

#SASGF

The logo features the word "VIRTUAL" in a large, bold, white-outlined font. Each letter is filled with a colorful, abstract pattern of diagonal stripes in shades of blue, red, green, and purple. Below "VIRTUAL" is the text "SAS® GLOBAL FORUM 2021" in a clean, white, sans-serif font. The entire logo is centered on a dark blue horizontal band.

**VIRTUAL**  
SAS® GLOBAL FORUM 2021

# How SAS 9.x enriched your SAS programmer life?

Angelo Tinazzi, Cytel Inc.

Senior Director, Standards, Systems, CDISC Consulting Group



- Born in Milan, Italy
- Lived and Worked in Italy, UK, Switzerland and France
- Biostat by Education



- Working for Cytel in Geneva, Switzerland for 8 years
- >25 Years of Experience
- Pharma, CRO and Academic experience



- Living in France for about 10 years just the other side of the Swiss border
- 3 kids, 1 cat ... and 1 wife



- Passion for Clinical Data Standards
- Various Collaborations
- CDISC ADaM Trainer
- CDISC EU Committee
- PHUSE-EU Stream Chair
- Regular Speaker > 50

 <https://www.linkedin.com/in/angelotinazzi>

# SEUGI Madrid 1997 .... That was a long time ago

[https://support.sas.com/resources/papers/proceedings-archive/SEUGI1997/TINAZZI\\_TECSHOW.PDF](https://support.sas.com/resources/papers/proceedings-archive/SEUGI1997/TINAZZI_TECSHOW.PDF)



## **Do it yourself: tips and tricks in building your own application using SAS<sup>®</sup> macro language**

**Angelo Tinazzi**

Department of Oncology - Unit of Biometry -Laboratory of Cancer Clinical Epidemiology  
Mario Negri Institute Milan (Italy)

### **Abstract**

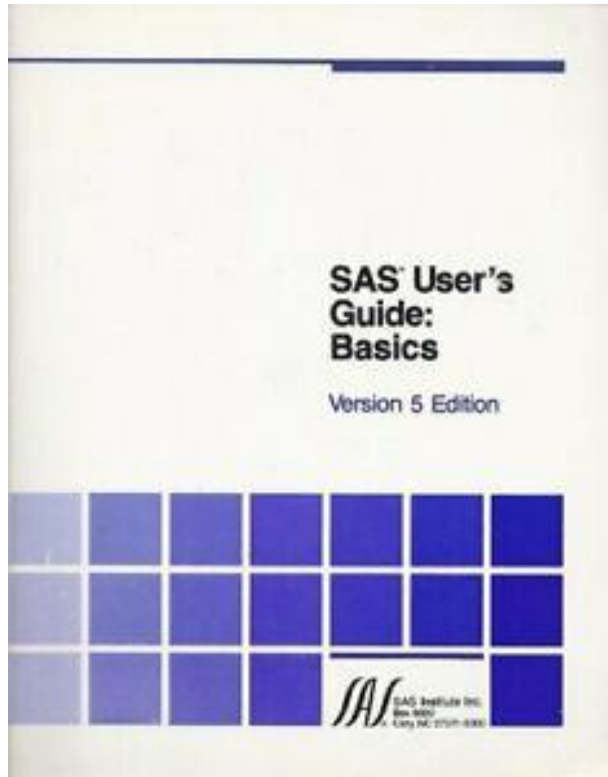
SAS macro language is probably one of the most used "features" of SAS System. The basic definition of SAS macro language could be "generator of SAS code". Its main characteristic is to give the possibility to develop ad-hoc application.

This paper is not intended to introduce SAS macro language, but the examples that reported could help also new users to better understand the "magnificent intrinsic language" of SAS System.

Some examples on how to use macro language will be presented and discussed; tips and tricks will be deeply analysed.

# SEUGI Madrid 1997 .... That was a long time ago

[https://support.sas.com/resources/papers/proceedings-archive/SEUGI1997/TINAZZI\\_TECSHOW.PDF](https://support.sas.com/resources/papers/proceedings-archive/SEUGI1997/TINAZZI_TECSHOW.PDF)



# AGENDA

## A bit of SAS History

SAS 9 is the longest-running SAS Version

How old is your SAS?

