

SAS® and Excel, A Winning Combination, Part 1: Easy Solutions for Multi-Sheet Excel Workbooks, Color-Coding, Imbedded Graphs, etc.

LeRoy Bessler PhD, Bessler Consulting and Research
Fox Point, Milwaukee, Wisconsin, USA, Le_Roy_Bessler@wi.rr.com

Abstract

SAS is my personal favorite computer tool, but the world's commonest non-SAS tool for data presentation, and working with data, is Microsoft Excel. People often want your SAS report formatted as a spreadsheet, so that they and others can post-process or reformat data however they like. This short tutorial on SAS-to-Excel solutions is for anyone who needs to deliver information via Excel from a SAS program. In regard to the popular "traffic lighting" concept, this paper will explain how to color-code data so as to address the commonest form of color blindness. The paper will reach the limits of what this author can easily do with the tools selected. For another solution for SAS-with-Excel, the reader is referred to the author's companion paper on Dynamic Data Exchange (DDE). Please see Reference 1.

Introduction

Below is a list of the examples. Each output is displayed in its own section, with supporting code below it. The only explanation provided, if any, is in the comment(s) for the code.

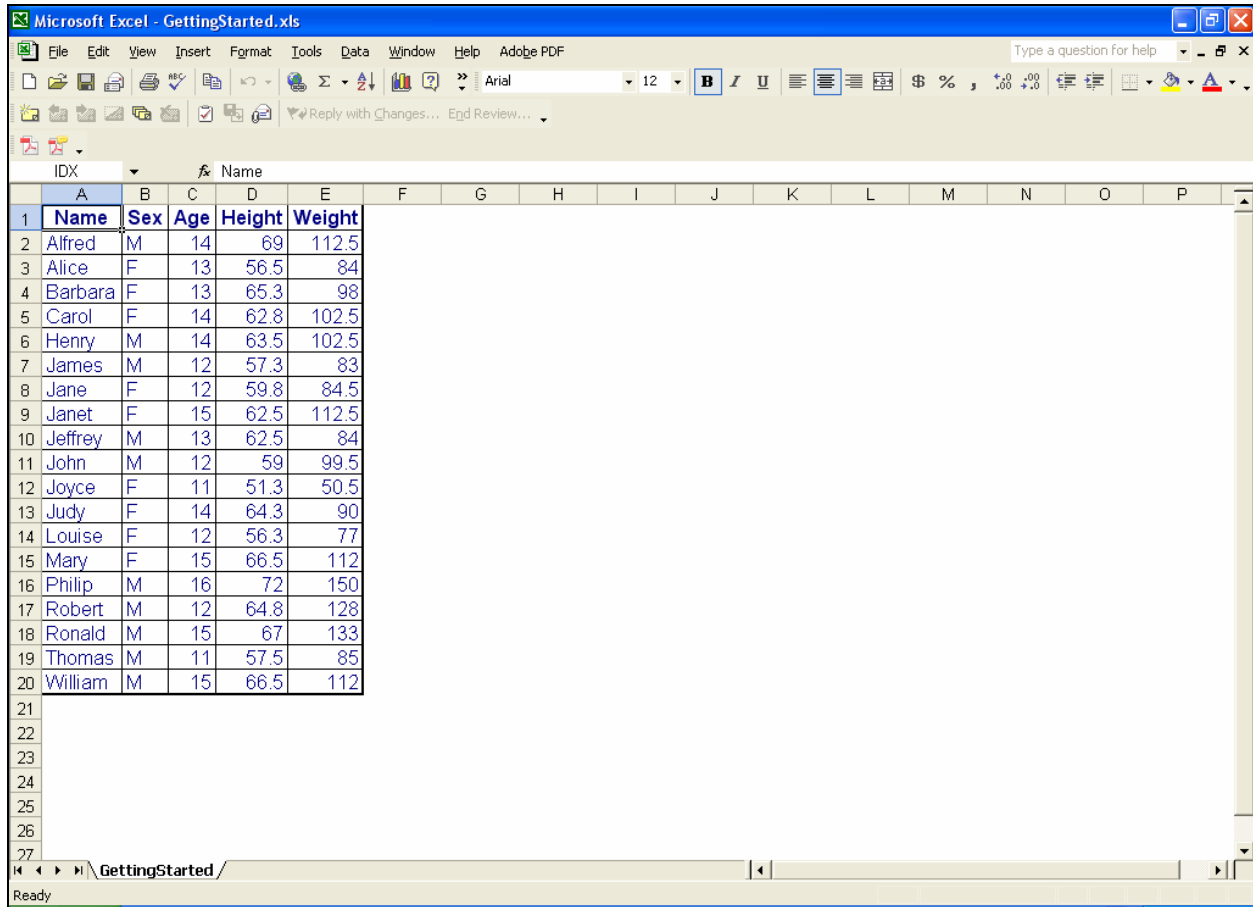
Solutions Using ODS HTML

1. Non-Customized Excel Output
2. Using ODS Style Minimal
3. Default Handling of Title (Unacceptable)
4. Using HTML Column Spanning Tag (Getting Better)
5. Title Spanning Seven Columns (The Correct Solution)
6. Conventional TITLE Statement Controls for Appearance of Text Fail with Column Spanning
7. Irrelevant Choice: Customize Text Appearance in Stretched Column
8. Customized Column-Spanning Title (the solution for a customized title in a spreadsheet from SAS)
9. Table with Footnote
10. Color-Coded Data (sometimes called "traffic lighting")
11. PROC PRINT Font and Color Control:
Everything that you can do except for STYLE(OBS) and STYLE(OBSHEADER)
12. Table Above Graph

Solutions Not Able To Use ODS HTML

13. Table at Left of Graph and Excel Sheet Name Customization
(requires MSOffice2K_x tagset, which can not support multi-sheet workbooks)
14. Multi-Sheet Workbook
(requires ExcelXP tagset, which can not support graphs)

Non-Customized Excel Output



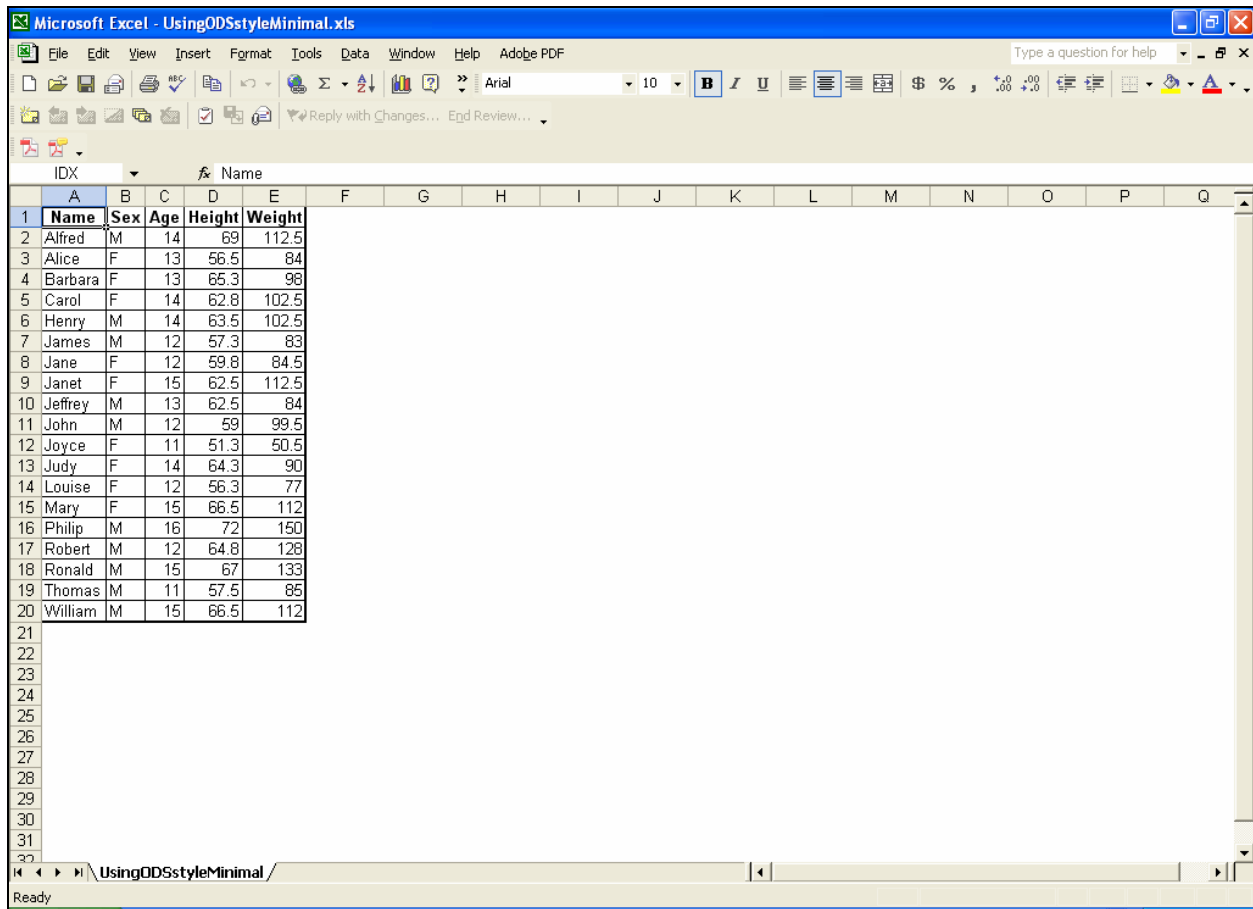
The screenshot shows a Microsoft Excel window titled "GettingStarted.xls". The spreadsheet contains a table with 20 rows and 5 columns. The columns are labeled "Name", "Sex", "Age", "Height", and "Weight". The data is as follows:

