

Get in line! How to make your data alignment easy!

Ying (Evelyn) Guo, PAREXEL International, Durham, North Carolina

ABSTRACT

On data-driven output, we strive for alignment of data that allows for ease of analysis and greater readability. While some outputs need left- or right-alignment, others need decimal-alignment or even alignment with brackets. To make the outputs visually pleasing, column centering is usually preferred, requiring that final datasets are lined up as desired. In order to achieve this, spaces are added to each column. When a column has values of various types, such as with a table having means, standard deviations, and ranges, consistent alignment is especially time-consuming. Instead of manually adding spaces for each value in that column through trial-and-error, the SAS® macro program which I developed automatically lines up the values in each column according to a user-specified character. It can automatically search values in a given column, identify the location of the specified character, and add necessary spacing before or after the value to meet the alignment rules. The use of this macro will significantly decrease the programming time spent on the aesthetics of outputs thus increasing table, figure and listing (TFL) production efficiency.

INTRODUCTION

Standardized and readable output is critical especially for those which are being sent for review to an external committee or the Food and Drug Administration (FDA). In most cases, programmers spend a lot of time preparing the final report. Manipulation of final datasets is necessary to ensure readability. Alignment is the one of the biggest challenges in outputting the final report. Sometimes, the variables need to be aligned by the decimal place other times, the values need to be aligned based on the bracket placement. In order to make the outputs visually pleasing, column centering is usually preferred, which makes alignment more difficult. When a column has values of various types, such as a demographic summary table, where the value can include mean, median and range, appropriate alignment can be a time-consuming task.

CURRENT ISSUE

To achieve the proper alignment, spaces are added in the value. The programmer needs to review the entire data first and then determine how many spaces need to be added for each value. The programmer will then classify the data into several categories, create a rule for each category, and then apply the rules to each data point by category. For example, in order to align by decimal, the programmer needs to calculate the maximum length of each value before the decimal place. If the maximum length is four (showed in Figure 1), then the programmer would add three spaces for values of 0 up to, but not including, 10. They would add two spaces for values of 10 up to, but not including, 100. And finally would add one space for values of 100 up to, but not including, 1000. No space would need to be added for values of more than 1000. The programmer could write a SAS code to accomplish this, but many have found it to be a very time-consuming, especially for large datasets with complicated value structures. In addition, when the data change and exceed the previous maximum length, this process needs to be modified to align with the new data structure. This will cause a lot of rework during the re-execution of the program (Figure 2 showed process for this task). Many programmers will agree that aligning values in a timely manner with little to no modification during rerun is often a challenge.

SOLUTION

In order to make the alignment process more efficient, and increase the flexibility of program to allow for versatile data, a SAS macro has been developed. This macro will automatically locate the special character using index function. This function can return a value with the position of the aligned character. It also calculates the length for each value and calculates the maximum length of the value. Then, it calculates the spaces need added to the value,

Get in line! How to make your data alignment easy!, continued

and automatically add the appropriate number of space(s) which are needed for perfect alignment within the report (Figure 3).

```
/*identify the position of the special character*/;
data test;
  set &indata;
  cpos=index(test, "%str(&char)");
  olen=length(test);
run;

/*define the maximum length of the variable*/;
proc sql;
  create table test1 as
  select test, cpos, olen, max (cpos) as maxpos
  from test;
quit;
```

When using this solution, there will be no need to manually calculate the spaces for each value. Even after the data is modified, the macro will automatically re-apply the correct alignment, which means there is no more need to modify the program after each data refresh (Figure 4).

Term	Value
Number (%) of Subjects With At Least One Disease/Synd	1053 (43.1)
Blood and lymphatic system disorders	39 (31.7)
Anaemia	139 (31.7)
Gastrointestinal disorders	3 (2.4)
Nausea	3 (2.4)
Vomiting	1 (0.8)
Infections and infestations	28 (22.8)
Body tinea	3 (2.4)
Infection parasitic	23 (18.7)
Urinary tract infection	9 (7.3)
Nervous system disorders	2 (1.6)
Dizziness	2 (1.6)
Skin and subcutaneous tissue disorders	1 (0.8)
Rash papular	1 (0.8)
UNCODED	1 (0.8)
Cough Xy52	1 (0.8)

