SAS Summer Drive-In

Macros from the Wild: What we need to know to survive

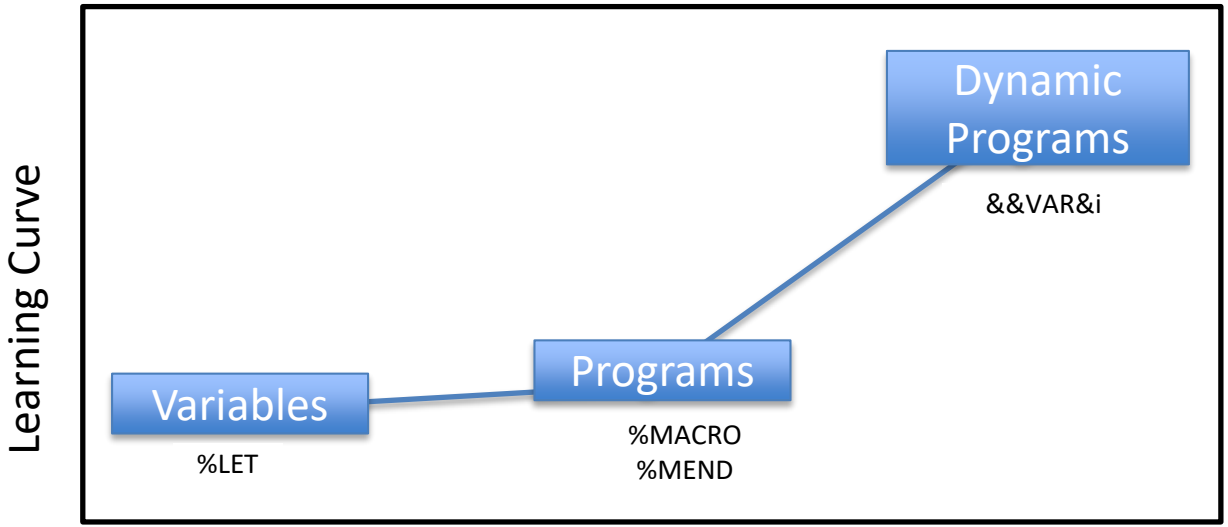
Andrew Walker

26JUL2022

Goals

Give a preview of what is possible with macros and provide resources for additional investigating

Natural Learning Progression of Macros

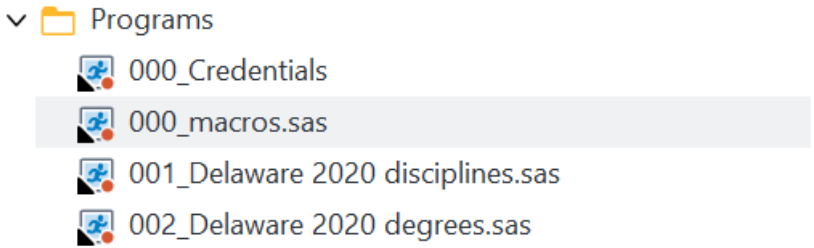


Overview of Patterns

1) %LET = Find and Replace
    1) Manual
    2) Self-defining
    3) Calculating
    4) Functions
2) Wide Lists SQL :INTO
3) Long Lists CALL SYMPUTX
4) Loop Through Lists &&Var&i
5) Document our Macros

Situation – I'm asked to run a program and update the required filters etc.

❶%LET Name = Andrew;




```
3  %let term = 20216;
4  %let termFA = &Term; /*Fall 2020*/
5  %let termSP = %SYSEVALF(%SYSFUNC(SUBSTR(&TERM,1,4))+1)1; *Spring Terms;
6  %PUT &TERMSP;
7  %let termfy1 = 20216; /*Fall 2021*/
8  %let termfy2 = &termSP; /*Spring 2022*/
9  %let termtypefy = 1; /*Census*/
```

%LET = Replace Text

❶%LET Name = Andrew;

When I call “&NAME” it resolves to “Andrew”

Ex.

```
Data new;
      set old;
Where name = ❷ “&NAME”;
Run;
```

```
Data new;
      set old;
Where name = ❷ “Andrew”;
Run;
```

Enhance %LET with %SYSEVAL

❶%LET ReportingTerm=2022;

❷%LET Past1Term = 2021;
❸%LET Past2Term = 2020;
❹%LET Past3Term = 2019;

&Past1Term resolves to 2021
&Past2Term resolves to 2020.