## Significant changes in SAS Language

My “Favorites”

A Quick Focus on

- HASH Objects

- Perl Regular Expression

## Significant changes in SAS Macro Language

%IF/%THEN in open code

New Comparison Operators

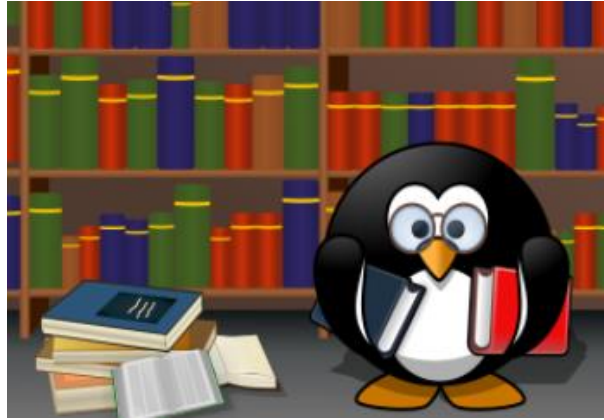
DOSUBL Function

New Alternatives to Macro, Programming Techniques and Cool Stuff

## Conclusions

References

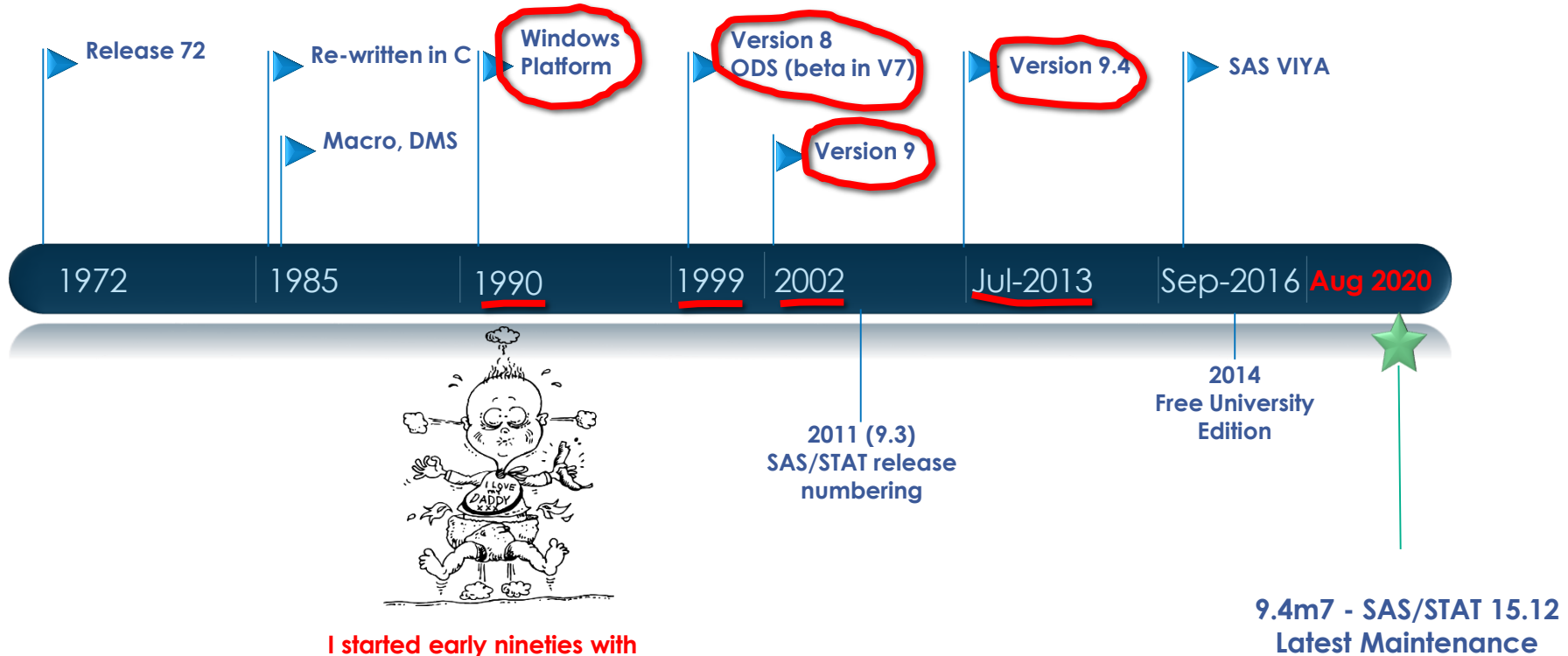
# My Assistant Librarian



**A Close Look at How DOSUBL Handles Macro Variable Scope**– Q. McMullen - SAS GLOBAL FORUM 2020  
**Submitting SAS® Code On The Side**– R. Langston; SUGI 2013

