Paper CC-24

How Many Licks to the Center of that Column?

Spencer Childress, Rho®, Inc., Chapel Hill, NC

ABSTRACT

Sometimes centering statistics just does not quite present data the way it should. Scanning down a column of means, ranges, p-values, et cetera takes more effort than it should due to alignment issues. What if all these stats could be aligned more readably? This article aims to solve this question by presenting a dynamic algorithm to align statistics along a vertical column, on a common integer or punctuation mark across any given statistic. As input, the algorithm needs the column width in order to properly indent the data. Additionally, to display each column alike, it needs the number of columns. The techniques presented herein offer a good overview of basic data step and array programming appropriate for all levels of SAS® capability. While this article targets a clinical computing audience, the techniques apply to a broad range of computing scenarios.

INTRODUCTION

As an alternative to centering statistics, the algorithm presented herein aims to align statistics on a common column of text, for enhanced readability. In order to align text, three assumptions come into play:

Assumption #1: The display outputs a monospaced font, such as Courier New.

A monospaced font is a font where every character occupies the same amount of space. That is, each character is the same height and width. Such fonts simplify alignment since width of text is a function of the number of characters, in both SAS and the output file.

Assumption #2: Column of alignment defaults to the ones digit of the first number in a statistic.

Given a monospaced font, the next order of business in aligning text is finding the column of alignment. For the purposes of this article, the ANYPUNCT function will determine the position of punctuation, e.g. the period, comma, parenthesis, hyphen, etc. ANYPUNCT returns the position of the first punctuation mark. The text just before the punctuation mark corresponds to the ones digit of the first number in a statistic, the column of alignment.

Assumption #3: Statistics never contain more than a single space, consecutively.

The third assumption merely limits the number of conditions the algorithm accounts for, for simplicity's sake.

COMPARISON

Centered statistics look like so:

Characteristics	Boring N=9	Drugs! N=10
Age at screening (years)		
n	9	10
Mean	36.3	36.7
SD	7.16	6.53
Median	35.0	38.5
Range (Min, Max)	(23, 44)	(27, 47)
Ethnicity - n (%)		
Hispanic or Latino	1 (11.1)	0
Not Hispanic or Latino	8 (88.9)	10 (100.0)
Not Reported	0	0
Unknown	0	0

Figure 1 Centered Table

Mean, standard deviation, and median look like a block of text, not easily readable. While centered table text has an appealing symmetry, it lacks the functionality alignment has.

Aligned statistics look like so:

Characteristics	Boring N=9	Drugs! N=10
Age at screening (years)		
n	9	10
Mean	36.3	36.7
SD	7.16	6.53
Median	35.0	38.5
Range (Min, Max)	(23, 44)	(27, 47)
Ethnicity - n (%)		
Hispanic or Latino	1 (11.1)	0
Not Hispanic or Latino	8 (88.9)	10 (100.0)
Not Reported	0	0
Unknown	0	0

Figure 2 Aligned Table

Alignment splits the table text into two parts, the number to the left of the punctuation mark, and the number to the right, if there is one. This orientation gives a more immediate sense of magnitude.

METHOD

In order to align text, the width of the table column(s) is needed. For this explanation, a width of 20 is assumed. A proper place to position the statistic is around the midpoint of a cell, which with a width of 20 is between the 10th and 11th characters in that cell. For appearance's sake, the algorithm rounds down, to the 10th spot. Beginning at the left side of the cell, the statistic can then be padded with spaces until positioned correctly. Thus, for a single digit integer, say 5, the statistic would be padded with nine spaces so the 5 is positioned as the 10th character in the cell. For a four-digit integer, say 9999, the last nine would occupy the 10th character in the cell.

CONCLUSION

A little additional programing can improve the readability and appearance in statistical displays. Instead of relying on built in text justification, such as centering, text can be programmatically positioned for optimal results. The algorithm, presented in a transportable macro in the appendix, takes three arguments: the input dataset, &DS; the column(s) to be aligned, &COLS; and the table column width, &WIDTH.

CONTACT INFORMATION

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

Spencer Childress
Rho, Inc
6330 Quadrangle Dr., Ste. 500
Chapel Hill, NC 27517
Phone: 919-595-6638
Fax: 919-408-0999
Email: Spencer_Childress@rhoworld.com
Web: www.rhoworld.com

APPENDIX

```
%macro align(ds    = ,
             cols  = ,
             width = );

  data &ds._aligned;
    set &ds;
```

```
    array cols (*) &cols;

    do i = 1 to dim(cols);
      *Strip variables to reset position.;
                     cols(i) = strip(cols(i));
      *Pad statistic with spaces.;
      *For statistics which contain a punctuation mark.;
      if anypunct(cols(i)) then do;
          *For statistics whose first character is a puncutation mark, search for
second punctuation mark.;
               if anypunct(substr(cols(i), 1, 1))
                  then cols(i) = repeat(' ', floor(&width/2) - anypunct(cols(i),
anypunct(cols(i)) + 1) - 1) || cols(i);
          *For statistics with a space immediately preceding punctuation mark, e.g. of
N (%) form.;
          else if substr(cols(i), anypunct(cols(i)) - 1, 1) = ' '
                  then cols(i) = repeat(' ', floor(&width/2) - anypunct(cols(i))) ||
cols(i);
          *For all other statistics which contain a punctuation mark.;
          else         cols(i) = repeat(' ', floor(&width/2) - anypunct(cols(i)) - 1)
|| cols(i);
      end;
      *For integers.;
      else             cols(i) = repeat(' ', floor(&width/2) - length(cols(i)) - 2) ||
cols(i);
    end;
  run;

%mend align;
```