



SESUG Speaker Sharing Program

Presentation Titles, Abstracts and Author Bios

Program Goals:

- Speaker sharing is one of the many ways SESUG and SAS support local SAS user groups.
- SESUG and SAS will co-sponsor and provide SAS user groups in the SouthEast a yearly presentation selected from our list of topics and delivered by our SESUG EC featured speakers. Speaker Travel Costs are fully paid by SESUG and SAS.

Program Specifics:

- Local and in-house SAS user groups in the SESUG region (Alabama, Florida, Georgia, Kentucky, Maryland, Mississippi, North Carolina, South Carolina, Tennessee, Virginia, West Virginia, Washington DC, Puerto Rico) are eligible to participate.
- Sponsorship is limited to one speaker each calendar year. Speakers **do** facilitate requests to provide more than one presentation at a User Group Meeting.
- SESUG fulfills local SAS user group requests for speakers based upon speaker availability and reasonable travel costs.
- In support of the SESUG Speaker Sharing Program, SAS provides matching funds to assist SESUG in maximizing the effectiveness and coverage of the Speaker Sharing Program.

Requesting a Speaker:

- Please plan as far in advance as possible so that SESUG can accommodate your request.
- After reviewing the available topics, speakers, and speaker bios **BELOW**, select your “top three choices” of speakers, and email your request to [Marje Fecht](#). Please include:
 - Desired **date / time / location** of meeting
 - User Group name and contact information
 - **Three** preferred speakers, and the presentation titles of interest.
- SESUG will review your request and, if fulfilled, SESUG will provide you with confirmation of:
 - Speaker / title(s)
 - Dates
 - Travel arrangements.
- NOTE: Requests are fulfilled based on speaker availability and reasonable travel expenses.



SESUG Speaker Sharing Program

Alphabetic List of Presentation Titles

- A Cup of Coffee and Proc FCMP: I Cannot Function Without Them Peter Eberhardt
- A Hands-On Introduction to SAS® DATA Step Programming Debbie Buck
- A “SAS® Programmer’s” Guide to the SAS® Enterprise Guide® Marje Fecht
- Analysis of Longitudinal Data: Comparison between PROC GLM and PROC MIXED Maribeth Johnson
- Are Your SAS® Programs Running You? Marje Fecht
- Building the Better Macro: Best Practices for the Design of Reliable, Effective Tools Frank Dilorio
- ColdFusion and SAS: No Pain Meds Required Carol Martell
- DATALINES, Sequential Files, CSV, HTML and More – Using INFILE and INPUT Statements to Introduce External Data into the SAS® System Andrew T. Kuligowski
- Data Set Options: Beyond DROP, KEEP, RENAME, and WHERE Ed Heaton
- Demystifying the SAS® Macro Facility - by Example Marje Fecht
- Dictionary Tables and Views: Essential Tools for Serious Applications Frank Dilorio
- Efficiencies with Large Datasets Peter Eberhardt
- Elementary Statistics Using Base SAS Debbie Buck
- Explaining Unexpected Log Messages and Output Results from DATA Step Code Debbie Buck
- Evaluating Sample Code for an Interview Stephanie Thompson
- Four Thousand Reports Three Ways Stephanie Thompson
- Fun with Functions Marje Fecht
- Getting Data into SAS®: INFILE and INPUT (*tutorial format*) Andrew T. Kuligowski
- Getting Your Data into SAS® Stephanie Thompson
- Graphing the Easy Way with SAS® Enterprise Guide® (or How to Look Good With Less Effort) Stephanie Thompson
- Here’s the Data, Here’s the Report I Want - How Do I Get There? Debbie Buck



SESUG Speaker Sharing Program

- How to Incorporate Old SAS® Data into a New DATA Step, or “What is S-M-U?” Andrew T. Kuligowski
- Individual Growth Analysis Using PROC MIXED Maribeth Johnson
- In Search of the LOST CARD Andrew T. Kuligowski
- Labels, Labels, and More Labels (note: 30 minutes) Stephanie Thompson
- Looking Beneath the Surface of Sorting Andrew T. Kuligowski
- Many to One Using a SAS® DATA Step and PROC MEANS Jennifer Waller
- More than Models: The Data Mining Team Stephanie Thompson
- Moving Forward with FORMATS David Maddox
- New to SAS® and New to Programming? What You Need to Do Before Typing Code Stephanie Thompson
- Notes from an Intersection: Google Earth @ SAS Carol Martell
- PROC REPORT...Your How-to Guide for Producing Customized Summary Tables Debbie Buck
- Programming for Failure: When Programs Faw Down and Go Boom Gary Schlegelmilch
- Pruning the SASLOG – Digging into the Roots of NOTEs, WARNINGs, and ERRORs Andrew T. Kuligowski
- Rules for Tools - The SAS Utility Primer Frank Dilorio
- SAS as a Utility Language: Using SAS for Solving Little Problems Gary Schlegelmilch
- SAS® System Options Are Your Friends Ed Heaton
- SAS® XML Mapper to the Rescue Carol Martell
- Set, Match, Merge ... Don't You Love SAS® Peter Eberhardt
- Shuffling, Shifting, and Structuring Using the Program Editor Gary Schlegelmilch
- So, Your Data are in Excel! Ed Heaton
- Summarizing Data with Base SAS® PROCs Debbie Buck
- Talking Past Each Other? How to Communicate with Medical Writers When Preparing Clinical Research Manuscripts for Journal Submission Stephanie Thompson



SESUG Speaker Sharing Program

- [The Design and Use of Metadata: Part Fine Art, Part Black Art](#) Frank Dilorio
- [The Effect of Missing Data on Repeated Measures Models](#) Maribeth Johnson
- [The SAS® Community: How to Be a Part of It!](#) Gary Schlegelmilch
- [The SAS Data Step: Where Your Input Matters](#) Peter Eberhardt
- [The SAS® Hash Object: It's Time To .find\(\) Your Way Around](#) Peter Eberhardt
- [The SAS Debugging Primer](#) Frank Dilorio
- [Things Dr Johnson Did Not Tell Me: An Introduction to SAS® Dictionary Tables](#) Peter Eberhardt
- [THINK Before You Type... Best Practices Learned the Hard Way](#) Marje Fecht
- [Using SAS® to Parse External Data](#) Andrew T. Kuligowski
- [Where's The Match? Matching Study and Comparison Subjects after Data Collection](#) Jennifer Waller

Author Bios

Debbie Buck

Debbie Buck has been programming in SAS for over 25 years and has been an independent consultant in Statistical Consulting and SAS Programming for over 20 years, primarily in the Pharmaceutical and Medical Device fields. She has also been a Contract Instructor for the SAS Institute.

