

**Change is Good, or at Least Expected:
Techniques for Visualizing Categorical Values Over Time**
Lisa Horwitz, SAS Institute, Inc.

ABSTRACT

Many techniques exist for showing how numeric values change over time. Bar charts, line charts, plots, and many other graph types are all excellent ways to demonstrate how temperature, expenses, and other measures increase and decrease over minutes, months or decades. On the other hand, such graphs don't lend themselves to showing how and when categorical values such as grade, rating, score, and status change over time. A simple combination of data manipulation, file merging, custom formatting, and the Output Delivery System (ODS) can produce a wide range of useful, easy-to-interpret, and effective reports. By using color, fonts, custom messages, and other features to indicate a change in a data value, these reports make it easy to monitor progress or to detect when things are going in the wrong direction.

INTRODUCTION

As Partner Program Manager in the SAS Alliances and Channels organization, I monitor the progress of SAS® partner organizations in the acquisition of SAS Global Certification Program certifications and in the meeting of SAS Partner Program requirements. For our partners, it's a mark of distinction to have ever-increasing numbers of consultants who are well-trained in a wide range of SAS technologies and who have passed SAS certification exams.

Reporting on the number of certifications that a partner's consultants has can be done in a variety of ways. Numeric values can be tracked in bar and line charts and in other graphical formats, and these reports clearly show upward or downward trends across time. An upward trend shows an organization whose employees are passing more certification exams or who is hiring consultants that bring their certifications with them. A downward trend is usually not good; it indicates that consultants have left the organization, taking their certifications with them.

Depending on a partner organization's tier (Silver, Gold, or Platinum), a certain number of certifications are required for the partner to be in good standing at the yearly program renewal time. Tracking a partner's status from month to month requires a reporting format that can show me exactly when the requirement for number of certifications is met and whether the status reverts to "not met" due to employee turnover or other factors. This paper describes two variations on trafficlighting reports with a time interval that grew out of this requirement.

REPORT #1: MONITORING STATUS CHANGE FROM MONTH TO MONTH

Before talking about how the status change report is created, let's look at report samples for four months (four "time intervals" – your time interval might be minutes or weeks or any other interval rather than months) to see how status changes are depicted. I'm using a few rows of sample data for illustration.

JANUARY REPORT

Partners who have met their credential requirement are displayed in green. Partners who have not yet met their credential requirement are displayed in yellow:

Partner Name	January
Partner1	Has Met Requirement
Partner2	Has Met Requirement
Partner3	Has Not Met Requirement
Partner4	Has Met Requirement
Partner5	Has Not Met Requirement
Partner6	Has Met Requirement
Partner7	Has Met Requirement
Partner8	Has Not Met Requirement

Figure 1. January Report

FEBRUARY REPORT

Data from February is displayed next to the data from January. Thanks to the trafficlighting, it's easy to see the change in status from one time interval to the next. Can you tell what changes have occurred?

Partner Name	January	February
Partner1	Has Met Requirement	Has Met Requirement
Partner2	Has Met Requirement	Has Met Requirement
Partner3	Has Not Met Requirement	Has Met Requirement
Partner4	Has Met Requirement	Has Met Requirement
Partner5	Has Not Met Requirement	Has Not Met Requirement
Partner6	Has Met Requirement	Has Met Requirement
Partner7	Has Met Requirement	Has Met Requirement
Partner8	Has Not Met Requirement	Has Not Met Requirement
Partner9		Has Not Met Requirement

Figure 2. February Report

This report shows that Partner3 has now met their requirement, Partner5 and Partner8 have still not met their requirements, and Partner9 joined the SAS Partner Program and has not yet met their requirement.

MARCH REPORT

Data from March is displayed next to the data from January and February. What changes have occurred?

Partner Name	January	February	March
Partner1	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner2	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner3	Has Not Met Requirement	Has Met Requirement	Has Met Requirement
Partner4	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner5	Has Not Met Requirement	Has Not Met Requirement	
Partner6	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner7	Has Met Requirement	Has Met Requirement	
Partner8	Has Not Met Requirement	Has Not Met Requirement	Has Met Requirement
Partner9		Has Not Met Requirement	Has Not Met Requirement
PartnerSev			Has Met Requirement

Figure 3. March Report

Partner8, after not meeting their requirement for two months, has now met their requirement. But what happened to Partner5, who had not met their requirement for two months, and Partner7, who had met their requirement for two months? Also, who is PartnerSev, and how have they seemingly come into the SAS Partner Program already meeting their requirement?

