1 Subject

Implementing K-Means Clustering Algorithm with various Data Mining Tools.

K-means is a clustering (unsupervised learning) algorithm[1]. The aim is to create homogeneous subgroups of examples. The individuals in the same subgroup are similar; the individuals in different subgroups are as different as possible.

The K-Means approach is already described in several tutorials (http://data-mining-tutorials.blogspot.com/search?q=k-means). The goal here is to compare its implementation with various free tools. We study the following tools: **Tanagra 1.4.28**; **R 2.7.2** without additional package; **Knime 1.3.5**; **Orange 1.0b2** and **RapidMiner Community Edition**.

The steps of the data analysis are the following:

- Importing the data file;
- Computing some descriptive statistics indicators;
- Standardizing the variables;
- Implementing the k-means algorithm on the standardized variables;
- Visualizing the cluster membership of each individual;
- Interpreting the clusters with conditional descriptive statistics indicators or graphical representations;
- Comparing the clusters with a pre-specified grouping defined by an illustrative categorical variable;
- Exporting the dataset in a file, including the new cluster membership column.

These steps are usual in a clustering approach. **The main interest of this tutorial is to show that we can implement these steps whatever the tools used**. Of course, I cannot master the functionalities of all the tools. Sometimes perhaps I do not use the most efficient procedure in some situations.

2 Dataset

We use the « cars_dataset.txt »[2] data file. It describes the characteristics of 392 vehicles. The active variables, which participate to the creation of the clusters, are the consumption (MPG), the DISPLACEMENT, the HORSEPOWER, the WEIGHT and the ACCELERATION. The illustrative variable, which is used only to strengthen the interpretation of the clusters, is ORIGIN (Japan, Europe and USA).

3 K-Means with TANAGRA

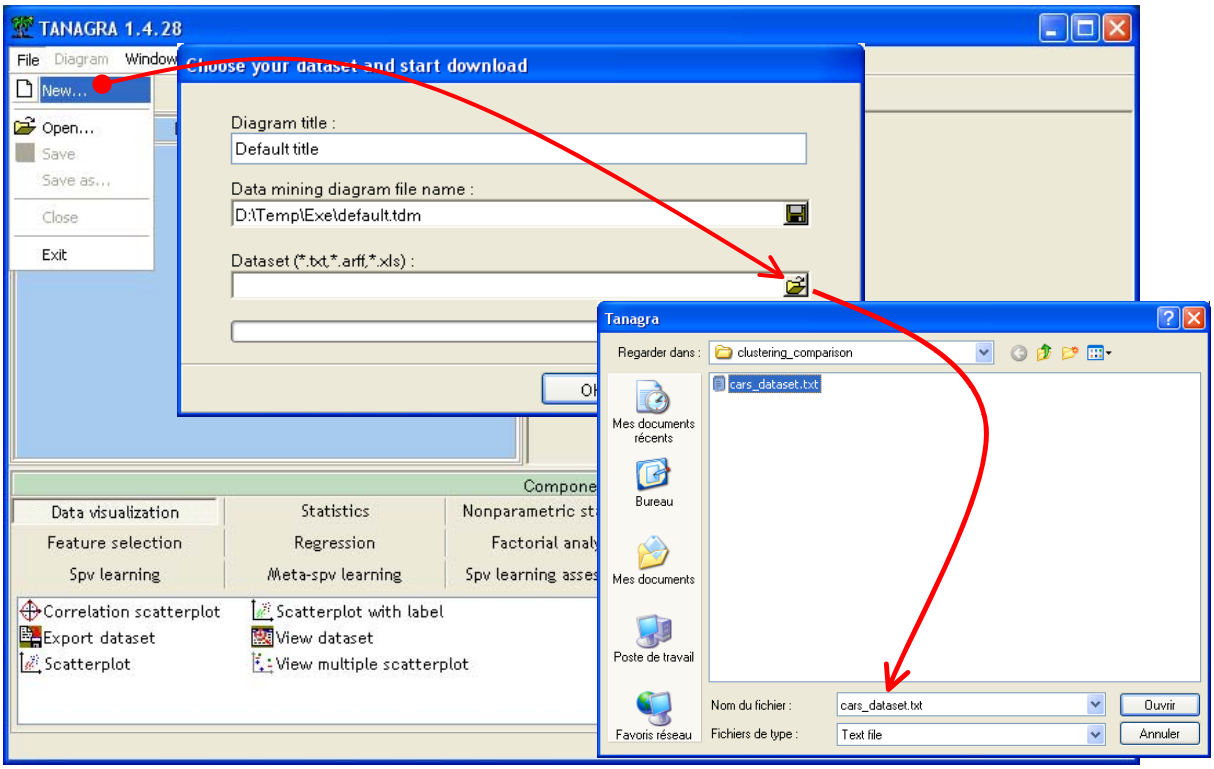
In this section, we give the details of operations with Tanagra. We give only the instruction and the resulting output for the other tools.