	A	B	C	D	E
1	Name	Sex	Age	Height	Weight
2	Alfred	M	14	69	112.5
3	Alice	F	13	56.5	84
4	Barbara	F	13	65.3	98
5	Carol	F	14	62.8	102.5
6	Henry	M	14	63.5	102.5
7	James	M	12	57.3	83
8	Jane	F	12	59.8	84.5
9	Janet	F	15	62.5	112.5
10	Jeffrey	M	13	62.5	84
11	John	M	12	59	99.5
12	Joyce	F	11	51.3	50.5
13	Judy	F	14	64.3	90
14	Louise	F	12	56.3	77
15	Mary	F	15	66.5	112
16	Philip	M	16	72	150
17	Robert	M	12	64.8	128
18	Ronald	M	15	67	133
19	Thomas	M	11	57.5	85
20	William	M	15	66.5	112

```
ods listing close;
ods noresults; * Do not open output in SAS *;

goptions reset=all; * Always do this. *;
ods html file="C:\SAS\Xloutput\GettingStarted.xls";
* start your reporting code here *;
proc print data=sashelp.class noobs label;
run;
* end your reporting code here *;
ods html close;
```

Using ODS Style Minimal



Microsoft Excel - UsingODSstyleMinimal.xls

Type a question for help

File Edit View Insert Format Tools Data Window Help Adobe PDF

Reply with Changes... Egd Review...

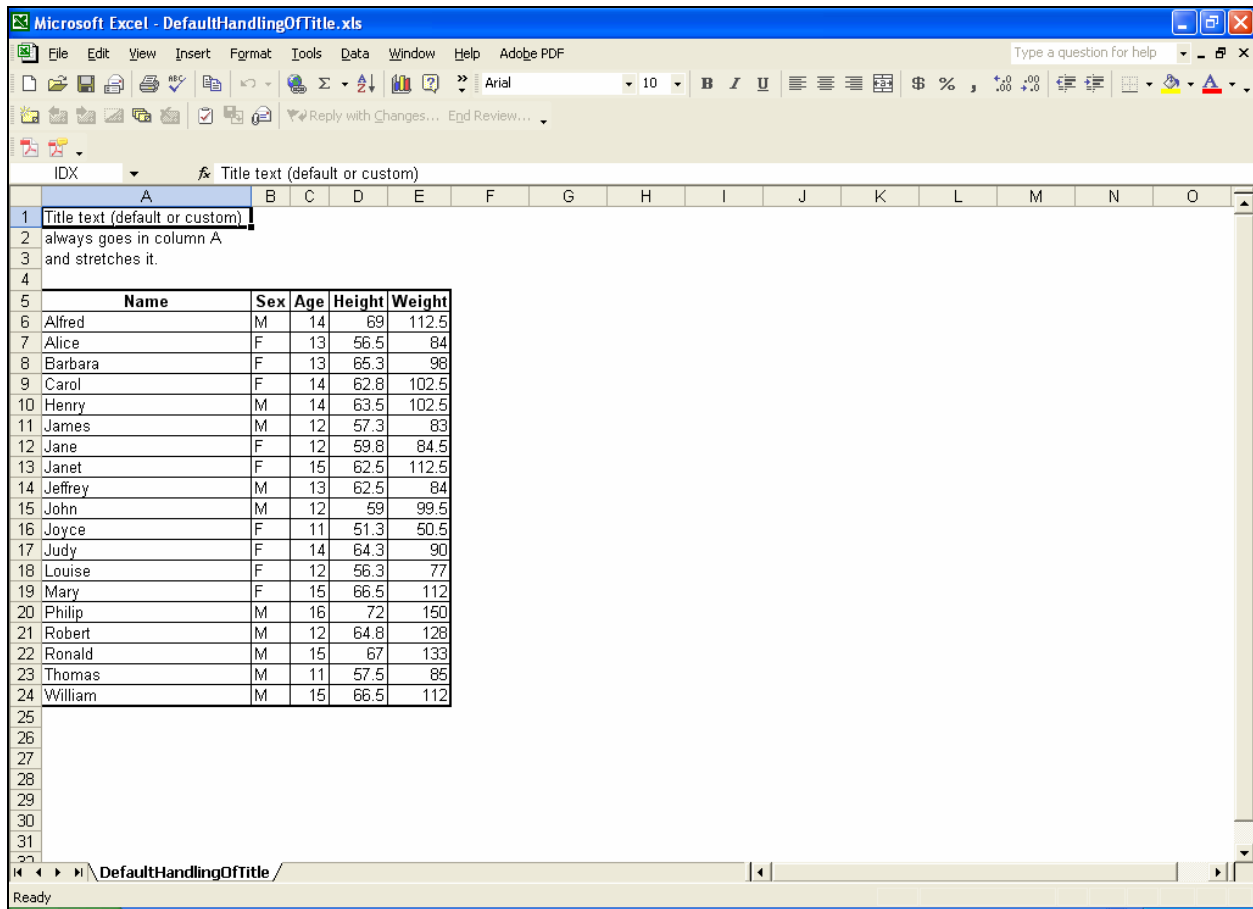
1	Name	Sex	Age	Height	Weight
2	Alfred	M	14	69	112.5
3	Alice	F	13	56.5	84
4	Barbara	F	13	65.3	98
5	Carol	F	14	62.8	102.5
6	Henry	M	14	63.5	102.5
7	James	M	12	57.3	83
8	Jane	F	12	59.8	84.5
9	Janet	F	15	62.5	112.5
10	Jeffrey	M	13	62.5	84
11	John	M	12	59	99.5
12	Joyce	F	11	51.3	50.5
13	Judy	F	14	64.3	90
14	Louise	F	12	56.3	77
15	Mary	F	15	66.5	112
16	Philip	M	16	72	150
17	Robert	M	12	64.8	128
18	Ronald	M	15	67	133
19	Thomas	M	11	57.5	85
20	William	M	15	66.5	112
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					

