Paper 2625-2018

Using the COPULA Procedure to Simulate Multivariate Data

Bill Qualls, First Analytics; Ben Pineda, Kellogg;
Rob Stevens, First Analytics

ABSTRACT

Simulating data is a common task for data scientists. In our scenario, our client needed to simulate a large number of observations of multivariate data based on a small number of real observations. When faced with such a task, an analyst will typically take a univariate approach, perhaps using PROC UNIVARIATE to produce a histogram of the data to determine a candidate distribution, as well as to obtain the best available estimates of the underlying parameters of the distribution. Having done so, they will then use one of the many distribution specific random number generators to simulate more rows. Data generated in this way should reflect the original distribution well. But there is a serious shortcoming with this univariate approach: it ignores any correlations which existed between the columns. A better method of generating such data is by using the COPULA procedure. When data is generated with PROC COPULA, not only will the columns within fields have the same parameters, but so will columns between fields. This paper will use SAS code to demonstrate the need for PROC COPULA, as well as how to use PROC COPULA. It will also discuss how PROC COPULA was used to better address our client's needs.

INTRODUCTION

Kellogg engaged First Analytics in a project which required simulating multivariate promotional data to produce guidelines for promotion planning purposes. Traditional methods of simulating univariate data were inappropriate as they would ignore the highly correlated nature of the original data. The solution was to use SAS® PROC COPULA. This paper will demonstrate how to use SAS® PROC COPULA to simulate multivariate data.

BUSINESS PROBLEM

Most consumer package goods (CPG) companies offer incentives to retailers so that they will in turn promote their goods. Once a retailer decides to run a promotion, it can take the form of (1) a temporary price reduction ("discount") off the shelf price to the customer, (2) including the company's products in advertising ("feature"), and/or (3) giving the company's products a prime location within the store ("display"). For example, in a promotion of Kellogg's Frosted Flakes, Safeway might agree to (1) a temporary price reduction of 10% to the consumer, (2) 50% of Safeway stores will include Frosted Flakes in feature and display promotions (FD), (3) another 20% of Safeway stores will include Frosted Flakes in feature promotions only (FO), and (4) another 15% of Safeway stores will include Frosted Flakes in display promotions only (DO).

It was Kellogg's desire to provide their planners with a tool which would include 75% and 95% "guardrails" of reasonable combinations of discount, FD, FO, and DO percentages. Kellogg faced two challenges in doing so. First, there was a shortage of data. The median number of data points for a given retailer and product was only 226. Second, the values were highly correlated. Indeed, FD + FO + DO cannot exceed 100%, and the higher the sum, the higher the discount. It was determined that more data was needed, and that the data would be obtained through simulation.

This paper will demonstrate the shortcomings of using a univariate approach to simulate multivariate data, and how these shortcomings are overcome by using SAS® PROC COPULA.

OUR DATA

Clearly, we cannot publish Kellogg data in this paper. But any multivariate data will suffice to demonstrate. For the readers' convenience, we will use Fisher's Iris data, which is found in SASHELP.IRIS. Recall this dataset contains fifty observations of four measures (SepalLength, SepalWidth, PetalLength, and PetalWidth) for each of three Species (Setosa, Versicolor, and Virginica). We will limit ourselves to the Setosa rows. We begin by creating the work data set, determining summary statistics (specifically, mean and standard deviation), and producing a correlation matrix. Throughout this paper we will be showing the same results for three different datasets, so we have written a "helper" macro (%show, included at the end of this paper) to simplify that work.

