Using the DATA Step to Create Bar Charts: The ODS Report Writing Interface

Randy Herbison, Westat

ABSTRACT

Introduced in SAS® 9.0, the ODS Report Writing Interface is an object-oriented addition to the DATA step. The ODS Report Writing Interface brings the full power of the Output Delivery System to the rich DATA step programming environment.

Many SAS programmers know that the ODS Report Writing Interface can be used to create tables and text. It is less well-known that the ODS Report Writing Interface can also be used to create certain types of graphics. This paper will explain how the ODS Report Writing Interface can be used to create horizontal bar charts. It will cover how to use text formatting methods to display text, how to use table methods to create horizontal bars, and how to use gridded and absolute layout methods to manage the entire arrangement.

This paper is suitable for intermediate and advanced SAS programmers.

INTRODUCTION

Available experimentally in Version 9.1 and preproduction in Version 9.2 TS Level 3MO, the ODS Report Writing Interface is a Component Object that provides the DATA step with an object-oriented interface to the Output Delivery System (ODS). It enables you to combine the rich DATA step programming environment with the full capabilities of ODS to create both tabular and non-tabular output. Using the ODS Report Writing Interface, you can take advantage of DATA step programming features such as arrays, by-group processing, and conditional logic while using ODS features such as colors, proportional fonts, and layout management.

The ODS Report Writing Interface has much to offer, including ease of use, but there are three things that make it especially useful for creating some types of horizontal bar charts. It enables you to build tables one cell at a time and to customize the table cells to create high-quality horizontal bars. It provides the ability to display high-quality text, and it offers layout tools that make it easy to combine relatively large amounts of text and tables (or bars).

The survey questions and composite labels used in this paper are from the **Nursing Home Survey on Patient Safety Culture** sponsored by the U. S. Department of Health and Human Services**,** Agency for Healthcare Research and Quality. The survey toolkit is available free to the public via the AHRQ's website at http://www.ahrq.gov/qual/patientsafetyculture/nhsurvindex.htm. No real survey response data have been used anywhere in this paper.

For this paper, we'll use the ODS Report Writing Object to create 2 types of horizontal bar charts: 1) A stacked horizontal bar chart with equal length bars; and 2) A simple horizontal bar chart with varying length bars.

The ODS Report Writing Interface supports the HTML, Printer (PDF, PS, PCL), and RTF destinations.

Both example bar charts were designed for the PDF destination. Some ODS features used to create the example bar charts are supported only by the PDF destination.

Getting Started With The ODS Report Writing Interface

Since the ODS Report Writing Interface is an object-oriented Component Object, using it is unlike traditional DATA step programming. The DATA step Component Interface enables you to use statements and methods to create and manipulate the ODSOUT object. You use DATA step object dot notation to access the ODSOUT object's methods. You create an ODSOUT object during the execution of the DATA step. Consequently, you may need to use programming logic to ensure the object gets created only once. While an instance of the ODSOUT object exists, you access ODS features by calling the object's methods.

Use a DECLARE statement to declare the ODSOUT class.

```
DECLARE ODSOUT  objectName;
```

or

```
DCL ODSOUT  objectName;
```

Although both statements declare the ODSOUT class, neither actually creates an instance of an ODSOUT object. You can use the _NEW_ operator to create (instantiate) an object.

```
objectName=_NEW_ ODSOUT( );
```

An easier way to declare and create an ODSOUT object combines the DECLARE statement with the functionality of the _NEW_ operator.

```
DCL ODSOUT objectName( );
```

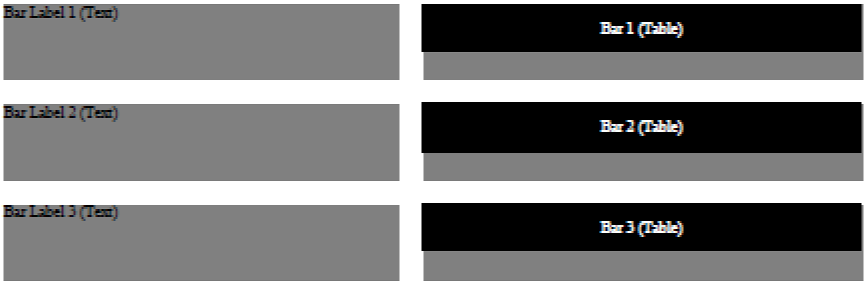
After the ODSOUT object has been created, you use its methods to perform actions. The general syntax of a method call is

```
objectName.method( < method arguments > );
```

Where “method” is the action to be performed, and “method arguments “ are additional information that the method understands how to use.