She has written and presented papers at SAS Global Forum/SUGI, regional, and local SAS user group meetings. Debbie chaired the SCSUG 1997 Conference and co-chaired SSU 2001, the Joint Conference of the SouthEast and South Central SAS Users Groups and is active on the SESUG and SCSUG Executive Councils.

Debbie is the SAS Global Forum 2011 Conference Chair, to be held at Caesars Palace in Las Vegas, NV in April 2011.

Frank Dilorio

A SAS programmer since 1975, Frank Dilorio is President of CodeCrafters, Inc. and the author of "SAS Applications Programming: A Gentle Introduction" and "Quick Start to Data Analysis with SAS." He is past President of the SouthEast SAS Users Group, and co-chaired its 1994 and 1996 conferences. He is also active in several local SAS Users Groups and was a co-founder of the Research Triangle CDISC Users Group. A native New Yorker, he lived in North Carolina from 1974 until 2008, when he returned to his Yankee roots, relocating to Philadelphia.



SESUG Speaker Sharing Program

Peter Eberhardt

Peter is a long time SAS consultant, and his company, Fernwood Consulting Group Inc., is a SAS Alliance partner. Peter is a regular speaker about, and a strong advocate of the SAS Business Intelligence architecture and the new tools it brings to the SAS community.

Peter co-chaired SESUG 2008 in St. Pete Beach and serves on the Executive Council of SouthEast SAS Users Group.

Marje Fecht

Marje Fecht is a Senior Partner with Prowerk Consulting, and has been a SAS software user since 1979.

Her initial use of SAS focused on statistical applications and she has been teaching and using computer software and SAS for over 30 years. She developed and taught a broad curriculum of courses for SAS Institute including Applications Development, Data Management and Access, Graphics, Reporting, Macros, and Statistics.

Most recently, her consulting work has focused on developing efficient systems for reporting, analysis, and Business Intelligence at major financial and retail organizations. In addition to her focus on developing hands-free production coding techniques, she enjoys sharing tips for graphical representation of data as well as applications development.

Marje is a frequent presenter at Users Groups in the US and Canada, and she is an active participant in conference planning for SESUG and SAS Global Forum.

Her favorite SAS pastime is *reducing spaghetti code to a single noodle*.

Ed Heaton

Ed Heaton has been programming in SAS® since 1988 - professionally since 1993. He has presented papers at SAS Global Forum (SUGI) and in-house, local, and regional SAS Users Groups since 2000. Ed was the academic co-chair of the 2007 conference of the Southeast SAS Users Group and has co-chaired various academic sections for other SESUG conferences.

As a project manager and senior systems analyst for Data & Analytic Solutions, Inc., Ed researches and detects fraud in Medicaid billing for the U.S. Department of Health and Human Services.

Maribeth Johnson

Maribeth Johnson has been an Assistant Professor in the newly formed Department of Biostatistics at the Medical College of Georgia since 2005. For 11 years prior to that she was a consulting research statistician in the Office of Biostatistics at the same institution. She previously worked in the University of Georgia's Department of Animal and Dairy Science for 11 year as a statistical programmer and cow hand. She has MS degrees in Animal Breeding from VA Tech and in Statistics from UGA. She has been using SAS since 1980. Maribeth serves on the Executive Council of the SouthEast SAS Users Group where she is currently the President.



SESUG Speaker Sharing Program

Andrew T. Kuligowski

Andrew T. Kuligowski has been a SAS user for well over 25 years, and is currently a Sr. Software Developer for FCCI Insurance Group in Sarasota, Florida. He has augmented his professional coding experiences in the insurance, media, and petrochemical fields by speaking at various SAS conferences and user group meetings. Andrew was conference co-chair for SESUG'97 in Jacksonville, Florida and Tennessee SAS Users Day in Knoxville, Tennessee, and is slated to chair SAS Global Forum 2012 in Orlando, Florida. In his spare time, Andrew can also be found volunteering at the Florida Aquarium in Tampa.

Carol Martell

Carol is a Senior Applications Specialist for the Highway Safety Research Center (HSRC) at UNC Chapel Hill. She has over 30 years of SAS experience, and has been with HSRC since 1989. Over the course of time, she ushered HSRC onto the internet and as head of HSRC computer services, she managed an expansion into what is now a small computer center. This experience enhanced her integration skills, which include dynamic web application and web data management. She maintains HSRC source data for NC traffic records, querying and building population-based analysis files for contracts, many of which involve the creation of dynamic websites querying those data. She has presented at SUGI, SAS Global Forum, SSU, SESUG and RTSUG, is webmaster for RTSUG and is Operations Chair for SESUG 2009.

David Maddox

David Maddox is a Business Banking Reporting Analyst for Regions Financial Corporation and has been a SAS user for approximately twenty years. David has been active in SAS user groups for over ten years, including the Birmingham Users Group for SAS (BUGS) and the SouthEast SAS Users Group (SESUG). In 2002, he served as conference co-chair for SESUG 2002 in Savannah, GA

Gary Schlegelmilch

Gary Schlegelmilch has been enthusiastically working in SAS for 12 years. He has worked in SAS primarily using Base SAS, SAS/CONNECT, and Macros on UNIX and VMS. His programming career spans four decades, and he has used and informally taught several computer languages. Gary enjoys presenting papers at local, regional, and international SAS conferences, and co-chaired the 2003 SouthEast SAS Users Group conference in St. Pete Beach, Florida. He is an active member of the SESUG Executive Council, and is co-chairing the SESUG 2009 Simple But Clever section.

Stephanie Thompson

Stephanie Thompson has over fifteen years experience applying statistical and modeling techniques to business problems in various manufacturing, retail, and academic environments using SAS and other programming languages. She has a strong understanding of data structures, analytical tools, and operating environments. She holds a B.S. in Industrial Engineering from Rochester Institute of Technology and an M.B.A. from St. Bonaventure University.