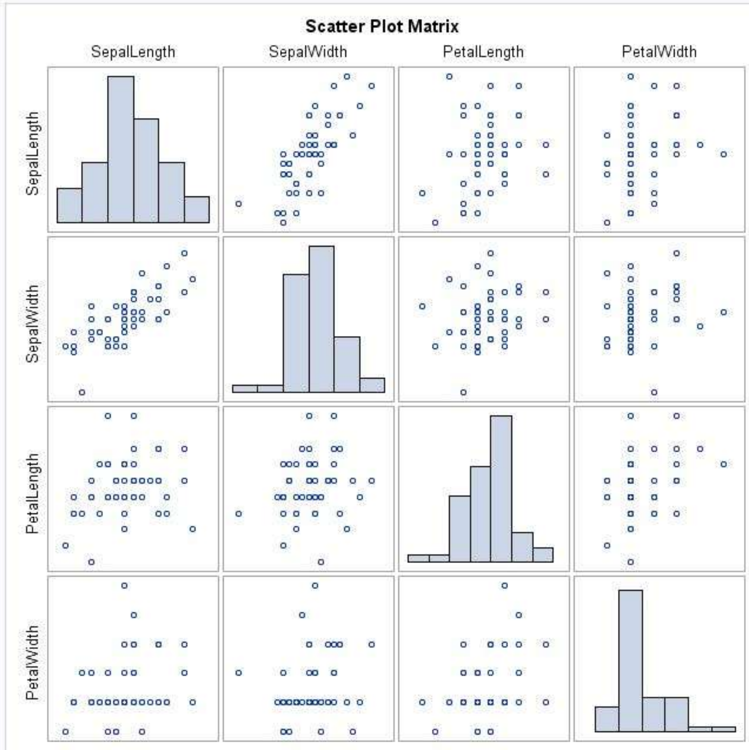
Creating our Setosa data set:

```
*------------------------------------ ;
*     O R I G I N A L   D A T A
*------------------------------------ ;

* for this example I will use a subset of iris dataset;
%let VARS=SepalLength SepalWidth PetalLength PetalWidth;
data work.orig_data (keep=&VARS);
set sashelp.iris
 (where=(species = "Setosa"));
run;

%show(TBL=orig);
```

Outputs from PROC CORR for our Setosa data follow:

Simple Statistics							
Variable	**N**	**Mean**	**Std Dev**	**Sum**	**Minimum**	**Maximum**	**Label**
SepalLength	50	50.06000	3.52490	2503	43.00000	58.00000	Sepal Length (mm)
SepalWidth	50	34.28000	3.79064	1714	23.00000	44.00000	Sepal Width (mm)
PetalLength	50	14.62000	1.73664	731.00000	10.00000	19.00000	Petal Length (mm)
PetalWidth	50	2.46000	1.05386	123.00000	1.00000	6.00000	Petal Width (mm)

Pearson Correlation Coefficients, N = 50 Prob > |r| under H0: Rho=0				
	SepalLength	**SepalWidth**	**PetalLength**	**PetalWidth**
SepalLength Sepal Length (mm)	1.00000	0.74255 <.0001	0.26718 0.0607	0.27810 0.0505
SepalWidth Sepal Width (mm)	0.74255 <.0001	1.00000	0.17770 0.2170	0.23275 0.1038
PetalLength Petal Length (mm)	0.26718 0.0607	0.17770 0.2170	1.00000	0.33163 0.0186
PetalWidth Petal Width (mm)	0.27810 0.0505	0.23275 0.1038	0.33163 0.0186	1.00000



We see that the correlation coefficients vary from a high of 0.74 for SepalLength and SepalWidth to a low of 0.18 for SepalWidth and PetalLength. We also observe that the individual variables appear to be sufficiently normal.

THE NAÏVE APPROACH

The naïve approach to simulating data would be to use the mean and standard deviation for each variable to produce random data from a distribution with those parameters. We call this the naïve approach because it ignores the correlations between the variables. Let's do this anyway and see what happens.

