MWSUG 2018 - Paper SB-093

Quality Control for Big Data: How to Utilize High Performance Binning Techniques

Deanna Schreiber-Gregory, Henry M Jackson Foundation, Bethesda, MD
Karlen Bader, Henry M Jackson Foundation, Bethesda, MD

ABSTRACT

It is a well-known fact that the structure of real-world data is rarely complete and straightforward. Keeping this in mind, we must also note that the quality, assumptions, and base state of the data we are working with has a strong influence on the selection and structure of the statistical model chosen for analysis and/or data maintenance. If the structure and assumptions of the raw data are altered too much, then the integrity of the results as a whole are grossly compromised. The purpose of this paper is to provide programmers with a simple technique which will allow the aggregation of data without losing information. This technique will also check for the quality of binned categories in order to improve the performance of statistical modeling techniques. The SAS® high performance analytics procedure, HPBIN, gives us a basic idea of syntax as well as various methods (Bucket, Winsor, Quantile, and Pseudo_Quantile), tips, and details on how to bin variables into comprehensible categories. We will also learn how to check whether these categories are reliable and realistic by reviewing the WOE (Weight of Evidence), and IV (Information Value) for the binned variables. This paper is intended for any level of SAS User interested in quality control and/or SAS high performance analytics procedures.

INTRODUCTION

With the emergence of big data, programmers have been trying to find the most efficient ways to work through their now exponentially larger workloads. In addition to having larger numbers of observations and variables to work with, they also have a wider range of predictor and outcome values that must be considered. When handling these large quantities of datapoints, it is important to note that the existence of this wide variability in values can result in significantly more chaotic information when utilizing data mining applications such as decision trees as well a much higher probability of losing information during the process of splitting. To overcome these issues data reduction can be used as an unsupervised discretization technique for data smoothing methods.

This paper provides detailed information about how the continuous variable can be split into binning categories and how the resulting model can be precisely enhanced. In response to this need, High Performance analytical procedures were created. These procedures are capable of working in either Single-Machine mode or Distribution mode. By using these procedures, we can increase the quality of data and ultimately improve a model's performance.

This paper provides several techniques through which you can bin variables into a number of categories for the model without losing information. PROC HPBIN gives the results of descriptive analytics- missing variables, outliers, generic multiple effects, and non-linear information. This procedure can perform various binning methods – bucket binning, winsorized binning and pseudo-quantile binning with WOE (weight of evidence) and IV (information value) for grouping values, each of which will be covered in detail.

GENERAL SYNTAX

The following syntax is pretty basic for the HPBIN procedure:

```
PROC HPBIN DATA=dataset name <options>;
       CODE FILE = filename;
       ID variables;
       <INPUT> variables <options>;
       PERFORMANCE <performance options>
```

```
        TARGET variable/LEVEL=level ORDER=order;
RUN;
```

The following table provides several options that can be utilized within the HPBIN procedure:

Option	Description
BINS_META=SAS-data-set	*BINS-META dataset specifies the dataset which contains the mapping table information which is generated by PROC HPBIN*
BUCKET | **WINSOR WINSORRATE=**number| **PSEUDO_QUANTILE**	*Specifies the binning method to categorize the data. The default is **BUCKET***
COMPUTEQUANTILE	*Computes the quantile result, which contains the following percentages:* *0% (Min), 1%, 5%, 10%, 25%* *(Q1), 50% (Median), 75% (Q3), 90%, 95%, 99%, and 100% (Max)*
COMPUTESTATS	*Computes the statistic result which is descriptive statistics.*
DATA=SAS-data-set	*Specifies the input SAS data set or database table. For single-machine mode, the input must be a SAS dataset.*
NOPRINT	*Suppresses the generation of ODS outputs.*
NUMBIN=integer	*Specifies the global number of levels for the binning variables which is between 2 and 1000 inclusive. The default number is 6.*
OUTPUT=SAS-data-set	*Creates an output SAS dataset either in single-machine mode or as a database table for the distributed database in distributed mode.* ****In order to avoid duplicate data for large datasets, the variables in the input dataset are not included in the output dataset.*
WOE	*This option computes the Weight of Evidence and Information Values for the input variables.*
WOEADJUST=number	*Specifies the adjustment value for WOE which is 0.0-1.0 inclusive. Default value is 0.5.*
CODE FILE=filename **;**	*This statement generated the score code and is saved in a file and used for scoring purposes.* ****If you specify multiple CODE statements, only the first one is used.*
FREQ variable **;**	*The frequency of occurrence of each observation which is appeared n times where n is the value of variable observation.*
ID variables **;**	*Lists the variables to transform to output dataset which are from input dataset.*
INPUT variables < / option > **;**	*INPUT statement names for one or more continuous variables for binning process.*

