



THE  
POWER  
TO KNOW.

**SAS<sup>®</sup> Enterprise Miner<sup>™</sup> and  
SAS<sup>®</sup> Text Miner Procedures  
Reference for SAS<sup>®</sup> 9.1.3  
The DMSPLIT 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).

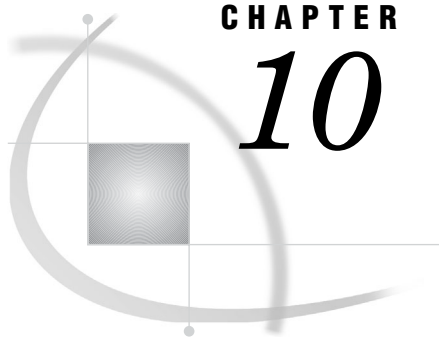
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

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](http://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

## 10

## The DMSPLIT Procedure

---

<i>Overview: DMSPLIT Procedure</i>	217
<i>Syntax: DMSPLIT Procedure</i>	217
<i>PROC DMSPLIT Statement</i>	217
<i>FREQ Statement</i>	219
<i>TARGET Statement</i>	219
<i>VARIABLE Statement</i>	219
<i>WEIGHT Statement</i>	220
<i>Details: DMSPLIT Procedure</i>	220
<i>Missing Values</i>	220
<i>Examples: DMSPLIT Procedure</i>	221
<i>Example 1: Creating a Decision Tree for a Binary Target with the DMSPLIT Procedure</i>	221

---

### Overview: DMSPLIT Procedure

The DMSPLIT procedure performs variable selection using binary variable splits for maximizing the Chi-Square value of a 2 X 2 frequency table. The cutoff threshold is chosen so that the Chi-Square value of the table is maximized.

PROC DMINE and PROC DMSPLIT are underlying procedures for the Variable Selection node.

---

### Syntax: DMSPLIT Procedure

```
PROC DMSPLIT <option(s)>;
  FREQ variable;
  TARGET variable;
  VARIABLE variable-list;
  WEIGHT variable;
```

### PROC DMSPLIT Statement

Invokes the DMSPLIT procedure.

```
PROC DMSPLIT <option(s)>;
```

## Required Arguments

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

Specifies an input data set containing the training data.

**Default:** None.

### **DMDBCAT=<libref.> SAS-catalog**

Identifies an input metadata catalog generated by PROC DMDB. The metadata catalog is associated with a valid data set specified by the DATA= option. The catalog contains important information (for example, the range of variables, number of missing values of each variable, moments of variables) that is used by many other Enterprise Miner procedures that require a DMDB data set. The DMDBCAT= catalog and the DATA= data set must be appropriately related to each other in order to obtain proper results.

**Default:** None.

## Options

### **BINS=*integer***

Specifies the number of categories in which the range of a numeric (interval) variable is divided for splits.

**Range:** Integer > 0

**Default:** 100

### **CHISQ=*number***

Specifies a low bound for the Chi-Square value still eligible for variable splits. The value of CHISQ governs the number of splits that are performed: the higher the value of CHISQ, the fewer splits and passes of the input data will be performed.

**Range:** *number* is a real number > 0

**Default:** 0; even the smallest Chi-Square values are eligible.

### **OUTVARS=<libref.>SAS-data-set**

Specifies an optional output data set containing most of the output table information for the splits.

### **PASSES=*integer***

Specifies an upper bound for the number of passes through the input data set that are used for performing the binary splits.

**Range:** Integer > 0

**Default:** 12

### **PRINT | NOPRINT**

Specifies whether to suppress all output printed in the Output window.

**Default:** NOPRINT

---

## FREQ Statement

**Alias:** FREQUENCY

**Tip:** Specify the FREQ variable in PROC DMDB or PROC DMSPLIT. Specify the FREQ variable in PROC DMDB so that the information is saved in the catalog and so that the variable is automatically used as a FREQ variable in PROC DMSPLIT. This also ensures that the FREQ variable is automatically used by all other Enterprise Miner procedures in the project.

