



THE
POWER
TO KNOW.

**SAS[®] Enterprise Miner[™] and
SAS[®] Text Miner Procedures
Reference for SAS[®] 9.1.3
The SEQUENCE 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).

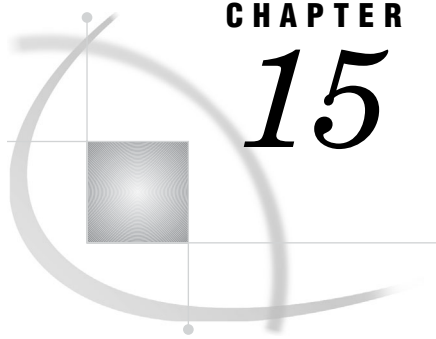
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

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

15

The SEQUENCE Procedure

Overview: <i>SEQUENCE</i> Procedure	331
Syntax: <i>SEQUENCE</i> Procedure	331
<i>PROC SEQUENCE</i> Statement	332
<i>CUSTOMER</i> Statement	333
<i>TARGET</i> Statement	333
<i>VISIT</i> Statement	334
Details: <i>SEQUENCE</i> Procedure	334
<i>SAME</i> and <i>WINDOW</i> Parameters	334
Examples: <i>SEQUENCE</i> Procedure	335
Example 1: Performing a Simple 2-Item Sequence Discovery	335
Example 2: Specifying the Maximum Number of Item Events and Setting the Lower Timing Limit	340
References	342

Overview: SEQUENCE Procedure

The SEQUENCE procedure enables you to perform sequence discovery. Sequence discovery goes one step further than association discovery by taking into account the ordering or timing of the relationship among items, for example, “Of those customers who purchase a new computer, 25% of them will purchase a laser printer in the next quarter”. To perform a sequence discovery, you must first run the ASSOCIATION procedure to create and output the data set of the assembled items.

PROC SEQ produces rules similar to PROC RULEGEN, however the rules additionally imply an element of timing. A rule $A \Rightarrow B$ implies that event B occurred ‘after’ event A occurred. The visit or sequence variable is used for timing comparison. The sequence variable can have any numeric value, including date or time values. Transactions with missing sequence values are ignored entirely during the sequence computation.

In order to determine the timing element, SEQUENCE uses a sequence variable or time-stamp that enables you to measure the time span from observation to observation. This procedure is useful for businesses such as banks or mail-order houses.

Syntax: SEQUENCE Procedure

```
PROC SEQUENCE <option(s)>;
    CUSTOMER variable(s);
```

TARGET *variable*;
VISIT *variable* /<*visit-option(s)*>;

PROC SEQUENCE Statement

Invokes the SEQUENCE procedure.

PROC SEQUENCE <*option(s)*>;

Required Arguments

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

Specifies the SAS data set that was output from PROC ASSOC and which is also one of the inputs to PROC SEQ.

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

Identifies the input data source. To perform sequence discovery, the input data set must have a separate observation for each product purchased by each customer. You must also assign the ID model role to a variable, the TARGET model role to another variable, and SEQUENCE role to the timestamp or ordering variable.

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

Identifies the metadata catalog associated with the input data source.

Options

NITEMS=*integer*

Specifies the maximum number of events for which rules, or chains, are generated. If you request more than 2-event chains, (*integer*- 2) additional passes through the input file are required.

Default: 2

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

Specifies the output data set to which the rules are written. The output data set has the following variables: RULE, COUNT, SUPPORT, CONF, ISET1, ISET2, ..., ISETn.

RULE

Contains the rule text, for example, A & B ==> C & D

COUNT

Contains the number of the transactions meeting the rule.

SUPPORT

Contains the percent of support, that is the percent of the total number of transactions that qualify for the rule.

Definition: SUPPORT= COUNT/total, where *total* is the total number of transactions in the data set. The support level is an integer that represents how frequently the combination occurs in the database.

CONF

Contains the percent of confidence.

Definition: CONF= COUNT/lhs_count where *lhs_count* is the number of transactions satisfying the left side of the rule.

ISET1, ISET2,..., ISETn

Contain, in order, the events that form the event chain. PROC SEQUENCE can detect multiple events occurring at the same time and can report them as rules of the type A & B ==> C & D. This means that events A and B occurred at the same time, followed by C and D, which occurred simultaneously afterwards.

SUPPORT=*integer*

Specifies the minimum number of transactions that must be considered in order for a rule to be accepted. Rules that do not meet the support level are rejected. Note that this support is in addition to the SUPPORT parameter on proc assoc call, which in a sense defines a superset of associations forming potential sequences. The final reported sequences are limited by the SUPPORT option on proc sequence call.

