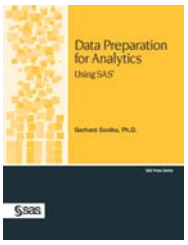
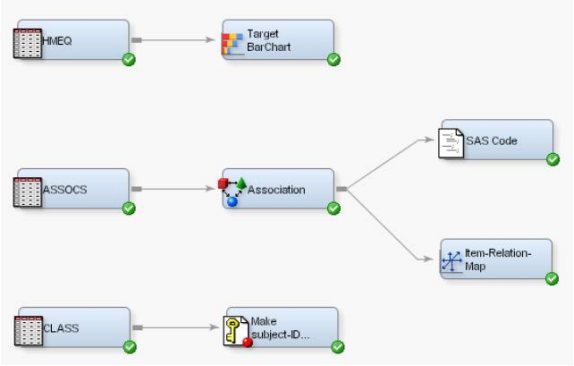


SAS®Enterprise Miner Extension Nodes by Gerhard Svolba

Data Preparation and other useful tools (GTOOLS)



Dr. Gerhard Svolba
SAS-Austria

Email: sastools.by.gerhard@gmx.at

Data Preparation for Analytics →
http://www.sascommunity.org/wiki/Data_Preparation_for_Analytics

Document: GTOOLS - EXTENSION NODES BY GERHARD SVOLBA.DOC

Table of Contents

Description of this document

Change History of this Document

Date	Version	Changes and Enhancements
2006-12-06	2.2	Initial Creation of the SAS Files. Inclusion of the “Concentration” Node for SAS Enterprise Miner
2007-04-07	3.1	Creation of 3 New Nodes: Correlation, Trend Regression, Category Count
2007-05-30	3.2	Validation and Documentation of the New Nodes
2007-05-30	3.2	Completion of the first public version
2007-05-20	3.3	Creation of a separate SAS Catalog for the source files of the nodes (thanks to David Duling for pointing this out)
2008-11-08	3.4	Preparation of the version for sascommunity.org
2010-12-01	4.1	Integration of the Data Preparation and GTOOLS nodes. Provision of the EM 6.x version, a 64bit Windows and a platform independent version.

For any questions and comments please contact:
sastools.by.gerhard@gmx.net

Disclaimer and error tracking

Note that neither SAS nor the author takes responsibility for errors and/or their consequences in the SAS code or the SAS tools provided here. There is no SAS technical support for the content of the macros or the SAS Enterprise Miner extension nodes.

Anyhow, the author is happy to receive your feedback, comments, usage experiences and improvement suggestions when using these SAS tools. If possible and applicable, the author will provide updated versions of the SAS tools through the SAS Press Companion page.

Development Environment

The original development environment for the macros, programs and extension nodes was a desktop installation of SAS 9.1.3 and EM 5.2 on a WindowsXP operating system. The environment has been upgraded and tested to SAS 9.2 and Enterprise Miner 6.1 and 6.2 on a 32bit and 64bit Windows systems.

Abstract

For many data mining analyses, a one-row-per-subject data mart has to be created. If source data is available in a multiple-row-per-subject structure, data needs to be transposed or aggregated. During aggregation, the creation of meaningful derived variables is a key task. The features that describe the behaviour of a subject shall be made available in various variables in the one-row-per-subject structure.

In the book “Data Preparation for Analytics Using SAS” [1] a lot of ideas, macros and coding examples are presented, that describe how meaningful derived variables can be created.

This paper describes how SAS®Enterprise Miner 6.2 has been extended by creating additional nodes that create meaningful derived variables from transactional data and provide them in a one-row-per-subject structure. It is illustrated how easy these nodes can be used to create a rich set of candidate predictors for predictive modelling, which possibly have explanatory value with respect to the target variables. The code has been programmed that way, that it automatically loops over the available list of input variables and combines the resulting derived variables into one dataset.

Extension nodes for the following types of derived variables are provided; Correlation, Concentration, Trend Regression and Category Counts. The paper describes how these nodes can be used and parameterised. It is also shown how so called Extension Nodes can be created and installed in a SAS® Enterprise Miner environment.

Introduction

The creation of meaningful derived variables for data mining is a key task. In many cases however, the analyst does not want to program every derived variable by hand. Especially in the case of several input variables automatic creation of meaningful derived variables is very important.

SAS®Enterprise Miner extension nodes can be used to embed powerful macros into the familiar SAS®Enterprise Miner environment and provide the user tools to easily create derived variables from transactional data.

Using the SAS Enterprise Extension Nodes for Data Preparation -- “Creating a one-row-per-subject data mart from transactional data”

General

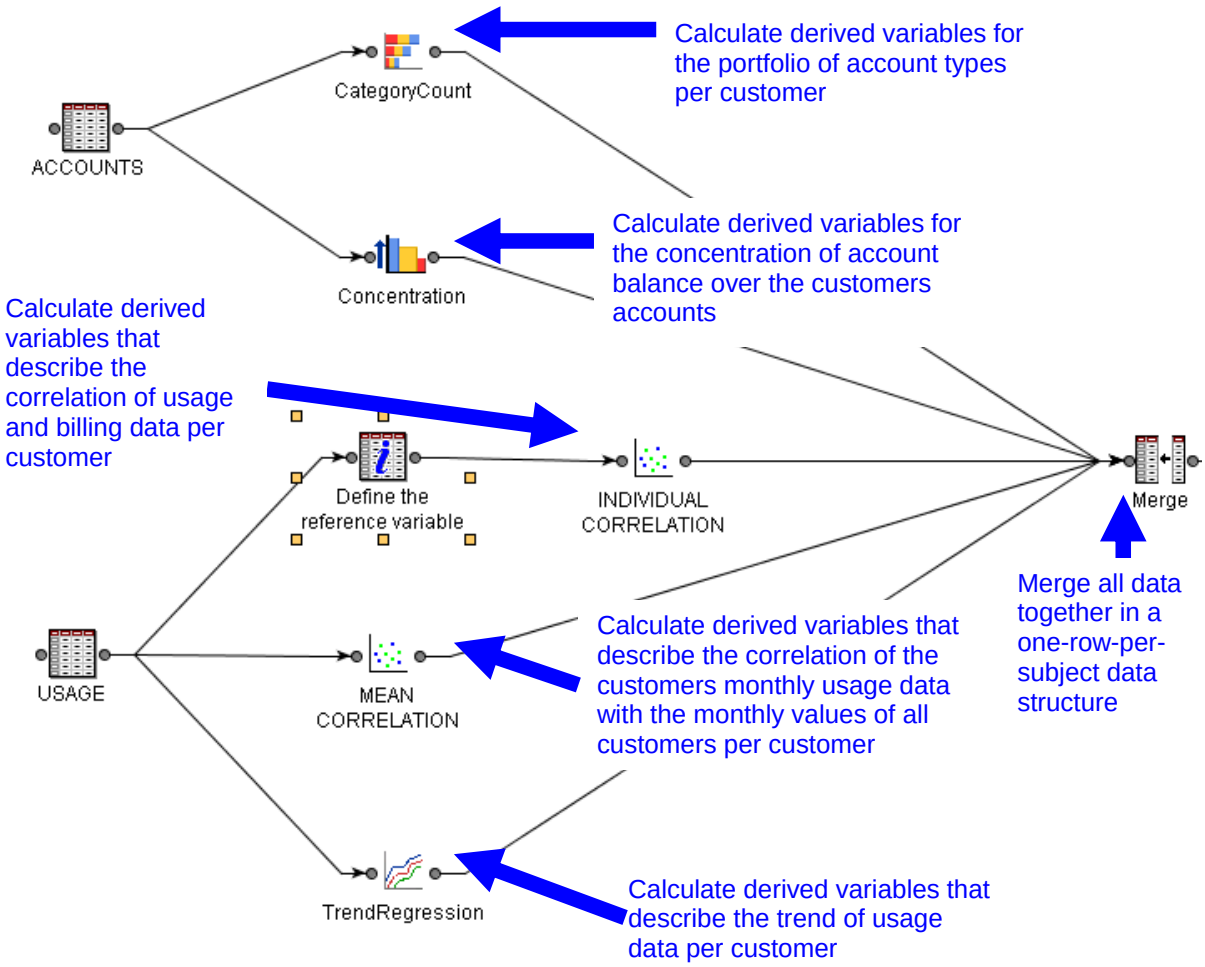
After installation the extension nodes for data preparation will appear in a separate tab called “DATA PREP” in SAS®Enterprise Miner.

