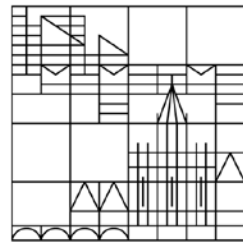




**Nycomed Chair for  
Bioinformatics and Information Mining**

Universität  
Konstanz



# **The New Iris Data – Modular Data Generators**

Iris Adä  
Michael R. Berthold

[Iris.Adae@Uni-Konstanz.de]

ACM SIGKDD 2010, Washington



# Outline

- Motivation
  - Why generate data?
  - What do current generators?
  - Why use modular generators?
- Modular Data Generation
  - Creating clusters (Demo)
  - Creating association rules
  - Combining two motifs
- Outlook





# Motivation

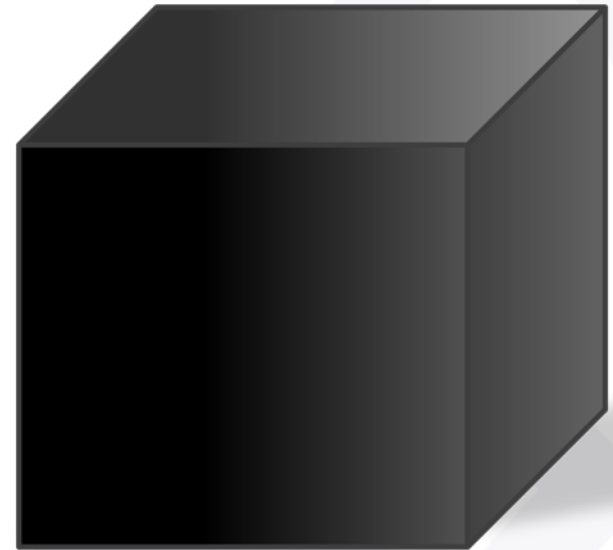
- Reasons to generate artificial data
  - Teaching
  - Method Development, Validation, Testing
  - Demonstration





# Motivation

- **Current tools**
  - Generate data for one method/purpose (cluster, association rules, own algorithm evaluating)
  - Difficult to
    - extend,
    - modify or
    - reuse for other purposes





# Modular Data Generators

- Splitting t

Table "default" - Rows: 3000 Spec - Columns: 6 Properties

Row ID	Date a...	S haircolor	D height	I shoesize	S agency	I nrofjobs
c_5455	05.Jul. 1969	blond	125.317	30	lightC	39
c_5458	02.Aug. 1964	red	230.497	44	redishC	25
c_5463	30.Apr. 1960	brunette	139.96	39	darkC	14
c_5467	05.Nov. 1971	blond	167.017	40	lightC	38
c_5473	18.Aug. 1966	red	200.84	44	redishC	16
c_5481	07.Jan. 1962	red	224.546	42	redishC	15
c_5486	19.Aug. 1988	blond	138.525	41	lightC	50
c_5489	22.Jun. 1960	blond	148.335	35	lightC	31
c_5493	09.Mar. 1950	red	214.431	42	lightC	10
c_5495	15.Jun. 1979	red	125.847	39	redishC	25
c_5505	02.Mar. 1988	red	32.781	37	redishC	22
c_5515	08.Mar. 1952	blond	150.102	37	lightC	21
c_5520	17.Oct. 1964	blond	95.156	39	lightC	16
c_5522	20.Sep. 1984	brunette	75.937	39	darkC	9
c_5526	19.Nov. 1976	blond	171.612	42	lightC	10
c_5531	11.Mar. 1981	blond	61.005	39	lightC	36
c_5539	18.Mar. 1965	blond	173.543	41	darkC	7
c_5543	04.Jul. 1987	blond	40.686	35	lightC	21
c_5546	29.May. 1989	brunette	101.669	37	darkC	19
c_5547	19.Apr. 1956	red	158.543	38	redishC	15
c_5554	25.Feb. 1988	red	96.379	37	redishC	26
c_5559	22.Jul. 1962	blond	128.926	34	lightC	26
c_5560	28.Jan. 1966	blond	206.344	44	lightC	17
c_5562	17.Mar. 1983	blond	82.253	33	lightC	39
c_5571	05.Dec. 1983	brunette	160.624	38	darkC	31
c_5579	24.Oct. 1967	blond	108.098	39	lightC	14
c_5585	08.May. 1975	brunette	124.595	35	darkC	30
c_5591	16.Jul. 1958	blond	145.315	34	lightC	31
c_5599	21.Feb. 1979	brunette	204.536	46	darkC	27
c_5600	26.Apr. 1970	brunette	149.641	31	darkC	7
c_5607	26.Feb. 1964	brunette	126.118	40	darkC	20
c_5615	29.Mar. 1953	blond	118.225	32	lightC	21
c_5619	30.Jan. 1961	brunette	149.374	36	darkC	35
c_5627	11.Sep. 1989	red	217.861	39	redishC	14
c_5636	26.Oct. 1961	red	54.03	39	redishC	28

Empty Table

create 100 i

Time Ge

create dates betw

Gauss Distributed Assigner

resize (height-class)

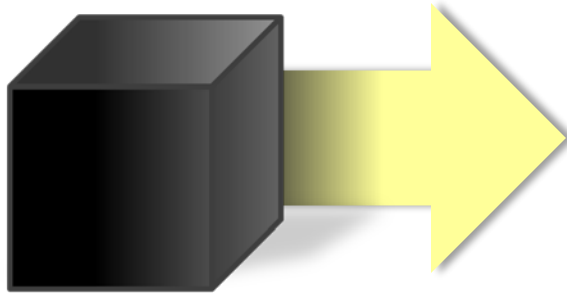
Histogram (interactive)

cy)



# Modular Data Generators

- Splitting the generation into small modules



Modular  
Data  
Generators

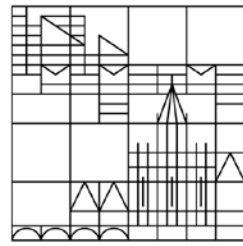
- Each module can then be modified, exchanged and/or extended to meet new requirements
- Integrated in an existing data analysis platform (open source, free, reusable, ...)