We always want to reduce the opportunities for errors.

Enhance %LET with %EVAL

❶%LET ReportingTerm=2022;

❷%LET Past1Term = %EVAL(&ReportingTerm -1);
❸%LET Past2Term = %EVAL(&ReportingTerm -2);
❹%LET Past3Term = %EVAL(&ReportingTerm -3);

&Past1Term resolves to 2021
&Past2Term resolves to 2020.

%EVAL lets us immediately evaluate numeric calculations.

%EVAL and introduce %SYSFUNC?

```
%LET ReportingTerm=2022;
%LET PastTerm1=2021;
%LET PastTerm2=2020;
%LET PastTerm3=2019;
```

```
%LET PastTerm2= %EVAL(&ReportingTerm-1);
%LET PastTerm3= %EVAL(&ReportingTerm-2);
%LET PastTerm4= %EVAL(&ReportingTerm-3);
```

%LET with %SYSEVAL and %SYSFUNC

❶ %LET ReportingTerm=**2022**;
❷ %LET PastTerm = %EVAL(&ReportingTerm -1);

Updated Example
❸ %LET ReportingTerm=**2020**FA;
❸ %LET PastTerm = %EVAL(%SYSFUNC(❹SUBSTR(&ReportingTerm,1,4) ❹) -1)FA;

%SYSFUNC lets us use other SAS functions inside the macro facility – we don't need the data step to execute functions.

I want a report on the past five years.

```
%MACRO Past;
❶%LET ReportingTerm=%EVAL(%SYSFUNC(YEAR(%SYSFUNC(Today()))))-2);
    ❷%DO a=1 %to 6;
    ❸%GLOBAL PastFATerm&a PastSPTerm&a PastSU1Term&a PastSU2Term&a;
        ❹%LET PastFATerm&a=%SYSEVALF(&ReportingTerm-&a.)FA;
        ❹%LET PastSPTerm&a=%SYSEVALF(&ReportingTerm-&a.)SP;
        ❹%LET PastSU1Term&a=%SYSEVALF(&ReportingTerm-&a.)SU1;
        ❹%LET PastSU2Term&a=%SYSEVALF(&ReportingTerm-&a.)SU2;
    ❺%END;
%MEND Past;

❼%Past;
```

Can I see what I just made?

Macro Variable Name	Macro Variable Value
PASTFATERM1	2019FA
PASTFATERM2	2018FA
PASTFATERM3	2017FA
PASTFATERM4	2016FA
PASTFATERM5	2015FA
PASTFATERM6	2014FA
PASTSPTERM1	2019SP
PASTSPTERM2	2018SP
PASTSPTERM3	2017SP
PASTSPTERM4	2016SP
PASTSPTERM5	2015SP
PASTSPTERM6	2014SP
PASTSU1TERM1	2019SU1
REPORTINGTERM	2020

```
17  proc sql ;
18       Select name, value
19       from dictionary.macros
20                         where scope='GLOBAL'
21                   and not name contains 'SYS_SQL_IP_' and
22                    FIND(name,'SQL')=0  and
23                    FIND(name,'SAS')=0  and
24                    FIND(name,'SYS')=0  and
25                    SUBSTR(name,1,1) NE '_';
26       quit;
```

I want to create a macro variable based on data from a data set.

```
PROC SQL Noprint;
Select Distinct Acad_Term
❶Into :CUTerm
From CU;
Quit;

%PUT &=CUTerm;
```

Creates a macro called **&CUTerm** from the value Acad_Term table CU.

Title “This is the Title I Never Need to Update For - &CUTerm”

Creating Lists with PROC SQL :INTO

```
PROC SQL Noprint;
Select Distinct Acad_Term
❶Into: CUTerm Separated by ", "
From CU;
Quit;

%PUT &=CUTerm;
```

Creates a macro called &CUTerm from the value Acad_Term table CU.

Title "This is the Title I Never Need to Update For - &CUTERM"

Creating Lists with PROC SQL :INTO

```
PROC SQL Noprint;
Select Distinct QUOTE(COMPRESS(Acad_Term))
as Acad_Term
Into: CUTerm Separated by ", "
From CU;
Quit;
```

Acad_Term
2018FA
2019SP
2019SU1
2019SU2

```
%PUT &=CUTerm;
```
"2018FA", "2019SP", "2019SU1", "2019SU2"

Creates a macro called &CUTerm from the value Acad_Term table CU.