The following nodes are available for data preparation. Note that all nodes assume the data in a transactional (multiple-row-per-subject) structure and calculate derived variables on a one-row-per-subject basis.

	Data Preparation for Analytics - Overview	Prints the description, parameterisation and functionality of the nodes into the output window
	TrendRegression	Calculates derived variables that describe the trend of an interval variable in up to two time intervals and creates a concatenated group variable. (see also chapter 18.6 of Data Preparation for Analytics [1])
	Correlation	Calculates derived variables that describe the correlation of values with its overall mean per timeid or with other input variables. (see also chapter 18.3 of Data Preparation for Analytics [1])
	CategoryCount	Calculates derived variables for categorical data. Aggregations like counts, distinct counts or proportions are calculated. (see also chapter 19.4 of Data Preparation for Analytics [1])
	Concentration	Calculates derived variables that describe the concentration of an interval variable in a sub-hierachy of the analysis subject. (see also chapter 18.4 of Data Preparation for Analytics [1])
	TP2Wide	Transpose a dataset from a multiple-row-per-subject structure into a one-row-per-subject structure (see also chapter 14 of Data Preparation for Analytics [1])

The diagram

The following SAS Enterprise Miner diagram shows how the nodes are used to transfer data from 2 transactional data sets (ACCOUNTS, USAGE) into a one-row-per-subject data mart. A XML-File that contains the following diagram can be found in the ZIP file.



Available Data

The two data sets can be found in the \data directory.

ACCOUNT

The ACCOUNT data contain one row per customer and account. For each account the type and a subtype is given. For each account the balance has already been aggregated in the data.

Using the CONCENTRATION and the CATEGORYCOUNT node, we will calculate a concentration of balances per customer and we will calculate number of accounts, number of different account types, and proportion of possible account types from this data.

CustID	AccountID	Balance	Interest	TYPE	TYPESUB
1000002	10001	1102.23	1.00	Saving Account	Saving Account Premium
1000002	10002	813.38	3.00	Loan	Loan fixed
1000005	10003	1844.06	10.00	Saving Account	Saving Account Premium
1000006	10004	696.56	5.00	Funds	Funds international
1000007	10005	1030.26	2.00	Funds	Funds local
1000007	10006	1878.33	1.00	Saving Account	Saving Account Standard
1000008	10007	326.12	0.00	Funds	Funds international
1000009	10008	781.13	2.00	Saving Account	Saving Account Standard
1000009	10009	1522.69	4.00	Funds	Funds international
1000009	10010	1450.78	4.00	Loan	Loan fixed
1000014	10011	734.72	6.00	Saving Account	Saving Account Premium
1000014	10012	379.33	4.00	Saving Account	Saving Account Premium
1000015	10013	766.92	4.00	Saving Account	Saving Account Premium
1000016	10014	829.28	3.00	Saving Account	Saving Account Premium
1000016	10015	276.75	2.00	Saving Account	Saving Account Standard
1000018	10016	702.01	4.00	Funds	Funds international
1000019	10017	132.58	4.00	Funds	Funds international
1000019	10018	1284.79	4.00	Loan	Loan variable

The following picture shows, how the metadata (ROLE and LEVEL) shall be specified for the ACCOUNT data.

Name	Role	Level
AccountID	**Rejected**	**Interval**
Balance	**Input**	**Interval**
CustID	**ID**	**Interval**
Interest	**Input**	**Interval**
Type	**Input**	**Nominal**
TypeSub	**Input**	**Nominal**

USAGE

The USAGE data contain usage information for the last six months per customer. For each month the billing amount and the usage amount is given.

Using the CORRELATION and the TRENDREGRESSION node, we will calculate derived values that describe the course over time and the correlation between billing and usage and the correlation of billing and usage with its monthly mean.

The following pictures shows, how the metadata (ROLE and LEVEL) shall be specified for the USAGE data for the MEAN CORRELATION and the TRENDREGRESSION.

Name	Role	Level
Billing	**Input**	**Interval**
CustID	**ID**	**Interval**
Month	**Time ID**	**Interval**
Usage	**Input**	**Interval**

The following pictures shows, how the metadata (ROLE and LEVEL) shall be specified for the USAGE data for the INDIVIDUAL CORRELATION.

Name	Role	New Role
Billing	Input	**Censor**
CustID	ID	**Default**
Month	Time ID	**Default**
Usage	Input	**Default**

CustID	MONTH	BILLING	USAGE
1000002	1	12	60
1000002	2	12	64
1000002	3	19	69
1000002	4	10	55
1000002	5	10	48
1000002	6	11	25
1000005	1	11	33
1000005	2	10	30
1000005	3	9	43
1000005	4	10	39
1000005	5	10	35
1000005	6	9	33
1000006	1	16	111
1000006	2	19	124
1000006	3	19	116
1000006	4	18	118
1000006	5	22	108
1000006	6	18	103
1000007	1	9	50
1000007	2	14	47
1000007	3	10	49
1000007	4	.	.
1000007	5	11	49
1000007	6	12	50

The Resulting Data Mart

Using the extension nodes will create the following variables in the resulting data mart:

Variable	Description	Created By Node
CustID	CustomerID	
Billing_corr	Correlation of customers montly BILLING values with the overall monthly mean	Correlation (MEAN Mode)
Usage_corr	Correlation of customers montly USAGE values with the overall monthly mean	Correlation (MEAN Mode)
Usage_Billing_corr	Correlation of USAGE and BILLING per customer	Correlation (INDIV Mode)
Billing_FIRST	Trend over time for BILLING values for the first interval	TrendRegression
Billing_SECOND	Trend over time for BILLING values for the second interval	TrendRegression
Billing_GROUP	Concatenated grouped values for the first and second trend of BILLING	TrendRegression
Usage_FIRST	Trend over time for USAGE values for the first interval	TrendRegression
Usage_SECOND	Trend over time for USAGE values for the second interval	TrendRegression
Usage_GROUP	Concatenated grouped values for the first and second trend of USAGE	TrendRegression
Type_Count	Number of distinct ACCOUNT per customer	CategoryCount
TypeDistinctCount	Number of distinct ACCOUNT TYPES per customer	CategoryCount
TypeDistinctProp	Proportion of distinct ACCOUNT TYPES per customer	CategoryCount
TypeOnlyDistinct	Indicator if customer has only distinct ACCOUNT	CategoryCount

	SUB TYPES	
TypePossibleProp	Proportion of possible ACCOUNT SUB TYPES per customer	CategoryCount
TypeAllPossible	Indicator if customer has all possible ACCOUNT TYPES	CategoryCount
TypeSub_Count	Number of SUB TYPES ACCOUNT per customer	CategoryCount
TypeSubDistinctCount	Number of distinct ACCOUNT SUB TYPES per customer	CategoryCount
TypeSubDistinctProp	Proportion of distinct ACCOUNT SUB TYPES per customer	CategoryCount
TypeSubOnlyDistinct	Indicator if customer has only distinct ACCOUNT TYPES	CategoryCount
TypeSubPossibleProp	Proportion of possible ACCOUNT SUB TYPES per customer	CategoryCount
TypeSubAllPossible	Indicator if customer has all possible ACCOUNT SUB TYPES	CategoryCount
Balance_conc	Concentration of BALANCE values over accounts per subject	Concentration
Interest_conc	Concentration of INTEREST values over accounts per subject	Concentration

CUSTID	BALANCE_...	INTEREST_...	TYPE_COUNT	TYPEDISTIN...	TYPEDISTIN...	TYPEONLY...	TYPEPOSSI...	TYPEALLP...	TYPESUB_...	TYPESUBDI...	TYPESUBDI...	TYPESUBO...	TYPESUBP...	TYPESUBA...	USAGE_BIL...
1000002	0.575395	0.75	2	2	100.0	1	66.7	0	2	2	100.0	1	33.3	0	0.771429
1000005	1	1	1	1	100.0	1	33.3	0	1	1	100.0	1	16.7	0	-0.51471
1000006	1	1	1	1	100.0	1	33.3	0	1	1	100.0	1	16.7	0	0.028571
1000007	0.645787	0.666667	2	2	100.0	1	66.7	0	2	2	100.0	1	33.3	0	-0.52705
1000008	1	.	1	1	100.0	1	33.3	0	1	1	100.0	1	16.7	0	0.314286
1000009	0.598754	0.6	3	3	100.0	1	100.0	1	3	3	100.0	1	50.0	0	-0.54286
1000014	0.659503	0.6	2	1	50.0	0	33.3	0	2	1	50.0	0	16.7	0	0.2
1000015	1	1	1	1	100.0	1	33.3	0	1	1	100.0	1	16.7	0	-0.25714
1000016	0.749781	0.6	2	1	50.0	0	33.3	0	2	2	100.0	1	33.3	0	0.289886
1000018	1	1	1	1	100.0	1	33.3	0	1	1	100.0	1	16.7	0	0.666737
1000019	0.906463	0.5	2	2	100.0	1	66.7	0	2	2	100.0	1	33.3	0	0.542857
1000021	1	1	1	1	100.0	1	33.3	0	1	1	100.0	1	16.7	0	0.257143
1000022	1	1	1	1	100.0	1	33.3	0	1	1	100.0	1	16.7	0	-0.02857

CUSTID	USAGE_BIL...	BILLING_CO...	USAGE_CO...	BILLING_FIR...	BILLING_SE...	BILLING_GR...	USAGE_FIR...	USAGE_SE...	USAGE_GR...
1000002	0.771429	0.028571	0.257143	-0.70571	0.7	== ...	-6.77143	-23	-- ...
1000005	-0.51471	0.637748	-0.52179	-0.31143	-0.2	== ...	0.314286	-2	=- ...
1000006	0.028571	-0.37143	0.714286	0.497143	-3.5	=- ...	-2.45714	-5	-- ...
1000007	-0.52705	0.3	-0.10541	0.226744	1.1	=+ ...	0.174419	1	=+ ..
1000008	0.314286	-0.31429	-0.2	0.04	-2.5	=- ...	3.257143	15	++ ..
1000009	-0.54286	-0.54286	-0.02857	0.405714	3.1	=+ ...	2.057143	-9	+- ...
1000014	0.2	-0.77143	0.142857	0.385714	-2.2	=- ...	-2.42857	8	-+ ...
1000015	-0.25714	-0.82857	0.257143	-0.09714	-3.8	=- ...	1.314286	-8	+- ...
1000016	0.289886	0.142857	0.20292	-1.33143	0.9	-= ...	-0.17143	9	=+ ..
1000018	0.666737	0.885714	0.753702	-0.26	3.3	=+ ...	-2.6	53	-+ ...
1000019	0.542857	-0.08571	0.485714	-0.33143	-2.5	=- ...	-5.6	-15	-- ...
1000021	0.257143	-0.2	-0.71429	0.52	-0.5	== ...	1.2	-5	+- ...
1000022	-0.02857	0.2	0.714286	0.048571	-2.3	=- ...	-2.65714	-3	-- ...

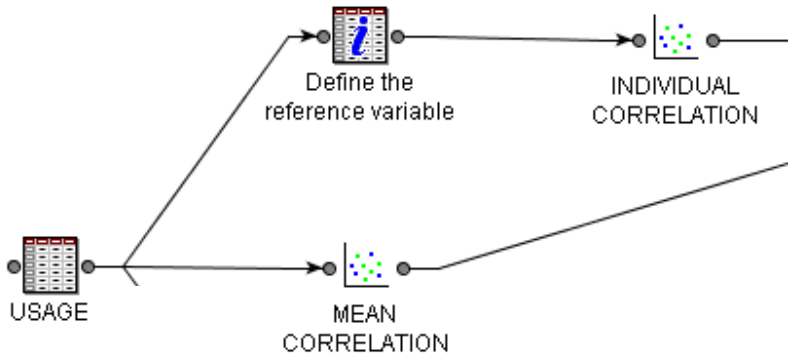
We see that the nodes calculate 23 derived variables from the source data. Assuming k variables that qualify as INPUT variables with USE = YES for the respective node, the following number of derived variables is calculated:

Correlation (INDIV Mode)	k
Correlation (MEAN Mode)	k
TrendRegression	3*k
CategoryCount	6*k
Concentration	K
Transpose to WIDE	k * n (where n is the number of time intervals in the transactional data)

Description of the Extension Nodes for Data Preparation

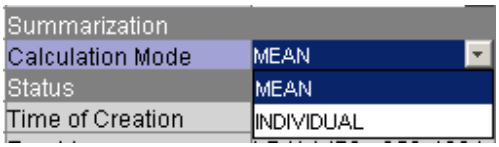
Correlation Node

The correlation node calculates correlations of interval values per subject. Two modes are possible; The INDIVIDUAL mode calculates correlation per subject between input variables in the source data. The MEAN mode calculates correlation per subject between the customer individual value per TIMEID and the overall mean value per TIMEID.