Using the ODS Report Writing Interface to Create Horizontal Bar Charts

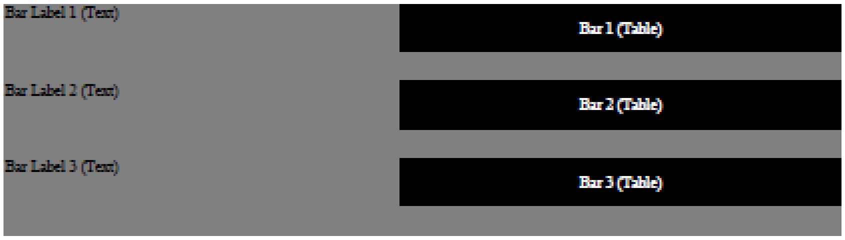
We'll use the ODS Report Writing Interface's text formatting, table, and layout management methods as the basic building blocks for the bar charts. The FORMAT_TEXT method will enable us to specify the color, justification, and font used for displaying the survey questions as bar labels. The table methods will allow us to use tables as bars. Using the table method FORMAT_CELL, we can control individual cell widths, and we can selectively inhibit any cell border. The LAYOUT_GRIDDED method will provide an easy way to control the arrangement of text and bars in a 2-dimensional row and column structure.



When using a layout method, we must use the REGION method to create regions as output containers for the bar labels and bars. The regions above are colored gray (and partlyly covered by the bars in column 2). For the bar labels, we can use the entire region to display one or more lines of survey question text per bar. This will allow us to display a variable number of lines of text per bar. Setting the region height greater than the bar height allows the text height to exceed the bar height. The code used to create the structure above is below.

```
data _null_;
  dcl odsout ods();

  * Start a 2-column layout;
  ods.layout_gridded(columns: 2);

  * Create 3 bars labels and bars;
  do i=1 to 3;

    * Create the region for the bar label;
    ods.region(width: '45pct',
               height: '.7in',
               overrides: 'backgroundColor=gray');

    * Display the bar label;
    ods.format_text(text: catx(' ', 'Bar Label', i, '(Text)'));

    * Create the region for the bar;
    ods.region(width: '50pct',
               height: '.7in',
               overrides: 'backgroundColor=gray');

    * Display the bar;
    ods.table_start(overrides: 'width=100pct
                                backgroundColor=black');
      ods.row_start();
        ods.format_cell(text: catx(' ', 'Bar', i, '(Table)'),
                        overrides: 'height=.4in
                                    color=white
                                    vjust=m
                                    just=c');
      ods.row_end();
    ods.table_end();
  end;


  * End the layout;
  ods.layout_end();
run;
```

To maximize the number of bars that will fit on a page, we'll eliminate the row gutters between the rows. The region height, which is greater than the bar height, will leave adequate spacing between the bars.



Ideally, we would use the LAYOUT_GRIDDED method's optional ROW_GUTTER argument to eliminate the row gutters, and we would keep a small column gutter. But, as of SAS 9.2 TS Level 2M3, it seems that neither the COLUMN_GUTTER nor the ROW_GUTTER arguments can be used to make the gutters smaller than the defaults. As a workaround, we'll use a CELLSPACING override for the LAYOUT_GRIDDED method.

```
     ods.layout_gridded(columns: 2  overrides: 'cellspacing=0');
```

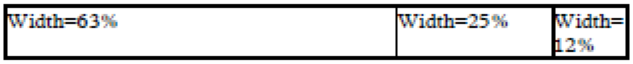
We'll lose the column gutter, but word wrapping will probably leave some space between the text and the bars (we might need to adjust the column widths).

Stacked Horizontal Bar Chart: Creating the Bar Segments

The FORMAT_CELL method's optional OVEERIDE argument allows us to specify a cell's background color, color (foreground), font, text justification, and width. We'll use each of these cell attributes to create the bar segments for the stacked bar chart, but let's begin with the WIDTH attribute. The WIDTH attribute makes the stacked bar chart possible by enabling us to creating cells of varying widths within a row.

```
data _null_;
  dcl odsout ods();

  ods.layout_gridded(columns: 1);

  ods.region(width: '50pct');
  ods.table_start(overrides: 'width=101pct  cellpadding=0 cellSpacing=0');
```

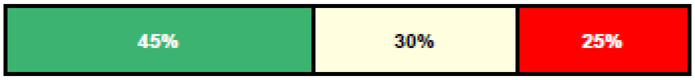
```
    ods.row_start();

        ods.format_cell(data: 'Width=63%', overrides: 'width=63pct');
        ods.format_cell(data: 'Width=25%', overrides: 'width=25pct');
        ods.format_cell(data: 'Width=12%', overrides: 'width=12pct');


    ods.row_end();
  ods.table_end();

  ods.layout_end();

run;
```



