Paper 152-2010

Turning Raw Data into Polished Reports

Lora D. Delwiche, University of California, Davis

Susan J. Slaughter, Avocet Solutions, Davis, CA

Abstract

This paper shows how to turn raw data into polished looking reports. Starting at the beginning with reading your raw data file, we show how to sort and print your data, format data values, and add summary information to your reports. Using features of the Output Delivery System that are easy to learn, we also show how to create different types of output, and control style, font and color in your reports. With these techniques you can create reports that are attractive and draw attention to your main points to impress your boss or client.

Introduction

In the book "The Phantom Tollbooth," a boy named Milo sees life as a waste of time. Then one day he is magically transported to the Kingdom of Wisdom. His journey starts in Expectations, then, without thinking, he finds himself in the Doldrums. During his travels, Milo meets many interesting and unusual characters, and learns about the Princess of Sweet Rhyme and the Princess of Pure Reason. The princesses have been banished from the Kingdom, but Milo rescues them thereby restoring rhyme and reason to the Kingdom. And, in so doing, Milo finds meaning in his own life.

When confronted with a large amount of data, and told to write a report, your journey will have many similarities to Milo's journey through the Kingdom of Wisdom. You, and likely your boss, will start with some expectations about what the report will be like and what information it will convey. You may end up in the Doldrums, as Milo did, not knowing how to go about producing the report. Your journey will likely take many turns, and you will learn new things along the way. But, in the end, your goal is to bring rhyme and reason to your report thereby making sense out of your data.

Expectations

One of the interesting places Milo visits during his journey is the Word Market in the city of Dictionopolis. You can buy any word you like at the Word Market, and they come in all shapes, sizes and flavors. Suppose you have been given sales data for several vendors at the market, and are expected to produce a nice report. Table 1 shows the data file in CSV (comma separated value) format. Note that the first data line contains descriptions of the data fields.

Table 1. CSV file of Word Market data.

```
Vendor,Customer,Residence,WordPurchased,WordType,DatePurchased,Quantity,Price
c,Milo,,quagmire,n,10/31/03,1,98
a,Milo,,happy,a,1/1/04,20,30
b,Whether Man,Expectations,welcome,,12/3/03,235,45
b,Duke of Definition,Dictionopolis,nonsense,n,1/8/04,26,56
b,Earl of Essence,Dictionopolis,fantastic,a,1/8/04,26,71
a,Spelling Bee,Dictionopolis,vegetable,n,3/3/04,67,9
a,King Azaz ,Dictionopolis,unabridged,a,12/24/03,1200,67
b,Humbug,Dictionopolis,balderdash,n,11/17/03,34,87
a,Dodecahedron,Digitopolis,faces,n,2/23/04,100,65
b,Mathemagician,Digitopolis,numbers,n,11/16/03,500,20
c,Wordsnatcher,Mts of Ignorance,take,v,1/16/04,,0
c,Terrible Trivium,Mts of Ignorance,work,v,1/16/04,39,22
c,Senses Taker,Castle in the Air,sight,n,3/10/04,450,50
b,Princess Reason,Castle in the Air,learn,v,3/19/04,1500,75
b,Princess Rhyme,Castle in the Air,wonderful,a,12/30/03,2000,50
```

Reading the Data

To start your journey, your first task is to read the data and get it into a SAS® data set. The IMPORT procedure makes reading CSV files easy. The IMPORT procedure is available for Windows, UNIX and OpenVMS operating environments. (If you are using a z/OS system, you can read this file using the DATA step with the DSD option on the

INFILE statement.) The following program reads the data file, marketplace.csv, and creates a SAS data set named WORDSALES using the first line in the data file as variable names.

```
PROC IMPORT DATAFILE = 'C:\WordMarket\Data\marketplace.csv' OUT = wordsales
REPLACE;
RUN;
```

A Simple PROC PRINT

The PRINT procedure is a basic, easy to use, reporting tool available in SAS. A simple PROC PRINT, will format your data into nice columns, label the columns, and decide how to make the report best fit on the page. The basic PROC PRINT is an easy way to see if your data have been read correctly, and provides a simple report. The following, brief program will print the WORDSALES SAS data set. The results of the PROC PRINT are shown in Table 2.

```
PROC PRINT DATA = wordsales;
RUN;
```

Table 2. Results from simple PROC PRINT.

```
                                     The SAS System             09:49 Sunday, January 9, 2005   1



            Obs    Vendor    Customer                   Residence

              1      c       Milo
              2      a       Milo
              3      b       Whether Man                Expectations
              4      b       Duke of Definition         Dictionopolis
              5      b       Earl of Essence            Dictionopolis
              6      a       Spelling Bee               Dictionopolis
              7      a       King Azaz                  Dictionopolis
              8      b       Humbug                     Dictionopolis
              9      a       Dodecahedron               Digitopolis
             10      b       Mathemagician              Digitopolis
             11      c       Wordsnatcher               Mts of Ignorance
             12      c       Terrible Trivium           Mts of Ignorance
             13      c       Senses Taker               Castle in the Air
             14      b       Princess Reason            Castle in the Air
             15      b       Princess Rhyme             Castle in the Air

                   Word            Word          Date
            Obs    Purchased       Type     Purchased        Quantity          Price

              1    quagmire         n      10/31/2003               1             98
              2    happy            a      01/01/2004              20             30
              3    welcome                 12/03/2003             235             45
              4    nonsense         n      01/08/2004              26             56
              5    fantastic        a      01/08/2004              26             71
              6    vegetable        n      03/03/2004              67              9
              7    unabridged       a      12/24/2003            1200             67
              8    balderdash       n      11/17/2003              34             87
              9    faces            n      02/23/2004             100             65
             10    numbers          n      11/16/2003             500             20
             11    take             v      01/16/2004               .              0
             12    work             v      01/16/2004              39             22
             13    sight            n      03/10/2004             450             50
             14    learn            v      03/19/2004            1500             75
             15    wonderful        a      12/30/2003            2000             50
```

While this report is easier to read than the raw data, it certainly does not meet our expectations of what the finished report should look like. Here is perhaps where you find yourself in the Doldrums wondering what you can do to make this report better. When Milo found himself in the Doldrums, he discovered that the way to get out was to start thinking. So, let's think: the Customer and Residence information is interesting, but it is not relevant to this report. The sales for each vendor are all mixed up – that's a problem. And, what's "The SAS System" doing at the top of the report? Identifying problem areas is half the battle.

Sorting and Selecting Information to Print

After thinking his way out of the Doldrums, Milo soon finds himself at the Word Market. Milo discovers that you can select just about any word you want at the market, but some selections are better than others. For example, the letter X tastes like a truck full of stale air and, Milo learns, that's why people don't use X's very often. Milo also discovers how important it is to keep words sorted. After a commotion in the market place leaves all the words in a jumble, "no say could anything one correctly" (no one could say anything correctly).

Sorting your data is one of the easiest tasks you can do, and it can also be one of the most useful for organizing your data. In our case, we want to sort by Vendor so observations for each vendor will be together. Then within Vendor, we want to sort by DatePurchased so we can see the chronology of the purchases. Here are the SAS statements to sort the WORDSALES data set:

```
PROC SORT DATA=wordsales;
   BY Vendor DatePurchased;
RUN;
```

The BY statement tells SAS how to sort the observations. SAS sorts by the first variable, Vendor, then within Vendor, SAS sorts by the second variable, DatePurchased.