Default: If not specified, SUPPORT is set to a number that is 2% of the total transaction count; implying every sequence rule reported must be satisfied by at least 2% of the customer base.

CUSTOMER Statement

Specifies the ID variable that identifies each customer to be analyzed.

Alias: CUST

CUSTOMER *variable(s)*;

Required Argument

variable(s)

Specifies the customer to be analyzed.

TARGET Statement

Specifies the name of the product to be analyzed.

TARGET *variable*;

Required Argument

variable(s)

Specifies the name of the product to be analyzed.

VISIT Statement

VISIT *variable* </*visit-option(s)*>;

Required Argument

Specifies the timing variable. See “Details: SEQUENCE Procedure” on page 334 for an example of the SAME and WINDOW options in the VISIT statement.

variable

Specifies the time-stamp unit to measure. *Variable* is any numeric value, including date or time values.

Options

visit-option

SAME and WINDOW specify the upper and lower timing limits of a sequence rule. $\text{SAME} < \text{time difference} \leq \text{WINDOW}$.

Visit-option can be as follows:

SAME=*same-number*

Specifies the lower time-limit between the occurrence of two events that you want to associate with each other. If the time difference between the two events is less than or equal to *same-number* (that is, it is ‘too soon’), then the two events are treated as occurring in the same visit, and the transaction is not counted.

Default: 0

WINDOW=*window-number*

Specifies the maximum time difference between the occurrence of two events that you want to be treated as the same visit. If the time difference is greater than *window-number* (that is, it is ‘too late’), then the transaction is treated as falling outside of the timing window, and the transaction is not counted. For *NITEM-long* sequence chain, WINDOW applies to the entire chain.

Default: MAX

Details: SEQUENCE Procedure

SAME and WINDOW Parameters

Two optional parameters, SAME and WINDOW, are available to define what is ‘after’. The rule $A \Rightarrow B$ implies $\text{SAME} < \text{TimeB} - \text{TimeA} \leq \text{WINDOW}$.

Any time difference ($\text{TimeB} - \text{TimeA}$) less than or equal to SAME is considered the same time and is consolidated as the same visit and the same transaction. Any time difference exceeding WINDOW falls outside of the timing window and is ignored as well. In other words, SAME lets you define what is ‘too soon’, that is, event B occurred too soon after event A to qualify for $A \Rightarrow B$. Likewise, WINDOW defines ‘too late’, that is, event B occurred too late after event A occurred to be considered for $A \Rightarrow B$.

Consider the following example:

Customer	Visit	Product
1	1	soda
1	2	apples
1	3	juice
1	4	milk
1	5	bread
2	2	soda
2	6	apples
2	7	milk

With SAME=1, the visits are consolidated as follows:

Customer	Visit	Product
1	1	soda and apples
1	3	juice and milk
1	5	bread
2	2	soda
2	6	apples and milk

Customer 1 is counted for apples ==> milk, however, Customer 2 is not. Both customers are counted for soda ==> milk.

If WINDOW=3 was also specified, then only Customer 1 would count for soda ==> milk. Using the above criterion, Customer 2 would not qualify.

Examples: SEQUENCE Procedure

The following examples were executed using the Windows XP Professional operating system and the SAS software release 9.1.3.

Example 1: Performing a Simple 2-Item Sequence Discovery

Features: ASSOCIATION and SEQUENCE Procedures

- Specifying the Maximum Item-Set Size
- Setting the Support Level
- Specifying the Number of Events

The following example demonstrates how to perform a sequence discovery using the ASSOCIATION and SEQUENCE procedures. The example data set SAMPSIO ASSOCS (stored in the sample library) contains 7,007 separate customer transactions. CUSTOMER is an ID variable that identifies the customers. PRODUCT is the nominal target variable that identifies the items. TIME is the visit variable that measures the time span from observation to observation.

As a marketing analyst for a grocery chain, you want to identify likely 2-item purchase sequences. This information can help you make decisions, such as when to distribute coupons, when to put a product on sale, or how to present items in store displays.