Stephanie is a SAS Certified Base Programmer for SAS9 and has presented at SESUG, SUGI / SAS Global Forum, NESUG, and local users group events. She is a member of the SESUG Executive Council, has co-chaired various academic sections at SESUG, and is the Academic Program Chair for SESUG 2010.



SESUG Speaker Sharing Program

Jennifer Waller

Jennifer Waller obtained a PhD in Biostatistics from the School of Public Health at the University of South Carolina in 1994. She is an Associate Professor and Director of Graduate Programs in the Department of Biostatistics at the Medical College of Georgia where she has worked since 1997 teaching, consulting and doing collaborative research. She has used SAS since 1989 and served as conference co-chair for SESUG 2008 in St. Pete Beach, Florida.

Presentation Abstracts (in alphabetic order by presentation title)

A Cup of Coffee and Proc FCMP: I Cannot Function Without Them

Peter Eberhardt

How many times have you tried to simplify your code with LINK/RETURN statements?
How much grief have you put yourself through trying to create macro functions to encapsulate business logic? How many times have you uttered "If only I could call this DATA Step as a function"?

If any of these statements describe you, then the new features of PROC FCMP are for you. If none of these statements describe you, then you really need the new features of PROC FCMP. This paper will get you started with everything you need to write, test, and distribute your own "data step" functions with the new (v9.2) PROC FCMP. This paper is intended for beginner to intermediate programmers, although anyone wanting to learn about PROC FCMP can benefit.

A Hands-On Introduction to SAS® DATA Step Programming

Debbie Buck

The SAS DATA step is one of the most powerful and versatile software tools available for handling and manipulating data files, especially large data sets. It is also sometimes somewhat confusing to SAS programmers as to what the different statements do, what the correct syntax is for these statements, and how they can help you achieve your goals.

In this workshop we will examine how to get data into a SAS data set, how to create new variables or modify existing variables, how to conditionally handle variables or observations, how to handle a group of variables in the same way, and how to control which observations are written to the SAS data set.



SESUG Speaker Sharing Program

A “SAS® Programmer’s” Guide to the SAS® Enterprise Guide®

Marje Fecht and Rupinder Dhillon

Have you been programming in SAS for a while and just aren’t sure how Enterprise Guide can help you? It isn’t *just a pretty face!*

This presentation will demonstrate how SAS programmers can use SAS Enterprise Guide as their primary interface to the SAS system while maintaining the flexibility of writing their own customized code. We will look at

- navigating the views and menus
- using SAS Enterprise Guide to access your existing programs and enhance your processing
- leveraging the more complex built-in capabilities available in SAS Enterprise Guide to further enhance the information you deliver
- using Report Builder (new to Version 4.2 of SAS Enterprise Guide)
- adding Project Parameters and dynamic parameters, to generalize the usability of your programs and processes.

We will also share **best practices** for working in a server environment.

Version 4.2 of SAS Enterprise Guide will be used.

This presentation is appropriate for SAS users who understand the basics of SAS programming and want to learn how to use SAS Enterprise Guide.

Analysis of Longitudinal Data: Comparison between PROC GLM and PROC MIXED

Maribeth Johnson

Longitudinal data refers to datasets with multiple measurements of a response variable on the same experimental unit made over a period of time. These types of data require special attention because they involve correlated data. The relationships between repeated measurements are important in assessing reliability and tracking of those measurements. The proper variance-covariance structure in the analysis model is essential to the understanding and interpretation of those relationships. The assumption of compound symmetry necessary for correctly using the intraclass correlation as a measure of tracking can be tested against other variance structures using PROC MIXED. This paper compares the variance, covariance and correlation estimates obtained from the GLM and MIXED procedures of SAS/STAT® on two sets of data, one of which has missing data.



SESUG Speaker Sharing Program

Are Your SAS® Programs Running You?

Marje Fecht (Co-author: Larry Stewart)

Most programs are written on a tight schedule, using the most accessible knowledge of the programmer. Often the programmer mistakenly assumes that the program will never be used again; five years later the spaghetti code is still in use. While the tasks are accomplished and the results are accurate, the program may not be as efficient as possible and subsequent submissions may require tedious and time-consuming input and modifications.

This presentation looks at typical SAS Code, and then suggests changes to improve the efficiency and maintenance of the programs. If you are a programmer who has inherited code that was written "*many SAS versions ago*", you will benefit from examples of "*updating your code to the current decade*".

This tutorial focuses on maintenance - free, efficient coding techniques so that you can spend your work time being more productive!

Topics include:

- macro coding techniques
- efficient programming tips
- code reduction tricks
- maintenance-free programming suggestions.

Building the Better Macro: Best Practices for the Design of Reliable, Effective Tools

Frank Dilorio

The SAS® macro language has power and flexibility. When badly implemented, however, it demonstrates a chaos-inducing capacity unrivalled by other components of the SAS System. It can generate or supplement code for practically any type of SAS application, and is an essential part of the serious programmer's tool box. Collections of macro applications and utilities can prove invaluable to an organization wanting to routinize work flow and quickly react to new programming challenges. But the language's flexibility is also one of its implementation hazards. The syntax, while sometimes rather baroque, is reasonably straightforward and imposes relatively few spacing, documentation, and similar requirements on the programmer. In the absence of many rules imposed by the language, the result is often awkward and ineffective coding. Some amount of self-imposed structure must be used during the program design process, particularly when writing systems of interconnected applications. This paper presents a collection of macro design guidelines and coding best practices. It is written primarily for programmers who create systems of macro-based applications and utilities, but will also be useful to programmers just starting to become familiar with the language.



SESUG Speaker Sharing Program

ColdFusion and SAS: No Pain Meds Required

Carol Martell

This paper describes a method of integrating dynamic SAS output into an Adobe ColdFusion® application. The SAS/IntrNet Application Dispatcher responds to a request from the ColdFusion server, and the results are displayed in a ColdFusion page having the .cfm extension. The advantages to using this approach include isolating the SAS programmer from the design aspects of Web development.

The technique requires that SAS/IntrNet Application Dispatcher be available with the accompanying SAS/IntrNet Application Server, and that the user have access to a ColdFusion Server. The paper does not address setting up or configuring either Application Dispatcher or ColdFusion Server. The intended audience is Web application developers with experience using Application Dispatcher and with the desire to stop using PUT statements for enhancing Web design.

