

PHUSE EU Connect 2020

9-13 November 2020

Paper CT02 – Coding Tips and Tricks (CT)

Mind the Gap: Make Sure you are Upskilled with SAS 9.x

Angelo Tinazzi, Cytel Inc.
Geneva – Switzerland
angelo.tinazzi@cytel.com

➤ A bit of SAS History

SAS 9 is the longest-running SAS Version

How old is your version of SAS Release?

➤ 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

New Alternatives to Macro, Programming Techniques and Cool Stuff

➤ Conclusions

References

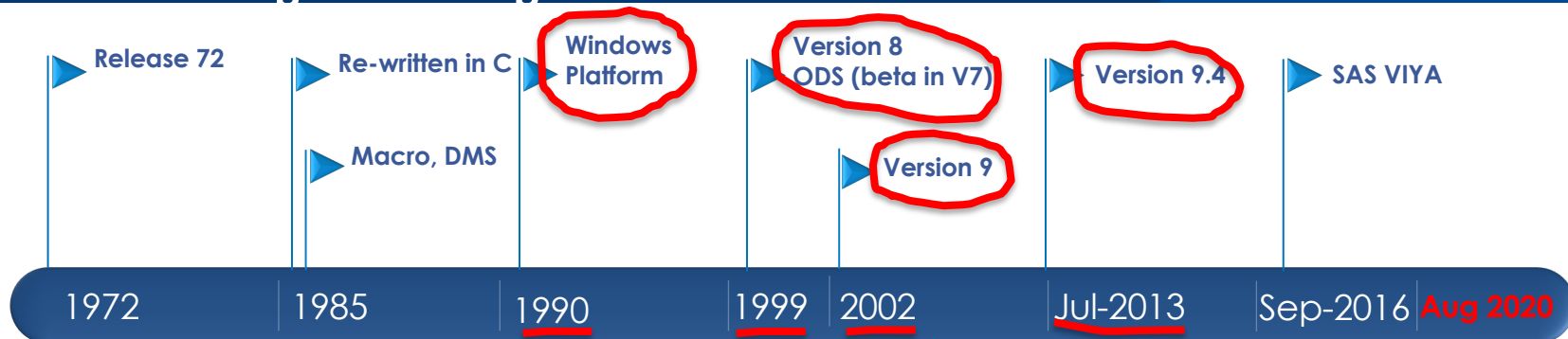
AGENDA

A bit of SAS History

SAS 9 is the longest-running SAS Version

How old is your Version of SAS Release?