Note the following for the usage of the node

- Note that the data source that is used for the node must have the role TRANSACTION.
- The variables that identifies the analysis subject, for which correlations measures are created must have the role ID.
- The variables that identifies the repetition per subject must have the role TIMEID.
- The node will calculate correlations measures for all INPUT, INTERVAL variables, that have a USE = “YES” in the variables dialog (see below).
- The node can run in two modes; MEAN and INDIVIDUAL. The run mode is specified with the following parameter in the property sheet.



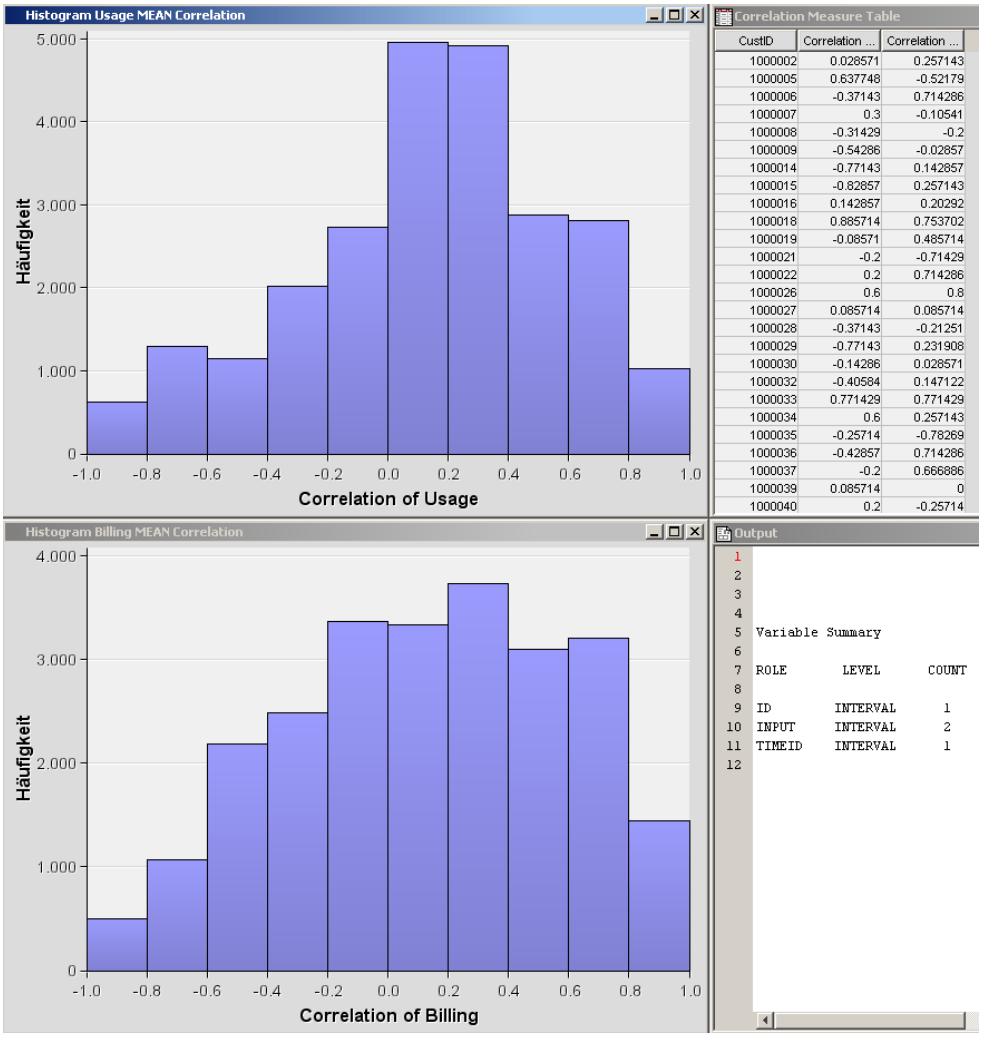
- If MEAN is specified, correlations for each value of TIMEID per subject with the overall mean per TIMEID are calculated.
- If INDIVIDUAL is specified, one interval variable must have the role CENSOR, the correlation of all INPUT INTERVAL variables to this variable are calculated

The following results are created

- The node creates an output dataset with one row per ID. This dataset holds one variable with correlation measures for each INTERVAL variable, that is set to USE = YES. This

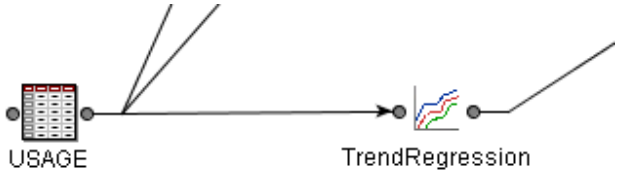
dataset can be merged to a dataset with one-row per ID value using the MERGE node of SAS®Enterprise Miner.

- A histogram is created for each concentration measure variable.



TrendRegression Node

The TrendRegression node calculates derived variables that describe the trend of interval values per subject. Trends can be calculated for up to two intervals and the trend can be grouped into a class variable and concatenated for the two intervals



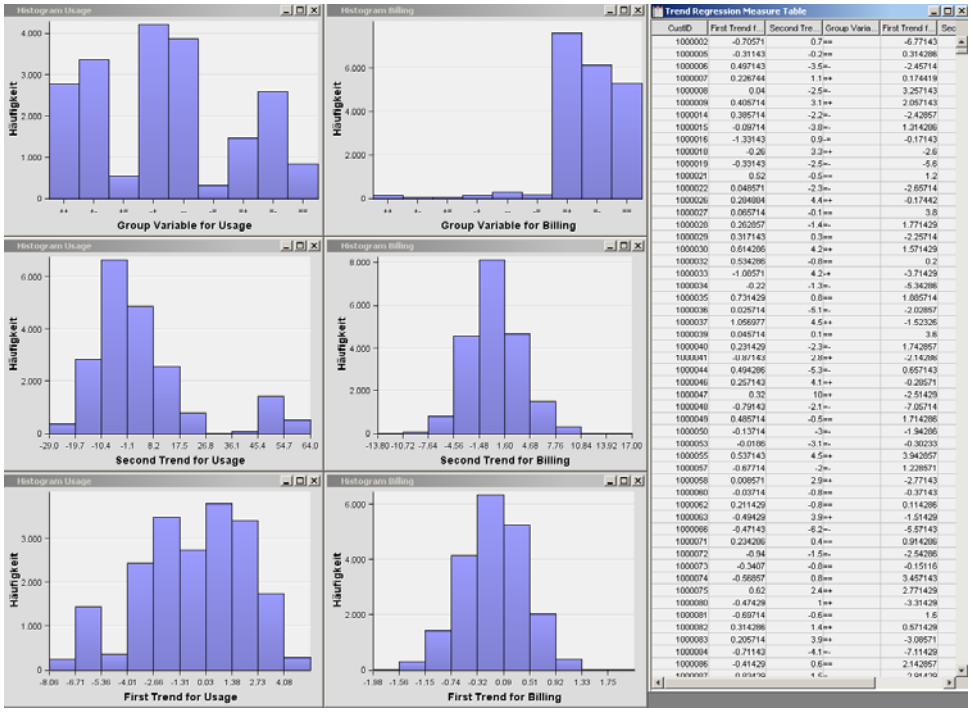
Note the following for the usage of the node