For this report, we decide to just look at sales to customers that live in Dictionopolis. An easy way to control which observations are used in a procedure is with the WHERE statement. For example, if you place the following statement in the PRINT procedure, SAS will only print the observations that satisfy the condition (customers that live in Dictionopolis).

```
WHERE Residence='Dictionopolis';
```

Sometimes you don't want to see all the variables in a data set when you print it. To print only selected variables, use the VAR statement in the PRINT procedure. Simply list the variables you want to print in the order that you want them to appear. For example:

```
VAR Vendor DatePurchased WordPurchased Quantity Price;
```

Also, by default, SAS prints the observation number as a column in your PRINT procedure output. If you don't care about the observation number, it may be distracting. To drop the Obs column from your output, use the NOOBS option to the PROC PRINT statement:

```
PROC PRINT DATA=wordsales NOOBS;
```

The last thing we want to do to improve the appearance of the report, is to give the report a title. By default, SAS uses "The SAS System" as a title for all output, and SAS prints the current date at the top of each page of output. The date can be very useful, but often it is distracting. To eliminate the date, use the NODATE system option:

```
OPTIONS NODATE;
```

The OPTIONS statement can go anywhere in your program, but it is usually best to put it as the first line.

Use the TITLE statement to give your report an appropriate title. You can put the TITLE statement in the procedure, or before the PROC statement. Enclose the text you want in quotes after the TITLE keyword:

```
TITLE 'Words Purchased by Residents of Dictionopolis';
```

Table 3 shows the results of the following SAS program that will sort the WORDSALES data set by Vendor and DatePurchased, then print the observations where Residence equals Dictionopolis for selected variables and give the report an appropriate title:

```
OPTIONS NODATE;

PROC SORT DATA=wordsales;
   BY Vendor DatePurchased;
RUN;

PROC PRINT DATA=wordsales NOOBS;
   WHERE Residence='Dictionopolis';
   VAR Vendor DatePurchased WordPurchased WordType Quantity Price;
   TITLE 'Words Purchased by Residents of Dictionopolis';
RUN;
```

Table 3. Results after sorting and selecting variables and observations.

```
                         Words Purchased by Residents of Dictionopolis                         2

                         Date      Word            Word
          Vendor    Purchased      Purchased       Type        Quantity            Price

            a      12/24/2003      unabridged       a              1200               67
            a      03/03/2004      vegetable        n                67                9
            b      11/17/2003      balderdash       n                34               87
            b      01/08/2004      nonsense         n                26               56
            b      01/08/2004      fantastic        a                26               71
```

This report is much better than the last, but there is still room for improvement. It would be nice to see the actual amount the customer paid for the words they bought (Quantity * Price), and since many of those numbers are in the thousands, as are some of the values for Quantity, it would be nice to see the numbers with commas to make them easier to read. Also while the date is perfectly readable as is, you know your boss prefers to see dates written as 01JAN2005, so it would be nice to change the format for the date as well.

Computing the Amount Paid

The amount the customers paid for their purchases can easily be computed in a DATA step. Use the SET statement to read the WORDSALES data set, then use an assignment statement to compute the new variable AmountPaid:

```
* Compute amount paid;
DATA wordsales;
   SET wordsales;
   AmountPaid = Quantity * Price;
RUN;
```

Formatting Values Using Standard Formats

In the city of Dictionopolis, Milo meets King Azaz's cabinet members: the Count of Connotation, the Minister of Meaning, the Earl of Essence, the Duke of Definition, and the Undersecretary of Understanding. Whenever any of the King's advisors say a word, each one of them says the same thing using a different word. Here Milo learns that there are often many ways to represent the same idea. In the SAS language, we use formats to represent data values in different ways.

SAS has many standard formats that you can use to change the appearance of the data values in your output. For example, there are many ways to represent dates: 10/31/04, 10-31-04, 31OCT2004, 31.10.04 to show a few. To change the format that SAS uses, you can assign a format to a variable in a FORMAT statement. You will almost always want to do this if you are using SAS date values. A SAS date value is the number of days since January 1st, 1960, and if there is no format assigned to a date variable, SAS will print just the number. You might wonder why the output we have shown so far, shows the dates in nice readable forms. The reason is that the IMPORT procedure assigns a date format (MMDDYY10.) to variables that appear to be dates. This is a nice feature of the IMPORT procedure. Since we want to print the date in the 01JAN2005 format, we will assign the variable, DatePurchased, the format DATE9.. Also, we will assign the COMMA8. format to AmountPaid and to Quantity. Here is the FORMAT statement:

```
FORMAT DatePurchased DATE9. Quantity AmountPaid COMMA8.;
```

Table 4 shows the results of the following program, which computes a new variable, AmountPaid, and uses formats to change the appearance of the values in the report.

```
DATA wordsales;
   SET wordsales;
   AmountPaid = Quantity * Price;
RUN;
PROC PRINT DATA=wordsales NOOBS;
   WHERE Residence='Dictionopolis';
   VAR Vendor DatePurchased WordPurchased WordType Quantity Price AmountPaid;
   FORMAT DatePurchased DATE9. Quantity AmountPaid COMMA8.;
   TITLE 'Words Purchased by Residents of Dictionopolis';
RUN;
```

Table 4. Word Market sales data using formats.

```
                         Words Purchased by Residents of Dictionopolis                         3

                      Date       Word            Word                                 Amount
      Vendor     Purchased       Purchased       Type       Quantity       Price        Paid

        a        24DEC2003       unabridged       a            1,200          67      80,400
        a        03MAR2004       vegetable        n               67           9         603
        b        17NOV2003       balderdash       n               34          87       2,958
        b        08JAN2004       nonsense         n               26          56       1,456
        b        08JAN2004       fantastic        a               26          71       1,846
```

Labeling Data Using User-Defined Formats

Our report is looking better, but it would be nice to know which Vendor is which. Vendor “a” is not very descriptive. Vendor “Spelling B Ranch” would be a lot better. In Dictionopolis, Milo meets Faintly Macabre, the Official Which. The job of the Official Which is to decide which words to use for every occasion. When Faintly was a young Which, she had trouble determining which words to use, and would often use too few. Faintly ended up in prison for dishing out so few words that no one could say a thing. When you write your SAS program to create a report, you can play the role of the Official Which by deciding what SAS prints. But don’t end up in Faintly’s position by using too few words.

Both the Vendor and WordType variables have coded values, which save space in data files, but are not very useful in reports. Here is a case where too little, just a single character, could put you in a difficult spot because no one will understand the report. To print out a better description of the data value, create a user-defined format using the FORMAT procedure, and apply the format to the variable in a FORMAT statement.

The following FORMAT procedure creates two character formats: $ven. and $wt.. When applied, the $ven. format causes SAS to print “Spelling B Ranch” instead of printing the actual variable’s value, “a”. Likewise, the $wt. format causes SAS to print “noun” instead of “n”. Notice that in the FORMAT procedure, the format names do not end in a period, but when you use them in a FORMAT statement, the formats must end in a period.

```
* Create formats for labels;
PROC FORMAT;
   VALUE $ven 'a' = 'Spelling B Ranch'
              'b' = 'Sentences Farm'
              'c' = 'Typos R Us';
   VALUE $wt  'n' = 'Noun'
              'v' = 'Verb'
              'a' = 'Adjective'
            other = 'Other';
RUN;
```

To apply the formats to the proper variables, put them in a FORMAT statement:

```
FORMAT Vendor $ven. WordType $wt.;
```

Table 5 shows the results of the following program which includes the new user-defined formats:

```
PROC PRINT DATA=wordsales NOOBS;
   WHERE Residence='Dictionopolis';
   VAR Vendor DatePurchased WordPurchased
       WordType Quantity Price AmountPaid;
   FORMAT DatePurchased DATE9. Quantity AmountPaid COMMA8.
          Vendor $ven. WordType $wt.;
   TITLE 'Words Purchased by Residents of Dictionopolis';
RUN;
```

Table 5. Word Market sales data using user-defined and standard formats.

```
                         Words Purchased by Residents of Dictionopolis                        4

                           Date       Word                                                Amount
         Vendor         Purchased     Purchased       WordType     Quantity          Price      Paid

   Spelling B Ranch     24DEC2003     unabridged      Adjective       1,200             67    80,400
   Spelling B Ranch     03MAR2004     vegetable       Noun               67              9       603
   Sentences Farm       17NOV2003     balderdash      Noun               34             87     2,958
   Sentences Farm       08JAN2004     nonsense        Noun               26             56     1,456
   Sentences Farm       08JAN2004     fantastic       Adjective          26             71     1,846
```

Using the ID and BY statements with PROC PRINT

When you want to print out your data and group them by values of a particular variable, then adding an ID and BY statement to your PROC PRINT can produce an appealing report. The ID statement tells SAS which variable to use to identify the row in the output. Normally, SAS will label each row with the observation number unless you use the NOOBS option like we have in the last several examples. Using the ID statement, SAS will, in place of the observation number, use the value of the ID variable for that observation. When you also include a BY statement with the same variable as in the ID statement, then you get a report which lists each unique value of the variable only once.