**Nycomed Chair for  
Bioinformatics and Information Mining**

Universität  
Konstanz



## **Demo : Creating cluster with MDG's**

The New Iris Data – Modular Data Generators



# Creating cluster using MDG's

The screenshot displays the KNIME software interface. The main workspace shows a workflow with a single node named 'Cluster(gauss)'. The 'Node Repository' on the left lists various nodes, with 'Empty Table Creator' highlighted under the 'Iris Modular Data Generation' category. The 'Empty Table Creator' node is shown in the workspace with a 'Create Rows' button. The right-hand pane displays the configuration dialog for the 'Empty Table Creator' node, which includes the following sections:

- Empty Table Creator**  
Creates an empty file. Just lines with RowIDs.
- Dialog Options**
- Number**  
the number of rows
- Prefix**  
the prefix of the rowkey (before a unique number)
- Skip**  
a random number of rowkeys between 0 and this value will be skipped (if less than 0 no keys will be skipped)
- Seed**  
the random seed
- Ports**
- Output Ports**  
0 Empty table with rowkeys.





# Creating cluster using MDG's

The screenshot displays the KNIME software interface. The main workspace shows a workflow with a single node, 'Empty Table Creator', connected to a 'Cluster(gauss)' node. The 'Empty Table Creator' node is highlighted with a red box. A configuration dialog for the 'Empty Table Creator' node is open, showing the following settings:

- File: Table "default" - Rows: 3000 | Spec - Column: 0 | Properties
- Row ID: A list of row keys including c\_6, c\_13, c\_23, c\_25, c\_26, c\_32, c\_42, c\_52, c\_62, c\_66, c\_76, c\_79, c\_86, c\_90, c\_98, c\_108, c\_117, and c\_124.

The 'Output Ports' section shows the following output:

0 Empty table with rowkeys.



# Creating cluster using MDG's

The screenshot displays the KNIME software interface. On the left, the Node Repository is open, showing the 'Random Label Assigner' node selected under the 'Iris Modular Data Generation' > 'Categorical' category. The main workspace shows a workflow with two nodes: 'Empty Table Creator' (labeled 'Create Rows') and 'Random Label Assigner' (labeled 'Assign Cluster Identity'). The 'Random Label Assigner' node is highlighted, and its dialog options are shown on the right.

## Random Label Assigner

Assigns the labels based on the probabilities to the rows. Here we use the classnames and the probabilities given in the dialog to assign the new class column. category with empty name or probability less or equal 0 will be ignored.

### Dialog Options

**Column Name**  
the name of the new column

**seed**  
the random seed to get a deterministic result

**Name of column**  
The name of the new column

**Probability**  
the probability of this category

### Ports

**Input Ports**



# Creating cluster using MDG's

The screenshot displays the KNIME interface with a workflow in progress. The workflow consists of two nodes: 'Empty Table Creator' (labeled 'Create Rows') and 'Random Label Assigner' (labeled 'Assign Cluster Identity'). The 'Random Label Assigner' node is highlighted, and its configuration dialog is open. The dialog shows the 'Column Name' set to 'clusterID' and a 'seed' of '-5090049309689418586'. A table in the dialog lists categories and their probabilities:

Category	Probability
Cluster0	0.2
Cluster1	0.2
Cluster2	0.2
Cluster3	0.2
Cluster4	0.2

The dialog also includes 'Options' and 'Memory Policy' tabs, and buttons for 'OK', 'Apply', and 'Cancel'.



# Creating cluster using MDG's

The screenshot shows the KNIME software interface. The main window displays a workflow with three nodes: 'Empty Table Creator' (labeled 'Create Rows'), 'Random Label Assigner' (labeled 'Assign Cluster Identity'), and 'Gauss Distributed Assigner' (labeled 'Create 1. Dimension'). A dialog box titled 'Dialog - 0:3 - Gauss Distributed Assigner (Create 1. Dimension)' is open, showing the configuration for the 'Gauss Distributed Assigner' node. The dialog has two tabs: 'Options' and 'Memory Policy'. The 'Options' tab is active, showing the following settings:

- Dependency Column: **S clusterID**
- Column Name: **Dim1**
- Seed:  **1279533108338** (with a 'New' button)
- Bounds min:  -5,000  max:  5,000