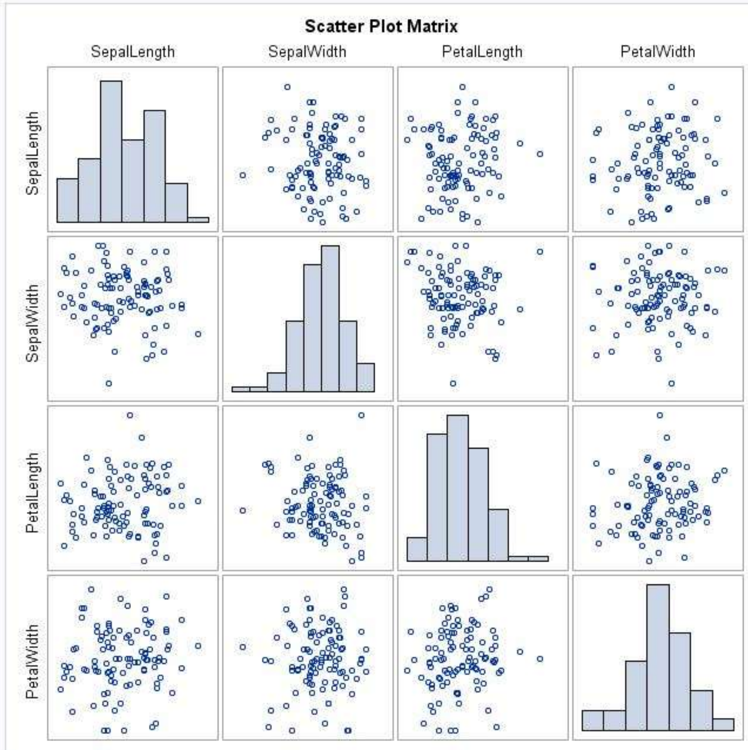
Recall that SAS' rannor returns $z \sim N(0,1)$. We can convert these z values to values from $N(\mu, \sigma^2)$ by using $x = \mu + z\sigma$ or, more correctly, $x = \bar{x} + zs$:

```
*------------------------------------ ;
*    N A I V E   A P P R O A C H
*------------------------------------ ;

* naive attempt at generating more data, ignores correlations;
%let GENERATE = 100;
%let SEED = 1234;

data work.naive_data (keep=&VARS);
set work.orig_stats;
SEED = &SEED;
do i = 1 to &GENERATE;
   call rannor(SEED, z);
   SepalLength = SepalLength_Mean + (z * SepalLength_StdDev);
   SepalLength = max(SepalLength, 0);
   call rannor(SEED, z);
   SepalWidth  = SepalWidth_Mean  + (z * SepalWidth_StdDev );
   SepalWidth = max(SepalWidth, 0);
   call rannor(SEED, z);
   PetalLength = PetalLength_Mean + (z * PetalLength_StdDev);
   PetalLength = max(PetalLength, 0);
   call rannor(SEED, z);
   PetalWidth  = PetalWidth_Mean  + (z * PetalWidth_StdDev );
   PetalWidth = max(PetalWidth, 0);
   output;
end;
run;

%show(TBL=naive);
```

Outputs from PROC CORR for our naïve approach data follow:

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
SepalLength	100	50.08999	3.51986	5009	42.93365	58.73144
SepalWidth	100	34.31659	4.01349	3432	21.29352	42.11967
PetalLength	100	15.15906	1.79437	1516	11.25408	21.10015
PetalWidth	100	2.46109	1.04527	246.10925	0	4.85048

Pearson Correlation Coefficients, N = 100 Prob > |r| under H0: Rho=0				
	SepalLength	SepalWidth	PetalLength	PetalWidth
SepalLength	1.00000	-0.08875	0.11051	0.09333
		0.3799	0.2737	0.3557
SepalWidth	-0.08875	1.00000	-0.13474	0.04160
	0.3799		0.1814	0.6811
PetalLength	0.11051	-0.13474	1.00000	0.14155
	0.2737	0.1814		0.1601
PetalWidth	0.09333	0.04160	0.14155	1.00000
	0.3557	0.6811	0.1601	



We see that the sample statistics ("Mean" and "Std Dev" columns) compare favorably to those of the original data, but the correlations do not. In fact, none of the correlations are statistically significant.

USING THE COPULA PROCEDURE

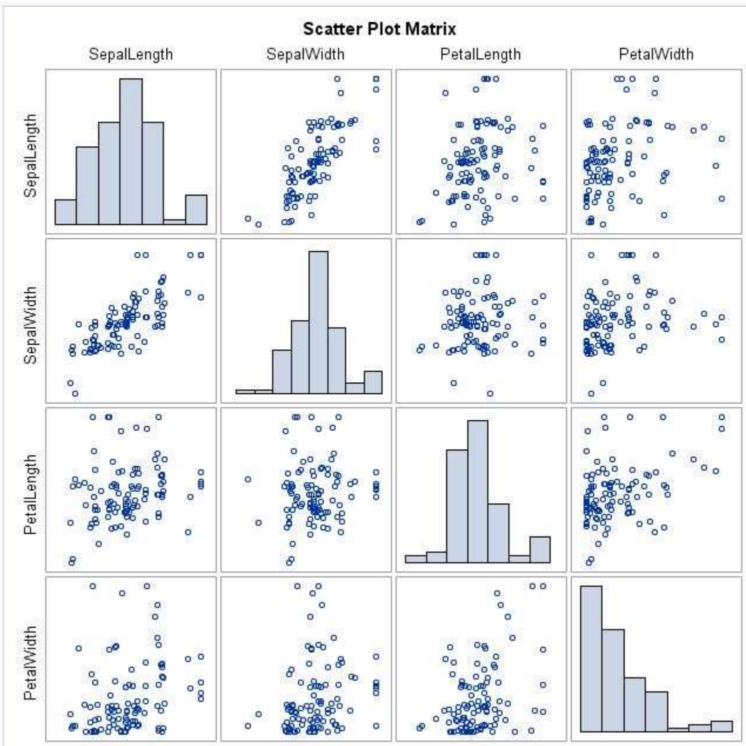
The naïve approach failed because it did not consider the underlying structure of the data; that is, the correlations. SAS® PROC COPULA will do so. This paper will not attempt to explain how PROC COPULA works, only demonstrate how to use it and show that it works. Note PROC COPULA requires SAS/ETS®:

```
*------------------------------------ ;
*  U S I N G   P R O C   C O P U L A
*------------------------------------ ;

* using proc copula to generate more data;
title "PROC COPULA";
proc copula data=work.orig_data;
var &VARS;
fit normal;
simulate / ndraws=&GENERATE
  SEED=&SEED
  out=work.copula_data;
run;
title;

%show(TBL=copula);
```