Title "This is the Title I Never Need to Update For - &CUTERM"

Creating Lists with PROC SQL :INTO

```
PROC SQL Noprint;
❷ Select Distinct QUOTE(COMPRESS(Acad_Term)) as Acad_Term,
❶ Count(Distinct Student_ID) as Stu_Count
❹Into:CUTerm Separated by ", " ❺:Stu_Count
From student_data;
Quit;
```

Creates a macro called &CUTerm from the value Acad_Term table CU.
Creates a macro called &Stu_Count from the distinct count of students from CU.

Title "This is the Title I Never Need to Update For - &CUTERM with &Stu_Count"

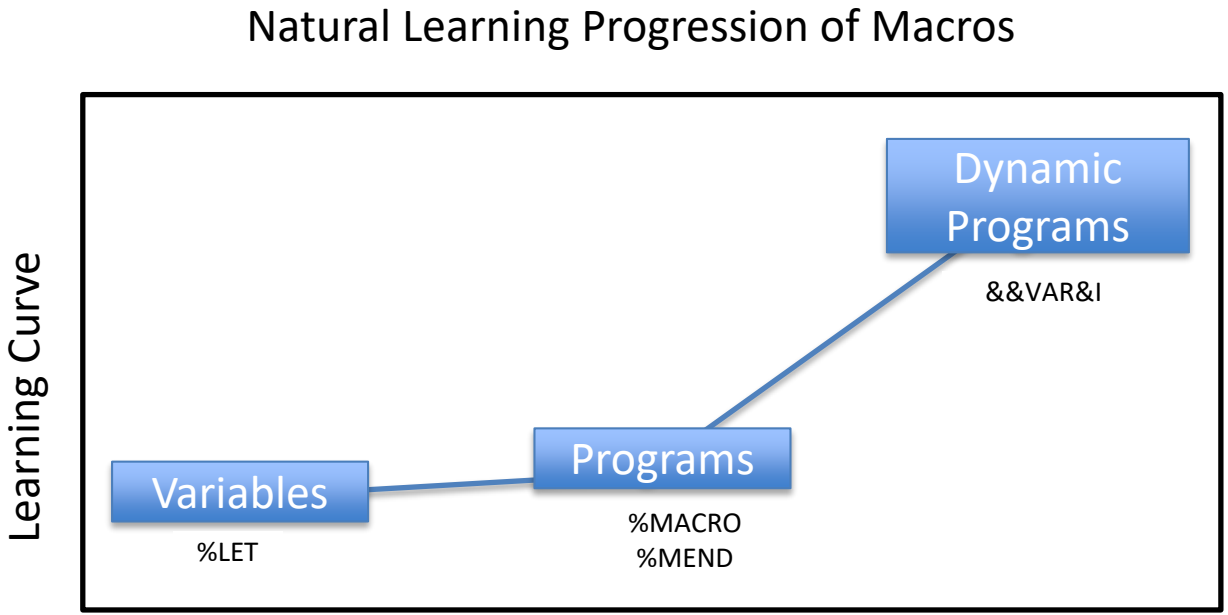
I want to create a list of self-created global macro variables!

```
17  proc sql ;
18       Select name, value
19       INTO :Name separated by ', ', :Val separated by ', '
20       from dictionary.macros
21                         where scope='GLOBAL'
22                   and not name contains 'SYS_SQL_IP_' and
23                    FIND(name,'SQL')=0  and
24                    FIND(name,'SAS')=0  and
25                    FIND(name,'SYS')=0  and
26                    SUBSTR(name,1,1) NE '_';
27       quit;
28
29  %PUT &NAME;
```

```
32  A, PASTFATERM, PASTFATERM1, PASTFATERM2, PASTFATERM3, PASTFATERM4, PASTFATERM5, PASTFATERM6, PASTSPTERM, PASTSPTERM1, PASTSPTERM2,
33  PASTSPTERM3, PASTSPTERM4, PASTSPTERM5, PASTSPTERM6, PASTSU1TERM, PASTSU1TERM1, PASTSU1TERM2, PASTSU1TERM3, PASTSU1TERM4,
34  PASTSU1TERM5, PASTSU1TERM6, PASTSU2TERM, PASTSU2TERM1, PASTSU2TERM2, PASTSU2TERM3, PASTSU2TERM4, PASTSU2TERM5, PASTSU2TERM6,
35  REPORTINGTERM
```