UsingODSstyleMinimal/

Ready

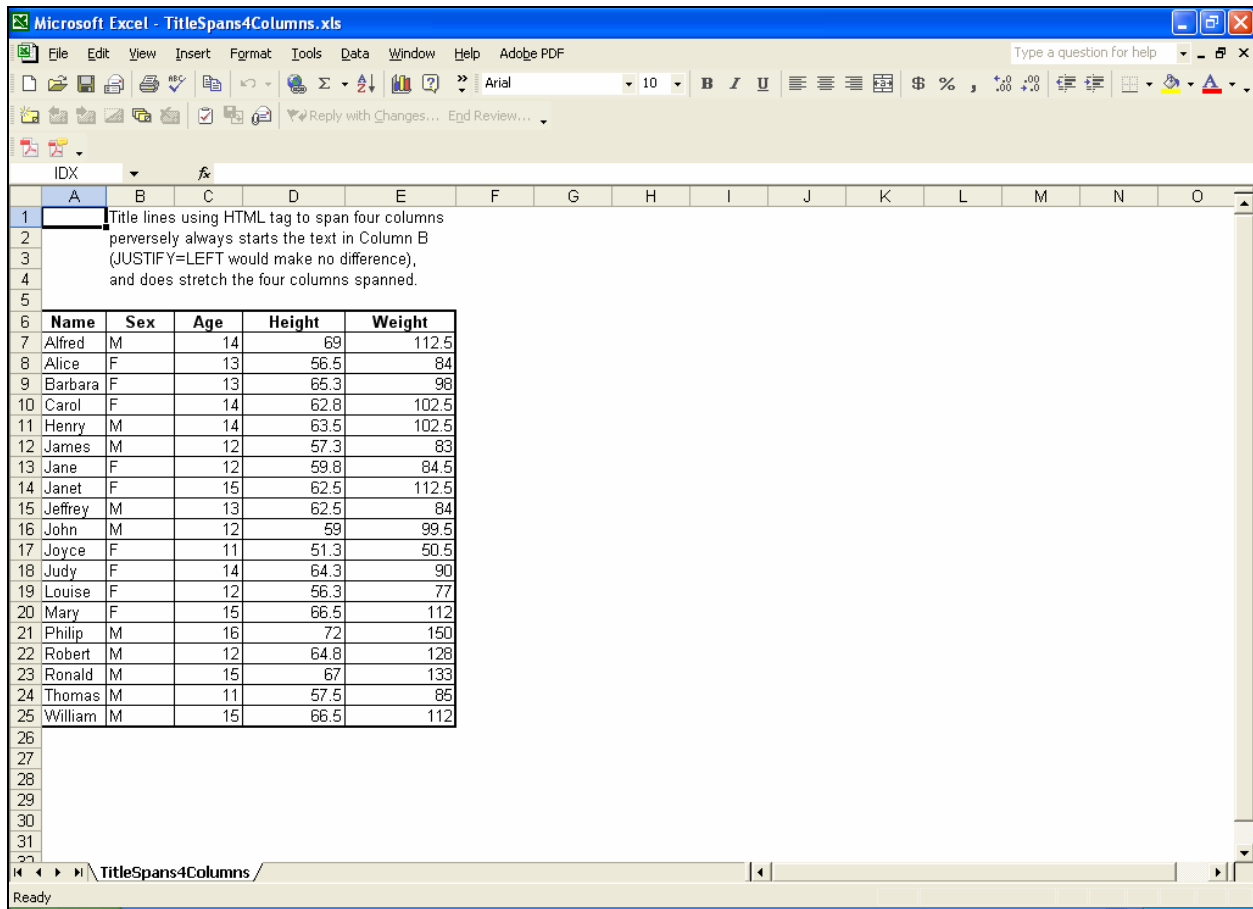
```
options reset=all; * Always do this. *;
ods html file="C:\SAS\toX\output\UsingODSstyleMinimal.xls"
style=Minimal;
proc print data=sashelp.class noobs label;
run;
ods html close;
```

Default Handling of Title (Unacceptable)



```
options reset=all; * Always do this. *;
ods html file="C:\SAS\toX\output\DefaultHandlingOfTitle.xls"
style=Minimal;
title1 'Title text (default or custom)';
title2 'always goes in column A';
title3 'and stretches it.';
proc print data=sashelp.class noobs;
run;
ods html close;
```

Using HTML Column Spanning Tag (Getting Better)

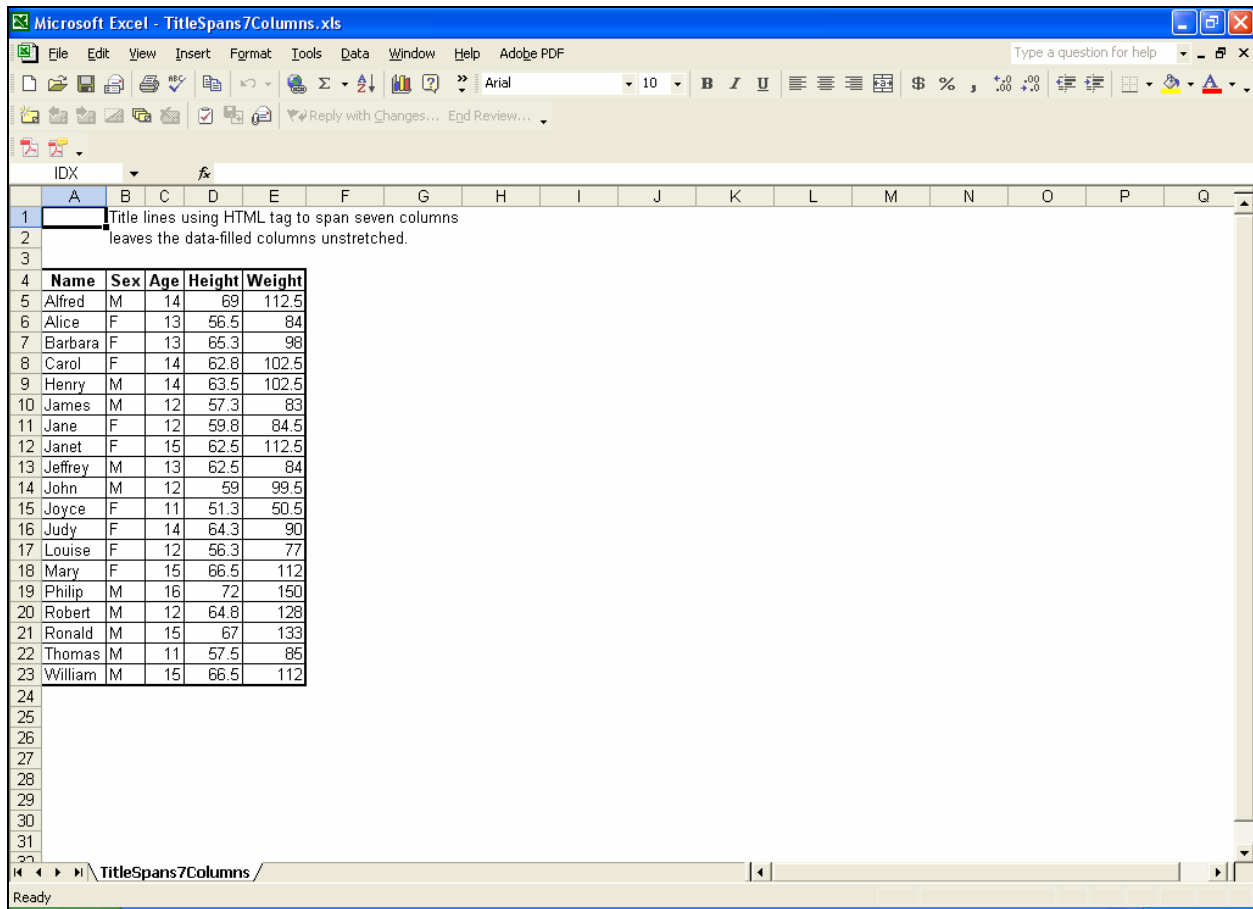


The screenshot shows a Microsoft Excel window titled "TitleSpans4Columns.xls". The spreadsheet has columns A through O and rows 1 through 31. Row 1 contains a cell with a colspan=4 attribute, which spans columns A, B, C, and D. The text in this cell is: "Title lines using HTML tag to span four columns". Row 2 contains a cell with a colspan=4 attribute, which spans columns A, B, C, and D. The text in this cell is: "perversely always starts the text in Column B". Row 3 contains a cell with a colspan=4 attribute, which spans columns A, B, C, and D. The text in this cell is: "(JUSTIFY=LEFT would make no difference)". Row 4 contains a cell with a colspan=4 attribute, which spans columns A, B, C, and D. The text in this cell is: "and does stretch the four columns spanned." Below these rows is a table with 5 columns: Name, Sex, Age, Height, and Weight. The data in this table is as follows:

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83
Jane	F	12	59.8	84.5
Janet	F	15	62.5	112.5
Jeffrey	M	13	62.5	84
John	M	12	59	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90
Louise	F	12	56.3	77
Mary	F	15	66.5	112
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112

```
options reset=all; * Always do this. *;
ods html file="C:\SAS\toX\output\TitleSpans4Columns.xls" style=Minimal;
title1
'<td COLSPAN=4>Title lines using HTML tag to span four columns</td>';
title2
'<td COLSPAN=4>perversely always starts the text in Column B</td>';
title3
'<td COLSPAN=4>(JUSTIFY=LEFT would make no difference),</td>';
title4
'<td COLSPAN=4>and does stretch the four columns spanned.</td>';
proc print data=sashelp.class noobs;
run;
ods html close;
```

Title Spanning Seven Columns (The Correct Solution)



The screenshot shows a Microsoft Excel window titled "TitleSpans7Columns.xls". The spreadsheet has two rows with titles that span seven columns (A-G). The first row title is "Title lines using HTML tag to span seven columns" and the second row title is "leaves the data-filled columns unstretched." Below these titles is a table with five columns: Name, Sex, Age, Height, and Weight. The table contains 15 rows of data.

