SAS139-2022

Generating Simple Statistics with Base SAS Procedures

Jane Eslinger, SAS Institute Inc.

ABSTRACT

Multiple Base SAS procedures generate simple statistics such as mean, min, and max. As a programmer, it is always good to know which procedures do what. This paper and presentation compares and contrasts generating simple statistics with the MEANS, UNIVARIATE, TABULATE, and REPORT procedures.

INTRODUCTION

Programmers often need to generate both simple statistics and descriptive statistics. The Base SAS procedures use a standardized set of keywords to refer to statistics. After you learn the keyword to the statistic that you need, it can be used with all appropriate procedures.

This paper focuses on four Base procedures that are often used by programmers to generate needed statistics. The MEANS, UNIVARIATE, TABULATE, and REPORT procedures have different overall purposes, but all are capable to generating statistics. Through examples, this paper presents the code required by each procedure.

The goal of this paper is to empower you as a programmer. Exploring multiple ways to generate the same statistic gives you more choices for writing robust code. You will be able to make choices about the procedure that works best and most efficiently for your program.

LEVEL SETTING

Base SAS procedures can generate dozens of statistics. This paper uses only the sum, min, max, and mean statistics to illustrate how to request the statistics within each procedure. The same concepts apply to the other statistics. Please see the Reference section of this paper for a link to the documentation page with a list of available statistics.

Furthermore, each example is designed to be as straightforward as possible. The intention is to illustrate the statistics, not all the statements that could possibly be used within a procedure. The same data set, sashelp.shoes, is used for all examples. Two classification variables are used to make the examples more realistic.

In addition, this paper contains syntax for generating printed output and syntax for generating data sets. Typically, a programmer chooses one or the other. Some procedures such as PROC UNIVARIATE generate large amounts of printed out. ODS SELECT and ODS EXCLUDE statements are used in the examples to reduce the printed output to the tables of interest. Please see the Reference section of this paper for a link to the documentation page on these statements. Also, the documentation for each procedure contains a list of the output object names for each piece of generated output.

PRINTED OUTPUT

This section illustrates the syntax required by the MEANS, UNIVARIATE, TABULATE, and REPORT procedures, for printed output. Though they generate the same statistics, each procedure displays the numbers differently. The programmer must determine whether one procedure creates output that is more aesthetically pleasing than the others.

PROC MEANS

The MEANS procedure provides data summarization tools to compute descriptive statistics for variables across all observations and within groups of observations.

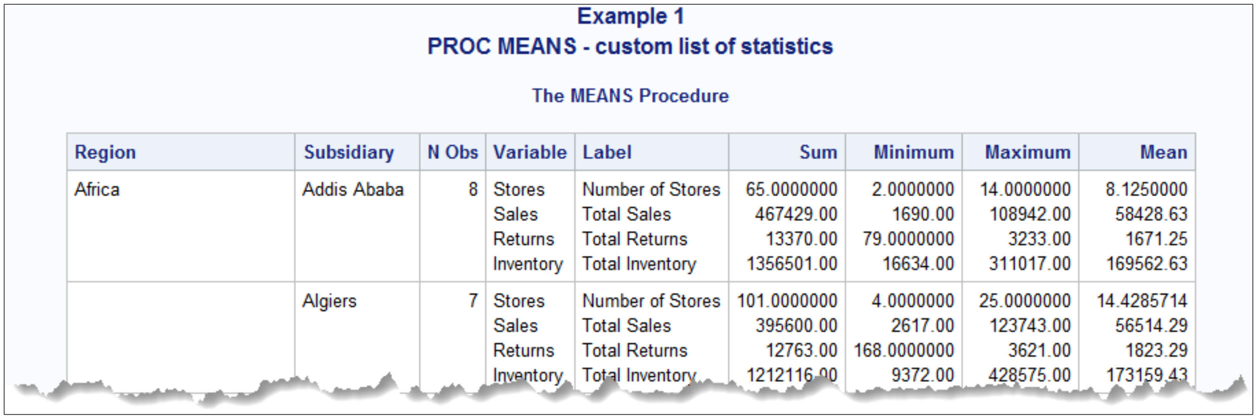
By default, when generating printed output, PROC MEANS provides the following statistics:

- number of observations
- number of nonmissing values
- mean
- standard deviation
- minimum
- maximum

A custom selection of desired statistics is listed in the PROC MEANS statement. Example 1 illustrates the syntax for requesting the printing of the sum, min, max, and mean statistics for four numeric variables within two classification variables.