DATALINES, Sequential Files, CSV, HTML and More – Using INFILE and INPUT Statements to Introduce External Data into the SAS® System

Andrew T. Kuligowski

The SAS® System has numerous capabilities to store, analyze, report, and present data. However, those features are useless unless that data is stored in, or can be accessed by, the SAS System. This presentation is designed to review the INFILE and INPUT statements. It has been set up as a series of examples, each building on the other, rather than a mere recitation of the options as documented in the manual. These examples will include various data sources, including DATALINES, sequential files, CSV files, and HTML files.

Data Set Options: Beyond DROP, KEEP, RENAME, and WHERE

Ed Heaton (co-author: Sarah Woodruff)

While the `DROP=`, `KEEP=`, `RENAME= ()`, and `WHERE= ()` data set options are extremely useful and well known, there are numerous other data set options that can greatly enhance computer efficiency and programmer productivity. However, finding them and the documentation can be daunting because not all options are a part of Base SAS® and their descriptions exist in various locations.

This paper will provide a quick reference to the documentation for data set options and then go on to discuss a handful of data set options that deserve special consideration. Some of these are Base SAS data set options and some are provided with SAS/ACCESS® software. Our discussion of the documentation will cover both that which is available through the Help menu in the SAS® Display Manager and what is available online via support.sas.com.



SESUG Speaker Sharing Program

Demystifying the SAS® Macro Facility - by Example

Marje Fecht (co-author: Harry Droogendyk)

The SAS macro facility enables you to apply a wealth of useful, uncomplicated, real-world solutions to enhance your coding pleasure, reduce coding effort, and minimize error. As your business applications inevitably become more complex, the SAS macro facility is indispensable to:

- reduce code repetition
- increase control over program execution
- minimize manual intervention
- create modular code.

Unfortunately, the SAS macro facility is often perceived as confusing, difficult to use, and only really comprehensible by a few long-time macro masochists. This presentation removes the mystery of macros and provides coding tips, macro solutions, and methodologies you can take away and implement immediately.

Dictionary Tables and Views: Essential Tools for Serious Applications

Frank DiIorio

Dictionary tables were introduced to the SAS System in during the mid-life of Version 6. Laden with information that is often difficult, and sometimes impossible, to get through other means, they still appear to be on the outside of many programmers' Bag of Tricks. This is both perplexing and unfortunate for as we will see in this paper, once their content and organization is understood, they are readily adapted for a range of applications that "are only limited by your imagination." Indeed, it is difficult to think of a robust, generalized system utility that would not benefit from use of this metadata.

This paper describes dictionary tables and their associated SASHELP library views. It:

- presents scenarios that show how they can be used
- gives high-level descriptions of some of the more important (a relative term, to be sure) tables
- identifies features of SQL and the macro language that are commonly used when writing programs that effectively use the tables
- shows examples of the tables' use, emphasizing the use of SQL and the macro language interface

The reader should come away from the discussion with an understanding of the tables as well as with a checklist of SQL and macro skills that are required to use the tables most effectively.



SESUG Speaker Sharing Program

Efficiencies with Large Datasets

Peter Eberhardt

This presentation will show you some simple tricks to help you manage large datasets. It starts by looking at ways of making large datasets smaller. It follows up with issues of sorting, matching and lookups – three of the major hurdles of large datasets. Finally, it touches on a number of programming issues that have a positive impact.

Elementary Statistics Using Base SAS

Debbie Buck

Base SAS includes a number of procedures that will allow you to perform elementary statistical analyses. Most of these PROCs produce descriptive statistics, but there are some capabilities for inferential statistics (hypothesis testing) as well. This paper will discuss which Base SAS PROCs will provide you with the desired statistics and show examples of code and output for a research study.

Explaining Unexpected Log Messages and Output Results from DATA Step Code

Debbie Buck (Co-author: Larry Stewart)

Everyone knows that you should always read the SAS® log every time you execute a SAS program, especially if it includes a DATA step. When you read the log, do you ever find SAS notes or warnings that puzzle you? You know the ones like "The variable XYZ is uninitialized" or "The variable ABC in the DROP, KEEP, or RENAME has never been referenced." But when you look at your output everything seems to be okay. So what do those messages mean and should you be concerned?

Do you submit DATA steps that produce no notes or error messages in the log, but the code produces unexpected results? For example, you write a basic IF/THEN statement to delete specific observations, but it deletes all the observations. If you answered yes to any of the questions above, then this might be the presentation for you. This presentation focuses on common unexpected notes in the SAS log and why they occur. It also addresses common logic errors in DATA steps that lead to unexpected results.

Evaluating Sample Code for an Interview

Stephanie Thompson

Requesting sample SAS® code from job candidates before an interview is a good way to gauge their level of experience, ability, and style. Once you have reviewed the code, prepare your questions for the candidate. You want to make sure that they 1) wrote the code themselves, 2) understand what they wrote, and 3) are familiar enough with it to explain why things were done the way they were. This paper will provide you with some guidelines on how to get answers to these questions in an interview. Code samples that I have received over the last few years and the questions that arose will be used as examples. This is also a good time to ask if the candidate can think of another way to accomplish the same task using a different approach. Evaluating work samples can be a way to gain invaluable insight into your candidate's skills and this paper will help you get the most out of it that you can.



SESUG Speaker Sharing Program

Four Thousand Reports Three Ways

Stephanie Thompson

How do you go about generating over four thousand PDF reports in up to three different versions? When a large, southern research university decided to add up to five optional questions per class section and up to five more questions at the prefix level to their core set of fifteen questions on the student evaluation of faculty survey, it seemed like a project that would never be completed. If the additional questions weren't enough, the reports were being revamped at the same time to improve their appearance for delivery on the web. Each report had a tabular section and two customized box and whisker plots. Thanks to ODBC / SAS Access, PROC SQL, macro, DATA Step programming, PROC GPLOT, goptions, and ODS it all came together. This paper summarizes how each SAS® component was used and contributed to the completion of the project.

Fun with Functions

Marje Fecht

Functions can be fun (and useful) if you know when, how, and why to use them. Functions can be frustrating if you don't know the tricks behind them.

