



Alternative programming approach for Conditional processing in SAS

Balram Chauhan, Cytel

**PhUSE US Connect 2018
Transformative Current and Emerging Best Practices**

Outline

- **Introduction**
- **Examples on alternative methods**
- **Conclusion**
- **References**

Introduction

- Conditional processing is unavoidable and found in nearly every SAS program.
- Primary tool is IF ...THEN ...ELSE / WHERE / SELECT statements.
- Although IF ... THEN... ELSE / WHERE are powerful and easy to use however there are other alternative methods.
- Those are majorly used for :
 - To make code compact
 - To make code readable
 - To make code reusable

Scenario #1

- Combine PCTESTCD and PCSPECS to get unique PARAMCD

Standard Method

```
if PCTESTCD = 'RADICCON' AND PCSPEC = 'STOOL' THEN PARAMCD = 'RCON_FE';
if PCTESTCD = 'RADICCON' AND PCSPEC = 'URINE' THEN PARAMCD = 'RCON_UR';
if PCTESTCD = 'RADICCON' AND PCSPEC = 'STOOL/URINE' THEN PARAMCD = 'RCON_TOT';

if PCTESTCD = 'RADIOAMT' AND PCSPEC = 'STOOL' THEN PARAMCD = 'RAMT_FE';
if PCTESTCD = 'RADIOAMT' AND PCSPEC = 'URINE' THEN PARAMCD = 'RAMT_UR';
if PCTESTCD = 'RADIOAMT' AND PCSPEC = 'STOOL/URINE' THEN PARAMCD = 'RAMT_TOT';

if PCTESTCD = 'RADIOPCT' AND PCSPEC = 'STOOL' THEN PARAMCD = 'RPCT_FE';
if PCTESTCD = 'RADIOPCT' AND PCSPEC = 'URINE' THEN PARAMCD = 'RPCT_UR';
if PCTESTCD = 'RADIOPCT' AND PCSPEC = 'STOOL/URINE' THEN PARAMCD = 'RPCT_TOT';
```

- Alternative Method :** Using IFC/IFN Function

```
PARAMCD1 = catx('_',
  ifc (PCTESTCD = 'RADIOAMT', 'RAMT', ifc(PCTESTCD = 'RADIOPCT', 'RPCT', ifc(PCTESTCD = 'RADICCON', 'RCON', ''))),
  ifc (PCSPEC = 'STOOL', 'FE', ifc (PCSPEC = 'URINE', 'UR', ifc (PCSPEC = 'STOOL/URINE', 'TOT', ''))));
```

OUTPUT

PCSPEC	PCTESTCD	PARAMCD	PARAMCD1
STOOL	RADICCON	RCON_FE	RCON_FE
STOOL/URINE	RADICCON	RCON_TOT	RCON_TOT
URINE	RADICCON	RCON_UR	RCON_UR
STOOL	RADIOAMT	RAMT_FE	RAMT_FE
STOOL/URINE	RADIOAMT	RAMT_TOT	RAMT_TOT
URINE	RADIOAMT	RAMT_UR	RAMT_UR
STOOL	RADIOPCT	RPCT_FE	RPCT_FE
STOOL/URINE	RADIOPCT	RPCT_TOT	RPCT_TOT
URINE	RADIOPCT	RPCT_UR	RPCT_UR

IFN/IFC Function

IFC/IFN function is similar to IF THEN ELSE statement which can compact the standard conditional processing.

Syntax : IFC(logical-expression, value-returned-when-true, value-returned-when-false)

The IFN function is similar to the IFC function except that IFN returns a numeric value while IFC returns a character value.

Scenario #2

- To create AE grade from severity variable AESEV

Standard Method

```
if AESEV = "MILD" then AGRD = 'GRADE I';  
else if AESEV = "MODERATE" then AGRD = 'GRADE II';  
else if AESEV = "SEVERE" then AGRD = 'GRADE III';  
else if AESEV = "VERY SEVERE" then AGRD = 'GRADE IV';  
else if AESEV = "DEATH" then AGRD = 'GRADE V';
```

- Alternative Method** : Using WHICHC/WHICHN Function

Using CHOOSEC/CHOOSEN Function

```
AGRD1 = choosec(whichc(AESEV, 'MILD', 'MODERATE', 'SEVERE', 'VERY SEVERE', 'DEATH'),  
                'GRADE I', 'GRADE II', 'GRADE III', 'GRADE IV', 'GRADE V');
```

1 2 3 4 5

Compares Index Value

Returns Index Value

Output

AESEV	AGRD	AGRD1
MODERATE	GRADE II	GRADE II
MILD	GRADE I	GRADE I
MODERATE	GRADE II	GRADE II
MODERATE	GRADE II	GRADE II
MILD	GRADE I	GRADE I
MILD	GRADE I	GRADE I
MILD	GRADE I	GRADE I
VERY SEVERE	GRADE IV	GRADE IV

WHICHC/WHICHN Function

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.

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

- If string is missing, then WHICHC returns a missing value and If string does not equal any subsequent argument, then WHICHC returns 0.
- WHICHN is similar to WHICHC, Searches for a numeric value that is equal to the first argument, and returns the index of the first matching value.

CHOOSEC/CHOOSEN Function

The CHOOSEN function uses the value of index-expression to select from the arguments that follow

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

- The CHOOSEN function is similar to the CHOOSEC function except that CHOOSEN returns a numeric value while CHOOSEC returns a character value.