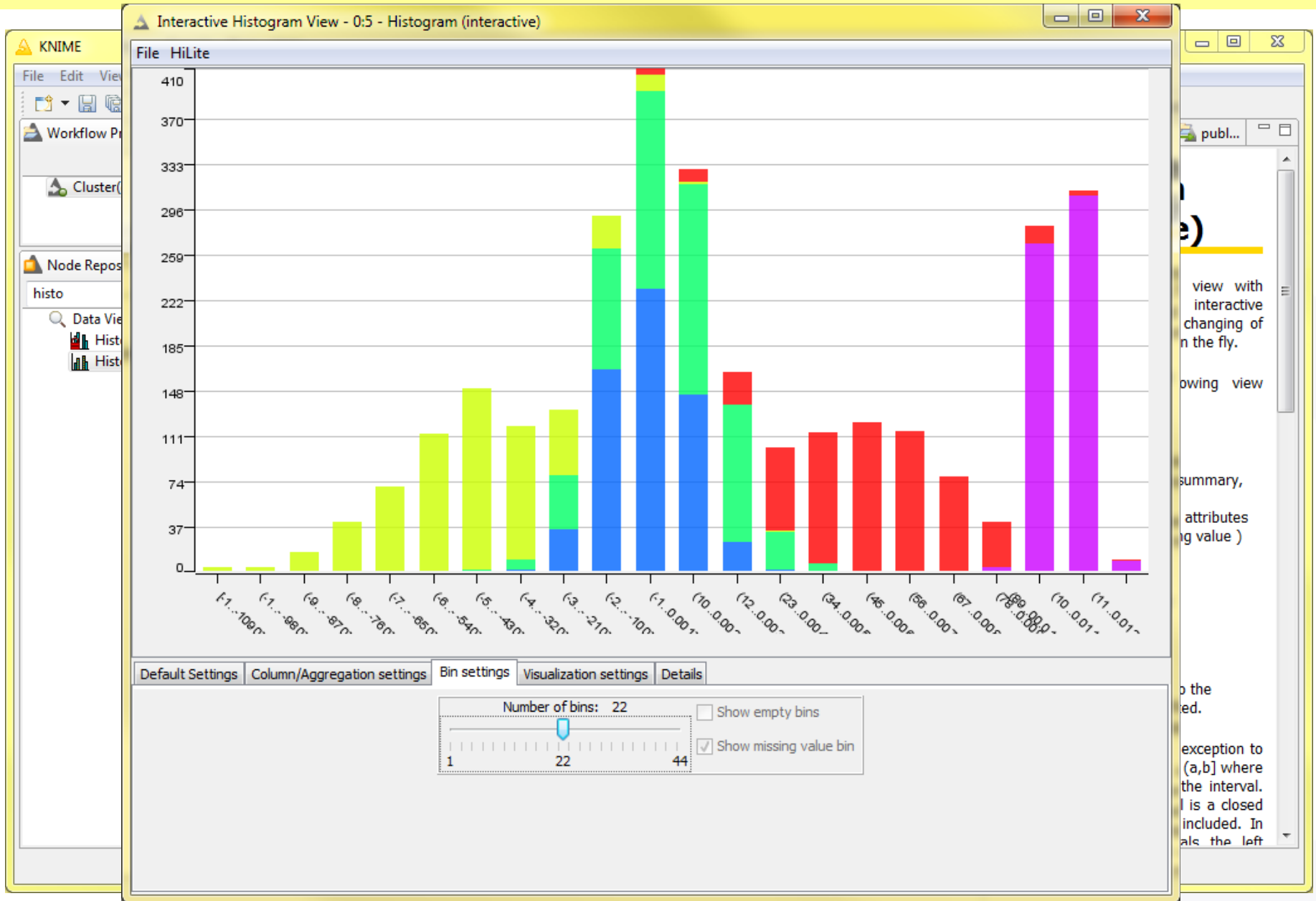
Below the settings is a table with the following data:

Name	Mean	Var
Cluster4	0	150
Cluster0	-50	100
Cluster1	500	200
Cluster2	1,000	50
Cluster3	-500	200

The dialog also has 'OK', 'Apply', and 'Cancel' buttons at the bottom. On the right side of the main window, there is a help panel titled 'Gauss Distributed Assigner' with the following text: 'assigns a value based on the class column, this value is gauss distributed as defined in the configuration by its mean and max'.



# Creating cluster using MDG's





# Creating cluster using MDG's

The screenshot displays the KNIME software interface with a workflow titled "Cluster(gauss)". The workflow consists of the following nodes:

- Empty Table Creator** (labeled "Create Rows")
- Random Label Assigner** (labeled "Assign Cluster Identity")
- Gauss Distributed Assigner** (labeled "Create 1. Dimension")
- Gauss Distributed Assigner** (labeled "Dim2")
- Gauss Distributed Assigner** (labeled "Dim3")
- Color Manager**
- Histogram (interactive)**

The workflow is connected as follows: Empty Table Creator → Random Label Assigner → Gauss Distributed Assigner (Create 1. Dimension) → Color Manager → Histogram (interactive). Additionally, the Gauss Distributed Assigner (Create 1. Dimension) node is connected to two other Gauss Distributed Assigner nodes (Dim2 and Dim3).

The right-hand side of the interface shows the **Gauss Distributed Assigner** dialog options:

## Gauss Distributed Assigner

assigns a value based on the class column, this value is gauss distributed as defined in the configuration by its mean and max

### Dialog Options

**Dependency Column**  
choose a class column

**Category Name**  
the name of the category

**seed**  
the random seed

**Bounds**  
these bounds are used for the whole column.

**values**  
for each value of the given class column, you can define the mean and the variance

**Ports**



# Creating cluster using MDG's

Scatter Matrix - 0:8 - Scatter Matrix

File HiLite

Dim1

Dim2

Dim3

Dim1

Dim2

Dim3

Exclude

Column(s):

Select all search hits

clusterID

Select

add >>

add all >>

<< remove

<< remove all

Include

Column(s):

Select all search hits

Dim1

Dim2

Dim3

## Matrix

element  $E_{ij}$  is at row  $i$ , column  $j$ , where the row is displayed at the x-axis and the column at the y-axis are displayed