# A bit of SAS History

## SAS 9 is the longest-running SAS Version



I started early nineties with  
SAS 5.x in VMS OS

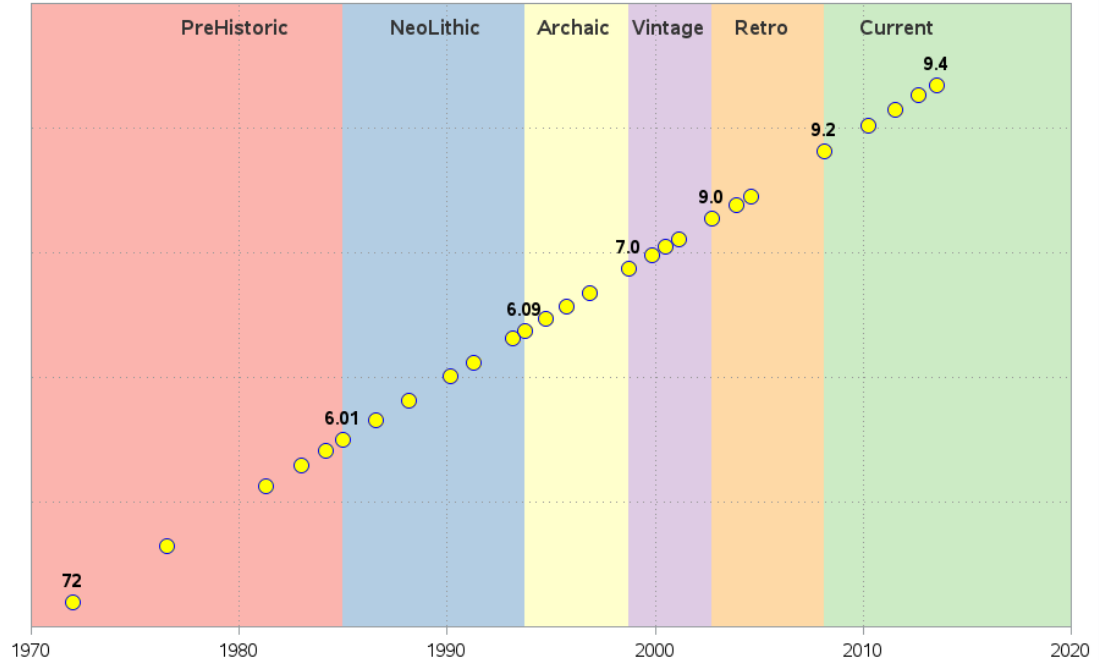
[https://en.wikipedia.org/wiki/SAS\\_\(software\)](https://en.wikipedia.org/wiki/SAS_(software))

# A bit of SAS History

## How old is your SAS?



### SAS Releases Timeline



<https://blogs.sas.com/content/iml/2013/08/02/how-old-is-your-version-of-sas-release-dates-for-sas-software.html>

