



THE
POWER
TO KNOW.

**SAS[®] Enterprise Miner[™] and
SAS[®] Text Miner Procedures
Reference for SAS[®] 9.1.3
The TAXONOMY Procedure
(Book Excerpt)**

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., SAS® *Enterprise Miner™* and SAS® *Text Miner Procedures: Reference for SAS® 9.1.3*, Cary, NC: SAS Institute Inc.

SAS® Enterprise Miner™ and SAS® Text Miner Procedures: Reference for SAS 9.1.3

Copyright © 2008 by SAS Institute Inc., Cary, NC, USA.

All rights reserved. Produced in the United States of America.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

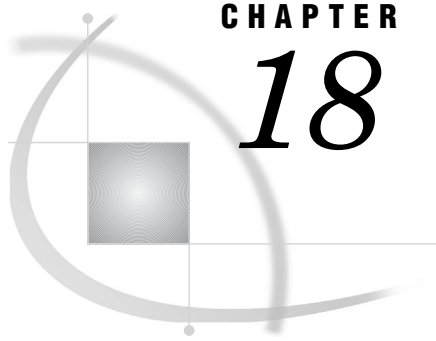
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, October 2008

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at support.sas.com/publishing or call 1-800-727-3228.

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 registered trademarks or trademarks of their respective companies.



CHAPTER

18

The TAXONOMY Procedure (Experimental)

<i>Overview: TAXONOMY Procedure</i>	389
<i>Syntax: TAXONOMY Procedure</i>	390
<i>PROC TAXONOMY Statement</i>	391
<i>CUSTOMER or ID Statement</i>	393
<i>TARGET Statement</i>	393
<i>HIERARCHY Statement</i>	393
<i>Examples: TAXONOMY Procedure</i>	394
<i>Example 1: PROC TAXONOMY</i>	394

Overview: TAXONOMY Procedure

The TAXONOMY procedure performs association rule mining over transaction data in conjunction with item taxonomy. This procedure is an extension of the market basket analysis capability available in Enterprise Miner. It is useful in retail marketing scenarios that involve tens of thousands of distinct items where the items are grouped hierarchically into subcategories, categories, departments, and so on. The hierarchical grouping of the items is called item taxonomy. PROC TAXONOMY uses the taxonomy data and generates rules at multiple levels in the taxonomy known as generalized association rules. This type of rule solves some of the problems in simple market basket analysis as explained below.

One of the problems with simple market basket analysis that is performed over the transaction or point-of-sale data is that significant and potentially useful associations often go undetected.

For example, consider a supermarket that sells different types of breads and a wide selection of wines. Further, assume that a large number of customers buy some type of bread with some type of wine. However, market basket analysis that is performed using, for example, PROC ASSOC might not find any rules linking bread and wine at the transaction level. This is because it computes the support for the combination of specific types of bread and specific types of wine, none of which might be large enough to generate a rule. On the other hand, by combining the item taxonomy with the transaction data, PROC TAXONOMY computes the support for the combination of any type of bread and any type of wine in addition to specific types of bread and specific types of wine. Note that reducing the minimum support is often not a solution to this problem because it can lead to the generation of a very large number of associations, many of them potentially insignificant.

Another problem with simple association rule mining is that often a large number of obvious and uninteresting rules are generated along with the useful ones. In practice, this is considered one of the major drawbacks of association rule mining: if the support is set high, fewer rules are generated but most of them might be obvious and hence useless (for example, "Cereal → Milk"). On the other hand, if the support is set low,

too many rules are generated and the domain experts have to evaluate the generated rules and identify the ones that are useful. Generalized association rule mining provides an ability to generate an interestingness measure called *support lift* for the rules based on of the deviation of a rule's support from its estimated support. The estimated support for a rule is computed based on the support of ancestors of the items in the rule. From an objective perspective, the higher the deviation the more surprise the rule holds. Consequently the rule is likely to be more interesting. To give a different perspective, item taxonomy represents a limited form of domain knowledge that is used by PROC TAXONOMY to generate more interesting rules.

Generalized associations also enable the use of focused interactive mining. For example, one might consider rules only at the higher level of the taxonomy, which will be fewer and more understandable. Subsequently, the rules that are of interest can be drilled down to lower and lower levels of the taxonomy for more specific and actionable rules.

PROC TAXONOMY does not limit rules to items at the same level of the taxonomy. Often how far up the hierarchy an item is generalized depends on its overall frequency in the data. If no item taxonomy is specified, PROC TAXONOMY generates simple association rules.