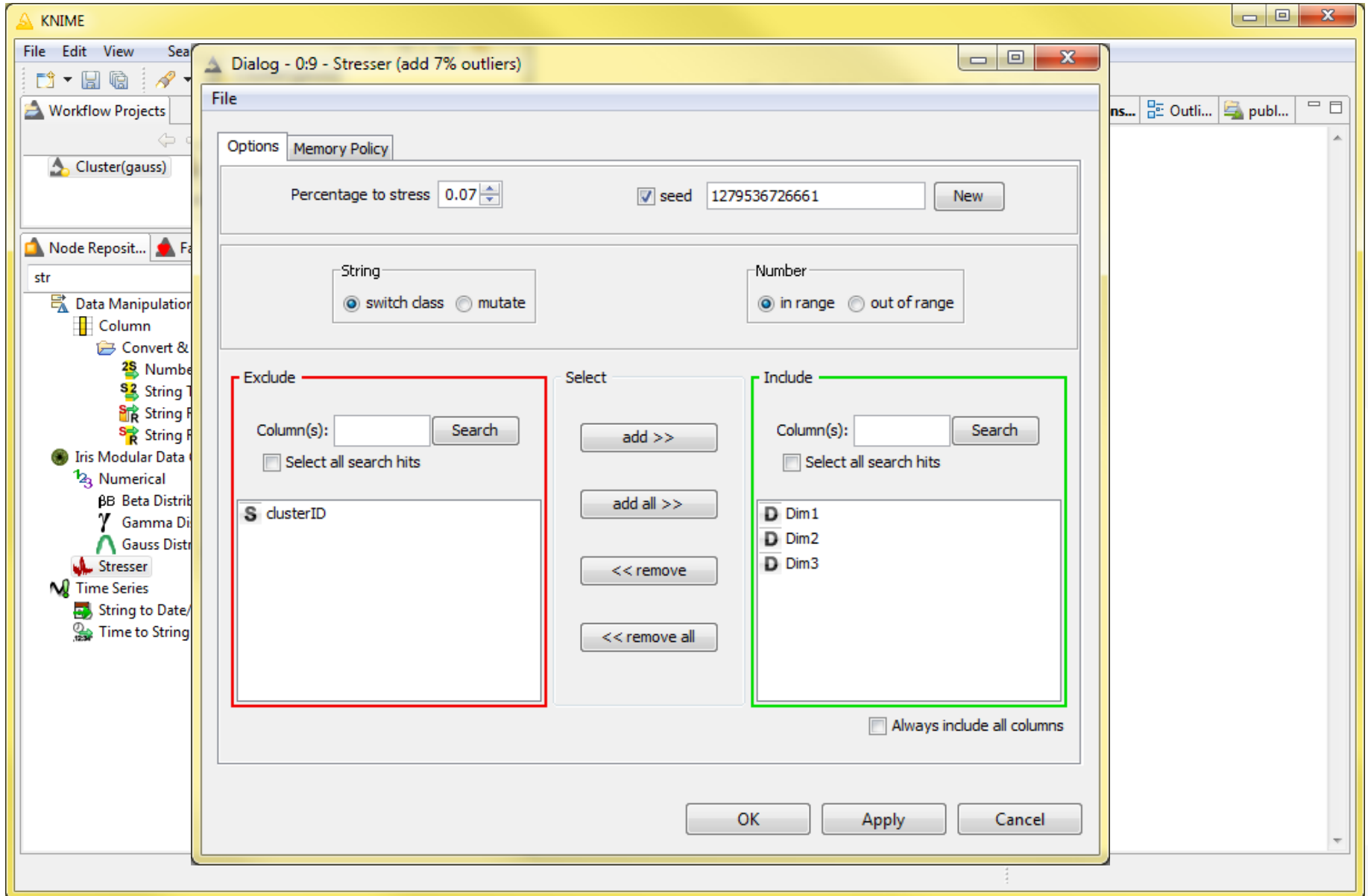
**domain** are **sure that the** **d or set the** **ator node!**

ted by either  
ouse over the  
. Hold control  
The selected  
or right-click to  
hilitate menu in  
e displayed as  
ments.

points to get  
the value.



# Creating cluster using MDG's



The screenshot shows the KNIME interface with the 'Stresser' node selected in the workflow. A dialog box titled 'Dialog - 0:9 - Stresser (add 7% outliers)' is open, showing the 'Memory Policy' tab. The dialog is divided into several sections:

- Options:** 'Percentage to stress' is set to 0.07. A 'seed' checkbox is checked with the value 1279536726661 and a 'New' button.
- String:** Radio buttons for 'switch class' (selected) and 'mutate'.
- Number:** Radio buttons for 'in range' (selected) and 'out of range'.
- Exclude:** A red-bordered box containing a search field with 'clusterID' entered and a 'Search' button. A 'Select all search hits' checkbox is present.
- Select:** A central area with buttons: 'add >>', 'add all >>', '<< remove', and '<< remove all'.
- Include:** A green-bordered box containing a search field and a 'Search' button. A list of columns is shown: 'Dim 1', 'Dim 2', and 'Dim 3'. A 'Select all search hits' checkbox is present.
- Always include all columns:** A checkbox at the bottom right.

At the bottom of the dialog are 'OK', 'Apply', and 'Cancel' buttons.





# Creating cluster using MDG's

The screenshot shows the KNIME Scatter Matrix node interface. The main window displays a 3x3 matrix of scatter plots for dimensions Dim1, Dim2, and Dim3. The diagonal plots show a linear relationship, while the off-diagonal plots show clusters of data points in various colors. The 'Appearance' tab is active, showing an 'Exclude' list with 'clusterID' and an 'Include' list with 'Dim1', 'Dim2', and 'Dim3'.

**Scatter Matrix Data:**

Dimension	Min	Max
Dim1	-1138.526	1160.958
Dim2	0.309	7.683
Dim3	-2.626E3	1.831E3

**Appearance Tab Settings:**

- Exclude:** clusterID
- Include:** Dim1, Dim2, Dim3



# Creating cluster using MDG's

The screenshot displays the KNIME software interface with a workflow titled '\*0: Cluster(gauss)'. The workflow consists of the following nodes:

- Empty Table Creator** (Create Rows)
- Random Label Assigner** (Assign Cluster Identity)
- Gauss Distributed Assigner** (Create 1. Dimension)
- Gauss Distributed Assigner** (Dim2)
- Gauss Distributed Assigner** (Dim3)
- Stresser** (add 7% outliers)
- Normalizer**
- k-Means**
- Color Manager**
- Scatter Matrix**

The right-hand pane provides details for the **k-Means** node:

## k-Means

This node outputs the cluster centers for a predefined number of clusters (no dynamic number of clusters). K-means performs a crisp clustering that assigns a data vector to exactly one cluster. The algorithm terminates when the cluster assignments do not change anymore. The node can be configured as follows:

### Dialog Options

- number of clusters**  
The number of clusters (cluster centers) to be created.
- max number of iterations**  
The number of iterations after which the algorithm terminates, independent of the accuracy improvement of the cluster centers.