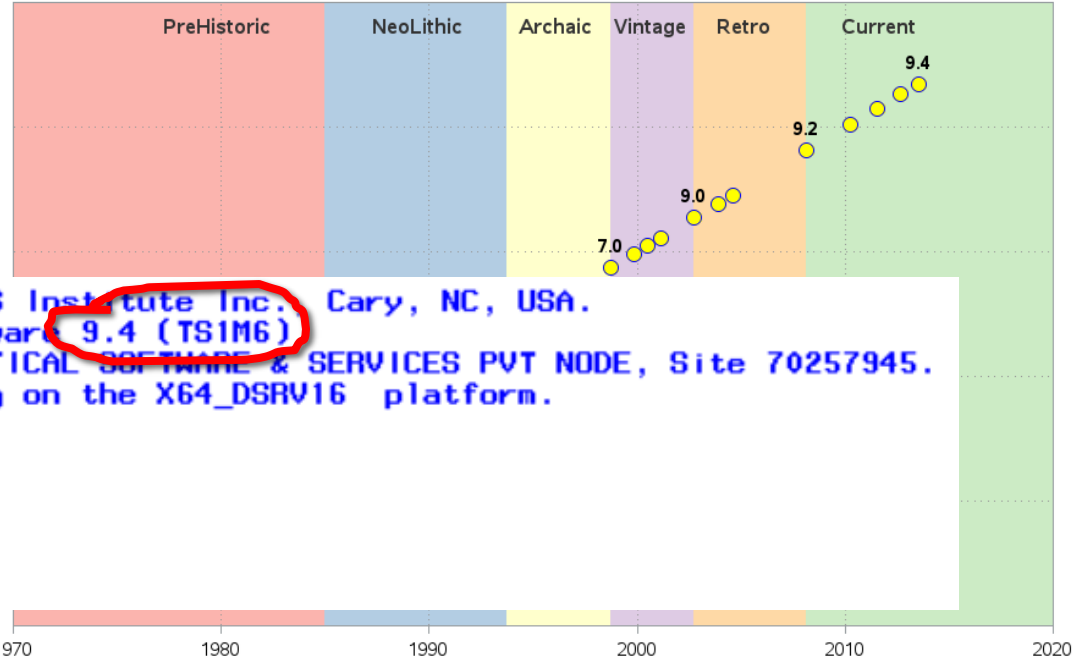


# A bit of SAS History

## How old is your SAS?



### SAS Releases Timeline



NOTE: Copyright (c) 2016 by SAS Institute Inc. Cary, NC, USA.  
NOTE: SAS (r) Proprietary Software 9.4 (TSIM6)  
Licensed to CYTEL STATISTICAL SOFTWARE & SERVICES PVT NODE, Site 70257945.  
NOTE: This session is executing on the X64\_DSRV16 platform.

NOTE: Analytical products:

SAS/STAT 15.1

<https://blogs.sas.com/content/iml/2013/08/02/how-old-is-your-version-of-sas-release-dates-for-sas-software.html>

# Significant changes in SAS Language

## My « Favorites » - New Functions

- New numeric / stats functions i.e. GEOMEAN, MEDIAN, LARGEST...
- ANY Functions i.e., ANYDIGIT
- CAT(s) Functions i.e., CATX
- COALESCE and COALESCEC (like in SQL)
- “V” Functions
- INDEX vs FIND vs COUNT(W)
- IFC(N), CHOOSEC(N) and WHICHC(N)

# Significant changes in SAS Language

## My « Favorites » - New Functions

### CAT(s) Functions i.e. CATX

#### Before

```
old = put(n,1.)||" "||trim(a)|| " "||  
      trim(b)|| " "||c;
```

#### With SAS 9.x

```
new = CATX(" ", n , a, b, c);
```



Let the CAT Out of the Bag: String Concatenation in SAS® 9 – J. Horstman – SCSUG 2016

# Significant changes in SAS Language

## My « Favorites » - New Functions

### “V” Functions

#### Create dynamic decode variables

```
data demo_decode;
  set demo;
  array ORIGvar (2) racen sexn;
  array DECODEvar (2) $ race sex;
  do i=1 to 2;
    DECODEvar(i)=putn(ORIGvar(i), VFORMAT(ORIGvar(i)));
  end;
run;
```

#### Before SAS 9.x

Get attributes from dictionary, macros, etc.



'V' for ... Variable Information Functions to the Rescue – R. Watson and K. Miller, MWSUG 2015

# Significant changes in SAS Language

## My « Favorites » - New Functions

### “V” Functions

#### Create dynamic decode variables

```
data demo_decode;
  set demo;
  array ORIGvar (2) race sex;
  array DECODEvar (2) _decode_;
  do i=1 to 2;
    DECODEvar(i) = VVALUE(ORIGvar(i));
  end;
run;
```

**Shorter Version (get variable value formatted as per assigned format)**

#### Before SAS 9.x

Get attributes from dictionary, macros, etc.