In this presentation, we will explore the most commonly used functions to streamline data processing and reduce your programming effort. You will learn how to use summary functions, string functions, conversion functions, and date functions. We will also explore some of the 60 functions added to Version 9 SAS software.

This presentation will also help you understand WHERE to use functions, since they aren't "just for the DATA Step".

Getting Data into SAS®: INFILE and INPUT

Andrew T. Kuligowski

The SAS System has numerous capabilities to store, analyze, report, and present data. However, those features are useless unless that data is stored in, or can be accessed by, the SAS System. This presentation includes an introduction to the INPUT and INFILE statements, which combine to provide a simple, yet powerful, method to pass data into the SAS System. Matters to be addressed are reading fixed and variable length files, .CSV files, in-stream data, informats, and more.



SESUG Speaker Sharing Program

Getting Your Data into SAS®

Stephanie Thompson

Where is your data stored?

- Oracle tables
- SQL Server tables
- Microsoft Access
- Microsoft Excel
- Text file
- All over the place

Over the years there has been a proliferation of ERP systems and other ways to collect and store data. Many times you need data from different systems to complete a single analysis. Sometimes getting data out can seem like quite a challenge. Luckily, SAS has the capability to access many different types of data but also different ways to do so. This workshop will cover how to use SAS to access data from a variety of sources through both presentation and live demonstration. Some practical tips on which methods work best or are fastest will also be covered.

The following methods to access data will be covered in the workshop:

- SAS libname engine
- PROC SQL
- ODBC
- PROC IMPORT (including the Import Wizard)

Graphing the Easy Way with SAS® Enterprise Guide® (or How to Look Good With Less Effort)

Stephanie Thompson

Have you ever wanted to make some fancy graphs but were intimidated by coding them from scratch? Let Enterprise Guide® do the hard part for you. This hands-on workshop will provide a brief overview of Enterprise Guide then walk you through creating different types of graphs. Bar charts, pie charts, scatter plots and more will be created. Once the graphs are done, we'll "borrow" the generated code, put it in another SAS® program, make a few modifications, and then voila - fancy graphs from your program. We will use both the programming interface and Enterprise Guide in this workshop.

Note: This 90 minute hands-on workshop can be offered as a 60 minute tutorial.



SESUG Speaker Sharing Program

Here's the Data, Here's the Report I Want - How Do I Get There?

Debbie Buck

SAS® programmers are often given a set of data and asked to generate a specific report. This can be a daunting task, especially for a relatively new SAS user. Where to start? This process involves two separate, but related phases. Phase one concerns the data itself. What form are the data in – is it raw data, an existing SAS data set or sets, or data from another software program? Also, does the data need to be manipulated, restructured or summarized to provide the information needed for the SAS procedure? Phase two involves determining which PROC will provide the output in the desired form. Knowing whether the report needs a specific layout, customized information, or statistics is essential in deciding on the most appropriate PROC. This paper will present some guidelines to follow for producing the desired report in a logical and organized manner using Base SAS.

Individual Growth Analysis Using PROC MIXED

Maribeth Johnson

Individual growth models are designed for exploring longitudinal data on individuals over time. PROC MIXED allows the growth parameters for each individual to be examined as random effects in the model. Individual-level covariates can be entered into the model as fixed effects to determine their impact on the dependent variable alone and in interaction with the growth parameters. The structure of the variance-covariance matrix of the repeated measurements can also be examined and entered into the model. A model building exercise will be demonstrated using up to eight systolic blood pressure measurements of youths aged 7-22

In Search of the LOST CARD

Andrew T. Kuligowski

“Everyone who’s not here, raise your hand”. It’s an old joke, but it points out the difficulty of identifying persons or things that are not present. The SAS® system has its own version of this chestnut, the SASLOG message indicating that there are one or more gaps in one’s input data:

NOTE: LOST CARD.

This presentation will focus on the creation and use of adhoc to explore the positions where the input data might be incomplete. The goal will be to identify where the missing data should be, so that you can code around the limitations of your data.

NOTE: This is only a 20-minute presentation. The user group who requests this might want to consider it as a second paper to another being presented by the author.



SESUG Speaker Sharing Program

Labels, Labels, and More Labels

Stephanie Thompson

SAS® datasets include labels as optional variable attributes in the descriptor portion. Labels are not a nuisance but something that you can get a lot of use out of. Learn how to get the most out of them in your programs. Here are just a few of the topics that will be covered: how labels can make reports more readable, how to permanently or temporarily create or change labels, how to use them in procedures or not use them, and how to extract them into a dataset or a macro variable. Consider this paper a primer on the variable label.

Note: 30 minute talk

Looking Beneath the Surface of Sorting

Andrew T. Kuligowski

Many things that appear to be simple turn out to be a mask for various complexities. For example, as we all learned early in school, a simple drop of pond water reveals a complete and complex ecosystem when viewed under a microscope. A single snowflake contains a delicate crystalline pattern. Similarly, the decision to use data in a sorted order can conceal an unexpectedly involved series of processing and decisions.

This presentation will examine multiple facets of the process of sorting data, starting with the most basic use of PROC SORT and progressing into options that can be used to extend its flexibility. It will progress to look at some potential uses of sorted data, and contrast them with alternatives that do not require sorted data. For example, we will compare the use of the BY statement vs. the CLASS statement in certain PROCs, as well as investigate alternatives to the MERGE statement to combine multiple datasets together.

Many to One Using a SAS® DATA Step and PROC MEANS

Jennifer Waller

Medical claims data from Medicaid, Medicare, or private insurance companies, financial data, and other industry data contain information for various needs. For example, medical claims data contains information regarding medical procedures, diagnoses, prescriptions, and more. When using these data, often a single individual will have multiple observations within a given time period of data extraction. To add to this complexity, each observation can have pertinent data in many variables within the observation. Obtaining a single observation regarding whether an individual had a particular outcome (e.g. did the individual have the diagnosis, did they buy a particular brand) requires examining many variables within a single observation and then examining the many observations that pertain to the individual. One might think that determining an individual's outcomes a daunting task. However, there is a somewhat easy solution. Using ARRAY's and DO loops in the DATA step and then PROC MEANS, data across many variables and many observations for each individual can be combined into a single observation to determine outcome measures. The explanation of how to creating a single observation from multiple variables and multiple observations will be done using insurance claims data.



