



ABSTRACT

Version 8 of the SAS System has provided statisticians with several new features and tools that not only increase the availability of statistical methods, but also decrease the level of output manipulation and therefore time, required for reporting. This paper will demonstrate some of the new summary and graphical features available in Base SAS procedures, new features in commonly used SAS/STAT procedures and introduces new procedures such as boxplot, mi, and mianalyze. By using ODS objects the statistician can save time and code extracting statistical results in preparation for reporting. These features will be demonstrated, showing how to extract just the information you wish to report into data sets or directly to an output file. Furthermore this paper will consider how to control output layout and style to represent ICH (International Conference on Harmonisation) guidelines.

INTRODUCTION

There are many new features in version 8 of SAS which both increase the availability of statistical methods and allow these results to be reported more efficiently; saving time and SAS code.

This paper is divided into three sections, initially describing the enhancements and new features in base SAS summary procedures, introducing and demonstrating features of the output delivery system and finally discussing new procedures and enhancements in SAS/STAT.

EXPLORATORY DATA ANALYSIS

Eany Meany Miny Mo

One of the most commonly performed tasks as a statistician or statistical programmer is the summarisation of data. In version 6.12 the choice of means, summary or univariate procedures depends on which statistics are to be presented, the class statement in proc means or summary meant less code because a proc sort was not required, however if the median is needed only proc univariate could be used.

In version 8 the median and percentiles are available in proc means and proc summary. With a class statement added to proc univariate the choice of which procedure to use for simple statistics could be at the user's discretion.

Proc Univariate: If A Picture is Worth a Thousand Words...

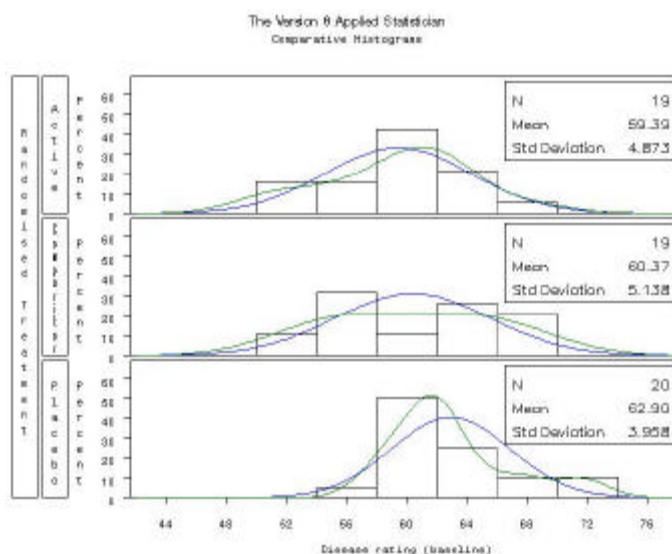
If a picture is worth a thousand words then how many statistics is a plot of the data worth?

Not only does proc univariate include numerous new statistics and considerably improved control over what it produces, but it can also now produce high resolution graphics.

Histogram, probability and quartile – quartile plots can be produced each having one or more theoretical distributions superimposed. Furthermore a table of summary statistics can be inset into this graphic.

The following code demonstrates how proc univariate can be used to generate a high resolution comparative histogram for groups of data. In V8 this graphic can also be easily exported to an ever increasing number of formats via the Output Delivery System (ODS).

```
proc univariate
  data=data.efficacy
  normal;
  class trtcd;
  var baseline;
  histogram baseline /
    normal(color=blue)
    kernel(color=green)
    nrows=3 ;
  inset n (5.0) mean
        (5.2) std (5.3) /
    pos=ne;
run;
```



The following lists some of the new statistics available from proc univariate:

Option	Description of statistic requested
CIBASIC	Generates confidence limits based on normally distributed data for the mean, standard deviation and variance.
CIPCTLDF	Generates confidence limits for quantiles using a distribution free method
CIPCTLNORMAL	Generates confidence limits for quantiles based on normally distributed data.
MU0=	Allow the user to change the mean location parameter from the default of zero.
NORMAL	Generates Shapiro-Wilk, Kolmogorov-Smirnov, Cramer-von Mises and Anderson-Darling goodness of fit tests.
ROBUSTSCALE	Generates robust estimates of scale
ROUND=	Specifies round-off units for the analysis variable, prior to analysing.
TRIMMED=	Generates trimmed means
WINSORIZED=	Generates winsorized means

More Than a Mean in a Summary

Although proc means or proc summary are frequently used only to generate descriptive statistics, there are many enhanced uses in version 8.