---

**FREQ** *variable*;

### Required Argument

*variable*

Specifies one numeric (interval scaled) FREQUENCY variable.

**Range:** Any integer. A rational value is truncated to the next integer.

**CAUTION:**

If the FREQ variable specified in PROC DMSPLIT differs from that in PROC DMDB, no FREQ variable will be used.  $\Delta$

---

## TARGET Statement

**Tip:** One or more variables might be specified already in PROC DMDB.

---

**TARGET** *variable*;

### Required Argument

*variable*

Specifies the target variable. One variable name can be specified identifying the target (response) variable.

**CAUTION:**

If a target is specified in PROC DMDB, it must not be specified in PROC DMSPLIT.  $\Delta$

---

## VARIABLE Statement

**Alias:** VAR

---

**VARIABLE** *variable-list*;

## Required Argument

### *variable-list*

Specifies all the variables (numeric and categorical, that is, INTERVAL and CLASS) that can be used for independent variables in the prediction or modeling of the target variable.

---

## WEIGHT Statement

**Alias:** WEIGHTS

**Tip:** Specify the WEIGHT variable in PROC DMDB so that the information is saved in the catalog and so that the variable is used automatically as a WEIGHT variable in PROC DMSPLIT.

---

**WEIGHT** *variable*;

## Required Argument

### *variable*

Specifies one numeric (interval scaled) variable that is used to weight the input variables.

### **CAUTION:**

**If the WEIGHT variable is specified in PROC DMDB, it must not be specified in PROC DMSPLIT.  $\triangle$**

---

## Details: DMSPLIT Procedure

---

### Missing Values

For numeric variables, missing values are replaced by the (weighted) mean of the variable. For categorical (CLASS) variables, missing values are treated as an additional category.

## Examples: DMSPLIT Procedure

The following examples were executed on the Windows XP Professional operating system; the version of the SAS System was 9.1.3.

### Example 1: Creating a Decision Tree for a Binary Target with the DMSPLIT Procedure

#### Features:

- Specifying the target and input variables.
- Setting the number of categories in which the range of each interval variable is divided for splits.
- Setting the number of passes the procedure makes to determine the optimum number of splits.
- Setting the chi-square lower bound for evaluating the splits.
- Importing the DMSPLIT tree to the SPLIT procedure.
- Producing summary statistics for the training data.
- Saving the decision tree from within PROC SPLIT.
- Scoring/validating with a test data set.

As a marketing analyst at a catalog company, you want to determine the inputs that best predict whether a customer will make a purchase from your new fall outerwear catalog. The fictitious catalog mailing data set is named SAMPSIO.DMEXA1 (stored in the sample library). The data set contains 1,966 customer cases. The binary target (PURCHASE) contains a formatted value of “Yes” if a purchase was made and a formatted value of “No” if a purchase was not made.

Although there are 48 input variables available for predicting the target, only 17 inputs are used to construct the tree. Note that AMOUNT is an interval target and ACCTNUM is an id variable; these variables are not suitable model inputs.

To demonstrate how to score a data set, a sample of customers is selected from the SAMPSIO.DMEXA1 training data set.

#### Program

Before you analyze the data using the DMSPLIT procedure, you must create a DMDB catalog that contains the metadata for the input data set being analyzed. For more information about how to do this, see “Example 1: Getting Started with the DMDB Procedure” in the DMDB procedure documentation.

```
proc dmdb batch data=sampsio.dmexa1 dmdbcat=catexa1;
  id acctnum;
  var amount income homeval frequent recency age
      domestic apparel;
  class purchase(desc) marital ntitle gender telind
      origin job statecod numcars edlevel;
run;
```

The PROC DMSPLIT statement invokes the procedure. The DATA= option identifies the DMDB encoded training data set that is used to fit the model. The DMDBCAT= option identifies the DMDB training data catalog.

```
proc dmsplit data=sampsio.dmexal dmdbcat=catexal
```