- Note that the data source that is used for the node must have the role TRANSACTION.
- The variables that identifies the analysis subject, for which correlations measures are created must have the role ID.
- The variables that identifies the repetition per subject must have the role TIMEID.
- The node will calculate trend measures for all INPUT, INTERVAL variables, that have a USE = “YES” in the variables dialog (see below).
- For the first interval (= stage) FROM and TILL values can be specified that are used in a WHERE condition for the TIMEID values.
- Parameters are available to control whether a second stage or the creation of a group variable shall take place.
- For the creation of the grouping variable 2 limits can be specified. The trend is classified into 3 groups +, =, and -. Lower Limit separates – and =, Upper Limit separates = and +; where the limit itself is placed in the lower group (lower equal).

Property	Value
Node ID	TrendRegression
Imported Data	...
Exported Data	...
Variables	...
First Stage	
From for first stage	-9.99999999E8
Till for first stage	9.99999999E8
Second Stage	
TwoStage	YES
From for 2nd stage	5.0
Till for 2nd stage	6.0
Grouping	
Concatenated Group	YES
Lower Limit	-1.0
Upper Limit	1.0

The following results are created

- The node creates an output dataset with one row per ID. This dataset holds up to three variables with trend measures for each INTERVAL variable, that is set to USE = YES. This dataset can be merged to a dataset with one-row per ID value using the MERGE node of SAS®Enterprise Miner.
- A histogram is created for each set of trend measure variables.



Concentration Node

The concentration node creates the concentration measure as described in section 18.4 (page 187) of Data Preparation for Analytics.

Note the following for the usage of the node

- Note that the data source that is used for the node must have the role TRANSACTION.
- The variables that identifies the analysis subject, for which concentration measures are created must have the role ID.
- The node will calculate concentration measures for all INPUT, INTERVAL variables, that have a USE = “YES” in the variables dialog (see below).
- Per default, the node assumes the data to have one row per analysis subject and sub-hierarchy. E.g. one line per customer and account.
- If the data does have multiple rows per analysis subject and sub-hierarchy the data need to be pre-aggregated. This can be achieved by the following steps:

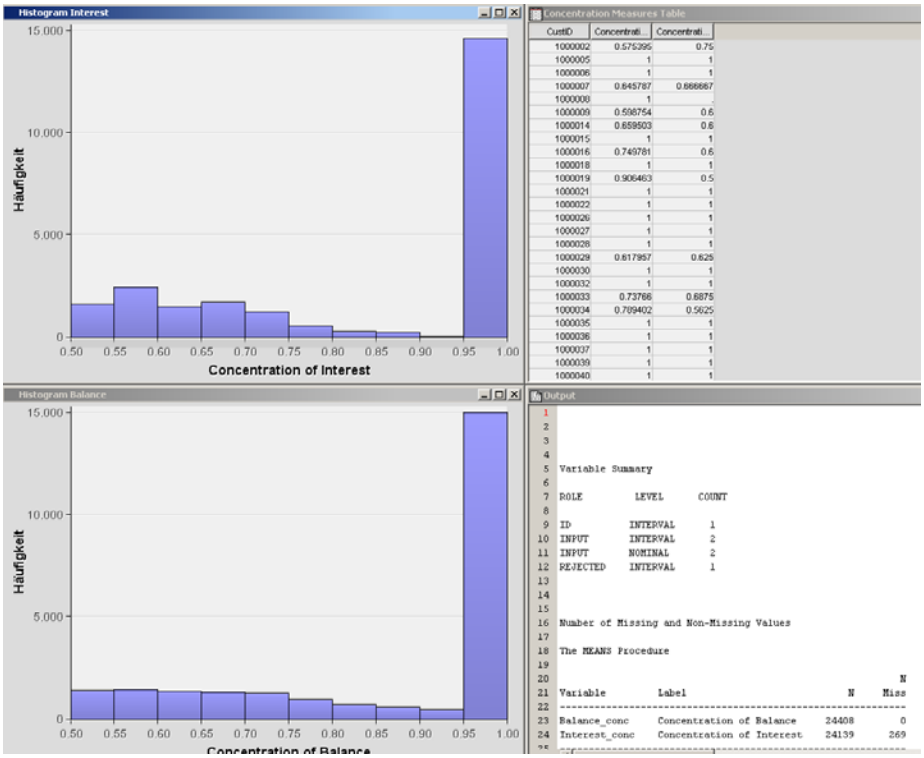
- Set role = SEGMENT to the variables that identifies the sub-hierachies (e.g. AccountID)
- Note that in this situation we misuse the role SEGMENT for a second level hierarchy.
- Setting the node parameter SUMMARIZE to YES (default = NO)
- Select a desired SUMMARIZATION MEASURE (default = SUM), other options are MEAN or MEDIAN

- Note that the definition and calculation of the concentration measure assumes non-negative input-variables. The calculated concentration" measure for an ID which has at least one negative value, results in a" missing value." There is no logical replacement value for a missing value in this case." In the case of only a few missing values (e.g. max 10%), it is recommended" to replace the missing values with the Replacement Node.

Property	Value
Node ID	Concentration2
Imported Data	...
Exported Data	...
Variables	...
Summarization	
Summarize	NO
Summarization Measure	SUM

The following results are created

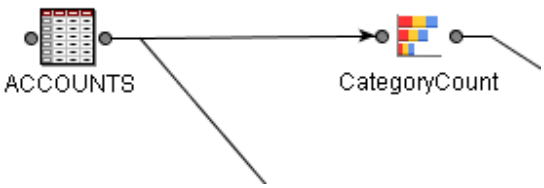
- The node creates an output dataset with one row per ID. This dataset holds one variable with concentration measures for each INTERVAL variable, that is set to USE = YES. This dataset can be merged to a dataset with one-row per ID value using the MERGE node of SAS®Enterprise Miner.
- The node also creates textual output showing information about the number of missing values.
- A histogram is created for each concentration measure variable.



CategoryCount Node

The CategoryCount node calculates derived variables the contain counts and distinct counts for categorical variables.

- the number of multiple observations per subject
- the number of distinct categories
- the proportion of distinct categories on all multiple observations per subject
- the indicator that a subject has only distinct categories
- the proportion of distinct categories on all available categories
- the indicator that a subject has all possible categories



Note the following for the usage of the node

- Note that the data source that is used for the node must have the role TRANSACTION.
- The variables that identifies the analysis subject, for which concentration measures are created must have the role ID.