### Ports

#### Input Ports

- 0 Input to clustering. All numerical values and only these are considered for clustering.



# Used Modules

Empty Table Creator



- Creates starting table  
(No columns, only rows with identifiers)

Random Label  
Assigner



- Assigns random labels.  
(based on specified labels / probabilities)

Gauss Distributed  
Assigner



- Assigns random numbers.  
(based on Gaussian distribution)

Stresser



adding 7% stress

- Adds noise to a column  
(numerical or categorical outliers)



# Other Modules

Conditional  
Label Assigner



- Assigns conditional random labels (based on values of another column)

Rule Engine

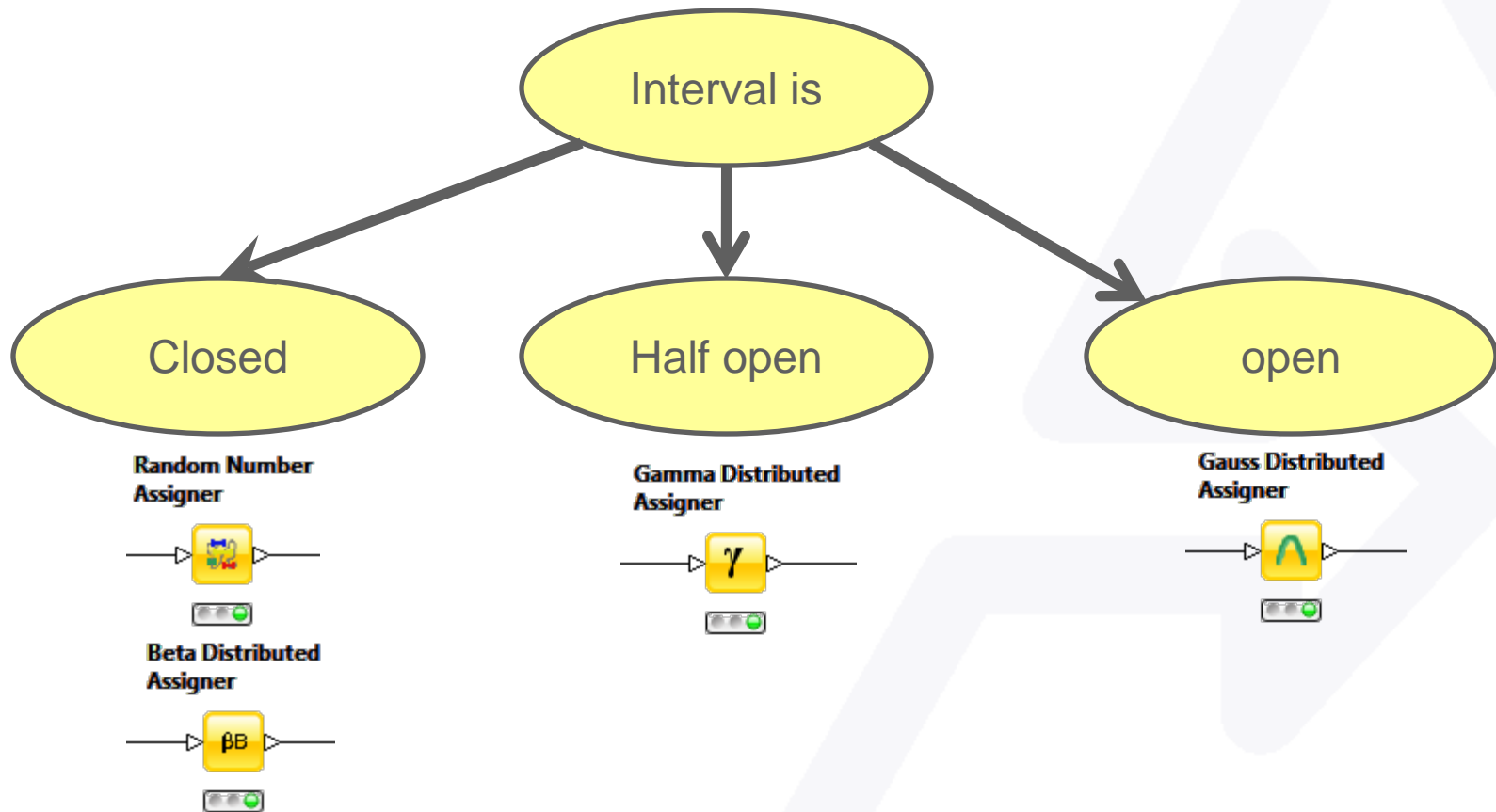


- Inserts various kind of rules



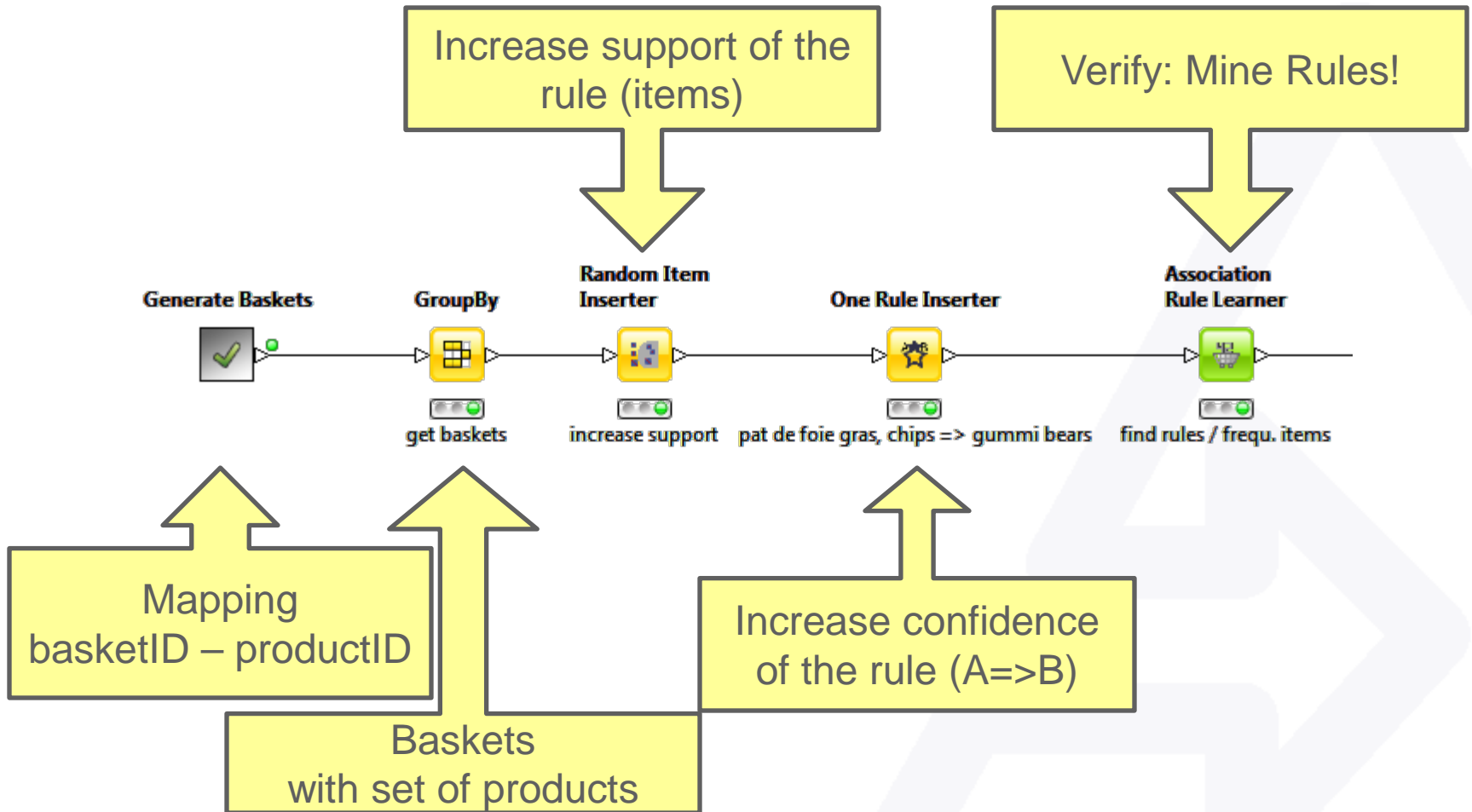
# Other Modules for Random Numbers

- Independent: one distribution for entire column
- Dependent: distribution for each value of a second column





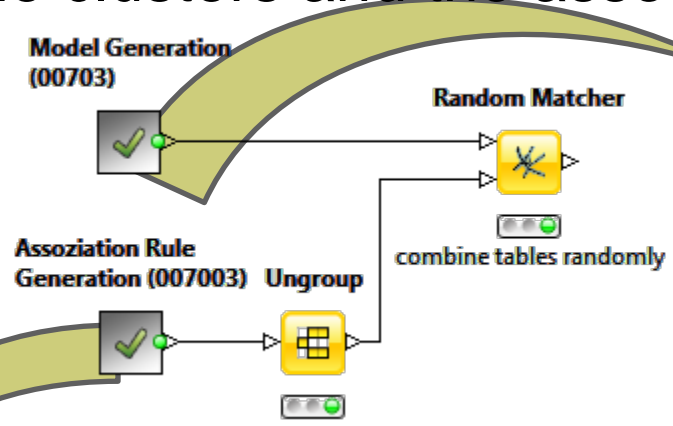
# Association Rules





# Combining tables randomly

- E.g. combine the clusters and the association rules



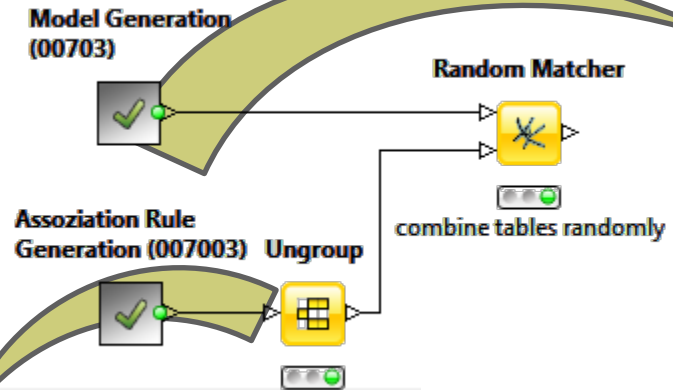
Row ID	S BasketID	(...) pr...
Row270	c_117BID_11	[Bio Coke,Ch...
Row271	c_117BID_12	[Bio Coke,Celery,Cran...