Goals

Give a preview of what is possible with macros and provide resources for additional investigating



I want to run the same chunk of code, over and over, and define a variable.

```
34 %Macro Import(Y=);❶
35 libname Ex&Y ❷"S:\My Excel File-&y ❷xlsx";
36
37 Data SCH&Y label&y;❷
38     set ex&Y.❷ &Y ❷ata$"n;
39     TwoDigit = SCAN(ciporig ,1,'.');
40     Year=&Y.;❷
41 Run;
42
43 libname ex&y clear;
44
45 %MEND Import;
46
48 %Import(Y=2016);
49 %Import(Y=2017);
50 %Import(Y=2018);
51 %Import(Y=2019);
52 %Import(Y=2020);
53 %Import(Y=2021);
54 %Import(Y=2022);
```

Steps

1) Start with your model
    1) Get it to run once hard-coded,
    2) Then replace all the values with a macro call/variable.
2) Wrap it in a name
3) Define your variable
4) Call the macro

Step 1 Create your Model

```
57  libname Ex2016. "S:\My Excel File-2016.xlsx";
58
59  Data SCH2016;
60      set ex2016."2016 data$"n;
61      Year=2016;
62  Run;
63
64  libname ex2016 clear;
```

```
66  %MACRO IMPORT(Y=);
67  libname Ex&Y. "S:\My Excel File-2016.xlsx";
68
69  Data SCH&Y;
70      set ex&Y."&Y data$"n;
71      Year=&Y;
72  Run;
73
74  libname ex&Y clear;
75
76  %MEND IMPORT;
```

Long Lists of Macros – CALL SYMPUTX

I want a separate PDF by _____ and I don't want to manually define each report.

I want the data to determine the file name, filter, and titles.

Long Lists of Macros – CALL SYMPUTX

```
13 Data Terms;
14  Infile Datalines;
15  Input Acad_Term :$7.;
16  Datalines;
17  2018FA
18  2019SP
19  2019SU1
20  2019SU2
21  ;
```

I want a macro for each term so I can automatically loop through each one and make four different reports.

Steps

Find the Terms table

Use CALL SYMPUTX to assign a macro to each value

Long Lists of Macros – SYMPUTX

Acad_Term
2018FA
2019SP
2019SU1
2019SU2

```
Data _NULL_;
    ❶Set Terms End=no_more;
    ❷Call symputx(CATT("Terms",_n_),Acad_Term);
    ❸If no_more THEN CALL symputx('NUMROWS',_n_);
Run;
```

Long Lists of Macros – SYMPUTX

```
Data _NULL_;
    Set Terms End=no_more;
    Call symputx(CATT("Terms",_n_),Acad_Term);
    If no_more THEN CALL
symputx('NUMROWS',_n_);
Run;
```

Macro Variable Name	Macro Variable Value
NUMROWS	4
TERMS1	2018FA
TERMS2	2019SP
TERMS3	2019SU1
TERMS4	2019SU2

Long Lists of Macros – SYMPUTX

```
13 Data Terms;
14  Infile Datalines delimiter=',';
15  Input Acad_Term :$7. Acad_year :$7.;
16  Datalines;
17  2018FA,2018-19
18  2019SP,2018-19
19  2019SU1,2018-19
20  2019SU2,2018-19
21  ;
22
```

```
29 Data _NULL_;
30      Set Terms End=no_more;
31      Call symputx(CATT("Terms",_n_),Acad_Term);
32      Call symputx(CATT("Year",_n_),Acad_Year);
33      If no_more THEN CALL symputx('NUMROWS',_n_);
34  Run;
35
```

Long Lists of Macros – SYMPUTX

```
13 Data Terms;
14  Infile Datalines delimiter=',';
15  Input Acad_Term :$7. Acad_year :$7.;
16  Datalines;
17  2018FA,2018-19
18  2019SP,2018-19
19  2019SU1,2018-19
20  2019SU2,2018-19
21  ;
```

Macro Variable Name	Macro Variable Value
NUMROWS	4
TERMS1	2018FA
TERMS2	2019SP
TERMS3	2019SU1
TERMS4	2019SU2

```
%Macro MyMacro;
%DO a=1 %to &NUMROWS;
ODS PDF
File="C:\Outcomes_&&Term&a...PDF";
Title1 "&&Acad_Year&a";

... My reports...;

%MEND Program5;

% MyMacro
```