After inspection of your initial report, you decide that the date of the sale is not that important after all, and what you really want to see is a report showing all the purchases broken down by vendor, and including the customer name.

Table 6 shows the results of the following program which uses the variable Vendor for both the ID and BY statements. Notice that we have included the Customer variable, and have removed the subsetting WHERE statement from this program, so now all the observations are listed. The data set must be sorted by the variable listed in the BY statement. Because the WORDSALES data set has already been sorted by the Vendor variable, we do not need an additional PROC SORT in this program.

```
PROC PRINT DATA=wordsales;
   ID Vendor;
   BY Vendor;
   VAR Customer WordPurchased Quantity Price AmountPaid;
   FORMAT Quantity AmountPaid COMMA8.Vendor $ven.;
   TITLE 'Word Market Vendor Sales';
RUN;
```

Table 6. Word Market sales data using ID and BY statements.

```
                                   Word Market Vendor Sales                                  5

                                                 Word                                     Amount
  Vendor            Customer                     Purchased      Quantity         Price      Paid

  Spelling B Ranch  King Azaz                    unabridged        1,200            67    80,400
                    Milo                         happy                20            30       600
                    Dodecahedron                 faces               100            65     6,500
                    Spelling Bee                 vegetable            67             9       603

  Sentences Farm    Mathemagician                numbers             500            20    10,000
                    Humbug                       balderdash           34            87     2,958
                    Whether Man                  welcome             235            45    10,575
                    Princess Rhyme               wonderful         2,000            50   100,000
                    Duke of Definition           nonsense             26            56     1,456
                    Earl of Essence              fantastic            26            71     1,846
                    Princess Reason              learn             1,500            75   112,500

  Typos R Us        Milo                         quagmire              1            98        98
                    Wordsnatcher                 take                  .             0         .
                    Terrible Trivium             work                 39            22       858
                    Senses Taker                 sight               450            50    22,500
```

Creating HTML Output and Choosing Styles

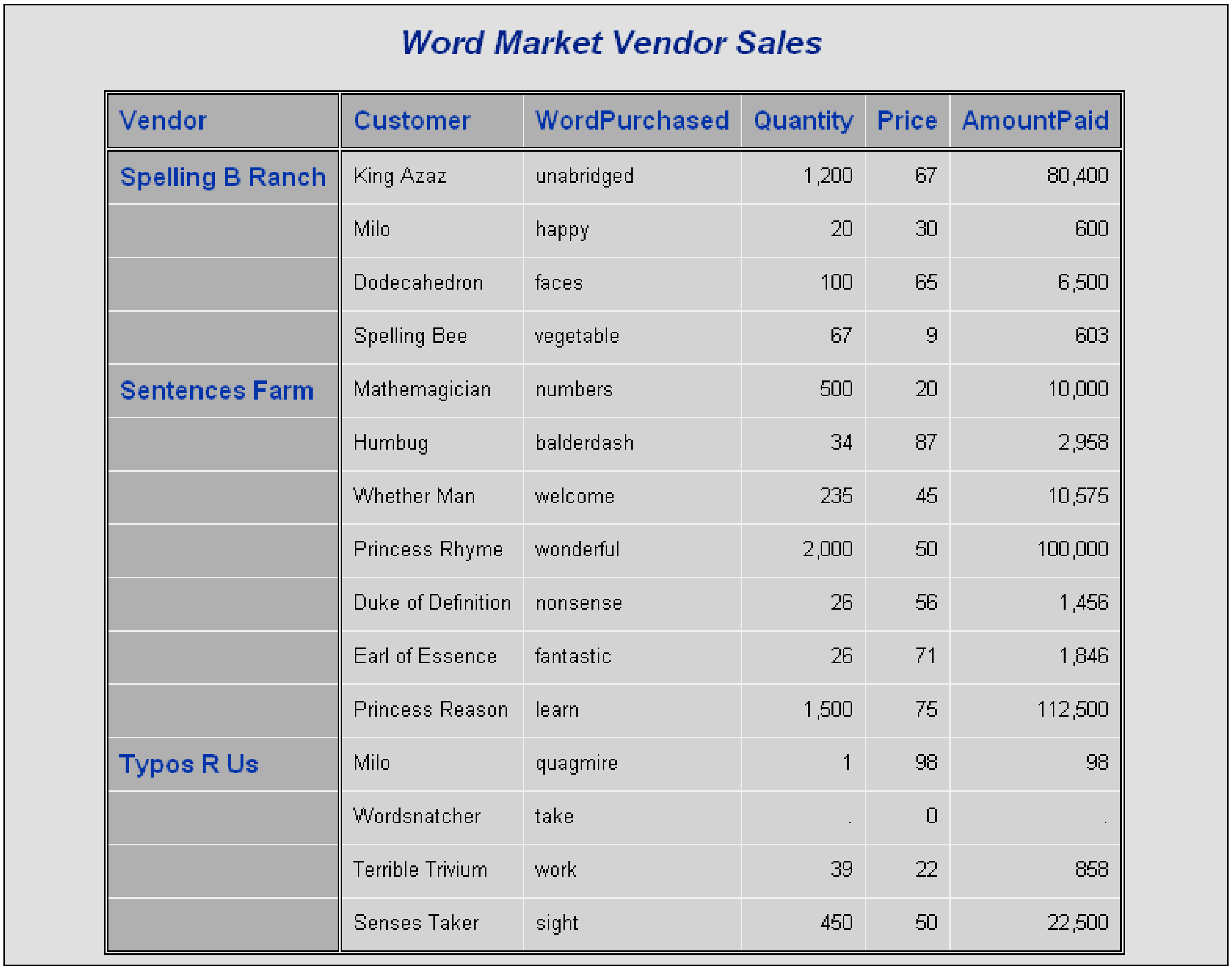
Now we have a nice simple looking report with the information that we want, but it could use some style and color. One of many obstacles Milo faces on his way to rescue the princesses, is the Senses Taker. The Senses Taker fools Milo and his traveling companions by making them see things that aren't there, smell odors that don't exist, and hear sounds never made. Milo learns how important the senses are, and that something visually appealing can be quite attractive.

In addition to text output, you can also easily create HTML reports. HTML reports are great for posting results on the Web, and for e-mailing results to colleagues. The HTML files produced from SAS procedures have more style than the regular output listing, and they have color.

You can create HTML files programmatically, but if you are using a Windows version of SAS, you can simply create HTML output by changing a setting in your menu items. Go to Tools►Options►Preferences from the pull-down menus, then select the Results tab. Put a check mark next to "Create HTML". Now HTML output will appear when you run your SAS programs. To save the HTML output to a file, make the HTML Results Viewer window active (click on it), then click on the File menu item and choose "Save As…".

Table 7 shows the previous output using the DEFAULT HTML style. To change styles for the HTML files, go to the same Results tab (Tools►Options►Preferences) and choose a style from the Style list. Table 8 shows what the BARRETTSBLUE style looks like.

Table 7. HTML report using the DEFAULT style.