SESUG Speaker Sharing Program

More than Models: The Data Mining Team

Stephanie Thompson

Gathering data from various sources, preparing it for modeling, imputing, partitioning, testing various models, choosing the best, presenting it to your boss, then failure? Data mining is about more than just variables and models. Developing an understanding of each variable may take more than just running some summary statistics and deciding if it is ordinal or nominal. The need for comprehension is even more critical when the data you use are from different areas of the organization. Putting together a group of subject matter experts in the early phases of a data mining project can make a big difference in the outcome of your project. They can help you eliminate extraneous or duplicative variables and put others in context to help you better understand and interpret results. This paper will discuss how subject matter experts can aid in data mining using examples from several actual projects. Learn how to leverage the knowledge to derive a better conclusion and avoid costly errors.

Moving Forward with FORMATS

David Maddox

FORMATS provide an instruction or template that SAS® uses to output data values. They are used primarily to control the written appearance of data values. But, FORMATS have other uses as well. This presentation will feature some of those additional uses, including table lookups and traffic lighting. This is a basic to intermediate presentation.

New to SAS® and New to Programming? What You Need to Do Before Typing Code

Stephanie Thompson

People from all kinds of backgrounds are using SAS® software to meet their business needs. Some come with previous programming experience and some have never typed a line of code in their career. Have you been told what each proc does, that each DATA step must end in a run command, and then been told to write a SAS program? Have you been given code to launch and told to “just change the date and run it every week”? Beware. There is a lot more to SAS programming than knowing some syntax and launching a job. Being a successful SAS programmer means bringing yourself up to speed with some basic programming skills. Before typing even the first line of code into the Enhanced Editor, some preparation needs to be done. Doing this will help you more than you can imagine. This paper provides the basics of programming that can help you write good code and ensure you are getting the answer you intended to get.



SESUG Speaker Sharing Program

Notes from an Intersection: Google Earth @ SAS®

Carol Martell

This paper examines the issues one encounters using SAS to create data-driven placemarks viewable in Google Earth. The introduction of bike lanes in certain areas of St. Petersburg, Florida, provides a scenario for comparing before-and-after traffic counts. The traffic count data, while providing an intersection label, does not provide latitude and longitude. Come along for a ride on the SAS train from proprietary traffic counts to Google Earth; there are sure to be many detours. Is it a rollercoaster or a kiddie ride?

At the 2008 SAS Global Forum, two integrative techniques were unveiled. Kuiper and Vyverman showed us how to use SAS XML Mapper to read, and in SAS 9.2, write Google Earth XML (KML). Drukenbrod and Mintz showed us how to provide dynamic query access at placemarks using SAS/IntrNet® Application Dispatcher and how to create county-area colored overlays for Google Earth. With these techniques in hand, we set out in search of some appropriate ways to display our data.

PROC REPORT...Your How-to Guide for Producing Customized Summary Tables

Debbie Buck

A number of years ago SAS® developers responded to requests from SAS programmers for a single report-writing procedure that incorporated the features of PROC PRINT, PROC MEANS, PROC TABULATE, and the DATA _NULL_ step. The resulting procedure, PROC REPORT, is highly versatile with the capabilities to produce both detail and summary reports.

In this workshop we will focus on PROC REPORT statements that will produce complex summary tables which include multiple dimensions, frequently reported descriptive statistics, columns of information not included in the data set, and customized summary statements.

Although PROC REPORT can be used in a windowing environment with point-and-click report building, we will use a programming approach to create the reports.

This workshop is intended primarily for beginning to intermediate SAS users, or anyone who would like to become more familiar with PROC REPORT, and is applicable to all operating systems.



SESUG Speaker Sharing Program

Programming for Failure: When Programs Faw Down and Go Boom

Gary Schlegelmilch

Programs fail.

Despite the best efforts of the best programmers, a non-working combination of data, software, and processing is invariably going to happen - and the program is not going to run. Typical result is a sometimes instantaneous, sometimes arduous effort to find where the program has failed - and fix it.

This presentation attempts to cover two major topics: (1) a few ideas for finding the elusive bug, and (2) planning and structuring a SAS program so that perhaps a few less of them happen.

Rules for Tools - The SAS[®] Utility Primer

Frank Dilorio

Let's start with the premise that good programmers are lazy by nature. They want to use tools such as formats and ODS for execution-time efficiency or to pretty-up our output, functions to perform calculations, and so on. Another hallmark of a good programmer is a keen eye for pattern recognition. Rather than rewrite basically the same program over and over, they identify similarities and parameterize the program, making it into a general-purpose program, a "utility."

This paper steps through the life cycle of a simple utility. It starts with "naïve" code that doesn't exploit program similarities, then illustrates how a general-purpose utility may be developed. It ends with the initial program becoming a call to a simple, powerful routine in a macro library. The transition from simple, brute-force programming into a compact, general-purpose utility isn't a random event. The last sections of the paper present a set of design principles for utilities.

Although we focus on Base SAS in Version 9.0, the principles and techniques are readily extended across SAS versions and products. The reader will come away from this paper with an appreciation of both the process and the tool set required to build generalized programs.

SAS[®] as a Utility Language: Using SAS for Solving Little Problems

Gary Schlegelmilch

SAS, from its very conception, was intended to provide methods for determining and handling statistics cleanly, clearly, and effectively. It does so with speed, strength, and versatility. However, the tools that affect its versatility lend themselves with equal ease to nonstatistical projects as well. Small day-to-day tasks – impromptu reports, quick reviews of data, and other small utility programs – are handled in SAS quickly and efficiently.

Because SAS provides minimal data field definition, a modular approach in the DATA step, and clearly defined processes in the PROC step, small tasks can be programmed with a minimum of effort. In addition, the turnaround time will frequently be less than the same routine written with a compiled/linked language.

When time is of the essence, SAS may very well be the best candidate.



SESUG Speaker Sharing Program

SAS® System Options Are Your Friends

Ed Heaton

Does SAS® always do things the way you want? Have you ever made a simple little mistake of omission with disastrous consequences? Do you find the SAS log difficult to read and less informative than you need? Do you keep doing the same little tasks over and over because SAS doesn't remember what you want?