The following section describes the procedure syntax and usage. An example that clarifies the concepts of generalized association rule mining and the procedure usage is introduced in the subsequent section.

Syntax: TAXONOMY Procedure

PROC TAXONOMY

```

DATA=<libref.>SAS-data-set

<OUT=<libref.>SAS-data-set>
  <OUTFREQ=<libref.>SAS-data-set>
  <OUTRULE=<libref.>SAS-data-set>
  <ITEMS=<max rule size>>
  <SUPPORT=<support count>>
  <PCTSUP=<percent support> (% | PERCENT)>
  <CONF=<minimum confidence> (% | PERCENT)>
  <SUP_LIFT<minimum support lift>>
  <LIFT=<minimum lift>>
  <MINBSKTSZ=<minimum basket size>>
  <MAXBSKTSZ=<maximum basket sizebasket size>>
  <MAXCANDS=<maximum number of candidates>>
  <NORM>
  <MAXRULES=<maximum number of rules>>
  <NUMRULES=<number of top rules>>
  <SORTBY (CONF | SUPPORT | LIFT | SUP_LIFT | SIZE | LHSSZ |
RHSSZ)>;

CUSTOMER|ID <variable name(s)>;

TARGET variable name;
<HIERARCHY
  DATA=><<libref.>SAS-data-set...<libref.>SAS-data-set>><DIMENSION
  DATA=><<libref.>SAS-data-set>><DIMENSION CHILD=<variable
  name>><DIMENSION PARENT=<variable name>>;

```

PROC TAXONOMY Statement

PROC TAXONOMY *<option(s) >*;

This statement invokes the TAXONOMY procedure. The options are described below.

Required Arguments

DATA=*<libref.>SAS-data-set*

specifies the data set that contains the input data, typically from a transactional system (ROLE = TRANSACTION in Enterprise Miner). The data set must contain the variables specified in the CUSTOMER or ID statement and the TARGET statement.

Note: The input data set must be sorted by the variables in the CUSTOMER or ID statement (described below). △

Optional Arguments

OUT=*<libref.>SAS-data-set*

Specifies the data set that contains the following variables: COUNT, SUPPORT, ITEM1, ITEM2, ..., ITEMn., where n is the smaller of the maximum frequent set that is found or the size that is specified in the ITEMS option.

Default: No frequent set data set will be created.

OUTFREQ=*<libref.>SAS-data-set*

Specifies the data set that contains the following variables: ITEM, CODE, COUNT, SUPPORT, and optionally, LEVEL, if a hierarchy has been specified by means of the optional HIERARCHY statement.

Default: No frequency table data set will be created.

OUTRULE=*<libref.>SAS-data-set*

Specifies the data set that contains the following variables: LHS, RHS, COUNT, SUPPORT, CONF, EXP_CONF, LIFT, and RULE. It might also contain LMIN, LMAX, RMIN, and RMAX, if a hierarchy has been specified by means of the optional HIERARCHY statement.

Default: No rules data set will be created.

Note: At least one of OUT, OUTRULE or OUTFREQ must be specified. △

ITEMS=*Integer*

Specifies the maximum size of the frequent sets (consequently, the number of items in the rule) to be mined. The size should be between 1 and 100. If a value smaller or larger is specified, it will be reset to 1 or 100.

Default: If OUT or OUTRULE are specified, ITEMS=2. Otherwise, ITEMS=1.

SUPPORT=*Integer* OR **PCTSUP**=*Real number between 0 and 1 %* | **PERCENT**

These options specify the minimum count for the frequent sets. Support specifies the absolute value of the count whereas PCTSUP specifies it as a percentage of the number of baskets in the input data. Either SUPPORT or PCTSUP can be specified. If both are specified, SUPPORT takes precedence over PCTSUP. If neither is specified, PCTSUP is set to the default value.

Default: 2.0%

CONF=Real number between 0 and 1 % | PERCENT

Specifies the minimum confidence for the rules. This must be a real number between 0 and 100.

Default: 50.0%

LIFT=Real number

This optional parameter specifies the minimum lift value for a rule to be generated.

Default: 1.0

SUP_LIFT=Real number

This optional parameter specifies the minimum support-lift value for a rule to be generated.

Default: 1.0

MINBSKTSZ=Integer

Basket is a grouping of all the target items by the customer ID. This option specifies the minimum size of the "baskets" to be considered valid in the input data set.