Word Market Vendor Sales

Vendor	Customer	WordPurchased	Quantity	Price	AmountPaid
Spelling B Ranch	King Azaz	unabridged	1,200	67	80,400
	Milo	happy	20	30	600
	Dodecahedron	faces	100	65	6,500
	Spelling Bee	vegetable	67	9	603
Sentences Farm	Mathemagician	numbers	500	20	10,000
	Humbug	balderdash	34	87	2,958
	Whether Man	welcome	235	45	10,575
	Princess Rhyme	wonderful	2,000	50	100,000
	Duke of Definition	nonsense	26	56	1,456
	Earl of Essence	fantastic	26	71	1,846
	Princess Reason	learn	1,500	75	112,500
Typos R Us	Milo	quagmire	1	98	98
	Wordsnatcher	take	.	0	.
	Terrible Trivium	work	39	22	858
	Senses Taker	sight	450	50	22,500

If you are not using SAS in the Windows operating environment, or if you want more control over the HTML files SAS creates, you can use ODS HTML statements in your program. To generate HTML output all you need are two statements—one to open the HTML file, and one to close it. There are other types of ODS statements, and other types of HTML files that SAS can write, not to mention other types of output (such as PostScript, Rich Text Format, and output data sets). However, in this example we simply want to create an HTML file containing the report. We do that with these two basic statements.

```
    ODS HTML FILE='path\filename.html';

    ODS HTML CLOSE;
```

The ODS HTML statement opens a file, called a body file, which will contain the report. With the ODS HTML statement, you can create other HTML files, such as a table of contents, but in this paper we just want the report itself. The ODS HTML CLOSE statement closes the file when the report is done. Generally speaking, you want to put the first statement just before the procedure, and the closing statement just after the RUN statement, but before any other procedures. You can specify a different style in the ODS statement using a STYLE= option:

```
    ODS HTML FILE='path\filename.html' STYLE=style-name;
```

where style-name is the name of one of the styles provided with SAS software. The following are just a few of the styles that come with SAS 9.

Some ODS Styles

DEFAULT
ANALYSIS
MONEY
BROWN
BARRETTSBLUE
BRICK
STATDOC
CURVE
SASWEB
MINIMAL
FANCYPRINTER
STATISTICAL
D3D
JOURNAL
WATERCOLOR

Here are the same PROC PRINT statements, this time including ODS HTML statements, and choosing the BARRETTSBLUE style.

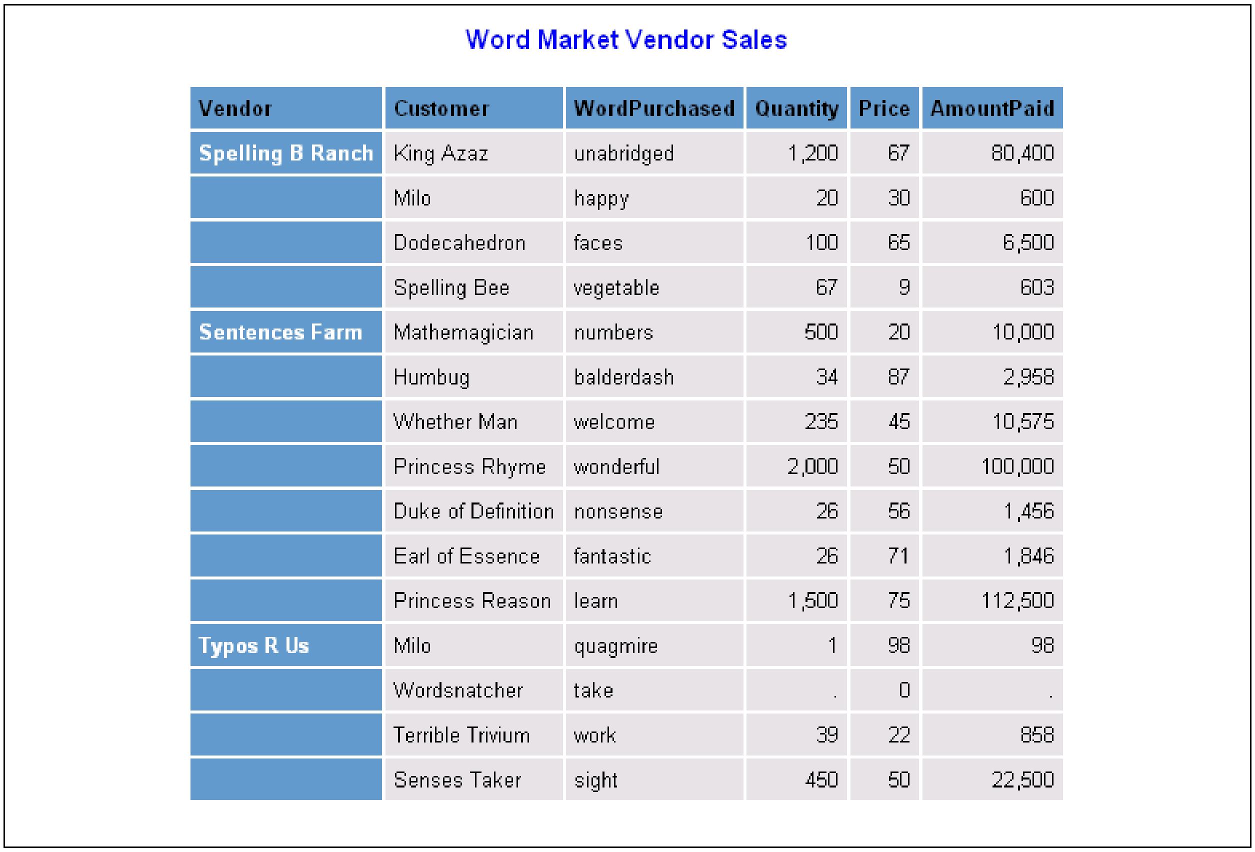
```
ODS HTML FILE='C:\WordMarket\Reports\VendorSales1.html' STYLE=barrettsblue;

PROC PRINT DATA=wordsales;
   ID Vendor;
   BY Vendor;
   VAR Customer WordPurchased Quantity Price AmountPaid;
   FORMAT Quantity AmountPaid COMMA8. Vendor $ven.;
   TITLE 'Word Market Vendor Sales';
RUN;

ODS HTML CLOSE;
```

SAS will write the output from PROC PRINT in a file named VendorSales1.html in a directory named WordMarket\Reports on the C drive. The resulting HTML file is the same as the file you would get by using the HTML option in the pull down menus and choosing the BARRETTSBLUE style. The results are shown in Table 8.

Table 8. HTML report using the BARRETTSBLUE style.



Word Market Vendor Sales

Vendor	Customer	WordPurchased	Quantity	Price	AmountPaid
Spelling B Ranch	King Azaz	unabridged	1,200	67	80,400
	Milo	happy	20	30	600
	Dodecahedron	faces	100	65	6,500
	Spelling Bee	vegetable	67	9	603
Sentences Farm	Mathemagician	numbers	500	20	10,000
	Humbug	balderdash	34	87	2,958
	Whether Man	welcome	235	45	10,575
	Princess Rhyme	wonderful	2,000	50	100,000
	Duke of Definition	nonsense	26	56	1,456
	Earl of Essence	fantastic	26	71	1,846
	Princess Reason	learn	1,500	75	112,500
Typos R Us	Milo	quagmire	1	98	98
	Wordsnatcher	take	.	0	.
	Terrible Trivium	work	39	22	858
	Senses Taker	sight	450	50	22,500

Other types of output

So far in this paper we have shown you two types of output: listing and HTML. The listing is what you see in the OUTPUT window in interactive mode, or in the LISTING or OUTPUT file, if you use batch mode. You can create HTML output using either the ODS HTML statements, or by choosing HTML in the Tools►Options►Preferences pull down menu (Windows only). Using ODS statements you can send your results to many types of destinations including:

- LISTING
- HTML
- PRINTER
- PDF
- PS
- PCL
- RTF

The PRINTER destination, by default, produces whatever type of output the current system printer uses such as PostScript. By default, PRINTER also sends output to your printer automatically, or using options, you can send PRINTER output to a file. The PDF destination produces files in Portable Document Format which can be read using Adobe Acrobat software. The PS destination produces PostScript output, while the PCL destination produces Printer Control Language output, both of which are meant for printing. RTF output becomes a Word table when opened in a Word document, and as a Word table it can be easily resized and edited.