Title lines using HTML tag to span seven columns						
leaves the data-filled columns unstretched.						
4	Name	Sex	Age	Height	Weight	
5	Alfred	M	14	69	112.5	
6	Alice	F	13	56.5	84	
7	Barbara	F	13	65.3	98	
8	Carol	F	14	62.8	102.5	
9	Henry	M	14	63.5	102.5	
10	James	M	12	57.3	83	
11	Jane	F	12	59.8	84.5	
12	Janet	F	15	62.5	112.5	
13	Jeffrey	M	13	62.5	84	
14	John	M	12	59	99.5	
15	Joyce	F	11	51.3	50.5	
16	Judy	F	14	64.3	90	
17	Louise	F	12	56.3	77	
18	Mary	F	15	66.5	112	
19	Philip	M	16	72	150	
20	Robert	M	12	64.8	128	
21	Ronald	M	15	67	133	
22	Thomas	M	11	57.5	85	
23	William	M	15	66.5	112	

```
options reset=all; * Always do this. *;
ods html file="C:\SAS\toX\output\TitleSpans7Columns.xls" style=Minimal;
title1
'<td COLSPAN=7>Title lines using HTML tag to span seven columns</td>';
title2
'<td COLSPAN=7>leaves the data-filled columns unstretched.</td>';
proc print data=sashelp.class noobs;
run;
ods html close;
```

Conventional TITLE Statement Controls for Appearance of Text Fail with Column Spanning

The screenshot shows an Excel spreadsheet with the following data:

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83
Jane	F	12	59.8	84.5
Janet	F	15	62.5	112.5
Jeffrey	M	13	62.5	84
John	M	12	59	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90
Louise	F	12	56.3	77
Mary	F	15	66.5	112
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112

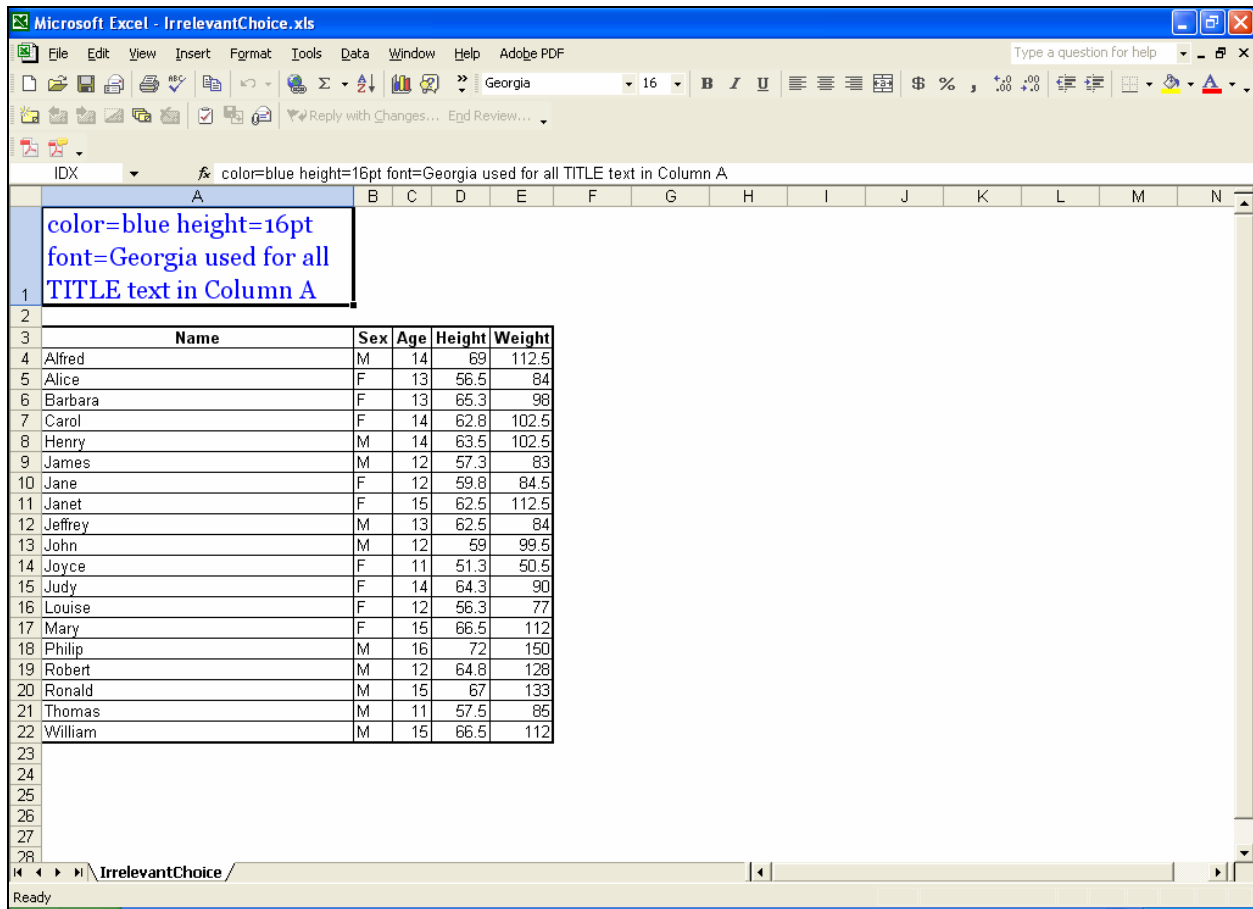
The first row of the spreadsheet (row 1) contains a single cell with the text: "color=blue height=16pt font=Georgia used for this TITLE statement". This cell spans across columns A through Q.

```

options reset=all; * Always do this. *;
ods html file="C:\SAS\toX\output\CustomizedTitleAppearance.xls"
style=Minimal;
title1 color=blue height=16pt font=Georgia
'<td COLSPAN=8>color=blue height=16pt font=Georgia used for this TITLE
statement</td>';
proc print data=sashelp.class noobs;
run;
ods html close;

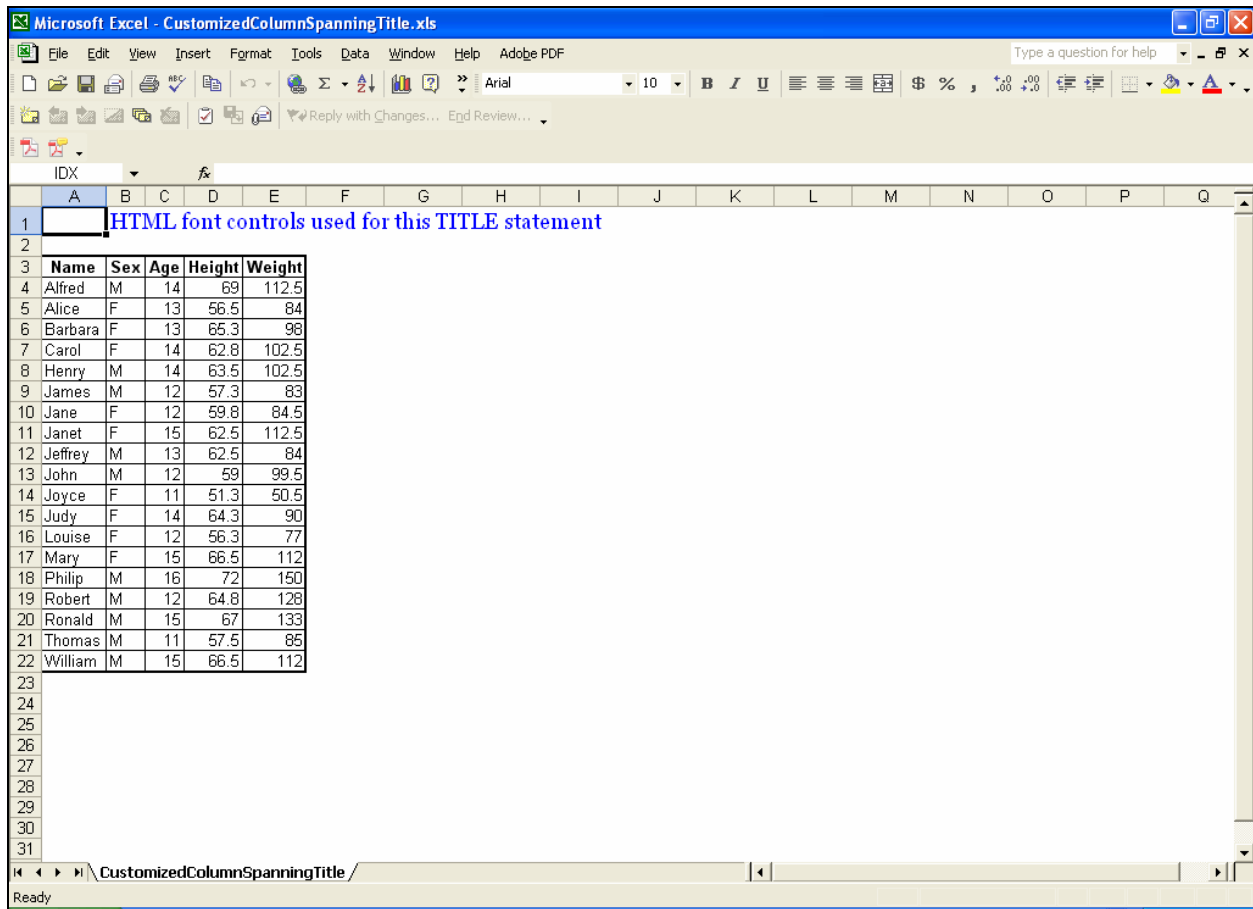
```