3.1 Creating a diagram and importing the dataset

After we launch Tanagra, we click on the FILE / NEW menu in order to create a new diagram. We select the CARS_DATASET.TXT data file.

[1] http://faculty.chass.ncsu.edu/garson/PA765/cluster.htm

[2] http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/cars_dataset.zip ; from the STATLIB server, http://lib.stat.cmu.edu/datasets/cars.desc



392 observations and 6 variables are loaded.

Dataset description

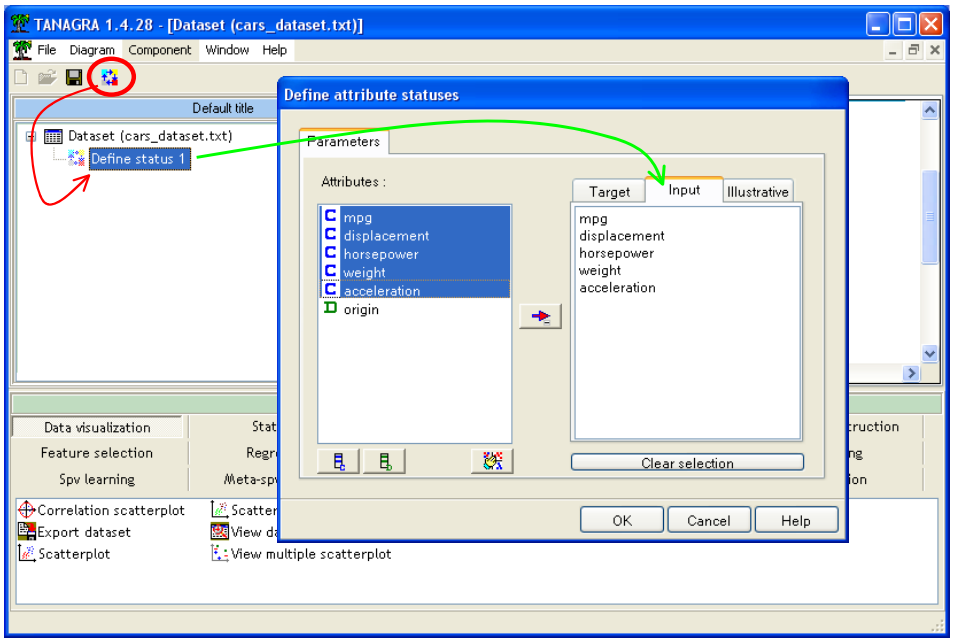
6 attribute(s)
392 example(s)

Attribute	Category	Informations
mpg	Continue	-
displacement	Continue	-
horsepower	Continue	-
weight	Continue	-
acceleration	Continue	-
origin	Discrete	3 values

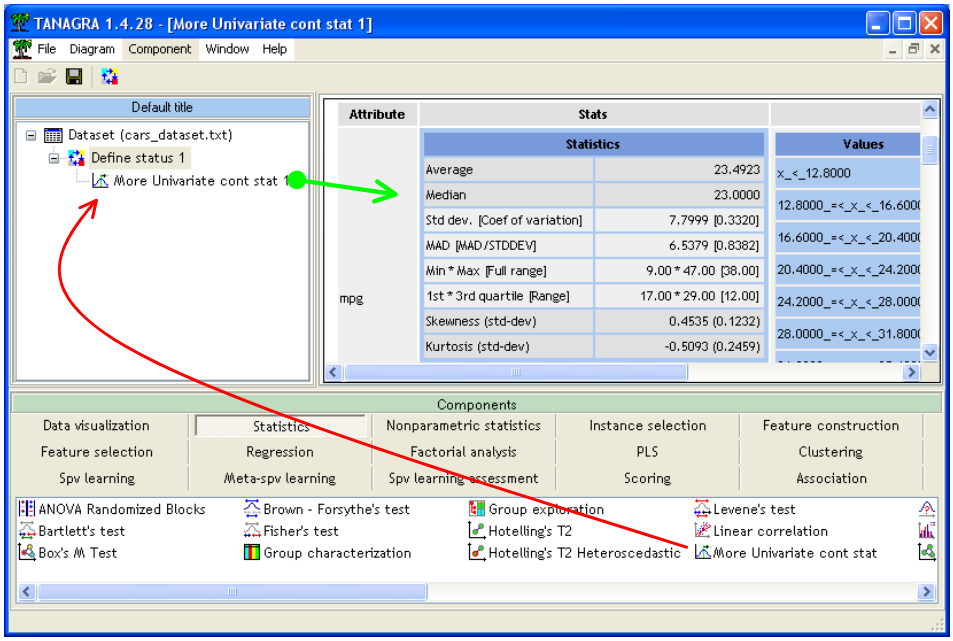
3.2 Descriptive statistics

We want to obtain an overview of the main characteristics of the dataset. We add a DEFINE STATUS component into the diagram. We set all the continuous variables as INPUT.

