PhUSE 2017

Paper CT02

Efficient Coding Techniques In SAS®

Geetha Kesireddi, GCE Solutions Inc, Hyderabad, India

ABSTRACT:

SAS is most widely used language in Clinical industry for statistical analysis and its usage is improving drastically. Regular habituated way of programming in SAS is usage of some of the old functions, regular 'if then else' statements. These are basic techniques which are widely used. However, as the industry is growing, programming using these basic techniques is making the code humongous and difficult for maintenance. Continuous upgrade of both hardware and software over time, is becoming very difficult with these basic methods/approach. One of the solutions for this problem is improving the coding techniques by using new functions like IFC, IFN, CHOOSEC and CHOOSEN. Which can be used in day to day programming, to make the code much easy for maintenance and to drastically brings down the time to upgrade.

INTRODUCTION:

Almost every SAS programmer uses 'If then else' statements often to implement conditional logics in the program. IFC, IFN, CHOOSEC and CHOOSEN functions are great alternatives to the regular 'if then else' statements. These functions are very handy to use. These functions help in reducing the lines of code and helps in handling missing values.

IFC, IFN functions are very useful for encoding, decoding and flagging values. These functions can be used in data step and in proc SQL, making it more powerful.

This paper illustrates various scenarios where these functions can be used.

IFC FUNCTION:

The IFC function returns a character value based on whether an expression is true, false, or missing.

SYNTAX:

IFC(logical-expression, value-returned-when-true, value-returned-when-false <, value-returned-when-missing>)

EXAMPLE 1:

To derive treatment group variables (TRTA and TRTP). Based on the respective APERIOD values, TRTA and TRTP values were assigned from TRT01A/TRT01P, TRT02A/TRT02P, TRT03A/TRT03P. Suppose if APERIOD is 1 then TRTA/TRTP will be TRT01A/TRT01P etc.

	SUBJID	APERIOD	TRT01A	TRT02A	TRT03A	TRT01P	TRT02P	TRT03P
1	1001	1	Placebo	xx100mg	xx200mg	xx100mg	Placebo	xx100mg
2	1002	2	xx100mg	Placebo	xx200mg	Placebo	xx100mg	xx200mg
3	1003	3	Placebo	xx100mg	xx100mg	Placebo	xx100mg	xx200mg

SOLUTION 1: Using 'if then else' statements.

```
if APERIOD=1 then TRTA=TRT01A;
  else if APERIOD=2 then TRTA=TRT02A;
    else if APERIOD=3 then TRTA=TRT03A;
```

```
if APERIOD=1 then TRTP=TRT01P;
  else if APERIOD=2 then TRTP=TRT02P;
    else if APERIOD=3 then TRTP=TRT03P;
```

SOLUTION 2: Using 'IFC' function.

```
TRTA=ifc(APERIOD=1,TRT01A, ifc(APERIOD=2,TRT02A, ifc(APERIOD=3,TRT03A,'')));
TRTP=ifc(APERIOD=1,TRT01P, ifc(APERIOD=2,TRT02P, ifc(APERIOD=3,TRT03P,'')));
```

which yields the results:

	SUBJID	APERIOD	TRT01A	TRT02A	TRT03A	TRT01P	TRT02P	TRT03P	TRTA	TRTP
1	1001	1	Placebo	xx100mg	xx200mg	xx100mg	Placebo	xx100mg	Placebo	xx100mg
2	1002	2	xx100mg	Placebo	xx200mg	Placebo	xx100mg	xx200mg	Placebo	xx100mg
3	1003	3	Placebo	xx100mg	xx100mg	Placebo	xx100mg	xx200mg	xx100mg	xx200mg

EXAMPLE 2:

Conditional execution of statements using IFC. ENREF is created using AEENRF, AEENDTC, AEENDY variables where in, the 'U' value in AEENRF should be changed to 'UNKNOWN' and if the AEENRF values are missing then in that place AEENDTC and AENDY should be populated.