When the report shows unexpected results, such as data for two partners “disappearing” and another partner appearing with requirements already met, some research might be necessary to understand what these changes mean and how to interpret these results. In addition to the next month’s data, I added some descriptive text to the report to explain the missing data.

APRIL REPORT

Partner5 became inactive in March, so I added a note to explain the absence of data for them starting in March. Partner7 is now doing business as (DBA) PartnerSev as of March, so I added a note to explain the absence of data for Partner7 starting in March and to direct our attention to PartnerSev, for all subsequent news about their status. Another new partner, Partner10, has joined the program.

Partner Name	January	February	March	April
Partner1	Has Met Requirement	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner2	Has Met Requirement	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner3	Has Not Met Requirement	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner4	Has Met Requirement	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner5	Has Not Met Requirement	Has Not Met Requirement	Inactive as of 15Mar	
Partner6	Has Met Requirement	Has Met Requirement	Has Met Requirement	Has Met Requirement
Partner7	Has Met Requirement	Has Met Requirement	Now DBA PartnerSev	
Partner8	Has Not Met Requirement	Has Not Met Requirement	Has Met Requirement	Has Met Requirement
Partner9		Has Not Met Requirement	Has Not Met Requirement	Has Not Met Requirement
Partner10				Has Not Met Requirement
PartnerSev			Has Met Requirement	Has Met Requirement

Figure 4. April Report

These traffic light reports with a time interval enable us to easily monitor progress over time and detect anomalies.

HOW TO CREATE THE REPORT

Step 1. Identify and Validate the ID Fields in the Data

This report requires that each time interval file contains a consistent identification field or set of fields. The identification field is used to match-merge with the previous time interval files. In my monthly files, partner_name is the ID field.

	partner_name
1	Partner1
2	Partner2
3	Partner3
4	Partner4
5	Partner5
6	Partner6
7	Partner7
8	Partner8

Figure 5. ID Field

Because data for your reporting might come from different sources or might have data quality issues, it might be necessary to perform data cleansing and reconciliation to ensure that the identification fields will match when the time interval files are merged together. Otherwise, your report could be mismatched and include invalid ID values.

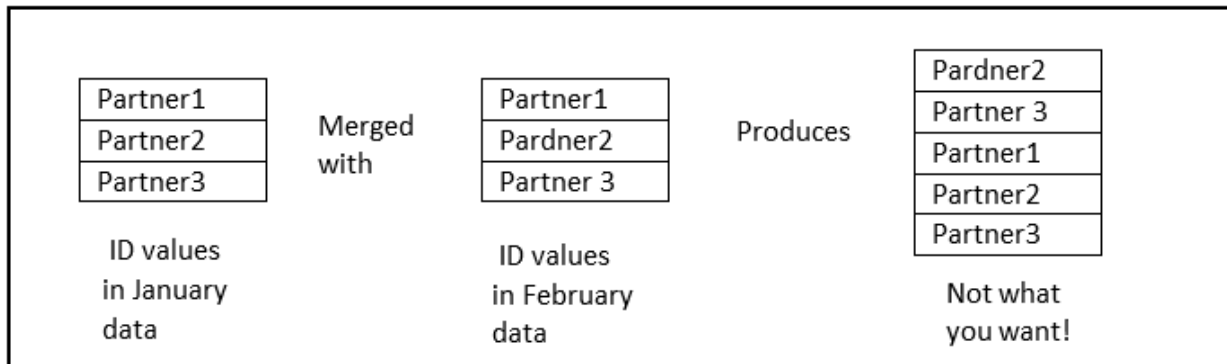


Figure 6. Mismatched Values

Use fuzzy matching or other programming techniques in SAS or use SAS® Data Management software to cleanse and reconcile your data as necessary.