Baskets smaller than this size are rejected. This option is useful to filter baskets by size which the user might consider useful.

Default: 1

MAXBSKTSZ=Integer

Specifies the maximum size of the baskets to be considered valid in the input data set. A basket is a grouping of all the target items by the customer ID. This option is useful to filter baskets by size which the user might consider useful.

Default: 1000

MAXCANDS=Integer

Specifies the maximum number of candidate sets to consider. If the candidates to be considered exceed this value, PROC TAXONOMY exits with an error message.

Note: Maximum number of candidates is in general limited by the computational resources. Since it is inversely related to the SUPPORT specification, if the number of candidates exceeds MAXCANDS, the SUPPORT value can be increased to successfully run the procedure without increasing computational resources. Δ

Default: 1,000,000,000

NORM

This flag will result in the target variable normalization, that is, character string values are left-justified, up cased and trailing blanks are trimmed.

Note: The strings are not padded with blanks. This might result in change in the variable length. If NORM is specified, values in the item taxonomy are also normalized. Δ

Default: No normalization

MAXRULES=Number

This optional argument specifies the limit on the number of rules generated. After this limit is reached no more rules are generated.

Default: None

NUMRULES=Number

This optional argument specifies the number of rules to output to the OUTRULE data set. The default is to output all the rules. If the number of generated rules is smaller than the specified number, this argument will have no effect.

Default: None

SORTBY CONF|SUPPORT|LIFT|SUP_LIFT|SIZE|LHSSZ|RHSSZ

Specifies the sort criteria for the rules written to the OUTFRULE data set.

Default: The rules are sorted by SUPPORT.

CUSTOMER or ID Statement

CUSTOMER or ID *<variable list>*;

This statement specifies one or more variables for grouping the target into baskets. The basket is identified by the unique concatenated value of all the variables specified in the statement. If all the variables specified in this statement are not present in the input data set, the procedure ends with an error.

TARGET Statement

TARGET *<variable[(ASCENDING|ASC)|()] [/NOMISS];>*;

This statement specifies a single nominal variable used as the target. If the variable specified is not present in the input data set, the procedure ends with an error.

HIERARCHY Statement

HIERARCHY DATA=*<list of data sets>*;

This statement specifies the item taxonomy in the form of flattened data sets. The data sets contain the values of the child and parent items in the hierarchy at each level beginning with the lowest level. Each data set must contain at least two variables such that the first variable is the child and the second variable is the parent. See the example below for a description of the hierarchy and sample hierarchy data sets.

Note:

- The HIERARCHY statement is optional. If no hierarchy is specified, PROC TAXONOMY will perform simple association analysis with the input data without the hierarchy.
- Each distinct item that appears in the input (transactional) data set must have a parent in the lowest level of the hierarchy (the first data set in the hierarchy statement).
- Multiple parents are not supported in the hierarchy. If multiple parents are specified for a child (at any level), all except the last one are ignored.
- Rugged hierarchy is not supported. That is, at any level if a parent exists for an item, the item must appear in the immediately next level.

Δ

Examples: TAXONOMY Procedure

Example 1: PROC TAXONOMY

The following example, taken from a grocery store context, will serve to clarify the concepts of generalized association rule mining. For simplicity, we assume the lowest level in the taxonomy represents the actual items in the customer baskets. The item taxonomy for this example is shown in Figure 1. The example transaction data is shown in Table 1. Note that in reality the market baskets contain SKUs or some other unique identification code for specific products on sale. The identifying code indicates, among other things the brand, size, type, and so on (for example, "Kraft low fat Swiss cheese, sliced, 8Oz").