	*The options are :**NUMBIN=**integer*
PERFORMANCE < performance-options > **;**	*This statement defines the performance parameters for a multithreaded and distributed computing environment as well as controlling whether the HPBIN procedure executes in either a single-machine or distributed mode.*
TARGET variable / **LEVEL=**level **ORDER=**order **;**	*Defines the target variable to calculate WOE and IV values.* *The values of level can be BINARY or NOMINAL.* *The values of order can be ASCENDING or DESCENDING. The default is DESCENDING.*

Table 1: PROC HPBIN options and descriptions

By using the procedure we could minimize the variance of variable length using the following methods:

- **Bucket Binning:** In this method, the predictor is minimized into a number of categories by displaying the *numbin* statement.
- **Winsorized Binning:** By using this method, the outliers are discarded to obtain smooth binning categories with the *winsorate* option.
- **Quantile Binning:** Quantile binning aims to assign the same number of observations to each bin, if the number of observations is evenly divisible by the number of bins. As a result, each bin should have the same number of observations, provided that there are no tied values at the boundaries of the bins. This method can be accomplished by using the *quantile* option.
- **Pseudo-Quantile Binning Method:** In this method, the predictor values should be categorized into quantile values by using the *pseudo-quantile* option.

DATASET INTRODUCTION

The dataset we will be using in this paper provides detailed customer information from a Portuguese banking institution. The premise behind this data is to predict the rate of success for the bank's current telemarketing strategies.

The following is a screenshot of our input variables and observations:

Obs	age	marital	default	housing	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp_var_rate	cons_price_idx	cons_conf_idx	euribor3m	nr_employed	y
1	29	single	no	no	may	mon	137	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191	no
2	28	single	no	yes	may	tue	49	1	999	0	nonexistent	1.1	93.994	-36.4	4.856	5191	no
3	50	married	no	yes	may	fri	152	4	999	0	nonexistent	1.1	93.994	-36.4	4.864	5191	no
4	48	married	no	yes	jun	fri	384	2	999	0	nonexistent	1.4	94.465	-41.8	4.967	5228.1	no
5	45	married	unknown	no	jun	wed	199	3	999	0	nonexistent	1.4	94.465	-41.8	4.959	5228.1	no
6	25	married	no	yes	jun	wed	121	2	999	0	nonexistent	1.4	94.465	-41.8	4.962	5228.1	no
7	32	divorced	no	no	jul	tue	131	5	999	0	nonexistent	1.4	93.918	-42.7	4.961	5228.1	no
8	25	single	no	no	jul	fri	662	4	999	0	nonexistent	1.4	93.918	-42.7	4.957	5228.1	no
9	47	divorced	no	no	jul	thu	393	3	999	0	nonexistent	1.4	93.918	-42.7	4.962	5228.1	no
10	56	married	unknown	yes	aug	thu	674	1	999	0	nonexistent	1.4	93.444	-36.1	4.968	5228.1	yes
11	60	married	unknown	no	aug	tue	73	1	999	0	nonexistent	1.4	93.444	-36.1	4.966	5228.1	no
12	34	married	no	no	aug	thu	122	1	999	0	nonexistent	1.4	93.444	-36.1	4.964	5228.1	no
13	52	married	no	yes	nov	thu	134	3	999	0	nonexistent	-0.1	93.2	-42	4.076	5195.8	no
14	40	married	no	yes	nov	fri	111	1	999	0	nonexistent	-0.1	93.2	-42	4.021	5195.8	no
15	40	married	no	yes	apr	wed	169	1	999	1	failure	-1.8	93.075	-47.1	1.445	5099.1	no
16	50	married	no	yes	apr	fri	198	1	999	0	nonexistent	-1.8	93.075	-47.1	1.405	5099.1	no
17	40	single	unknown	no	apr	fri	266	2	999	0	nonexistent	-1.8	93.075	-47.1	1.405	5099.1	no
18	29	married	unknown	no	may	mon	332	2	999	0	nonexistent	-1.8	92.893	-46.2	1.299	5099.1	no
19	45	married	no	yes	may	thu	8	5	999	0	nonexistent	-1.8	92.893	-46.2	1.266	5099.1	no
20	35	married	no	yes	oct	wed	608	1	9	2	failure	-1.1	94.601	-49.5	0.959	4963.6	yes