Width=63%	Width=25%	Width=12%

Now that we have seen how to create variable width cells, let's customize the colors, font, and text justification.

```
data _null_;

  input positive neutral negative;

  array segments {3} positive neutral negative;
  array segmentColors {3} $24 _temporary_ ('mediumseagreen','lightYellow','red');
  array segmentFontColors {3} $24 _temporary_ ('white', 'black', 'white');

  dcl odsout ods();

  ods.layout_gridded(columns: 1);

  ods.region(width: '55pct');
  ods.table_start(overrides: 'width=101pct  cellpadding=0 cellSpacing=0');

    ods.row_start();

    do i=1 to dim(segments);

        text=catt(segments{i},'%');
        segmentColor=segmentColors{i};
        segmentFontColor=segmentFontColors{i};

        attrs=catt('Width=',segments{i},'pct color=',segmentFontColor,
                   ' backgroundcolor=', segmentColor, ' just=c vjust=m',
                   ' fontFamily="ARIAL" fontWeight=bold fontSize=9pt',
                   ' height=.4in');

        ods.format_cell(data: text, overrides: attrs);

    end;

    ods.row_end();
  ods.table_end();
  ods.layout_end();
cards;
45 30 25
;
```

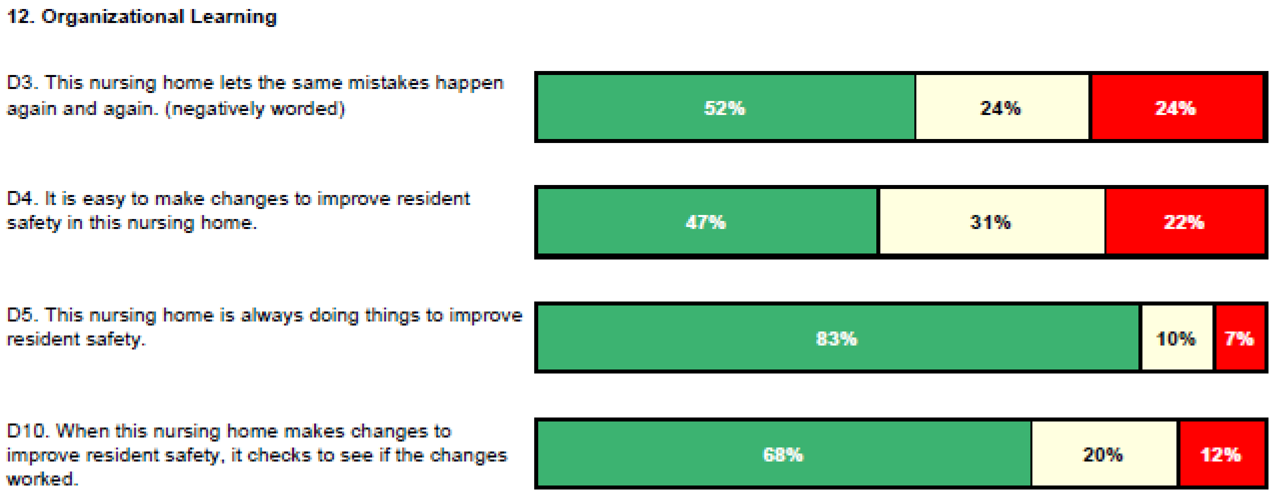


Adding the Text

We now have a nice stacked bar, but it's not very useful without a label to describe what it represents. The next step is to add the composite label and the question text as the bar label. We need to create 3 more regions and use the FORMAT_TEXT method to add all of the text. The first region will accommodate the composite text. The second region will be a placeholder for the second column. The third region will contain the question text, and the fourth region will contain the bars.

All data used in the following examples are included in the appendix.

```
data _null_;

  set bc(where=(composite='12')) end=_last;
  by composite notsorted;

  array segments {3} positive neutral negative;
  array segmentColors {3} $24 _temporary_ ('mediumseagreen','lightYellow','red');
  array segmentFontColors {3} $24 _temporary_ ('white', 'black', 'white');

  if _n_=1 then do;
    dcl odsout ods();
    ods.layout_gridded(columns: 2, overrides: 'cellspacing=0');
  end;

  if first.composite then do;
    ods.region(height: '.4in');
    ods.format_text(data: composite,  format: '$composite.',
                    overrides: 'just=left
                                fontFamily="ARIAL"
                                fontSize=9pt
                                fontWeight=bold');
    ods.region( );
  end;

  ods.region(width: '40pct', height: '.7in');

  ods.format_text(text: item,   format: '$question.',
                  overrides: 'just=left
                              fontFamily="ARIAL"
                              fontSize=9pt');
  ods.region(width: '55pct', height: '.7in');
  ods.table_start(overrides: 'width=101pct cellpadding=0 cellSpacing=0');

    ods.row_start();

    do i=1 to dim(segments);

      text=catt(segments{i},'%');
      segmentColor=segmentColors{i};
      segmentFontColor=segmentFontColors{i};

      attrs=catt('Width=',segments{i},'pct color=',segmentFontColor,
                 ' backgroundcolor=', segmentColor, ' just=c vjust=m',
                 ' fontFamily="ARIAL" fontWeight=bold fontSize=9pt',
                 ' height=.4in');

      ods.format_cell(data: text, overrides: attrs);

     end;

    ods.row_end();
  ods.table_end();

  if _last then ods.layout_end();
run;
```