	SUBJID	AETERM	AEENDTC	AEENRF	AENDY
1	2007	COLD	2014-09-08		756
2	2007	DIARRHEA	2014-04-14		609
3	2007	INSOMNIA		U	.
4	2007	HEADACHE	2013-03-04		203
5	2007	SORE-THROAT	2014-07-20		706
6	2010	ARYTHMIA		AFTER	.

SOLUTION 1: Using 'if then else' statements.

```
data test;
 set ae;
  if AEENRF ne '' then do;
   if AEENRF='U' then AEENRF='UNKNOWN';
     ENREF=AEENRF; end;
      else ENREF=cats(AEENDTC,'/',put(AENDY,best.));
run;
```

SOLUTION 2: Using 'IFC' in proc SQL.

```
proc sql;
 create table test
 as select SUBJID,AETERM,AEENDTC,AAEENRF,AENDY,
   ifc(AEENRF ^='',ifc(AEENRF='U','UNKNOWN',AEENRF),
   ifc(AEENDTC ^='',cats(AEENDTC,'/',put(AENDY,best.)),'')) as ENREF
 from ae;
quit;
```

which yields the results:

	SUBJID	AETERM	AEENDTC	AEENRF	AENDY	ENREF
1	2007	COLD	2014-09-08		756	2014-09-08/756
2	2007	DIARRHEA	2014-04-14		609	2014-04-14/609
3	2007	INSOMNIA		U	.	UNKNOWN
4	2007	HEADACHE	2013-03-04		203	2013-03-04/203
5	2007	SORE-THROAT	2014-07-20		706	2014-07-20/706
6	2010	ARYTHMIA		AFTER	.	AFTER

EXAMPLE 3:

For creation of flags like treatment completion flag (TRTCPLFL). If the subject's decode value in disposition dataset is 'COMPLETED' in treatment epoch then treatment completion flag will be 'Y', else the reason for non-completion will be specified in the treatment discontinuation (TRTDISC) column.

	SUBJID	dsdecod_trt
1	1001	COMPLETED
2	1001	COMPLETED
3	1001	COMPLETED
4	1001	COMPLETED
5	1001	COMPLETED
6	1001	SCREEN FAILURE
7	1001	STUDY TERMINATED BY SPONSOR
8	1001	COMPLETED
9	1001	COMPLETED
10	1001	SUBJECT/GUARDIAN DECISION
11	1001	COMPLETED

SOLUTION 1: Using 'if then else' statements.

```
if upcase(DSDECOD_TRT) eq 'COMPLETED' then TRTCPLFL='Y';
 else TRTCPLFL='N';
if upcase(DSDECOD_TRT) eq 'COMPLETED' then TRTDISC='';
 else TRTDISC = DSDECOD_TRT;
```

SOLUTION 2: Using 'IFC' function.

```
TRTCPLFL = ifc((upcase(DSDECOD_TRT) eq 'COMPLETED'), 'Y', 'N');
TRTDISC  = ifc((upcase(DSDECOD_TRT) eq 'COMPLETED'), ' ', DSDECOD_TRT);
```

which yields the results:

	SUBJID	DSDECOD_TRT	TRTCPLFL	TRTDISC
1	1001	COMPLETED	Y	
2	1001	COMPLETED	Y	
3	1001	COMPLETED	Y	
4	1001	COMPLETED	Y	
5	1001	COMPLETED	Y	
6	1001	STUDY TERMINATED BY SPONSOR	N	STUDY TERMINATED BY SPONSOR
7	1001	COMPLETED	Y	
8	1001	COMPLETED	Y	
9	1001	SUBJECT/GUARDIAN DECISION	N	SUBJECT/GUARDIAN DECISION
10	1001	COMPLETED	Y	

IFN FUNCTION:

The IFN function returns a numeric value based on whether an expression is true, false, or missing.

SYNTAX: IFN(logical-expression, value-returned-when-true, value-returned-when-false<, value-returned-when-missing>)

EXAMPLE 1:

To derive ATOXGRN and BTOXGRN (numeric variables of toxicity grades) in LAB dataset.