Display 1: Dataset variables and observations

Our main target before applying a machine learning algorithm, would be to precisely define the continuous variables of age and duration to predict whether or not a particular client would subscribe to a term deposit (the outcome variable).

FIRST METHOD: BUCKET BINNING

The first method we will explore is that of bucket binning.

MATHEMATICAL THEORY

For bucket binning, the length of the bucket is:

$$L = \frac{\max(x) - \min(x)}{n}$$

where $\max(x) = maxinum\ value\ of\ a\ variable$

$\min(x) = minimum\ value\ of\ a\ variable$

$n = number\ of\ buckets$

while the split points are:

$$s_k = \min(x) + L * k$$

where $k = 1, 2, \ldots., numbin - 1$

SYNTAX

```
/*Bucket Binning */
proc hpbin data=bank_details numbin=5 bucket computestat
code file="/home/lakshminirmala.b/publications/binning_code.sas";
input duration/numbin=4;  input age;
ods output mapping=result_bucket;
run;
```

RESULTS

Summary Statistics								
Variable	N	N Missing	Mean	Median	Std Dev	Minimum	Maximum	N Bins
age	41188	0	40.0240604	38	10.4212500	17	98	5
duration	41188	0	258.285010	180	259.279249	0	4918	4

Mapping				
Variable	Binned Variable	Range	Frequency	Proportion
age	BIN_age	age < 33.2	13009	0.31584442
		33.2 <= age < 49.4	20124	0.48858891
		49.4 <= age < 65.6	7436	0.18053802
		65.6 <= age < 81.8	520	0.01262504
		81.8 <= age	99	0.00240361
duration	BIN_duration	duration < 1229.5	40725	0.98875886
		1229.5 <= duration < 2459	436	0.01058561
		2459 <= duration < 3688.5	24	0.00058269
		3688.5 <= duration	3	0.00007284

Output 1. Results of bucket binning process

SECOND METHOD: WINSORIZED BINNING

The second method we will explore is that of winsorized binning.

MATHEMATICAL THEORY

In the case of Winsorized binning, the Winsorized statistics are the first to be computed. After the minimum and maximum values are found, the split points are calculated the same way as in bucket binning.

Winsor Minimum

1. Calculate the tail count by $wc = ceil(WinsorRate * n)$, and find the smallest *I*, such that $\sum_{i=Il}^{Ir} c_i \geq wc$.
2. Then, calculate the left tail count by $lwc = \sum_{i=1}^{I} c_i$. When you are finished, find the next I_l, such that $\sum_{i=1}^{Ir} c_i \geq lwc$.
3. Considering these calculations, we can calculate Winsor Minimum as $WinsorMin = min_{I_t}$.