The PRELOADFMT option instructs proc means to preload all formats associated with class variables. This can be used in combination with the COMPLETETYPES option for displaying a category for each class variable, even when there are no source data to summarise.

For example, the source data contains no female subjects in the comparator treatment group. The following code produces a summary even for this class of data:

```
proc means data=data.efficacy n mean std completetypes maxdec=1;
  class trtcd sex / preloadfmt;
  var age;
run;
```

Produces the output below:

Analysis Variable : AGE Age at Baseline (Years)				
Randomised Treatment	Subject Gender	N	Mean	Std Dev
ff				
Active	Male	8	32.4	12.0
	Female	12	48.1	12.6
Comparator	Male	20	41.6	14.3
	Female	0		
Placebo	Male	9	38.9	9.2
	Female	11	42.1	14.4
ff				

The new CLASSDATA= option allows a data set to be specified which contains combinations of class variables to include in an analysis. This can also be used with the EXCLUSIVE option to summarise only the combinations of class levels in the CLASSDATA data set.

One further new feature is the MLF option on the CLASS statement. Used with proc format it allows overlapping ranges to create subgroup analyses. The following example categorises age into two main groups (below 30 years and over 30 years), but also categorises age into smaller groups. These overlapping ranges are then summarised in a single proc means step.

```
title2 "Overlapping Format Ranges with the MLF option";
proc format ;
  value agefmt (multilabel)
    15 - 30='Below 30 Years'
    31 - high='Over 30 Years'
    15 - 19='15 to 19'
    20 - 25='20 to 25'
    25 - 39='25 to 39'
    40 - 55='40 to 55'
    56 - high='56 +';
run;

proc means data=data.efficacy fw=8 maxdec=1 n mean std min max
nonobs;
  class age / mlf order=fmt;
  var baseline;
  format age agefmt.;
run;
```

The MLF option in the class statement above produces the overlapping ranges as follows:

The Version 8 Applied Statistician					
Overlapping Format Ranges with the MLF option					
The MEANS Procedure					
Analysis Variable : BASELINE Disease rating (baseline)					
Age at Baseline (Years)	N	Mean	Std Dev	Minimum	Maximum
15 to 19	3	64.6	9.6	53.8	72.1
20 to 25	7	61.6	4.2	54.0	67.7
25 to 39	14	59.8	4.8	50.3	68.8
40 to 55	24	60.9	4.4	50.7	71.2
56 +	10	61.1	5.0	53.0	67.5
Below 30 Years	16	61.7	5.5	53.8	72.1
Over 30 Years	42	60.6	4.6	50.3	71.2



AT ODS WITH YOUR OUTPUT?

In my opinion the enhancement which has the largest impact on the way I work in version 8 is the output delivery system (ODS).

Not only can reports be written directly to any number of formats such as Word documents, web pages, pdf files, the default output window or even concurrently to any of these destinations; but ODS also allows the user to capture any part of a procedure's output into a data set.

This paper could very easily devote much discussion to the subject of ODS, worthy of a paper of its own. Indeed there are many such papers, so the my aims here are only to introduce these features.

ODS Output

Any part of a procedure's output can be stored in a data set.

Each procedure's output is divided up into parts called objects. Each object within each procedure has a unique name by which it can be identified.

To determine the name of these objects, turn on ODS trace which prints information about each output object to the log file:

For example, proc means creates one output object, we can see this by expanding the proc means results in the results window. We can determine the name of this output object from the log file as follows:

```
ods trace on;
proc means data=data.efficacy;
  class trtcd;
  var age;
run;
ods trace off;
```

Partial log file:

```
Output Added:
-----
Name:      Summary
Label:     Summary statistics
Template:  base.summary
Path:     Means.Summary
-----
```

The name of the output object for proc means is therefore "summary". Storing this object in a data set is done via an ods output statement.

```
ods output object_name = data_set_name;
```

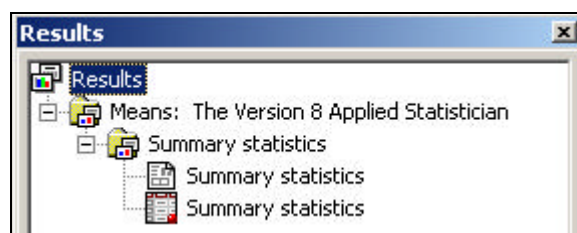
Applying this we can create a data set called "mydataset". The following partial log file is produced:

```
1  ods output summary=mydataset ;
2  proc means data=data.efficacy;
3    class trtcd;
4    var age;
5    run;

NOTE: The data set WORK.MYDATASET has 3 observations and 7 variables.
NOTE: There were 60 observations read from the data set DATA.EFFICACY.

6  ods output close;
```

If we expand the proc means output in the results window we can also see the data set created.