Overall Pattern

Overall Pattern

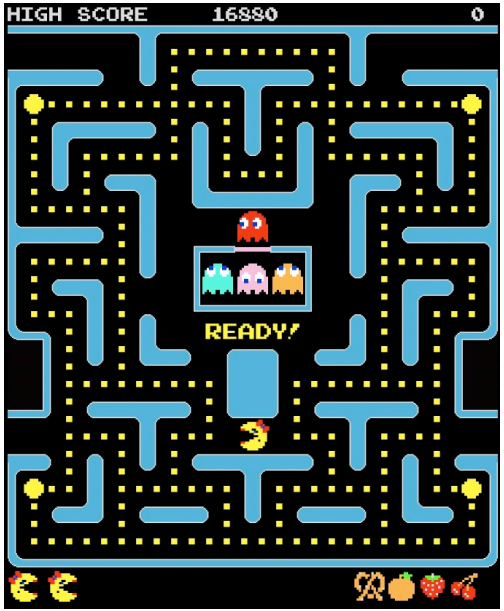
```
%DO a=1 %to &NUMROWS;

&&Term&a

%End
```

Overall Pattern

```
%DO a=1 %to &NUMROWS;
&&Term&a
%End
```

The SAS Complier can only “eat” one “&” at a time.

Think of PacMan



Overall Pattern

```
%DO a=1 %to &NUMROWS;
&&Term&a
%End
```

The SAS Complier can only “eat” one “&” at a time.

First Loop

❶&&Term&a = ❷&TERM1 = ❸2018FA

Macro Variable Name	Macro Variable Value
NUMROWS	4
TERMS1	2018FA
TERMS2	2019SP
TERMS3	2019SU1
TERMS4	2019SU2

I don’t want to update predictable if statements!

```
150  if datepart(DEGR_CONFER_DT) > '30JUN2009'd and datepart(DEGR_CONFER_DT) < '01JUL2010'd then IPEDS_year='2009-10';
151  if datepart(DEGR_CONFER_DT) > '30JUN2010'd and datepart(DEGR_CONFER_DT) < '01JUL2011'd then IPEDS_year='2010-11';
152  if datepart(DEGR_CONFER_DT) > '30JUN2011'd and datepart(DEGR_CONFER_DT) < '01JUL2012'd then IPEDS_year='2011-12';
153  if datepart(DEGR_CONFER_DT) > '30JUN2012'd and datepart(DEGR_CONFER_DT) < '01JUL2013'd then IPEDS_year='2012-13';
154  if datepart(DEGR_CONFER_DT) > '30JUN2013'd and datepart(DEGR_CONFER_DT) < '01JUL2014'd then IPEDS_year='2013-14';
155  if datepart(DEGR_CONFER_DT) > '30JUN2014'd and datepart(DEGR_CONFER_DT) < '01JUL2015'd then IPEDS_year='2014-15';
156  if datepart(DEGR_CONFER_DT) > '30JUN2015'd and datepart(DEGR_CONFER_DT) < '01JUL2016'd then IPEDS_year='2015-16';
157  if datepart(DEGR_CONFER_DT) > '30JUN2016'd and datepart(DEGR_CONFER_DT) < '01JUL2017'd then IPEDS_year='2016-17';
158  if datepart(DEGR_CONFER_DT) > '30JUN2017'd and datepart(DEGR_CONFER_DT) < '01JUL2018'd then IPEDS_year='2017-18';
159  if datepart(DEGR_CONFER_DT) > '30JUN2018'd and datepart(DEGR_CONFER_DT) < '01JUL2019'd then IPEDS_year='2018-19';
160  if datepart(DEGR_CONFER_DT) > '30JUN2019'd and datepart(DEGR_CONFER_DT) < '01JUL2020'd then IPEDS_year='2019-20';
161  if datepart(DEGR_CONFER_DT) > '30JUN2020'd and datepart(DEGR_CONFER_DT) < '01JUL2021'd then IPEDS_year='2020-21';
162  if datepart(DEGR_CONFER_DT) > '30JUN2021'd and datepart(DEGR_CONFER_DT) < '01JUL2022'd then IPEDS_year='2021-22';
163  if datepart(DEGR_CONFER_DT) > '30JUN2022'd and datepart(DEGR_CONFER_DT) < '01JUL2023'd then IPEDS_year='2022-23';
```

