Paper 202-30

Using Procedure-Based ODS Data Components in Statistical Reporting

Vincent J. Faber, DM-STAT, Inc., Malden, MA

ABSTRACT

The objective of this paper is to illustrate the use of the ODS (Output Delivery System) Data Component in *SAS® system* procedures. It is designed to be an introduction for the elementary or intermediate programmer. I first provide a description of procedure-based ODS data components and discuss their applications. I then provide step-by-step instructions on how to extrapolate results from procedures and turn them into data sets that can then be used for statistical reporting. For example, we can extract both the mean and the p-value from the TTEST procedure using ODS data components and produce a table to report means and a test for significance.

Using a deidentified subset of a large national study of pregnant women, I will illustrate the use of procedure-based ODS with five commonly used procedures: UNIVARIATE, TTEST, FREQ, GLM, and LOGISTIC. Then I will conclude with a more advanced application where propensity scores are developed and used to adjust for potential selection bias in a logistic regression analysis.

INTRODUCTION

When working with data sets that contain a large quantity of variables, statistical reporting can be cumbersome. Some procedures, such as the FREQ Procedure, are simple to report. The output will rarely exceed one or two pages and the components are easy to read and understand. Other procedures, such as GLM and LOGISTIC, can be more difficult to report. The output can exceed multiple pages depending on the specified options and may be more difficult for an investigator to locate, read, and understand.

ODS is a powerful tool that can enhance the efficiency of statistical reporting and meet the needs of the investigator. Using ODS in procedure statements has unlimited benefits for biostatisticians, epidemiologists, and other statistical programmers. In particular, it is most popular for its ability to include or exclude *select* components of procedure output and turn them into more condensed, understandable, and reportable tables. ODS also minimizes transcription errors. By using ODS, we can make changes and easily reproduce data sets and tables.

For many years, SAS has worked diligently to provide optimal reporting for an abundance of statistical measures. However, it is becoming a more common practice for programmers to report *select* pieces of procedure output. Many times, investigators are only concerned with certain statistical measures. Investigators need to produce tables that are suitable to report in abstracts, manuscripts, and other industry papers. Programmers can meet these needs, and others, through the use of ODS.

ODS will break procedures down into sections. Each section is a set of raw data defined by *the SAS System*. These data sets store the information that is displayed in the procedure output. They can be turned into output data sets that can be used as tables for statistical reporting.

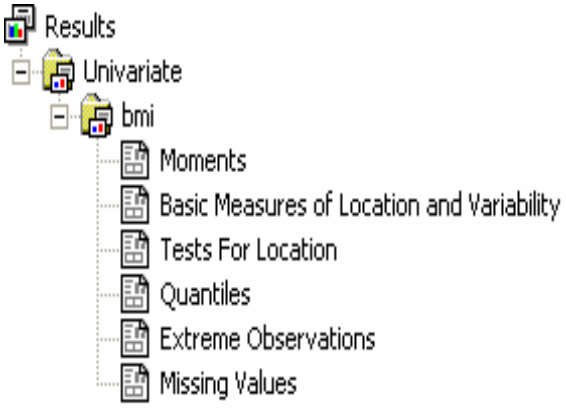
In Section 1, we will use the RESULTS window to learn about procedure output. This will help us use ODS to trace procedures in Section 2. In Section 3, we will then identify and create data sets from the procedure components. In Section 4, we will use data manipulation techniques to manage the data sets. Before moving on to some applied examples, we will summarize the process of using ODS into 5 easy steps in Section 5. In Sections 6, 7, and 8, we will use multiple ODS statements with the FREQ, GLM, and LOGISTIC procedures, respectively. Finally, in Section 9, we will use ODS in an advanced statistical procedure known as propensity scoring.

1. USING THE RESULTS WINDOW

Prior to using Procedure-Based ODS, we need to understand the output generated by the procedure. The RESULTS window will help you identify the components of the procedure to include, exclude, and manipulate.

After a procedure is run, the RESULTS window will display a list of the output sections produced by the procedure. Each section is called an output object and can be made into a data set. The procedure's components will become the observations and variables in the data set.

FIGURE 1



RESULTS window displaying the UNIVARIATE output objects.