Scenario #3

- Filtering dataset using single/multiple population flags

Standard Method :

```
data TEST;  
set ADSL;  
where SAFFL = 'Y';  
run;
```



```
data TEST;  
set ADSL;  
where SAFFL = 'Y' and PKFL = 'Y';  
run;
```



```
data TEST;  
set ADSL;  
where SAFFL = 'Y' and PKFL = 'Y' and complfl = 'Y';  
run;
```



DATA :

	USUBJID	SAFFL	PKFL	COMPLFL
1	103-01-01	Y	Y	Y
2	103-01-02	Y	Y	Y
3	103-01-03	Y	Y	Y
4	103-01-04	N	N	N
5	103-01-05	Y	Y	Y
6	103-01-06	N	Y	Y
7	103-01-07	Y	Y	N
8	103-01-08	N	Y	Y
9	103-01-09	Y	N	Y
10	103-01-10	Y	N	Y

Alternative for Scenario #3

- Shorten expressions using user defined functions :

PROGRAM: Filtering with one Population variable

```
data TEST;  
set ADSL;  
where pop1(SAFFL) ; /*  
run;
```

	USUBJID	SAFFL	PKFL	COMPLFL
1	103-01-01	Y	Y	Y
2	103-01-02	Y	Y	Y
3	103-01-03	Y	Y	Y
4	103-01-04	N	N	N
5	103-01-05	Y	Y	Y
6	103-01-06	N	Y	Y
7	103-01-07	Y	Y	N
8	103-01-08	N	Y	Y
9	103-01-09	Y	N	Y
10	103-01-10	Y	N	Y

LOG:

```
NOTE: There were 7 observations read from the data set WORK.ADSL.  
WHERE POP1(SAFFL);  
NOTE: The data set WORK.TEST has 7 observations and 4 variables.  
NOTE: DATA statement used (Total process time):  
      real time           0.01 seconds  
      cpu time            0.01 seconds ■
```

Alternative for Scenario #3

PROGRAM: Filtering with Two Population variables

```
data TEST;  
set ADSL;  
where pop2 (SAFFL, PKFL)  
run;
```



	USUBJID	SAFFL	PKFL	COMPLFL
1	103-01-01	Y	Y	Y
2	103-01-02	Y	Y	Y
3	103-01-03	Y	Y	Y
4	103-01-04	N	N	N
5	103-01-05	Y	Y	Y
6	103-01-06	N	Y	Y
7	103-01-07	Y	Y	N
8	103-01-08	N	Y	Y
9	103-01-09	Y	N	Y
10	103-01-10	Y	N	Y

LOG:

```
NOTE: There were 5 observations read from the data set WORK.ADSL.  
      WHERE POP2(SAFFL, PKFL);  
NOTE: The data set WORK.TEST has 5 observations and 4 variables.  
NOTE: DATA statement used (Total process time):  
      real time           0.01 seconds  
      cpu time            0.01 seconds
```

Alternative for Scenario #3

PROGRAM: Filtering with Three Population variables

```
data TEST;  
set ADSL;  
where pop3 (SAFFL, PKFL, COMPLFL) ;  
run;
```



	USUBJID	SAFFL	PKFL	COMPLFL
1	103-01-01	Y	Y	Y
2	103-01-02	Y	Y	Y
3	103-01-03	Y	Y	Y
4	103-01-04	N	N	N
5	103-01-05	Y	Y	Y
6	103-01-06	N	Y	Y
7	103-01-07	Y	Y	N
8	103-01-08	N	Y	Y
9	103-01-09	Y	N	Y
10	103-01-10	Y	N	Y

LOG:

NOTE: There were 4 observations read from the data set WORK.ADSL.
WHERE POP3(SAFFL, PKFL, COMPLFL);

NOTE: The data set WORK.TEST has 4 observations and 4 variables.

NOTE: DATA statement used (Total process time):

real time 0.01 seconds

cpu time 0.01 seconds

Alternative for Scenario #3 : FCMP proc

```
/*To create user define functions */
options cmplib = work.functions;

proc fcmp outlib = work.functions.conf;
/*User define function : for POP1 */
    function pop1(str1 $);
    if upcase(strip(str1)) eq 'Y' then return(1);
    else return(0);
    endsub;
/*User define function : for POP2 */
    function pop2(str1 $, str2 $);

    if upcase(strip(str1)) eq 'Y' and upcase(strip(str2)) eq 'Y' then
return(1);
    else return(0);
    endsub;
/*User define function : for POP3 */
    function pop3(str1 $, str2 $, str3 $);
    if upcase(strip(str1)) eq 'Y' and upcase(strip(str2)) eq 'Y' and
upcase(strip(str3)) eq 'Y' then return(1);
    else return(0);
    endsub;

run;
```

User define Functions : FCMP proc

- FCMP procedure allows you to create new Functions.
- POP1, POP2, POP3 are user define functions which can be used across all programs and studies/projects once created and stored in permanent/temporary library.
- Functions always return an explicit result.
- You can create your own functions for most commonly used tasks like Study day calculation, BMI using height and weight, date imputation, calculating age ... etc.

CONCLUSION

- There are several ways to do same task in SAS
- Each way has its own advantages and disadvantages
- Effective use of appropriate methods can make programmers life easy
- Alternative methods can simplify programming, readability & reusability

References

- <http://support.sas.com/documentation>
- **The FCMP Procedure - SAS Support:**
<https://support.sas.com/documentation/onlinedoc/base/91/fcmp.pdf>

Any Questions ?

Thank You