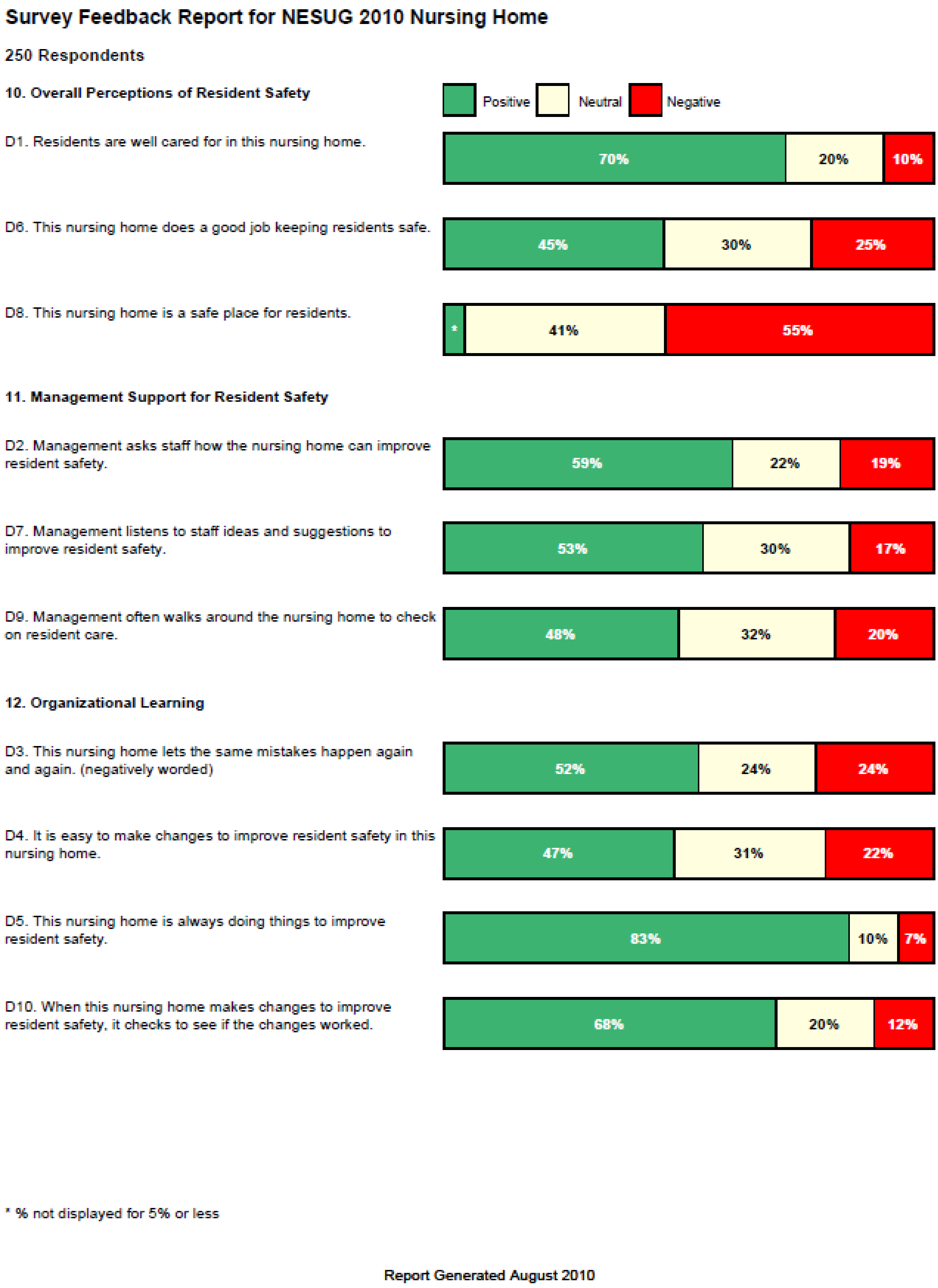
Completing the Stacked Horizontal Bar Chart Program

The remaining steps to completing the bar chart include creating the legend, adding a note, and adding titles and a footnote. As you may suspect, creating the legend is very similar to creating the bars. The differences include the number of cells and inhibiting the display of some cell borders. We'll use the FORMAT_TEXT method to add the titles and the note, and we'll use the LAYOUT_ABSOLUTE method to place the note exactly where we want it.

