

West Chester University
Graduate Program in Applied Statistics
presents
Full Day Training Workshops in
SAS SQL and SAS Macros

SAS SQL

Saturday December 19

8:30 AM to 4:30 PM

SAS Macros

Saturday January 9

8:30 AM to 4:30 PM

25 University Ave., Room 161, West Chester University

All West Chester University students, alumni, and friends of the Program are invited to attend.

Admission is free.

E-mail Dr. Randall Rieger at rrieger@wcupa.edu if you have any questions.

Workshop descriptions are included below.

Overview of SAS SQL Seminar

Audience for the seminar: Students with very little SAS experience, and SAS programmers with a few years experience, can benefit from this seminar. *People new to SAS should feel free to attend. This seminar is especially helpful for people with less than a year of Proc SQL.*

The seminar leader, Russ Lavery, has given these presentations both internationally (Europe and Asia) and state-side at such institutions as Harvard, UPenn, Rutgers and Philadelphia University. Parts of this presentation have won two awards at international conferences.

Overview: PROC SQL is very a powerful tool for querying data and is so versatile that it should be the entry point for learning SAS. If you know PROC SQL, *you can use it in place of many other SAS procedures*. It often allows you to “get the job done in SAS” without knowing all of SAS. It is the Swiss army knife of procedures and should be understood by every programmer and statistician.

In addition to PROC SQL being important in SAS, learning SQL is the key to learning other software and is an important career skill. SQL is the basis for Queries in M.S. Access, Oracle, Business Objects, and many other packages.

SQL has been difficult to learn because SQL has been a “black box”. The historical teaching method has been to have the student run dozens of SQL queries until s/he recognizes a pattern – but it has been impossible to explain why results come out as they do. This seminar shows a graphical model of previously undocumented PROC SQL internal processes and makes PROC SQL easy to learn.

The major deliverable of this seminar is the graphical representation of the SQL process and we use that graphical model to develop rules for describing, and predicting, the SQL process. These descriptive rules translate directly into coding rules and allow a programmer to quickly code PROC SQL queries.

The graphical model of SQL Queries	Displaying Query Results Summarizing data	Presenting data
The graphical model of SQL sub-Queries	Non-correlated sub-queries	Correlated sub-queries
SQL Joins	SQL joins vs Data step Joins Multiple SQL joins and the pivot table	Reflexive joins
Set Operators – so useful Venn you need them	introduction to set operators the EXCEPT operator the UNION operator	The INTERSECT operator The OUTER UNION operator
SQL as an environment	Creating views with the SQL procedure Integrity constraints SQL options	Dictionary tables
SQL Performance	Indexes and the where clause The SQL optimizer and the where clause	

There are no in-seminar *computer* exercises and laptops will not be needed. However, there will be group exercises during the seminar, to aid in understanding and increase retention of the presented material. Groups will be formed at 8:30 AM. **Please be on time so that you do not disrupt the others in the class.**

Overview of SAS Macro Seminar

The seminar leader, Russ Lavery, has given these presentations both internationally (Europe and Asia) and state-side at such institutions as Harvard, UPenn, Rutgers and Philadelphia University.

Audience for this seminar: This will NOT teach “basic SAS” and is NOT for people with less than 3 months (preferably people have 6 months) of SAS experience. Experienced programmers will see a graphical model of SAS that greatly simplifies thinking about macros. This graphically presented paradigm will help experienced programmers explain their macros to junior programmers and is very valuable. It is expected that this full-day seminar will reduce, by 200 man-hours, the time needed to learn macros.

Overview of SAS Macro Seminar

This seminar is presented to help students transition from doing analysis in SAS, as a student, to doing analysis in SAS as a working professional. While students often spend 90% of their time in statistical procedures, working SAS professionals spend 90% of their time in the data step and the macro language. Macros are used in almost every SAS corporate/consulting environment.

Learning the SAS macro language is often a time consuming, and stressful part of transitioning into the working world. Macros are used to make SAS code simpler and easier to maintain. Additionally, macros are used to automate processes and make processes/programs “production”. Without knowledge of SAS macros, a new employee cannot maintain, or sometimes even read, legacy SAS code (legacy code is code written by the last person who last had your job or by the person who sits in the next cube). Not being able to read code is a very unpleasant situation, with severe career implications.

For a high level summary, the seminar will cover:

The SAS supervisor Tokenization The Macro Catalog The Macro Symbol Table
% let and & Evaluating the &&& Call Symput & Symget Call Execute
Using macros to automate a process (two common ways to make a program),
Using macros to create data driven programs

Important “Take Aways” from the seminar are:

1000+ lines of SAS macro examples (all the examples in Seminar)

Sample Code for how to perform 4 common Macro tasks. This can be used as “starter code” when on a job as well as a “self-test on Macro” concepts.

Below is a detailed list of topics covered in the macro seminar

Four essential skills in 6 examples

Skill 1) wrap code in a macro and call it with different parameters

Skill 2) Automate jobs (have your program write your program)

a) % do loop and the &&state&i

b) automate with SQL and the %scan loop

Skill 3) Conditionally executing code with a %if

Skill 4) Making “macro code snippets” (macro’s that generate part of a SAS statement)

Components of the SAS system

Tokenization and the 4 types of tokens

The three compiles/executes that happen when running a macro program

The Macro Symbol Table and the Macro Catalog

Moving tokens into and out of the Macro Symbol Table and the Macro Catalog

Evaluating multiple ampersands (&StateNo &&StateNo &&&StateNo &&&&StateNo)

The effect of single vs double quotes

Global vs local variables

Rules that are applied to tokens as they flow into the Macro Catalog

Rules that are applied to tokens as they flow out of the Macro Catalog

Call symput Basics

Symget Basics

An example of Call Symput and Symget

Call execute: automating a series of reports with Call Execute and a 'driver" file

The rules for the parameters in: Call Symput, Symget and Call Execute.

Single vs double quotes in Call Execute parameters.

There are no in-seminar *computer* exercises and laptops will not be needed. However, there will be group exercises during the seminar, to aid in understanding and increase retention of the presented material. Groups will be formed at 8:30 AM. **Please be on time so that you do not disrupt the others in the class.**

You can download seminar materials from: <http://www.hcp.med.harvard.edu/node/2635>. You will want to scroll down the web page and download:

BW Handout

IS01 Macro Phuse

Macro Code to match slides with IAN examples

Overview of 4 Macros Tasks in 6 Macros