Example 1:

```
proc means data=sashelp.shoes sum min max mean;
    class region subsidiary;
    var stores sales returns inventory;
run;
```



Example 1
PROC MEANS - custom list of statistics

The MEANS Procedure

Region	Subsidiary	N Obs	Variable	Label	Sum	Minimum	Maximum	Mean
Africa	Addis Ababa	8	Stores Sales Returns Inventory	Number of Stores Total Sales Total Returns Total Inventory	65.0000000 467429.00 13370.00 1356501.00	2.0000000 1690.00 79.0000000 16634.00	14.0000000 108942.00 3233.00 311017.00	8.1250000 58428.63 1671.25 169562.63
	Algiers	7	Stores Sales Returns Inventory	Number of Stores Total Sales Total Returns Total Inventory	101.0000000 395600.00 12763.00 1212116.00	4.0000000 2617.00 168.0000000 9372.00	25.0000000 123743.00 3621.00 428575.00	14.4285714 56514.29 1823.29 173159.43

Display 1. Partial PROC MEANS Output

PROC UNIVARIATE

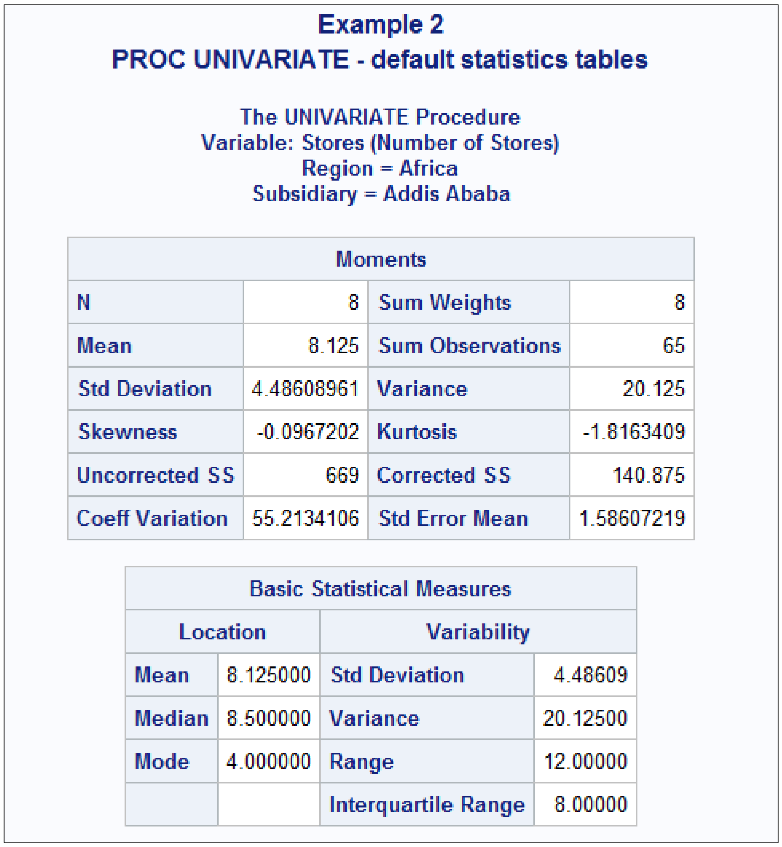
PROC UNIVARIATE calculates summary statistics, generates quantiles and percentiles, performs tests for goodness of fit, and produces probability plots.

By default, the other procedures in this paper place all combinations in the same table. By default, PROC UNIVARIATE generates separate tables for each combination of classification variables. It also separates various sets of statistics into multiple tables. This results in PROC UNIVARIATE generating a large amount of output. Example 2 uses an ODS SELECT statement to limit the output to the two tables that display the statistics of interest: sum, min, max, and mean.