Figure 1 is a RESULTS window listing the six output objects for the UNIVARIATE procedure on the variable "Body Mass Index" ("BMI"): Moments, Basic Measures of Location and Variability, Tests for Location, Quantiles, Extreme Observations, and Missing Values. The listing in the RESULTS window will match the sections of output generated by the procedure. The RESULTS window can be hidden to increase the working space on your computer monitor; however, when using ODS for the first time, it might be helpful to leave the RESULTS window open. The best way to visualize the output objects generated by a procedure is to view the output in the RESULTS window.

2. IDENTIFYING OUTPUT OBJECTS

By using the RESULTS window in the BMI example, we see that the UNIVARIATE procedure has six output objects. In practice, it may not be necessary to report the six output objects. Perhaps the investigator only needs to see the 10^{th} and the 90^{th} percentiles. Conceivably, the investigator may only need to see the most extreme observations. In both situations, procedure-based ODS can be used to select pieces of output to report. Using ODS in the procedure statement we can select the QUANTILES output object, turn it into a data set, and manipulate it to report only the desired percentiles.

ODS identifies the output objects in a way similar to the RESULTS window through a process called tracing. The general syntax for tracing is listed below:

```
ods trace on;
procedure syntax;
ods trace off;
```

We begin by instructing ODS to start tracing the procedure. Then we apply the appropriate procedure syntax and stop the tracing. The following syntax is used to trace the UNIVARIATE procedure for the BMI example:

```
ods trace on;
proc univariate data=one;
var BMI;
run;
ods trace off;
```

As a result of the tracing, information about each of the output objects will be generated in the log. *The SAS System* will still generate a listing of the procedure's output objects in the RESULTS window. Information that is more pertinent and that is not captured in the RESULTS window will be found in the log.

Figure 2 shows the results of tracing the UNIVARIATE procedure from a section of the log. The tracing results provide the Name, Label, Template and Path for the output objects. For now, we are only concerned with the name of the output object.

FIGURE 2

```
Output Added:
-------------
Name:        BasicMeasures
Label:       Basic Measures of Location and Variability
Template:    base.univariate.Measures
Path:        Univariate.bmi.BasicMeasures
-------------

Output Added:
-------------
Name:        TestsForLocation
Label:       Tests For Location
Template:    base.univariate.Location
Path:        Univariate.bmi.TestsForLocation
-------------
```

Tracing results from a section of the log for the UNIVARIATE procedure.

The name of the output object may differ from what is listed in the RESULTS window. This is because the RESULTS window is displaying the label defined for the output object and not the actual name of the object. As a novice programmer, you can use the LISTING option preceded by a forward slash with the tracing syntax to move the tracing result from the log to the LST. This may further enhance your learning of the procedure's components and its output objects.

3. INVOKING ODS TO CREATE DATA SETS

Suppose an investigator would like to test the equality of the mean birth weight of babies born to women who developed gestational diabetes as compared to those who did not. While all of the statistical information reported by the TTEST procedure is pertinent, she would like only to report the means of the two groups and the p-value testing the equality of means.

If you are unfamiliar with the TTEST procedure, you should begin by running the procedure and looking at the listing in both the RESULTS window and the LST. With a clear understanding of the output generated by the procedure, trace the procedure to learn the names of the output objects. We will need to know the names of the output objects so that we can instruct ODS to output them into a data set. The following code traces the TTEST procedure and identifies the names of the output. Since we are using the LISTING option, it will place the tracing information directly into the LST.

```
ods trace on / listing;
proc ttest data=one;
class g_diabt2;
var baby_wt;
run;
ods trace off;
```

The names of the output objects are now in the LST. STATISTICS and TTESTS are the names of the two output objects that store the means and the p-values, respectively. To construct data sets from the output objects, use an ODS output statement.

The general format of the ODS output statement is as follows:

```
proc <procedure>;
procedure syntax;
ods output <output object> = <data set name>;
run;
```

First we apply the procedure syntax. Then we add the ODS output statement. This statement extracts the specified output object and creates a data set.

```
proc ttest data = one;
class g_diabt2;
var baby_wt;
ods output STATISTICS = MEANS;
ods output TTESTS = PVAL;
run;
```

To turn the STATISTICS and TTESTS output objects into data sets, we added the ODS output statements to the syntax for the TTEST procedure. When we check the log we will see that the data set MEANS has 3 observations and 14 variables, and PVAL has 2 observations and 6 variables.

An output object is a set of raw data as defined by *the SAS System*. This data set stores the information that is displayed in procedure output. Therefore, the structure of the data set created by ODS will be similar, if not identical, to the structure of the output.

