the agent, and select **Advanced Character Options**. Clear the **Play spoken audio** and **Play character sound effects** check boxes. You may also want to set the **Speaking speed** to its fastest setting. This will improve the typing speed.

Change the style from EG Default to Printer: Select **Tools** → **Options** and click the **Results** tab. In the Style box, replace **EG Default** with **Printer** and click **OK**.

Select RTF output as well as HTML: This is only necessary if you want to paste your results into a WORD document. To choose the RTF file type or set other results options, select **Tools** → **Options** and click the **Results** tab. Make your selections and click **OK**.

Define EGTASK: We recommend you define the EGTASK library at the beginning of each EG session so output data sets will be stored in your SAS data library. For instructions, see the **SAS Data Libraries** section of this paper.

EXERCISE 1

Objective: Perform Linear Regression using Crime Data

Data Description: This data set contains several variables from the statistical abstracts of the United States for the 50 states and District of Columbia (DC).

Variables: Murder Rate, Violent Crime Rate, Metropolitan Residents %, White %, High School Graduates %, Poverty Rate % and Single Parent %.

Data Source: Statistical Methods for the Social Sciences by Alan Agresti and Barbara Finlay, chapter 9.

Add the crime data set to your project and create the following:

1. Scatter plot of Murder versus Poverty (Task: Graph → Line)
2. Boxplot for Murder (Task: Analysis → Descriptive → Distribution Analysis)
3. Regression of Murder versus Poverty. Include the following plots: Observed vs. Predicted, Observed vs. Poverty with Prediction Line, and Residual Plot (Task: Analysis → Regression → Linear);
4. Omit the DC observation from the data set and repeat the Linear Regression (Task: Data → Filter then repeat step #3)
5. Prepare a document of your results; be sure to include your interpretation.

CONCLUSION

EG greatly simplifies data access. Students should be able to easily access a wide variety of data types from various platforms with few if any problems. It is important to realize that no software can deal with every possible data file, so it is up the student to verify that the data is imported correctly.

EG greatly simplifies many statistical tasks. Students should be able to generate statistical output without having to learn SAS syntax. This will allow the students to concentrate on their data and learning statistics.

EG greatly simplifies the creation of SAS graphs. SAS/GRAPH code is difficult, so only dedicated students attempt to learn SAS/GRAPH. However, the simplicity with which graphs can be created in EG will lead to their increased use. The use of graphs will enhance the data analysis and interpretation of statistics.

EG should make simple data management tasks easy to perform. However, students with little data analysis experience will probably need instructions on joining data, filtering data, and creating new variables. Learning the SAS code for these tasks would probably be harder than learning the EG methods.

EG provides students an interactive and fairly convenient way to learn SAS programming. The generated code can be viewed in the code window. The generated code usually contains many options and statements that are not required. These are interesting to a programmer who is comfortable with SAS code, but for a beginning SAS user, they probably make SAS seem more complex than it has to be. Students can also copy and run sample programs that are available from many sources. EG will provide very limited exposure to the data step.

Understanding the data step is definitely a core concept in SAS programming. On the other hand, EG will provide exposure to SQL.

Traditional SAS programmers often rely on their programs and log files for documentation. EG provides logs for the tasks, but not for data sets created from Query Builder or modified in the data grid. It is true that the SQL code is saved with the query data view, but no code or log is saved with the SAS data set.

EG will probably be useful to graduate students and researchers at NCSU. However, because not all analyses are available as tasks, they may also need to write code. Students often run large simulations and batch jobs. We are not sure how these will work in our environment using EG. It will also be interesting to see how EG changes the consulting we do with these clients.

EG provides a rich selection of results file types; such as HTML, txt, PDF and RTF. This variety of file types provides many options for reporting results. We do not believe that the Document Builder is full featured enough for our needs, but the smooth incorporation of RTF files into Word documents will certainly be used extensively. Some campus members will use the HTML file formats for reporting on the Web. The email feature will be useful, especially to students who need to share their results with professors or committee members.

We intend to introduce EG in a statistics class before the meetings. We plan to evaluate whether EG can be used to provide beginning statistics students with a relatively painless introduction to SAS. We hope that EG will do this and that we will see the use of EG and SAS increase in these classes in the future.

REFERENCES

Rigsbee, Carol and Hemedinger, Chris (2001)
"Delivering Information to the People Who Need to Know". Proceedings of the 26th SAS Users Group International Conference, Paper 145-26.

TRADEMARKS

SAS is a registered trademark or a trademark of SAS Institute Inc. in the USA and other countries.
® Indicates USA registration.

Other brand or product names are registered trademarks or trademarks of their respective companies.

ACKNOWLEDGMENTS

We sincerely appreciate Terry Bryon, Joe Wells, and Laura Grady for helping us get the software, installing it on the distributed platform, and

answering questions related to its use in this environment. Thanks to Dr. Jackie Dietz for the information on how she uses SAS in her introductory statistics classes. Thanks to Bill Sawyer of SAS Technical Support for his patient and thorough responses to our questions.

CONTACT INFORMATION

Sandy Donaghy
Email: sandy_donaghy@ncsu.edu

Joy Smith
Email: joy_smith@ncsu.edu