Example 2 illustrates PROC UNIVARIATE syntax.

Example 2:

```
ods select moments basicmeasures;
proc univariate data=sashelp.shoes;
    class region subsidiary;
    var stores sales returns inventory;
run;
```



Example 2

PROC UNIVARIATE - default statistics tables

The UNIVARIATE Procedure
Variable: Stores (Number of Stores)
Region = Africa
Subsidiary = Addis Ababa

Moments			
N	8	Sum Weights	8
Mean	8.125	Sum Observations	65
Std Deviation	4.48608961	Variance	20.125
Skewness	-0.0967202	Kurtosis	-1.8163409
Uncorrected SS	669	Corrected SS	140.875
Coeff Variation	55.2134106	Std Error Mean	1.58607219

Basic Statistical Measures			
Location		Variability	
Mean	8.125000	Std Deviation	4.48609
Median	8.500000	Variance	20.12500
Mode	4.000000	Range	12.00000
		Interquartile Range	8.00000

Display 2. Partial PROC UNIVARIATE OUTPUT

There are no options in the PROC UNIVARIATE procedure for requesting a subset of the statistics. An output data set must be created. An example in another section of this paper shows the syntax for creating an output data set.

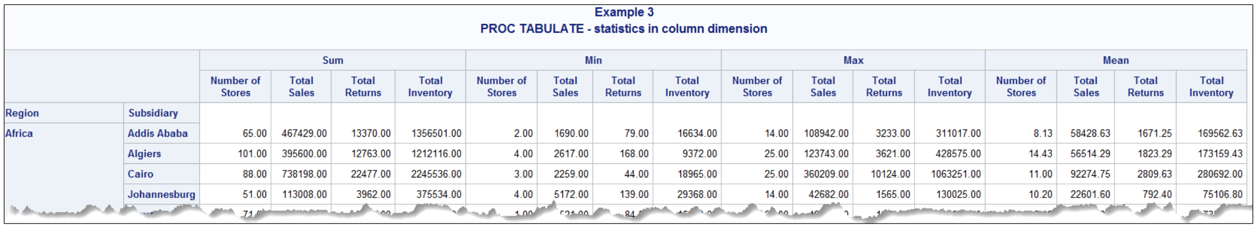
PROC TABULATE

The TABULATE procedure displays descriptive statistics in tabular format, using some or all of the variables in a data set. Analysis variables are listed in the VAR statement. The keywords for the desired statistics are placed in the TABLE statement. The placement of the statistical keyword, along with the use of parentheses and asterisks, determine which analysis variable it is calculated for. It is possible to request a different statistic for each analysis variable.

The TABLE statement in Example 3 requests all four statistics for the four analysis variables.

Example 3:

```
proc tabulate data=sashelp.shoes;
   class region subsidiary;
   var stores sales returns inventory;
   table region*subsidiary,
                (sum min max mean)*(stores sales returns inventory);
run;
```