These are the active variables of the analysis i.e. they are used during the clustering process.



We add the MORE UNIVARIATE CONT STAT component (STATISTICS tab). We click on the contextual VIEW menu.

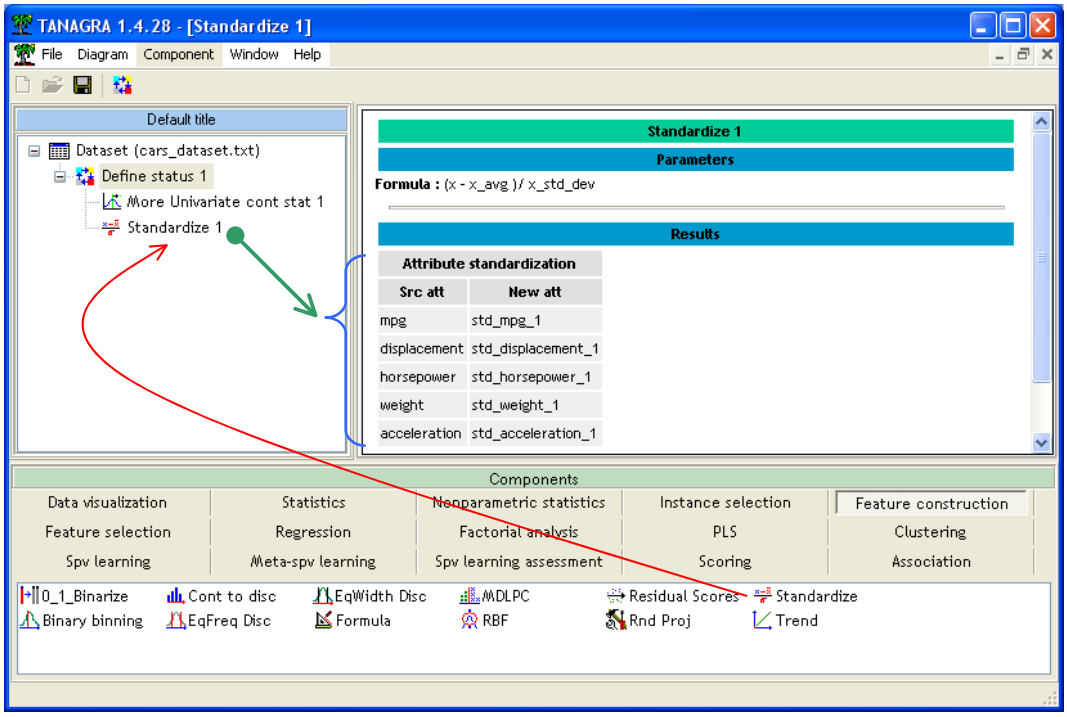


It seems that there are not anomalies or something which requires a specific pre-treatment in our dataset.

3.3 Standardizing the active variables

We want to standardize the variables before performing the k-means approach. The aim is to eliminate the discrepancy of scales between the variables[3]. We add the STANDARDIZE component (FEATURE CONSTRUCTION tab) into the diagram. Then, we click on the VIEW menu.

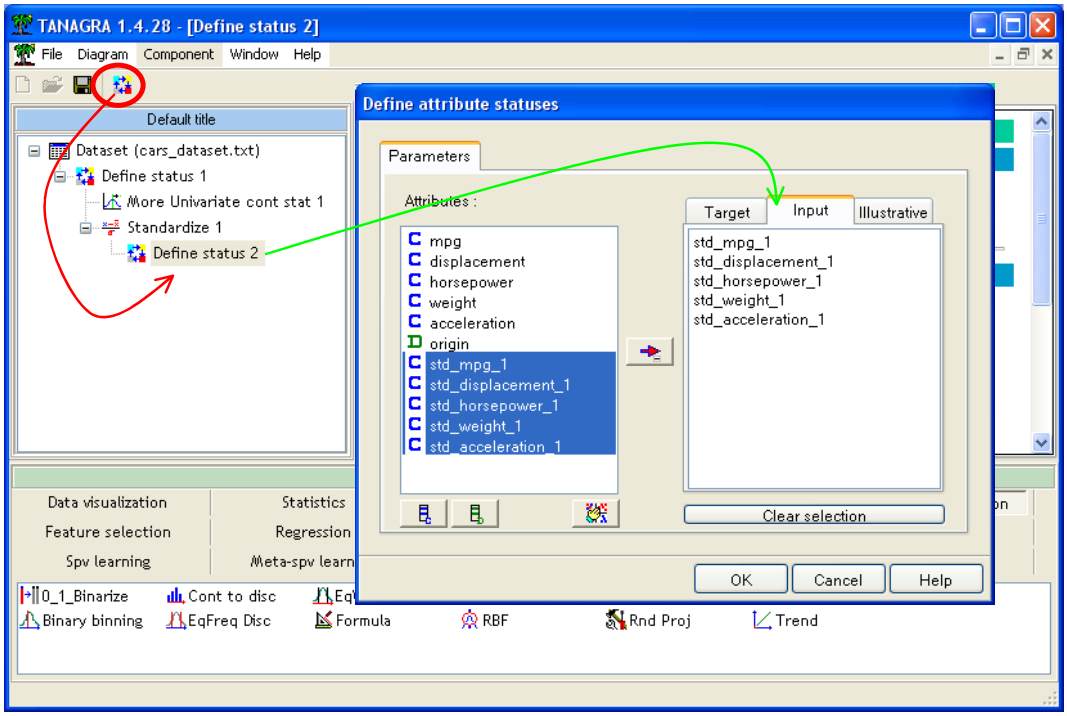
[3] In fact, this operation is not necessary with Tanagra. It can automatically standardize the variables with the K-Means component. We use explicitly this step for the comparison with the other tools.



