

PHUSE EU Connect 2025 (Presentation – CT04)

Efficient SAS programming using CHOOSEC/CHOOSEN and WHICHC/WHICHN over traditional IF-THEN-ELSE clause

Nimit Agrawal, Principal Statistical Programmer, Syneos Health[®]

Sumit Pratap Pradhan, Manager, Statistical Programming, Syneos Health[®]

November 17, 2025

AGENDA

- 01 Introduction
- 02 Overview of CHOOSEC/CHOOSEN
- 03 Overview of WHICHC/WHICHN
- 04 Combination of CHOOSEC/WHICHC
- 05 Conclusion
- 06 Q&A

Introduction

- CHOOSESEC/CHOOSESEN and WHICHHC/WHICHN are the new set of functions introduced in SAS 9.2, these set of functions provide handy and efficient alternative to traditional IF-THEN-ELSE statement for implementation of conditional logic
- These functions can be used effectively in many scenarios over traditional IF-THEN-ELSE clause e.g.
 - Assignment of numeric value to different Overall Response values (CR, PR, etc.) for the derivation of BOR.
 - assignment of CRF text to a numeric toxicity grading scale
- This approach will help reduce the long time we spend for writing repetitive IF-THEN-ELSE clauses and as a result, will make our program more efficient, readable, precise and robust.

Overview of CHOOSEC Function

Syntax: CHOOSEC (index-expression, selection-1 <,...selection-n>)

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.

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.

Example of CHOOSEC Function

Scenario: To derive AE toxicity grades text present on CRF from AETOXGR

```
ATOXGR = choossec(input(AETOXGR,best.), "Grade 1 - Mild", "Grade 2 - Moderate", "Grade 3 - Severe",  
                  "Grade 4 - Very Severe", "Grade 5 - Life-Threatning");
```

```
if AETOXGR = "1" then ATOXGR = "Grade 1 - Mild";  
else if AETOXGR = "2" then ATOXGR = "Grade 2 - Moderate";  
else if AETOXGR = "3" then ATOXGR = "Grade 3 - Severe";  
else if AETOXGR = "4" then ATOXGR = "Grade 4 - Very Severe";  
else if AETOXGR = "5" then ATOXGR = "Grade 5 - Life-Threatning";
```

Traditional code using
IF-THEN-ELSE

Output

AETOXGR	ATOXGR
1	Grade 1 - Mild
2	Grade 2 - Moderate
3	Grade 3 - Severe
4	Grade 4 - Very Severe
5	Grade 5 - Life-Threatning

Overview of CHOSEN Function

Syntax: CHOSEN (index-expression, selection-1 <,...selection-n>)

CHOSEN function is the numeric counterpart of the CHOOSE function, returns a numeric value that represents the results of choosing from a list of arguments.

Example: This function will be useful in the scenario where the sequential numeric values need to be decoded to the discrete numeric values.

```
ORD_DE = chosen(ORD, 7, 13, 17);
```

Output:

ORD	ORD_DE
1	7
2	13
3	17

Overview of WHICHC Function

Syntax: WHICHC (string, value-1 <, value-2, ...>)

Basics: The WHICHC function searches the second and subsequent arguments for a value that is equal to the first argument and returns the index of the first matching value.

Example: Assignment of numeric value to corresponding different Overall Response values

```
ORESPN = whichc(OVERRESP, "CR", "PR", "SD", "PD", "NE");
```

```
if OVERRESP = "CR" then ORESPN = 1;  
else if OVERRESP = "PR" then ORESPN = 2;  
else if OVERRESP = "SD" then ORESPN = 3;  
else if OVERRESP = "PD" then ORESPN = 4;  
else if OVERRESP = "NE" then ORESPN = 5;
```

Traditional code using IF-THEN-ELSE

Output

OVERRESP	ORESPN
CR	1
PR	2
SD	3
PD	4
NE	5

Overview of WHICHN Function

The WHICHN function is the numeric counterpart of WHICHC function, searches for a numeric value that is equal to the first argument and returns the index of the first matching value.

Syntax: WHICHN (argument, value-1 <, value-2, ...>)

Example: Conversion of Questionnaire Scores into sequential numeric value as part of primary endpoint analysis

AVAL = **whichn**(QSSTRESN, 10, 20, 35, 43, 47) ;

Output:

⊕ QSSTRESN	⊕ AVAL
10	1
20	2
35	3
43	4
47	5

Application of CHOOSEC and WHICHC Function in combination

Scenario: Assignment of full CRF text corresponding to decoded Overall Response values

```
if OVERRESP = 'PD' then PARAM = 'Progressive Disease';  
else if OVERRESP = 'PR' then PARAM = 'Partial Response';  
else if OVERRESP = 'CR' then PARAM = 'Complete Response';  
else if OVERRESP = 'SD' then PARAM = 'Stable Disease';  
else if OVERRESP = 'NE' then PARAM = 'Not Evaluable';
```

Alternative Scenario: Using CHOOSEC and WHICHC function

```
PARAM = choosec (whichc (OVERRESP, 'PD', 'PR', 'CR', 'SD', 'NE'),  
                'Progressive Disease', 'Partial Response', 'Complete Response', 'Stable Disease', 'Not Evaluable');
```

Output:

OVERRESP	PARAM
CR	Complete Response
PR	Partial Response
SD	Stable Disease
PD	Progressive Disease
NE	Not Evaluable

Conclusion

Conditional processing in SAS can be performed in multiple ways - one way is to follow traditional approach by using IF-THEN-ELSE statement.

An alternative approach to perform conditional processing using CHOOSEC/CHOOSEN and WHICHC/WHICHN is illustrated above with multiple examples and use cases. These two sets of SAS functions are still under the unexplored territory, but they are very efficient and versatile and if used regularly in day-to-day programming activities could result in simplified programming, readability, reusability and robustness.



Contact Information

Nimit Agrawal

Principal Statistical Programmer

nimit.agrawal@syneoshealth.com

<https://www.linkedin.com/in/nimit-agrawal-171bb333/>

Sumit Pratap Pradhan

Manager, Statistical Programming

sumit.pradhan@syneoshealth.com

<https://www.linkedin.com/in/sumit-pradhan-71133345/>