Step 2. Summarize the Data into One Status Field

The status field might be an assessment or summation of more than one field. For example, a year's worth of test scores might determine a "pass" or "fail" status. In my data, status is determined by reviewing whether credentials have been earned in any of six categories. My incoming January detail data looks like Figure 7.








	 partner_name	 NumSales	 NumTechSales	 NumProg	 NumInst	 NumImp	 NumAdmin
1	Partner1	1	1	0	0	0	0
2	Partner2	1	0	0	0	0	0
3	Partner3	0	0	0	0	0	0
4	Partner4	0	0	1	0	0	0
5	Partner5	0	0	0	0	0	0
6	Partner6	1	1	1	0	0	1
7	Partner7	0	0	0	1	0	0
8	Partner8	0	0	0	0	0	0

Figure 7. January Detail Data

A SAS program processes the values of NumSales, NumTechSales, and the other numeric variables to determine the status. The program produces a boiled-down file for January, which contains just the partner_name (ID) field and the status field, as shown in Figure 8.



	 partner_name	 status
1	Partner1	Has Met Requirement
2	Partner2	Has Met Requirement
3	Partner3	Has Not Met Requirement
4	Partner4	Has Met Requirement
5	Partner5	Has Not Met Requirement
6	Partner6	Has Met Requirement
7	Partner7	Has Met Requirement
8	Partner8	Has Not Met Requirement

Figure 8. Boiled-down Data

Step 3. Create the Base File

We want to automate the process of merging each incoming time interval file with the earlier interval files so that only very few manual changes are needed each time the process takes place. Let's refer to the file with which we merge the new time interval file as the base file. Besides allowing us to standardize the program code, this approach will also enable us to keep backup copies of the previous base files, which is recommended for auditing and review purposes, and to provide a way to "roll back" the data in case of any sort of disruption.

I tend to err on the side of caution by saving copies of each time interval's detail data and summary file (as seen in Step 2), as well as saving the previous time interval base report and the current time interval base report. My files are not large, so saving these files doesn't place an undue burden on storage resources. Your situation might be different. For example, you might decide that keeping all those versions of the data is not necessary or even possible. At minimum, I would recommend keeping the previous month's base file. In the case of some sort of system or programming glitch, the detail information can be reconstructed from the saved file.

Let's assume that the boiled-down January data (seen in Figure 8) is the starting time interval and is the file that we'll consider as the base file. Its two-level name is sgf.partners_jan, where sgf is the assigned libref that refers to the SAS library in which the SAS data set is stored, and partners_jan is the name of the data set. I recommend that you use a standard naming format that can be repeated each time interval. In this example, SAS data sets for subsequent months' time intervals are named partners_feb, partners_march, and so on.

Because `partners_jan` is the first time interval file, let's make a copy of it and name it `sqf.base`. In addition, we will rename the status field with a name that refers to the time interval in the file. Here is one way to accomplish these tasks. This code is run only one time, in order to create the original base file.

```
proc datasets;
  copy in=sqf out=work;
  select partner_jan;
run;
  change partner_jan=base;
  modify base;
  rename status=january;
run;
  copy in=work out=sqf;
  select base;
quit;
```

Step 4. Merge the Base File with the New Time Interval File

When data for the next time interval is available, we'll merge it with the base file. Depending on how the merging is performed, the files might need to be sorted ahead of time to ensure that they have the same sort sequence. Be sure to change the name of the status field in the new file to reference the new time interval. Also, if you want to keep a copy of the previous base file that contains only the January data, be sure to make a copy of it before overwriting it.

```
proc datasets lib=sqf;
  modify partners feb;
  rename status=february;
quit;

proc sort data=sqf.base out=base;
  by partner_name;
run;

proc sort data=sqf.partner_feb out=feb;
  by partner_name;
run;

proc datasets lib=sqf;
  change base=base_jan;
quit;

data sqf.base;
  merge base feb;
  by partner_name;
run;
```

The new base file looks like Figure 9.

	partner_name	january	february
1	Partner1	Has Met Requirement	Has Met Requirement
2	Partner2	Has Met Requirement	Has Met Requirement
3	Partner3	Has Not Met Requirement	Has Met Requirement
4	Partner4	Has Met Requirement	Has Met Requirement
5	Partner5	Has Not Met Requirement	Has Not Met Requirement
6	Partner6	Has Met Requirement	Has Met Requirement
7	Partner7	Has Met Requirement	Has Met Requirement
8	Partner8	Has Not Met Requirement	Has Not Met Requirement
9	Partner9		Has Not Met Requirement