The BINS= option specifies the number of categories in which the range of each interval variable is divided for splits.

```
bins=30
```

The CHISQ= option specifies a minimum bound for the Chi-Square value that is still eligible for making a variable split. The value of CHISQ governs the number of splits that are performed. As you increase the CHISQ value, the procedure performs fewer splits and passes through the input data.

```
chisq=2.00
```

The PASSES = option specifies an upper bound for the number of passes that are made through the data.

```
passes=20
```

The OUTVARS = option creates a data set containing splitting information.

```
outvars=vout;
```

The VAR statement specifies the numeric and categorical inputs (independent variables).

```
var amount income homeval frequent recency age
    domestic apparel marital ntitle gender telind origin
    job statecod numcars edlevel;
```

The TARGET statement defines the target (response) variable.

```
target purchase;
title 'DMSPLIT: Binary Target';
run;
```

PROC PRINT creates a partial report of the OUTVARS= data set.

```
proc print data=vout(obs=20);
title2 'OUTVARS= Summary Data';
run;
```

The PROC SPLIT statement invokes the procedure. The INDMSPLIT option specifies to read the tree created from PROC DMSPLIT. The DMSPLIT tree information is stored in the DMDB catalog.

```
title 'Import and Save Tree from DMSPLIT';
```

```
proc split dmdbcat=catexal indmsplit
```

The OUTTREE= option names the data set that contains tree information.

```
    outmatrix=trtree
```

The OUTLEAF= option names the data set that contains statistics for each leaf node.

```
    outleaf=leafdata
```

The OUTTREE= option specifies the output data set that describes the tree. The OUTTREE data set can be used as input in subsequent executions of PROC SPLIT.

```
    outtree=savetree;
run;
```

PROC PRINT creates a report of the training statistics.

```
proc print data=trtree label;
    title2 'Training Statistics';
run;
```

PROC PRINT creates a partial report of the leaf statistics for the training data.

```
proc print data=leafdata(obs=10) label;
    title2 'Leaf Statistics';
run;
```

The DATA step creates a fictitious score data set.

```
data testexal(drop=ran);
    set sampsio.dmexal;
    ran=ranuni(3333);
    if ran lt 0.08;
    title 'Create Fictitious Score Data Set';
run;
```

The INTREE = option reads the tree that was saved from the previous PROC SPLIT step.

```
proc split intree=savetree;
```

The SCORE statement scores the DATA= data set. The OUTFIT= option names the output data set containing fit statistics. The OUT= option names the output data set that contains tree statistics for the scored data set. Typically you would want to score a truly mutually exclusive data set that might or might not contain the target values (the WORK.TESTEXA1 data set is a random subset of the SAMPSIO.DMEXA1 training data set).

```
score data=testexal nodmdb
```

```

outfit=tfit out=tout;

title 'Input Tree and Score Test Data';

```

PROC PRINT creates a report of the fit statistics for the scored data set.

```

proc print data=tfit label;
  title2 'Fit Statistics for the Scored Data Set';
run;

```

PROC FREQ creates a misclassification table for the scored data set. The F\_PURCHASE variable is the actual target value for each customer and the I\_PURCHASE variable is the target value into which the customer is classified.

```

proc freq data=tout;
  tables f_purchase*i_purchase;
  title2 'Scored Data';
  title3 'Misclassification Table';
run;

```

PROC PRINT creates a partial report of selected variables from the OUT= score information data set.

```

proc print data=tout(obs=10) label;
  var _node_ _leaf_ i_purchase u_purchase f_purchase
      p_purchaseyes p_purchaseno r_purchaseyes
      r_purchaseno;
  title2 'Score Summary Data';
run;

```

## PROC DMSPLIT Output

### Partial Listing of the Splitting Table

The splitting table contains the following information for each split:

- node number
- parent node
- chi-square value for the split
- splitting variable
- the average of the splitting variable if it is an interval input, or the number of levels if the splitting variable is non-interval.