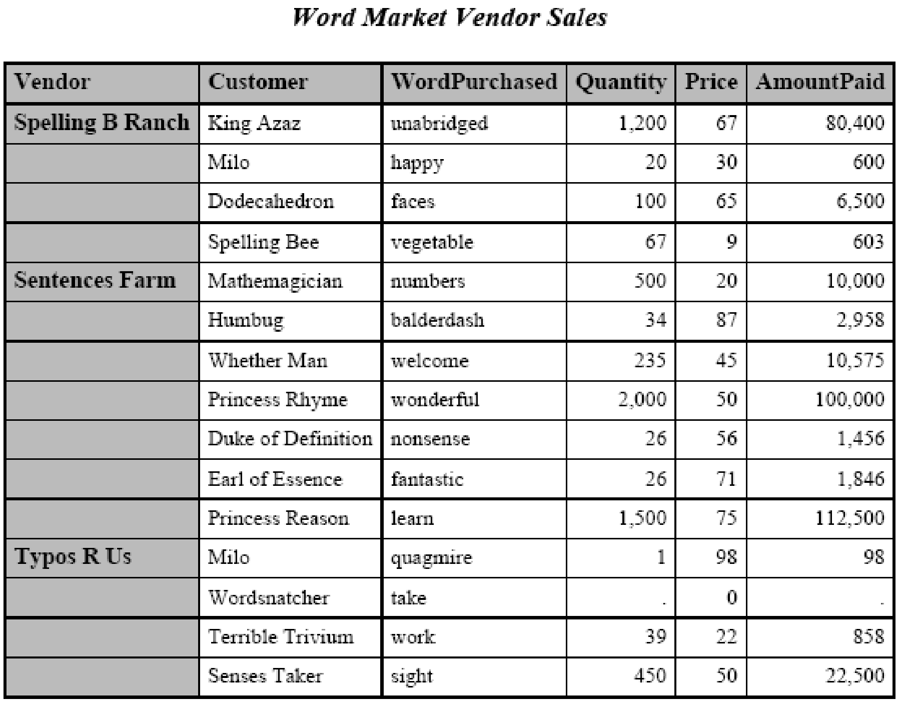
Which destination you use depends on your needs and what printers are available to you. HTML gives you more choices (such as more colors) than other destinations, but the other destinations have advantages too. HTML reports are great for posting results on the Web, and for e-mailing reports to colleagues who can read HTML attachments. On the other hand, if you need to print long reports, the other destinations will give you logical page breaks. Because it’s designed to be displayed on a screen rather than a page, HTML doesn’t provide page breaks.
Table 9 shows the results of the following program which uses the same PROC PRINT statements, but this time uses the ODS PDF statement to create a pdf file.

```
ODS PDF FILE='C:\WordMarket\Reports\VendorSales1.pdf';

PROC PRINT DATA=wordsales;
   ID Vendor;
   BY Vendor;
   VAR Customer WordPurchased Quantity Price AmountPaid;
   FORMAT Quantity AmountPaid COMMA8.Vendor $ven.;
   TITLE 'Word Market Vendor Sales';
RUN;

ODS PDF CLOSE;
```

Table 9. PDF report using the default style for pdf files.



Word Market Vendor Sales

Vendor	Customer	WordPurchased	Quantity	Price	AmountPaid
Spelling B Ranch	King Azaz	unabridged	1,200	67	80,400
	Milo	happy	20	30	600
	Dodecahedron	faces	100	65	6,500
	Spelling Bee	vegetable	67	9	603
Sentences Farm	Mathemagician	numbers	500	20	10,000
	Humbug	balderdash	34	87	2,958
	Whether Man	welcome	235	45	10,575
	Princess Rhyme	wonderful	2,000	50	100,000
	Duke of Definition	nonsense	26	56	1,456
	Earl of Essence	fantastic	26	71	1,846
	Princess Reason	learn	1,500	75	112,500
Typos R Us	Milo	quagmire	1	98	98
	Wordsnatcher	take	.	0	.
	Terrible Trivium	work	39	22	858
	Senses Taker	sight	450	50	22,500

Summarizing Data in your Report

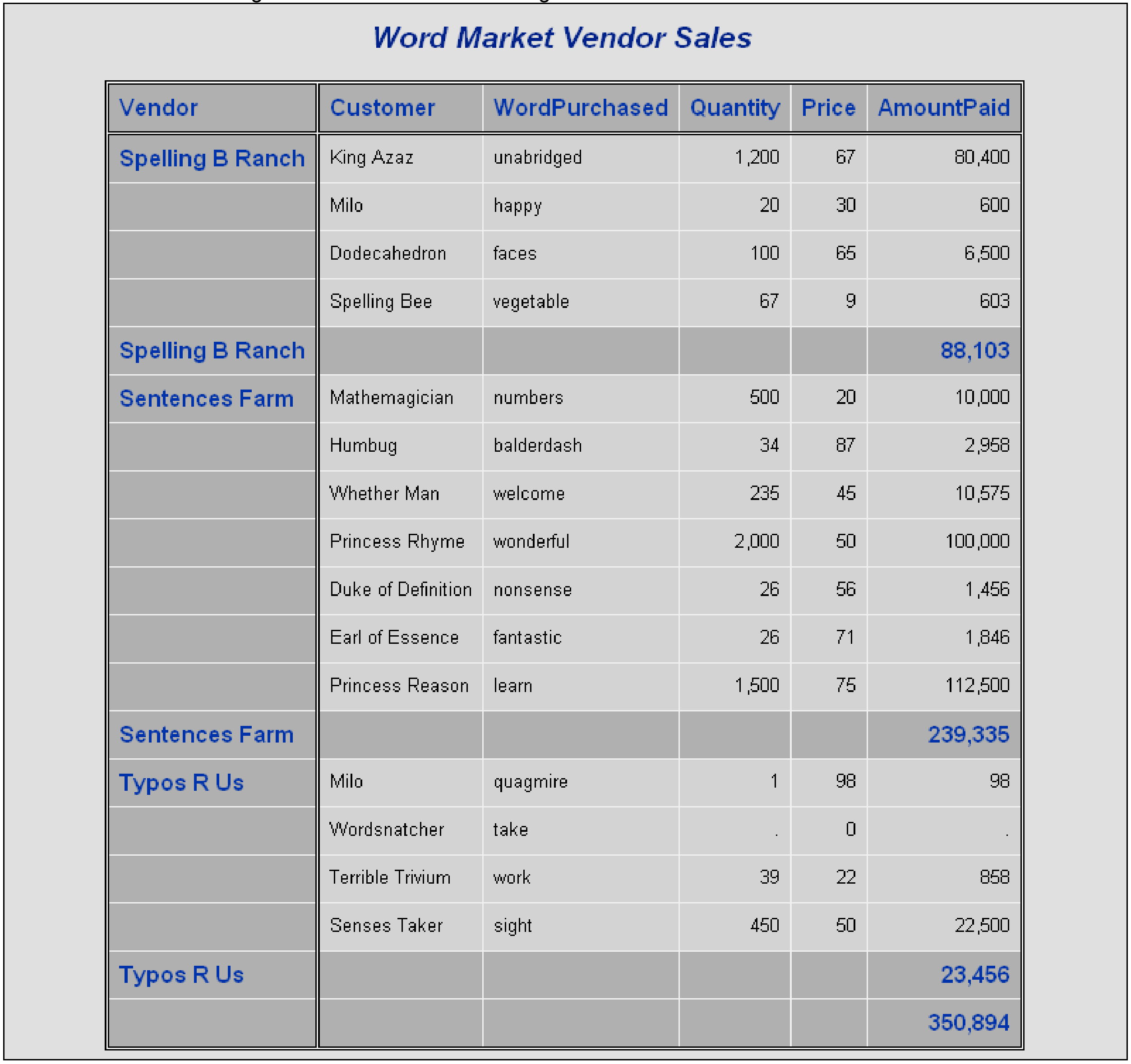
When Milo visits the city of Digitopolis, he learns about the importance of numbers. The Mathemagician shows him around the numbers mine, and serves him subtraction stew (the more you eat the hungrier you get). And, while trying to reach Infinity, Milo meets half a boy, neatly divided from top to bottom. The boy explains that he is from the average family with 2.58 children, and he is the .58. Milo begins to realize that sums and averages—and all things mathematical—are important in understanding the world he lives in.