Winsor Maximum

1. Calculate the largest I, such that $\sum_{i=1}^{N} c_i \geq wc$.
2. The right tail count is calculated by $rwc = \sum_{i=1}^{N} c_i$. When you are finished, find the next I_r, such that $\sum_{i=I_r}^{N} c_i \geq rwc$.
3. Considering these calculations, we can calculate Winsor Maximum is $WinsorMax = max_{I_r}$.

The mean is then calculated by using the formula:

$$WinsorMean = \frac{lwc*WinsorMin + \sum_{i=Il}^{Ir} sum_i + rwc*WinsorMax}{n}$$

As stated earlier, the split points are still:

$$s_k = \min(x) + L * k$$

where $k = 1, 2, \ldots., numbin - 1$

SYNTAX

```
/* Winsorized Binning */
proc hpbin data=bank_details winsor  winsorrate=0.02 computestats;
code file="/home/lakshminirmala.b/publications/binning_code.sas";
input duration/numbin=4; input age/numbin=5;
ods output mapping=result_winsorized;
run;
```

RESULTS

Mapping

Variable	Binned Variable	Range	Frequency	Proportion
age	BIN_age	age < 32.2	11176	0.27134117
		32.2 <= age < 39.4	11431	0.27753229
		39.4 <= age < 46.6	7780	0.18888997
		46.6 <= age < 53.8	5887	0.14292998
		53.8 <= age	4914	0.11930659
duration	BIN_duration	duration < 276.5	28647	0.69551811
		276.5 <= duration < 535	8186	0.19874721
		535 <= duration < 793.5	2551	0.06193552
		793.5 <= duration	1804	0.04379916

Winsorized Statistics

Variable	Mean	Std Error Mean	N Left Tail	Percent Left Tail	N Right Tail	Percent Right Tail	DF
age	39.8690152	0.05007594	1068	2.59298825	837	2.03214529	39282
duration	251.339613	1.14483818	879	2.13411673	824	2.00058269	39484

Trimmed Statistics

Variable	Mean	Std Error Mean	N Left Tail	Percent Left Tail	N Right Tail	Percent Right Tail	DF
age	39.8230278	0.05007591	1068	2.59298825	837	2.03214529	39282
duration	239.825402	1.14483758	879	2.13411673	824	2.00058269	39484

Output 2: Results of winsorized binning process

THIRD METHOD: QUANTILE BINNING

The third method we will explore is that of quantile binning. The goal of Quantile binning is to assign the same number of observations to each bin, as long as the number of observations is evenly divisible by the number of bins. The end result will be individual bins with the same number of observations, provided that there are no tied values at the boundaries of the bins. Since PROC HPBIN alwyas assigns observations with the same value to the same bin, quantile binning might create unbalanced bins if any variable has tied values.

For example, if an observation whose value is x is assigned to bin k, then every observation whose value is x is assigned to bin k for this variable, and no observation whose value is x is assigned to the next bin, bin $k + 1$. Therefore, bin k might have more observations than bin $k + 1$, because the tied values at the

boundaries between bin k and bin $k+1$ are all assigned to bin k. That is, tied values at the boundaries between two bins are always assigned to the lower-numbered bin.