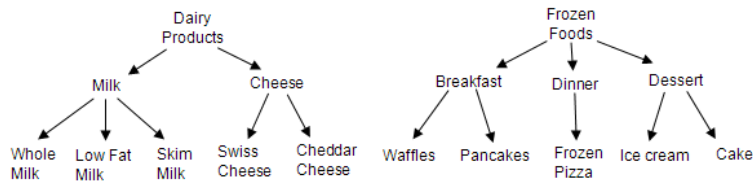


Figure 1. Item Taxonomy

Run the following code:

```

data work.Category;
  length Item $20 Category $20;
  Item='Whole Milk';      Category='Milk';      output;
  Item='Low Fat Milk';    Category='Milk';      output;
  Item='Skim Milk';      Category='Milk';      output;
  Item='Swiss Cheese';    Category='Cheese';    output;
  Item='Cheddar Cheese'; Category='Cheese';    output;
  Item='Waffles';         Category='Breakfast'; output;
  Item='Pancakes';        Category='Breakfast'; output;
  Item='Frozen Pizza';    Category='Dinner';    output;
  Item='Ice Cream';       Category='Dessert';   output;
  Item='Cake';            Category='Dessert';   output;
  Item='Milk';            Category='Dairy Products'; output;
  Item='Cheese';          Category='Dairy Products'; output;
  Item='Breakfast';       Category='Frozen Foods'; output;
  Item='Dinner';          Category='Frozen Foods'; output;
  Item='Dessert';         Category='Frozen Foods'; output;
run;

data work.Dept;
  length Category $20 Department $20;
  Category='Milk';        Department='Dairy Products'; output;
  Category='Cheese';      Department='Dairy Products'; output;
  Category='Breakfast';   Department='Frozen Foods'; output;
  Category='Dinner';      Department='Frozen Foods'; output;

```

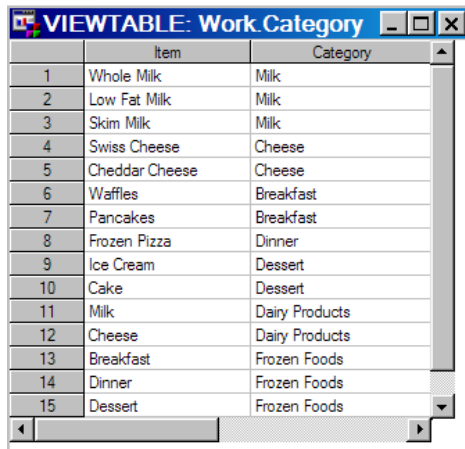
```

    Category='Dessert';    Department='Frozen Foods';  output;
run;
data work.Basket;
  length Customer $ 10 Item $ 20 ;
  retain Customer;
  Customer='Anne';
  Item='Low Fat Milk';    output;
  Item='Cheddar Cheese'; output;
  Item='Cake';           output;
  Item='Frozen Pizza';   output;
  Item='Ice Cream';      output;
  Item='Pancakes';       output;
  Customer='Bob';
  Item='Low Fat Milk';    output;
  Item='Swiss Cheese';   output;
  Item='Frozen Pizza';   output;
  Item='Ice Cream';      output;
  Customer='Chris';
  Item='Skim Milk';      output;
  Item='Swiss Cheese';   output;
  Item='Ice Cream';      output;
  Item='Cake';           output;
run;

```

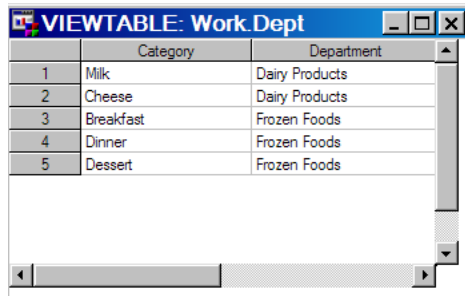
	Customer	Item
1	Anne	Low Fat Milk
2	Anne	Cheddar Cheese
3	Anne	Cake
4	Anne	Frozen Pizza
5	Anne	Ice Cream
6	Anne	Pancakes
7	Bob	Low Fat Milk
8	Bob	Swiss Cheese
9	Bob	Frozen Pizza
10	Bob	Ice Cream
11	Chris	Skim Milk
12	Chris	Swiss Cheese
13	Chris	Ice Cream
14	Chris	Cake

Table 1: Basket data used in the example.



	Item	Category
1	Whole Milk	Milk
2	Low Fat Milk	Milk
3	Skim Milk	Milk
4	Swiss Cheese	Cheese
5	Cheddar Cheese	Cheese
6	Waffles	Breakfast
7	Pancakes	Breakfast
8	Frozen Pizza	Dinner
9	Ice Cream	Dessert
10	Cake	Dessert
11	Milk	Dairy Products
12	Cheese	Dairy Products
13	Breakfast	Frozen Foods
14	Dinner	Frozen Foods
15	Dessert	Frozen Foods

Table 2: Taxonomy data set for the first level



	Category	Department
1	Milk	Dairy Products
2	Cheese	Dairy Products
3	Breakfast	Frozen Foods
4	Dinner	Frozen Foods
5	Dessert	Frozen Foods

Table 3: Taxonomy data set for the second level

The taxonomy data consists of a set of flattened data sets. Each data set represents the child-parent relationship at a specific level in the taxonomy. For example, the Category data set shown in Table 2 tabulates all the relationships between the Items level and the Category level. Similarly, the Dept data set shows the relationships between the Category and the Dept levels. The order of specification of the taxonomy data sets must correspond to the lowest and up to the highest level of the taxonomy. Each taxonomy data set must consist of two variables. The first variable represents the child role and the second variable represents the parent role. Notice that a variable in the parent role in one data set appears as a child variable in the next data set. Further, the name, type, length, and formats of such variables must match.