ODS Destinations

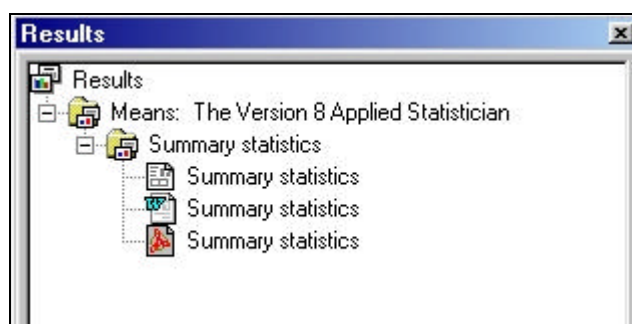
At the same time as storing output in data sets from procedures, we can also direct the output to one or several destinations. For example the summary statistics produced above could also be saved in RTF and PDF files. This is achieved by the use of multiple ODS statements:

```
ods rtf file="microsoft word.rtf";  
ods pdf file="adobe acrobat.pdf";
```

```
proc means data=data.efficacy fw=8 maxdec=1 n mean std min max  
nonobs;  
  class trtcd;  
  var age;  
run;
```

```
ods rtf close;  
ods pdf close;
```

The results window then displays one icon per output destination created:



Viewing the output of these files shows the default "style" used by SAS to display fonts, background shading and border drawing etc:

Output viewed in Adobe Acrobat:

The Version 8 Applied Statistician

Analysis Variable : AGE Age at Baseline (Years)					
Randomised Treatment	N	Mean	Std Dev	Minimum	Maximum
Active	20	41.8	14.4	20.0	62.0
Comparator	20	41.6	14.3	19.0	62.0
Placebo	20	40.7	12.1	19.0	62.0

Almost always the applied statistician will wish to define and apply a standard look and feel for the output. In the pharmaceutical industry, for example, the appearance of the output should adhere to ICH guidelines for output included in a regulatory submission.

Defining and Applying Style to ODS Output

An efficient way of controlling the global look and feel of your output is to define and store a style using proc template.

To adhere to guidelines recommended by ICH our output should have at least a one inch for all margins, and supposing we wished to remove cell shading and alter the font to resemble that of traditional SAS output, we can use proc template as follows:

```
libname project 'c:\project';

proc template ;
  define style ICH / store=project.styles;
    parent = styles.prnter;

    replace fonts /
      'TitleFont2' = ("Courier New",8pt)
      'TitleFont' = ("Courier New",8pt)
      'StrongFont' = ("Courier New",8pt)
      'EmphasisFont' = ("Courier New",8pt)
      'FixedEmphasisFont' = ("Courier New",8pt)
      'FixedStrongFont' = ("Courier New",8pt)
      'FixedHeadingFont' = ("Courier New",8pt)
      'BatchFixedFont' = ("Courier New",8pt)
      'FixedFont' = ("Courier New",8pt)
      'headingEmphasisFont' = ("Courier New",8pt)
      'headingFont' = ("Courier New",8pt)
      'docFont' = ("Courier New",8pt);
    replace Table from output /
      background = _undef_
      frame = HSIDES
      rules = GROUPS
      cellpadding = 5pt
      cellspacing = 0pt
      borderwidth = 1pt;

    replace color_list /
      "bg" = white
      "fg" = black
      "bgH" = white
```



```
"link" = blue ;

replace body /
  topmargin=1in
  bottommargin=1in
  leftmargin=1in
  rightmargin=1in;
end;
run;
```

The above style definition, called ICH, is stored in an "item store" which is located in the SAS library "project". This can be made available to our SAS session with the ODS PATH statement.

```
ods path project.styles(read) sashelp.tmplmst(read);
```

To use this style definition we add the style= option to the ods rtf statement.

```
ods rtf file="microsoft word.rtf" style=ICH;
```

```
proc means data=data.efficacy fw=8 maxdec=1 n mean std min max
nonobs;
  class trtcd;
  var age;
run;
```

```
ods rtf close;
```

Produces the output as viewed in Microsoft Word:

The Version 8 Applied Statistician					
Analysis Variable : AGE Age at Baseline (Years)					
Randomised Treatment	N	Mean	Std Dev	Minimum	Maximum
Active	20	41.8	14.4	20.0	62.0
Comparator	20	41.6	14.3	19.0	62.0
Placebo	20	40.7	12.1	19.0	62.0

STATISTICAL ANALYSES

An Old Way – A New Way

For many reports the default output of our analysis procedure is taken and manipulated before being reported. In version 6.12 considerable manipulation was often required to capture the results into data sets for reporting.

