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 beneficiary information from a public access de-identified health insurance database.

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

DESYNPUF_ID	BENE_BIRTH_DT	BENE_DEATH_DT	BENE_SEX_IDENT_CD	BENE_RACE_CD	BENE_ESRD_IND	SP_STATE_CODE	BENE_COUNTY_CD	BENE_HMO_CVRAGE_TOT_MONS	BENE_SSI_CVRAGE_TOT_MONS	BENE_HI_CVRAGE_TOT_MONS	PLAN_CVRG_MOS_NUM	SP_ALZHDMTA	SP_CHF	SP_CHRNKIDN	SP_CNCR	SP_COPD	SP_DEPRESSN	SP_DIABETES	SP_ISCHMCHT	SP_OSTEOPRS	SP_RA_OA	SP_STRKETIA	MEDREIMB_IP	BENRES_IP	PPPYMT_IP	MEDREIMB_OP	BENRES_OP	PPPYMT_OP	MEDREIMB_CAR	BENRES_CAR	PPPYMT_CAR
00013D2E	19E-07		1	1	0	26	950	12	12	12	12	2	2	2	2	2	2	2	2	2	2	2	0	0	0	50	10	0	0	0	0
00016F74	19E-07		1	1	0	39	230	12	12	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	700	240	0
0001FDD	19E-07		2	1	0	39	280	12	12	0	12	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
00021CA	19E-07		1	5	0	6	290	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
00024B3	19E-07		1	1	0	52	590	12	12	0	0	2	2	2	2	2	2	2	1	2	2	2	0	0	0	30	40	0	220	80	0
0002DAE	19E-07		1	2	0	33	400	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
0002F28	19E-07		1	1	0	39	270	12	12	0	0	2	1	1	2	2	1	1	1	2	1	2	0	0	0	1010	270	0	3330	940	0
0003084	19E-07		1	1	0	24	680	10	10	0	0	2	2	2	2	2	2	2	1	2	2	2	0	0	0	150	160	0	870	340	80
000345A	2E-07		2	1	0	23	810	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
00036A2	19E-07		2	2	0	1	570	12	12	12	12	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
000489E	19E-07		2	1	0	15	140	12	12	0	12	2	2	2	2	2	2	1	1	1	2	2	0	0	0	0	0	0	1980	420	0
00048EF	19E-07		1	1	0	44	230	11	11	8	8	2	2	2	2	2	2	2	2	2	2	2	0	0	0	50	30	0	400	250	30
0004F0A	19E-07		2	1	0	41	30	12	12	0	12	1	2	2	1	1	2	1	1	2	2	2	0	0	0	4580	560	0	3760	1350	0
0005270	19E-07		1	1	0	14	982	12	12	0	5	2	1	2	1	2	1	1	1	2	2	2	14000	1024	0	4240	3130	0	6740	2130	1200
00070B6	19E-07		2	1	0	13	270	12	12	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	100	20	0	930	220	0
0007E57	19E-07		1	1	0	6	140	12	12	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
0007F12	19E-07		2	2	0	34	400	12	12	12	12	1	1	2	2	1	1	1	1	1	1	1	5000	1024	0	5510	1380	0	4980	1170	200
000A005	19E-07		2	2	0	50	160	12	12	12	12	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
000B466	19E-07		2	1	Y	46	170	12	12	0	0	1	1	1	1	1	1	1	1	1	1	2	0	0	0	20650	4960	5000	19030	5260	30
000B97E	19E-07		1	1	0	22	20	12	12	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	1020	290	0
000C748	19E-07		1	2	0	25	350	12	12	0	10	2	2	1	2	2	2	1	1	2	1	2	30000	1024	0	0	0	0	720	290	0
000D6D6	19E-07		1	5	0	32	0	9	0	12	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
000DDD3	19E-07		1	1	0	14	580	12	12	0	0	2	1	2	2	1	2	2	1	1	1	2	0	0	0	320	130	0	1350	370	0
000E374	19E-07		1	1	0	34	350	1	1	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
00108066	2E-07		1	1	0	3	90	12	12	0	12	1	1	2	2	2	1	1	1	1	2	2	0	0	0	1280	120	0	890	280	0
0010D6F	19E-07		2	1	0	54	0	12	0	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
0011714C	19E-07		2	1	0	38	10	12	12	12	12	2	2	2	1	2	2	2	1	2	2	2	0	0	0	70	50	0	350	200	90
0011CB1F	19E-07		1	1	0	20	130	12	12	9	12	1	1	1	1	2	1	1	1	2	2	2	11000	1024	0	0	0	0	4920	1230	80
0012AFE	19E-07		1	1	0	36	870	12	12	0	12	2	2	2	2	2	2	2	2	2	2	2	0	0	0	70	0	0	90	20	0
0013C35	2E-07		2	1	0	5	90	12	12	7	7	1	2	2	2	2	2	1	1	1	2	2	0	0	0	400	110	0	1860	390	0
00139C3	19E-07		2	1	0	10	150	12	12	0	12	2	2	2	1	2	1	1	1	1	2	2	0	0	0	1450	350	0	1200	470	0
0013E139	2E-07		2	1	0	18	170	12	12	0	12	2	2	2	2	2	1	2	1	2	2	2	0	0	0	190	100	0	340	100	0
0014FFD	19E-07		2	1	0	34	640	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0