“V” for ... Variable Information Functions to the Rescue – R. Watson and K. Miller, MWSUG 2015

# Significant changes in SAS Language

## My « Favorites » - Functions Modifiers

For example COMPRESS function [SAS 9.1]

```
STRING="A comma is removed, a semicolon is also removed;";  
STRINGNEW=compress(STRING,, 'p');  
PUT "STRING:" @12 STRING;  
PUT "STRINGNEW:" @12 STRINGNEW;
```

-----LOG-----

```
STRING: A comma is removed, a semicolon is also removed;  
STRINGNEW: A comma is removed a semicolon is also removed
```

# Significant changes in SAS Language

My « Favorites » - Other « cool » improvements and options

- Some “cool” SET statement improvements [SAS 9.2]
  - `set supp:` → it sets all SDTM supplemental qualifier datasets
  - `set lb1-lb50` → it sets all 50 lb datasets datasets
- IN Operator Improvement i.e. `IN (2 3 5:10)` true when a value is either 2,3, 5,6,7,8,9,10
- New Formats and Informats [SAS 9.2]
  - ISO 8601 dates, time, and intervals i.e., `E8601DT.` and `E8601DA.`



ISO 8601 and SAS®: A Practical Approach – D. Morgan - PharmaSUG 2017

# Significant changes in SAS Language

## A quick Focus on HASH Objects [SAS 9.1]

- Powerful table lookup facility
- First in-memory data structure accessible from the DATA step
- Extracting a subset of Larger Dataset i.e. gradable lab tests ~20 Millions Records

CPU  
Time

SQL [~23 seconds]	MERGE [~36 seconds]	HASH [~14 seconds]
<pre>proc sql noprint;   create table SQL_ as   select *   from lb   where lbtestcd in   (select distinct   lbtestcd from LB   where lbtoxgr&gt;0); quit;</pre>	<pre>proc sql noprint;   create table grades as   select distinct lbtestcd   from LB(where=(lbtoxgr&gt;0)); quit; proc sort data=LB out=LB_sort;   by lbtestcd; run; data MERGE_;   merge LB_sort grades (in=ingr);   by lbtestcd;   if ingr; run;</pre>	<pre>data HASH_;   if _n_=1 then do;     dcl hash g(dataset:"LB(       where=(lbtoxgr&gt;0)");     g.definekey("lbtestcd");     g.definedone();   end;   set LB;   if g.find()=0; run;</pre>



Innovative Clinical Programming Methods - R. Allen; PharmaSUG-China 2019

Data Management Solutions Using SAS HASH Table Operations – P. Dorfman and Don Enderson; 2018 [Book]



# Significant changes in SAS Language

## A quick Focus on Perl Regular Expressions

- Powerful pattern search in text strings
- Compact solution
- You can get the position pattern, extract a substring, or substitute a string

Checking for character YYYY-MM-DD ISO format, and no partial date, prior to apply conversion to SAS date

```
if PRXMATCH ("/\d{4}-\d{2}-\d{2}/o", aestdtc)=1 then  
  asdt = input(substr(aestdtc,1,10), yymmdd10.);
```



Four steps to get a quick start with Perl Regular Expressions in SAS® – Q. Ni, P. Burmenko – PharmaSUG-China 2019  
Unstructured Data Analysis – K. Matthew Windham - 2018

# Significant changes in SAS Macro Language

## %IF/%THEN in open code [SAS 9.4M5]

```
/* Check if a dataset exist and if so take an action */  
%if %sysfunc(exist(work.adlb)) %then %do;  
    proc means data=work.adlb;  
    run;  
%end;  
%else %do;  
    %PUT WARNING: Missing WORK.ADLB - Please check;  
%end;
```

- Your %IF/%THEN **must be followed by a %DO/%END block**
- **No nesting** of multiple %IF/%THEN constructs

# Significant changes in SAS Macro Language

## New IN Comparison Operator

- Problem: I want to check if for a macro parameter &OPTIONS the user selected a valid option i.e. DEBUG, STORE, REPORT are the valid options

- Before SAS 9

```
%macro mymacro(option=) ;  
    %if &option eq DEBUG or &option eq STORE or  
        &option eq REPORT %then .....
```