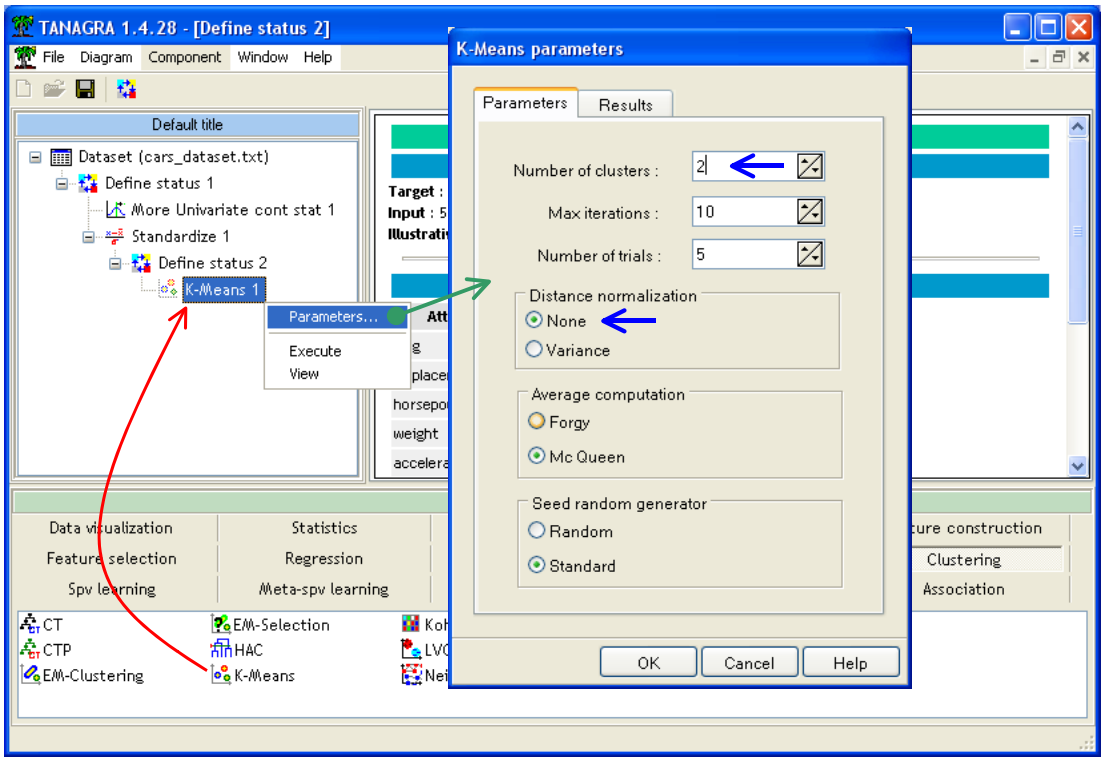
5 new variables are now available for the further processing.