Outputs from PROC CORR for the data generated by PROC COPULA follow:

Simple Statistics							
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum	Label
SepalLength	100	49.55318	3.49999	4955	43.00000	58.00000	Sepal Length (mm)
SepalWidth	100	34.03565	4.00626	3404	23.00000	44.00000	Sepal Width (mm)
PetalLength	100	14.30672	1.77260	1431	10.00000	19.00000	Petal Length (mm)
PetalWidth	100	2.11862	1.17418	211.86245	1.00000	6.00000	Petal Width (mm)

Pearson Correlation Coefficients, N = 100 Prob > |r| under H0: Rho=0				
	SepalLength	SepalWidth	PetalLength	PetalWidth
SepalLength	1.00000	0.77527	0.23128	0.29402
Sepal Length (mm)		<.0001	0.0206	0.0030
SepalWidth	0.77527	1.00000	0.09229	0.20925
Sepal Width (mm)	<.0001		0.3611	0.0367
PetalLength	0.23128	0.09229	1.00000	0.38740
Petal Length (mm)	0.0206	0.3611		<.0001
PetalWidth	0.29402	0.20925	0.38740	1.00000
Petal Width (mm)	0.0030	0.0367	<.0001	



CONCLUSION

The following table summarizes the various correlations. By comparing the "Original Data" column to the "PROC COPULA" column, we see that PROC COPULA does a good job of capturing the underlying structure of the data:

	Original Data	Naïve Approach	PROC COPULA
SepalLength:SepalWidth	0.74	-0.09	0.78
SepalLength:PetalLength	0.27	0.11	0.23
SepalLength:PetalWidth	0.28	0.09	0.29
SepalWidth:PetalLength	0.18	-0.13	0.09
SepalWidth:PetalWidth	0.23	0.04	0.21
PetalLength:PetalWidth	0.33	0.14	0.39

REFERENCES

"SAS/ETS® 13.2 User's Guide: The COPULA Procedure." Available at https://support.sas.com/documentation/onlinedoc/ets/132/copula.pdf.

CONTACT INFORMATION

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

Bill Qualls
First Analytics
bqualls@firstanalytics.com

Ben Pineda
Kellogg
Benjamin.Pineda@Kellogg.com

Rob Stevens
First Analytics
rstevens@firstanalytics.com

APPENDIX: THE SHOW MACRO

All code needed to try this on your own has been shown in this paper, except for the show macro, which is included here. This macro should be placed at the top of your code. Reminder: PROC COPULA requires SAS/ETS®:

```
*------------------------------------- ;
*       H E L P E R   M A C R O
*------------------------------------- ;

* Macro to show intermediate results;
%macro show(TBL=);

  * get descriptive statistics;
  proc means data=work.&TBL._data noprint;
  output out=work.&TBL._stats
    (drop=_type_ _freq_) mean= std= / autoname;
  run;

  * show descriptive statistics;
  title "Descriptive statistics for &TBL";
  proc print data=work.&TBL._stats;
  run;
  title;

  * show correlations;
  ods graphics on;
  title "Correlations for &TBL";
  proc corr data=work.&TBL._data
    plots(maxpoints=none)=matrix(histogram);
  run;
  title;
  ods graphics off;

%mend show;
```