ABSTRACT
If you've ever wished for a way to assign character values based on a logical expression in one line of code, well I've got great news for you: CHOOSEC, you'll see! Usually, character values are assigned in the data step using one line of code with a PUT using a format or $format. But this requires creation of the format in another procedure altogether, sometimes even requiring the programmer to actually scroll up, create the new format, and potentially lose track of the progress being made in the data step. At minimum, the format/$format method isn't a true One Line Solution™ since additional lines are needed to create the format before it's available in the data step. CHOOSEC can simulate a format or $format in one line of code in the data step.

INTRODUCTION
As a programmer, I strive to streamline my code to reduce overall length and demand on computational resources, while maximizing readability. In this regard, One Line Solutions™ are generally best.

I've long wished for a One Line Solution™ to assign character values based on some algorithm “on the fly” in the data step, but until the PharmaSUG 2021 paper “What Kind of WHICH Do You CHOOSE to be?” by Richann Watson and Louise Haddon, I didn’t know of one. In this paper I'll highlight the CHOOSEC function.

SAS® DOCUMENTATION (SAS.COM)
THE CHOOSEC FUNCTION
Returns a character value that represents the results of choosing from a list of arguments.

SYNTAX
CHOOSEC(index-expression, selection-1 <, …selection-n>)

REQUIRED ARGUMENTS
Index-expression: specifies a numeric constant, variable, or expression.
Selection: specifies a character constant, variable, or expression. The value of this argument is returned by the CHOOSEC function.

LENGTH OF RETURNED VARIABLE
In a DATA step, if the CHOOSEC function returns a value to a variable that has not previously been assigned a length, that variable is given a length of 200 bytes.

THE BASICS
The CHOOSEC function uses the value of index-expression to select from the arguments that follow. For example, if index-expression is 3, CHOOSEC returns the value of selection-3. If the first argument is negative, the function counts backward from the list of arguments and returns that value.
CHOOSSEC EXAMPLES

EXAMPLE #1 BASIC IMPLEMENTATION AND COMPARISON TO COMMON ALTERNATIVE METHODS

* The format method requires creation of the format *;

```sas
proc format;
value arm
  1 = '1.5 mg'
  2 = '5 mg'
  3 = '10 mg';
run;
```

```sas
data dummyrand;
set sdtm.dm;
* Actual randomization may come in as numeric variable RANDGRP;
randgrp = ceil(ranuni(1)*3); * <-- Index-expression ranges 1-3;
* length statement not required for CHOOSEC: default length $200.;
arm = CHOOSEC( randgrp , '1.5 mg' , '5 mg' , '10 mg');
* Format method generally requires an additional length statement;
length arm2 $200.;
arm2 = put( randgrp , arm. );
* Logical structures like IFN/IFC, IF/THEN, and SELECT are excluded due to obvious multi-line requirement where an extra line is required for each possible value;
run;
```

EXAMPLE #2 LOGICAL EXPRESSION ARITHMETIC; ON THE FLY FORMAT SIMULATION

You can turn any kind of input data into an index-expression using logical expression arithmetic.

```sas
data _null_;
set sdtm.dm;
* Simulate FORMAT in one line in the data step *;
CAT1 = choosec( 1 + 2*( age < 65 ) + 3*( age >= 65 ) , '003 Unk' , '001 <65 years' , '002 >=65 years' );
* Simulate $FORMAT in one line in the data step *;
CAT2 = choosec( (sex='M') + 2*(sex='F') , '001 Male' , '002 Female' );
run;
```

EXAMPLE #3 VARIABLE ASSIGNMENT CONSTRUCTION

* Pseudocode. FORMAT approach for illustrative purposes only. This code does not work as-is, not sure this is possible;

```sas
proc format;
value cat1f
  1 - 3 = [catx(' ',put(orresn,z3.),orresn,orres)];
  4 - high = '004 >=4 No Change, Minimally Worse, Much Worse, Very Much Worse';
run;
```
data _null_;  
set raw.cgic(where=(^missing(orresn)));  
  * Again: Theoretical only, FORMAT above does not work as written ;  
  length CAT1 $200.;  
  cat1 = put(orresn,cat1f.);  
run;  
  
* CHOOSEC Works! *;  
data _null_;  
set raw.cgic(where=(^missing(orresn)));  
  * Assigned value depends on variable value in SELECTION element(s);  
  cat1 = choosec( 1 + (orresn GE 4)  
    , catx(' ',put(orresn,23.),orresn,orres)  
    , '004 >=4 No Change, Minimally Worse, Much Worse, Very Much Worse' );  
run;  

CONCLUSION
CHOOSCEC is a versatile function that allows you to assign character values from logical expression. (A.k.a. Simulate a format with one line of code in the data step.)

REFERENCES


CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:

    Joel Campbell  
    Advanced Analytics, Inc.  
    Joel.Campbell@AdvancedAnalyticsCRO.com