



Salford Predictive Modeler<sup>®</sup>

## **Introduction to Data Binning**

*This guide provides a detailed introduction  
to the automated binning of data.*

© 2019 Minitab, LLC. All rights reserved.

Minitab®, SPM®, SPM Salford Predictive Modeler®, Salford Predictive Modeler®, Random Forests®, CART®, TreeNet®, MARS®, RuleLearner®, and the Minitab logo are registered trademarks of Minitab, LLC. in the United States and other countries. Additional trademarks of Minitab, LLC. can be found at [www.minitab.com](http://www.minitab.com). All other marks referenced remain the property of their respective owners.

## Data Binning

Data binning, also known variously as bucketing, discretization, categorization, or quantization, is a way to simplify and compress a column of data, by reducing the number of possible values or levels represented in the data. For example, if we have data on the total credit card purchases a bank customer has made in the last month we may prefer to maintain this information in approximate intervals such as: zero, up to \$10, \$10-\$100, \$100-\$500, etc. The categorization can be extremely fine, for example, if we were to simply round card purchases to the nearest dollar, or coarse, as exemplified above. The motivations for binning include:

- ◆ Preparation of data for learning machines that require binned data
- ◆ Protection against minor data errors
- ◆ Protection against outliers
- ◆ Vehicle for handling missing values (missings are assigned a special bin)
- ◆ Simplification, summarization, and reporting

SPM® leverages essential features of the CART® decision tree to deliver a variety of ways to discretize data. SPM discretization can be performed for either continuous or categorical data, and for either numeric or textual data.

## Types of Discretization

SPM includes several ways to process continuous variables into their discrete counterparts. This process is often carried out as a first step toward making them more suitable for the modeling stage.

### Simple or Naïve

Available only for continuous variables, simple discretization groups similar values of the data together, assigning all members of a group or level the same mean value. Thus, if we had a group of records each with credit card purchases of between \$0.01 and \$5.00 we would assign all records the same mean value (possibly something like \$2.50). The groupings are defined automatically following one of two rules:

- ◆ Equal Fraction of Data (e.g. 10% of the data in each of 10 bins)
- ◆ Equal Quantitative Ranges (intervals of equal width, e.g. \$500)

The analyst needs to decide only on the number of intervals to be created.

### Self-Guided via CART

Available only for continuous variables, the discretization is accomplished via CART to determine the optimum groupings and interval widths. The method underlying Self-Guided CART binning is explained here via an example. Let's say that you want to bin a continuous variable X. Behind the scenes we construct a CART regression tree with X as the target variable and an identical copy of X as the sole predictor. Each record will fall into a terminal node and then the X\_FILLED value is the predicted value in the CART tree (remember that in CART regression tree the predicted value is the average of the target variable in a terminal node). The X\_BIN value is a whole number representing simply a bin number. 0 is always used to represent missing values; otherwise, the numbers will simply go from 1 to K, where K is hoped to be close to the number of bins you requested. The bins will be ordered with 1 representing the

lowest values and K the highest values. Note: each bin value corresponds to a terminal node in the CART tree. The key user controls are:

- ◆ **Minimum bin size:** No bin may contain fewer than a specified number of learn sample records
- ◆ **Desired number of bins:** The number of bins the analyst specifies. There is no guarantee that SPM can create a binning with exactly the number of bins desired but SPM attempts to get close.
- ◆ **More or Fewer:** If the desired number of bins are not available should we generate the closest available binning with MORE or FEWER bins?

## Supervised Binning Via CART

Available for both continuous and categorical variables. The method requires a target variable to “supervise” the binning. Typically, this supervisory variable would be the variable serving as the ultimate target for the analysis being undertaken, but it could be any variable selected by the analyst. This style of binning is inspired by credit risk scorecard construction methods. Behind the scenes we run a series of CART models all of which have the same target and include just one of the predictors listed on the model setup dialog (also known as the KEEP list). If the KEEP list contains 200 predictors then we will run 200 CART trees. Each tree is pruned to a size equal to the number of bins requested, or as close to that number as possible if the CART tree sequence does not support an exact match. If an exact match is not possible we use the next smaller or the next larger number of bins following the preference you have indicated when the binning run was set up.

The binning is governed by the same controls listed above for self-guided binning: minimum bin size, desired number of bins, and preference for more or fewer bins when an exact match is not possible. In addition, for supervised binning we offer a FILL option which determines precisely what information is stored in the binned representation of the raw data. FILL is discussed below.

## Binned Variables Created

Normally one would run the data binning process with the goal of saving a new data set containing the transformed versions of the variables. The process will typically save three versions of the input data variables:

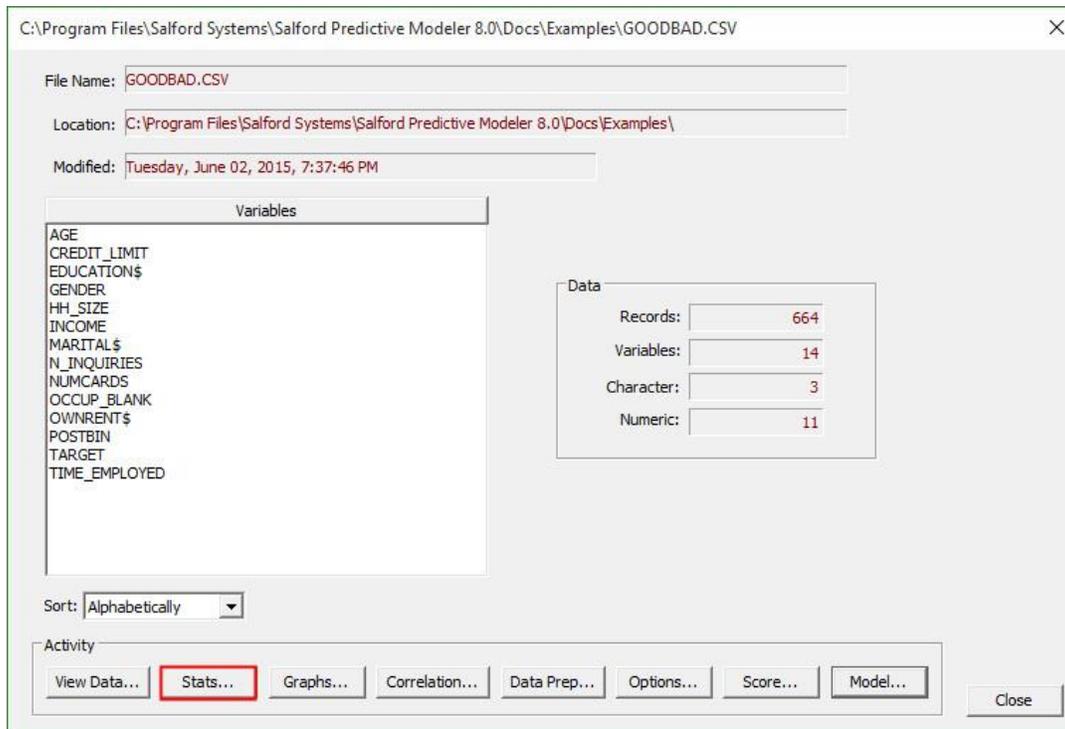
- ◆ **Original variable:** this is just a copy of what you started with
- ◆ **Variable\_BIN:** this is a whole number representing simply a bin number. 0 is always used to represent missing values; otherwise, the numbers will simply go from 1 to K, where K is hoped to be close to the number of bins you requested. The bins will be ordered with 1 representing the lowest values and K the highest values.
- ◆ **Variable\_FILLED:** this is a proper numeric representation of the original variable. By default, each bin will have been filled with the mean value of the variable for the learn sample records falling into that bin.

SPM offers the option of filling the \_FILLED version of the data with something other than the mean value of the records in the bin. First, instead of filling with the means of the original source variable you could substitute instead the mean of any other variable available. Thus, you might elect to fill with the mean of the target variable. Second, you could fill with “weights of evidence” (WOE) values, which are related to Naïve Bayes quantities. For a given variable X we look at each bin to determine how common that bin is for the two groups defined by the dependent variable Y. For example, suppose that among the subset of data for which the target  $Y=1$ , 40% of the learn data fall into a specific bin, while for the  $Y=0$  group the fraction falling into that bin is just 20%, then the WOE is based on the ratio  $40/20$ . A way of looking at this is to say that knowing that a record falls into a bin in question tells us that this would have been far more

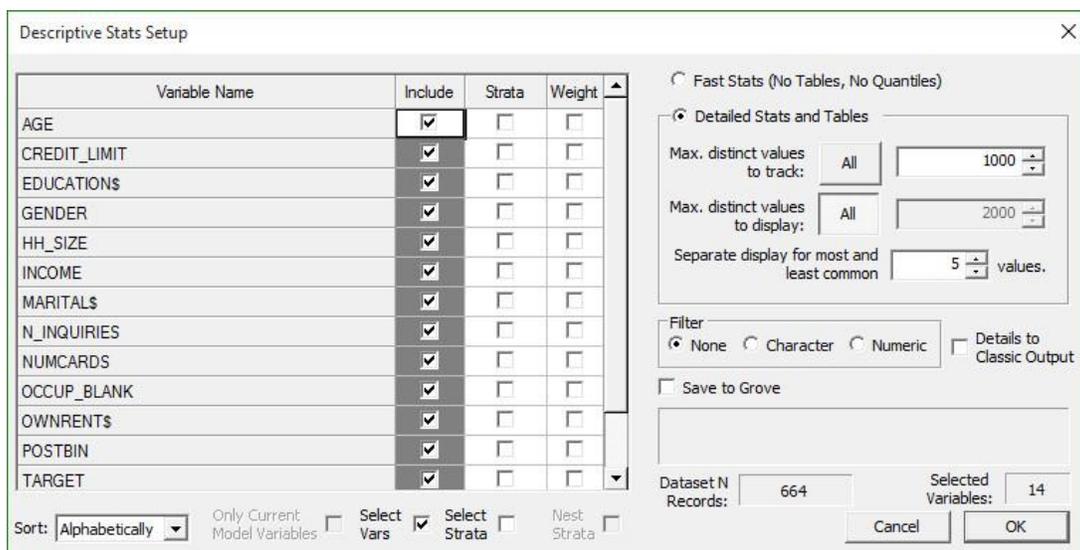
likely for a record with  $Y=1$ . WOE encoding has been common practice in credit risk scorecard development as a data preprocessing step prior to building a predictive model (say, with logistic regression).

## Working Examples:

We start with the **GOODBAD.CSV** data set included with the installation package. This data set is quite small but will serve our purposes.



We also click on the **[Stats]** button to reach the next dialog, where we select “Detailed Stats and Tables” and we make sure that all variables have been selected for reporting (by clicking on the “**Include**” column header and the “**Select Vars**” box below). Then we click on the **[OK]** button.



