The SEQUENCE Procedure

Overview
Procedure Syntax
PROC SEQUENCE Statement
CUSTOMER Statement
TARGET Statement
VISIT Statement
Details
Examples
Example 1: Performing a Simple 2-Item Sequence Discovery
Example 2: Specifying the Maximum Number of Item Events and Setting the Lower Timing Limit
References

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
Overview

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 ==> 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 utilizes 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.
Procedure Syntax

PROC SEQUENCE <option(s)>;

CUSTOMER variable(s);

TARGET variable;

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

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The SEQUENCE Procedure

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
   Specifies the input data source in its DMDB form. This data set is read in order to extract the
   timing information necessary to generate the sequence rules.

DMDBCAT=<libref.> SAS-catalog
   Identifies the metadata catalog associated with the input DMDB.

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.

**Default:** If not specified, SUPPORT is set to a number that is 2% of the total transaction count.

*Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.*
The SEQUENCE Procedure

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.

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
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 <\(\text{visit-option(s)}\)>;

Required Argument

Specifies the timing variable. See Details for an example of the SAME and WINDOW options in the VISIT statement.

\(\text{variable}\)

Specifies the time-stamp unit to measure. \(\text{Variable}\) is any numeric value, including date or time values.

Options

\(\text{visit-option}\)

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

\(\text{Visit-option}\) can be as follows:

SAME=\(\text{same-number}\)

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 \(\text{same-number}\) (that is, it is 'too soon'), then the two events are treated as occurring in the same visit, and the transaction is not counted.

\textbf{Default:} 0

WINDOW=\(\text{window-number}\)

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 \(\text{window-number}\) (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 \(\text{NITEM-long}\) sequence chain, WINDOW applies to the entire chain.

\textbf{Default:} MAX

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The SEQUENCE Procedure

Details

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:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Visit</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>soda</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>apples</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>juice</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>milk</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>bread</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>soda</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>apples</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>milk</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Customer</th>
<th>Visit</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>soda and apples</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>---------------</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>juice and milk</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>bread</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>soda</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>apples and milk</td>
</tr>
</tbody>
</table>

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

The following examples were executed using the HP-UX version 10.20 operating system and the SAS software release 6.12TS045.

Example 1: Performing a Simple 2-Item Sequence Discovery

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

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
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 SAMSIO. 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 may 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

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

proc dmdb batch data=sampsio.assocs out=dmseq dmbcat=catseq;
    id customer time;
    class product(desc);
run;

proc assoc data=dmseq dmbcat=catseq
    out=aout(label='Output from Proc Assoc')
    items=5 support=20;
    cust customer;
    target product;
run;
```
proc sequence data=dmseq dmdbcat=catseq
  assoc=aout
  out=sout(label='Output from Proc Sequence')
  nitems=2;
  cust customer;
  target product;
  visit time;
run;

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

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

<table>
<thead>
<tr>
<th>OBS</th>
<th>CUSTOMER</th>
<th>TIME</th>
<th>PRODUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>herring</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>corned_beef</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>olives</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>3</td>
<td>ham</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>turkey</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>5</td>
<td>bourbon</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>6</td>
<td>ice_cream</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>baguette</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>soda</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>herring</td>
</tr>
</tbody>
</table>
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.

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

Partial Log Listing

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

6   proc assoc data=dmseq dmdbcat=catseq
7       out=aout(label='Output from Proc Assoc')
8   run;
9
items=5 support=20;
10
cust customer;
11  target product;
12 run;
13
----- 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 dmdbcat=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.
The PROC PRINT procedure lists the first 10 observations in the SAMPSIO.ASSOCS data set.

```sas
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 DMBD step.

```sas
proc dmbd batch data=sampsio.assocs out=dmseq dmbdcat=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=dmseq 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.

```plaintext
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.

```plaintext
proc print data=sout(obs=10);
   var count support conf rule;
   title 'Partial Listing of the 2-Item Sequences';
run;
```
Example 2: Specifying the Maximum Number of Item Events and Setting 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.

```plaintext
proc sequence data=dmseq
dmdbcat=catseq
assoc=aout
  out=s4out(label = 'Output from Proc Sequence')
  nitems=4;

cust customer;
target product;

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

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

```plaintext
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
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.

<table>
<thead>
<tr>
<th>OBS</th>
<th>COUNT</th>
<th>SUPPORT</th>
<th>CONF</th>
<th>RULE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>235</td>
<td>23.4765</td>
<td>48.3539</td>
<td>herring ==&gt; heineken</td>
</tr>
<tr>
<td>2</td>
<td>225</td>
<td>22.4775</td>
<td>57.3980</td>
<td>baguette ==&gt; heineken</td>
</tr>
<tr>
<td>3</td>
<td>220</td>
<td>21.9780</td>
<td>69.1824</td>
<td>soda ==&gt; cracker</td>
</tr>
<tr>
<td>4</td>
<td>218</td>
<td>21.7782</td>
<td>68.5535</td>
<td>soda ==&gt; heineken &amp; cracker</td>
</tr>
<tr>
<td>5</td>
<td>218</td>
<td>21.7782</td>
<td>68.5535</td>
<td>soda ==&gt; heineken</td>
</tr>
<tr>
<td>6</td>
<td>215</td>
<td>21.4785</td>
<td>45.4545</td>
<td>olives ==&gt; turkey</td>
</tr>
<tr>
<td>7</td>
<td>213</td>
<td>21.2787</td>
<td>52.8536</td>
<td>bourbon ==&gt; cracker</td>
</tr>
<tr>
<td>8</td>
<td>209</td>
<td>20.8791</td>
<td>100.000</td>
<td>herring &amp; baguette ==&gt; heineken</td>
</tr>
<tr>
<td>9</td>
<td>201</td>
<td>20.0799</td>
<td>55.3719</td>
<td>avocado ==&gt; heineken</td>
</tr>
<tr>
<td>10</td>
<td>150</td>
<td>14.9850</td>
<td>30.8642</td>
<td>herring ==&gt; cracker</td>
</tr>
</tbody>
</table>

Partial Log Listing

```plaintext
1 proc sequence data=dmseq
dmdbcat=catseq
assoc=aout
out=s4out(label = 'Output from Proc Sequence')
nitems=4;
cust customer;
target product;
visit time / same=2;
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.
The NITEMS= option specifies the maximum number of events for which rules, or chains, are generated.

nitems=4;
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;
```
References