MATHEMATICAL THEORY

This procedure computes 0%, 1%, 5% 10%, 25%, 50%, 75%, 90%, 95%, 99%, and 100% percentiles respectively for each binning variable. The mathematical computation of this process is as follows:

Let *m* be the number of nonmissing values for a variable, and let $x_1, x_2, \ldots\ldots, x_m$ represent the ordered values of the variable. Let the t^{th} percentile be *y*, set $p = \frac{t}{100}$, and let *mp* = *j* + *g*, where *j* is the integer part of *mp* and *g* is the fractional part of *mp*. Then the t^{th} percentile is as described:

$$y = \begin{cases} x_j & if\ g = 0 \\ x_{j+1} & if\ g > 0 \end{cases}$$

SYNTAX

```
/*Quantile Binning  with woe and iv values*/
proc hpbin data=bank_details quantile computehist computequantile;
code file="/home/lakshminirmala.b/publications/binning_code.sas";
input duration/numbin=4;  input age/numbin=5;
ods output mapping=result_pseudo;
ods output histogram=histo_pseudo;
run;
```

RESULTS

Mapping

Variable	Binned Variable	Range	Frequency	Proportion
age	BIN_age	age < 31.0049	9330	0.22652229
		31.0049 <= age < 35.0063	7183	0.17439545
		35.0063 <= age < 41.0003	8533	0.20717199
		41.0003 <= age < 49.0031	8087	0.19634360
		49.0031 <= age	8055	0.19556667
duration	BIN_duration	duration < 102.2944	10313	0.25038846
		102.2944 <= duration < 180.4906	10392	0.25230650
		180.4906 <= duration < 319.1782	10197	0.24757211
		319.1782 <= duration	10286	0.24973293

Histogram

Variable	Binned Variable	Bin	Frequency	Range
age	BIN_age	0	0	
		1	9330	age < 31.0049
		2	7183	31.0049 <= age < 35.0063
		3	8533	35.0063 <= age < 41.0003
		4	8087	41.0003 <= age < 49.0031
		5	8055	49.0031 <= age
duration	BIN_duration	0	0	
		1	10313	duration < 102.2944
		2	10392	102.2944 <= duration < 180.4906
		3	10197	180.4906 <= duration < 319.1782
		4	10286	319.1782 <= duration

Output 3: Results of quantile binning process

FOURTH METHOD: PSEUDO-QUANTILE BINNING

The last method we will explore is that of pseudo-quantile binning.

MATHEMATICAL THEORY

This procedure is similar to quantile binning but with more complex sorting algorithms which estimate the quantile method.

For variable *x*, PROC HPBIN uses a simple bucket sorting method to obtain the basic information. Let N = 10,000 be the number of buckets, ranging from $\min(x)$ to $\max(x)$. For each bucket B_i, $i = 1, 2, \ldots, N$, PROC HPBIN keeps following information:

c_i: count of *x* in B_i

$\min_i$: minimum value of *x* in B_i

$\max_i$: maximum value of *x* in B_i

$\sum_i$: sum of *x* in B_i

$\sum 2_i$: sum of x^2 in B_i

To calculate the quantile table, let $P = \{0.00, 0.01, 0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95, 0.99, 1.00\}$. For each $p_k \in P$, $k = 1, 2, \ldots, 11$, find the smallest I_k such that $\sum_{i=1}^{I_k} c_i \geq p_k * n$. Therefore, the quantile value Q_k is obtained,

$$Q_k = \begin{cases} min_{I_k} & \text{if } \sum_{i=1}^{I_k} c_i > p_k * n \\ max_{I_k} & \text{if } \sum_{i=1}^{I_k} c_i = p_k * n \end{cases}$$

where $k = 1, 2, \ldots, 11$.

PROC HPBIN calculates the split points as follows:

For pseudo–quantile binning, let the base count $bc = \text{ceil}(\frac{n}{numbin})$. PROC HPBIN finds those integers $\{I_k | k = 1, 2, \ldots\}$ such that

$$\left(\sum_{i=1}^{I_k} c_i \geq \sum_{i=1}^{I_{k-1}} c_i + bc \quad or \quad \sum_{i=1}^{I_k} c_i \geq \frac{n}{numbin} * k \right)$$

$$and \sum_{i=1}^{I_k} c_i > \sum_{i=1}^{I_{k-1}} c_i$$

$$and \sum_{i=1}^{I_k} c_i < n$$

where *k* is the *k*th split.

