

A Macro for checking incorrect format of time variable

Lanlan PENG, Sanofi-Aventis, Chengdu, China

ABSTRACT

With the development of scientific and technological progress, the data analysis has become important part of the clinical trial. Usually in data analysis process, the date/time variable plays an indispensable role, but sometimes these data collected have incorrect format, which will affect the accuracy of analysis results. Therefore, the checking of incorrect format is an important step in the data analysis process.

Traditionally, these incorrect data format would be found out by visual check. The disadvantage of this method is that it's either time consuming or prone to errors. For this reason, we need to create a Macro to check automatically the incorrect format instead of visual check.

INTRODUCTION

In this paper, I propose a SAS Macro program which named %checktime to verify all date/time variables in different datasets, and record incorrect date/time variable format in the PDF file. Firstly, this paper demonstrates a brief process overview of %checktime, after that it will present the principal steps which contain, how to find out and collect all date/time variables in only one dataset, and how to check the incorrect date/time format.

PROCESS OVERVIEW

1. The date/time variables will be searched in sashelp.vcolumn, which provides detailed information about each variable contained in all the datasets present in the SAS session. Then, we generate a dataset for summarizing all date/time variables. Process 1 details this step.
2. Once the dataset is created, previous calling the SET routine, and then using FETCH() function to pass the fetched value from the Data Set Data Vector (DDV) to the corresponding data step. This process will be explained later in process 2.
3. For checking out the incorrect date/time variable format, PRXMATCH() function and regular expression will be used, this step is the core of Macro %checktime. Process 3 details this step.
4. The final step is to create a report in PDF in order of the dataset which contains the incorrect date/time variable format, the variable name, the domain. Process 4 will show how to generate the report in PDF format.

PROCESS 1 <FINDING OUT ALL DATE/TIME VARIABLES>

Before checking the incorrect date/time variable format in target library, first of all, the naming rules for data/time variables should be knew. Then, the date/time variables will be searched in sashelp.vcolumn, which contains detailed information about each variable. For example, the SDTM date/time variables are ended with "DTC" and they represent in ISO8601 format. The code and the output are presented below (the variable "memname" is the name of datasets).

```
data prepl(keep=libname memname name);
  set sashelp.vcolumn;
  where libname = "Library name";
  if substr(reverse(compress(name)),1,3) = "CTD" then output;
run;
```

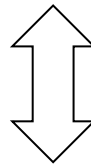
<A Macro for checking incorrect format of time variable>, continued

| libname | memname | name |
|--------------|---------|----------|
| Library name | AE_H | AESTDTC |
| Library name | AE_H | AEENDTC |
| Library name | AE_H | AEDTHDTC |
| Library name | APAE_H | AESTDTC |
| Library name | APAE_H | AEENDTC |
| Library name | CM_H | CMSTDTC |
| Library name | CM_H | CMENDTC |

Figure 1: All date/time variable in the SAS session

Once the output is produced, I make use of the RETAIN statement to retain the first observation value to the next that means that, we put the variable of same memname in same row and separate them by space. And then, the last record for same memname will be selected. The purpose of this step is to reduce the number of times the same dataset will be opened.

| libname | memname | name | var |
|--------------|---------|----------|--------------------------|
| Library name | AE_H | AESTDTC | AESTDTC |
| Library name | AE_H | AEENDTC | AESTDTC AEENDTC |
| Library name | AE_H | AEDTHDTC | AESTDTC AEENDTC AEDTHDTC |
| Library name | APAE_H | AESTDTC | AESTDTC |
| Library name | APAE_H | AEENDTC | AESTDTC AEENDTC |
| Library name | CM_H | CMSTDTC | CMSTDTC |
| Library name | CM_H | CMENDTC | CMSTDTC CMENDTC |



| libname | memname | name | var |
|--------------|---------|----------|--------------------------|
| Library name | AE_H | AEDTHDTC | AESTDTC AEENDTC AEDTHDTC |
| Library name | APAE_H | AEENDTC | AESTDTC AEENDTC |
| Library name | CM_H | CMENDTC | CMSTDTC CMENDTC |

Figure 2: Output

<A Macro for checking incorrect format of time variable>, continued

```
data prep2;
  set prep1;
  length var $200.;
  by memname;
  retain var;
  if first.memname then var = name;
  else var = strip(var)||" "||name;
run;
data prep3;
  set prep2;
  by memname;
  if last.memname;
run;
```

Figure 3: RETAIN statement

PROCESS 2 <USING FETCH() FUNCTION TO FETCH VALUES>

Before using FETCH() function for reading the next non-deleted observation from dataset(Figure 2) into the Data Set Data Vector (DDV), the SET routine must be called preciously, the values will be automatically passed from the DDV to the corresponding DATA step. FETCH returns a 0 if the operation is successful, not equal to 0 if it isn't successful. (Figure 4 below)

```
%let dsid = %sysfunc(open(prepare3));
%local libname memname name var;
%if &dsid %then %do;
  %syscall set(dsid);
  %do i = 1 %to &count.;
    %let rc = %sysfunc(fetch(&dsid.));
    %if &rc. = 0 %then %do;
      data time_1_&i(keep=VARIABLE DOMAIN_NAME ISSUE);
        length DOMAIN_NAME $10.;
        set &libname..&memname. ( keep=&var. );
    %end;
  %end;
```

Figure 4: FETCH function

PROCESS 3 <CHECKING THE INCORRECT DATE/TIME VARIABLE FORMAT>

As the date/time variables have multiple different substrings, it's difficult to link several INDEX function call together with OR conditions to match a correct date/time variable format. For this reason, we apply a conventional method which combines PRXMATCH function with Perl regular expression.