Program

The PROC PRINT procedure lists the first 10 observations in the SAMPSIO.ASSOCS data set.

```
proc print data=sampsio.assocs(obs=10);
    title 'Partial Listing of the ASSOCS Data Set';
run;
```

Before you can run the ASSOCIATION and SEQUENCE procedures, you must create the DMDB data set and the DMDB catalog by using a PROC DMDB step.

```
proc dmdb batch data=sampsio.assocs dmdbcat=catseq;
    id customer time;
    class product(desc);
run;
```

The ASSOCIATION procedure determines the products that are related. The DATA= and DMDB= options identify the DMDB data set and catalog, respectively. PROC ASSOC writes the related products to the OUT= data set; this data set is used as input by the SEQUENCE procedure.

```
proc assoc data=sampsio.assocs dmdbcat=catseq
    out=aout(label='Output from Proc Assoc')
```

The ITEMS= option specifies the maximum size of the item set to be considered (default=4). The SUPPORT= option specifies the minimum support level that is required for a rule to be accepted (default =5% of the largest frequency).

```
    items=5 support=20;
```

The CUST statement (alias = CUSTOMER) specifies the ID variable. The TARGET statement specifies the nominal target variable.

```
    cust customer;
    target product;
run;
```

The DATA= and DMDB= options identify the DMDB data set and catalog, respectively. The ASSOC= option identifies the name of the input data set from the previous PROC ASSOC run.

```
proc sequence data=dmseq dmdbcat=catseq
              assoc=aout
              out=sout(label='Output from Proc Sequence')
```

The NITEMS= option specifies the maximum number of events for which rules, or chains, are generated. By default, the SEQUENCE procedure computes binary sequences (NITEMS=2).

```
          nitems=2;
```

The CUST statement (alias = CUSTOMER) specifies the ID variable. The TARGET statement specifies the nominal target variable.

```
          cust customer;
          target product;
```

The VISIT statement names the timing or sequence variable.

```
          visit time;
run;
```

The SORT procedure sorts the observations in descending order by the values of support.

```
proc sort data=sout;
          by descending support;
run;
```

The PRINT procedure lists the first 10 observations in the sorted sequence data set.

```
proc print data=sout(obs=10);
          var count support conf rule;
          title 'Partial Listing of the 2--Item Sequences';
run;
```

Output

PROC PRINT Partial Listing of the SAMPSIO.ASSOCS Data Set

Partial Listing of the ASSOCS Data Set

OBS	CUSTOMER	TIME	PRODUCT
1	0	0	herring
2	0	1	corned_beef
3	0	2	olives
4	0	3	ham
5	0	4	turkey

6	0	5	bourbon
7	0	6	ice_cream
8	1	0	baguette
9	1	1	soda
10	1	2	herring

Partial Listing of the 2-Item Sequences

The OUT= sequence data set contains the following variables:

- COUNT – contains the number of transactions meeting the rule.
- SUPPORT – contains the percent of support, that is the percent of the total number of transactions that qualify for the rule.
- CONF – contains the percent of confidence.
- RULE – contains the text rule.
- ISET1, ISET, ISETn – contain, in order, the events that form the event chain. For this example, the ISET variables are not printed.

The first rule, cracker ==> unhidden, indicates that 337 customers bought unhidden after buying crackers. The confidence factor indicates that 69% of the time a customer will buy unhidden after they buy crackers. Thirty-three percent of the customer base supports this rule.

Partial Listing of the 2--Item Sequences

OBS	COUNT	SUPPORT	CONF	RULE
1	337	33.6663	69.0574	cracker ==> unhidden
2	235	23.4765	48.3539	herring ==> heineken
3	233	23.2767	49.2600	olives ==> bourbon
4	229	22.8771	47.1193	herring ==> corned_beef
5	226	22.5774	46.5021	herring ==> olives
6	225	22.4775	57.3980	baguette ==> heineken
7	220	21.9780	69.1824	soda ==> cracker
8	220	21.9780	56.1224	baguette ==> herring
9	220	21.9780	46.5116	olives ==> turkey
10	218	21.7782	68.5535	soda ==> heineken

Partial Log Listing

```

1
2  proc dmdb batch data=sampsio.assoc dmdbcat=catseq;
3      id customer time;
4      class product(desc);
5  run;
Records processed=    7007  Mem used = 511K.
NOTE: The PROCEDURE DMDB used 0:00:02.65 real 0:00:00.96 cpu.

6
7  proc assoc data=dmseq dmdbcat=catseq
8      out=aout(label='Output from Proc Assoc')
9
10
11      items=5 support=20;
12
13      cust customer;
```

```

14     target product;
15     run;

```

```

----- Potential 1 item sets = 20 -----
Counting items, records read:      7007
Number of customers:                1001
Support level for item sets:        20
Maximum count for a set:            600
Sets meeting support level:         20
Megs of memory used:                0.51

```

```

----- Potential 2 item sets = 190 -----
Counting items, records read:      7007
Maximum count for a set:            366
Sets meeting support level:         183
Megs of memory used:                0.51

```

```

----- Potential 3 item sets = 1035 -----
Counting items, records read:      7007
Maximum count for a set:            234
Sets meeting support level:         615
Megs of memory used:                0.51