Example 3
PROC TABULATE - statistics in column dimension

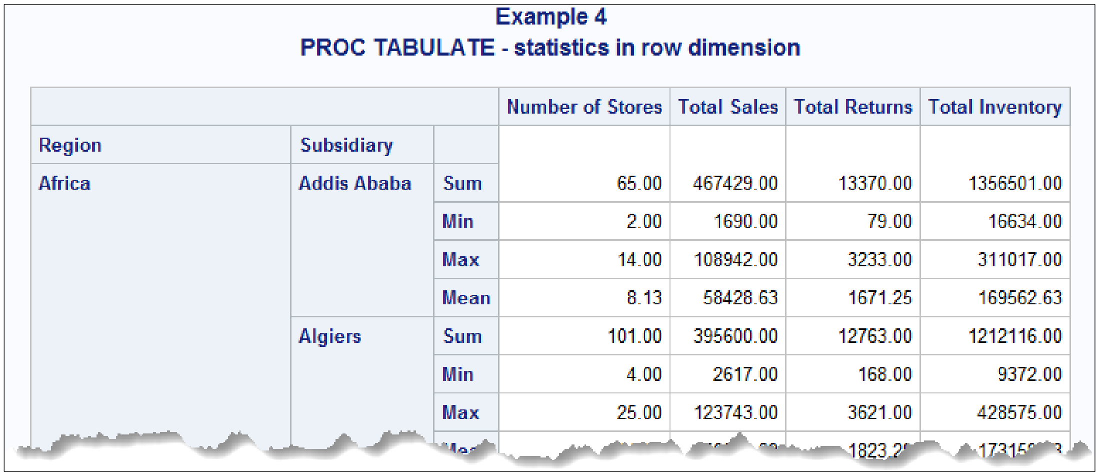
		Sum				Min				Max				Mean			
		Number of Stores	Total Sales	Total Returns	Total Inventory	Number of Stores	Total Sales	Total Returns	Total Inventory	Number of Stores	Total Sales	Total Returns	Total Inventory	Number of Stores	Total Sales	Total Returns	Total Inventory
Region	Subsidiary																
Africa	Addis Ababa	65.00	467429.00	13370.00	1356501.00	2.00	1690.00	79.00	16634.00	14.00	108942.00	3233.00	311017.00	8.13	58428.63	1671.25	169562.63
	Algiers	101.00	395600.00	12763.00	1212116.00	4.00	2617.00	168.00	9372.00	25.00	123743.00	3621.00	428575.00	14.43	56514.29	1823.29	173159.43
	Cairo	88.00	738198.00	22477.00	2245536.00	3.00	2259.00	44.00	18965.00	25.00	360209.00	10124.00	1063251.00	11.00	92274.75	2809.63	280692.00
	Johannesburg	51.00	113008.00	3962.00	375534.00	4.00	5172.00	139.00	29368.00	14.00	42682.00	1565.00	130025.00	10.20	22601.60	792.40	75106.80

Display 3. Partial PROC TABULATE Output

The TABLE statement provides flexibility when structuring the output table. Variables and statistics can be defined in the row dimension or the column dimension. Example 4 generates the same values as Example 3. However, the statistics are moved to the row dimension.

Example 4:

```
proc tabulate data=sashelp.shoes;
    class region subsidiary;
    var stores sales returns inventory;
    table region*subsidiary*(sum min max mean),
           (stores sales returns inventory);
run;
```



Example 4
PROC TABULATE - statistics in row dimension

			Number of Stores	Total Sales	Total Returns	Total Inventory
Region	Subsidiary					
Africa	Addis Ababa	Sum	65.00	467429.00	13370.00	1356501.00
		Min	2.00	1690.00	79.00	16634.00
		Max	14.00	108942.00	3233.00	311017.00
		Mean	8.13	58428.63	1671.25	169562.63
	Algiers	Sum	101.00	395600.00	12763.00	1212116.00
		Min	4.00	2617.00	168.00	9372.00
		Max	25.00	123743.00	3621.00	428575.00

Display 4. Partial PROC TABULATE Output

The circumstance dictates the best placement of the statistics within the TABLE statement.

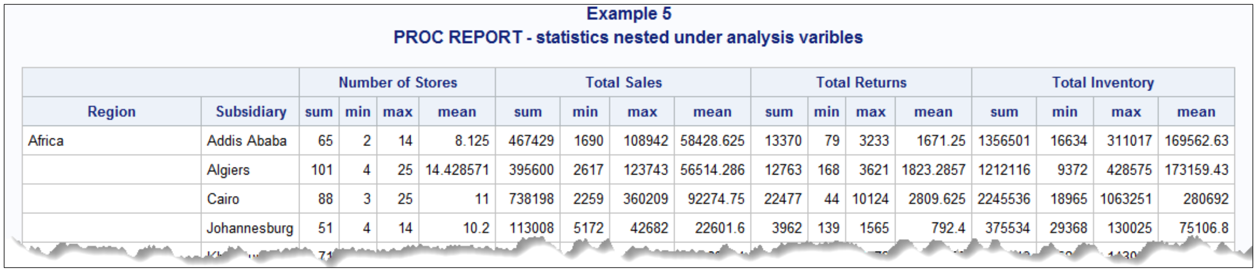
PROC REPORT

PROC REPORT combines aspects of PROC PRINT, PROC MEANS, and PROC TABULATE with features of the DATA step to produce a variety of reports.