Irrelevant Choice: Customize Text Appearance in Stretched Column A



```
goptions reset=all; * Always do this. *;
ods html file="C:\SAS\toX\output\IrrelevantChoice.xls" style=Minimal;
title1 color=blue height=16pt font=Georgia
'color=blue height=16pt font=Georgia used for all TITLE text in Column
A';
proc print data=sashelp.class noobs;
run;
ods html close;
```


Customized Column-Spanning Title (the solution for a customized title in a spreadsheet from SAS)



The screenshot shows a Microsoft Excel window titled "CustomizedColumnSpanningTitle.xls". The spreadsheet has a title in cell A1 that spans 8 columns: "HTML font controls used for this TITLE statement". Below the title is a table with 5 columns: Name, Sex, Age, Height, and Weight. The table contains 22 rows of data.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	HTML font controls used for this TITLE statement																
2																	
3	Name	Sex	Age	Height	Weight												
4	Alfred	M	14	69	112.5												
5	Alice	F	13	56.5	84												
6	Barbara	F	13	65.3	98												
7	Carol	F	14	62.8	102.5												
8	Henry	M	14	63.5	102.5												
9	James	M	12	57.3	83												
10	Jane	F	12	59.8	84.5												
11	Janet	F	15	62.5	112.5												
12	Jeffrey	M	13	62.5	84												
13	John	M	12	59	99.5												
14	Joyce	F	11	51.3	50.5												
15	Judy	F	14	64.3	90												
16	Louise	F	12	56.3	77												
17	Mary	F	15	66.5	112												
18	Philip	M	16	72	150												
19	Robert	M	12	64.8	128												
20	Ronald	M	15	67	133												
21	Thomas	M	11	57.5	85												
22	William	M	15	66.5	112												
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	

```
options reset=all; * Always do this. *;
ods html file="C:\SAS\Xloutput\CustomizedColumnSpanningTitle.xls"
style=Minimal;
title1
'<td COLSPAN=8><font color=blue size=4 face=Georgia>HTML font controls
used for this TITLE statement</font></td>';
* HTML font sizes are 1,2,3,4,5,6,7, which are not point sizes *;
proc print data=sashelp.class noobs;
run;
ods html close;
```

Table with Footnote

The screenshot shows a Microsoft Excel spreadsheet titled "TableWithFootnote.xls". The spreadsheet contains a table with 22 rows of data and a footnote. The table has columns for Name, Sex, Age, Height, and Weight. The footnote is located in cell A24 and spans 8 columns.

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83
Jane	F	12	59.8	84.5
Janet	F	15	62.5	112.5
Jeffrey	M	13	62.5	84
John	M	12	59	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90
Louise	F	12	56.3	77
Mary	F	15	66.5	112
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112

I am the FOOTNOTE1 statement text.

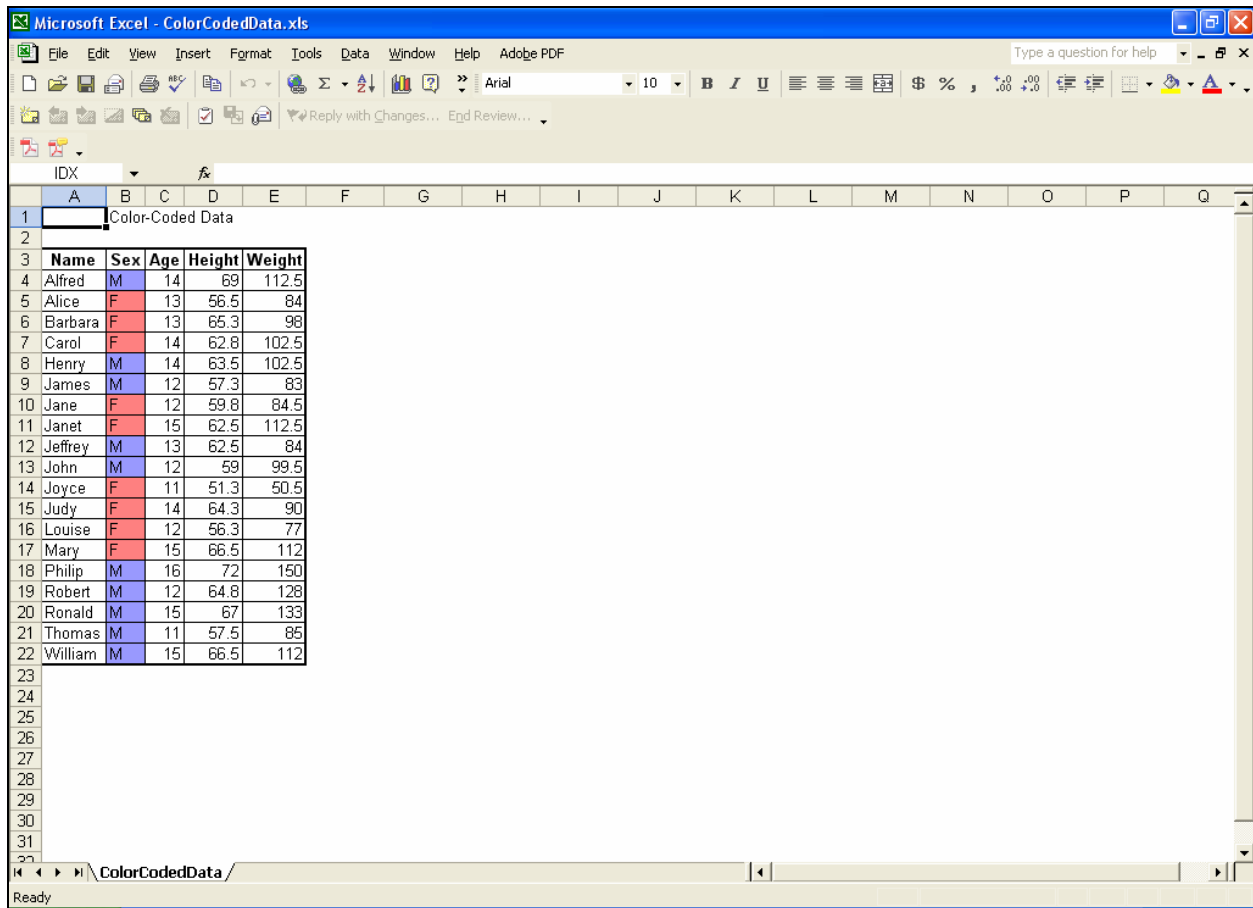
```

options reset=all; * Always do this. *;
ods html file="C:\SAS\toX\loutput\TableWithFootnote.xls" style=Minimal;
title1
'<td COLSPAN=8>Where do footnotes go in the spreadsheet?</td>';
footnote1
'<td COLSPAN=8>I am the FOOTNOTE1 statement text.</td>';
proc print data=sashelp.class noobs;
run;
ods html close;