There are several SAS procedures that will summarize data, but the PRINT procedure is useful if, in addition to summary information, you also want to print individual observations. You can add totals to PROC PRINT output by using a SUM statement. In the SUM statement you simply list the variables you want totaled. If you also specify a BY statement, you will get subtotals for each value of the by-variables plus a grandtotal at the end of the report. Without a BY statement, SAS gives you just the grand total.

The results of the following program are shown in Table 10. This program, which includes the BY, ID, and SUM statements, now summarizes the AmountPaid by Vendor:

```
PROC PRINT DATA=wordsales NOOBS;
   BY vendor;
   ID vendor;
   SUM AmountPaid;
   VAR Customer WordPurchased Quantity Price AmountPaid;
   FORMAT Quantity AmountPaid COMMA8.Vendor $ven.;
   TITLE 'Word Market Vendor Sales';
RUN;
```

Table 10. Results showing how to summarize data using PROC PRINT.



Word Market Vendor Sales

Vendor	Customer	WordPurchased	Quantity	Price	AmountPaid
Spelling B Ranch	King Azaz	unabridged	1,200	67	80,400
	Milo	happy	20	30	600
	Dodecahedron	faces	100	65	6,500
	Spelling Bee	vegetable	67	9	603
Spelling B Ranch					88,103
Sentences Farm	Mathemagician	numbers	500	20	10,000
	Humbug	balderdash	34	87	2,958
	Whether Man	welcome	235	45	10,575
	Princess Rhyme	wonderful	2,000	50	100,000
	Duke of Definition	nonsense	26	56	1,456
	Earl of Essence	fantastic	26	71	1,846
	Princess Reason	learn	1,500	75	112,500
Sentences Farm					239,335
Typos R Us	Milo	quagmire	1	98	98
	Wordsnatcher	take	.	0	.
	Terrible Trivium	work	39	22	858
	Senses Taker	sight	450	50	22,500
Typos R Us					23,456
					350,894

Customizing Reports using STYLE= options

When Milo visits the forest of sight, he meets Chroma the Great and learns the importance of colors. Chroma conducts the orchestra that “plays” the colors of the kingdom. Each morning, Chroma conducts the sunrise and the orchestra plays all day through the sunset. When Milo takes a turn conducting, he learns how terribly things can go wrong: flowers turn black, rocks are chartreuse, even his trusty travel companion, Tock the Watch Dog, turns to a magnificent ultra-marine. Colors are important, but the wrong color in the wrong place can send the wrong message.

The STYLE= option is one way to customize output and add color to your reports. When used in an ODS statement, the STYLE= option controls the overall look of a report. You can also use the STYLE= option in PROC PRINT (and in PROC REPORT and PROC TABULATE, but not in other procedures) to control specific features of your reports.

In the PRINT procedure, you can use STYLE= options in the PROC PRINT statement and using a location list, you specify which location(s) in your report you want the style to apply to:

```
PROC PRINT STYLE(location-list)= {style-attribute = value};
```

The location-list can take on the following values:

Location	**Table region affected**
DATA	all the data cells
HEADER	the column headers (variable names)
OBS	the data in the Obs or ID column
OBSHEADER	the header for the Obs or ID column
TOTAL	the by-group totals produced by a SUM statement
GRANDTOTAL	the grand total produced by a SUM statement

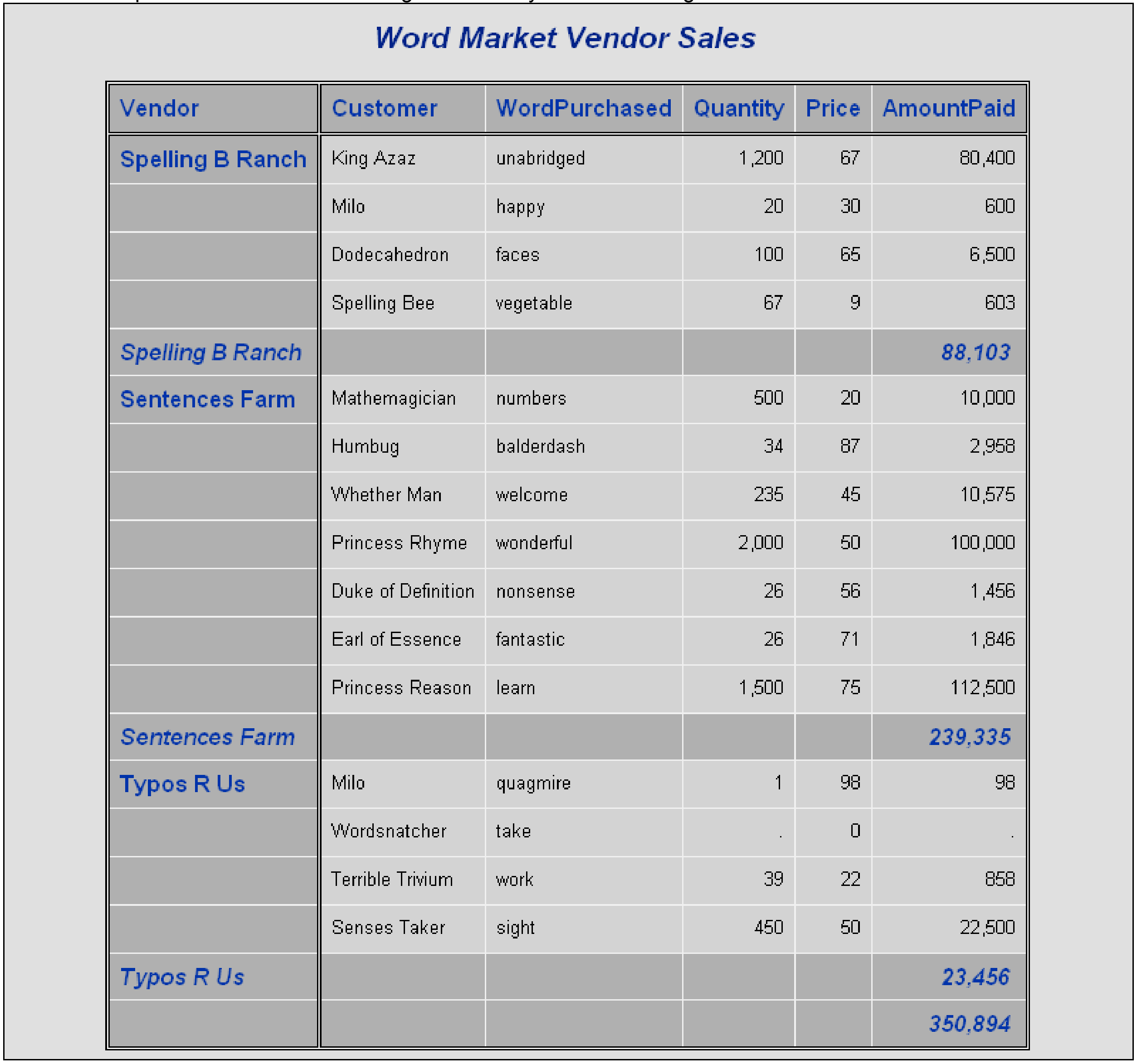
Here are a few examples of the attributes that you can control with the STYLE= option in PROC PRINT.