Like PROC TABULATE, PROC REPORT is very flexible when requesting multiple statistics for multiple analysis variables. In Example 5, the statistics are nested under the analysis variable using parentheses and a comma.

Example 5:

```
proc report data=sashelp.shoes;
    column region subsidiary (stores sales returns inventory),
            (sum min max mean);
    define region / group;
    define subsidiary / group;
run;
```



Example 5

PROC REPORT - statistics nested under analysis varibles

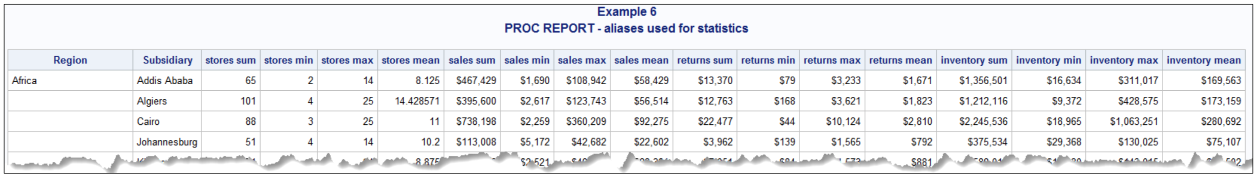
		Number of Stores				Total Sales				Total Returns				Total Inventory			
Region	Subsidiary	sum	min	max	mean	sum	min	max	mean	sum	min	max	mean	sum	min	max	mean
Africa	Addis Ababa	65	2	14	8.125	467429	1690	108942	58428.625	13370	79	3233	1671.25	1356501	16634	311017	169562.63
	Algiers	101	4	25	14.428571	395600	2617	123743	56514.286	12763	168	3621	1823.2857	1212116	9372	428575	173159.43
	Cairo	88	3	25	11	738198	2259	360209	92274.75	22477	44	10124	2809.625	2245536	18965	1063251	280692
	Johannesburg	51	4	14	10.2	113008	5172	42682	22601.6	3962	139	1565	792.4	375534	29368	130025	75106.8

Display 5. Partial PROC REPORT Output

Instead of nesting the statistics, Example 6 illustrates the syntax for using aliases within PROC REPORT. This method requires much more code, but it provides more control over each column for the purposes of formatting.

Example 6:

```
proc report data=sashelp.shoes;
   column region subsidiary stores stores=st2 stores=st3 stores=st4
         sales sales=s2 sales=s3 sales=s4
         returns returns=r2 returns=r3 returns=r4
         inventory inventory=i2 inventory=i3 inventory=i4;
   define region / group;
   define subsidiary / group;
   define stores / sum 'stores sum';
   define st2 / min 'stores min';
   define st3 / max 'stores max';
   define st4 / mean 'stores mean';
   define sales / sum 'sales sum';
   define s2 / min 'sales min';
   define s3 / max 'sales max';
   define s4 / mean 'sales mean';
   define returns / sum 'returns sum';
   define r2 / min 'returns min';
   define r3 / max 'returns max';
   define r4 / mean 'returns mean';
   define inventory / sum 'inventory sum';
   define i2 / min 'inventory min';
   define i3 / max 'inventory max';
   define i4 / mean 'inventory mean';
run;
```



Example 6

PROC REPORT - aliases used for statistics

Region	Subsidiary	stores sum	stores min	stores max	stores mean	sales sum	sales min	sales max	sales mean	returns sum	returns min	returns max	returns mean	inventory sum	inventory min	inventory max	inventory mean
Africa	Addis Ababa	65	2	14	8.125	$467,429	$1,690	$108,942	$58,429	$13,370	$79	$3,233	$1,671	$1,356,501	$16,634	$311,017	$169,563
	Algiers	101	4	25	14.428571	$395,600	$2,617	$123,743	$56,514	$12,763	$168	$3,621	$1,823	$1,212,116	$9,372	$428,575	$173,159
	Cairo	88	3	25	11	$738,198	$2,259	$360,209	$92,275	$22,477	$44	$10,124	$2,810	$2,245,536	$18,965	$1,063,251	$280,692
	Johannesburg	51	4	14	10.2	$113,008	$5,172	$42,682	$22,602	$3,962	$139	$1,565	$792	$375,534	$29,368	$130,025	$75,107