Figure 9. New Base File

Step 5. Revise the Program Using Macro Variables

You might have noticed that the program in Step 4 has many references to the time interval in the filenames. A simple revision will enable the program to run with fewer code changes from month to month.

```
%let old time=jan;
%let new time=feb;
%let new_status=february;

proc datasets lib=sgf;
  modify partner &new time;
  rename status=&new_status;
quit;

proc sort data=sgf.base out=base;
  by partner_name;
run;

proc sort data=sgf.partner_&new_time out=&new_time;
  by partner_name;
run;

proc datasets lib=sgf;
  change base=base_&old_time;
quit;

data sgf.base;
  merge base &new time;
  by partner_name;
run;
```

The next time the program is run, with the next new time interval file, the only changes to the program will be to assign new values to the macro variables.

```
%let old time=feb;
%let new time=mar;
%let new_status=march;
```

Step 6. Creating the Trafficlighting

Trafficlighting the report based on the merged data requires two actions: 1) creating a custom format to assign the color to the specific values, and 2) using ODS features to apply the format to the cells of the report.

The term trafficlighting as used here refers to the practice of color-coding parts of a report to draw the viewer's attention to certain values and increase the intuitive nature of the report. A traffic light that controls the flow of traffic on a busy street uses red, yellow and green lights to tell us whether to stop, move with caution, or go. Project management uses RAG (red, amber, green) or RYG (red, yellow, green) status to easily monitor and control the progress of business projects. Items displayed in red are usually defined as being behind their timeline or over budget, items in amber or yellow are within tolerance, and items in green are progressing according to plan. In reports, the colors red, yellow and green are often used to indicate thresholds or progress toward a certain conclusion.

I create a custom format called `status_fmt` to assign colors to current status. With just two possible values for status, I'm using just the two colors green and yellow.

```
proc format;
  value $status fmt 'Has Met Requirement'='lightgreen'
                  'Has Not Met Requirement'='yellow';
run;
```

Incorporating this custom format into the PRINT procedure code colors the report values appropriately.

The February report code is shown here.

```
proc print data=sgf.base label noobs;
  var partner name;
  var january / style (data) = [ background=$status_fmt.];
  var february / style (data) = [ background=$status_fmt.];
  label partner name='Partner Name';
  label january='January';
  label february='February';
run;
```

If you want, you can assign values to macro variables to reference the variables in the VAR and LABEL statements.

Here is the resulting report.

Partner Name	January	February
Partner1	Has Met Requirement	Has Met Requirement
Partner2	Has Met Requirement	Has Met Requirement
Partner3	Has Not Met Requirement	Has Met Requirement
Partner4	Has Met Requirement	Has Met Requirement
Partner5	Has Not Met Requirement	Has Not Met Requirement
Partner6	Has Met Requirement	Has Met Requirement
Partner7	Has Met Requirement	Has Met Requirement
Partner8	Has Not Met Requirement	Has Not Met Requirement
Partner9		Has Not Met Requirement

Figure 10. February Report

As additional data is added each month, additional VARIABLE and LABEL statements are added to the code to account for the growing number of time intervals. To fully automate the code, you can write a macro program to generate the new VARIABLE and LABEL statements at each time interval so that the statements don't have to be added manually.

REPORT #2: TRACKING COMPETENCY BADGE ATTAINMENT FROM MONTH TO MONTH

Partners earn competency badges when their consultants have acquired certification in six skill sets for one of the core technologies of SAS: analytics, data management, or visualization. The trafficlighting used in my Badge Attainment Report does not use the typical RYG colors. Instead, I selected three colors simply to differentiate between the three possible types of badges. I'm using a few rows of sample data for illustration.

THE REPORT

Timeline of Partner Competency Badges					
Obs	Partner Name	Badges Earned 1Feb17	Badges Earned 1Mar17	Badges Earned 1Apr17	Badges Earned 1May17
1	PartnerA	Analytics	Analytics	Analytics	Analytics
2	PartnerB	Data Management	Data Management	Data Management	Data Management
3	PartnerB		Visualization	Visualization	Visualization
4	PartnerC	Analytics	Analytics	Analytics	Analytics
5	PartnerC	Visualization	Visualization	Visualization	Visualization
6	PartnerD	Analytics	Analytics	Analytics	Analytics
7	PartnerD	Data Management	Data Management	Data Management	Data Management
8	PartnerD	Visualization	Visualization	Visualization	Visualization
9	PartnerE			Visualization	Visualization
10	PartnerF				Data Management

Figure 11. Partners Earning Competency Badges

The difference between this report and the previous one is that, instead of the trafficlighting being used to change columns (representing time intervals) on the same row whenever a partner's credential status

changed, a new row is added when a partner attains a new badge, with the row highlighted in the color associated with the new badge.

