WUSS 2019 Trip Report for Jack Hamilton

Western Users of SAS Software
September 4-6, 2019
Seattle

A Hands-on Introduction to SAS® DATA Step Hash Programming Techniques

Presented By Kirk Paul Lafler

1. A hash object is a data structure that contains an array of items that are used to map identifying values, known as keys (e.g., employee IDs), to their associated values (e.g., employee names or employee addresses). As implemented, it is designed as a DATA step construct and is not available to any SAS PROCedures.

2. A hash object permits table lookup operations to be performed considerably faster than other available methods found in the SAS system. Unlike a DATA step merge or PROC SQL join where the SAS system repeatedly accesses the contents of a table stored on disk to perform table lookup operations, a hash object reads the contents of a data set into memory once allowing the SAS system to repeatedly access it, as necessary.

3. Available in all operating systems running SAS 9 or greater, the hash object is called using methods. The syntax for calling a method involves specifying the name of the user-assigned hash table, a dot (.), the desired method (e.g., operation) by name, and finally the specification for the method enclosed in parentheses.

A Quick Look at Fuzzy Matching Programming Techniques Using SAS® Software

Presented By Kirk Paul Lafler and Stephen Sloan

1. When data sources and data sets contain consistent and valid data values, share common unique identifier(s), and have no missing data, the matching process rarely presents any problems. But, when data originating from multiple sources contain duplicate observations, duplicate and/or unreliable keys, missing values, invalid values, capitalization and punctuation issues, inconsistent matching variables, and imprecise text identifiers, the matching process is often compromised by unreliable and/or unpredictable results.

2. We adhere and recommend the following six step fuzzy matching process. Step 1: Determine the Likely Matching Variables using Metadata. Step 2: Understand the Distribution of Data Values. Step 3: Perform Data Cleaning. Step 4: Perform Data Transformations. Step 5: Process Exact Matches. Step 6: Match Key Fields using Fuzzy Matching Techniques.

3. SAS Institute offers four techniques for its users: the Soundex (phonetic matching) algorithm, and the SPEDIS, COMPLEV, and COMPGED functions to help make fuzzy matching easier and more effective.

A Survey of Some of the Most Useful SAS Functions (No paper available)

Presented By Ron Cody

1. Use CALL SORTN to sort WITHIN an observation

2. Use COUNTN and CATS together to eliminate DO loops and arrays

Connecting to Datasets through Python and SAS

Presented By Joe Matise

1. Instead of writing code to download data from major datasources, connect to Python and use already published code!

Decision Trees: a Gentle Introduction

Presented By Richard Hector

1. Demystify decision-analysis - so your AI implementation won't give you an unpleasant surprise

2. Solve the problem first then write the code of the solution - It is worth the effort to write and develop a statistical analysis plan.

3. Decision-analysis helps you think through the problem. It's worth the time and effort

Doing More with the SGPLOT Procedure

Presented By Josh Horstman

1. You can create combination plots in SGPLOT by including multiple plot request statements. Plots will be overlaid in the order listed within the procedure call.

2. Use the MARKERATTRS and/or LINEATTRS options on your plot request statement to customize the appearance of plot markers and lines.

3. Use the STYLEATTRS statement (added in SAS 9.4) to customize plot attributes on the basis of a grouping variable.

Easy Solutions To %LET You Reduce Repetitive Programming

Presented By Ekaterina Roudneva

1. Use %Let statement in the beginning of your programs to define commonly used dates and file paths

Everything is better with friends: Executing SAS code in Python scripts with SASPy

Presented By Isaiah Lankham and Matthew Slaughter

1. SASPy is a module for the Python programming language, allowing SAS procedures to be executed in Python scripts using Python syntax.

2. SASPy allows SAS programmers to take advantage of the flexibility of Python for flow control and Python programmers to incorporate SAS analytics into their scripts.

3. All workshop examples are available as an open-source project at https://github.com/saspy-bffs/. We will be adding examples in the near future, and we welcome your examples, as well. A good overview of how to contribute to an open-source project can be found at https://www.firsttimersonly.com/

Fifteen Functions to Supercharge Your SAS® Code

Presented By Josh Horstman

1. Use the IFC and IFN functions to incorporate IF/THEN/ELSE logic into a function call.

2. The CAT family of functions can concatenate any number of character arguments and provides several advantages over using the concatenation operator.

3. Use TRANWRD, not TRANSLATE, when you want to replace a specific substring with a different one. TRANSLATE operates instead on a character-by-character basis.