Style-attribute	**Example of value**
BACKGROUND	BLACK
FOREGROUND	ORANGE
FONT_FACE	COURIER
FONT_WEIGHT	BOLD
FONT_STYLE	ITALIC
JUST	C
VJUST	C

Table 11 shows the results of the following program which uses the STYLE= option in the PROC PRINT statement to make the by-group totals and grand total appear in italic font.

```
PROC PRINT DATA=wordsales STYLE(TOTAL GRANDTOTAL)= {FONT_STYLE=ITALIC};
   BY vendor;
   ID vendor;
   SUM AmountPaid;
   VAR Customer WordPurchased Quantity Price AmountPaid;
   FORMAT Quantity AmountPaid COMMA8. Vendor $ven.;
   TITLE 'Word Market Vendor Sales';
RUN;
```

Table 11. Output from PROC PRINT using italic font style for total and grand total rows.



Word Market Vendor Sales

Vendor	Customer	WordPurchased	Quantity	Price	AmountPaid
Spelling B Ranch	King Azaz	unabridged	1,200	67	80,400
	Milo	happy	20	30	600
	Dodecahedron	faces	100	65	6,500
	Spelling Bee	vegetable	67	9	603
Spelling B Ranch					*88,103*
Sentences Farm	Mathemagician	numbers	500	20	10,000
	Humbug	balderdash	34	87	2,958
	Whether Man	welcome	235	45	10,575
	Princess Rhyme	wonderful	2,000	50	100,000
	Duke of Definition	nonsense	26	56	1,456
	Earl of Essence	fantastic	26	71	1,846
	Princess Reason	learn	1,500	75	112,500
Sentences Farm					*239,335*
Typos R Us	Milo	quagmire	1	98	98
	Wordsnatcher	take	.	0	.
	Terrible Trivium	work	39	22	858
	Senses Taker	sight	450	50	22,500
Typos R Us					*23,456*
					350,894

You can also use the STYLE= option in the VAR statement in PROC PRINT.

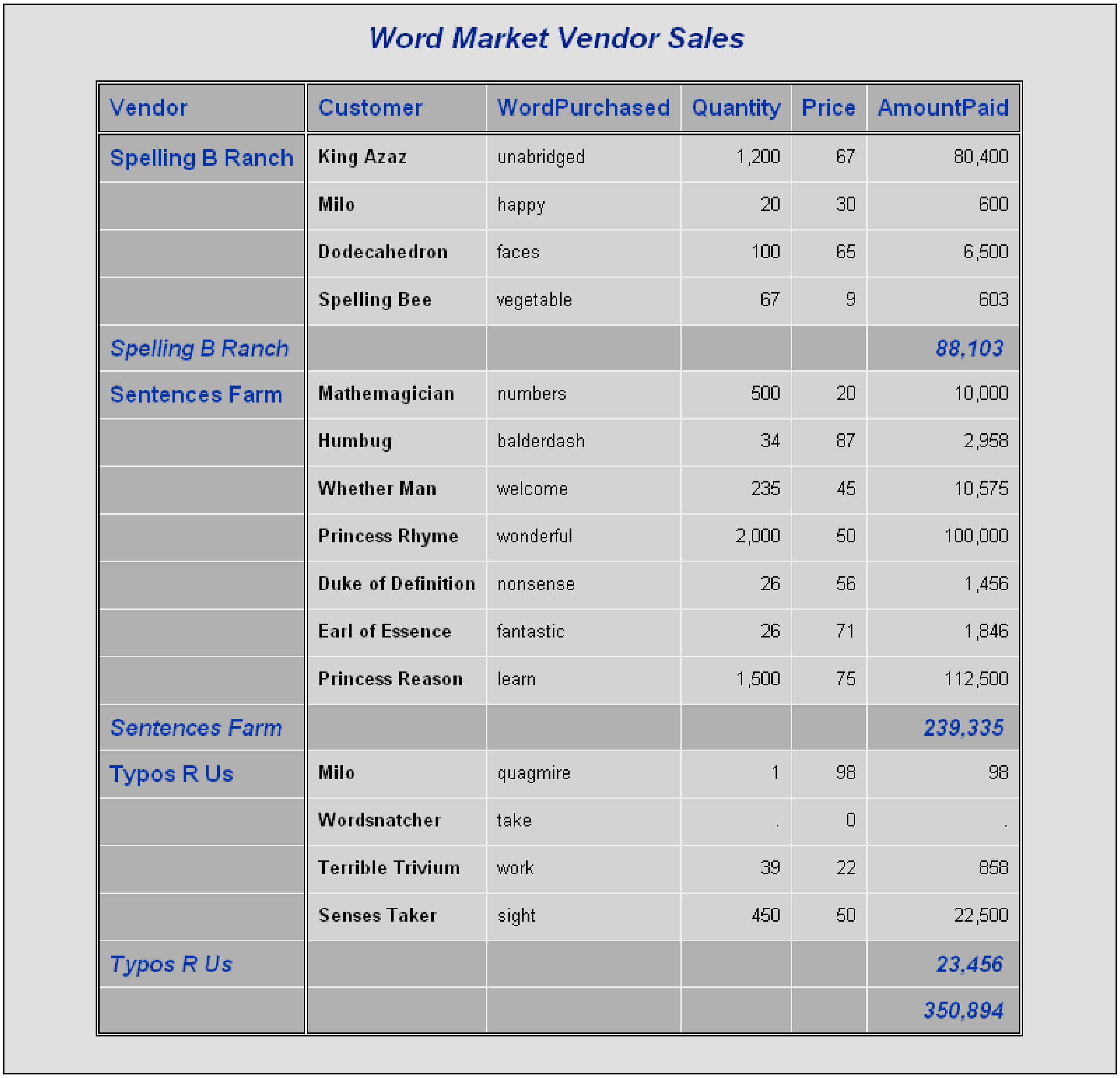
```
    VAR   variable-list / STYLE(location-list) = {style-attribute = value};
```

When you use the STYLE= option in the VAR statement, then the style only applies to the variables listed in the VAR statement, and the only valid locations are DATA and HEADER.

Table 12 shows the results of the following program which, in addition to using the STYLE= option in the PROC PRINT statement, uses the STYLE= option in a VAR statement to make the customer names appear in bold face. Because we only want the style to apply to one of the variables, Customer, it appears in its own VAR statement. The remaining variables are listed in a separate VAR statement.

```
PROC PRINT DATA=wordsales STYLE(TOTAL GRANDTOTAL)= {FONT_STYLE=ITALIC};
   BY vendor;
   ID vendor;
   SUM AmountPaid;
   VAR Customer/STYLE={FONT_WEIGHT=BOLD};
   VAR WordPurchased Quantity Price AmountPaid;
   FORMAT Quantity AmountPaid COMMA8. Vendor $ven.;
   TITLE 'Word Market Vendor Sales';
RUN;
```

Table 12. Output from PROC PRINT using bold type face for variable Customer.