Variable	N	N Missing	% Missing	N Distinct	Mean	Min	Max	1%	2.5%	5%	10%	20%	25% Q1	40%	50%
1 AGE	580	84	12.65	39	32.00172	9	58	19	21	23	24	26	26	29	
2 CREDIT_LIMIT	664	0	0.00	281	4,966.5738	0	104,622	0	0	0	0	0	0	5,374	
3 EDUCATIONS\$	653	11	1.66	3											
4 GENDER	589	75	11.30	2	-0.6893	-4	6	-4	-4	-4	-4	-4	-4	-4	
5 HH_SIZE	544	120	18.07	7	3.31434	1	7	1	1	1	1	2	2	3	
6 INCOME	664	0	0.00	352	081.37349	0	20,800	0	0	0	2,057	2,500	2,597.5	3,000	
7 MARITALS	663	1	0.15	3											
8 N_INQUIRIES	664	0	0.00	20	2.98494	0	23	0	0	0	0	0	0	1	
9 NUMCARDS	664	0	0.00	10	1.8012	0	9	0	0	0	0	0	0	1	
10 OCCUP_BLANK	664	0	0.00	2	0.05422	0	1	0	0	0	0	0	0	0	
11 OWNRENTS	521	143	21.54	4											
12 POSTBIN	664	0	0.00	5	3.0753	1	5	1	1	1	1	2	2	3	
13 TARGET	664	0	0.00	2	0.30572	0	1	0	0	0	0	0	0	0	
14 TIME_EMPLOYED	664	0	0.00	31	1.74623	0.5	33	0.5	0.5	0.5	0.5	0.5	0.5	0.5	

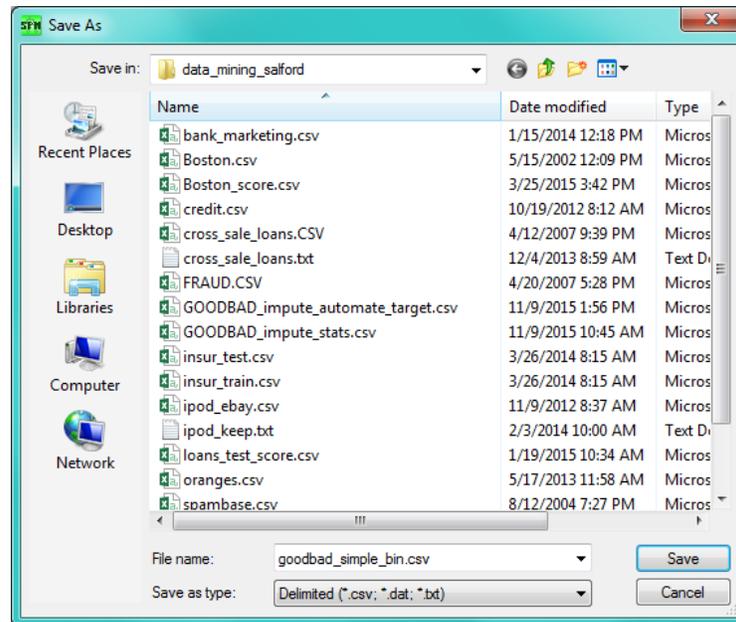
Here we are principally interested in the highlighted “N Distinct” column which displays the number of unique data values found for each variable. The CREDIT\_LIMIT variable is a prime candidate for binning with 281 distinct values, as is INCOME with 352 values. We may or may not be interested in binning variables such as EDUCATION\$ with only a handful of values, but if we want to make use of the FILL and CODING features of SPM Binning we will still need to pass the variables through the binning process.

Next we start with METHOD=SIMPLE binning. All we need to do is to specify the number of bins we are looking for the variables we are most concerned with. (The variables with just a few levels will probably end up with one bin for each original value). There is a literature with some fairly complex methods for determining the “optimal” number; for every day purposes we often see practitioners using between 10 and 20 bins and for now we will stick with a number in this range.

The screenshot shows the 'Model Setup' dialog box with the following settings:

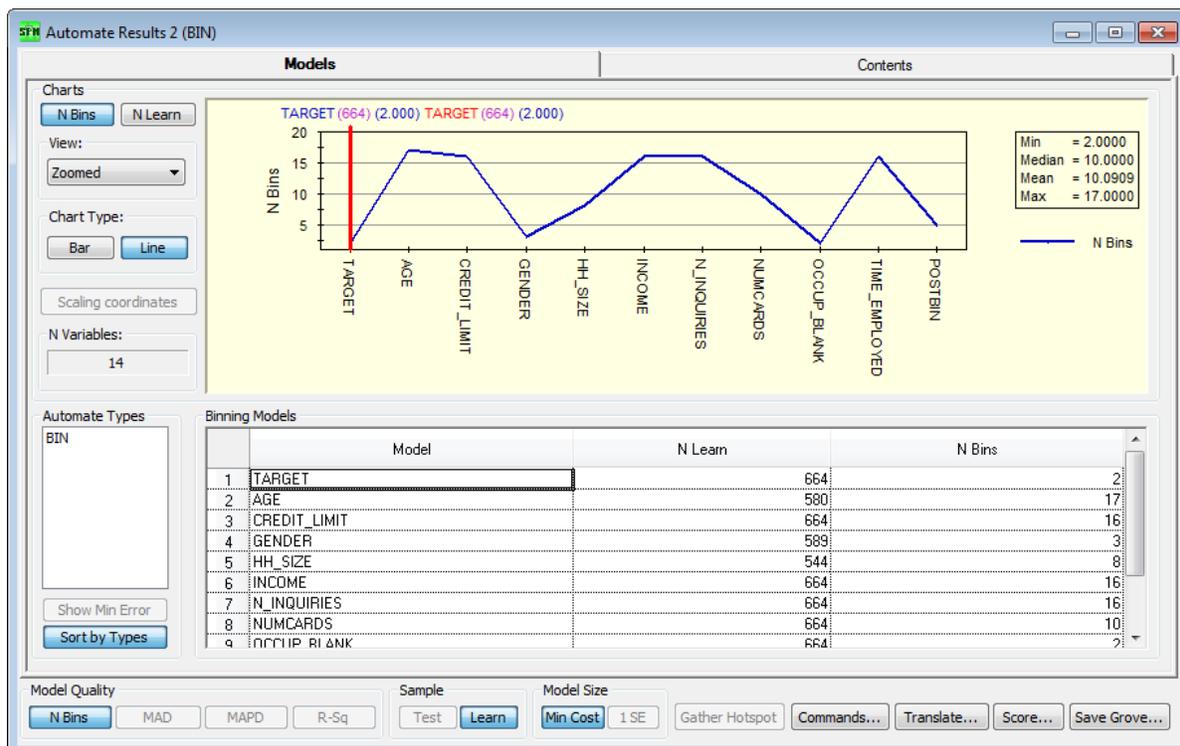
- Binning Method:** Equal Data Fraction (selected), Equal distance/spacing, CART-based
- Supervised by:** Target (selected), Self-guided
- Number of Bins:** Ideal Number of Bins: 16 (checked), If not ideal prefer: more (selected), fewer, Variable specific, Minimum Bin Size: 1
- Generate NAIVE BAYES tables:** Yes (unchecked), No (checked), Report: Short (unchecked), Long (checked)
- Save Output to File:** Save Output As...: C:\data\_mini...\goodbad\_simple\_bin.csv, Suffix for Constructed BIN vars: (empty)
- Coding:** Mean of Predictor (selected), Mean of Fill variable, WOE, Laplace adjustment: 1.0
- Automatic Best Predictor Discovery:** Off (selected), Discover only, Discover and run (Maximum variables for each class: 8)
- Number of Predictors in Model:** 14
- Analysis Engine:** Data Binning

Observe that in the **Binning setup** dialog we have opted for 16 bins (if possible), using the “Equal Data Fraction” policy for constructing the bins (1/16 will put about 6.25% of the data in each bin). For our data set, that will be about 40 records per bin if we use all the data. We also requested the “Long” form of reporting and we specified an output data set to save the results.



We click the **[Save]** button to complete this dialog and return to the **Binning** setup.

Finally, we need to decide whether to use all the data or to use some testing method and it is reasonable to go with no test method. After clicking the **[Start]** button we will arrive at the following summary which reports on the number of bins actually constructed:



Variables with fewer than 16 levels are left alone and those with more are reduced to fewer levels. Observe that AGE, originally with 39 distinct levels is reduced to 17 instead of 16 due to the technicalities of the CART tree sequence generated (the sequence includes an option for 17 but not for 16 terminal nodes – or bins). For more detailed information we need to turn to the classic output window:

Classic Output (Ctrl+Alt+C)

Binning Summary

[Binning Summary](#)  
[TARGET](#)  
[AGE](#) ←  
[CREDIT\\_LIMIT](#)  
[GENDER](#)  
[HH\\_SIZE](#)  
[INCOME](#)  
[N\\_INQUIRIES](#)  
[NUMCARDS](#)  
[OCCUP\\_BLANK](#)  
[TIME\\_EMPLOYED](#)  
[POSTBIN](#)  
[Binning Keep Lists](#)

