Handling Missing Values

Dr. Goutam Chakraborty

Outline For This Session

- Data exploration and data preparation before data mining
  - Handling missing (and/or unexpected) values,
  - Handling transformation ,
  - Handling extreme values (outliers)

Unexpected Values in a Column

- A numerical variable has values beyond the range it is supposed to have!
  - Best option: if we can go back and check and fix the data
  - Practical option 1: Replace nonsensical value by missing value.
  - Practical option 2: Replace the nonsensical value with something reasonable(?!)
- A binary variable have more than 2 values in a data set!!!
  - Two values that it is supposed to have plus some more!
- Missing and empty values (in a data base both of these may be called null values)
    - But, they often arise due to different reasons!

Missing versus Empty Values

- Missing means the value for the variable *exists in the data world* but has **not been entered** in the data base
- Empty means the value for the variable **does not exist** in the data world
- An example with Gender (M/F) of a person and type of cheese (A/S/C) on a Turkey sandwich ordered at a restaurant.
  - Missing value likely to occur in ….
  - Empty value likely to occur in …
- Point is, we need to be sensitive about how missing/null values may have been generated before attempting to fix it!

Dealing with Missing Values

- What is the problem if we have missing value?
  - Some algorithms will throw out any record with missing value
  - Some algorithms will automatically replace missing value
- Many methods exist to replace missing values with “some other values.” Some of the issues that I want you to keep in mind are:
  - Replacing missing values without capturing (usually with a separate flag variable) that there were missing values, actually removes information from the data set!!! Why?
  - In replacing missing values, analysts often opt for causing “least damage” to the data (i.e., create minimum bias)

Missing Value in SAS System

- Before performing data replacement, you should understand how the SAS System stores missing values.
- By default, missing numeric values are displayed as periods (.) and missing character values are displayed as blanks ( ) in SAS.
  - Numeric data can contain characters that represent special missing values (use MISSING statement when you want to distinguish between types of missing values).
    - The Impute node **should not be used** to replace special missing numeric values that are assigned with a MISSING statement.

Impact of Missing Values

- Missing Data = information (variable value) not available for a subject (or, case) about which other information is available.
- Key issue: what is the nature of the missing data.
  - Systematic?
  - Random?
- Analyst’s Concern = to identify the patterns
  and relationships underlying the missing data in order to maintain as close as possible the original distribution of values when any remedy is applied.
- Impact:
  - Reduces sample size available for analysis.
  - Can distort (bias) results.

Four-Step Process for Handling Missing Data

- Step 1: Determine the type of missing data.
- Step 2: Determine the extent of missing data.
- Step 3: Diagnose the randomness of the missing data processes - **MAR** (missing at random) or **MCAR** (missing completely at random).
- Step 4: Select an appropriate imputation method.

Choices for Missing Data Imputation in SAS Enterprise Miner

- SAS Enterprise Miner has an Imputation node that allows missing value imputation by many methods.
- **Default** imputation techniques are:
  - Interval variables by Mean
  - Class variables by Mode
- **Model-based** imputation techniques for interval or class variable include:
  - Decision Tree (replace missing variable values with replacement values that are estimated by analyzing each input as a target. The remaining input and rejected variables are used as predictors).
  - Consult SAS Enterprise Miner Help for other choices.

Imputation of Missing Data by Multiple Imputation (MI)

- Basic idea behind multiple imputation:
  - Any single replacement of a missing value (by mean, median, and so on) might create unknown biases in the data.
  - Simulate and generate multiple complete data sets from the original data with missing values by repeatedly replacing missing entries with imputed ones.
  - Use the mean of the multiple imputed values as a replacement for missing data.
- The MI procedure in Base SAS allows both finding the patterns of missing data and imputation of missing data.