SAS 9 is the longest-running SAS Version



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

2011 (9.3)
SAS/STAT release
numbering

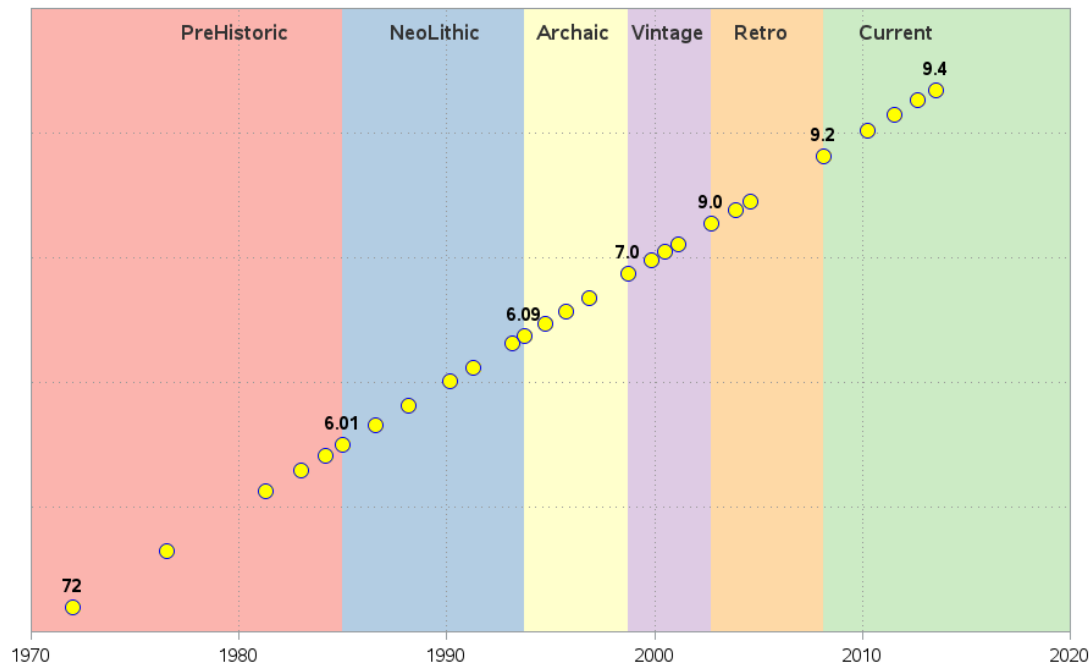
2014
Free University
Edition

9.4m7 - SAS/STAT 15.12
Latest Maintenance

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



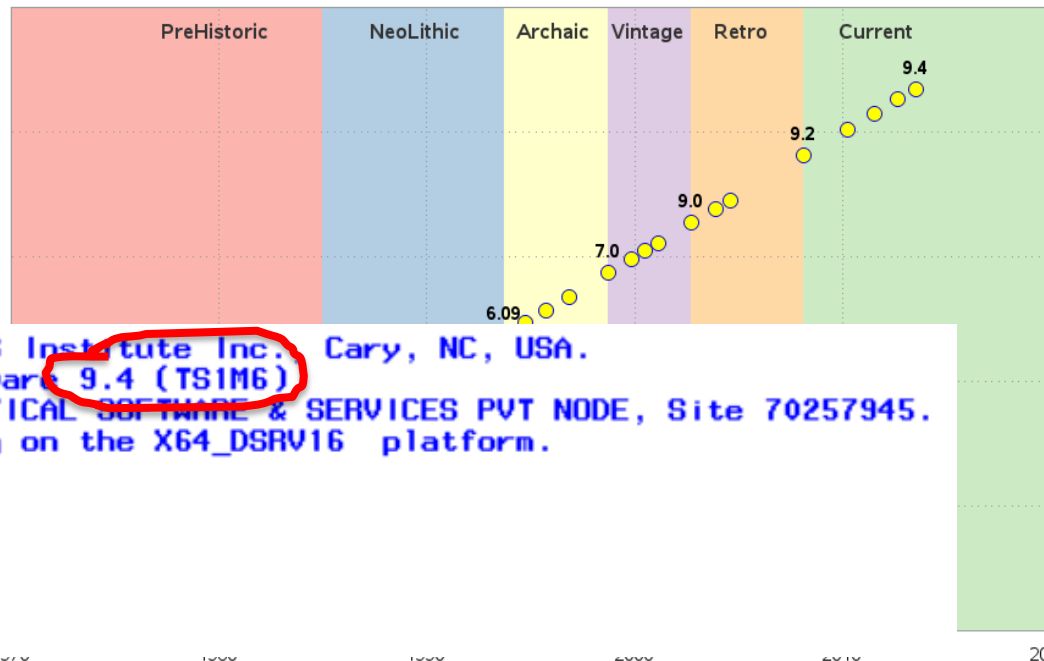
SAS Releases Timeline



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



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”

A Quick Focus on:

- HASH Objects
- Perl Regular Expression



Use at your own risk

- 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 and IFN

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);
```



“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” Functions

Create dynamic decode variables

```
data demo_decode;  
  set demo;  
  array ORIGvar    (2)    racen sexn;  
  array DECODEvar  
do i=1 to  
  DECODEvar  
end;  
run;
```

**Shorter Version (get variable value
formatted as per assigned format
DECODEvar(i)=VVALUE(ORIGvar(i));**

Before SAS 9.x

Get attributes from dictionary, macros, etc.



For example COMPRESS function [SAS 9.1]

```
STRINGNEW=compress (STRING, , 'p' ) ;
```

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

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.`





- 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>0); quit;</pre>	<pre>proc sql noprint; create table grades as select distinct lbtestcd from LB(where=(lbtoxgr>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>0))"); g.definekey("lbtestcd"); g.definedone(); end; set LB; if g.find()=0; run;</pre>





- 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

```
SC if _n_=1 then do;  
pr      dcl hash g(dataset:"LB(where=(lbtoxgr>0))");  
      g.definekey("lbtestcd");  
      g.definedone();  
end;  
qu      set LB;  
      if g.find()=0;  
run;
```





- 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.);
```



Significant changes in SAS Macro Language

%IF/%THEN in open code

New IN Comparison Operator

New Alternatives to Macro, Programming Techniques and Cool Stuff

```
/* 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

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 .....
```

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

EU Connect 2020

SM08 - PROC FCMP or Function you Create, Master and Proceed – I. Boyko

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

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

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

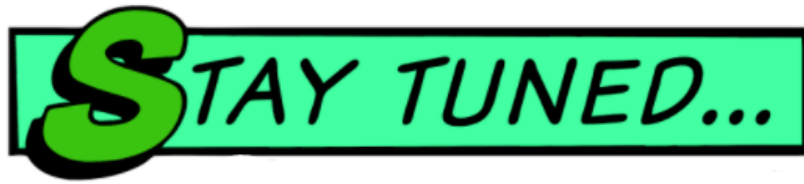
Expansion of Opportunities in Programming: DS2 Features and Examples of Usage Object Oriented Programming in SAS® – S. Voievutkyi – PharmaSUG 2017

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

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

Conclusions

References



PUUSE EU CONNECT 2020

Paper CT02

Mind the Gap: Make Sure you are Upskilled with SAS 9.x

Angelo Tinazzi, Cytel Inc., Geneva, Switzerland

ABSTRACT

SAS version 9 is the longest-running version as it was first released in 2002, with its last technical maintenance released at the end of 2018 (9.4 M6). With this presentation, I will go through some of the main SAS/BASE language enrichment you might have missed especially if you, like me, are a SAS programmer that started his or her career when SAS version 6, the first PC Dos version, was up and running. For example, the family of "V" functions, the hash objects, new functions to handle characters variables, new data steps options and new macro features. The focus will be on improvements that made mine, and every SAS programmer, a better life.

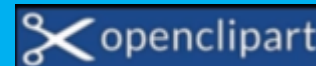
INTRODUCTION

.....
.....
.....
.....
.....
.....
.....
.....

- **'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
- **Innovative Clinical Programming Methods** - R. Allen; PharmaSUG-China 2019
- **Comparing 6 Techniques To Do Data Driven Programming** – J. Derks – PHUSE 2017
- **Submitting SAS® Code On The Side** – R. Langston; SUGI 2013
- **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]
- **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



Clipart sourced at openclipart.org



Angelo Tinazzi, Senior Director
Cytel Inc.
Standards, Systems, CDISC Consulting, Statistical Programming
Clinical Research Services
Route de Pré-Bois 20
C.P 1839, 1215 Geneva, SWITZERLAND
Email: angelo.tinazzi@cytel.com