```

Color-Coded Data (sometimes called “traffic lighting”)

The term “traffic lighting” in the context of data presentation refers to the use of red and green backgrounds to highlight bad and good, respectively. The problem is that the inability to distinguish red and green is the commonest form of color blindness (a condition wherein the two colors look the same). The simplest solution is to use red and blue. (Orange and green might be OK.) If using blue as a background for black text, it is important to use a light enough shade of blue. For even lighter shades of blue (and red), use AAAA or CCCC instead of 9999 in the color names below. However, on an LED monitor, the very light shades with CCCC might wash out to nearly white.



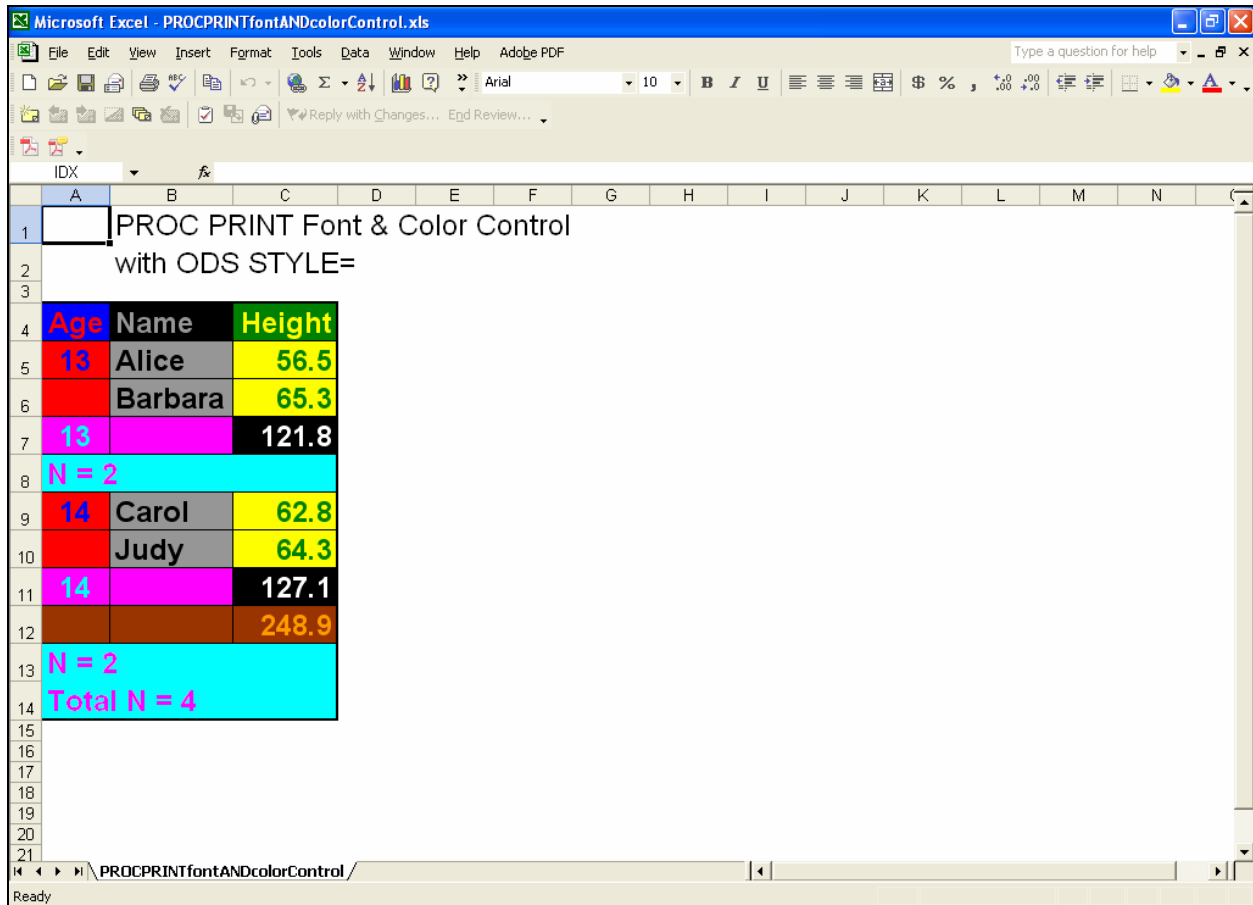
The screenshot shows a Microsoft Excel spreadsheet titled "ColorCodedData.xls". The spreadsheet contains a table with the following data:

IDX	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	67.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

```
proc format lib=work;
value $SexCol
'F'   = 'CXFF9999'
'M'   = 'CX9999FF';
run;
goptions reset=all; * Always do this. *;
ods html file="C:\SAS\Xloutput\ColorCodedData.xls" style=Minimal;
title1 '<td COLSPAN=4>Color-Coded Data</td>';
proc print data=sashelp.class noobs;
var Name;
var Sex / style = [background = $SexCol.];
var Age Height Weight;
run;
ods html close;
```

PROC PRINT Font and Color Control:

Everything that you can do except for STYLE(OBS) and STYLE(OBSHEADER)



The screenshot shows an Excel spreadsheet with the following content:

Age	Name	Height
13	Alice	56.5
	Barbara	65.3
13		121.8
N = 2		
14	Carol	62.8
	Judy	64.3
14		127.1
		248.9
N = 2		
Total N = 4		

```
proc sort data=sashelp.class out=ToPrint;
where name in ('Alice' 'Barbara' 'Carol' 'Judy');
by Age Name;
run;

%let FontFormatting = font_face=Arial font_weight=Bold font_size=5;
%let TitleFormatting = face=Arial weight=Bold size=5 color=black;

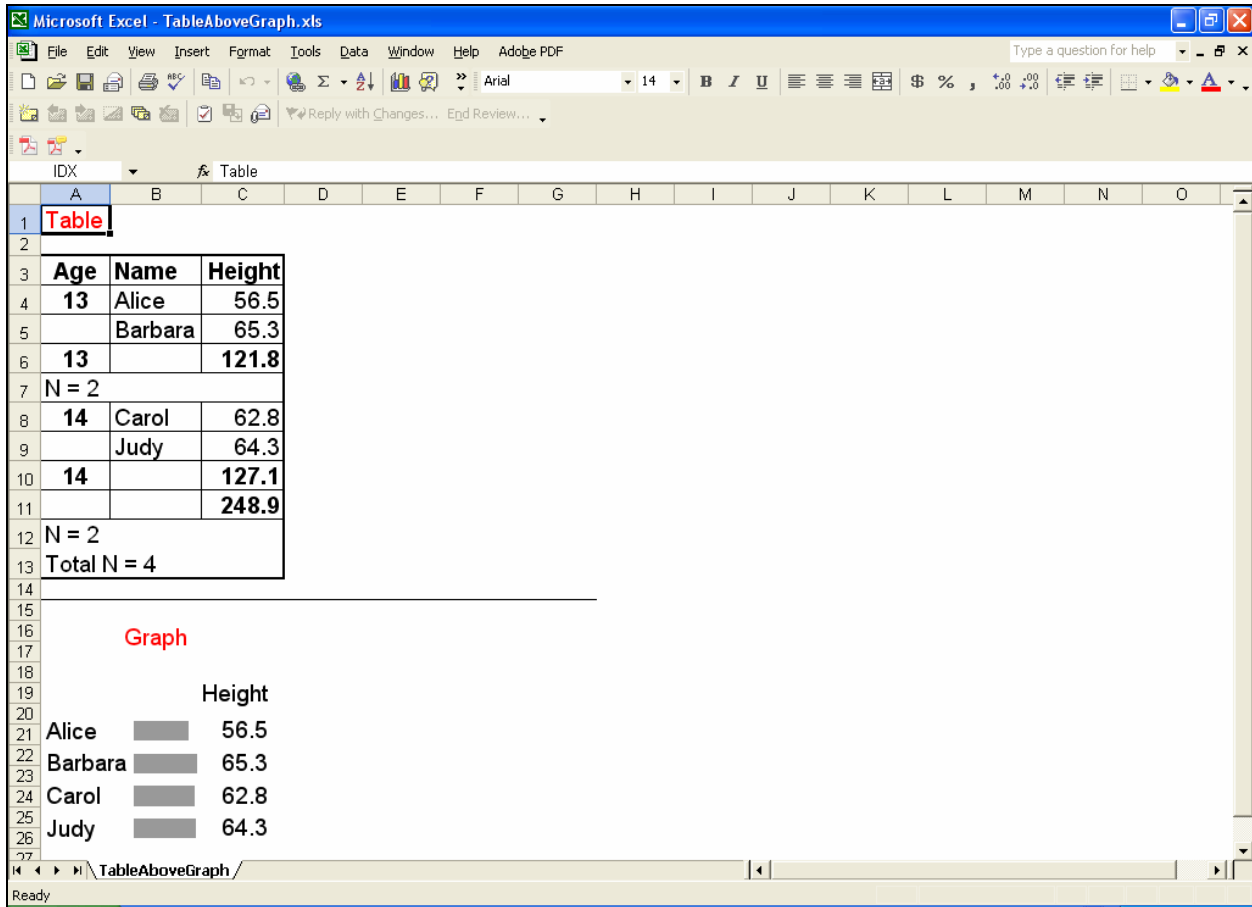
options reset=all; * Always do this. *;
ods html file="C:\SAS\toX\output\PROCPRINTfontANDcolorControl.xls"
style=Minimal;
title1 justify=left
  "<td COLSPAN=6><font &TitleFormatting>PROC PRINT Font & Color
Control</font></td>"
  justify=left
  "<td COLSPAN=6><font &TitleFormatting>with ODS STYLE=</font></td>";
proc print data=ToPrint label N
  style(header) = [&FontFormatting]
  style(data) = [&FontFormatting]
  style(total) = [&FontFormatting background=magenta foreground=cyan]
```

```

    style(bylabel)= [&FontFormatting background=magenta foreground=cyan
just=center]
    style(grand)   = [&FontFormatting background=brown foreground=orange]
    style(N)      = [&FontFormatting background=cyan foreground=magenta
just=left];
by age;
id age /
    style(header) = [&FontFormatting background=blue foreground=red]
    style(data)   = [&FontFormatting background=red foreground=blue];
var Name /
    style(header) = [background=black foreground=CX999999]
    style(data)   = [background=CX999999 foreground=black];
var Height;
sum Height /
    style(header) = [background=CX009900 foreground=yellow]
    style(data)   = [background=yellow foreground=CX009900]
    style(total) = [&FontFormatting background=black foreground=white];
sumby age;
run;
ods html close;

```

Table Above Graph



```

proc sort data=sashelp.class out=ToPrint;
where name in ('Alice' 'Barbara' 'Carol' 'Judy');
by Age Name;
run;

%let FontFormatting = font_face=Arial font_size=4;

goptions reset=all; * Always do this. *;
ods html path="C:\SAS\toX\output" file="TableAboveGraph.xls"
style=Minimal;
title1 font=Arial height=14pt color=red 'Table';
proc print data=ToPrint label N
  style(header) = [&FontFormatting]
  style(data) = [&FontFormatting]
  style(total) = [&FontFormatting]
  style(grand) = [&FontFormatting]
  style(bylabel)= [&FontFormatting just=center]
  style(N) = [&FontFormatting just=left];
by age;
id age /
  style(header) = [&FontFormatting]

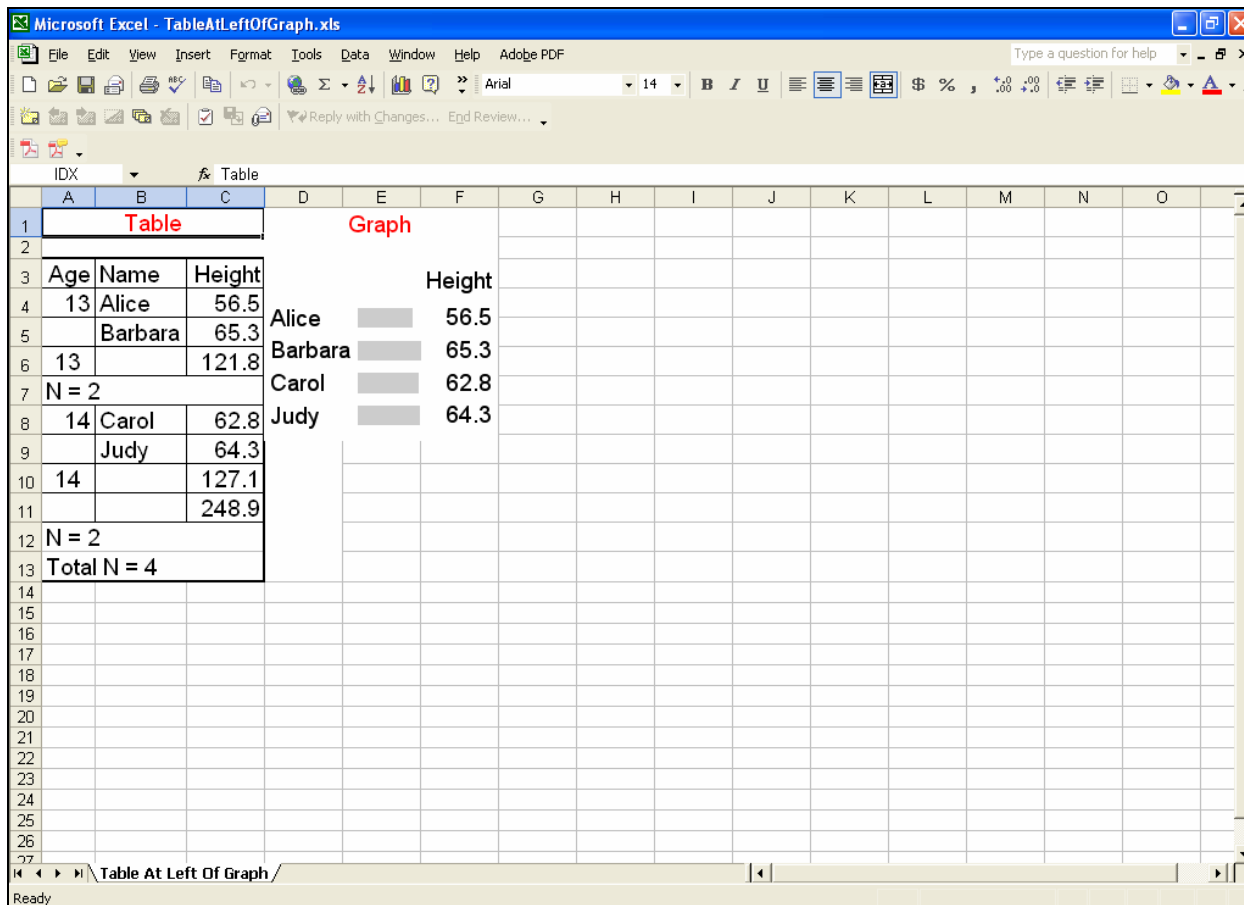
```

```

    style(data)    = [&FontFormatting];
var name height;
sum height;
run;
proc catalog c=work.gseg kill;run;quit;
goptions hsize=2in vsize=2in htext=10pct ftext='Arial' border;
                                * goption border
                                has no effect when imbedded in a spreadsheet *;
title color=red height=2pct ' '
justify=center height=10pct 'Graph';
footnote1 angle=-90 height=2pct ' ';
footnote2 angle=+90 height=1pct ' ';
proc gchart data=ToPrint;
pattern1 color=CX999999;
axis1 label=none style=0 major=none minor=none;
axis2 label=none style=0 major=none minor=none value=none;
hbar name / name='Below'
    sumvar=height sum sumlabel='Height'
    maxis=axis1 raxis=axis2
    width=4 space=3 coutline=same;
run; quit;
ods html close;