The split value is

$$s_k = \min(x) + \frac{\max(x) - \min(x)}{N} * I_k$$

where $k = 1, 2, \ldots$, and $k < numbin$.

The time complexity for pseudo–quantile binning is $C\mathcal{O}(n)/(nodes * cpus)$, where C is a constant that depends on the number of sorting bucket N, *n* is the number of observations, $nodes$ is the number of computer nodes on the appliance, and $cpus$ is the number of CPUs on each node.

SYNTAX

```
/*Pseudo-Quantile Binning  with woe and iv values*/
proc hpbin data=bank_details pseudo_quantile computehist computequantile;
code file="/home/lakshminirmala.b/publications/binning_code.sas";
input duration/numbin=4;  input age/numbin=5;
```

```
ods output mapping=result_pseudo;
ods output histogram=histo_pseudo;
run;
```

RESULTS

Mapping

Variable	Binned Variable	Range	Frequency	Proportion
age	BIN_age	age < 31.0049	9330	0.22652229
		31.0049 <= age < 35.0063	7183	0.17439545
		35.0063 <= age < 41.0003	8533	0.20717199
		41.0003 <= age < 49.0031	8087	0.19634360
		49.0031 <= age	8055	0.19556667
duration	BIN_duration	duration < 102.2944	10313	0.25038846
		102.2944 <= duration < 180.4906	10392	0.25230650
		180.4906 <= duration < 319.1782	10197	0.24757211
		319.1782 <= duration	10286	0.24973293

Histogram

Variable	Binned Variable	Bin	Frequency	Range
age	BIN_age	0	0	
		1	9330	age < 31.0049
		2	7183	31.0049 <= age < 35.0063
		3	8533	35.0063 <= age < 41.0003
		4	8087	41.0003 <= age < 49.0031
		5	8055	49.0031 <= age
duration	BIN_duration	0	0	
		1	10313	duration < 102.2944
		2	10392	102.2944 <= duration < 180.4906
		3	10197	180.4906 <= duration < 319.1782
		4	10286	319.1782 <= duration

Output 3: Results of pseudo-quantile binning process

BINNING AND WEIGHT OF EVIDENCE

PROC HPBIN is also able to compute the weight of evidence and the information value.

Weight of evidence (WOE) is a measure of how much the evidence supports or undermines a hypothesis. WOE measures the relative risk of an attribute for a binning level. The value depends on whether the value of the target variable is a non-event or an event. An attribute's WOE can be defined as follows:

$$WOE_{attribute} = ln\frac{p^{non-event}_{attribute}}{p^{event}_{attribute}} = ln\frac{\frac{N^{attribute}_{non-event}}{N^{total}_{non-event}}}{\frac{N^{attribute}_{event}}{N^{total}_{event}}}$$

The definitions of the quantities in the preceding formula are as follows:

$N^{attribute}_{non-event}$: the number of non-event records that exhibit the attribute

$N^{total}_{non-event}$: the total number of non-event records

$N^{attribute}_{event}$: the number of event records that exhibit the attribute

N^{total}_{event}: the total number of event records

To avoid an undefined WOE, an adjustment factor, *x*, is used:

$$WOE_{attribute} = ln\frac{\frac{N^{attribute}_{non-event}+x}{N^{total}_{non-event}}}{\frac{N^{attribute}_{event}+x}{N^{total}_{event}}}$$

You can use the WOEADJUST= option to specify a value between [0, 1] for *x*. The default value of *x* is 0.5.

The information value (IV) is a weighted sum of the WOE of the characteristic's attributes. In short, the weight is the difference between the conditional probability of an attribute given an event and the conditional probability of that attribute given a non-event. Consider the following formula of IV, where *m* is the number of bins of a variable:

$$IV = \sum_{i=1}^{m} \left(\frac{N_{non-event}^{attribute}}{N_{non-event}^{total}} - \frac{N_{event}^{attribute}}{N_{event}^{total}} \right) * WOE_i$$

An information value can be any real number. Generally speaking, the higher the information value, the more predictive a characteristic is likely to be.