It all comes together

```
66 Data Datesz(Drop=i);
67      Do i=1990 to 2049;
68          Year1_IPEDS=CATS("'",'1JUL',PUT(i,8.),"'",'d');
69          Year2_IPEDS=CATS("'",'30JUN',PUT((i+1),8.),"'",'d');
70          Year1_Acad=CATS("'",'01SEP',PUT(i,8.),"'",'d');
71          Year2_Acad=CATS("'",'31AUG',PUT((i+1),8.),"'",'d');
72          Year1_Cal=CATS("'",'01JAN',PUT(i,8.),"'",'d');
73          Year2_Cal=CATS("'",'31DEC',PUT((i+1),8.),"'",'d');
74          Yearz = CATS(PUT(i,8.),'-',SUBSTR(COMPRESS(PUT((i+1),8.)),3,2));
75          Output;
76  End;
77  Run;
```

Data

	1		2		3		
	Year1_IPEDS	Year2_IPEDS	Year1_Acad	Year2_Acad	Year1_Cal	Year2_Cal	Yearz
1	'1JUL1990'd	'30JUN1991'd	'01SEP1990'd	'31AUG1991'd	'01JAN1990'd	'31DEC1991'd	1990-91
2	'1JUL1991'd	'30JUN1992'd	'01SEP1991'd	'31AUG1992'd	'01JAN1991'd	'31DEC1992'd	1991-92
3	'1JUL1992'd	'30JUN1993'd	'01SEP1992'd	'31AUG1993'd	'01JAN1992'd	'31DEC1993'd	1992-93
4	'1JUL1993'd	'30JUN1994'd	'01SEP1993'd	'31AUG1994'd	'01JAN1993'd	'31DEC1994'd	1993-94
5	'1JUL1994'd	'30JUN1995'd	'01SEP1994'd	'31AUG1995'd	'01JAN1994'd	'31DEC1995'd	1994-95
6	'1JUL1995'd	'30JUN1996'd	'01SEP1995'd	'31AUG1996'd	'01JAN1995'd	'31DEC1996'd	1995-96
7	'1JUL1996'd	'30JUN1997'd	'01SEP1996'd	'31AUG1997'd	'01JAN1996'd	'31DEC1997'd	1996-97
8	'1JUL1997'd	'30JUN1998'd	'01SEP1997'd	'31AUG1998'd	'01JAN1997'd	'31DEC1998'd	1997-98
9	'1JUL1998'd	'30JUN1999'd	'01SEP1998'd	'31AUG1999'd	'01JAN1998'd	'31DEC1999'd	1998-99
10	'1JUL1999'd	'30JUN2000'd	'01SEP1999'd	'31AUG2000'd	'01JAN1999'd	'31DEC2000'd	1999-00
11	'1JUL2000'd	'30JUN2001'd	'01SEP2000'd	'31AUG2001'd	'01JAN2000'd	'31DEC2001'd	2000-01
12	'1JUL2001'd	'30JUN2002'd	'01SEP2001'd	'31AUG2002'd	'01JAN2001'd	'31DEC2002'd	2001-02
13	'1JUL2002'd	'30JUN2003'd	'01SEP2002'd	'31AUG2003'd	'01JAN2002'd	'31DEC2003'd	2002-03
14	'1JUL2003'd	'30JUN2004'd	'01SEP2003'd	'31AUG2004'd	'01JAN2003'd	'31DEC2004'd	2003-04
15	'1JUL2004'd	'30JUN2005'd	'01SEP2004'd	'31AUG2005'd	'01JAN2004'd	'31DEC2005'd	2004-05
16	'1JUL2005'd	'30JUN2006'd	'01SEP2005'd	'31AUG2006'd	'01JAN2005'd	'31DEC2006'd	2005-06