- With SAS 9

```
%macro mymacro(option=) / minoperator mindelimiter=" , " ;  
    %if &option IN(DEBUG,STORE,REPORT) %then .....
```

# Significant changes in SAS Macro Language

## DOSUBL Function [SAS 9.3M2]

- Allows the execution of data steps and proc steps within another data step
- Enable immediate execution of SAS after a text string is passed
- It is an alternative to CALL EXECUTE
  - CALL EXECUTE generated code executes at the end of the dataset
  - DOSUBL code executes immediately
- Allows use of DATA and PROC step in **“function-style” macro**



A Close Look at How DOSUBL Handles Macro Variable Scope – Q. McMullen - SAS GLOBAL FORUM 2020  
Submitting SAS® Code On The Side – R. Langston; SUGI 2013

# Significant changes in SAS Macro Language

## DOSUBL Function [SAS 9.3M2] - Example Extracting Summary Statistics

```
/*Macro to Get Denominator*/
```

```
%macro getDen(in=,where=);
```

```
  %local rc den;
```

```
  %let den=0;
```

```
  %let rc=%sysfunc(dosubl(%nrstr(
```

```
    proc sql noprint;
```

```
      select count(*) into :den from &in where &where;
```

```
    quit;
```

```
  )))
```

```
  &den
```

```
%mend getden;
```

```
/*Macro to Get Numerator*/
```

```
%macro getCount(in=,where=);
```

```
  %local rc count;
```

```
  %let count=0;
```

```
  %let rc=%sysfunc(dosubl(%nrstr(
```

```
    proc sql noprint;
```

```
      select count(*) into :count from &in where &where;
```

```
    quit;
```

```
  )))
```

```
  &count
```

```
%mend getCount;
```

# Significant changes in SAS Macro Language

## DOSUBL Function [SAS 9.3M2] - Example Extracting Summary Statistics

```
/*Calculate % of Events stats by Arm Treatment and Study Population*/
```

```
data eventPop;
```

```
length pop $3 arm $10;
```

```
/*Event in ALL Pop, %*/
```

```
pop='ALL';
```

```
arm='Trt';
```

```
pct=( %getCount(in=adef,where=all eq 'Y' and arm eq 'Trt' and event='Dead') / %getDen(in=adsl,where=all eq 'Y' and arm eq 'Trt') ) * 100;
```

```
proc sql noprint;
```

```
select count(*) into :count
```

```
from adef
```

```
where all eq 'Y' and arm eq 'Trt' and  
event='Dead';
```

```
quit;
```

```
proc sql noprint;
```

```
select count(*) into :den
```

```
from adef
```

```
where all eq 'Y' and arm eq 'Trt';
```

```
quit;
```

```
%getDen(in=adsl,where=saf eq 'Y' and arm eq 'Placebo') ) * 100;
```

```
/*Event in EFF Pop, %*/
```

```
pop='EFF';
```

```
arm='Trt';
```

```
pct=( %getCount(in=adef,where=eff eq 'Y' and arm eq 'Trt' and event='Dead') / %getDen(in=adsl,where=eff eq 'Y' and arm eq 'Trt') ) * 100;
```

```
output;
```

```
arm='Placebo';
```

```
pct=( %getCount(in=adef,where=eff eq 'Y' and arm eq 'Placebo' and event='Dead') / %getDen(in=adsl,where=eff eq 'Y' and arm eq 'Placebo') ) * 100;
```

```
output;
```

```
format pct 8.1;
```

```
run;
```