Display 1: Dataset variables and observations

Our main target before applying a machine learning algorithm, would be to precisely define the continuous variables of reimbursement and claims amounts in order to predict a pre-defined set of demographic and health variables.

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=BenSummary_num numbin=5 bucket computestats;
input MEDREIMB/numbin=4; input BENRES;
ods output mapping=result_bucket;
run;
```

RESULTS

Summary Statistics								
Variable	N	N Missing	Mean	Median	Std Dev	Minimum	Maximum	N Bins
MEDREIMB	3436888	0	1117.96965	670	1416.59748	0	23770	4
BENRES	3436888	0	314.874171	190	393.331737	0	5670	5

Mapping				
Variable	Binned Variable	Range	Frequency	Proportion
MEDREIMB	BIN_MEDREIMB	MEDREIMB < 5942.5	3398484	0.98882594
		5942.5 <= MEDREIMB < 11885	35876	0.01043851
		11885 <= MEDREIMB < 17827.5	2464	0.00071693
		17827.5 <= MEDREIMB	64	0.00001862
BENRES	BIN_BENRES	BENRES < 1134	3276024	0.95319487
		1134 <= BENRES < 2268	151658	0.04412655
		2268 <= BENRES < 3402	8208	0.00238821
		3402 <= BENRES < 4536	940	0.00027350
		4536 <= BENRES	58	0.00001688

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=BenSummary_num winsor  winsorrate=0.02 computestats;
input MEDREIMB/numbin=4; input BENRES/numbin=5;
ods output mapping=result_winsorized;
run;
```

RESULTS

Summary Statistics

Variable	N	N Missing	Mean	Median	Std Dev	Minimum	Maximum	N Bins
MEDREIMB	3436888	0	1117.96965	670	1416.59748	0	23770	4
BENRES	3436888	0	314.874171	190	393.331737	0	5670	5

Mapping

Variable	Binned Variable	Range	Frequency	Proportion
MEDREIMB	BIN_MEDREIMB	MEDREIMB < 1297.5	2340672	0.68104401
		1297.5 <= MEDREIMB < 2585	660860	0.19228442
		2585 <= MEDREIMB < 3872.5	255004	0.07419619
		3872.5 <= MEDREIMB	180352	0.05247538
BENRES	BIN_BENRES	BENRES < 294	2108302	0.61343343
		294 <= BENRES < 578	676698	0.19689265
		578 <= BENRES < 862	333334	0.09698716
		862 <= BENRES < 1146	161972	0.04712752
		1146 <= BENRES	156582	0.04555924

Winsorized Statistics							
Variable	Mean	Std Error Mean	N Left Tail	Percent Left Tail	N Right Tail	Percent Right Tail	DF
MEDREIMB	1087.99022	0.92866950	834362	24.2766712	69004	2.00774654	2.53E6
BENRES	309.176860	0.26745342	895722	26.0620073	70568	2.05325283	2.47E6

Trimmed Statistics							
Variable	Mean	Std Error Mean	N Left Tail	Percent Left Tail	N Right Tail	Percent Right Tail	DF
MEDREIMB	1332.09669	0.92866945	834362	24.2766712	69004	2.00774654	2.53E6
BENRES	385.630030	0.26745341	895722	26.0620073	70568	2.05325283	2.47E6

