

How to modify a variable's length without changing its position in dataset

Yucheng Pan, Pfizer (China) Research and Development Co.,Ltd. China

ABSTRACT

When working with SAS datasets, there may be times when programmers need to modify the length of variables. The LENGTH statement is often used in SAS data steps to modify the length of existing variables and these variables listed in the LENGTH statement will be moved to front position as well. Although variable's position won't affect any further data steps, it can make the dataset look unorganized and difficult to read.

In this article, we will explore several methods and provide examples of how to modify a variable's length without changing its position in dataset.

INTRODUCTION

Modifying the length of a variable is a common task when working with SAS datasets. This may be necessary when programmers need to increase the length of a character variable to avoid data truncation.

Below dataset is a sample dataset which collects the information of electrocardiogram test results. The variable EGSTRESU shows the standard unit for the corresponding test and in this case the value should be "beats/min". However the default length of variable EGSTRESU is set to 8 which truncated the value "beats/min" into "beats/mi".

STUDYID	DOMAIN	USUBJID	EGTESTCD	EGTEST	EGORRES	EGSTRESC	EGSTRESU	EGDTC
A1234567	EG	10011001	EGHRMN	ECG Mean Heart Rate	85	85	beats/mi	2023-01-01

Display 1. Sample raw dataset

By solving this problem, we can use the LENGTH statement prior to the SET statement to modify the length of variable EGSTRESU.

Example:

```
data eg1;
  length egstresu $9.;
  set eg;
  egstresu="beats/min";
run;
```

EGSTRESU	STUDYID	DOMAIN	USUBJID	EGTESTCD	EGTEST	EGORRES	EGSTRESC	EGDTC
beats/min	A1234567	EG	10011001	EGHRMN	ECG Mean Heart Rate	85	85	2023-01-01

Display 2. Sample output dataset

The value of variable EGSTRESU is corrected now, but the LENGTH statement also moved the variable EGSTRESU to the first position, which makes the dataset look unorganized and difficult to read. The LENGTH statement can move all the variables listed to front, and variables not listed in the LENGTH statement retain their origin position.

In this article, we will explore four methods to modify the variable's length without changing its position in dataset and provide examples and tips to help you achieve this task efficiently and effectively.

METHODS

All the methods provided in this article can achieve desired result, which may vary depending on the data structure and your needs.

METHOD 1

Since the LENGTH statement will move every variable listed within to front position, why don't we just name all the variables in the LENGTH statement? If all the variables are moved to front, it means no variables are moved at all. Problem solved!

Example:

```
data eg1;
  length studyid $8 domain $2 usubjid $8 egtestcd $6 egtest $19
        egorres $2 egstresc $2 egstresu $9 egdtc $10;
  set eg;
  egstresu="beats/min";
run;
```

STUDYID	DOMAIN	USUBJID	EGTESTCD	EGTEST	EGORRES	EGSTRESC	EGSTRESU	EGDTC
A1234567	EG	10011001	EGHRMN	ECG Mean Heart Rate	85	85	beats/min	2023-01-01

Display 3. Sample output dataset

METHOD 2

Alternatively, we can use the RETAIN statement combined with the LENGTH statement.

The RETAIN statement is most often used to reorder variables simply because no other variable attribute specifications are required. Similar with the LENGTH statement, only the variables listed will be moved, the variables not listed in the RETAIN statement retain their original position.

Remember to use the RETAIN statement prior to the SET, MERGE, UPDATE statement in order to reorder the variables.

Example:

```
data eg1;
  retain studyid domain usubjid egtestcd egtest egorres egstresc
        egstresu egdtc;
  length egstresu $9;
  set eg;
  egstresu="beats/min";
run;
```

STUDYID	DOMAIN	USUBJID	EGTESTCD	EGTEST	EGORRES	EGSTRESC	EGSTRESU	EGDTC
A1234567	EG	10011001	EGHRMN	ECG Mean Heart Rate	85	85	beats/min	2023-01-01

Display 4. Sample output dataset

But what if there are too many variables in the dataset? Let's say over 100 variables, then it will become too complex and tedious to write each of them in the RETAIN statement. Fortunately, we can use SASHELP.VCOLUMN (DICTIONARY.COLUMNS or PROC CONTENTS also works) to generate a metadata dataset which contains the attributes for each variable. Below dataset shows partially how it looks like:

Library Name	Member Name	Member Type	Column Name	Column Type	Column Length	Column Position	Column Number in Table
WORK	EG	DATA	STUDYID	char	8	0	1
WORK	EG	DATA	DOMAIN	char	2	8	2
WORK	EG	DATA	USUBJID	char	8	10	3
WORK	EG	DATA	EGTESTCD	char	6	18	4
WORK	EG	DATA	EGTEST	char	19	24	5
WORK	EG	DATA	EGORRES	char	2	43	6
WORK	EG	DATA	EGSTRESC	char	2	45	7
WORK	EG	DATA	EGSTRESU	char	8	47	8
WORK	EG	DATA	EGDTC	char	10	55	9

Display 5. Sample metadata dataset

This dataset is already sorted by every variable's origin position by default, hence we can use PROC SQL statement to read all the variables' names into a macro variable, which can be used in the RETAIN statement later.

Example:

```
proc sql;
    select name into:variable_list separated by " "
    from sashelp.vcolumn
    where libname="WORK" and memname="EG"
    ;
quit;

data eg1;
    retain &variable_list.;
    /*Macro variable variable_list resolves to STUDYID DOMAIN USUBJID EGTESTCD
    EGTEST EGORRES EGSTRESC EGSTRESU EGDTC*/
    length egstresu $9;
    set eg;
    egstresu="beats/min";
run;
```

STUDYID	DOMAIN	USUBJID	EGTESTCD	EGTEST	EGORRES	EGSTRESC	EGSTRESU	EGDTC
A1234567	EG	10011001	EGHRMN	ECG Mean Heart Rate	85	85	beats/min	2023-01-01

Display 6. Sample output dataset

METHOD 3

If you just need to modify the length of one specific variable (or several consecutive variables) and you happen to know the name of the last variable in the dataset as well, you can use the trick of double SET statements.

For any existed character variables, the LENGTH statement is only effective prior to the SET statement. But in some scenario the LENGTH statement can be placed after the SET statement, this only works for new variables and these new variables will be placed in the last position. The double-dash in the DROP statement refers to specify all the variables between these two based on the position of variables as they appear in the dataset.

Example:

```
data eg1;
    set eg(drop=egstresu--egdtc);
```

```

length egstresu $9;
set eg;
egstresu="beats/min";
run;

```

STUDYID	DOMAIN	USUBJID	EGTESTCD	EGTEST	EGORRES	EGSTRESC	EGSTRESU	EGDTC
A1234567	EG	10011001	EGHRMN	ECG Mean Heart Rate	85	85	beats/min	2023-01-01

Display 7. Sample output dataset

METHOD 4

Last but not least, we can use PROC SQL statement to solve this. PROC SQL in SAS sometimes can be very powerful to deal with the data compared to SAS data steps.

The ALTER TABLE statement can add, modify and delete columns in existing tables. You can use the MODIFY clause to change the length, informat, format and label of a column. The best part about MODIFY is that it won't change the variable's position in the dataset.

The following code shows how to modify and update the EGSTRESU column.

Example:

```

data eg1;
  set eg;
run;

proc sql;
  alter table eg1
  modify egstresu char(9)
  ;

  update eg1
  set egstresu="beats/min"
  ;
quit;

```

STUDYID	DOMAIN	USUBJID	EGTESTCD	EGTEST	EGORRES	EGSTRESC	EGSTRESU	EGDTC
A1234567	EG	10011001	EGHRMN	ECG Mean Heart Rate	85	85	beats/min	2023-01-01

Display 8. Sample output dataset

CONCLUSION

There are several methods for modifying a variable's length without changing its position in dataset. You can control the order in which variables are displayed in SAS output by using the LENGTH statement or RETAIN statement, while using the SQL procedure allows you to change column attributes as well. You can choose the method that best suits your needs and workflow to achieve the results more efficiently.

REFERENCES

Examples: Reorder and Align Variables

Available at

https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/lepg/p0zwi72ea2v9o8n1lermkpu2z3mu.htm

Altering Columns

Available at

https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/sqlproc/p0rdzn3mhqxl2zn13uvxkjt1svg.htm

CONTACT INFORMATION

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

Name: Yucheng Pan

Enterprise: Pfizer (China) Research and Development Co.,Ltd. China

E-mail: Yucheng.Pan@pfizer.com

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