System Options can be your solution to some of these problems. It isn't hard to pick out some System Options to make your code easier to manage. SAS provides a lot of System Options. There are about 258 in v8.2 for Windows. With version 9.1, ten of them went away and 68 new options were introduced. SAS 9.2 has about 370 system options. (The available system options vary with your operating system and your licensed products.)

Although their implementation can sometimes be confusing, system options should not be ignored. The author will explain a little about their usage in a Windows environment and will present a handful that will make your work easier and less prone to simple mistakes.

SAS® XML Mapper to the Rescue

Carol Martell

Have you ever been given XML data without the requisite Document Type Definition (DTD) or XML Schema describing the data? Certain XML formats are supported in SAS through the XMLTYPE= option in the XML libname statement. If your data conforms to one of the supported formats, then you're in luck. If not, you have a problem. The tags surrounding each data item can seduce an accomplished programmer into writing a DATA step program to read the data. While this is certainly possible, it isn't necessary. The SAS XML Mapper is a separate Java application from SAS that can, among other things, create a custom SAS XMLMap. The SAS XMLMap is used by the XML engine to interpret an XML document. The XML libname engine becomes your friend again, when the XMLMAP= option points to your new XMLMap file.

Set, Match, Merge ... Don't You Love SAS®

Peter Eberhardt

One table has accounts, another has new transactions. One table has patient data, another has treatments. No matter what your business or application it is rare that a single data set or table has everything you need to answer the questions you are expected to answer. SAS® has a simple and powerful mechanism to bring your tables together: the DATA Step Merge. But this simple operation can easily run awry. In this paper we will explain the basics of the DATA step merge and the issues and problems you will encounter if you do not understand its workings. This paper is geared towards beginning SAS programmers, but it can be a useful refresher for anyone.



SESUG Speaker Sharing Program

Shuffling, Shifting, and Structuring Using the Program Editor

Gary Schlegelmilch

Invariably, when we go to a SAS® programming class, we are taught the basics of how the language works and ways to lay out a program. The problem is that once we leave an academic arena and move into the professional one, we don't always develop the habits we need to write readable, maintainable code. Or, we inherit code that was written by someone else who never learned the skills and are required to update it for new requirements. In this paper, we'll discuss some of the basic structuring tools available within the SAS Program Editor® window, and how to use them effectively. This will help even the most advanced programmer, as well as the novice in the business, to both write new code and to update outmoded, unstructured, and just generally bad coding.

So, Your Data are in Excel!

Ed Heaton

You say your customer sent you the data in an Excel workbook. Well then, I guess you'll have to work with it. This paper will discuss some of the quirks you will find when your data are stored in an Excel workbook. It will cover such things as

- * naming conventions,
- * named ranges,
- * character length issues,
- * numeric precision
- * dates, time, and datetime values,
- * mixed data types, and
- * caching.

This paper will demonstrate - through the Display Manager - SAS code and techniques to make your life with Excel more predictable and your work less prone to error. We will use the EXCEL engine that became available in SAS 9 with your license for the SAS/ACCESS® Interface to PC Files.

Summarizing Data with Base SAS® PROCs

Debbie Buck

Base SAS provides a number of procedures designed to aid the SAS user in developing data summary reports. These procedures include MEANS, UNIVARIATE, FREQ, REPORT and TABULATE. Additionally, PROC FORMAT is also available to modify the appearance of data values or combine values into desired categories within PROCs without changing actual data values or creating new variables.

This presentation explores various common types of summary reports and what factors you should consider in deciding which procedure is best suited to your reporting needs. It is not meant to be a detailed tutorial on PROC step programming code for all of these procedures, but rather a guide on how to decide which procedures fits the requirements for a given report.

Summary reports generated using Base SAS PROCs are presented along with the code producing these reports.



SESUG Speaker Sharing Program

Talking Past Each Other? How to Communicate with Medical Writers When Preparing Clinical Research Manuscripts for Journal Submission

Stephanie Thompson

Clinical research manuscripts are often a blend of qualitative prose and quantitative data. And, the people who help the author(s) prepare each section — introduction, methods and materials, results, and discussion — are typically from very different parts of the research organization and have different backgrounds. Speaking a common language makes the process easier. This paper will present some tips on how to improve communication between medical writers and SAS® programmers, such as biostatisticians. Presented from both points of view, what does and doesn't work will be discussed in a point, counter-point fashion.

The Design and Use of Metadata: Part Fine Art, Part Black Art

Frank DiIorio

The complexity of even small pharmaceutical projects can be daunting. Consider the deliverables: patient profiles, listings, domain and analysis data sets, Define files, tables, and figures. Even in a single study, these routinely total hundreds of files. For NDA submissions, these are but a single piece of a larger “puzzle.”

Consider as well the documentation and human resources pushing the study through its life cycle. Project managers need to monitor the completion status of the files. Statisticians and analysts have to identify data requirements and lay out “dummy” displays. Programmers have to write the programs to create the data and reports using specifications that are often, to be kind, “fluid.” Creation of high-quality output requires coordination of effort and clear and immediate communication of results. Rho has migrated much of the requisite project management and data and display specifications to carefully designed and utilized metadata. By moving items that describe data sets and displays from documents and low-level programs into data sets, we have realized significant gains in productivity and quality of output.

This paper describes the current use of metadata at Rho. It:

- o Discusses the motivation for using metadata
- o Describes the metadata architecture
- o Identifies tools that access the tables
- o Presents examples, comparing metadata and non metadata-driven programs.

The paper is largely conceptual and nearly code-free. While we emphasize application development in the pharmaceutical industry, we feel the underlying concepts regarding metadata design and implementation are valid across industries.



SESUG Speaker Sharing Program

The Effect of Missing Data on Repeated Measures Models

Maribeth Johnson

Researchers involved with longitudinal studies are faced with the problem of trying to get study subjects to return for every follow-up visit. There is always some amount of missing data when looking at these types of studies. The MIXED procedure of the SAS[®] enables examination of correlational structures and variability changes between repeated measurements on experimental units across time. While PROC MIXED has the capacity to handle unbalanced data when the data are missing at random, a question arises as to when the degree of sparseness jeopardizes inference regarding the variance-covariance error structure of the multivariate repeated measures. Simulation is a tool that can be used to answer these types of questions. This paper shows the application of simulation to determine inference problems regarding the correlational structure in a data set with a specific pattern of missing data. This technique is also applied to the topic of initial sample size determination.