Impute Node (Modify Tab)

The Impute node provides the following imputations for missing interval variables:

- Andrew's Wave
- Default Constant
- Distribution
- Huber
- **Mean**
- **Median**
- Mid-Minimum Spacing
- Midrange
- None
- **Tree**
- **Tree Surrogate**
- Tukey's Biweight

The Impute node provides the following imputations for missing class variables:

- **Count**
- Default Constant
- Distribution
- None
- **Tree**
- **Tree Surrogat**e

Demonstration of Impute Node

- Add **Impute** node (under **Modify** tab) and Connect the data node to the Impute node
- Check property panel
  - Look at default properties settings for interval and Class variables
  - Change properties as below:
    - Indicator variables Type to Unique and Role to Input
- Right-Click Impute node, select Run. Examine results.

Outline For This Session

- Data exploration and data preparation before data mining
  - Handling missing (and/or unexpected) values,
  - Handling transformation ,
  - Handling extreme values (outliers)

Data Transformation

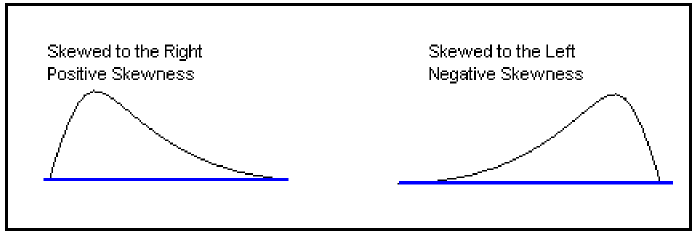
- Why should you consider transforming data?
  - Skewed distribution of numerical variables create problems in many modeling algorithms.
  - Numerical variables with very high variance is likely to emerge as more important in some modeling algorithms.
  - For categorical variables, it is often impractical to use a very large number of classes in models.
  - Often, predictive ability of models improve (particularly when model is applied on unseen data) when independent variables are transformed to be more symmetric (or, Normal).
  - In segmentation problems, skewed variables often create many small segments
- Is there a downside of transformations?

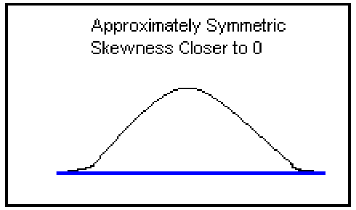
Normalization or Scaling Transformation

- Also, called r**ange-standardization** or **linear scaling**
- Transformed X= (X- Min. X)/(Max. X – Min. X)
- Range of New X is (0,1)
  - Produces no distortion in the data as it retains relative position of values (if X1>X2, then NewX1 > NewX2)
  - Does not change shape of distribution (if X was skewed so is Transformed X)
  - Some modeling tools may do it automatically (Neural Net)
    - Others could benefit from it!
- What if we get out-of-range values in new data?
  - Clip it to maximum/minimum
  - Use special scaling such as SoftMax

(Distributive) Transformation for Numerical Variables

- Power Series Transformation : Goal is to apply a transformation such that the transformed variable closely resemble a Normal distribution. Changes the **shape as well as range** of the distribution:
  - Log — Variable is transformed by taking the natural log of the variable.
  - Square Root — Variable is transformed by taking the square root of the variable.
  - Inverse — Variable is transformed by using the inverse of the variable.
  - Square — Variable is transformed by using the square of the variable.





Useful (Distributive) Transformations in SAS EM

- **Best Power Transformations** : These transformations are a subset of the general class of transformations that are known as Box-Cox transformations. The Transform Variables node in SAS EM supports the following best power transformations:
    - **Maximize Normality** — This method chooses the transformation that yields sample quantiles that are closest to the theoretical quantiles of a normal distribution.
    - **Maximize Correlation with Target** — This method chooses the transformation that has the best squared correlation with the target. **This method requires an interval target**.

Useful (Distributive) Transformations in SAS EM

- **Binning Transformations** **:** These transformations enable you to collapse an interval variable, such as debt to income ratio, into an ordinal grouping variable. There are three types of binning transformations.
  - **Bucket** — Buckets are created by dividing the data values into equally spaced interval based on the difference between the maximum and the minimum values.
  - **Quantile** — Data is divided into groups that have approximately the same frequency in each group.
  - **Optimal Binning for Relationship to Target** — Data is binned in order to optimize the relationship to the target. This method requires a **binary target**.
  - **Group Rare Levels Transformation** **:**This is available for **class variables only**. This method combines the rare levels (if any) into a separate group, _OTHER_. You can use the **Cutoff Value** property to specify the cutoff percentage. Rare levels are the variable values that occur less than the specified cutoff value.