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 always 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=BenSummary_num quantile computehist computequantile;
input MEDREIMB/numbin=4;  input BENRES/numbin=5;
ods output mapping=result_quant;
ods output histogram=histo_quant;
run;
```

RESULTS

Binning Information	
Method	Quantile Binning
Number of Bins Specified	Between 4 and 5
Number of Variables	2

Mapping				
Variable	Binned Variable	Range	Frequency	Proportion
MEDREIMB	BIN_MEDREIMB	MEDREIMB < 25	868550	0.25271408
		25 <= MEDREIMB < 675	860490	0.25036894
		675 <= MEDREIMB < 1635	854946	0.24875585
		1635 <= MEDREIMB	852902	0.24816113
BENRES	BIN_BENRES	BENRES < 5	895722	0.26062007
		5 <= BENRES < 105	497196	0.14466459
		105 <= BENRES < 285	682876	0.19869021
		285 <= BENRES < 555	675588	0.19656969
		555 <= BENRES	685506	0.19945544

Histogram				
Variable	Binned Variable	Bin	Frequency	Range
MEDREIMB	BIN_MEDREIMB	0	0	
		1	868550	MEDREIMB < 25
		2	860490	25 <= MEDREIMB < 675
		3	854946	675 <= MEDREIMB < 1635
		4	852902	1635 <= MEDREIMB
BENRES	BIN_BENRES	0	0	
		1	895722	BENRES < 5
		2	497196	5 <= BENRES < 105
		3	682876	105 <= BENRES < 285
		4	675588	285 <= BENRES < 555
		5	685506	555 <= BENRES

Quantiles and Extremes		
Variable	Quantile Level	Quantile
MEDREIMB	Max	23770
	.99	6100
	.95	3940
	.90	2940
	.75 Q3	1630
	.50 Median	670
	.25 Q1	20
	.10	0
	.05	0
	.01	0
	Min	0
BENRES	Max	5670
	.99	1690
	.95	1110
	.90	830
	.75 Q3	470
	.50 Median	190
	.25 Q1	0
	.10	0
	.05	0
	.01	0
	Min	0

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=BenSummary_num pseudo_quantile computehist computequantile;
input MEDREIMB/numbin=4;  input BENRES/numbin=5;
ods output mapping=result_pseudo;
ods output histogram=histo_pseudo;
run;
```

RESULTS

Binning Information	
Method	Pseudo-Quantile Binning
Number of Bins Specified	Between 4 and 5
Number of Variables	2

Mapping				
Variable	Binned Variable	Range	Frequency	Proportion
MEDREIMB	BIN_MEDREIMB	MEDREIMB < 21.393	868550	0.25271408
		21.393 <= MEDREIMB < 670.314	860490	0.25036894
		670.314 <= MEDREIMB < 1630.622	854946	0.24875585
		1630.622 <= MEDREIMB	852902	0.24816113
BENRES	BIN_BENRES	BENRES < 0.567	895722	0.26062007
		0.567 <= BENRES < 100.359	497196	0.14466459
		100.359 <= BENRES < 280.098	682876	0.19869021
		280.098 <= BENRES < 550.557	675588	0.19656969
		550.557 <= BENRES	685506	0.19945544

Histogram				
Variable	Binned Variable	Bin	Frequency	Range
MEDREIMB	BIN_MEDREIMB	0	0	
		1	868550	MEDREIMB < 21.393
		2	860490	21.393 <= MEDREIMB < 670.314
		3	854946	670.314 <= MEDREIMB < 1630.622
		4	852902	1630.622 <= MEDREIMB
BENRES	BIN_BENRES	0	0	
		1	895722	BENRES < 0.567
		2	497196	0.567 <= BENRES < 100.359
		3	682876	100.359 <= BENRES < 280.098
		4	675588	280.098 <= BENRES < 550.557
		5	685506	550.557 <= BENRES

Quantiles and Extremes		
Variable	Quantile Level	Quantile
MEDREIMB	Max	23770
	.99	6100
	.95	3940
	.90	2940
	.75 Q3	1630
	.50 Median	670
	.25 Q1	20
	.10	0
	.05	0
	.01	0
	Min	0
BENRES	Max	5670
	.99	1690
	.95	1110
	.90	830
	.75 Q3	470
	.50 Median	190
	.25 Q1	0
	.10	0
	.05	0
	.01	0
	Min	0

Output 4: 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_{event}^{total}: the total number of event records

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

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

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=BenSummary_num woe woeadjust=0.08 bins_meta=result_bucket;
target BENE_RACE_CD/level=nominal;
run;
proc hpbin data=BenSummary_num woe bins_meta=result_winsorized;
target BENE_RACE_CD/level=nominal;
run;
proc hpbin data=BenSummary_num woe bins_meta=result_quant;
target BENE_RACE_CD/level=nominal;
run;
```

```
proc hpbin data=BenSummary_num woe bins_meta=result_pseudo;
target BENE_RACE_CD/level=nominal;
run;
```

RESULTS

Technique	Results
Bucket Binning	Variable Information Value Variable: Information Value MEDREIMB: 0.00057662 BENRES: 0.00292979
Winsorized Binning	Variable Information Value Variable: Information Value MEDREIMB: 0.01450072 BENRES: 0.01905862
Quantile Binning	Variable Information Value Variable: Information Value MEDREIMB: 0.02997109 BENRES: 0.03210373
Pseudo-Quantile Binning	Variable Information Value Variable: Information Value MEDREIMB: 0.02997109 BENRES: 0.03210373

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

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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