3.4 K-Means

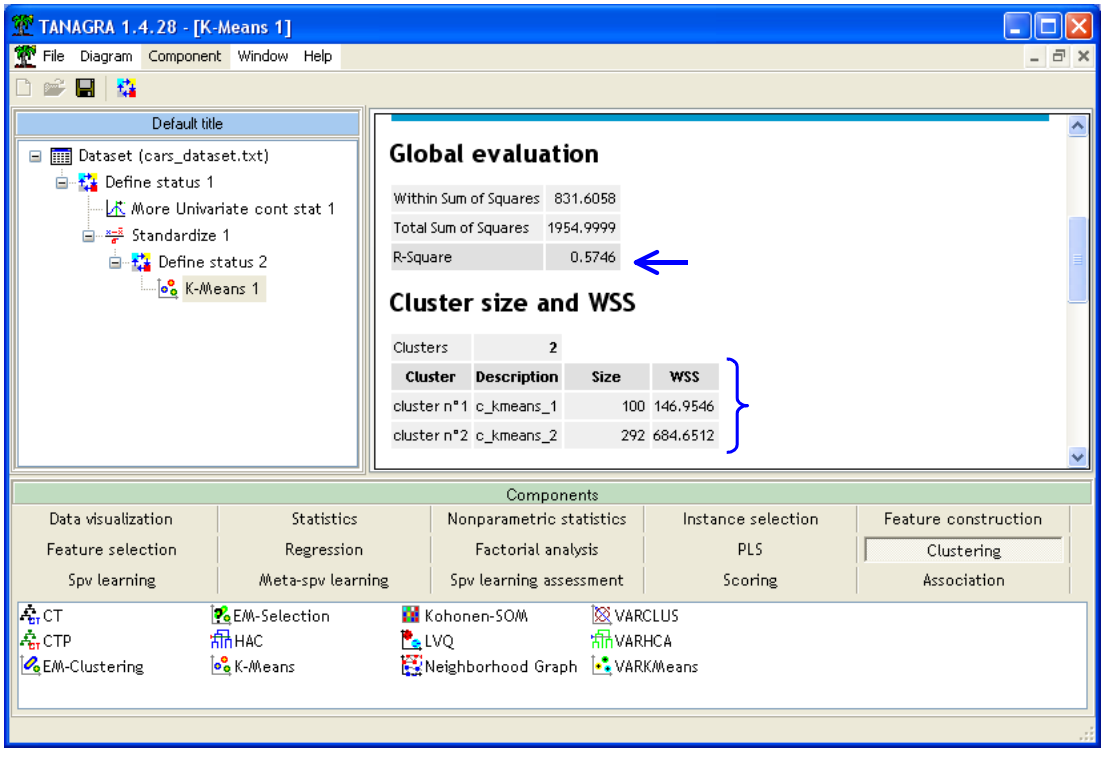
We want to use these transformed variables for the analysis. We insert a new DEFINE STATUS component into the diagram. We set as INPUT the computed attributes (from STD_MPG_1 to STD_ACCELERATION_1).



We insert the K-MEANS component (CLUSTERING tab). We click on the PARAMETERS contextual menu. We set the following parameters.



We ask a partitioning into two groups. It is not necessary to normalize the distance because we use already standardized variables. We validate and we click on the VIEW menu.



The TSS (Total sum of squares) is 1954.9999; the WSS (Within sum of squares) is 831.6058. The BSS (Between sum of squares) explained by the partitioning is (1954.9999 - 831.6058) = 1123.3941. The resulting ratio is (1123.3941 / 1954.9999) =57.46%.

There are 100 examples in the first cluster; 292 examples in the second one.

In the low part of the window, the CLUSTERS CENTROIDS section gives the average for each variable according to the clusters.

Cluster centroids

Attribute	Cluster n°1	Cluster n°2
std_mpg_1	-1.120825	0.383844
std_displacement_1	1.451760	-0.497178
std_horsepower_1	1.459572	-0.499853
std_weight_1	1.355324	-0.464152
std_acceleration_1	-1.018068	0.348653