Sample data as follows:

	SUBJID	ATOXGR	BTOXGR
1	1008	Grade 1	Grade 1
2	1008	Grade 1	No Grade
3	1008	Grade 2	Grade 1
4	1008	Grade 2	No Grade
5	1008	Grade 3	No Grade
6	1008	No Grade	Grade 1
7	1008	No Grade	No Grade

SOLUTION 1:

- Using 'if then else' statements.

```
if ATOXGR ne '' and BTOXGR ne '' then do;

     if ATOXGR='No grade' then ATOXGRN=0;
       else ATOXGRN=input(scan(ATOXGR,2),best.);

     if BTOXGR='No grade' then BTOXGRN=0;
       else BTOXGRN=input(scan(BTOXGR,2),best.);
end;
```

SOLUTION 2:

- Using 'IFN' Function

```
if cmiss(ATOXGR,BTOXGR) eq 0 then do;

ATOXGRN = ifn(compress(ATOXGR,'No Grade') = '', 0,input(compress(ATOXGR,
'No Grade'),best.));

BTOXGRN = ifn(compress(BTOXGR,'No Grade') = '', 0,input(compress(BTOXGR,
'No Grade'),best.));

end;
```

which yields the results:

	SUBJID	ATOXGR	ATOXGRN	BTOXGR	BTOXGRN
1	1008	Grade 1	1	Grade 1	1
2	1008	Grade 1	1	No Grade	0
3	1008	Grade 2	2	Grade 1	1
4	1008	Grade 2	2	No Grade	0
5	1008	Grade 3	3	No Grade	0
6	1008	No Grade	0	Grade 1	1
7	1008	No Grade	0	No Grade	0

EXAMPLE 2:

To derive VISITNUM from VISIT variable.

Sample visit data as follows:

	SUBJID	VISIT
1	101	RANDOMIZATION
2	101	UNP_IVR
3	101	WEEK 1
4	101	WEEK 2
5	101	WEEK 4
6	101	WEEK 8
7	101	WEEK 12
8	101	WEEK 13
9	101	WEEK 14
10	101	WEEK 15
11	101	WEEK 16
12	101	WEEK 20

```
proc format;
 value $vsnum
  'RANDOMIZATION' =  100
  'WEEK 1'        =  101
  'WEEK 2'        =  102
  'WEEK 4'        =  103
  'WEEK 8'        =  104
  'WEEK 12'       =  105
  'WEEK 13'       =  106
  'WEEK 14'       =  107
  'WEEK 15'       =  108
  'WEEK 16'       =  109
  'WEEK 20'       =  110
  ;
run;
```

SOLUTION 1:

- Using 'if then else' statements.

```
data test;
 set visit;
   if VISIT = 'UNP_IVR' then VISITNUM = 999;
    else VISITNUM = input(VISIT, $vsnum.);
run;
```

SOLUTION 2:

- Using 'IFN' Function

```
data test;
  set visit;
     VISITNUM = ifn((VISIT eq 'UNP_IVR'), 999, input(VISIT, $vsnum.));
 run;
```

which yields the results:

	SUBJID	VISIT	VISITNUM
1	101	RANDOMIZATION	100
2	101	UNP_IVR	999
3	101	WEEK 1	101
4	101	WEEK 2	102
5	101	WEEK 4	103
6	101	WEEK 8	104
7	101	WEEK 12	105
8	101	WEEK 13	106
9	101	WEEK 14	107
10	101	WEEK 15	108
11	101	WEEK 16	109
12	101	WEEK 20	110

CHOOSEC FUNCTION:

SYNTAX:

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

Index-expression value is to select from the arguments that follow, selection-1 to selection-n are the character values from which to choose. If the first argument is negative, the function counts backwards from the list of arguments, and returns that value.

Purpose is to select a character value from a list of character values (supplied as the 2nd to nth arguments of the function)

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

EXAMPLE:

To assign APERIODC values based on APERIOD. If the APERIOD value is 1 then APERIODC will be 'INDUCTION', if APERIOD value is 2 then APERIODC will be 'MAINTENANCE', if the APERIOD value is 3 then APERIODC will be 'TREATMENT'.