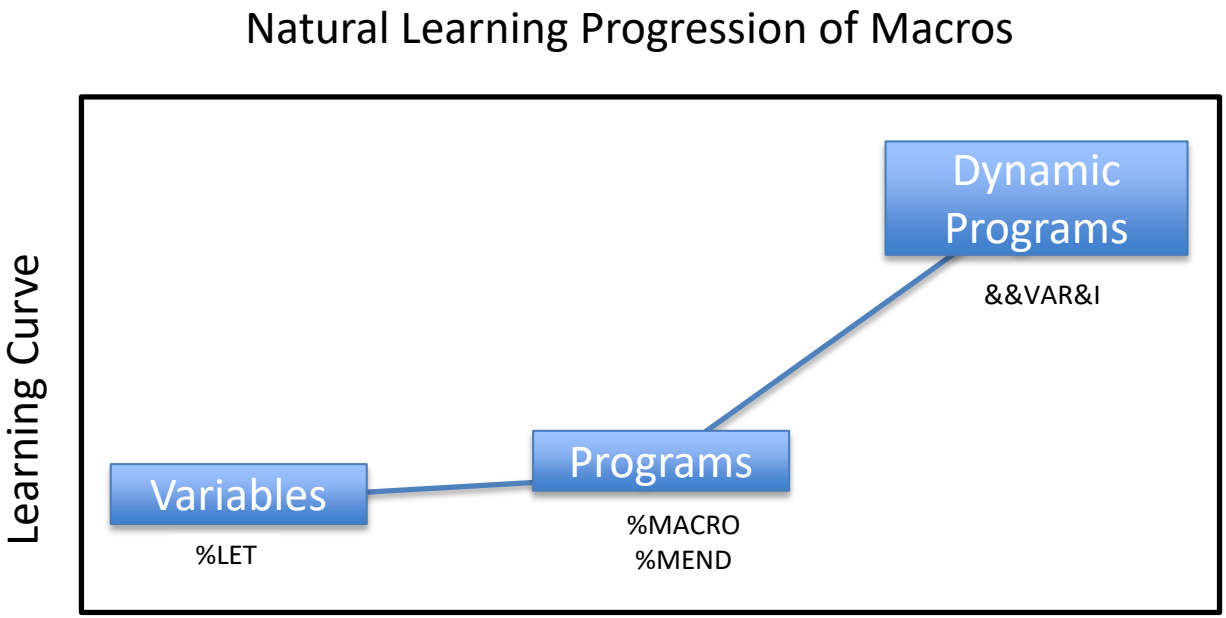
Create Macro Program

```
❶%MACRO D3;
      data degrees_2009_4;
      set degrees_2009_3;
            ❷%Do a=1 %to &Numrows;
            ❸if datepart(DEGR_CONFER_DT) > &&Year1_IPEDS&a and
                  datepart(DEGR_CONFER_DT) < &&Year2_IPEDS&a then
                  IPEDS_year="&&IPEDS_Year&a";
                        %End;
      Run;

❶%MEND D3;

%D3;
```

Code Generated – OPTIONS MPRINT;

```
183  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01SEP2031'd and datepart(DEGR_CONFER_DT) < '31AUG2032'd then acad_Year="2031-32";
184  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01JAN2031'd and datepart(DEGR_CONFER_DT) < '31DEC2032'd then cal_year="2031-32";
185  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '30JUN2032'd and datepart(DEGR_CONFER_DT) < '01JUL2033'd then IPEDS_year="2032-33";
186  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01SEP2032'd and datepart(DEGR_CONFER_DT) < '31AUG2033'd then acad_Year="2032-33";
187  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01JAN2032'd and datepart(DEGR_CONFER_DT) < '31DEC2033'd then cal_year="2032-33";
188  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '30JUN2033'd and datepart(DEGR_CONFER_DT) < '01JUL2034'd then IPEDS_year="2033-34";
189  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01SEP2033'd and datepart(DEGR_CONFER_DT) < '31AUG2034'd then acad_Year="2033-34";
190  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01JAN2033'd and datepart(DEGR_CONFER_DT) < '31DEC2034'd then cal_year="2033-34";
191  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '30JUN2034'd and datepart(DEGR_CONFER_DT) < '01JUL2035'd then IPEDS_year="2034-35";
192  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01SEP2034'd and datepart(DEGR_CONFER_DT) < '31AUG2035'd then acad_Year="2034-35";
193  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01JAN2034'd and datepart(DEGR_CONFER_DT) < '31DEC2035'd then cal_year="2034-35";
194  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '30JUN2035'd and datepart(DEGR_CONFER_DT) < '01JUL2036'd then IPEDS_year="2035-36";
195  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01SEP2035'd and datepart(DEGR_CONFER_DT) < '31AUG2036'd then acad_Year="2035-36";
196  MPRINT(D3):   if datepart(DEGR_CONFER_DT) > '01JAN2035'd and datepart(DEGR_CONFER_DT) < '31DEC2036'd then cal_year="2035-36";
```

Segway Into Level II – Macro Programs



What is Dynamic Programming

Art Carpenter

Dynamic programs **adjust** to their environment. Rather than telling the program which data sets or variables process against, these programs tend to make these determinations at execution time.