Great Time to Learn GTL: A Step-by-Step Approach to Creating the Impossible

Presented By Richann Watson

1. The power of GTL allows you to customize your graphs.

2. The different LAYOUTS are utilized to help produce the desired graph (or graphs) for the output.

3. Breaking the graph into components you know how to do and then combining makes complicated graphs manageable.

Have it Your Way: Using the ODS Excel Destination with the Data Step Report Writing Interface

Presented By Pete Lund

1. The data step Report Writing Interface gives you complete control over the appearance of your tabular output and can send those results to any ODS destination - including Excel.

2. Native Excel formats and formulas can be used in creating spreadsheets from a data step.

I Don't Use Python or R - Why Should I Use Jupyter Notebooks?

Presented By Jack Hamilton

1. You don't need to be a Python or R user to make use of Jupyter Notebooks.

Interactive Graphs!

Presented By Richann Watson and Kriss Harris

1. Interactive graphs/drill down can be achieved using the standard GTL code with a few additional options.
2. Interactive graphs can even be done using SAS University Edition.

Keyboard Macros! An awesome tool you may have never heard of - once you do, you will never program the same again (It's that amazing!)

Presented By Steven C. Black

1. Use the Keyboard Shortcut keys CTRL + SHIFT + L to lower case highlighted text.
2. Use the Keyboard Shortcut keys CTRL + SHIFT + U to upper case highlighted text.
3. CTRL + ALT + (negative sign on keypad '-') to Collapse all folding blocks.

Learning Data Science with SAS University Edition and JupyterLab (No paper available)

Presented By Brian Gaines

1. SAS University Edition, which is available at no charge for noncommercial purposes and can be downloaded from the SAS website, now includes the popular JupyterLab interface.
2. Code cells with their corresponding output and Markdown cells with formatted text and math notation can be combined to create a reproducible analytical document that effectively communicates your results.
3. In a Jupyter notebook, you can use the built-in magic functions, %showLog and %showFullLog, to display the SAS log of the previous code submission.

Merge with Caution: How to Avoid Common Problems when Combining SAS Datasets

Presented By Josh Horstman

1. The safest way to merge datasets is to include only the DATA, MERGE, BY, and RUN statements in the DATA step and move any additional logic to a separate step.
2. When merging datasets on a BY variable of mismatched lengths, use a LENGTH statement before the MERGE statement to explicitly set the length and avoid truncation issues.
3. When merging datasets having variables in common other than your BY variables, drop or rename the variables from all but one dataset to avoid subtle problems that can result from overlapping variables.

Modern data visualization for SAS users

Presented By Jay Ahn

1. SAS can be used to create JSON objects that are accessible by any web browsers.
2. Using the HTML 5 canvas, visualize the data dynamically including most 3D charts available, using the JSON objects output by the SAS programming.
3. Use JavaScript libraries such as reveal.js to present the data with a full control (drawing board, highlighter, and text to speech.)

Multi-way Splits in Decision Trees Where The Dependent Variable Has More Than Two Levels

Presented By Russ Lavery and YuTing Tian

1. SAS Enterprise Miner can create decision trees that split more than two ways out of a node.
2. Do a PROC FREQ to find out the number of levels of Y in a decision tree and split your leaves that number of ways.

3. You can request the slides from this presentation from the author and repeat this talk for your coworkers.

Order, Order! Four Ways to Reorder Your Variables, Ranked by Elegance and Efficiency

Presented By Louise Hadden

1. There are at least seven ways to reorder variables, six data step statements (ARRAY, ATTRIB, FORMAT, INFORMAT, LENGTH and RETAIN) and PROC SQL. Each of these methods have different advantages and disadvantages.

2. It is possible to create ARRAY, ATTRIB, FORMAT, INFORMAT, LENGTH and RETAIN statements programmatically instead of manually using SAS metadata (this technique can also be utilized for other purposes.)

3. PROC SQL allows users to "alias" variable names while reordering, including changing variable case where that is a concern. The "as select" statement can also be partially constructed using SAS metadata.

PARSING: Using SAS® When the Data Are Hiding in a Non-Standard Format

Presented By Andrew Kuligowski

1. Many SAS character functions are used in parsing - the pulling of data from unusual formats - and can be used for a myriad of other programming problems, as well!

2. The CATX function can replace the combination of TRIM and "||" (concatenate) that used to be required to concatenate character variables.