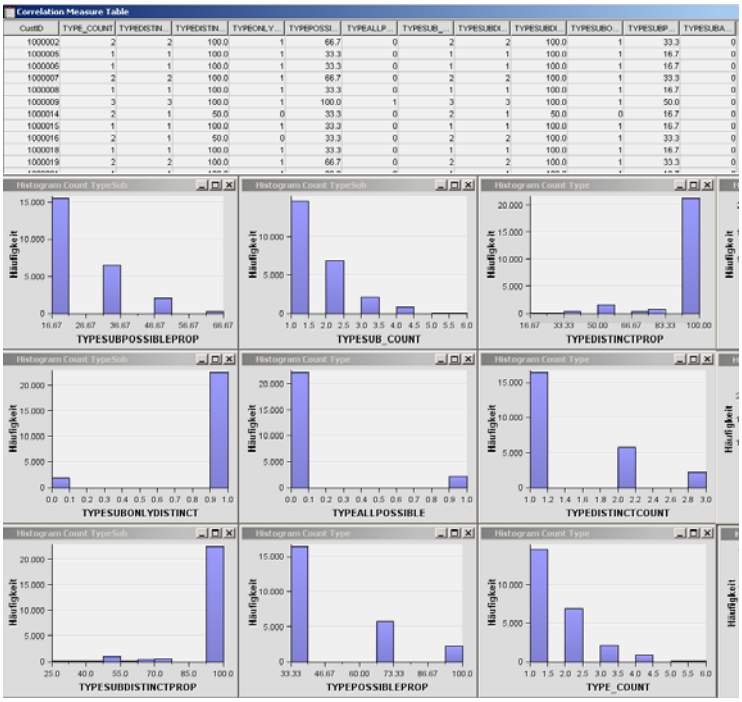
- The node will calculate concentration measures for all INPUT, NOMINAL variables, that have a USE = “YES” in the variables dialog.
- Per default, the node assumes the data to have one row per analysis subject and sub-hierarchy. E.g. one line per customer and account.

The following results are created

- The node creates an output dataset with one row per ID. This dataset containts the following variables for each nominal <category> variable that is set to USE=YES.

<category>_Count	Number of distinct <CATEGORY> per customer
<category>DistinctCount	Number of distinct <CATEGORY> per customer
<category>DistinctProp	Proportion of distinct <CATEGORY> per customer
<category>OnlyDistinct	Indicator if customer has only distinct ACCOUNT SUB <CATEGORY>
<category>PossibleProp	Proportion of possible <CATEGORY> per customer
<category>AllPossible	Indicator if customer has all possible <CATEGORY>

- This dataset can be merged to a dataset with one-row per ID value using the MERGE node of SAS®Enterprise Miner.
- A histogram is created for each concentration measure variable.



Transpose to Wide Node (TP2WIDE)

This node transposes transactional input data to a one-row-per-subject data structure. For each input variable, the node creates a new columns every timeid that is available in the input data. Note that the name of these columns is concatenated of the name of the respective input variable and the timeid value.

If the input data contain two variables, BILLING and USAGE with TIMEIDs 1-6 for every variable, the result of the transpose node is as follows:

CustID	Billing1	Billing2	Billing3	Billing4	Billing5	Billing6	Usage1	Usage2	Usage3	Usage4	Usage5	Usage6
1000002	12	12	19	10	10	11	60	64	69	55	48	25
1000005	11	10	9	10	10	9	33	30	43	39	35	33
1000006	16	19	19	18	22	18	111	124	116	118	108	103
1000007	9	14	10	.	11	12	50	47	49	.	49	50
1000008	12	9	13	15	12	10	29	31	42	47	34	49
1000009	9	8	11	8	9	12	23	34	27	33	41	32
1000014	16	12	15	13	18	15	85	72	70	58	66	74

Note that the transpose node uses PROC TRANSPOSE and a special macro logic that allows to transpose more than one variable in one go. See also SAS sample 31122 (http://support.sas.com/kb/32/122.html).

GTOOLS – Useful tools in SAS Enterprise Miner

General

After installation the extension nodes for GTOOLS will appear in a separate tab called “GTOOLS” in SAS®Enterprise Miner.

The following nodes are available.

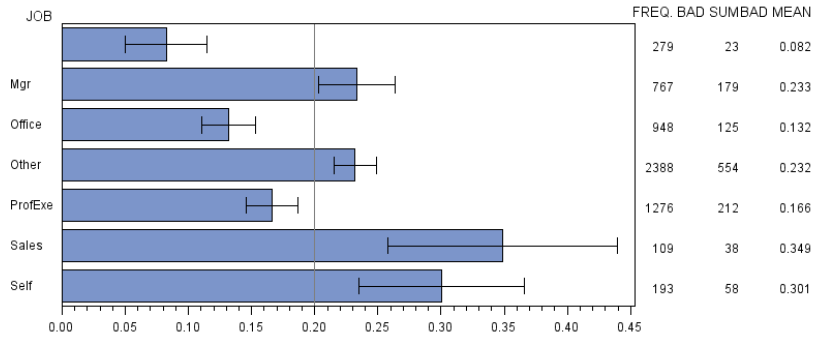
	TargetChart	Displays the relationship between an input variable and the target variables in a bar chart (see also chapter 20 of Data Preparation for Analytics [1])
	Anonymous	Removes the id variable of a subject from the table and stores it in a separate table. Optionally creates a surrogate key. This node shall be used if a dataset shall be made anonymous.
	Itemmap	Displays the 2-way association rules in a hierarchical tree structure.
	EM Dataset Copy	This node is not finished yet! It will copy to metadata of an existing data source in SAS Enterprise Miner.

Details for the Nodes

TargetChart

Displays the relationship between an input variable and the target variables in a bar chart (see also chapter 20 of Data Preparation for Analytics. This nodes needs a dataset with a TARGET variable and with INPUT variables that are NOMINAL and/or INTERVAL.

A target chart is created for each input variable that shows the overall proportion as a vertical line and the proportions per class in the bars.