FIGURE 3

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
baby_wt	Pooled	Equal	34E3	-1.77	0.0774
baby_wt	Satterthwaite	Unequal	1248	-1.43	0.1525

Test for Equality of Means from the TTEST output.

FIGURE 4

Variable	Method	Variances	t Value	DF	Pr > |t|
baby_wt	Pooled	Equal	-1.77	34E3	0.0774
baby_wt	Satterthwaite	Unequal	-1.43	1248	0.1525

PVAL data set created using the ODS output statement.

For example, in Figures 3 and 4, you can compare the output from the LST to a data set created from the output object. The PVAL data set holds the raw data components Variable, Method, Variances, DF (Degrees of Freedom), tValue, and Pr > |t|. These components are held in the data set as observations and variables that can be included, excluded, and manipulated following standard DATA step rules.

4. MANAGING ODS OUTPUT OBJECTS

As data sets, ODS output objects can be manipulated easily. Through the rest of this paper, examples of including, excluding, and manipulating data sets created from ODS output objects will be presented.

In the BMI example, we used the TTEST procedure to test for a significant difference in mean birth weights. The investigator asked for only the means of the two groups and the significance of the test. ODS was used to create the two data sets, MEANS and PVAL. To satisfy the investigator's request and present one combined data set, we merge the information from the two data sets into one. We exclude unnecessary variables and merge the data sets through a common variable.

In simple examples, constructing one table from many ODS output objects can be done through one or two data steps. More advanced techniques, such as the propensity scoring example that concludes this paper, will require multiple procedures and data steps. One of the most efficient ways to manage data is to use data manipulation techniques directly in the ODS output statements. The ODS output statement can handle KEEP, WHERE, DROP, and RENAME statements. We will modify the existing code from the birth weight example as follows:

```
proc ttest data=one;
class g_diabt2;
var baby_wt;
ods output STATISTICS = MEANS (keep=variable mean class where=(Class in "No","Yes")));
ods output TTESTS = PVAL (where = (Variances = "Equal")keep = variable variances
probt);
run;

data meancomp;
merge MEANS PVAL (drop=variances);
by variable;
run;
```

We added KEEP and WHERE statements to include and exclude variables and observations. Since the investigator was not interested in the mean difference, we used a WHERE statement to keep only the means for each level of the grouping variable and not the difference. We can also use a KEEP statement to include only the variables VARIABLE, MEAN, and CLASS. VARIABLE is the merging identifier between the two tables. We can use the same logic to create the PVAL data set. In this example, the test for the equality of variances was not significant. As a result, we should only report the Equal Variances p-value.

FIGURE 5

Testing the Equality of Means
between Women with and without Gestational Diabetes

Variable	G_diabt2	Mean	P-Value
baby_wt	No	3349.2	0.0774
	Yes	3377	

Data set MEANCOMP created by merging ODS data sets MEANS and PVAL from the TTEST

The data step finalizes the work. We merge the two data sets by the variable named VARIABLE. The merging variable may change depending on the procedure and your data. Sometimes, you may find it necessary to create a dummy variable to use as the merging variable.

The variable VARIANCES was solely used for data manipulation. It is no longer needed and is dropped from the data set. The final table in Figure 5 meets the request of the investigator.

5. SUMMARIZING THE STEPS

We can summarize the process of using procedure-based ODS in five simple steps.

Step 1: Run the procedure leaving the results window open

This is useful for elementary or novice programmers. It is an important, but not critical, first step in being proficient in procedure-based ODS.

Step 2: Identify the names of output objects using ODS

Trace the procedure with or without the LISTING option. This may be unnecessary as you become familiar with ODS.

Step 3: Identify the output objects needed

If you used the LISTING option with the tracing syntax, the results will appear in the LST. Without the LISTING option, results will appear in the LOG.

Step 4: Add the ODS output statement and apply data manipulation techniques

Apply the ODS output statement and be as creative as possible in applying data manipulation techniques. Use the names identified during tracing to create the data sets.

Step 5: Use the data sets as needed

Combine and manipulate the data sets and components as needed.

6. FREQ PROCEDURE EXAMPLE

A team of investigators would like to see if baby girls are more or less likely to have birth weights greater than 2500 grams as compared to baby boys. They would like to report the odds and the significance of women delivering baby girls that weigh more than 2500 grams. Additionally, they would like to report the 95% confidence interval for the odds ratio.