Transform Node (Modify Tab)

- This node enables you to create new variables that are transformations of existing variables in a data.
  - Transformations can be used to stabilize variances, remove nonlinearity, and correct non-normality in variables to help improve model performance.
- The default settings for this node is no transformation
  - You choose the variables you want to transform
  - You choose the type of transformations

Demonstration of Transform

- Add a **Transform** node (under **Modify** tab) to the right of the data node.
- Connect the data node to the Transform node
- Check property panel
  - Under Sample properties, change Method to Random, Size to Max.
  - Under Score, **change Hide to No**.
- Right-Click Transform node, select edit variables, Select GiftCntCardAll, GiftCntcasrd3. Select Explore
  - Note the skewness in the variables. Close Explore.
  - Change Method from Default to **Max Normal**.
- Right-click on Transform node, select Run. Examine results.
- Explore distributions of original and transformed variables via Metadata node

Outline For This Session

- Data exploration and data preparation before data mining
  - Handling missing (and/or unexpected) values,
  - Handling transformation ,
  - Handling extreme values (outliers)

Handling Extreme Values (Filter)

- Extreme values in variables are problematic because they may have undue influence on model.
- Often in predictive modeling applications, training data set is filtered to exclude observations, such as outliers or other extreme observations, that we **do not want** to include in model building.
- Filtering extreme values from the training data tends to produce better models because the parameter estimates are more stable.
  - Sometimes transformation fixes extreme values; other times filtering extreme values may negate the need for transformation.
- Filtering observations typically should not be used to in the **validation, or score** data sets. Why?
  - Since the validation are used for model assessment, you should not filter observations in these data sets.
  - You typically are interested in scoring the outcome of every observation in the score data set. Therefore, you should not filter the score data set.

General Methods of Outlier Detection

- Univariate outliers are cases that have an unusual value for a single variable.
  - For numerical variables, consider values beyond ±3SD or, ±4SD or, ±5SD away from mean as potential outliers.
    - What’s the rationale?
- Multivariate outliers are cases that have an unusual combination of values for a number of variables. The value for any of the individual variables may not be a univariate outlier, but, in combination with other variables, the case is an outlier. This can be detected by
  - Density approach
  - Distance (Mahalanobis) approach

After Outlier Detection

- Change outlier observation
  - Winsorize it
    - Change the value to the highest or lowest range that you want to consider
    - Use the winsorized data in your analysis
- Delete (keep aside) outlier observation
  - Sample size gets reduced