3. LENGTH function gives character string length without trailing blanks, LENGTHC gives it counting them.

Polynomial Regression for Modeling Curvilinear Data: A Biological Example

Presented By Elisha Johnston

1. Polynomial regression is useful for analyzing curvilinear data.

2. A second order polynomial regression model (quadratic) assume symmetry.

3. BIC has a larger penalty for additional parameters than AIC and consequently BIC may be a better tool for defending against overfitting.

Python and R made easy for the SAS Programmer

Presented By Janet Li and Varaprasad Ilapogu

1. Two very useful tools for SAS programmers wishing to expand their AI knowledge are Python and R.

2. Some of the parallels in the syntax of SAS, python, R may help allay concerns that the clinical SAS programmer may have with learning a new language.

Quote the SASLOG

Presented By Andrew Kuligowski

1. A SAS routine can be used to analyze a SASLOG.

SAS Jeopardy! 2019 Edition (No paper available)

Presented By Natasha Chapman

1. The BLANKLINE= option in PROC Print can be used to print a blank space after every n rows.

2. In the IF statement, <> resolves to MAX (as opposed to "not equal to").

3. PROC Datasets allows you to modify your data set without using a DATA step.

SAS to Excel: ODS Tagsets to Hundred Files or Hundred Tabs

Presented By Abdus Shahid

1. Use ODS Tagsets to Output files with minimum manual formatting.
2. Utilize Table of Contents option to navigate between multiple tabs.

Simple Steps for Great Graphics: Beyond the Default Settings for More Effective Data Visualization

Presented By Kristi Metzger

1. The default settings for graphs in SAS can be improved dramatically with just a few extra lines of code.

TB Fuzzy-Matches Made Easy: Designing shareable SAS code to accurately and efficiently identify exact and close genetic matching tuberculosis isolates

Presented By Evan Timme

1. Use the SUBSTR function in a DO LOOP to count variable similarities “count = 00; do i=1 to 15; if substr(var_a,i,1) = substr(var_b,i,1) then count +1;”
2. Keep code adaptability in-mind from the beginning. Pre-planning your macros saves time and reduces future errors.
3. ZIPCITYDISTANCE function is an easy tool for quick distance assessments

Tips for Correctly and Efficiently Comparing Two Files in SAS®

Presented By Aaron Brown

1. Be wary of formatting errors when reading Excel files into SAS via PROC IMPORT.
2. PROC COMPARE can let you compare two versions of the same data.
3. PROC SQL's EXCEPT set operator can let you examine records in one dataset but not in another.

User-Defined Multithreading with the SAS® DS2 Procedure: Performance Testing DS2 Against Functionally Equivalent DATA Steps

Presented By Troy Hughes

1. PROC DS2 enables SAS developers to create user-defined DATA steps that are multithreaded, performing some tasks in parallel.
2. PROC DS2 can execute some computationally complex (i.e., requiring intensive CPU resources) operations faster than functionally equivalent DATA steps; however, DS2 will perform less complex tasks slower and less efficiently than functionally equivalent DATA steps.
3. To understand whether PROC DS2 can benefit a particular program or problem set, SAS practitioners must have a firm grasp on both their environment's system resource availability and the system resource utilization (and computational complexity) of the program or problem set.

Using Jupyter to Take Your Data Science Workflow to the Next Level

Presented By Hunter Glanz

1. Jupyter is a fantastic tool for including live SAS code into your report on the fly!
2. Learn more about Jupyter Notebooks through SAS University Edition.
3. You can use Jupyter with tons of languages in addition to SAS, like Python and R.

Using PROC FCMP to the Fullest: Getting Started and Doing More

Presented By Art Carpenter

1. The FCMP procedure allows the user to define, compile, and store their own functions. These functions can then be easily shared within a group of SAS users and programmers.

2. FCMP functions and routines are automatically configured for use with the macro language. %SYSFUNC and %SYSCALL can utilize these functions and routines without any additional actions required. This means that the writing of complex functions for the macro language does not necessarily require the use of complex macro code.

3. In current versions of SAS, single argument functions can now be called directly by a user defined format. This allows the creation of continuous value formats which are otherwise unavailable.

Using SAS to enhance data sharing across REDCap projects: Reducing errors, streamlining management, and improving quality

Presented By Rachel Myers

1. Use %LET macro statement to define the specific list of variables, records, or filters to be applied to the export API parameter sent to REDCap to streamline data exports.

2. Using a series of API export parameters, you can identify unique records to be imported into the destination REDCap database and avoid overwriting or duplicating existing records.

3. To allow you to successfully use the API import function, ensure that you have API import/update privileges assigned to your REDCap user permissions and that your destination database is not restricted to auto-numbering of records.