What does the report tell us? PartnerA has held the Analytics badge for the entire time period in the report. PartnerB, who held the Data Management badge at the first time interval, earned their Visualization badge in the second time interval. PartnerC held both the Analytics and Visualization badges at the first time interval. PartnerD has held all three badges for the entire time period represented in this report - there are three rows in the report for PartnerD, with each row representing a different badge. PartnerE earned their Visualization badge in the third time interval, and PartnerF earned their Data Management badge in the fourth time interval.

HOW TO CREATE THE REPORT

The basics for creating this report are similar to the previous report:

- Ensure that there is a unique identifier in each time interval file. The identifier will be used to merge this time interval's file with the base file.
- Write the SAS program to summarize status detail data into one summary status field.
- Use PROC FORMAT to create a custom format associating text values with colors.
- Reference the custom format in PROC PRINT to traffic-light the report.

Here is the file from February, showing partners who have earned badges.

	partner_name	Badges_1Feb17
1	PartnerA	Analytics
2	PartnerB	Data Management
3	PartnerC	Analytics
4	PartnerC	Visualization
5	PartnerD	Analytics
6	PartnerD	Data Management
7	PartnerD	Visualization

Figure 12. February Badge Data

Here is the file from March, showing partners who have earned badges.

	partner_name	badges_1Mar17
1	PartnerA	Analytics
2	PartnerB	Data Management
3	PartnerB	Visualization
4	PartnerC	Analytics
5	PartnerC	Visualization
6	PartnerD	Analytics
7	PartnerD	Data Management
8	PartnerD	Visualization

Figure 13. March Badge Data

Let's merge the data using the same logic we used in the previous example.

```
proc sort data=sqf.feb_badge out=feb_sorted;
  by partner_name;
run;
```

```
proc sort data=sqf.mar_badge out=mar_sorted;
  by partner_name;
run;
```

```
data both;
```

```

merge feb sorted mar_sorted;
by partner_name;
run;

```

This code does not produce the results we expect.

	partner_name	Badges_1Feb17	badges_1Mar17
1	PartnerA	Analytics	Analytics
2	PartnerB	Data Management	Data Management
3	PartnerB	Data Management	Visualization
4	PartnerC	Analytics	Analytics
5	PartnerC	Visualization	Visualization
6	PartnerD	Analytics	Analytics
7	PartnerD	Data Management	Data Management
8	PartnerD	Visualization	Visualization

Figure 14. Unexpected Results

Note the value for PartnerB at the 1Feb17 time period. We want that cell to be blank to indicate that the partner had not yet earned the Visualization badge, but the merge operation causes the value Data Management to be propagated to the new row.

The solution is to merge the data using not only the partner_name field, but also on the badge field. However, the problem is that the DATA step merge operation requires that the BY variables have the same name in the tables being merged. Here the variables are named badges_1Feb17 and badges_1Mar17, which are not the same.

There are a number of ways to address this problem. One approach is to create a new variable in each time interval file that contains the value of the badge earned. This variable, called matchfield in this example, uses the same name in all the files. Note that the SQL procedure is an alternate approach. Using the SQL procedure does not require that the BY variables have the same name or that the data be sorted before joining.

Here is the code to accomplish this.

```

data add field 1feb17;
  set sqf.feb badge;
  matchfield=badges_1feb17;
run;

data add field 1mar17;
  set sqf.mar badge;
  matchfield=badges_1mar17;
run;

proc sort data=add field 1feb17 out=sorted_feb;
  by partner_name matchfield;
run;

proc sort data=add field 1mar17 out=sorted_mar;
  by partner_name matchfield;
run;

```

```

data sqf.mar report (drop=matchfield);
  merge sorted feb sorted mar;
  by partner_name matchfield;
run;

```

The output is now correct.

	partner_name	Badges_1Feb17	badges_1Mar17
1	PartnerA	Analytics	Analytics
2	PartnerB	Data Management	Data Management
3	PartnerB		Visualization
4	PartnerC	Analytics	Analytics
5	PartnerC	Visualization	Visualization
6	PartnerD	Analytics	Analytics
7	PartnerD	Data Management	Data Management
8	PartnerD	Visualization	Visualization