Word Market Vendor Sales

Vendor	Customer	WordPurchased	Quantity	Price	AmountPaid
Spelling B Ranch	**King Azaz**	unabridged	1,200	67	80,400
	Milo	happy	20	30	600
	Dodecahedron	faces	100	65	6,500
	Spelling Bee	vegetable	67	9	603
Spelling B Ranch					*88,103*
Sentences Farm	**Mathemagician**	numbers	500	20	10,000
	Humbug	balderdash	34	87	2,958
	Whether Man	welcome	235	45	10,575
	Princess Rhyme	wonderful	2,000	50	100,000
	Duke of Definition	nonsense	26	56	1,456
	Earl of Essence	fantastic	26	71	1,846
	Princess Reason	learn	1,500	75	112,500
Sentences Farm					*239,335*
Typos R Us	**Milo**	quagmire	1	98	98
	Wordsnatcher	take	.	0	.
	Terrible Trivium	work	39	22	858
	Senses Taker	sight	450	50	22,500
Typos R Us					*23,456*
					350,894

Adding traffic-lighting with a STYLE= option

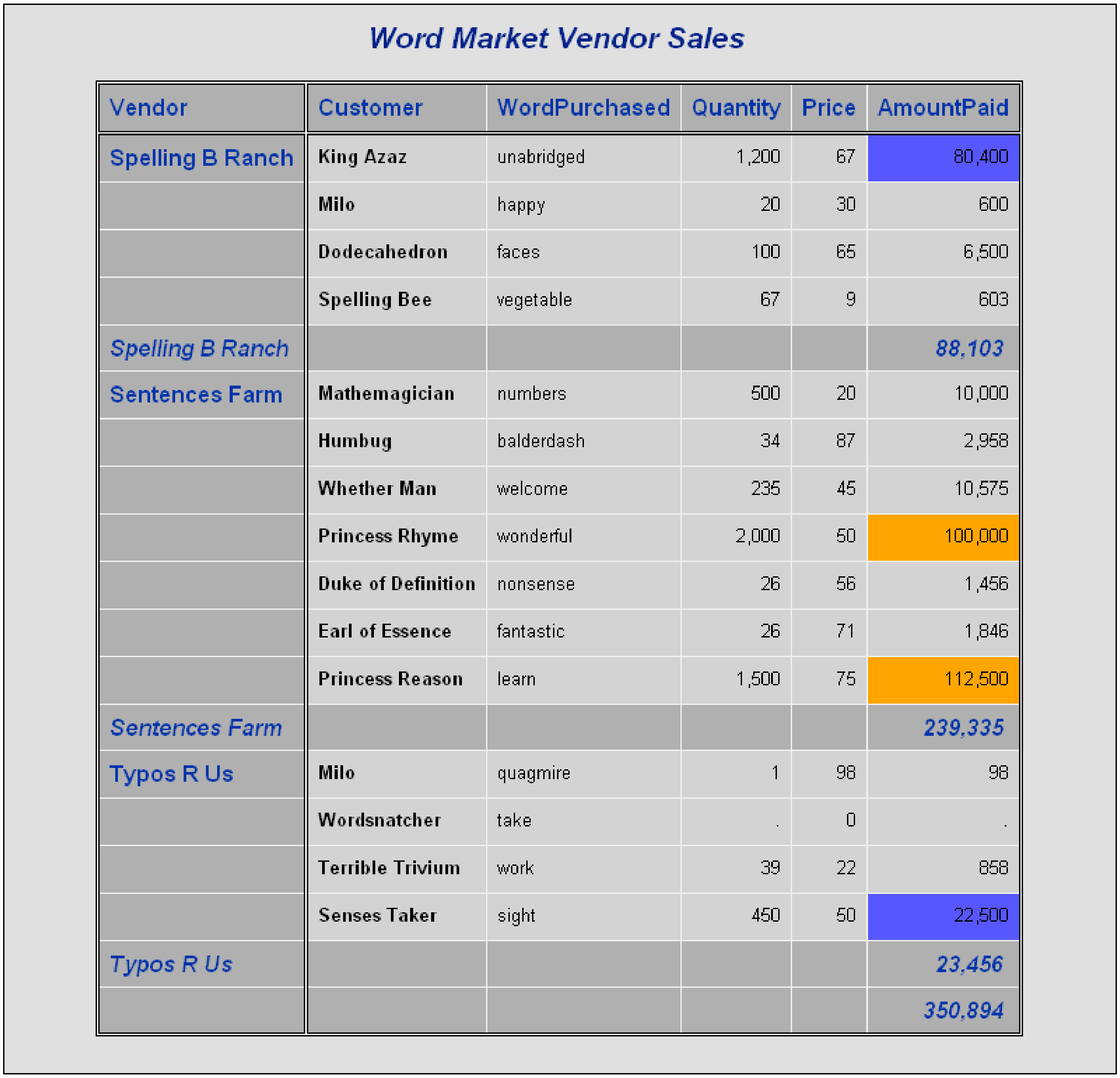
One of the most impressive effects you can create with the STYLE= option is traffic-lighting. With traffic-lighting, the values of the data determine their appearance. In this way, you can draw attention to the most important data, or highlight relationships between the values. To add traffic-lighting, you create a user-defined format specifying different style attributes for each range of values, and then use that format as the value of a style element.

The following program is exactly like the preceding one except that a VAR statement has been created for the AmountPaid variable and a STYLE= option has been added to it. A user defined format, PAID., is created by a PROC FORMAT. Then the format is used as the value for BACKGROUND= in the STYLE= option in the VAR statement for the AmountPaid variable. The report produced by this PROC PRINT appears in Table 13.

```
* Create user-defined formats for traffic-lighting;
PROC FORMAT;
   VALUE paid  20000-<100000 = 'light blue'
               100000-HIGH = 'orange'
               OTHER = 'light grey';
RUN;

PROC PRINT DATA=wordsales STYLE(TOTAL GRANDTOTAL)= {FONT_STYLE=ITALIC};
   BY vendor;
   ID vendor;
   SUM AmountPaid;
   VAR Customer/STYLE={FONT_WEIGHT=BOLD};
   VAR WordPurchased Quantity Price;
   VAR AmountPaid/STYLE={Background=paid.};
   FORMAT Quantity AmountPaid COMMA8. Vendor $ven.;
   TITLE 'Word Market Vendor Sales';
RUN;
```

Table 13. Output from PROC PRINT using traffic-lighting for the AmountPaid variable.



Word Market Vendor Sales

Vendor	Customer	WordPurchased	Quantity	Price	AmountPaid
Spelling B Ranch	**King Azaz**	unabridged	1,200	67	80,400
	Milo	happy	20	30	600
	Dodecahedron	faces	100	65	6,500
	Spelling Bee	vegetable	67	9	603
Spelling B Ranch					***88,103***
Sentences Farm	**Mathemagician**	numbers	500	20	10,000
	Humbug	balderdash	34	87	2,958
	Whether Man	welcome	235	45	10,575
	Princess Rhyme	wonderful	2,000	50	100,000
	Duke of Definition	nonsense	26	56	1,456
	Earl of Essence	fantastic	26	71	1,846
	Princess Reason	learn	1,500	75	112,500
Sentences Farm					***239,335***
Typos R Us	**Milo**	quagmire	1	98	98
	Wordsnatcher	take	.	0	.
	Terrible Trivium	work	39	22	858
	Senses Taker	sight	450	50	22,500
Typos R Us					***23,456***
					350,894

Conclusions

After rescuing Rhyme and Reason, Milo returns home. Inspired by his adventures, Milo finds that his life now has meaning. We hope that you feel inspired to find rhyme and reason in your data. By sorting, selecting, and summarizing; and by controlling style, font, and color, you can reduce a jumble of data into concise, meaningful, and polished reports.

Trademark Citation

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 of their respective companies.

References

Delwiche, Lora D. and Susan J. Slaughter (2008). *The Little SAS® Book: A Primer, Fourth Edition*. Cary, NC: SAS Institute Inc.

Norton, Juster (1961). *The Phantom Tollbooth*. Random House, NY.

About the Authors

Lora Delwiche and Susan Slaughter are authors of *The Little SAS® Book: A Primer* published by SAS Institute. They may be contacted at:

Lora D. Delwiche	(530) 752-9321 llddelwiche@ucdavis.edu
Susan J. Slaughter	(530) 756-8434 susan@avocetsolutions.com