Row272	c_117BID_13	[gummi bears]
Row273	c_117BID_14	[Apple cake,Bio Coke,Champagne Esprit du Siècle,Donuts,chips,H
Row274	c_117BID_15	[Chinese Food,Coke,gummi bears,Mouton Rothschild 1945,T-bon
Row275	c_117BID_16	[Broccoli,Champagne Esprit du Siècle,Chodate cake,Dornfelder(R
Row276	c_117BID_17	[Avocado,Bordeaux (Red Wine),Celery,chips,gummi bears,Jackfr
Row277	c_117BID_18	[Broccoli,Cherimoya,Chinese Food,Coconut,Coke,Ice Tea,Ketchu
Row278	c_117BID_19	[chips,Bordeaux (Red Wine),Broccoli,Chicory,Corn,Ravioli,Sauce
Row279	c_117BID_2	[American cheese,Bio Coke,Bio Coke,Brezels,Chinese Food,Frenc
Row280	c_117BID_20	[Beet,Bordeaux (Red Wine),Chodate cake,Donuts,Hamburger,Tc
Row281	c_117BID_21	[Artichoke,Beet,gummi bears,Bio Coke,apple,Mouton Rothschild 1
Row282	c_117BID_22	[gummi bears]
Row283	c_117BID_23	[Avocado,Avocado,Hamburger ,gummi bears,alkopops(wodka/citr
Row284	c_117BID_24	[Egg sandwich,Ham sandwich,apple,chips,dam chowder,duck]
Row285	c_117BID_25	[Apple cake,Okra,Potatoes,Ranch Dressing Mix,Sauvignon Blanc(
Row286	c_117BID_26	[American cheese,Apple cake,chips,Coconut,Egg sandwich,Frenc
Row287	c_117BID_27	[Apple cake,Apricot,Beet,Bio Coke,Clementine,French fries,apple