What is the PRXMATCH function?

Syntax:

```
PRXMATCH (pattern-id or regular-expression, source)
```

It searches source with the Perl regular expression and returns the position at which the string begins. If there is no match, PRXMATCH returns a zero.

What is the Perl regular expression?

The Perl regular expressions consist of a string of characters and special characters that are called metacharacters, which describes characteristics of a piece of text, for example, phone number, ID number and so on. The table below will display some metacharacters that be used for creating the Macro %checktime (Figure 5).

<A Macro for checking incorrect format of time variable>, continued

| Special characters | Signification |
|--------------------|---|
| \w | matches any of the characters a-z, A-Z, 0-9, or the underscore |
| \d | matches the numbers 0 to 9 only |
| \s | matches any whitespace character, including the space and the tab |
| ? | matches exactly 0 or 1 occurrences of the preceding pattern |

Figure 5: Metacharacters

On the basis of PRXMATCH() function and Perl regular expression, the correct date/time variable format will be defined, and then use IF-THEN/ELSE statement to find out the incorrect date/time variable format. Therefore, 5 types correct date/time variable format are summarized:

If the format for date/time variable is a partial date/time

- Date/time variables present the year (e.g. 2014)
- Date/time variables present the year, month (e.g. 2014-01)
- Date/time variables present the year, month, day (e.g. 2014-01-21)

If the format for date/time variables is complete date/time

- Date/time variable present the year, month, day, hour, minute (e.g. 2014-01-21T13:14)
- Date/time variable present the year, month, day, hour, minute and second (e.g. 2014-01-21T13:14:23)

According to these correct data formats, I can display them by Perl regular expression (Figure 6):

| Date/time variable | Length | Perl regular expression |
|---------------------|---------------|--|
| 2014 | 4 | \d{4} |
| 2014-01 | 7 | \d{4}-\d{2} |
| 2014-01-21 | 10 | \d{4}-\d{2}-\d{2} |
| 2014-01-21T13:14 | >10 and <= 19 | \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}?\d{2}? |
| 2014-01-21T13:14:23 | | |

Figure 6: Perl regular expression

The detailed code is showed below (Figure 7), it presents that if the length of date/time variable “var2 (j)” equal to 4, the PRXMATCH function searches \d{4} in var2 (j), if it matches, variable “cf” equal to the position at which \d{4} begins, else variable “cf” equal to -1 or 0. The IF-THEN/ELSE statement is used to select the incorrect date/time variable format which named “ISSUE”.

```

if length(var2(j)) = 4 then cf = prxmatch("/\d{4}/", var2(j));
else if length(var2(j)) = 7 then cf = prxmatch("/\d{4}-\d{2}/", var2(j));
else if length(var2(j)) = 10 then cf = prxmatch("/\d{4}-\d{2}-\d{2}/", var2(j));
else if 10 < length(var2(j)) =< 19 then cf = prxmatch("/\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}?\d{2}?", var2(j));
else cf = -1;
if cf <= 0 then ISSUE = var2(j);
if not missing(ISSUE) then output;

```

Figure 7: Perl regular expression

PROCESS 4 <GENERATING THE RESULT IN PDF FORMAT>

Once the all incorrect date/time variable formats are found out, the final step is to merge them in only one dataset that named time_issue. Then, the ODS PDF statement should be used to generate the result in PDF format. For clarifying

<A Macro for checking incorrect format of time variable>, continued

more detailed the result, two variables “domain_name” and “variable” will be added, one present the dataset name, it means where the incorrect date/time variable formats come from, the other one is the date/time variable name.

```
data time_issue;
set
  %do j=1 %to &count. ;
    time_1_&j
  %end;
;
run;
ods pdf file="c:/QC/OUTPUT/checkissue.pdf";
proc print data=time_issue;
  var DOMAIN_NAME VARIABLE ISSUE;
run;
ods pdf close;
```

Figure 8: Merge datasets

The final result in PDF format is presented below (Figure 9):

| Obs | DOMAIN_NAME | VARIABLE | ISSUET |
|-----|-------------|----------|----------|
| 1 | CM_H | CMSTDTC | 2017--11 |
| 2 | CM_H | CMSTDTC | 2008--15 |
| 3 | CM_H | CMSTDTC | 2011--01 |
| 4 | MH_H | MHSTDTC | --11 |
| 5 | MH_H | MHENDTC | --11 |

Figure 9: Final result in PDF

CONCLUSION

In summary, this Macro can be used to automate the process of checking the incorrect date/time variable format, hence eliminating the time consuming and manual process. But it still needs to be improved, because it can't check the logical error of date/time variable values, and it needs to use unified date/time variable formats, like ISO 8601 format. And if we don't know the naming rules for time variables, it will be difficult to find all date/time variables in sashelp.vcolumn.

REFERENCES

SAS 94 Online Doc

SDTM IG Version 3.2

SAS Support

CONTACT INFORMATION

Your comments and question are valued. Contact the author at:

Name: Lanlan PENG
Phone: (+86)15178819923
Email: Lanlan.peng92@gmail.com