Score 3

=====

Binning AGE

=====

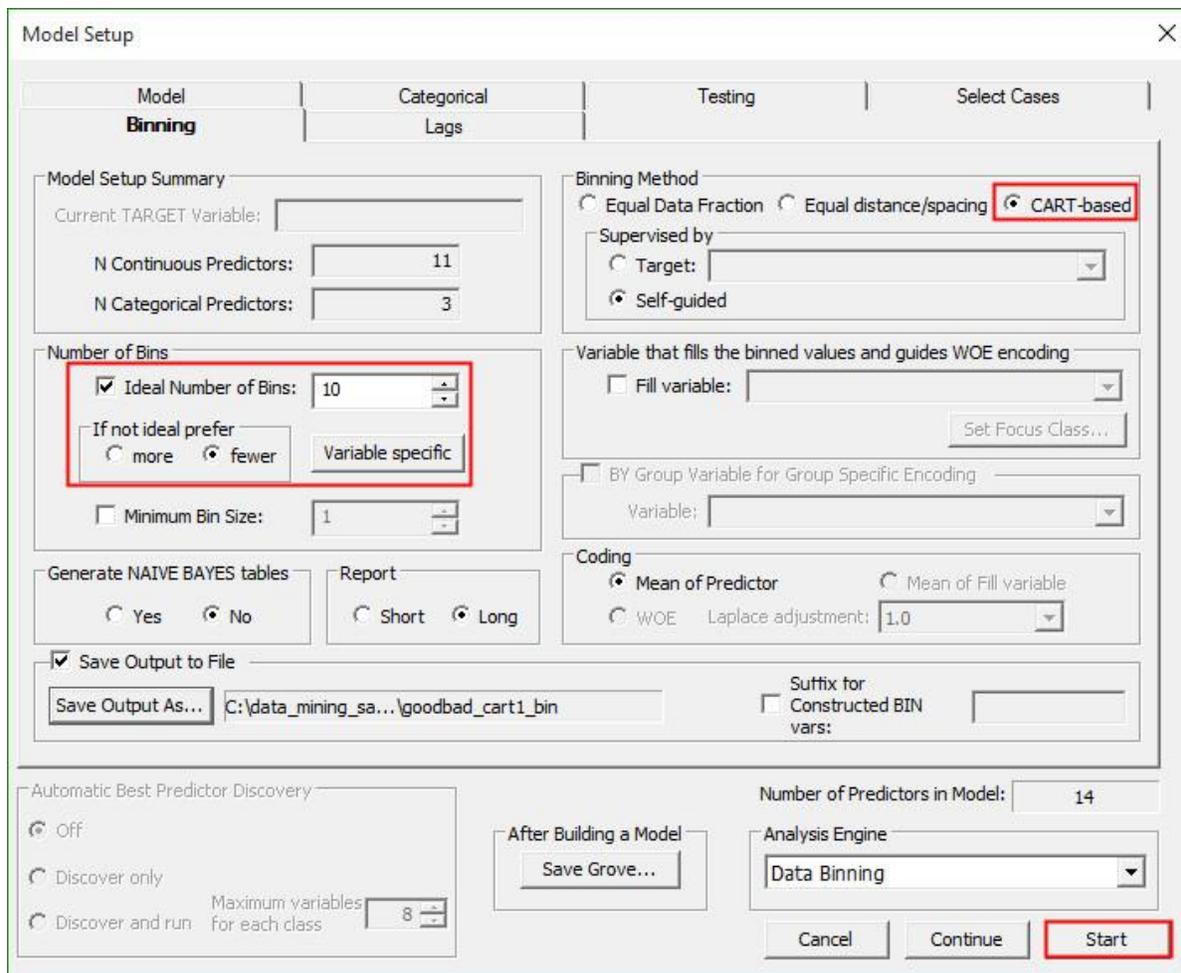
AGE - 17 bins

Bin	N	W	\$	Cut Point	Mean	StdDev	Source
0	84	84.00	.	.			Missing
1	27	27.00	4.66	22.50000	19.81481	2.96177	9.000
2	42	42.00	7.24	24.50000	23.61905	0.49151	23.000
3	38	38.00	6.55	25.50000	25.00000	0.00000	25.000
4	43	43.00	7.41	26.50000	26.00000	0.00000	26.000
5	29	29.00	5.00	27.50000	27.00000	0.00000	27.000
6	30	30.00	5.17	28.50000	28.00000	0.00000	28.000
7	34	34.00	5.86	29.50000	29.00000	0.00000	29.000
8	39	39.00	6.72	30.50000	30.00000	0.00000	30.000
9	35	35.00	6.03	31.50000	31.00000	0.00000	31.000
10	57	57.00	9.83	33.50000	32.59649	0.49496	32.000
11	37	37.00	6.38	35.50000	34.62162	0.49167	34.000
12	26	26.00	5.00	36.50000	36.00000	0.00000	36.000

Each variable has a hyperlink pointing to table with binning details. For the variable AGE we did get the 16 bins requested plus one extra for the missing values; the latter of course will have a size determined by the data. We see for bin number 1 the mean age is almost 20 with a maximum of 22 and a surprise minimum of 9 (probably a data error). The remaining bins are relatively narrow, with many comprising just one age value. A similar table will be created for every variable in the data set, or for every variable checked in the predictor column in the model setup dialog.

Starting with a simple binning can be a very useful way to summarize the data and can be thought of as a tabular histogram. We could readily have requested fewer bins to obtain an even more compact summary.

Now, we move on to CART-based binning, which will not necessarily give us such even bins, but may do a better job of grouping very similar values together.



Note the highlighted areas in the screen shot. We have opted for just 10 bins maximum for any variable having requested 10 or fewer. The binning method is CART and “self-guided” and we will save the resulting output data set to location and with a name of our choosing. (If you click on the [Save Output As..] button the dialog for naming and saving will be displayed).

**Example Binning for CREDIT\_LIMIT:**

```

Binning CREDIT_LIMIT
=====
CREDIT_LIMIT - 10 bins
-----

```

Bin	N	W	\$	Cut Point	Mean	StdDev	Source	Min	Max
1	233	233.00	35.09	3151.50000	61.53219	351.46376		0.00000	2858.00000
2	129	129.00	19.43	10250.00000	6699.54264	1761.66151	3445.00000	10000.00000	
3	96	96.00	14.46	15501.00000	12832.15625	1410.36017	10500.00000	15339.00000	
4	56	56.00	8.43	20344.50000	17989.58929	1384.86866	15663.00000	20074.00000	
5	35	35.00	5.27	26373.00000	23765.42857	1544.01269	20615.00000	26246.00000	
6	27	27.00	4.07	33423.50000	29025.33333	2019.55404	26500.00000	33000.00000	
7	29	29.00	4.37	43376.50000	38106.41379	2930.22010	33847.00000	43000.00000	
8	22	22.00	3.31	60234.50000	48777.40909	4080.07120	43753.00000	57500.00000	
9	22	22.00	3.31	84854.00000	71254.77273	5456.51504	62969.00000	82396.00000	
10	15	15.00	2.26	.	97243.66667	5703.54079	87312.00000	.104622E+06	

Opening the saved data set and opting to view it, you can see the data set details (you could readily open the .CSV data set in any spreadsheet such as Excel or OpenOffice). Here, we used Excel and selected specific columns to show:

	A	H	I	J	Q	R	S
1	CASEID	CREDIT_LIMIT_BIN	CREDIT_LIMIT_FILLED	CREDIT_LIMIT	INCOME_BIN	INCOME_FILLED	INCOME
2	1	4	17989.6	19387	2	2413.1	2399
3	2	1	61.5322	0	9	10175.1	9618
4	3	1	61.5322	0	1	24.26	0
5	4	1	61.5322	0	3	2958.64	2700
6	5	3	12832.2	12000	2	2413.1	2665
7	6	3	12832.2	11516	4	3767.28	4358
8	7	4	17989.6	19000	4	3767.28	3922
9	8	3	12832.2	14000	3	2958.64	2900
10	9	2	6699.54	7000	4	3767.28	3500
11	10	1	61.5322	0	2	2413.1	2500
12	11	1	61.5322	0	2	2413.1	1976
13	12	4	17989.6	17929	5	4967.66	4500
14	13	10	97243.7	100000	10	17386.3	15000
15	14	3	12832.2	12000	4	3767.28	3500
16	15	2	6699.54	8000	4	3767.28	4000
17	16	2	6699.54	5120	3	2958.64	2700
18	17	5	23765.4	25670	6	6147.81	5915
19	18	5	23765.4	25700	4	3767.28	4000

Observe that for every variable binned we now have three variables: the original, a categorical version containing just a bin number, and a “filled” version with a real value corresponding to each bin. Thus, the first line of the dataset has a record with an actual credit limit of 19387, which is assigned to bin number four (in the variable CREDIT\_LIMIT\_BIN) and a common bin credit limit of 17989.6.

At the end of the classic output we produce some handy commands for later use as it can be useful to experiment with alternative versions of your predictors. The saved KEEP lists make it easy to specify models consisting of (a) original variables, (b) categorical bins, or (c) compressed versions of the original predictors reduced to a smaller number of possible values. There may be situations in which the compressed versions are not only more convenient but also perform better.

REM Source variables:

```
KEEP AGE, CREDIT_LIMIT, GENDER, HH_SIZE, INCOME, N_INQUIRIES, NUMCARDS, OCCUP_BLANK,
      TIME_EMPLOYED, POSTBIN
```

REM Bin assignments:

```
KEEP AGE_BIN, CREDIT_LIMIT_BIN, GENDER_BIN, HH_SIZE_BIN, INCOME_BIN,
      N_INQUIRIES_BIN, NUMCARDS_BIN, OCCUP_BLANK_BIN, TIME_EMPLOYED_BIN,
      POSTBIN_BIN
```

REM Binned variables:

```
KEEP AGE_FILLED, CREDIT_LIMIT_FILLED, GENDER_FILLED, HH_SIZE_FILLED, INCOME_FILLED,
      N_INQUIRIES_FILLED, NUMCARDS_FILLED, OCCUP_BLANK_FILLED,
      TIME_EMPLOYED_FILLED, POSTBIN_FILLED
```

To use these variables you will need to open the dataset you just saved within SPM. Immediately after running a binning operation you remain attached to your original unbinned data. This is desirable as you may want to save several different binning versions of your data before moving on to working with your processed versions of the data.

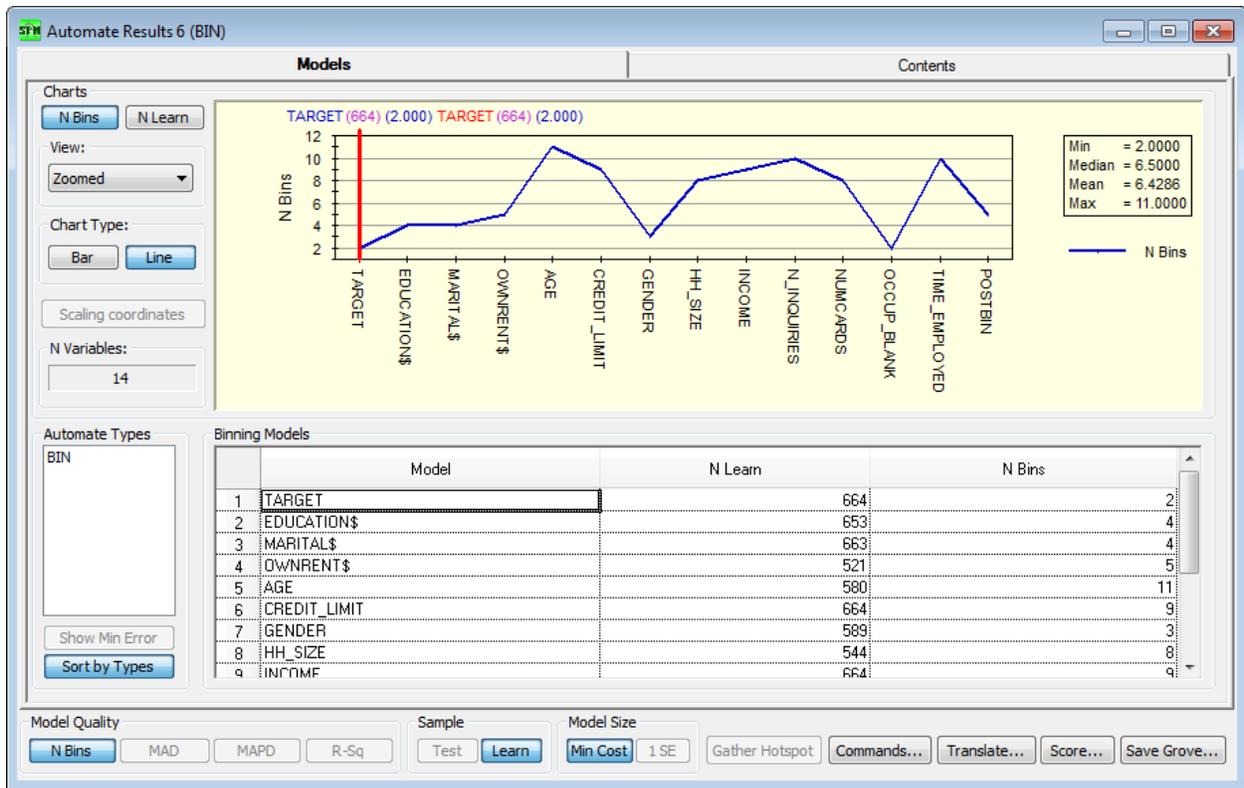
Binning supervised by CART is perhaps the most interesting variation offered in SPM binning. A similar method, but without the benefit of CART, is well known in credit risk scorecard construction where it has been used for decades. The idea is to construct bins in a predictor X while reviewing the ratio of “goods” to “bads” in each bin. The credit risk methodology, now also widely used in insurance and other industries, is an essentially manual, trial and error process, with a goal of producing a transformed version of the predictor. In SPM, the methodology is enhanced by letting CART search for optimal bin boundaries.