	SUBJID	PARAM	PARAMCD	APERIOD	ADT
1	101	Body mass index (kg/m2)	BMISI	1	2012-04-03
2	101	Body mass index (kg/m2)	BMISI	2	2012-06-03
3	101	Body mass index (kg/m2)	BMISI	2	2012-09-03
4	101	Body mass index (kg/m2)	BMISI	2	2013-01-03
5	101	Body mass index (kg/m2)	BMISI	3	2013-04-03
6	101	Body mass index (kg/m2)	BMISI	3	2013-07-03
7	101	Body mass index (kg/m2)	BMISI	3	2013-10-03

SOLUTION 1:

- Using 'if then else' statements.

```
data test;
 length APERIODC $20;
 set vs;
```

```
   if APERIOD=1 then APERIODC='INDUCTION';
    else if APERIOD=2 then APERIODC='MAINTENANCE';
     else if APERIOD=3 then APERIODC='TREATMENT';
      else APERIODC='';
run;
```

SOLUTION 2:

- Using 'CHOOSEC' function.

```
data test;
 length APERIODC $20;
 set vs;
  APERIODC=choosec(APERIOD, 'INDUCTION', 'MAINTENANCE', 'TREATMENT', '');
run;
```

Hence based on the index expression value of aperiod, CHOOSEC function assigns the appropriate aperiodc values which are listed.

Which yields the results:

	SUBJID	PARAM	PARAMCD	APERIOD	APERIODC	ADT
1	101	Body mass index (kg/m2)	BMISI	1	INDUCTION	2012-04-03
2	101	Body mass index (kg/m2)	BMISI	2	MAINTENANCE	2012-06-03
3	101	Body mass index (kg/m2)	BMISI	2	MAINTENANCE	2012-09-03
4	101	Body mass index (kg/m2)	BMISI	2	MAINTENANCE	2013-01-03
5	101	Body mass index (kg/m2)	BMISI	3	TREATMENT	2013-04-03
6	101	Body mass index (kg/m2)	BMISI	3	TREATMENT	2013-07-03
7	101	Body mass index (kg/m2)	BMISI	3	TREATMENT	2013-10-03

CHOOSEN FUNCTION:

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.

Index-expression value is to select from the arguments that follow, selection-1 to selection-n are the numeric values from which to choose. If the first argument is negative, the function counts backwards from the list of arguments, and returns that value.

EXAMPLE:

```
data _null_;
  ITEMNUMBER = choosen(1,200,250,78,48,79);
  RANK       = choosen(-2,1,2,3,4,5);
  SCORE      = choosen(3,220,187,197,302,234);
  VALUE      = choosen(-5,-20,34,45,-90,45);
  put ITEMNUMBER=  RANK=   SCORE=   VALUE=  ;
run;
```

SAS writes the following line to the log: **ITEMNUMBER=200 RANK=4 SCORE=197 VALUE=-20.**

CONCLUSION:

These new functions are valuable additions to existing SAS functions. Efficient and timely usage of these functions makes the code elegant, simple and easy to maintain.

REFERENCES:

SAS Institute, Inc. SAS(R) 9.2 Language Reference: Dictionary, Fourth Edition. Online documentation:

IFC Function:
http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002604573.htm

IFN Function:
http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002604573.htm

CHOOSEC Function:
http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002604617.htm

CHOOSEN Function:
http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a002604619.htm

ACKNOWLEDGMENTS

I would like to thank my management and friends at GCE Solutions, for motivating me to write this paper and inspiring me to present at the conference.

CONTACT INFORMATION

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

Name: Geetha Kesireddi
Enterprise: GCE Solutions
Work Phone: 040-40273548
Email: geetha.kesireddy@gcesolutions.com
Website: https://gcesolutions.com

SAS® and all other SAS®

Institute Inc. product or service names are registered trademarks or trademarks of SAS®
Institute Inc. in the USA and other countries. ®indicates USA registration.

Other brand and product names are trademarks of their respective companies.