Figure 15. Correct Output for March Badges Data

Now, using the same coding logic as above, let's add data for April. We add matchfield=badges_1Mar17 from March's data to the base data, add matchfield=badges_1Apr17 to the April data, and merge the files. Here is the result.

	partner_name	Badges_1Feb17	badges_1Mar17	Badges_1Apr17
1	PartnerA	Analytics	Analytics	Analytics
2	PartnerB	Data Management	Data Management	Data Management
3	PartnerB		Visualization	Visualization
4	PartnerC	Analytics	Analytics	Analytics
5	PartnerC	Visualization	Visualization	Visualization
6	PartnerD	Analytics	Analytics	Analytics
7	PartnerD	Data Management	Data Management	Data Management
8	PartnerD	Visualization	Visualization	Visualization
9	PartnerE			Visualization

Figure 16. April Badges Data

The same technique is repeated for all subsequent time intervals. Here is May's data.

	partner_name	Badges_1Feb17	badges_1Mar17	Badges_1Apr17	badges_1may17
1	PartnerA	Analytics	Analytics	Analytics	Analytics
2	PartnerB	Data Management	Data Management	Data Management	Data Management
3	PartnerB		Visualization	Visualization	Visualization
4	PartnerC	Analytics	Analytics	Analytics	Analytics
5	PartnerC	Visualization	Visualization	Visualization	Visualization
6	PartnerD	Analytics	Analytics	Analytics	Analytics
7	PartnerD	Data Management	Data Management	Data Management	Data Management
8	PartnerD	Visualization	Visualization	Visualization	Visualization
9	PartnerE			Visualization	Visualization
10	PartnerF				Data Management

Figure 17. May's Badges Data

To add the trafficlighting, we use PROC FORMAT to associate the colors with the possible badge values - analytics, data management, and visualization. To produce the trafficlighting report, we reference the custom format in PROC PRINT.

```
proc format;
  value $badge fmt 'Analytics'='pink'
                    'Data Management'='very light blue'
                    'Visualization'='light green';

run;

Title 'Timeline of Partner Competency Badges';
Footnote;

proc print data= sqf.may_report;
  var partner name;
  var badges 1Feb17 / style (data) = [ background=$badge fmt. ] ;
  var badges 1Mar17 / style (data) = [ background=$badge fmt. ] ;
  var badges 1Apr17 / style (data) = [ background=$badge fmt. ] ;
  var badges_1May17 / style (data) = [ background=$badge_fmt. ] ;

run;
```

The traffic lighted report is shown in Figure 11 above.

CONCLUSION

Trafficlighting is frequently used to highlight ranges of values in reports or to draw our attention to certain features of the report. The use of color catches our attention in a way that standard black and white reports can't. In this paper we've seen that the addition of a time element to trafficlighting provides an easy way to track categorical values. Such reports enable us to see changes and anomalies that we might not easily detect otherwise, allowing us to take action as necessary.

REFERENCES

Numerous papers have been written for presentation at SAS User Group conferences on the topic of trafficlighting, and much documentation is available. To name just a few papers:

Zender, Cynthia L. and Allison M. Booth, 2013. "Turn Your Plain Report into a Painted Report Using ODS Styles." *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings13/366-2013.pdf>.

Karp, Andrew H. 2011. "Traffic-Lighting Your Reports the Easy Way with PROC REPORT and ODS." *Proceedings of the SAS Global Forum 2011 Conference*. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings11/290-2011.pdf>.

Hadden, Louise, 2006. "STOP! WAIT: GO!: See What Traffic-Lighting Can Do For You!" *Proceedings of the SAS Global Forum 2006 Conference*. Cary, NC: SAS Institute Inc Available <http://www2.sas.com/proceedings/sugj31/142-31.pdf>.

ACKNOWLEDGMENTS

Thanks go to Robert Allison (Principal Test Engineer, SAS Institute), Rick Wicklin (Distinguished Research Statistician Developer, SAS Institute), Dan Heath (Principal Software Developer, SAS Institute), Warren Kuhfeld (Distinguished Research Statistician Developer, SAS Institute), and Cynthia Zender (Principal Curriculum Consultant, SAS Institute) for their helpful suggestions.

CONTACT INFORMATION

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

Lisa Horwitz
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Lisa.Horwitz@sas.com
<http://www.sas.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.