Display 6. Partial PROC REPORT Output

OUTPUT DATA SETS

This section illustrates the syntax required by the MEANS, UNIVARIATE, TABULATE, and REPORT procedures for generating output data sets. The syntax of the procedure dictates the structure of the data set. For example, the value of the statistics might be in their own variables, or they might be in one variable with each row representing a different statistic. As with the printed output, the programmer has a decision to make. The choice is based on how the data set is used downstream within the program and, often, which procedure the programmer is the most comfortable using.

PROC MEANS

In PROC MEANS, the OUTPUT statement creates the output data set. The OUT= names the new data set. The keywords for the desired statistics are listed. Example 7 uses the syntax of the keyword followed by an equal sign and the AUTONAME option. However, this is not the only acceptable syntax for requesting statistics. The example requests the four statistics for all four analysis variables. Alternatively, it could request different combinations of analysis variables and statistics. The programmer could also supply the names for the output variables.

Also note that no statistical keywords are listed in the PROC MEANS statement. The PROC MEANS statement controls the printed output, which is being suppressed with the NOPRINT option. Therefore, it is unnecessary and ineffective to list statistics in the PROC MEANS statement if only the output data set is needed.

Example 7:

```
proc means data=sashelp.shoes noprint;
   class region subsidiary;
   var stores sales returns inventory;
   output out=ds_means sum= min= max= mean= / autoname;
run;
```

PROC UNIVARIATE

As noted previously, an output data set must be created with PROC UNIVARIATE for requesting a subset of the statistics.

Unlike PROC MEANS, the OUTPUT statement within PROC UNIVARIATE must contain a specification of the form keyword=names. Any combination of analysis variable and statistical keyword is acceptable. Example 8 illustrates the syntax for requesting all four statistics for the four analysis variables.

Example 8:

```
proc univariate data=sashelp.shoes noprint;
   class region subsidiary;
   var stores sales returns inventory;
   output out=ds_univariate
         sum= storessum salessum returnssum inventorysum
         min= storesmin salesmin returnsmin inventorymin
         max= storesmax salesmax returnsmax inventorymax
         mean= storesmean salesmean returnsmean inventorymean;
run;
```

PROC TABULATE

The NOPRINT option is not valid in the PROC TABULATE statement like it was in the PROC MEANS and PROC UNIVARIATE statements. To suppress printed output, the ODS EXCLUDE statement is required.

For PROC TABULATE, an output data set is created with the OUT= option in the PROC TABULATE statement. PROC TABULATE determines the names of the variables in the output set. The naming convention for the statistics in the output data set is variablename_statisticname. For a programmer to control the names, the RENAME= data set option must be used.

Example 9 illustrates the syntax of using the OUT= option.

Example 9:

```
ods exclude all;
proc tabulate data=sashelp.shoes out=ds_tabulate;
   class region subsidiary;
   var stores sales returns inventory;
   table region*subsidiary,
         (sum min max mean)*(stores sales returns inventory);
run;
ods select all;
```

In Example 4, the statistical keywords were placed in the row dimension of the printed table. An output data set generated with that syntax generates the same table as Example 9 does.

PROC REPORT

The elemental purpose of PROC REPORT is to create a final printed report. However, it does have an OUT= option for generating an output data set. As with PROC TABULATE, an ODS EXCLUDE or ODS SELECT statement is required to suppress the printed output.

Example 10 nests the statistics under the analysis variables. The variables in the output data set are named by column number, _cn_. Most programmers do not like this naming convention because it is hard to determine what variable and statistic it represents without looking at the code. The RENAME= data set option must be used to change the names created by PROC REPORT. That can be cumbersome though.