Portability

Dynamic Programs

	name	county
1	Andrew	Wilson
2	Britt	Wake
3	Brian	Onslow

	county	ctype
1	Wake	Within
2	Wilson	Adjacent
3	Onslow	Other

```
Data _NULL_;
    ❶Set county ❷end=no_more;
    ❸Call symputx(CATT("Countyz",_n_),County);
    ❹Call symputx(CATT("Ctype",_n_),Ctype);
    ❺If no_more THEN CALL symputx('NUMROWS',_n_);
Run;
```

Macro If Then with Else

```
❶%MACRO TestIf;
Data Recode;
    set Base;
    ❷%DO a=1 %To &Numrowsa;
    ❸%IF &a>1 %Then Else;
        ❹if UPCASE(county)=UPCASE("&&Countyz&a") Then
            Recode = "&&Ctype&a";
    %END;
Run;
❶%MEND TestIF;

%TESTIF;
```

Macro Program Resolution

```
❷%DO a=1 %To &Numrowsa;
❸%IF &a>1 %Then Else;
    ❹if UPCASE(county)=UPCASE("&&Countyz&a") Then
    Recode = "&&Ctype&a";
```

```
43  MPRINT(TESTIF):   Data Recode;
44  MPRINT(TESTIF):   set Base;
45  MPRINT(TESTIF):   if UPCASE(county)=UPCASE("Wake") Then Recode = "Within";
46  MPRINT(TESTIF):   Else if UPCASE(county)=UPCASE("Wilson") Then Recode = "Adjacent";
47  MPRINT(TESTIF):   Else if UPCASE(county)=UPCASE("Onslow") Then Recode = "Other";
48  MPRINT(TESTIF):   Run;
49
```

Export Macros for documentation

```
%LET b = %SYSFUNC(SUBSTR(%SYSFUNC(dequote(&_CLIENTPROJECTPATH)),1,
                %EVAL((%SYSFUNC(FIND((%SYSFUNC(deQUOTE(&_CLIENTPROJECTPATH))),
                %SYSFUNC(deQUOTE(&_CLIENTPROJECTNAME)))))-2)));

%PUT &B;

DATA _NULL_;
 FILE "&b.Parameters &TERM %SYSFUNC(Today(),YYMMDD10.).txt";
PUT
@1 'term = ' "&Term" /
@1 'termfy1 = ' "&termfy1" /
@1 'termfy2 = ' "&termfy2" /
@1 'FYMin = ' "&FYMin" /
@1 'FYMax = ' "&FYMax" /
@1 'Yr3Min = ' "&Yr3Min" /
@1 'Yr3Max = ' "&Yr3Max";
Run;
```

Overview

%LET enhance with %SYSFUNC and %EVAL

PROC SQL INTO :VarName separated by “, “

CALL SYMPUTX for looping through reports

1) Create your model,
2) Create the macro,
3) Create the direct call or loops.

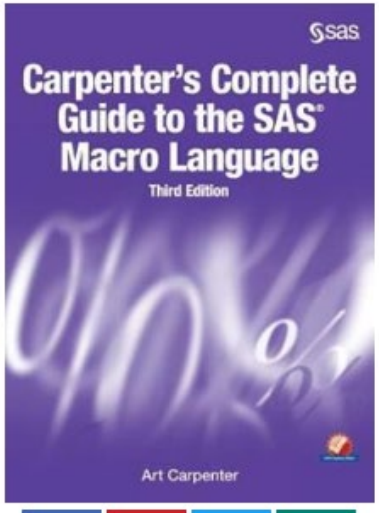
Next Steps

Infuse, enhance, level-up your SAS programs to

- reduce opportunity **error**
- self-documenting
- self-guiding
- give the user clear guidance on errors
- Start your programs with variables that change/repeat

Every time you open a program, think about **reducing error** and **enhance** the user's experience – make it simple for the next person.

Resource



Carpenter's Complete Guide to the SAS Macro Language,
Art Carpenter, 2016, Third Edition
SAS 9.4 Macro Language: Reference, Fifth Edition

Useful Website: everyone and no one is talking about
www.lexjansen.com
www.WalkerDataRanger.com

SAS® 9.4 Macro Language: Reference, Fifth Edition