Row ID	Date a...	S haircolor	D h...	T sh...	S agency	I
c_1916	07.May. 1965	red	133.64	35	redishC	14
c_1924	21.Feb. 1967	brunette	135.81	40	lightC	27
c_1926	01.Aug. 1981	red	211.936	41	redishC	14
c_1932	23.Apr. 1972	blond	185.083	41	darkC	18
c_1942	09.May. 1956	brunette	155.215	38	darkC	13
c_1950	15.Feb. 1974	brunette	34.312	37	darkC	12
c_1960	25.Aug. 1982	brunette	234.306	44	lightC	18
c_1966	21.Oct. 1980	red	199.217	46	redishC	14
c_1970	03.Jun. 1974	blond	105.104	39	lightC	11
c_1979	03.Mar. 1960	brunette	104.911	36	darkC	17
c_1982	06.Aug. 1973	brunette	219.129	39	darkC	25
c_1986	23.Dec. 1960	brunette	196.091	43	redishC	17
c_1988	11.Sep. 1981	brunette	231.792	41	darkC	15
c_1994	24.Mar. 1951	blond	246.774	46	redishC	21
c_1995	11.Mar. 1978	blond	247.151	44	lightC	17
c_2003	03.Aug. 1966	red	114.957	38	redishC	21
c_2008	26.Nov. 1971	blond	241.238	39	darkC	19
c_2014	17.Jun. 1970	blond	203.772	40	lightC	28



# Combining tables randomly

- E.g. combine the clusters and the association rules



Row ID	S BasketID	S product
Row301_2	c_117BID_4	Cherry ke
Row301_3	c_117BID_4	apple
Row301_4	c_117BID_4	Papaya
Row301_5	c_117BID_4	apples
Row301_6	c_117BID_4	omlett
Row301_7	c_117BID_4	wodka
Row302_1	c_117BID_40	apple
Row302_2	c_117BID_40	Clementine
Row302_3	c_117BID_40	Ham sandwich
Row302_4	c_117BID_40	chips
Row302_5	c_117BID_40	Plum
Row302_6	c_117BID_40	glenfiddich whisky
Row302_7	c_117BID_40	pork haunch
Row302_8	c_117BID_40	white truffles
Row302_9	c_117BID_40	wodka
Row303_1	c_117BID_5	Melon
Row303_2	c_117BID_5	Toast
Row303_3	c_117BID_5	Tuna

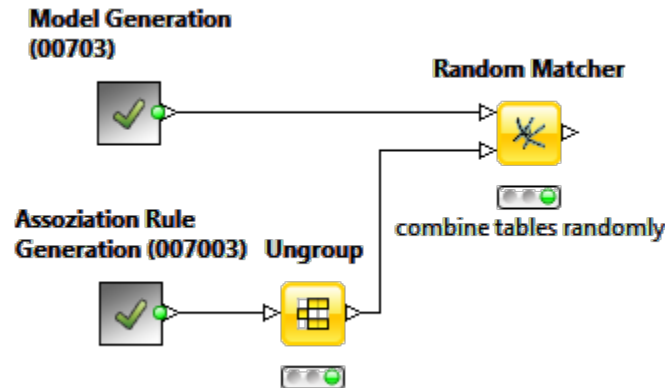
Row ID	Date a...	S haircolor	D height	...	S agency	I nrofj
c_1916	07.May.1965	red	133.64		redishC	14
c_1924	21.Feb.1967	brunette	135.81		lightC	27
c_1926	01.Aug.1981	red	211.936	41	redishC	14
c_1932	23.Apr.1972	blond	185.083	41	darkC	18
c_1942	09.May.1956	brunette	155.215	38	darkC	13
c_1950	15.Feb.1974	brunette	34.312	37	darkC	12
c_1960	25.Aug.1982	brunette	234.306	44	lightC	18
c_1966	21.Oct.1980	red	199.217	46	redishC	14
c_1970	03.Jun.1974	blond	105.104	39	lightC	11
c_1979	03.Mar.1960	brunette	104.911	36	darkC	17
c_1982	06.Aug.1973	brunette	219.129	39	darkC	25
c_1986	23.Dec.1960	brunette	196.091	43	redishC	17
c_1988	11.Sep.1981	brunette	231.792	41	darkC	15
c_1994	24.Mar.1951	blond	246.774	46	redishC	21
c_1995	11.Mar.1978	blond	247.151	44	lightC	17
c_2003	03.Aug.1966	red	114.957	38	redishC	21
c_2008	26.Nov.1971	blond	241.238	39	darkC	19
c_2014	17.Jun.1970	blond	203.777	40	lightC	28