There are at least a few other issues we should address to make the SAS program more robust. We cannot assume that the sum of the segment values for each survey item will add to 100. Nor can we assume that every segment value will be non-missing or that it will be large enough to create a cell size wide enough to display the percentage. To address these issues, we'll add code to avoid creating a segment when the value is missing, to display an error message when segments do not sum to 100, and to display as asterisk as the segment value when the value is 5% or less.

```
%let nsname=NESUG2010 Nursing Home;
%let numerator=250;
%let month=%sysfunc(today(),monname15.);
%let year=%sysfunc(today(),year.);
%let titleAttrs=just=left fontFamily='ARIAL' fontWeight=bold;
%let footnoteAttrs=fontFamily='ARIAL' fontStyle=roman fontWeight=light fontSize=9pt
just=c;

options nocenter orientation=portrait nodate nonumber;

title;
footnote "^{style [&footnoteAttrs]%sysfunc(trim(%sysfunc(putc(F1,$footnote.))))
&month &year}";

ods escapechar='^' noptitle;
ods listing close;
ods pdf file='c:\temp\bc1.pdf';
data _null_;

  set bc(where=(composite in('10','11','12'))) end=_last;
  by composite notsorted;

  array segments {3} positive neutral negative;
  array segmentColors {3} $24 _temporary_ ('mediumseagreen','lightYellow','red');
  array segmentFontColors {3} $24 _temporary_ ('white', 'black', 'white');
  array legendLabels {3} $ 24 ('POSITIVE','NEUTRAL','NEGATIVE');

  if _n_=1 then do;
    dcl odsout ods();

      * Title 1;
    text=catx(' ', put('T1',$title.), "&nsname");
    ods.format_text(text: text, overrides: "&titleAttrs fontSize=12pt");

    * Blank line;
    ods.format_text(text: ' ');

    * Title 2;
    text=catx(" ", "&numerator", put('T2',$title.));
    ods.format_text(text: text, overrides: "&titleAttrs fontSize=10pt");
      * Start the absolute layout;
    ods.layout_absolute();
      * Start the gridded layout;
    ods.layout_gridded(columns: 2, overrides: 'cellspacing=0');
  end;

  if first.composite then do;
    ods.region(height: '.4in');
    ods.format_text(data: composite,  format: '$composite.',
                    overrides: 'just=left
                                fontFamily="ARIAL"
                               fontSize=9pt
                               fontWeight=bold');
    ods.region( );

      * Add the legend;
    if _n_=1 then do;

      attrs=catx(' ','height=.2in width=.5in fontFamily="ARIAL" fontSize=8pt',
                     'color=black just=center vjust=b');

      ods.table_start(overrides: 'borderWidth=.0025in
                                  cellpadding=0
                                  cellSpacing=0');
        ods.row_start();


        do i=1 to dim(segments);

          segmentColor=segmentColors{i};

          ods.format_cell(overrides: 'height=.2in width=.25in
                                      backgroundcolor=' ||
                                      segmentColor);

          legendLabel=put(legendLabels{i},$legend.-l);
          ods.format_cell(text: legendLabel, inhibit: 'TBR',
                            overrides: attrs);
        end;

        ods.row_end();
      ods.table_end();
    end;
  end;

  ods.region(width: '45pct', height: '.7in');

  ods.format_text(text: item,   format: '$question.',
                overrides: 'just=left fontFamily="ARIAL" fontSize=9pt');

  ods.region(width: '50pct', height: '.7in');

  * Make sure segments sum to 100;
  total=sum(of segments{*});
  if total=100 then do;
    ods.table_start(overrides: 'width=101pct  cellpadding=0 cellSpacing=0');

      ods.row_start();

      do i=1 to dim(segments);

        value=segments{i};

        if value then do;
```

```
          if value > 5 then text=catt(value,'%');
          else text='*';

          attrs=catt('Width=',segments{i},'pct color=', segmentFontColors{i},
                     ' backgroundcolor=', segmentColors{i}, ' just=c vjust=m',
                     ' fontFamily="ARIAL" fontWeight=bold fontSize=9pt',
                     ' height=.4in');

            ods.format_cell(data: text, overrides: attrs);
        end;
      end;

      ods.row_end();
    ods.table_end();
  end;

  * Otherwise display error text;
  else ods.format_text(text: 'ERROR: Positive, neutral, ' ||
                            'and negative do not add to 100%.');

  if _last then do;

    * End the gridded layout;
    ods.layout_end();

      * Display the note;
    ods.region(x: '0in', y: '9.2in');
    text=put('F2',$footnote.);
    ods.format_text(text: text, overrides: 'fontFamily="ARIAL" fontSize=9pt');

    * End the absolute layout;
    ods.layout_end();
  end;
run;

ods listing;
ods pdf close;
```



Survey Feedback Report for NESUG 2010 Nursing Home

250 Respondents

10. Overall Perceptions of Resident Safety

🟩 Positive ⬜ Neutral 🟥 Negative

D1. Residents are well cared for in this nursing home.