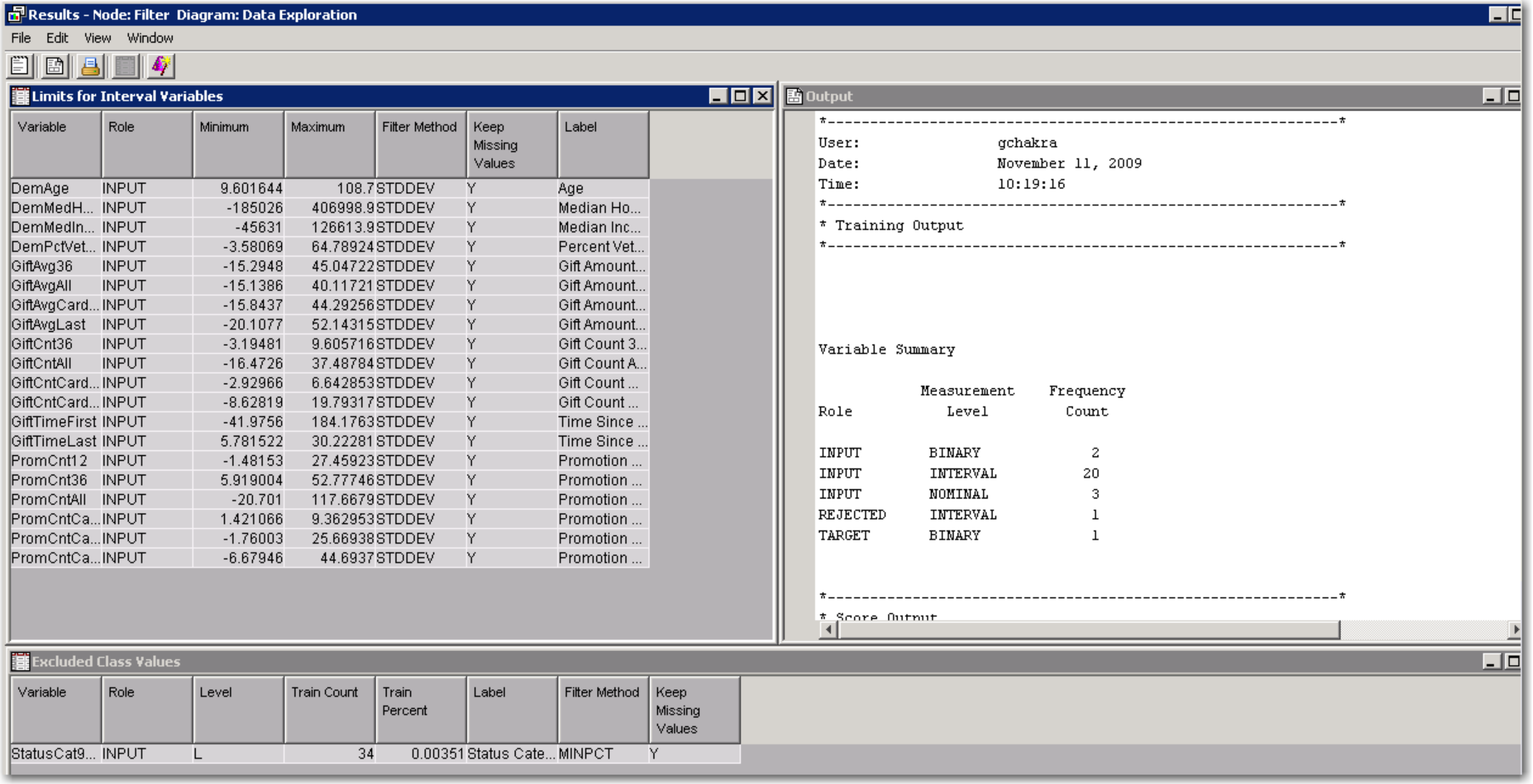
Filter Node (Sample Tab)

- Use the Filter node to create and apply filters to your training data set
  - You can use filters to exclude extreme outliers and errant data that you do not want to include in your analysis so that you can build a more stable (robust) model.
- Filtering can be applied either:
  - Automatically on your data (you can choose different rules for interval/class variables)
  - Or, *interactively* with one variable at a time

Demonstration of Filter Node

- Add a **Filter** node (under **Sample** tab) to the right of the data node.
- Connect the data node to the Filter node
- Check property panel
  - Look at default filtering for interval variables (3 std.dev from mean) and class variables (rare values less than 0.01%)
- Right-Click Filter node, select Run. Examine results.

Filter Results
(An Example Output)