```

```

----- Potential 4 item sets = 1071 -----
Counting items, records read:      7007
Maximum count for a set:            137
Sets meeting support level:         317
Megs of memory used:                0.51

```

```

----- Potential 5 item sets = 85 -----
Counting items, records read:      7007
Maximum count for a set:            116
Sets meeting support level:         71
Megs of memory used:                0.51

```

NOTE: The PROCEDURE ASSOC used 0:00:06.52 real 0:00:03.16 cpu.

```

16
17     proc sequence data=dmseq dmdbc=catseq
18         out=sout(label='Output from Proc Sequence')
19
20
21         nitems=2;
22
23         cust customer;
24         target product;
25
26         visit time;
27     run;
Large itemsets:                1206
Total records read:            7007
Customer count:                1001
Support set to:                20
Total Litem Sequences:        398
Number >= support              291
Memory allocated megs:         2

```

NOTE: The PROCEDURE SEQUENCE used 0:00:05.74 real 0:00:02.52 cpu.

Example 2: Specifying the Maximum Number of Item Events and Setting the Lower Timing Limit

SEQUENCE Procedure

- Using the NITEMS= option to Specify the Maximum Number of Event Items
- Using the SAME= option to Set the Lower Timing Limit

This example demonstrates how to specify the maximum number of item events and how to set the lower timing limit of a sequence rule. Before you run the example program, you should submit the PROC DMDB and PROC ASSOC steps from Example 1.

```
proc sequence data=dmseq
             dmdbcat=catseq
             assoc=aout
             out=s4out(label = 'Output from Proc Sequence')
```

The NITEMS= option specifies the maximum number of events for which rules, or chains, are generated.

```
             nitems=4;

             cust customer;
             target product;
```

The SAME= option specifies the lower time-limit between the occurrence of two events that you want to associate with each other (default = 0).

```
             visit time / same=2;
run;
```

The SORT procedure sorts the observations in descending order by the values of support.

```
proc sort data=s4out;
       by descending support;
run;
```

The PRINT procedure lists the first 10 observations in the sorted sequence data set.

```
proc print data=s4out(obs=10);
       var count support conf rule;
       title 'Partial Listing of the 4--Item Sequences';
       title2 'Lower Timing Limit Set to 2';
run;
```

Output

Partial PROC PRINT Listing of the 4-Item Sequence Data Set, Lower Time Set to 2

When the lower time limit is set to 2, the rule with the highest support is now a herring purchase followed by a heineken purchase. Twenty-three percent of the customer population supports it, with a 48% confidence.

Partial Listing of the 4-Item Sequences
Lower Timing Limit Set to 2

OBS	COUNT	SUPPORT	CONF	RULE
1	235	23.4765	48.3539	hering ==> heineken
2	225	22.4775	57.3980	baguette ==> heineken
3	220	21.9780	69.1824	soda ==> cracker
4	218	21.7782	68.5535	soda ==> heineken & cracker
5	218	21.7782	68.5535	soda ==> heineken
6	215	21.4785	45.4545	olives ==> turkey
7	213	21.2787	52.8536	bourbon ==> cracker
8	209	20.8791	100.0000	hering & baguette ==> heineken
9	201	20.0799	55.3719	avocado ==> heineken
10	150	14.9850	30.8642	hering ==> cracker

Partial Log Listing

```

1  proc sequence data=dmseq
2      dmdbcat=catseq
3      assoc=aout
4      out=s4out(label = 'Output from Proc Sequence')
5
6      nitems=4;
7  cust customer;
8  target product;
9
10 visit time / same=2;
11 run;
Large itemsets:          1206
Total records read:     7007
Customer count:         1001
Support set to:         20
Total Litem Sequences:  5641
Number >= support      466
--- Number Items:      3 ---
Total records read:     7007
Customer count:         1001
Total Litem Sequences:  5086
Number >= support       12
--- Number Items:      4 ---
Total records read:     7007
Customer count:         1001
Total Litem Sequences:  0
Number >= support       0
Memory allocated megs:  2
NOTE: The PROCEDURE SEQUENCE used 0:00:33.42 real 0:00:16.17 cpu.

```

References

- Agrawal, R., Imielinski, T., and Swami, A. (1993), “Mining Association Rules between Sets of Items in Large Databases”, *Proceedings, ACM SIGMOID Conference on Management of Data*, 207–216, Washington, D. C.
- Berry, M. J. A. and Linoff, G. (1997), *Data Mining Techniques for Marketing, Sales, and Customer Support*, New York: John Wiley and Sons, Inc.