DMSPLIT: Binary Target

The DMSPLIT Procedure

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---
1	0	92.337985	FREQUENT	2.363333	

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---	
2	1	29.482142	STATECOD	.	23	31
3	1	30.803968	DOMESTIC	3.200000		

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---	
4	2	11.582262	JOB	.	10	4
5	2	13.458240	STATECOD	.	20	11
6	3	30.883260	STATECOD	.	35	18
7	3	29.787117	STATECOD	.	21	24

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---	
8	4	7.279738	JOB	.	4	6
9	4	9.904762	STATECOD	.	2	9
10	5	7.028403	HOMEVAL	40000		
11	5	6.555114	HOMEVAL	240000		
12	6	8.285939	STATECOD	.	13	22
13	6	8.827120	JOB	.	5	9
14	7	7.158556	STATECOD	.	17	4
15	7	11.848364	APPAREL	1.666667		

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---	
17	8	6.059502	HOMEVAL	20000		
18	9	3.000000	INCOME	29040		
20	10	11.115789	STATECOD	.	5	11

DMSPLIT: Binary Target

The DMSPLIT Procedure

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---	
21	10	5.028481	JOB	.	10	3
22	11	6.422236	FREQUENT	2.287556		
24	12	6.159905	JOB	.	5	3
25	12	9.146654	EDLEVEL	.	2	2
27	13	6.575469	AMOUNT	1328.500000		

29	14	7.977941	INCOME	14520		
30	15	10.104654	STATECOD	.	13	8
31	15	15.128776	JOB	.	8	2

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---	
32	17	11.000000	RECENCY	789.366667		
33	17	4.562935	STATECOD	.	4	18
36	20	6.114656	AGE	29.700000		
37	20	8.333333	DOMESTIC	0.800000		
38	21	4.344165	TELIND	.	1	1
40	22	6.402489	HOMEVAL	96000		
41	22	4.000000	NTITLE	.	2	1
42	24	6.767263	TELIND	.	1	1
43	24	2.222222	AGE	28.066667		
44	25	5.400777	MARITAL	.	1	1
45	25	13.884196	INCOME	21780		
46	27	9.529412	APPAREL	1.666667		
47	27	15.079365	AGE	34.600000		
49	29	7.082388	FREQUENT	9.789556		
50	30	6.850370	APPAREL	0.055556		
51	30	5.204082	FREQUENT	6.606889		
52	31	7.296558	STATECOD	.	12	9

History of Node Splits

Node	Parent	ChiSqu	Split	Value	---Levels---	
56	33	4.800000	RECENCY	139.300000		
57	33	5.427649	FREQUENT	1.605556		
58	36	6.000000	AMOUNT	442.833333		
59	36	8.775000	HOMEVAL	29333		
60	37	9.183673	AMOUNT	885.666667		

**Effect Summary Table**

The Effect Summary table lists the node in which the effect was first split and the total number of times a split occurred for the effect.

The DMSPLIT Procedure

Effect Summary

Effect	Node 1st Split	Total Times Split
FREQUENT	1	50
STATECOD	2	65
DOMESTIC	3	15
JOB	4	34
HOMEVAL	10	26



- 959 of the 967 non-buyers were correctly classified; only 8 non-buyers were incorrectly classified as buyers.

The values in the STAT column enable you to identify the rows that pertain to counts (N), row and column percentages (Row% and Col%), and overall percentages (%).