Results - Node: Filter Diagram: Data Exploration

Limits for Interval Variables

Variable	Role	Minimum	Maximum	Filter Method	Keep Missing Values	Label
DemAge	INPUT	9.601644	108.7	STDDEV	Y	Age
DemMedH...	INPUT	-185026	406998.9	STDDEV	Y	Median Ho...
DemMedIn...	INPUT	-45631	126613.9	STDDEV	Y	Median Inc...
DemPctVet...	INPUT	-3.58069	64.78924	STDDEV	Y	Percent Vet...
GiftAvg36	INPUT	-15.2948	45.04722	STDDEV	Y	Gift Amount...
GiftAvgAll	INPUT	-15.1386	40.11721	STDDEV	Y	Gift Amount...
GiftAvgCard...	INPUT	-15.8437	44.29256	STDDEV	Y	Gift Amount...
GiftAvgLast	INPUT	-20.1077	52.14315	STDDEV	Y	Gift Amount...
GiftCnt36	INPUT	-3.19481	9.605716	STDDEV	Y	Gift Count 3...
GiftCntAll	INPUT	-16.4726	37.48784	STDDEV	Y	Gift Count A...
GiftCntCard...	INPUT	-2.92966	6.642853	STDDEV	Y	Gift Count ...
GiftCntCard...	INPUT	-8.62819	19.79317	STDDEV	Y	Gift Count ...
GiftTimeFirst	INPUT	-41.9756	184.1763	STDDEV	Y	Time Since ...
GiftTimeLast	INPUT	5.781522	30.22281	STDDEV	Y	Time Since ...
PromCnt12	INPUT	-1.48153	27.45923	STDDEV	Y	Promotion ...
PromCnt36	INPUT	5.919004	52.77746	STDDEV	Y	Promotion ...
PromCntAll	INPUT	-20.701	117.6679	STDDEV	Y	Promotion ...
PromCntCa...	INPUT	1.421066	9.362953	STDDEV	Y	Promotion ...
PromCntCa...	INPUT	-1.76003	25.66938	STDDEV	Y	Promotion ...
PromCntCa...	INPUT	-6.67946	44.6937	STDDEV	Y	Promotion ...

Output

```
*------------------------------------------------------------*
User:                gchakra
Date:                November 11, 2009
Time:                10:19:16
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

            Measurement    Frequency
Role           Level         Count

INPUT        BINARY            2
INPUT        INTERVAL         20
INPUT        NOMINAL           3
REJECTED     INTERVAL          1
TARGET       BINARY            1


*------------------------------------------------------------*
* Score Output
```

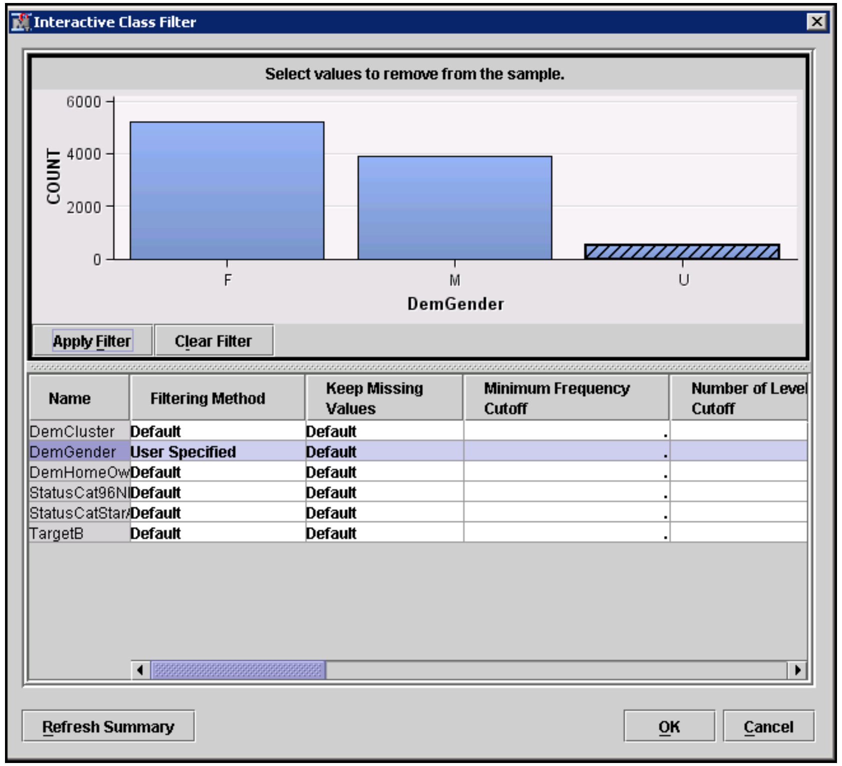
Excluded Class Values

Variable	Role	Level	Train Count	Train Percent	Label	Filter Method	Keep Missing Values
StatusCat9...	INPUT	L	34	0.00351	Status Cate...	MINPCT	Y