Consider a typical analysis of variance in using version 6.12. Proc glm would often be employed for model fitting and a test of effects using either the contrast or estimate statements as appropriate. The default output appears similar to the following:

Partial output:

The Version 8 Applied Statistician					
The GLM Procedure					
Class Level Information					
Class	Levels	Values			
TRTCD	3	1 2 3			
Number of observations		60			
NOTE: Due to missing values, only 56 observations can be used in this analysis.					
The Version 8 Applied Statistician					
The GLM Procedure					
Dependent Variable: ENDPOINT		Disease rating (end of study)			
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	1092.908036	546.454018	14.69	<.0001
Error	53	1971.087500	37.190330		
Corrected Total	55	3063.995536			
R-Square	Coeff Var	Root MSE	ENDPOINT Mean		
0.356694	10.92586	6.098388	55.81607		

It is very rare that reports are presented in the default format such as this. To achieve the desired layout results are captured into data sets and a reporting tool such as proc report is used. To actually capture many of the statistics considerable manipulation of results and even derivation of statistics is required, for example the calculation of confidence intervals about an inference.

The new way is of course ODS. Consider the output from proc glm displayed above. How much of the output is actually needed in your report? Perhaps all the output will be appended somewhere for documentation purposes and only key results used in another presentation.

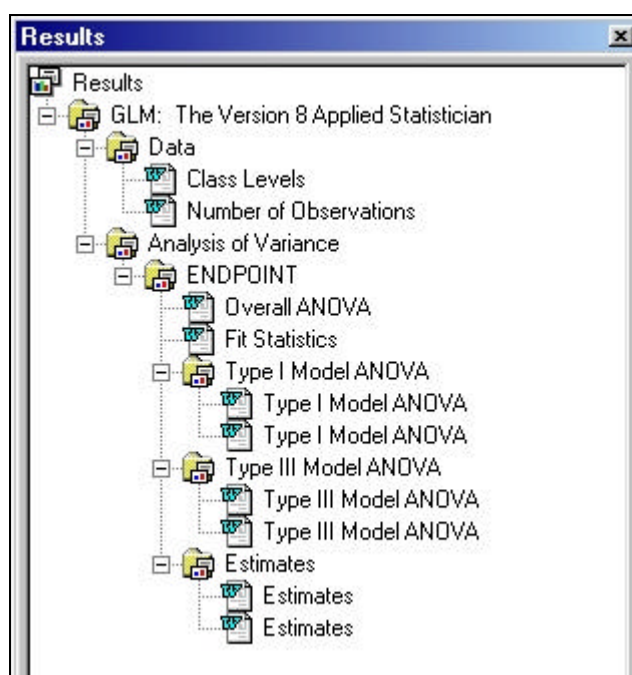
The following demonstrates how with ODS we can send output to rich text files (opened in Microsoft Word); in the first document keeping all the output and in the second document keeping only selected output:

```
ods rtf(1) file="report_all.rtf";
ods rtf(2) file="report_selected.rtf";

ods rtf(2) select modelanova estimates;

proc glm data=data.efficacy;
  class trtcd;
  model endpoint = trtcd ;
  estimate 'Active vs. All Others' trtcd 1 -0.5 -0.5;
run;
quit;
ods _all_ close;
```

Upon expanding the results window we can clearly see the sections of output are stored in Word documents. When a section was saved to both documents, two icons are shown.



Box and Whisker Plots

A new SAS/STAT procedure allows high resolution box and whisker plots to be drawn, with considerable control over both the methodology used in constructing the plots and the appearance of the output.

The basic syntax of the boxplot procedure is as follows:

```
proc boxplot < options > ;
  plot analysis-variable*group-variable < / options > ;
  by variables;
  id variables;
run;
```

Note that the input data set must be sorted by the group variable in the plot statement.