Use GROUP CHARACTERIZATION for detailed comparisons

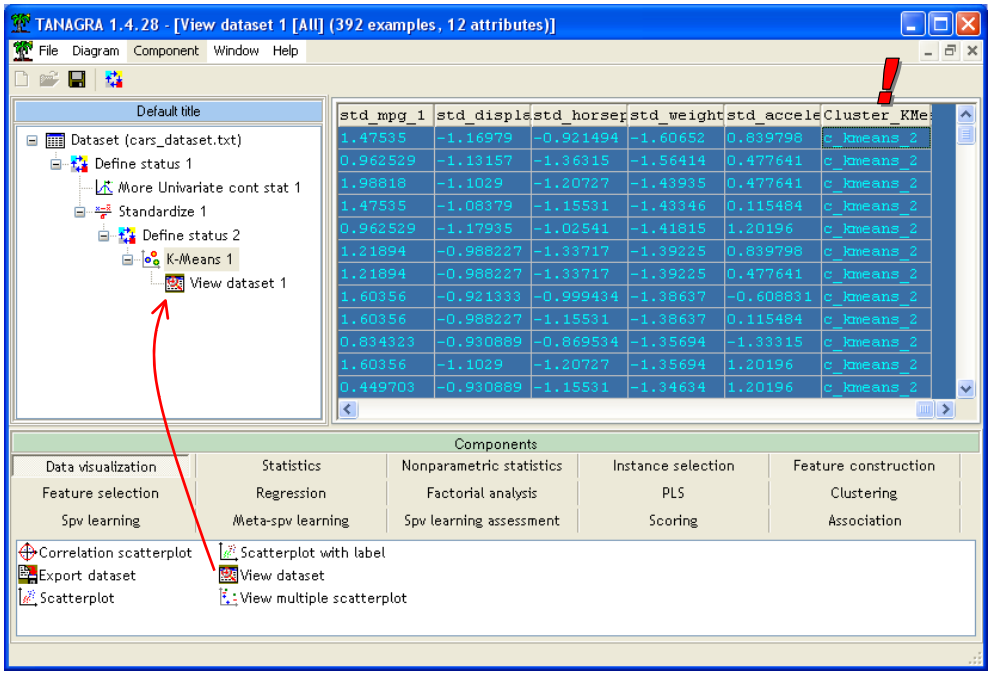
3.5 Interpretation of groups

We are now in the major step of the clustering process: we want to interpret the groups. What the characteristics of each cluster? What differentiate each others?

3.5.1 Group membership of individuals

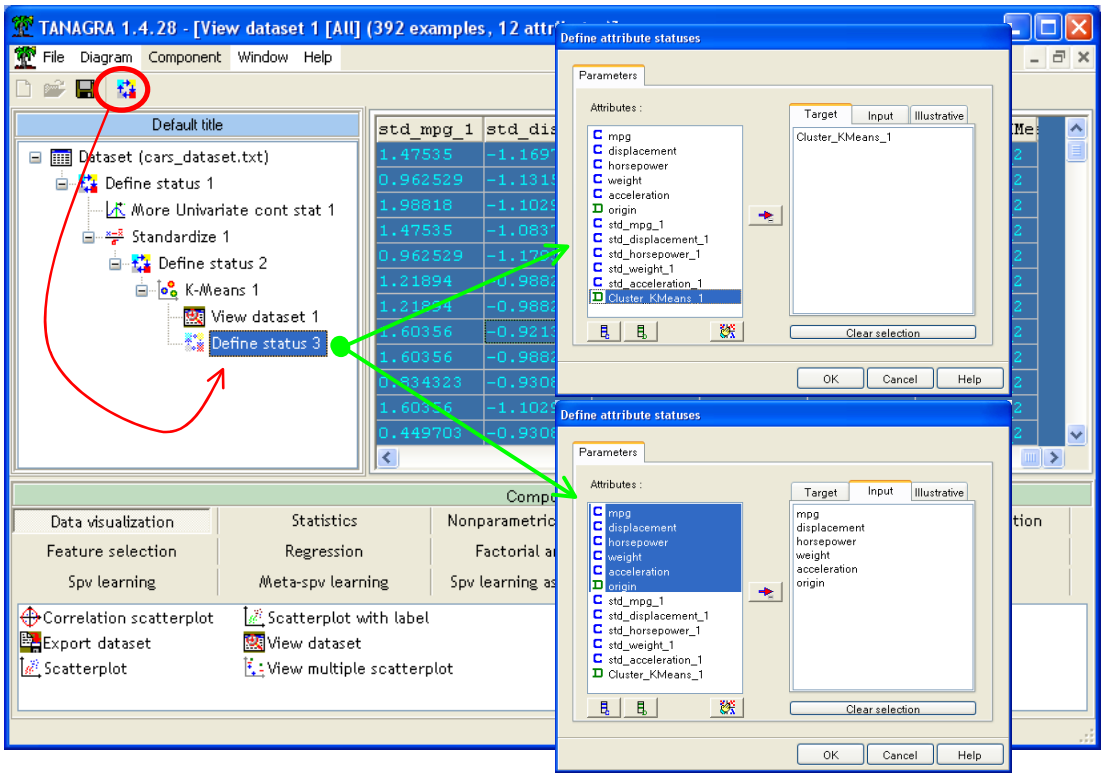
We can inspect the group membership of each individual. This approach is especially useful if we deal with a small dataset and if we can identify each instance (e.g. each individual is labeled).

TANAGRA computes and adds automatically a new column to the current dataset. We can visualize it with the VIEW DATASET component (DATA VISUALIZATION tab).