Figure 1 Unaligned outputs

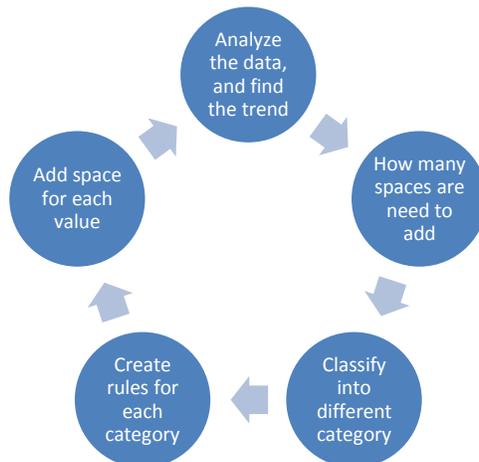


Figure 2 Traditional process

Get in line! How to make your data alignment easy!, continued

Term	Value
Number (%) of Subjects With At Least One Disease/Synd	1053 (43.1)
Blood and lymphatic system disorders	39 (31.7)
Anaemia	139 (31.7)
Gastrointestinal disorders	3 (2.4)
Nausea	3 (2.4)
Vomiting	1 (0.8)
Infections and infestations	28 (22.8)
Body tinea	3 (2.4)
Infection parasitic	23 (18.7)
Urinary tract infection	9 (7.3)
Nervous system disorders	2 (1.6)
Dizziness	2 (1.6)
Skin and subcutaneous tissue disorders	1 (0.8)
Rash papular	1 (0.8)
UNCODED	1 (0.8)
Cough Xy52	1 (0.8)

Figure 3 Aligned outputs

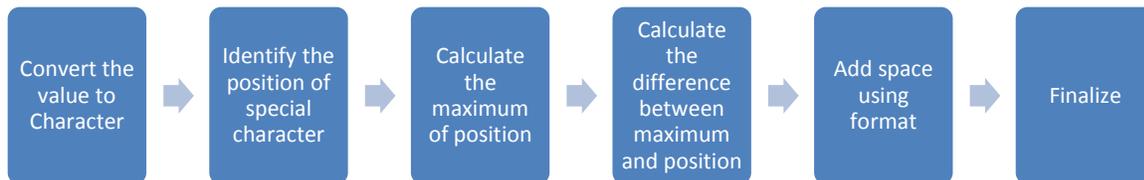


Figure 4 New process

SUMMARY

In summary, this macro is a very useful tool to align the variable output efficiently. It can align output by any specified character. In addition, the macro will perform automatically, regardless of different structures and length for variables, therefore, it also can be rerun on new data cut without modification. This macro will increase the efficiency of daily work, and also reduce potential rework.

CONCLUSION

This macro is simple and practical. It will be a very useful tool for programmers who want to align data quickly and accurately.

CONTACT INFORMATION

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

Name: Ying Guo
Enterprise: PAREXEL International
Address: 2520 Meridian Parkway, Suite 200
City, State ZIP: Durham NC 27713
Work Phone: 1-978-311-1811
E-mail: evelyn.guo@parexel.com
Web: www.pxl.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 registered trademarks or trademarks of their respective companies.

APPENDIX

```
%macro lineup(char=., lspace=3, indata=);

  /*format space*/;
  proc format;
    value space
      0 = 'aa'
      1 = 'a a'
      2 = 'a  a'
      3 = 'a   a'
      4 = 'a    a'
      5 = 'a   a'
      6 = 'a    a'
      7 = 'a   a'
      8 = 'a    a'
      9 = 'a   a'
     10 = 'a    a';
  run;

  /*identify the position of the special character*/;
  data test;
    set &indata;
    cpos=index(test, "%str(&char)");
    olen=length(test);
  run;

  /*define the maximum length of the variable*/;
  proc sql;
    create table test1 as
      select disprft1, test, cpos, olen, max (cpos) as maxpos
      from test;
  quit;

  data test2;
    set test1;
    lspace=&lspace;
    diff=maxpos-cpos;
    spa=diff+lspace;
  run;

  data test3;
    set test2;
    testf=strip(put(spa, space.))||strip(test);
    testff=tranwrd(testf, 'a', ' ');
  run;

%mend;
```