Every value in the parent variable must appear as a value in the child variable in the previous data set. This condition ensures that every parent has at least one child in the taxonomy and also ensures that the child-parent relationship does not span multiple levels. The taxonomy must be a strict, directed acyclic graph; that is, a child cannot have multiple parents. However, there is no error checking for the last condition. The data sets shown in Tables 2 and 3 corresponding to the taxonomy shown in Figure 1 satisfy all these conditions.

An example procedure statement is shown below:

```
PROC TAXONOMY DATA = Basket OUT = Sets RULEOUT = Rules
ITEMS = 5 SUPPORT = 2 CONF = 80 PERCENT;
CUSTOMER Customer;
TARGET Item;
HIERARCHY DATA = Category Dept;
```

Itemset ID	Set Size	Transact Count	Support(%)	Expected Support(%)	Item 1	Hierarchy Level 1	Item 2	Hierarchy Level 2	Item 3	Hierarchy Level 3	Item 4	Hierarchy Level 4
65	65	3	2	66.6667	66.6667	Dinner	2	Ice Cream	1	Dairy Products	3	
66	66	3	2	66.6667	66.6667	Dinner	2	Dessert	2	Dairy Products	3	
67	67	3	2	66.6667	66.6667	Ice Cream	1	Cake	1	Dairy Products	3	
68	68	4	2	66.6667	66.6667	Milk	2	Cheese	2	Frozen Pizza	1	Ice Cream
69	69	4	2	66.6667	66.6667	Milk	2	Cheese	2	Frozen Pizza	1	Dessert
70	70	4	2	66.6667	66.6667	Milk	2	Cheese	2	Dinner	2	Ice Cream
71	71	4	2	66.6667	66.6667	Milk	2	Cheese	2	Dinner	2	Dessert
72	72	4	2	66.6667	66.6667	Milk	2	Cheese	2	Ice Cream	1	Cake
73	73	4	2	66.6667	44.4444	Low Fat Milk	1	Cheese	2	Frozen Pizza	1	Ice Cream
74	74	4	2	66.6667	44.4444	Low Fat Milk	1	Cheese	2	Frozen Pizza	1	Dessert
75	75	4	2	66.6667	44.4444	Low Fat Milk	1	Cheese	2	Dinner	2	Ice Cream
76	76	4	2	66.6667	44.4444	Low Fat Milk	1	Cheese	2	Dinner	2	Dessert

Table 4: A list of frequent sets

Table 4 shows a partial listing of frequent sets data set generated by PROC TAXONOMY for the example data. The LEVEL numbers refer to the level at which that item appears in the taxonomy. This will be useful for filtering the frequent sets at specific levels. Table 5 shows a partial listing of the OUTRULE data set. LMIN and LMAX refer to the minimum and maximum levels respectively for the LHS of the rule. RMIN and RMAX refer to the RHS.

Rule ID	Size of Rule LHS	Size of Rule RHS	Transact Count	Support(%)	Support Lift	Confidence(%)	Lift	Rule	
1	140	3	1	2	66.6667	0.0000	100.0000	1.0000	Milk & Ice Cream & Cake ==> Cheese
2	129	2	2	2	66.6667	0.0000	100.0000	1.0000	Frozen Pizza & Ice Cream ==> Milk & Cheese
3	141	3	1	2	66.6667	0.0000	100.0000	1.0000	Milk & Cheese & Cake ==> Ice Cream
4	167	1	3	2	66.6667	0.5000	100.0000	1.5000	Low Fat Milk ==> Cheese & Frozen Pizza & Dessert
5	168	3	1	2	66.6667	0.5000	100.0000	1.5000	Cheese & Dinner & Ice Cream ==> Low Fat Milk
6	171	3	1	2	66.6667	0.5000	100.0000	1.0000	Low Fat Milk & Cheese & Dinner ==> Ice Cream
7	172	2	2	2	66.6667	0.5000	100.0000	1.5000	Dinner & Ice Cream ==> Low Fat Milk & Cheese

Table 5. A partial list of rules