Of the many plot options available two of the most prominent are BOXSTYLE and BOXWIDTHSCALE whose syntax takes the following form (for full description of the methodology used with each of these styles, see the On-Line Documentation for proc boxplot under SAS/STAT):

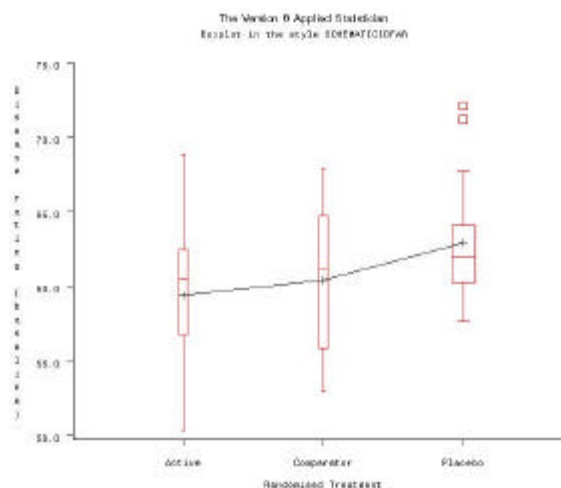
```
boxstyle = skeletal | schematid | schematicid | schematicidfar
boxwidthscale = value
```

When value of BOXWIDTHSCALE is positive, the widths are proportional to n^{value} . If value=0 the widths are proportional to $\log(n)$, where n is the sample size.

The following demonstrates a boxplot plot comparison of three treatment groups, joining each group's mean and identifying values beyond 1.5 x interquartile range. The width of the boxplot is also proportional to the sample size.

```
proc sort data=data.efficacy
    out=e1;
  by trtcd;
run;

proc boxplot data=e1;
  plot baseline * trtcd /
    boxwidthscale=1
    boxstyle=schematicidfar
    boxconnect;
run;
```



Multiple Imputations

The two procedures discussed in this section, Proc mi and mianalyze are only available in SAS 8.1 and above. Furthermore they remain experimental in version 8.2; hence principles of using these procedures are discussed rather than providing numeric demonstrations.

Most SAS procedures exclude from analyses an observation where a variable in the model has a missing value. This means the analysis loses information and does not take into account possible systematic differences between missing data, hence a resulting inference may not be applicable, especially where the sample size is small.

There are many methods of imputing missing values, however the method of multiple imputation can be used to represent a random sample of the missing values. This allows valid statistical inferences to be drawn, properly taking into account the uncertainty due to missing data.

Three steps are taken to draw inferences about parameters with multiple imputations:

1. Missing data are filled in m times to generate m complete data sets using proc mi.
2. The m complete data sets are each analysed using standard procedures, such as proc glm.
3. The results from the m analyses are combined to provide a valid inference using proc mianalyze.

There are three imputation mechanisms available in Proc MI. The method the statistician chooses depends on the pattern of missingness in the data:

- For data sets with monotone missing patterns, either a parametric or non-parametric regression method is appropriate



- or for data sets with arbitrary missing values a Markov Chain Monte Carlo (MCMC) method is appropriate.

New Features in Proc GLM

The clearest benefit of using proc glm, or any other SAS/STAT procedure in version 8 is ODS for the ability to capture any part of the output into a data set. There are some other useful new features too.

To generate confidence limits for parameter estimates and for the results of all estimate statements, use the CLPARM option on the model statement.

The new ALPHA= option on the proc glm statement specifies the level of statistical significance for confidence intervals.

The NAMELEN= option on the proc glm statement specifies the length of effect names in tables and output data sets. By default the length is 20, but can be increased up to 200.

Proc glm statements with new options are highlighted below:

```
proc glm data= namelen= alpha=;  
  class variables;  
  model dependent(s)=independent(s) / clparm ;  
  ...
```

New Features in Proc Logistic

A class statement has been added to proc logistic which allows interactions to be specified without the need for numerous dummy variables. The type of parameterisation can be specified, for example effect coding or reference coding.

There are several new model statement options for increased control over model fitting.

Exact tests of the parameters for the specified effects can be requested with the exact statement. Several exact statements can be used in a single proc logistic. Exact tests for the odds ratio can also be requested.

Proc logistic statements with new options are highlighted below:

```
proc logistic < options >;  
  class variables </cprefix= descending lprefix= order= param= ref=>;  
  model variable < (variable_options) > = < effects > < / absfconv=  
    fconv= gconv= xconv= ridging= >;  
  exact < 'label' > < Intercept >< effects > < / alpha= estimate=  
    joint jointly onesided outdist= > ;  
  ...
```

New Features in Proc Lifetest

The lifetest procedure now produces high resolution plots by default. The new options CENSORED SYMBOL= and EVENT SYMBOL= specify the symbol for the censored and event observations, respectively.