Import and Save Tree from DMSPLT  
Training Statistics

Obs	STAT	PURCHASE	==>		TOTAL
			YES	==> NO	
1	N	YES	992	7	999
2	N	NO	8	959	967
3	N	SUM	1000	966	1966
4	Row%	YES	99	1	100
5	Row%	NO	1	99	100
6	Row%	SUM	51	49	100
7	Col%	YES	99	1	51
8	Col%	NO	1	99	49
9	Col%	SUM	100	100	100
10	%	YES	50	0	51
11	%	NO	0	49	49
12	%	SUM	51	49	100

### Partial PROC PRINT Report of the Leaf Statistics (OUTLEAF=)

The leaf report contains the following information:

- Leaf identification number
- Number of customers in each leaf
- Percentages of the binary target values in each leaf.

Notice the purity of the leaf nodes.

Import and Save Tree from DMSPLT  
Leaf Statistics

Obs	Node	Leaf	N	N *		
				PRIORS	% YES	% NO
1	16	1	9	9	100.00	0.00
2	54	2	10	10	100.00	0.00
3	55	3	1	1	0.00	100.00
4	142	4	2	2	100.00	0.00
5	143	5	2	2	0.00	100.00
6	89	6	8	8	100.00	0.00
7	342	7	2	2	100.00	0.00
8	343	8	8	8	0.00	100.00
9	426	9	12	12	100.00	0.00
10	427	10	1	1	0.00	100.000

### PROC PRINT Report of the Score Fit Statistics (OUTFIT=)

The misclassification rate for the scored data set is almost zero. You can compare the maximum absolute error, sum of squared errors, average squared error, and root average squared error from this tree with other candidate trees (models). Small values for these test statistics are preferred.

Input Tree and Score Test Data  
Fit Statistics for the Scored Data Set

Obs	Test: Sum of Frequencies	Test: Sum of Weights Times Freqs	Test: Misclassification Rate	Test: Maximum Absolute Error
1	150	300	.006666667	0.72727

Obs	Test: Sum of Squared Errors	Test: Average Squared Error	Test: Root Average Squared Error	Test: Divisor for TASE
1	1.52661	.005088705	0.071335	300

**Misclassification Table for the Scored Data Set (OUT=)**

Only one customer in the test data set was incorrectly classified. Ideally, you should use a mutually exclusive test data set for validating the tree.

Input Tree and Score Test Data  
Scored Data  
Misclassification Table

The FREQ Procedure

Table of F\_PURCHASE by I\_PURCHASE

F\_PURCHASE(From: PURCHASE)  
I\_PURCHASE(Into: PURCHASE)

Frequency	Percent	Row Pct	Col Pct	NO	YES	Total
NO	64	1	65	42.67	0.67	43.33
	98.46	1.54		100.00	1.16	
YES	0	85	85	0.00	56.67	56.67
	0.00	100.00		0.00	98.84	
Total	64	86	150	42.67	57.33	100.00

**Partial PROC PRINT Report of the Score Summary Data Set**

Input Tree and Score Test Data

Score Summary Data

Obs	Node	Leaf	Unnormalized		
			Into: PURCHASE	Into: PURCHASE	From: PURCHASE
1	891	107	YES	1	NO
2	372	203	YES	1	YES
3	661	320	YES	1	YES
4	469	229	NO	0	NO
5	661	320	YES	1	YES
6	558	214	NO	0	NO
7	347	18	YES	1	YES
8	310	338	YES	1	YES
9	266	13	YES	1	YES
10	53	457	NO	0	NO

Obs	Predicted: PURCHASE=	Predicted: PURCHASE=	Residual: PURCHASE=	Residual: PURCHASE=
	Yes	No	Yes	No
1	0.72727	0.27273	-0.72727	0.72727
2	1.00000	0.00000	0.00000	0.00000
3	1.00000	0.00000	0.00000	0.00000
4	0.00000	1.00000	0.00000	0.00000
5	1.00000	0.00000	0.00000	0.00000
6	0.00000	1.00000	0.00000	0.00000
7	1.00000	0.00000	0.00000	0.00000
8	1.00000	0.00000	0.00000	0.00000
9	1.00000	0.00000	0.00000	0.00000
10	0.00000	1.00000	0.00000	0.00000