Example 10:

```
ods exclude all;
proc report data=sashelp.shoes out=ds_report1;
   column region subsidiary (stores sales returns inventory), (sum min max
mean);

   define region / group;
   define subsidiary / group;
run;
ods select all;
```

Example 11 uses the same syntax as Example 6, which uses aliases. The variable names in the output data set correspond to the alias name. Otherwise, the structure of the data sets created in Example 11 matches the structure in Example 10. This is means the RENAME= can be avoided.

Example 11:

```
ods exclude all;
proc report data=sashelp.shoes out=ds_report2;
   column region subsidiary stores stores=st2 stores=st3 stores=st4
          sales sales=s2 sales=s3 sales=s4
          returns returns=r2 returns=r3 returns=r4
          inventory inventory=i2 inventory=i3 inventory=i4;
   define region / group;
   define subsidiary / group;
   define stores / sum 'stores sum';
   define st2 / min 'stores min';
   define st3 / max 'stores max';
   define st4 / mean 'stores mean';
   define sales / sum 'sales sum';
   define s2 / min 'sales min';
   define s3 / max 'sales max';
   define s4 / mean 'sales mean';
   define returns / sum 'returns sum';
   define r2 / min 'returns min';
   define r3 / max 'returns max';
   define r4 / mean 'returns mean';
   define inventory / sum 'inventory sum';
   define i2 / min 'inventory min';
   define i3 / max 'inventory max';
   define i4 / mean 'inventory mean';
run;
ods select all;
```

CONCLUSION

The intention of this paper was to illustrate the syntax for generating simple statistics using multiple Base SAS procedures. It concentrated on the MEANS, UNIVARIATE, TABULATE, and REPORT procedures that are commonly used. The choice between which procedure to use is based on the look of the output or the structure of the output data set. Hopefully, this paper helps you, the programmer, determine which procedure is most beneficial for your program.

REFERENCES

Eslinger, Jane. 2019. "It's All about the Base—Procedures." In *Proceedings of the SAS Global Forum 2019 Conference*. Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3068-2019.pdf.

Eslinger, Jane. 2020. "It's All about the Base—Procedures, Part 2." In *Proceedings of the SAS Global Forum 2020 Conference*. Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4092-2020.pdf.

ACKNOWLEDGMENTS

This author is grateful to Kathryn McLawhorn for contributing to this paper, and to Susan Willard, who edited the paper.

RECOMMENDED READING

- SAS Institute Inc. 2022. "SAS Elementary Statistics Procedures." In *SAS® Viya® Programming Documentation*. Cary, NC: SAS Institute Inc. https://go.documentation.sas.com/doc/en/pgmsascdc/v_028/proc/n16h0f7xj8802hn1cdmr8745wdjp.htm.
- SAS Institute Inc. 2022. "Dictionary of ODS Language Statements: ODS SELECT Statement." In *SAS® Viya® Programming Documentation*. Cary, NC: SAS Institute Inc. https://go.documentation.sas.com/doc/en/pgmsascdc/v_028/odsug/p1b72ff70v3obrn16aanp1r9q2bu.htm.
- SAS Institute Inc. 2022. "Dictionary of ODS Language Statements: ODS EXCLUDE Statement." In *SAS® Viya® Programming Documentation*. Cary, NC: SAS Institute Inc. https://go.documentation.sas.com/doc/en/pgmsascdc/v_028/odsug/n0edvjbwu4y1bun1qgmzbk3s3vix.htm.
- SAS Institute Inc. 2014. "Usage Note 5250: Tips for determining which Base SAS® procedure to choose." https://support.sas.com/kb/52/520.html.

  McLawhorn, Kathryn. 2014. "Which Base procedure is best for simple statistics?" *SAS Users* (blog). SAS Institute Inc. https://blogs.sas.com/content/sgf/2014/05/09/which-base-procedure-is-best-for-simple-statistics/. *SAS Users* (blog). SAS Institute Inc. https://blogs.sas.com/content/sgf/. Last modified May 9, 2014.

CONTACT INFORMATION

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

Jane Eslinger
SAS Institute Inc.
Jane.Eslinger@sas.com
www.sas.com