Variable Name	Target	Predictor	Categorical	Weight
MARITALS	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
N_INQUIRIES	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NUMCARDS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OCCUP_BLANK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
OWNRENTS	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
POSTBIN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TARGET	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
TIME_EMPLOYED	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

To set up this style of binning (and optionally coding) select the indicated options on the **Binning** tab. Remember that we are now doing supervised binning, via CART, and therefore need a supervising variable. A convenient way to do this is to select a TARGET variable on the **Model Setup** dialog. We also recommend identifying the “Focus” class – the class label for the event we are interested in (usually the rare class). We complete the setup on the **Binning** tab, where we have indicated that we want Weights of Evidence (WOE) encoding in the bins, using the target variable to do the “filling”. When saving several variations of binning we will want the binned versions of the variables to have distinct names. The **Suffix** field allows us to customize the names to keep track of our experiments. Observe that we have also requested **Naïve Bayes** tables.

The screenshot shows the 'Model Setup' dialog box for the 'Binning' method. The 'Binning Method' section is highlighted with a red box, showing 'CART-based' selected. The 'Variable that fills the binned values and guides WOE encoding' section is also highlighted, showing 'Fill variable' set to 'TARGET'. The 'Generate NAIVE BAYES tables' section is highlighted, showing 'No' selected. The 'Start' button at the bottom right is highlighted with a red box.

Clicking the **[Start]** button initiates the process, which generates a fair bit of output. First, we see the overall Summary which reports the number of bins constructed for each variable and the number of records used to construct the binning. Differences in record counts are due entirely to missing values as missing values are automatically assigned to their own bin after the CART model is completed using the good values. Remember, each model contains only one predictor and is of the form  $MODEL\ Y=X$ . With no surrogate splitters we would only muddy the waters if we allowed the records missing on the predictor X to enter the CART Decision Tree analysis.



## Binning Tab Specifics

This section describes the options available on the **Binning** tab and the commands that are consequently executed.

The screenshot shows the 'Model Setup' dialog box with the 'Binning' tab selected. The 'Model Setup Summary' section displays the current TARGET Variable, N Continuous Predictors (11), and N Categorical Predictors (3). The 'Binning Method' section shows 'CART-based' selected, supervised by 'Target'. The 'Number of Bins' section has 'Ideal Number of Bins' set to 10 and 'Minimum Bin Size' set to 10. The 'Variable that fills the binned values and guides WOE encoding' section has 'Fill variable' selected. The 'Coding' section has 'Mean of Predictor' selected. The 'Save Output to File' checkbox is checked, and the output file is 'C:\Users\CHarrison\Desktop\age\_bin.csv'. The 'Analysis Engine' is set to 'Data Binning'.

### Model Setup Summary

Shows a brief summary of the configuration on the **Model** tab. Only variables marked as predictors will be binned.

The screenshot shows the 'Model Setup Summary' dialog box. It displays the current TARGET Variable, N Continuous Predictors (11), and N Categorical Predictors (3).

- ⚠ The target variable specified on the **Model** tab has no effect on binning. It is printed in the **Current TARGET Variable** window for reference purposes only.

## Number of Bins

AUTOMATE BIN IDEALBINS=<n>, PREFER=<MORE|FEWER>

There is no guarantee that SPM can create a binning with exactly the number of bins desired but we attempt to get close. If the desired number of bins is not available, you can control whether we should generate the closest available binning with MORE or FEWER bins.

You cannot create more bins than number of levels of a variable.

AUTOMATE BIN SMALLESTBIN=<n>

Sets a lower bound on the size of a bin. No bin may contain fewer than this number of learn sample records.

## Report

AUTOMATE BIN REPORT=<LONG|SHORT>

If **Long** is selected, additional information about binned variables will be printed to the Classic Output.

## Generate NAÏVE BAYES tables

AUTOMATE BIN NAIVEBAYES=<YES|NO>

Will estimate a Naïve Bayes model if the supervise and fill variables are categorical. The results are printed to the Classic Output.

## Binning Method

Determines the type of discretization.

Sections below provide details on each of the available types.