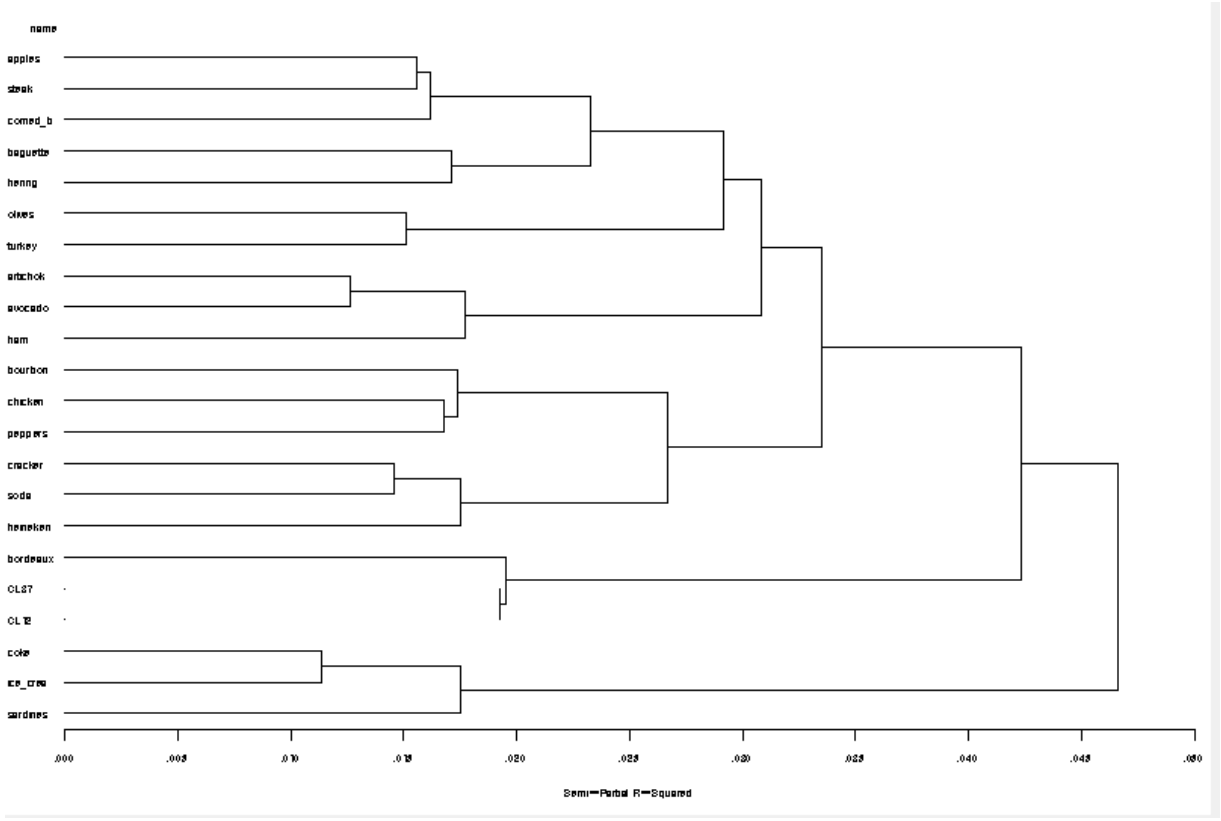
Note that the results of the node can be found in the results window under VIEW → SAS RESULTS → TRAIN GRAPHS.

Anonymous

This node shall be used if a dataset shall be made anonymous. The node removes the ID variable from the input dataset and stores it in a new dataset with the role DOCUMENT. Optionally you can specify that a new surrogate ID variable shall be created. You can also specify the name of this variable.

Itemmap

Displays the 2-way association rules in a hierarchical tree structure. Note that the node can only be run, if an association node is run before. In the association node the maximum number of items has to be set to 2. The node uses the association results to create a hierarchical tree structure to display the closeness of certain items. Note that an earlier version of the underlying macro for EM 4.x was provided to me by Hendrik Wagner. For the 5.x and 6.x version of Enterprise Miner some amendments in the Code had to be made as dataset names are different.



Note that the results of the node can be found in the results window under VIEW→ SAS RESULTS → TRAIN GRAPHS.

A Note on programming SAS Enterprise Miner Extension Nodes

General

Creating Extension Nodes for SAS®Enterprise Miner is a powerful tool to provide additional data mining functionality to users. Extension nodes allow seamless integration into the existing familiar SAS Enterprise Miner Environment. Users can benefit from their existing user experience with SAS Enterprise Miner. Extension Nodes go beyond the classic “SAS Code Node” by allowing providing macro parameters via selection lists and property sheets and present output like graphs and tables like standard Enterprise Miner Nodes.

Note that SAS®Enterprise Miner 6.x provides a template node for creating Enterprise Miner Extension Nodes. The “EXT DEMO” node can be found in the UTILITY tab.

Frequently Used Code Structures

Creating a Extension Nodes for SAS®Enterprise Miner has an extremely steep learning curve. The following hints allow the user to start off more quickly with creation of a extension nodes.

Provide XML to define the property sheet

The following example shows an example XML file, where a VARIABLES dialog is defined and a list of possible values for macro variable MODE is defined.

```
<?xml version="1.0" ?>
- <Component type="AF" serverclass="SASHELP.EMCORE.EMCODETOOL.CLASS" name="CORRELATION"
    group="DataPrep" prefix="CORRELATION" displayName="CORRELATION" description="CORRELATION"
    icon="CORRELATION.gif" resource="com.sas.analytics.eminer.visuals.PropertyBundle">
- <PropertyDescriptors>
  <Property type="String" name="Location" initial="CATALOG" />
  <Property type="String" name="Catalog" initial="SASHELP.DATAPREP.CORRELATION.SOURCE" />
  <Property type="String" name="ToolType" initial="DATAPREP" />
- <Property type="String" name="VariableSet" displayName="properties.common.variables.txt"
    description="properties.common.variables.desc.txt">
  <Control type="DIALOG" showValue="N" allowTyping="N"
    dialogClass="com.sas.analytics.eminer.visuals.VariablesDialog" />
    </Property>
- <Property type="String" name="Mode" displayName="Calculation Mode" description="Calculation Mode for
    Correlations" initial="MEAN" allowTyping="N">
- <Control type="CHOICE">
  <Choice rawValue="MEAN" />
  <Choice rawValue="INDIVIDUAL" />
    </Control>
    </Property>
    </PropertyDescriptors>
- <Views>
- <View name="Basic">
  <Property name="VariableSet" />
    </View>
- <View name="Advanced">
  <Property name="VariableSet" />
- <Group name="Summarization" displayName="Summarization" description="Summarization of Sub-
    Hierarchies">
  - <!--
```

```
  Property name="Task" /
   -->
<Property name="Mode" />
   </Group>
    </View>
  </Views>
  </Component>
```

Use parameter values from the property sheet

The parameter value MODE that has been defined in the above example can be used in the macro programm:

```
%IF &EM_PROPERTY_MODE = INDIVIDUAL %THEN %DO;
```

Control the number of loops with macro variables

The number of variables in a certain set can be retrieved from the respective macro variable:

```
%do i = 1 %to &em_num_nominal_input;
```

Extract the respective element of the variable list

The i^{th} element of a variable list can be extracted with the following statement.

```
%scan(%em_nominal_input,&i)
```

Re-Use the metadata from the input data source node

Metadata for variables that are defined in the Input Data Soruce Node or in the Metadata Node can be used to assign specific roles to variables. These roles, e.g. %EM_ID for the ID-variable, can be used to control merges and by processing.

```
%EM_ID
%EM_SEGMENT
%EM_REFERRER
```

Register Data for Output Generation

In order to use data for output generation in the result window, the dataset needs to be registertied with e.g. the following statement.

```
%EM_REGISTER(KEY=COUNT,TYPE=DATA);
DATA &EM_USER_COUNT;
 SET &EM_EXPORT_TRAIN;
RUN;
```

Create Graphical Output in the Results Window

Graphical Output in the Results Window can be generated with e.g. the following statement.

```
%DO I = 1 %TO &EM_NUM_NOMINAL_INPUT;
    %EM_REPORT(
           KEY=COUNT,
           VIEWTYPE=HISTOGRAM,
           X=%SCAN(%EM_NOMINAL_INPUT,&I)_COUNT,
           BLOCK=CORRELATION,
           description=Histogram Count
           %scan(%em_nominal_input,&i),
           autodisplay=Y);
```

Installing the SAS Enterprise Miner Extension Nodes

Installation for EM 6.x on Windows 32bit, Windows 64bit and UNIX environments

General

The two directories “ExtensionNodes_EM6x” and “SASSourceCode_EM6x” are needed for EM 6.x. Note that the “SASSourceCode_EM6x” directory contains the SAS Catalogs with the source code for the Enterprise Nodes which is available in different versions.