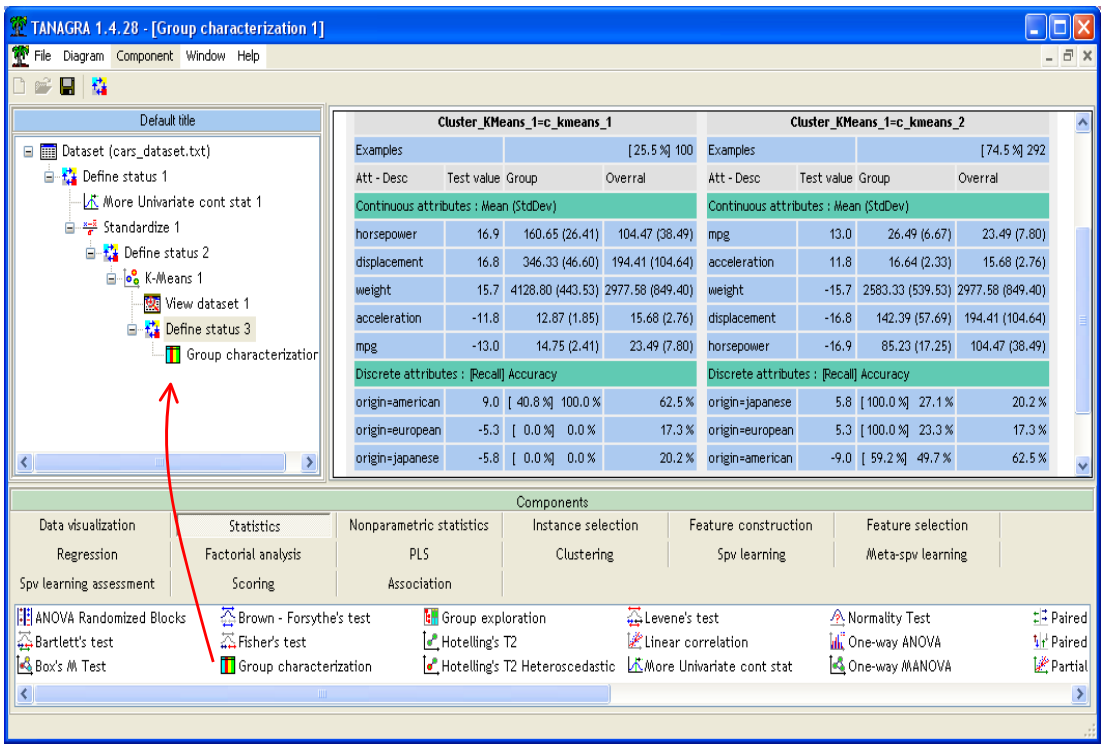
3.5.2 Conditional descriptive statistics

Another approach, more useful, is to compute the descriptive statistics indicators according to the cluster. By comparing them, we can understand the main characteristics of each cluster i.e. what are the variables which allow to differentiate the clusters.



We insert the DEFINE STATUS component into the diagram. We set as TARGET the computed column (CLUSTER_KMEANS_1), as INPUT the other attributes, including the illustrative variable (ORIGIN).

Then we add the GROUP CHARACTERIZATION component (STATISTICS tab).



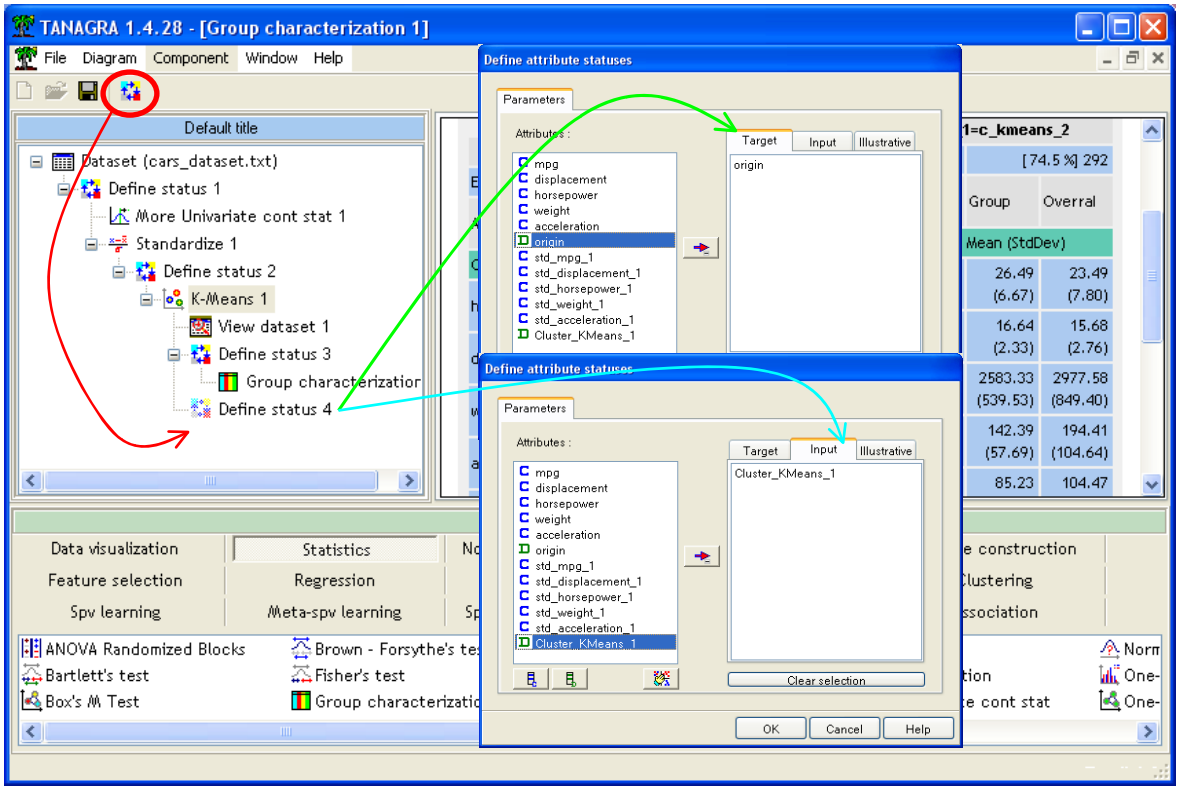
We note that the second cluster (C_K_MEANS_2) corresponds mainly to small cars with low consumption (the mean of MPG is 26.66 into the group while it is 23.49 in the whole dataset), with a small DISPLACEMENT, etc.