# Significant changes in SAS Macro Language

## DOSUBL Function [SAS 9.3M2] - Example Extracting Summary Statistics

```
/*Calculate % of Events stats by Arm Treatment and Study Population*/
```

```
data eventPop;
```

```
length pop $3 arm $10;
```

```
/*Event in ALL Pop, %*/
```

```
pop='ALL';
```

```
arm='Trt';
```

```
pct=( %getCount(in=adeff,where=all eq 'Y' and arm eq 'Trt' and event='Dead') / %getDen(in=adsl,where=all eq 'Y' and arm eq 'Trt') ) * 100;
```

```
output;
```

```
arm='Placebo';
```

```
pct=( %getCount(in=adeff,where=all eq 'Y' and arm eq 'Plac
```

```
output;
```

```
/*Event in SAF Pop, %*/
```

```
pop='SAF';
```

```
arm='Trt';
```

```
pct=( %getCount(in=adeff,where=saf eq 'Y' and arm eq 'Trt'
```

```
output;
```

```
arm='Placebo';
```

```
pct=( %getCount(in=adeff,where=saf eq 'Y' and arm eq 'Plac
```

```
output;
```

```
/*Event in EFF Pop, %*/
```

```
pop='EFF';
```

```
arm='Trt';
```

```
pct=( %getCount(in=adeff,where=eff eq 'Y' and arm eq 'Trt'
```

```
output;
```

```
arm='Placebo';
```

```
pct=( %getCount(in=adeff,where=eff eq 'Y' and arm eq 'Plac
```

```
output;
```

```
format pct 8.1;
```

```
run;
```

	pop	arm	pct
1	ALL	Trt	38.8
2	ALL	Placebo	33.3
3	SAF	Trt	46.3
4	SAF	Placebo	31.6
5	EFF	Trt	38.8
6	EFF	Placebo	32.0

# Significant changes in SAS Macro Language

## New Alternatives to Macro Programming Techniques and Cool Stuff

- PROC FCMP and PROC PROTO [**SAS 9.2**] To create user functions/call routines

PROC FCMP or Function you Create, Master and Proceed – I. Boyko – PHUSE-EU 2020

- PROC GROOVY [**SAS 9.3**] Java syntax compatible object-oriented programming

A GROOVY way to enhance your SAS life – K. Kennedy – PHUSE-EU 2019

- PROCS DS2 [**SAS 9.4**] SAS proprietary object-oriented programming language

Expansion of Opportunities in Programming: DS2 Features and Examples of Usage Object Oriented Programming in SAS® – S. Voievutkyi – PharmaSUG 2017

- PROC LUA [**SAS 9.4**] Lua language

Driving SAS® with Lua – P. Tomas – SAS GLOBAL FORUM 2015



# Conclusions

## References

- **'V' for ... Variable Information Functions to the Rescue** – R. Watson and K. Miller; MWSUG 2015
- **SAS Functions by Examples** – R. Cody – Second Edition 2010 [Book]
- **ISO 8601 and SAS®: A Practical Approach** – D. Morgan; PharmaSUG 2017
- **Let the CAT Out of the Bag: String Concatenation in SAS® 9** – J. Horstman; SCSUG 2016
- **Fifteen Functions to Supercharge Your SAS® Code** - J. Horstman; WUSS 2017
- **An Introduction to Perl Regular Expressions in SAS 9** – R. Cody and R. Wood; SUGI 29
- **Four steps to get a quick start with Perl Regular Expressions in SAS®** – Q. Ni, P. Burmenko; PharmaSUG-China 2019
- **Unstructured Data Analysis** – K. Matthew Windham - 2018
- **Innovative Clinical Programming Methods** - R. Allen; PharmaSUG-China 2019
- **Comparing 6 Techniques To Do Data Driven Programming** – J. Derks – PHUSE 2017
- **Efficient Coding Techniques In SAS** – G. Kesireddi; PHUSE 2017
- **Modernizing Legacy SAS® Applications and Program Code** KP. Lafler and C. Roberts; SCSUG 2017
- **Data Management Solutions Using SAS HASH Table Operations** – P. Dorfman and Don Enderson; 2018 [Book]
- **Submitting SAS® Code On The Side** – R. Langston; SUGI 2013
- **A Close Look at How DOSUBL Handles Macro Variable Scope** – Q. McMullen - SAS GLOBAL FORUM 2020
- **PROC FCMP or Function you Create, Master and Proceed** – I. Boyko – PHUSE 2020
- **A GROOVY way to enhance your SAS life** – K. Kennedy – PHUSE 2019
- **Expansion of Opportunities in Programming: DS2 Features and Examples of Usage Object Oriented Programming in SAS®** – S. Voievutkyi – PharmaSUG 2017
- **Driving SAS® with Lua** – P. Tomas – SAS GLOBAL FORUM 2015



# Thank you!

Contact Information  
[angelo.tinazzi@cytel.com](mailto:angelo.tinazzi@cytel.com)