For non-Windows environments the source code can be imported from the XPT-file with e.g. the following code:

```
*** Example to extract the source code files for Non-Windows plattforms;
*** Assuming that the XPT file has been stored in c:\tmp\nodesbygerrhard.xpt
    and the catalogs shall be exported to c:\tmp;
*** Dr. Gerhard Svolba - 1.12.2010;

libname newlib 'c:\tmp';
proc cimport library=newlib infile='c:\tmp\nodesbygerhard.xpt' memtype =
catalog;
run;
```

Step 1a: Installation for EM 5.x Full Desktop systems

Use the following steps to install the extension nodes on a system that is not running the EM 5.2 Analytics Platform (mid-tier). You can check your configuration by selecting

Help → Configuration from the main menu.

- Close the EM 5.x client application.
- Find the SAS root directory, for example: C:Program Files\SAS
- Find the SAS AP directory, for example: C:\Program Files\SAS\SASAPCore
- Copy the XML Files of the directory Production\ExtensionNodes_and_SourceCode_Windows_EM5x\XML to the Enterprise Miner extension subdirectory. C:\Program Files\SAS\SASAPCore\apps\EnterpriseMiner\ext
- Copy the GIF Files of the sub-directories gif16 and gif32 in Production\ExtensionNodes_and_SourceCode_Windows_EM5x\GIF to the Enterprise Miner extension subdirectory. C:\Program Files\SAS\SASAPCore\apps\EnterpriseMiner\ext\gif16 and gif32
- Restart the EM 5.2 client application

Installation for EM 5.x on Windows 32 bit systems

Note that the EM 5.x version does not contain the GTOOLS nodes but only the data preparation nodes.

Step 1a: Installation for EM 6.x Full Desktop systems

Use the following steps to install the extension nodes on a system that is not running the EM 6.x Analytics Platform (mid-tier). You can check your configuration by selecting "Help → Configuration" from the main menu.

- Close the EM 6.x client application.
- Navigate to the EnterpriseMiner\ext directory. If SAS in installed in C:\SAS whit will be under "C:\SAS\Config\Lev1\AnalyticsPlatform\apps\EnterpriseMiner\ext"
- Copy the XML Files of the directory \ExtensionNodes_EM6x\ to the Enterprise Miner extension subdirectory. …\Config\Lev1\AnalyticsPlatform\apps\EnterpriseMiner\ext
- Copy the GIF Files of the sub-directories gif16 and gif32 in …\ExtensionNodes_EM6x\GIF to the Enterprise Miner extension subdirectory. . …\Config\Lev1\AnalyticsPlatform\apps\EnterpriseMiner\ext\gif16 and gif32
- Restart the EM 6.x client application

Step 1b: Installation for EM 6.x Shared Platform systems

Follow these steps to install the extension nodes on the EM6.2 Shared Platform system. You do not need to update each individual end-user client. The extension nodes will be available to all users.

- Close the EM 6.2 shared platform
- Navigate to the EnterpriseMiner\ext directory. If SAS in installed in C:\SAS whit will be under "C:\SAS\Config\Lev1\AnalyticsPlatform\apps\EnterpriseMiner\ext"
- Copy the XML Files of the directory \ExtensionNodes_EM6x\ to the Enterprise Miner extension subdirectory. …\Config\Lev1\AnalyticsPlatform\apps\EnterpriseMiner\ext
- Copy the GIF Files of the sub-directories gif16 and gif32 in …\ExtensionNodes_EM6x\GIF to the Enterprise Miner extension subdirectory. . …\Config\Lev1\AnalyticsPlatform\apps\EnterpriseMiner\ext\gif16 and gif32
- Restart the EM 6.2 Shared Platform

Step 2: Install the Source Code

- Find the SAS-Catalog NODESBYGERHARD from the subdirectory in …\SASSourceCode_EM6x\ to your SASHELP Directory which corresponds to your operating system.
- E.g. you can copy the Code to "!SASROOT\dmine\sashelp". If SAS in installed under C:\SAS this will be most likely the directory: C:\SAS\SASFoundation\9.2\dmine\sashelp
- Alternatively you can copy this catalog to any directory and make the directory path available to the –SASHELP option in your SAS-config file. E.g.

```
-SASHELP        (
                 "!SASCFG\SASCFG"
                 "!sasroot\core\sashelp"
                 "!sasext0\dmine\sashelp"
```

Example Data and Example Diagrams

The Example Data and Example Diagrams can be found under: \Production\Example_Data and \Production\Example_EM_Diagrams. The SAS datasets are in windows formats. For Unix environments the datasets are provided in the XPT format as well. These datasets are needed by the example diagram: EM_Diagramm_01_DataPreparation_Examples.xml

For non-Windows environments the datasets can be imported from the XPT-file with e.g. the following code:

```
*** Example to extract the data for Non-Windows plattforms;
*** Assuming that the XPT file has been stored in c:\tmp\data.xpt and the
files shall be exported
    to c:\tmp;
*** Dr. Gerhard Svolba - 1.12.2010;
libname newlib 'c:\tmp';
proc cimport library=newlib infile="c:\tmp\data.xpt";
run;
```

Macro Usage

In order to avoid problems with non-existing macros definitions in a SAS-Session, the extension nodes do not call the same SAS macro instances as are described above and contained in the MACROS.SAS file which comes with my book “Data Preparation for Analytics”.

If one of the data preparation macros is needed in an extension node, the source code contains its own copy of the macro. Therefore any changes to the macros in MACROS.SAS have no influence on the extension nodes.

Summary

It has been discussed in this paper that the creation of meaningful derived variables from transactional data for data mining in a one-row-per-subject data structure is a key task. More ideas on this topic can be found in the book “Data Preparation for Analytics”. The extension nodes that are provided with this paper allow the easy creation of such derived variables within the SAS®Enterprise Miner environment. The nodes automatically create the derived variables for each input variable that is provided. Extension nodes in SAS®Enterprise Miner are a powerful way to provide now functionality to users in the familiar environment.

References

[1] Data Preparation for Analytics; Gerhard Svolba; SAS Press 2006

Gerhard Svolba: Bulding an efficient one-row-per-subject data mart for data mining; 2006 SUGI31; San Francisco

SAS Education Course: Extending SAS Enterprise Miner

Dave Duling “Building a data mining community”; M2006 Las Vegas

SAS White Paper: Integrating Your Favourite SAS® Program into a SAS® Data Mining Workbench
http://www.sas.com/resources/whitepaper/wp_6874.pdf