The SAS[®] Community: How to Be a Part of It!

Gary Schlegelmilch

OK, so now you've learned how to program in SAS. Or perhaps you're an end user, learning how to use the Enterprise Guide. Or you could be a student, just learning; or someone who's been involved with SAS since it became available for use in 1976.

So now what?

Invariably, we all have questions. We run into programming situations we can't quite figure out. A process that we believe should work – but we can't quite seem to line up the code correctly to do it. Or, the other side of that coin – you come up with a perfect solution to a problem you've seen a hundred times, thanks to a new version of an old function in SAS.

Most other languages have neither the active community that SAS does, nor do they have the staunch support that the SAS Institute has offered virtually since its inception. The users of SAS are communicators; we exchange ideas, techniques, and code easily and with a smile. So if you'd like to be a part of it – read on.

The SAS Data Step: Where Your Input Matters

Peter Eberhardt

Before the warehouse is stocked, before the stats are computed and the reports run, before all the fun things we do with SAS[®] can be done, the data need to be read into SAS. A simple statement, INPUT, and its close cousins FILENAME and INFILE, do a lot. This paper will show you how to define your input file and how to read through it, whether you have a simple flat file or a more complex formatted file.



SESUG Speaker Sharing Program

The SAS® Debugging Primer

Frank DiIorio

Meet an accomplished SAS programmer and you meet someone who's probably learned by making (and fixing) lots of mistakes along the way. The breadth of the SAS System's target applications, the variety of its "dialects" (Base SAS, macro, SCL, IML, SQL), and the quirky procedural/non-procedural environmental mix conspire to make mastery of the SAS System a slippery slope to ascend. Debugging is the art of gracefully recovering and learning from falls during the ascent.

This paper discusses techniques for debugging SAS programs. Its purpose is two-fold. First, it provides behavioral and technical tips for fixing code (how to read error messages in the SAS Log, knowing when there is a problem with the program even if SAS says there isn't, using the DATA step debugger, identifying system options, using PROCs for data validation, using macro variables to control debugging output, etc.) The second focus of the paper is its presentation of design and coding methods that make the programming process more reliable, thus reducing the need for debugging in the first place.

The paper's target audience is relative newcomers to the SAS System. More seasoned users may find or rediscover some of the techniques and features being discussed. Emphasis is placed on Base SAS and the macro language, although the techniques themselves are applicable to SCL and other products.

The SAS® Hash Object: It's Time To .find() Your Way Around

Peter Eberhardt

"This is the way I have always done it and it works fine for me."

Have you heard yourself or others say this when someone suggests a new technique to help solve a problem. Most of us have a set of tricks and techniques from which we draw when starting a new project. Over time we might overlook newer techniques because our old toolkit works just fine. Sometimes we actively avoid new techniques because our initial foray leaves us daunted by the steep learning curve to mastery. For me, the PRX functions and the SAS® hash object fell into this category.

In this presentation, we start with the fundamentals of setting up the hash object and work through a variety of practical examples to help you master this powerful technique.

Things Dr Johnson Did Not Tell Me: An Introduction to SAS® Dictionary Tables

Peter Eberhardt

SAS maintains a wealth of information about the active SAS session, including information on libraries, tables, files and system options; this information is contained in the Dictionary Tables. Understanding and using these tables will help you build interactive and dynamic applications. Unfortunately, Dictionary Tables are often considered an 'Advanced' topic to SAS programmers. This paper will help novice and intermediate SAS programmers get started with their mastery of the Dictionary tables.



SESUG Speaker Sharing Program

THINK Before You Type... Best Practices Learned the Hard Way

Marje Fecht

We have all done it...

You leave a meeting with a new project on your plate – you sit down at your computer and start programming – a day later and lots of code on the screen, you try to remember the real purpose of your latest task. Or, worse, you finish your task only to find that it is not what was requested. Discipline and planning aren't always easy but they pay off in the long run!

OR

You deliver the quick adhoc and move onto the next task on the list. A month later, someone wants the "adhoc" again with just a wee variation. WHAT??? You thought you would never look at that code again? Generalization was the last thing on your mind and now you have to dig through all that code... Discipline and planning aren't always easy but they pay off in the long run!

This presentation focuses on best practices to help you minimize effort and maximize results. Although the concepts are more project-focused than SAS[®]-specific, example SAS code will be shared so that you can apply the concepts more quickly.

Using SAS[®] to Parse External Data

Andrew T. Kuligowski

Most "Introduction to Programming" courses will include a section on reading external data; the first assumption they make will be that the data are stored in some sort of documented and consistent format. Fortunately, in the "real world", a lot of the data we deal with has the same basic assumption of occurring in a documented, consistent format - a lot of it, but not all of it.

This presentation will address some techniques that can be used when we are not dealing with cleanly formatted data, when the data we want is in a less-than-ideal format, perhaps intermingled or seemingly buried with unnecessary clutter. It will discuss the principles of using SAS[®] to parse a file to extract useful data from a normally unusable source. This will be accomplished by citing examples of unusual data sources and the SAS Code used to parse it



SESUG Speaker Sharing Program

Where's The Match? Matching Study and Comparison Subjects after Data Collection

Jennifer Waller

Statisticians, while not data managers, are often asked to perform data management tasks. One task is to match a study group to a comparison group one-to-one on several variables. While easy for small datasets and matching only categorical variables, matching thousands of study subjects to thousands of comparison subjects where data come from large administrative databases and the matching variables can be categorical and continuous is difficult. How do you match continuous variables with the comparison subject being within some range of the study subject (e.g. age within +/- 2 years)? The utilization of SAS Macros is one solution. The created macro uses a study subject data set, a control data set, and outputs data set containing the one-to-one matched subjects. The macro reads in a study subject, scans through the comparison subject data set until a match is found, writes the match to the matched data set, deletes the comparison subject from the comparison data set, and proceeds to the next study subject. If a match is not found, the macro proceeds to the next study subject. The resulting matched data set contains those study subjects and comparison subjects that were matched one-to-one on the necessary variables.