Since the two variables in question are both binary (Yes/No; Boy/Girl) we will use the FREQ procedure with the CHI SQUARE option to run the Chi-Square test. Additionally, we can use the MEASURES option to produce the odds ratio and 95% confidence interval. By tracing the procedure, we learn that all of the raw data we need is stored in 3 ODS output objects. We can use output object CROSSTABFREQS for the frequencies, CHISQ for the Chi-Square p-value, and RELATIVERISKS for the Odds Ratio and 95% Confidence Interval.

```
proc freq data=one;
tables bbwt25_2*genderb / chisq measures nopercent nocum;
ods output CROSSTABFREQS = FREQS (keep=table bbwt25_2 genderb frequency);
ods output CHISQ = PVAL (keep=table statistic prob where=(statistic = "Chi-Square"));
ods output RELATIVERISKS = ODDS(where=(studytype="Case-Control (Odds Ratio)"));
run;

data all;
merge FREQS PVAL (drop = statistic) ODDS (drop = studytype);
by table;
run;
```

We create the data set FREQS in the first ODS output statement. Next we create the data sets PVAL and ODDS. We again use the WHERE and KEEP statements to exclude unnecessary rows and variables.

FIGURE 6

Odds of Baby Girls delivering at More than 2500 Grams

Table	Odds Ratio	Lower C.L.	Upper C.L	P-Value
bbwt25_2_by_genderb	0.7818	0.7100	0.8609	<.0001

Data set ALL, merging each of the data sets FREQS, PVAL, and ODDS from the FREQ procedure.

We can maneuver the three data sets into one with a concluding data step. In this step we merge by the common variable TABLE. The final table for the investigator is displayed in Figure 6.

7. GLM PROCEDURE EXAMPLE

An investigator would like to know if maternal age is a determinant in the number of miscarriages a woman has. He wants to know if women in different age categories have more or less miscarriages as compared to other age categories. He has specifically requested a table showing the p-value associated with the test of significance, the significantly different age categories, and the difference between their means.

```
proc glm data=one;
class matama4;
model miscar = matama4;
means matama4;
means matama4 / tukey;
ods output OVERALLANOVA=PVAL (where=(source="Model")keep=source probf
              CLDIFFS = DIFF (where=(significance=1) keep = significance comparison
                              effect difference); run; quit;
data compare (drop = significance obs effect source);
merge DIFF PVAL; obs = _n_;
if obs in (2,3,4,5,7) then delete;
run;
```

Since the ANOVA model is simple (one independent variable with no adjustors), we can use the p-value from either the ANOVA Table, or from the Type III Sum of Squares Table. Suppose we want to also perform pairwise tests using the Tukey procedure. By tracing the GLM procedure, we learn OVERALLANOVA is the name of the output object that stores the p-value from the ANOVA table. CLDIFFS is the object that stores the Tukey multiple comparisons results.

FIGURE 7

Multiple Comparison Test
Mean Difference in Number of Miscarriages

Comparison	Difference Between Means	P-Value
=45 - 35=age<40	0.455176	<.0001
35=age<40 - 40=age<45	-0.328122	
15<age<35 - =45	-0.705415	
15<age<35 - 40=age<45	-0.578360	
15<age<35 - 35=age<40	-0.250239	

Data set COMPARE, merging the ODS data sets DIFF and PVAL from the GLM procedure.

Instead of using a separate ODS output statement for each output object, we use one output statement. We can still use KEEP and WHERE statements to manipulate the data. In the data step, we merge the two data sets, PVAL and DIFF. In this example, there is no common variable to merge the two data sets. We can merge without one or we can create a dummy variable in both data sets.

Since the Tukey test is a two element permutation of four categories of maternal age, the manipulation technique "obs =_n_" was used to remove duplicate comparisons. The final data set is displayed in Figure 7.