Proc lifetest statements with new options are highlighted below:

```
proc lifetest censoredsymbol=name eventsymbol=name  
  time variable < *tensor(list) > ;  
  ...
```



New Features in Proc Freq

The freq procedure now supports a test statement for asymptotic testing of selected measures of association and agreement. The BINOMIALC and BINOMIAL options request binomial proportions, confidence limits and tests for one-way tables, with and without a continuity correction respectively. Fleiss-Cohen scores can be requested and the individual row and column scores used for calculating statistics such as Pearson's correlation can also be displayed.

Proc freq statements with new options are highlighted below:

```
proc freq ;  
  table request(s) / binomial binomialc agree(wt=fc) scoreout ;  
  exact statistical-keyword(s) / option(s);  
  test statistical-keyword(s);  
  ...
```

New Features in Proc Npar1way

There are several new tests for scale differences (non-parametric test statistics) based on Ansari-Bradley, Klotz, Mood, and Siegel-Tukey scores. An exact statement allows exact p-values to be requested. Also Monte Carlo estimates of exact p-values can be requested.

Proc npar1way statements with new options are highlighted below:

```
proc npar1way < data= ab klotz mood st >  
  class variable;  
  exact statistic-options < / computation-options>;  
  freq variable;  
  ...
```

Compatibility with Version 6.12

Two procedures in version 6.12, proc genmod and proc mixed, used a preliminary version of the output delivery system via 'make' statements. These statements still function in version 8, however it is recommended that existing code is re-written to use the new ODS statements.

Whilst all other code written in version 6.12 runs in version 8 it is worth noting that some of the keywords used in the output have changed. On many occasions I have employed macros which read statistical output into a data set from a text file. This is done by scanning for specific key words in the output. As some of these key words have changed, for example in proc lifetest, the macros failed. Beware!

Other New Statistical Procedures

Several new procedures have been added to SAS/STAT since version 6.12. These are briefly described as follows:

- Proc PLS fits models using linear predictive models. The regression methods available are partial least squares, SIMPLS, principle components regression and reduced rank regression.
- Proc KRIGE2D performs ordinary kriging in two dimensions. Four semivariogram models are supported: the gaussian, exponential, spherical, and power models.
- Proc VARIOGRAM computes sample or empirical measures of spatial continuity for two-dimensional spatial data.



- Proc SURVEYSELECT selects probability based random samples, with any of the following methods: simple random sampling, unrestricted random sampling (with replacement), systematic random sampling or sequential random sampling.
- Proc SURVEYMEANS produces estimates of survey population means and totals from sample survey data.
- Proc SURVEYREG performs regression analysis for sample survey data. The procedure can handle complex survey sample designs, such as designs with stratification, clustering, and unequal weighting.
- Proc KDE performs either univariate or bivariate kernel density estimation.
- Proc LOESS implements a nonparametric method for estimating regression surfaces.
- Proc TPSPLINE uses the penalized least squares method to fit a nonparametric regression model.
- Proc NLMIXED fits nonlinear mixed models, that is, models in which both fixed and random effects enter nonlinearly

CONCLUSIONS

The version 8 applied statistician has both a broader range of statistical methods and considerably more flexibility in the way statistical output can be captured and reported.

Several new descriptive statistics can be obtained from base SAS procedures. Furthermore quantile statistics can be generated by all base SAS summarising procedures. The choice of which procedure to use in many cases may be down to individual preference. High resolution graphics can be easily obtained for exploratory data analysis.

The output delivery system provides a mechanism where any part of our procedures output can be captured into SAS data sets, but not only that, this output can be stored in numerous formats rather than the traditional fixed space font output. It is even possible to send the same output to multiple files simultaneously.

Statistical procedures have been enhanced and new ones introduced. Methods are available for which the statistician may have previously required an additional statistical package, such as exact tests or non-linear mixed models.

REFERENCES

Yang C. Yuan, "Multiple Imputation for Missing Data: Concepts and New

TRADEMARK INFORMATION

SAS® is a registered trademark of the SAS Institute Inc. Cary, NC, USA.

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



THE AUTHOR

I very much welcome your feedback on this paper and your thoughts on working as an applied statistician with SAS version 8. You may contact me by sending an email to david.shannon@amadeus.co.uk or even by the traditional method:

David Shannon
Amadeus Software Ltd.
Orchard Farm
Witney Lane
Leafield
Oxfordshire
OX29 9PG

www.amadeus.co.uk

Version 1.0, 13th August 2001.