D6. This nursing home does a good job keeping residents safe.

D8. This nursing home is a safe place for residents.

11. Management Support for Resident Safety

D2. Management asks staff how the nursing home can improve resident safety.

D7. Management listens to staff ideas and suggestions to improve resident safety.

D9. Management often walks around the nursing home to check on resident care.

12. Organizational Learning

D3. This nursing home lets the same mistakes happen again and again. (negatively worded)

D4. It is easy to make changes to improve resident safety in this nursing home.

D5. This nursing home is always doing things to improve resident safety.

D10. When this nursing home makes changes to improve resident safety, it checks to see if the changes worked.

* % not displayed for 5% or less

Report Generated August 2010

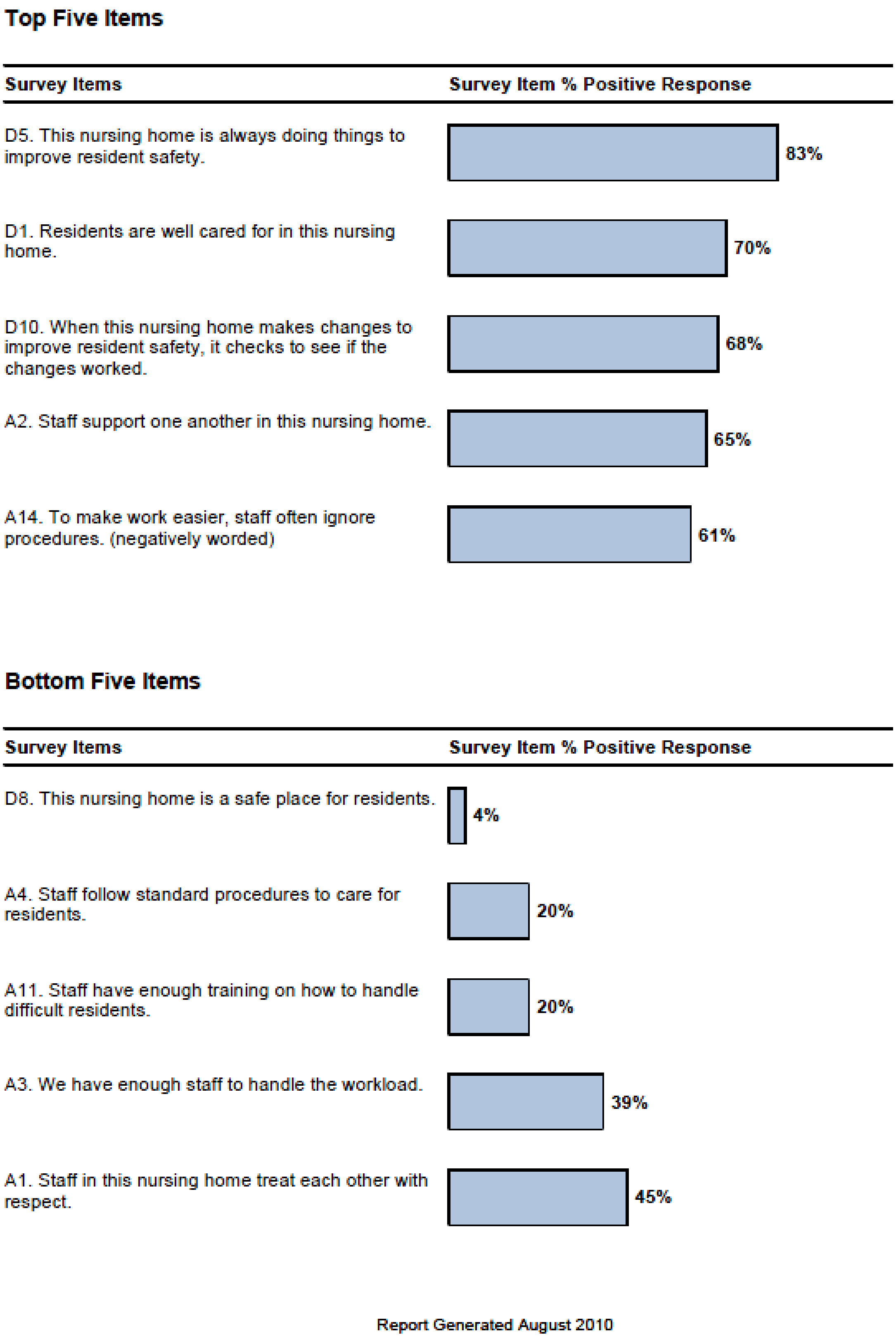
Horizontal Bar Chart: Creating the Varying Length Bars

Using the ODS Report Writing Interface to create the simple horizontal bar chart is very similar to creating the stacked horizontal bar chart. We use the same 2-column layout. The first column will display the survey question text as bar labels and the second column will be used to display the bars. The difference involves using the FORMAT_CELL method's optional INHIBIT argument to suppress the cell borders for the second cell which is used to display the bar value.