8. LOGISTIC REGRESSION EXAMPLE

A theory among some of the investigators in this study is that women who have a history of previous pre-term deliveries (gestational age at birth < 37 weeks) may be at higher risk of delivering pre-term in their current pregnancy. The investigators of the study would like to test this theory by running a logistic regression. They would like to know the odds ratio, 95% confidence interval, and the significance of the test.

```
proc logistic data=one descending;
model prebirt2 = preterm2;
ods output PARAMETERESTIMATES = PVAL (where=(variable="preterm2")
                                      keep=variable probchisq)
           ODDSRATIOS = ODDS (rename=(effect=Variable));
run;

data oddsrat (drop=variable);
merge PVAL ODDS;
by Variable;
run;
```

Although the LOGISTIC procedure generates an abundance of useful statistics, the most commonly requested statistics are contained in the output object ODDSRATIOS. This table stores the odds ratio and the 95% confidence interval for the odds ratio. The p-value to measure the significance of the odds ratio is contained in the output object PARAMETERESTIMATES. You can use a WHERE statement to exclude the information about the regression model's intercept.

FIGURE 8

Probability of Delivering Preterm
Given a History of Previous Pre-term Birth

Odds Ratio Estimate	Lower 95% Confidence Limit for Odds Ratio	Upper 95% Confidence Limit for Odds Ratio	P-Value
4.086	3.657	4.564	<.0001

Data set ODDSRAT, merging the ODS data sets ODDS and PVAL from the LOGISTIC procedure.

The PVAL and ODDS data sets have one common merging value among them. However, their variables names are different. To control this, we use a RENAME statement in one of the data sets and then merge by that variable. The final data set is displayed in Figure 8.

9. PROPENSITY SCORING

Background

In observational studies where it is of interest to compare groups (e.g. abnormal versus normal), the groups may have differences in a number of covariates.

A propensity score is the conditional probability of a subject being in one group, given their covariates. Mathematically, it is defined as $Pr(Z = i \mid X = x)$ where X is the main grouping variable and Z is/are the covariate(s).

For more information on Propensity Scoring, please see the reference article by Ralph B. D'Agostino, Jr.

We will first show the distribution of the covariates on the grouping variable before propensity scoring. Then we will model the effect of the covariates on the grouping variable. This model will determine the probability of subjects being in one of the groups, given their covariates. The propensities will then be sorted and grouped into quintiles. Next, we will show the distribution of the covariates on the grouping variable, adjusting for the quintiles of the propensity scores. Finally, we will model the effect of the grouping variable on the outcome, adjusting for the quintiles of the propensity score.

The propensity scoring procedure can be summarized as follows:

1. TTEST procedure or the FREQ procedure, with the CHI-SQUARE option, to show the distribution of the covariates on the grouping variable before propensity scoring.
2. LOGISTIC procedure to model the effect of covariates on the grouping variable and to generate the propensities.
3. RANK procedure to group the observations based on their predicted values.
4. GLM procedure to show the distribution of the covariates on the grouping variable after propensity scoring.
5. LOGISTIC procedure to show the odds of developing the outcome, adjusting for the propensity scores.

ODS can be used to summarize all of these results.

Applied Example

An investigator believes that women carrying fetuses with chromosomal or structural abnormalities may be at risk for pre-term birth. There are many differences between women carrying fetuses with abnormalities as compared to those carrying normal fetuses. To assess the effect of abnormal status on pre-term birth, we need to adjust for these important differences.

Using Procedure-Based ODS Data Components

To first show the differences between the two groups on the covariates, we will use the TTEST and FREQ procedures. We will report the mean or proportion of each covariate in relation to the grouping variable. Next, we will use the LOGISTIC and RANK procedures to generate the propensity for carrying a normal or abnormal fetus. We will then use the GLM procedure to show the differences between the two groups after adjusting for the propensities. Using procedure-based ODS data components, we will combine output objects from each of the procedures to produce the tables in Figures 9 and 10. To conclude the request, we again use procedure-based ODS with the LOGISTIC procedure to present the adjusted odds of delivering pre-term. The final table is seen in Figure 11.

FIGURE 9

Difference Between Groups Prior to Propensity Scoring

Variable	Normal	Abnormal	P-Value
abort	0.24	0.25	0.8831
art2	4.93	5.08	0.7053
bmi	24.91	25.61	<.0001
cocaine2	0.08	0.16	0.1473
diabete2	0.87	2.11	<.0001
educ	14.35	14.25	0.0383
folic2	47.38	47.2	0.8461
fulterm	0.82	0.74	<.0001
g_diabt2	3.23	5.83	<.0001
gravida	2.49	2.55	0.0727
mat_age	30.04	30.28	0.0268
miscar	0.37	0.44	<.0001
para	0.89	0.87	0.4344

ODS Table from the FREQ and TTEST procedure.