### Simple or Naïve (Equal Data Fraction or Equal distance/spacing)

```
AUTOMATE BIN METHOD=EQUALFRACTION|EQUALWIDTH
```

Available only for continuous variables, simple discretization groups similar values of the data together and assigns all members of a group the same mean value. Thus, if we had a group of records each with credit card purchases of between \$0.01 and \$5.00 we would assign all records the same mean value (possibly something like \$2.50). The groupings are defined automatically following one of two rules: Equal Fraction of Data rule (e.g. 10% of the data in each of 10 bins).

- ◆ Equal Fraction of Data (e.g. 10% of the data in each of 10 bins)
- ◆ Equal Quantitative Ranges (intervals of equal width, e.g. \$500)

The analyst needs to decide only on the number of intervals to be created.

### Self-Guided via CART (CART-based)

```
AUTOMATE BIN METHOD=CART
```

Available only for continuous variables. The discretization is accomplished via CART to determine the optimum groupings and interval widths. The variable being binned is used as the dependent variable for the model.

### Supervised Binning Via CART (Supervised by)

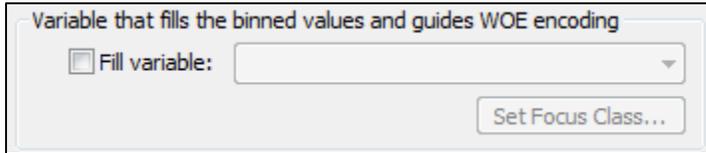
```
AUTOMATE BIN METHOD=CART SUPERVISE=<variable>
```

Available for both continuous and categorical variables. Typically, this supervisory variable would be the variable serving as the ultimate target for the analysis being undertaken, but it could be any variable selected by the analyst. This style of binning is inspired by credit risk scorecard construction methods. Behind the scenes we run a series of CART models, all of which have the same target and include each variable being binned as a single predictor. If we are binning 200 variables then we will run 200 CART trees. Each tree is pruned to a size equal to the number of bins requested, or as close to that number as possible. If an exact match is not possible we use the next smaller or the next larger number of bins following the preference you have indicated when the binning run was set up.

## Fill variable

 AUTOMATE BIN FILL=<variable>

With **Variable that fills the binned values and guides WOE encoding** you can control which variable will be used for the construction of *Variable\_FILLED* in the output dataset. By default, SPM will use the mean value of the learn records of the variable being binned. You could substitute it with the mean of any other variable available.

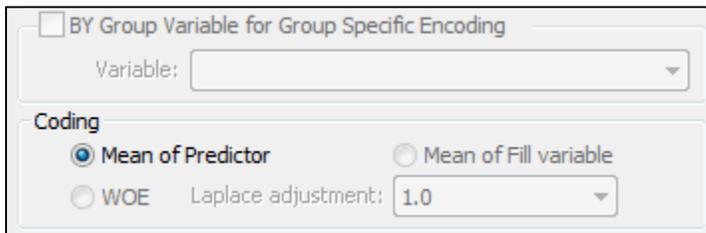


✓ You might elect to fill with the mean of the target variable.

## Coding

 AUTOMATE BIN CODING = <MEAN|WOE>, GROUP=<variable>

This option lets you specify an alternative way to fill *Variable\_FILLED* in the output dataset. Rather than using a fill variable, you could fill with “weights of evidence” (WOE) values, which are related to Naïve Bayes quantities. GROUP identifies a categorical grouping variable, which allows for group-specific fill values and/or WOE coding within bins.



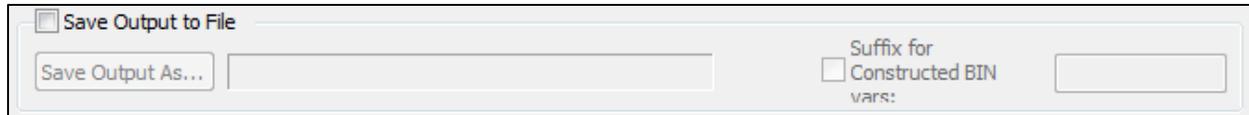
💡 WOE coding requires a binary supervising variable.

For a given variable X, we look at each bin to determine how common that bin is for the two groups defined by the dependent variable Y. For example, suppose that among the subset of data for which the target Y=1, 40% of the learn data fall into a specific bin, while for the Y=0 group the fraction falling into that bin is just 20%, then the WOE is based on the ratio 40/20. A way of looking at this is to say that knowing that a record falls into a bin tells us that this would have been far more likely for a record with Y=1. WOE encoding has been common practice in credit risk scorecard development as a data preprocessing step prior to building a predictive model (say with logistic regression).

## Save Output to File

 AUTOMATE BIN SAVE="filename.ext", SUFFIX=<"text">

Normally, one would run the Data Binning process with the goal of saving a new dataset containing the transformed versions of the variables. You can configure this via **Save Output to File** controls.



The screenshot shows a dialog box titled "Save Output to File". On the left, there is a "Save Output As..." button followed by an empty text input field. On the right, there is a checkbox labeled "Suffix for Constructed BIN vars:" followed by another empty text input field.

The saved dataset will contain three versions of input data variables selected for binning:

- ◆ Original variable: this is just a copy of what you started with.
- ◆ Variable\_BIN: this is a whole number representing simply a bin number. 0 is always used to represent missing values; otherwise, the numbers will simply go from 1 to K, where K is as close as possible to the number of bins you requested. The bins will be ordered with 1 representing the lowest values, and K the highest values.
- ◆ Variable\_FILLED: this is a proper numeric representation of the original variable. By default, each bin will have been filled with the mean value of the variable for the learn sample records falling into that bin. The actual values in this variable are determined by Fill Variable and Coding options.