The next example creates 2 separate bar charts on a single page. We use a separate gridded layout for each bar chart.

Although not required to create the horizontal bars, the example has lines before and after the column headings. Tle LINE method can be used to create the lines, and the FORMAT_TEXT method can be used to create the column headings. But, using a table with inhibited left and right borders to create the appearance of lines seems to produce better looking results and provides an easy way to align the column headings.

```
/*
   Create horizontal bar charts to represent top 5
   and bottom 5 % positive survey item responses.
*/
%let month=%sysfunc(today(),monname15.);

%let year=%sysfunc(today(),year.);

filename pdfout "c:\randy\nesug\2010\bc2.pdf";

options nonumber nodate nocenter topmargin=1in leftmargin=1in orientation=portrait;

ods noptitle escapechar='^';
ods listing close;
ods pdf file=pdfout notoc;

title;
footnote "^{style [fontFamily='ARIAL' fontStyle=roman "
         "fontWeight=light fontSize=9pt "
         "just=center]%sysfunc(trim(%sysfunc(putc(F1,$footnote.)))) "
         "&month &year}";

data _null_;
length attrs $ 256;
set top5_n_bottom5 end=_last;

attrs=catx(' ','fontFamily="ARIAL"','fontSize=12pt',
               'fontWeight=bold');

if _n_=1 then do;
  dcl odsout ods();

  * Title line;
  ods.format_text(data: 'T3', format: '$title.',
                  overrides: attrs);
  * Blank line;
  ods.format_text(data: ' ', overrides: 'height=.25in');
end;

if _n_=6 then do;

  * Blank line;
  ods.format_text(data: ' ', overrides: 'height=.45in');

  * Title line;
  ods.format_text(data: 'T4', format: '$title.',
                  overrides: attrs);

  * Blank line;
  ods.format_text(data: ' ', overrides: 'height=.25in');
end;

* Create the column headers;
if _n_ in(1,6) then do;

  * Table with inhibited left and right borders used
    to create appearance of lines before and after
    column headings;

  ods.table_start(overrides: 'width=95pct cellpadding=0');
  ods.row_start();

  attrs=catx(' ','height=.25in','vjust=c','fontFamily="ARIAL"',
                 'fontSize=10pt','fontWeight=bold');

  ods.format_cell(data: 'CL1', format: '$columnLabel.',
                  inhibit: 'LR', overrides: 'width=47.5pct ' || attrs);

  ods.format_cell(data: 'CL2', format: '$columnLabel.',
                  inhibit: 'LR', overrides: 'width=47.5pct ' || attrs);
  ods.row_end();
  ods.table_end();

  ods.layout_gridded(columns: 2, overrides: 'cellspacing=0');
end;

* Question text;
ods.region(width: '45pct', height: '.7in');
ods.format_text(data: item, format: '$question',
                overrides: 'just=left fontFamily="ARIAL" fontSize=10pt');

ods.region(width: '50pct', height: '.7in');
if positive > 0 then do;

  * Create the HBAR;
  ods.table_start(overrides: 'borderwidth=.25pct cellspacing=0');
    ods.row_start();
      ods.format_cell(data: ' ',overrides: 'width=' ||
                                           trim(put(positive * .80,best.)) ||
                                           ' pct height=.4in ' ||
                                           'backgroundcolor=lightsteelblue');
      * Create 2nd cell with no borders;
      ods.format_cell(inhibit: 'TRB', data: trim(left(put(positive,best.))) || '%',
                      overrides: 'just=left vjust=c fontFamily="ARIAL" ' ||
                                 'fontSize=10pt fontWeight=bold');
    ods.row_end();
  ods.table_end();

end;

* Don't create the bar.  Just display 0%;
else ods.format_text(data: '0%',overrides: 'cellpadding=0 vjust=c just=left ' ||
                                           'fontFamily="ARIAL" fontSize=10pt ' ||
                                           'fontWeight=bold');

if _last or _n_=5 then ods.layout_end();
run;
ods pdf close;
ods listing;
```



Top Five Items

Survey Items	Survey Item % Positive Response
D5. This nursing home is always doing things to improve resident safety.	83%
D1. Residents are well cared for in this nursing home.	70%
D10. When this nursing home makes changes to improve resident safety, it checks to see if the changes worked.	68%
A2. Staff support one another in this nursing home.	65%
A14. To make work easier, staff often ignore procedures. (negatively worded)	61%

Bottom Five Items

Survey Items	Survey Item % Positive Response
D8. This nursing home is a safe place for residents.	4%
A4. Staff follow standard procedures to care for residents.	20%
A11. Staff have enough training on how to handle difficult residents.	20%
A3. We have enough staff to handle the workload.	39%
A1. Staff in this nursing home treat each other with respect.	45%

Report Generated August 2010

CONCLUSION

The ODS Report Writing Interface can be used effectively to create some types of horizontal bar charts, and is especially useful when you need to display large amounts of text as bar labels.