```

Table at Left of Graph and Excel Sheet Name Customization
 (requires MSOffice2K_x tagset, **which can not support multi-sheet workbooks**)



```
proc sort data=sashelp.class out=ToPrint;
where name in ('Alice' 'Barbara' 'Carol' 'Judy');
by Age Name;
run;

%let FontFormatting = font_face=Arial font_size=4;

goptions reset=all; * Always do this. *;
ods tagsets.MSOffice2K_x path="C:\SASToXLOutput"
file="TableAtLeftOfGraph.xls" style=Minimal
options(panelcols="2" sheet_name="Table At Left Of Graph" doc="help");
title1 font=Arial height=14pt color=red 'Table';
title1 font=Arial height=14pt color=red 'Table';
proc print data=ToPrint label N
  style(header) = [&FontFormatting]
  style(data) = [&FontFormatting]
  style(total) = [&FontFormatting]
  style(grand) = [&FontFormatting]
  style(bylabel)= [&FontFormatting just=center]
  style(N) = [&FontFormatting just=left];
```

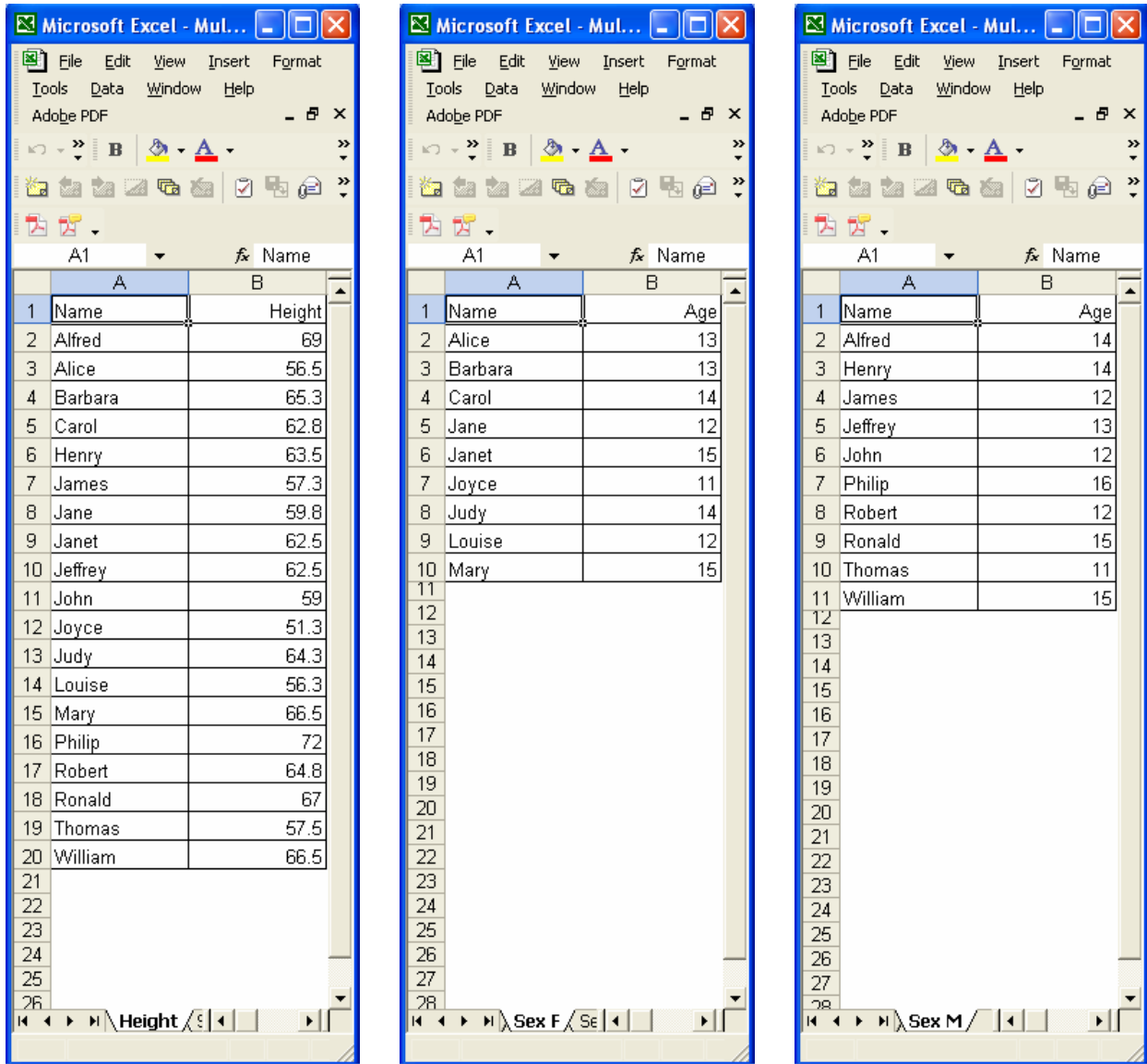


```

by age;
id age /
  style(header) = [&FontFormatting]
  style(data)   = [&FontFormatting];
var name height;
sum height;
run;
proc catalog c=work.gseg kill;run;quit;
goptions hsize=2in vsize=2in htext=10pct ftext='Arial' border;
                                * goption border
                                has no effect when imbedded in a spreadsheet *;
title color=red height=2pct ' '
justify=center height=10pct 'Graph';
footnote1 angle=-90 height=2pct ' ';
footnote2 angle=+90 height=1pct ' ';
proc gchart data=ToPrint;
pattern1 color=CXCCCCC;
axis1 label=none style=0 major=none minor=none;
axis2 label=none style=0 major=none minor=none value=none;
hbar name / name='AtRight'
  sumvar=height sum sumlabel='Height'
  maxis=axis1 raxis=axis2
  width=4 space=3 coutline=same;
run; quit;
ods _all_ close;

```

Multi-Sheet Workbook (requires ExcelXP tagset, **which can not support graphs)**



```
ods tagsets.excelxp style=Minimal
    file="C:\SASToXloutput\MultiSheetTablesOnlyWorkbook.xls";
ods tagsets.excelxp options(sheet_name="Height");
proc print data=sashelp.class noobs;
id name; var height; run;
ods tagsets.excelxp options(sheet_name="none");
ods tagsets.excelxp options(sheet_interval='bygroup'
    sheet_label='Sex'
    suppress_bylines='yes');
proc sort data=sashelp.class out=work.SortedToPrint;
by sex; run;
proc print data=work.SortedToPrint noobs;
by sex; id name; var age; run;
ods tagsets.excelxp close;
```

Conclusion

The methods shown address common needs when presenting data from SAS via Excel. For other Excel formatting requirements, the author has found Dynamic Data Exchange (DDE) to be a method that is extremely powerful, and, for him, easier to use than the alternatives. See Reference 1.

References (by this author)

1. LeRoy Bessler, "SAS and Excel, A Winning Combination, Part 2: Dynamic Data Exchange (DDE), a Popular Solution around the World", *Proceedings of the MidWest SAS Users Group Conference 2010*, MWSUG, Inc. (USA), 2010.

Acknowledgments

My thanks to Chevell Parker at SAS Technical Support for advice about ODS tagset MSOffice2K_x.

Author Information

Your questions, comments, suggestions, and other simple SAS-to-Excel solutions are welcome.

LeRoy Bessler PhD

Bessler Consulting and Research, Fox Point, Milwaukee, Wisconsin, USA

Le_Roy_Bessler@wi.rr.com

A SAS user since 1978, Dr. LeRoy Bessler has shared his knowledge and experience with other users at conferences throughout the USA and in Montreal, London, Heidelberg, and Dublin. Though a SAS generalist with long experience in Base SAS, SAS macro language, and SAS tools for access to non-SAS data, his special interests include communication-effective visual communication and reporting, web information delivery, highly formatted Excel reporting, SAS/GRAPH®, ODS, creation of unique tools to support the SAS BI server and its users, and Software-Intelligent Application Development for Reliability, Reusability, Extendibility, and Maintainability. He is a regular contributor to *VIEWS News*, the web newsletter of the VIEWS International SAS Programmer Community.

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries.

® indicates USA registration. Other product and brand names are trademarks or registered trademarks of their respective owners.