

Data Quality Control: Using High Performance Binning to Prevent Information Loss

Deanna Naomi Schreiber-Gregory, Henry M Jackson Foundation, Bethesda, MD;
Lakshmi Nirmala Bavirisetty, South Dakota State University, Cleveland, OH

ABSTRACT

It is a well-known fact that the structure of real-world data is rarely complete and straight-forward. Keeping this in mind, we must also note that the quality, assumptions, and base state of the data we are working with has a very 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, 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	<i>BINS-META dataset specifies the dataset which contains the mapping table information which is generated by PROC HPBIN</i>
BUCKET WINSOR WINSORRATE=number PSEUDO_QUANTILE	<i>Specifies the binning method to categorize the data. The default is BUCKET</i>
COMPUTEQUANTILE	<i>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)</i>
COMPUTESTATS	<i>Computes the statistic result which is descriptive statistics.</i>
DATA=SAS-data-set	<i>Specifies the input SAS data set or database table. For single-machine mode, the input must be a SAS dataset.</i>
NOPRINT	<i>Suppresses the generation of ODS outputs.</i>
NUMBIN=integer	<i>Specifies the global number of levels for the binning variables which is between 2 and 1000 inclusive. The default number is 6.</i>
OUTPUT=SAS-data-set	<i>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.</i>
WOE	<i>This option computes the Weight of Evidence and Information Values for the input variables.</i>
WOEADJUST=number	<i>Specifies the adjustment value for WOE which is 0.0-1.0 inclusive. Default value is 0.5.</i>
CODE FILE=filename ;	<i>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.</i>
FREQ variable ;	<i>The frequency of occurrence of each observation which is appeared n times where n is the value of variable observation.</i>
ID variables ;	<i>Lists the variables to transform to output dataset which are from input dataset.</i>
INPUT variables < / option > ;	<i>INPUT statement names for one or more continuous variables for binning process. The options are :NUMBIN=integer</i>

PERFORMANCE < performance-options > ;	<i>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.</i>
TARGET variable / LEVEL =level ORDER =order ;	<i>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.</i>

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.
- **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	4.864	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	4.078	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	94.801	-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 $\max(x)$ = maximum 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 $k = 1, 2, \dots, \text{numbin} - 1$

SYNTAX

```
/*Bucket Binning */
proc hpbins 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

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

Winsor Maximum

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

The mean is then calculated by using the formula:

$$\text{WinsorMean} = \frac{lwc * \text{WinsorMin} + \sum_{i=l_t}^{l_r} sum_i + rwc * \text{WinsorMax}}{n}$$

As stated earlier, the split points are still:

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

where $k = 1, 2, \dots, \text{numbin} - 1$

SYNTAX

```
/* Winsorized Binning */
proc hpbins 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: PSEUDO-QUANTILE BINNING

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

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, \dots, 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:

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

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	0.19834390
duration	BIN_duration	49.0031 <= age	8055	0.19556667
		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
duration	BIN_duration	5	8055	49.0031 <= age
		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

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 woadjust=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	<table border="1"> <thead> <tr> <th colspan="2">Variable Information Value</th> </tr> <tr> <th>Variable</th> <th>Information Value</th> </tr> </thead> <tbody> <tr> <td>age</td> <td>0.15598614</td> </tr> <tr> <td>duration</td> <td>0.15166427</td> </tr> </tbody> </table>	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.
Variable Information Value										
Variable	Information Value									
age	0.15598614									
duration	0.15166427									

Winsorized Binning	<table border="1"> <thead> <tr> <th colspan="2">Variable Information Value</th> </tr> <tr> <th>Variable</th> <th>Information Value</th> </tr> </thead> <tbody> <tr> <td>age</td> <td>0.09284183</td> </tr> <tr> <td>duration</td> <td>1.17686412</td> </tr> </tbody> </table>	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.
Variable Information Value										
Variable	Information Value									
age	0.09284183									
duration	1.17686412									
Pseudo-Quantile Binning	<table border="1"> <thead> <tr> <th colspan="2">Variable Information Value</th> </tr> <tr> <th>Variable</th> <th>Information Value</th> </tr> </thead> <tbody> <tr> <td>age</td> <td>0.08113676</td> </tr> <tr> <td>duration</td> <td>1.39599213</td> </tr> </tbody> </table>	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.
Variable Information Value										
Variable	Information Value									
age	0.08113676									
duration	1.39599213									

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:

Lakshmi Nirmala Bavirisetty
Masters Graduate
South Dakota State University
plnimmi@gmail.com

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

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.