Maximize output panel to get a feel for the effect of filtering

Interactive Filtering of a Class Variable

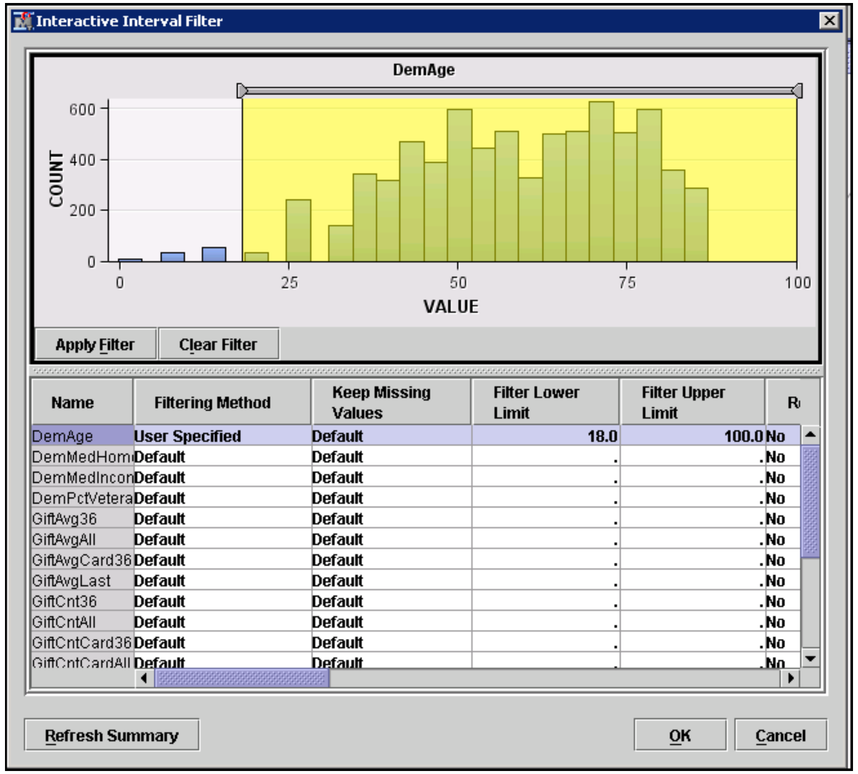
- Click on the **Class Variables ellipsis button** in the property sheet for Filter node
- Select DemGender variable.
- Suppose we want to filter out the U values of this variable.
  - Click on the U value in the graph (it should get shaded).
  - Click Apply Filter button. Note that the Filtering Method changes from Default to User Specified.
  - Click OK.
  - Run Filter node again and take a look at the results



Name	Filtering Method	Keep Missing Values	Minimum Frequency Cutoff	Number of Level Cutoff
DemCluster	Default	Default	.	
DemGender	User Specified	Default	.	
DemHomeOw	Default	Default	.	
StatusCat96NI	Default	Default	.	
StatusCatStar/	Default	Default	.	
TargetB	Default	Default	.	

Interactive Filtering of an Interval Variable

- Click on the **Interval Variables ellipsis button** in the property sheet for Filter node
- Select DemAge variable.
- Suppose we want to filter out all values less than 18 for this variable.
  - Click on the slider on top of the graph and drag the left slider to more than 18. The shaded area of the graph values will be kept after filtering.
  - Click Apply Filter button. Note that the Filtering Method changes from Default to User Specified. Note also the Filter lower limit (we could have directly enter the number in the limit)
  - Click OK.
  - Run Filter node again and take a look at the results



Name	Filtering Method	Keep Missing Values	Filter Lower Limit	Filter Upper Limit	R
DemAge	User Specified	Default	18.0	100.0	No
DemMedHom	Default	Default	.	.	No
DemMedIncon	Default	Default	.	.	No
DemPctVetera	Default	Default	.	.	No
GiftAvg36	Default	Default	.	.	No
GiftAvgAll	Default	Default	.	.	No
GiftAvgCard36	Default	Default	.	.	No
GiftAvgLast	Default	Default	.	.	No
GiftCnt36	Default	Default	.	.	No
GiftCntAll	Default	Default	.	.	No
GiftCntCard36	Default	Default	.	.	No
GiftCntCardAll	Default	Default	.	.	No