In order to check on how the data is categorized for each model, we use the WOE and IV values. The IV values are listed in order to check the quality of data:

< 0.02: unpredictive
0.02 to 0.1: weak
0.1 to 0.3: medium
0.3 to 0.5: strong
> 0.5: suspicious

We then apply these category boundaries to our next step. By following this process we discover how the variable is binned through predictive analytics:

SYNTAX

```
proc hpbin data=bank_details woe woeadjust=0.08 bins_meta=result_bucket;
target y/level=nominal;
run;
proc hpbin data=bank_details woe bins_meta=result_winsorized;
target y/level=nominal;
run;
proc hpbin data=bank_details woe bins_meta=result_pseudo;
target y/level=nominal;
run;
```

RESULTS

Technique	Results	Conclusion
Bucket Binning	Variable Information Value Variable: Information Value age: 0.15598614 duration: 0.15166427	With this method, our IV values are very low for both age and duration. Given these results, this method would not be a good choice.

Winsorized Binning	Variable Information Value Variable | Information Value age | 0.09284183 duration | 1.17686412	With this approach, we extract outliers for a winsorized splitting process, so the variable duration ends up having a reasonable IV value.
Quantile Binning	Variable Information Value Variable | Information Value age | 0.08113676 duration | 1.39599213	This approach also provides a reasonable result for binning the continuous variable of duration.
Pseudo-Quantile Binning	Variable Information Value Variable | Information Value age | 0.08113676 duration | 1.39599213	This approach also provides a reasonable result for binning the continuous variable of duration.

FEATURES OF HPBIN PROCEDURES

If the dataset is reading from the data source instead of SAS dataset, the input data can be read as parallel also gives the output data in parallel approach.

Data can be run either Single-machine code or Distribution code. If you run this procedure in a distribution method on a cluster of machines that distributes the data and calculated the computations. The Distribution mode can read data could be Client-Data mode, Database mode, HDFS mode, or LASR mode.

CONCLUSION

This paper provides a brief discussion of PROC HPBIN and its capabilities and a comparison to various methods that can be used via PROC HPBIN depending on the goal of the project.

REFERENCES / RECOMMENDED READING

Cody, R. May, 2008. Cody's Data Cleaning Techniques Using SAS: Second Edition. SAS Institute. Inc.

Svolba, G. November, 2006. Data Preparation For Analytics Using SAS. SAS Institute, Inc.

Lin, Aa. Z. 2013. "Variable Reduction in SAS by Using Weight of Evidence and Information Value." *Proceedings of the SAS Global 2013 Conference*, San Fransisco, CA : SAS. Available at http://support.sas.com/resources/papers/proceedings13/095-2013.pdf.

Refaat, M. September, 2006. Data Preparation for Data Mining Using SAS. Elsevier Science.

SAS. "Base SAS® 9.4 Procedures Guide: High-Performance Procedures Second Edition." SAS Documentation. 2013. Available at https://support.sas.com/documentation/cdl/en/prochp/66704/PDF/default/prochp.pdf.

SAS. "SAS/STAT® 14.1 User's Guide: High-Performance Procedures." SAS Documentation. 2015. Available at https://support.sas.com/documentation/cdl/en/stathpug/68163/PDF/default/stathpug.pdf.

CONTACT INFORMATION

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

Deanna Naomi Schreiber-Gregory
Data Analyst II / Research Associate
Henry M Jackson Foundation for the Advancement of Military Medicine
Uniformed Services University / Walter Reed Medical Center
d.n.schreibergregory@gmail.com

Karlen Bader
Research Assistant
Henry M Jackson Foundation for the Advancement of Military Medicine
Uniformed Services University / Walter Reed Medical Center