FIGURE 10

Differences Between Groups After Propensity Scoring

Variable	Normal	Abnormal	P-Value
abort	0.24	0.25	0.9876
art2	0.05	0.05	0.4578
bmi	24.98	24.99	0.8821
cocaine2	0	0	0.1629
diabete2	0.01	0.02	<.0001
educ	14.37	14.37	0.9962
folic2	0.48	0.48	0.8362
fulterm	0.81	0.79	0.3139
g_diabt2	0.03	0.05	<.0001
gravida	2.5	2.52	0.4053
mat_age	30.08	30.07	0.8925
miscar	0.37	0.38	0.4288
para	0.88	0.9	0.5014

ODS Table from the GLM procedure.

FIGURE 11

Odds of Delivering Preterm
Adjusting for Propensity Scores

Variable	Odds Ratio	Lower C.L.	Upper C.L.	P-Value
abnormal	5.551	5.016	6.143	<.0001

Final table combining ODS data sets PVAL and ODDS from the LOGISTIC procedure.

Syntax for Propensity Scoring

For brevity, some of the necessary data manipulation has been excluded from the code. Procedure-based ODS in the TTEST, FREQ, GLM, and LOGISTIC procedures are used here to reduce the abundance of output and focus on the most relevant covariates.

Compare the ODS output statements used in each of the previous examples to the procedures used in the propensity scoring example. You should notice that we are using the same output objects in different combinations.

The following is an example of the syntax involved in propensity scoring:

```
proc ttest data=include;
class abnormal;
var abort bmi educ fulterm gravida mat_age miscar para;
ods output TTESTS=TT (keep=variable variances tvalue probt);
ods output EQUALITY=EQ(keep=variable Fvalue Probf);
```

```
ods output STATISTICS=STAT (keep=variable class n mean); run;

proc freq data=one;
tables cocaine2*abnormal / chisq outpct out=TABLE (drop=percent pct_row count);
ods output CHISQ = PVAL;
run;

proc logistic data=one;
class race;
model abnormal = abort bmi educ fulterm gravida mat_age miscar para;
output out=pr pred = pr;
run;

proc rank data=pr groups = 5 out = rank;
ranks rnks;
var pr;
run;

data quint;
set rank;
quintile = rnks +1;
run;

proc glm data=quint;
class abnormal quintile;
model abort = abnormal quintile;
lsmeans abnormal;
ods output OVERALLANOVA = ANOVA (drop = DF SS MS Fvalue);
ods output LSMEANS = LSMEAN (drop = effect rename=(abortLSmean = LSMean));
run; quit;

proc logistic data=quint descending;
model prebirt2 = abnormal quintile;
ods output PARAMETERESTIMATES = PVAL (where=(variable="abnormal")
                                      keep=variable probchisq);
           ODDSRATIOS = ODDS (rename=(effect=variable)
                              where=(variable="abnormal"));
run;
```

CONCLUSION

Procedure-based ODS data components can be used to facilitate statistical reporting. Output that is too cumbersome can be distilled and turned into tables for easier reporting and understanding. When variables and models are changed, these tables can be easily modified, thereby reducing transcription error. ODS allows the analyst to selectively include or exclude components to meet the needs of investigators. As a result, the programmer can achieve more control over the efficiency and consistency of statistical reporting.

REFERENCES

D'Agostino, Jr., Ralph B., (1998), "Propensity Score Methods for Bias Reduction in the Comparison of a Treatment to a Non-Randomized Control Group," *Statistics in Medicine*, 17, 2265-2281.

Haworth, Lauren E., (2000), "Output Delivery System: The Basics", NC: SAS Institute Inc.

ACKNOWLEDGMENTS

I would like to offer my most sincere gratitude towards Drs. Kimberly Dukes and Lisa Sullivan, and John Vidaver for their academic, vocational, and personal support. Much appreciation is also afforded to my fellow colleagues, especially the Statistical Programmers Karen, Fay, Gina, Jamie, and Pam. Finally, I would like to thank Michelle Zabka for her unbounded assistance in my success.

CONTACT INFORMATION

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

Vincent J. Faber
DM-STAT, Inc.
One Salem Street, Suite 300
Malden, MA 02148
(781) 395-4523 ext. 150
Email1: Vincent.Faber@dmstat.com
Email2: vfaber@bu.edu
Web: www.dmstat.com