# Combining tables randomly

- E.g. combine the clusters and the association rules

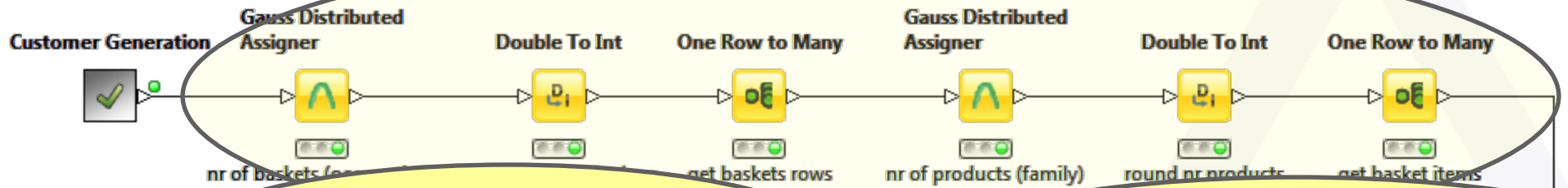


Row ID	Date a...	S haircolor	D height	I shoesize	S agency	I nrofjobs	S BasketID	S product
c_1916	07.May. 1965	red	133.64	35	redishC	14	c_7358ID_5	turkey
c_1924	21.Feb. 1967	brunette	135.81	40	lightC	27	c_5368ID_4	Jackfruit
c_1926	01.Aug. 1981	red	211.936	41	redishC	14	c_8688ID_6	Coke
c_1932	23.Apr. 1972	blond	185.083	41	darkC	18	c_6138ID_21	Sharon Fruit (Persimmon)
c_1942	09.May. 1956	brunette	155.215	38	darkC	13	c_6738ID_10	Persimmon
c_1950	15.Feb. 1974	brunette	34.312	37	darkC	12	c_6498ID_2	chips
c_1960	25.Aug. 1982	brunette	234.306	44	lightC	18	c_10058ID_3	lamm careé
c_1966	21.Oct. 1980	red	199.217	46	redishC	14	c_2848ID_2	liquorice
c_1970	03.Jun. 1974	blond	105.104	39	lightC	11	c_1628ID_12	gummibears
c_1979	03.Mar. 1960	brunette	104.911	36	darkC	17	c_6928ID_2	Apple Juice
c_1982	06.Aug. 1973	brunette	219.129	39	darkC	25	c_1798ID_2	mussels
c_1986	23.Dec. 1960	brunette	196.091	43	redishC	17	c_9748ID_11	ostrich eggs
c_1988	11.Sep. 1981	brunette	231.792	41	darkC	15	c_9868ID_9	chicken
c_1994	24.Mar. 1951	blond	246.774	46	redishC	21	c_5078ID_5	milk lactosefree
c_1995	11.Mar. 1978	blond	247.151	44	lightC	17	c_6928ID_49	Cherry coke
c_2003	03.Aug. 1966	red	114.957	38	redishC	21	c_598ID_1	wodka
c_2008	26.Nov. 1971	blond	241.238	39	darkC	19	c_2608ID_2	pork
c_2014	17.Jun. 1970	blond	203.722	40	lightC	28	c_4378ID_10	Riesling(White wine)
c_2018	07.Apr. 1983	blond	217.181	43	lightC	47	c_3948ID_4	Jackfruit
c_2027	21.Apr. 1975	brunette	100.422	30	darkC	16	c_2218ID_5	Coke



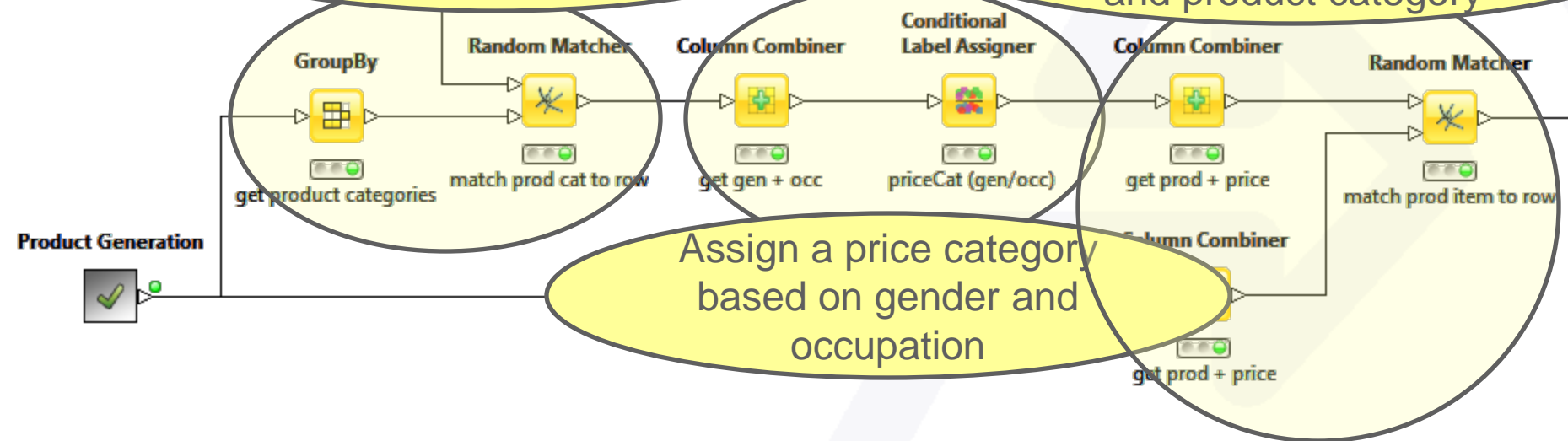
# Generate Customer Data

Generate one row per bought item



Assign a product category randomly

Assign a random real product with matching price and product category



Assign a price category based on gender and occupation



Other things you may want to know but I did not have time to talk about today...

- All generators use (user specifiable) random seeds:
  - Deterministic results
  - Allows to generate different data sets following the same pattern.
- KNIME Workflow variables allow to control global settings
- Batch Execution for automatic data generation



# Summary / Outlook

## Modular Data Generation in KNIME:

- Visual documentation of complex data generation processes
- Repeatability
- Extendibility

## Ongoing Work:

- Add functionality to generate time-dependent data

## Available now (nodes and workflows):

<http://www.knime.org/datageneration>

(demo tonight at the poster)