TRADEMARK CITATION

ACKNOWLEDGMENTS

The author would like to thank Mike Rhoads for his assistance, encouragement, patience, and support.

DISCLAIMER

The content of this paper is the work of the author and does not necessarily represent the opinions, recommendations, or practices of Westat.

REFERENCES

Daniel O’Connor, 2009, SAS Institute Inc., Cary NC - The Power to Show: Ad Hoc Reporting, Custom Invoices, and Form Letters

Daniel O’Connor, 2003, SAS Institute Inc., Cary NC - Next Generation Data _NULL_ Report Writing Using ODS OO Features

SAS 9.2 Language Reference, SAS Institute Inc., Cary NC

CONTACT INFORMATION

Randy Herbison
Westat
RandyHerbison@westat.com

APPENDIX

Bar Chart Data

```
proc format;
value $title
  'T1'='Survey Feedback Report for'
  'T2'='Respondents'
  'T3'='Top Five Items'
  'T4'='Bottom Five Items'
;
value $footnote
  'F1'='Report Generated'
  'F2'='* % not displayed for 5% or less'
;
value $legend
  'POSITIVE'='Positive'
  'NEUTRAL'='Neutral'
  'NEGATIVE'='Negative'
;
value $composite
  '1'='1. Teamwork'
  '2'='2. Staffing'
  '3'='3. Compliance With Procedures'
  '4'='4. Training & Skills'
  '10'='10. Overall Perceptions of Resident Safety'
  '11'='11. Management Support for Resident Safety'
  '12'='12. Organizational Learning'
;
value $question
  'A1'='A1. Staff in this nursing home treat each other with respect.'
  'A2'='A2. Staff support one another in this nursing home.'
  'A5'='A5. Staff feel like they are part of a team.'
  'A9'='A9. When someone gets really busy in this nursing home, other staff help
out.'
  'A3'='A3. We have enough staff to handle the workload.'
  'A8'='A8. Staff have to hurry because they have too much work to do. (negatively
worded)'
  'A16'='A16. Residents’ needs are met during shift changes'
  'A17'='A17. It is hard to keep residents safe here because so many staff quit
their jobs. (negatively worded)'
  'A4'='A4. Staff follow standard procedures to care for residents.'
  'A6'='A6. Staff use shortcuts to get their work done faster. (negatively worded)'
  'A14'='A14. To make work easier, staff often ignore procedures. (negatively
worded)'
  'A7'='A7. Staff get the training they need in this nursing home.'
  'A11'='A11. Staff have enough training on how to handle difficult residents.'
  'D2'='D2. Management asks staff how the nursing home can improve resident
safety.'
  'D7'='D7. Management listens to staff ideas and suggestions to improve resident
safety.'
  'D9'='D9. Management often walks around the nursing home to check on resident
care.'
  'A13'='A13. Staff understand the training they get in this nursing home.'
  'D3'='D3. This nursing home lets the same mistakes happen again and again.
(negatively worded)'
  'D4'='D4. It is easy to make changes to improve resident safety in this nursing
home.'
  'D5'='D5. This nursing home is always doing things to improve resident safety.'
  'D10'='D10. When this nursing home makes changes to improve resident safety, it
checks to see if the changes worked.'
  'D1'='D1. Residents are well cared for in this nursing home.'
  'D6'='D6. This nursing home does a good job keeping residents safe.'
  'D8'='D8. This nursing home is a safe place for residents.'
;
value
$columnLabel
  'CL1'='Survey Items'
  'CL2'='Survey Item % Positive Response'
  'CL3'='Column Label 3'
  'CL4'='Column Label 4'
;
run;

data bc;
  input composite $ item $ positive neutral negative;
cards;
1 A1 45 30 25
1 A2 65 20 15
1 A5 50 46 4
1 A9 60 40 20
2 A3 39 25 36
2 A8 45 30 25
2 A16 54 31 15
2 A17 50 25 25
3 A4 20 40 40
3 A6 45 30 25
3 A14 61 24 15
4 A7 50 25 25
4 A11 20 40 40
4 A13 45 30 25
10 D1 70 20 10
10 D6 45 30 25
10 D8 4   41 55
11 D2 59 22 19
11 D7 53 30 17
11 D9 48 32 20
12 D3 52 24 24
12 D4 47 31 22
12 D5 83 10 7
12 D10 68 20 11
;

proc sort data=bc out=top5;
  by descending positive;
run;

proc sort data=bc out=bottom5;
  by positive;
run;

data top5_n_bottom5;
  set top5(obs=5) bottom5(obs=5);
run;
```