In order to characterize the strength of the difference, we use the "test value" criterion (http://data-mining-tutorials.blogspot.com/2009/05/understanding-test-value-criterion.html).

We can use either the active or the illustrative variables in order to characterize the groups. In our dataset, we use the ORIGIN variable for the group interpretation. We note for instance that the first cluster (C_K_MEANS_1) is only constituted of American cars. They have a high consumption (MPG is 14.75 into the group), etc.

3.5.3 Cross tabulation between the group membership and an illustrative variable

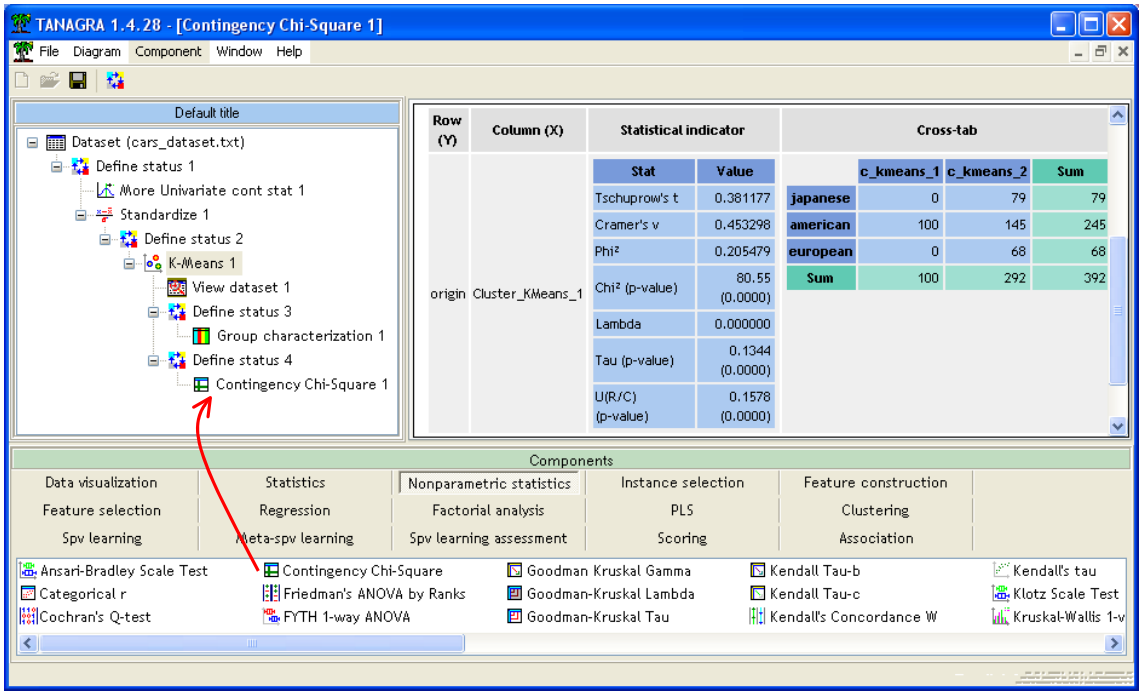
We can also highlight the association between the clusters membership and an illustrative variable using a cross tabulation. We insert a DEFINE STATUS component. We set ORIGIN as TARGET and C_KMEANS_1 as INPUT.



We add the CONTINGENCY CHI-SQUARE component (NONPARAMETRIC STATISTICS tab) into the diagram. We click on the VIEW menu.

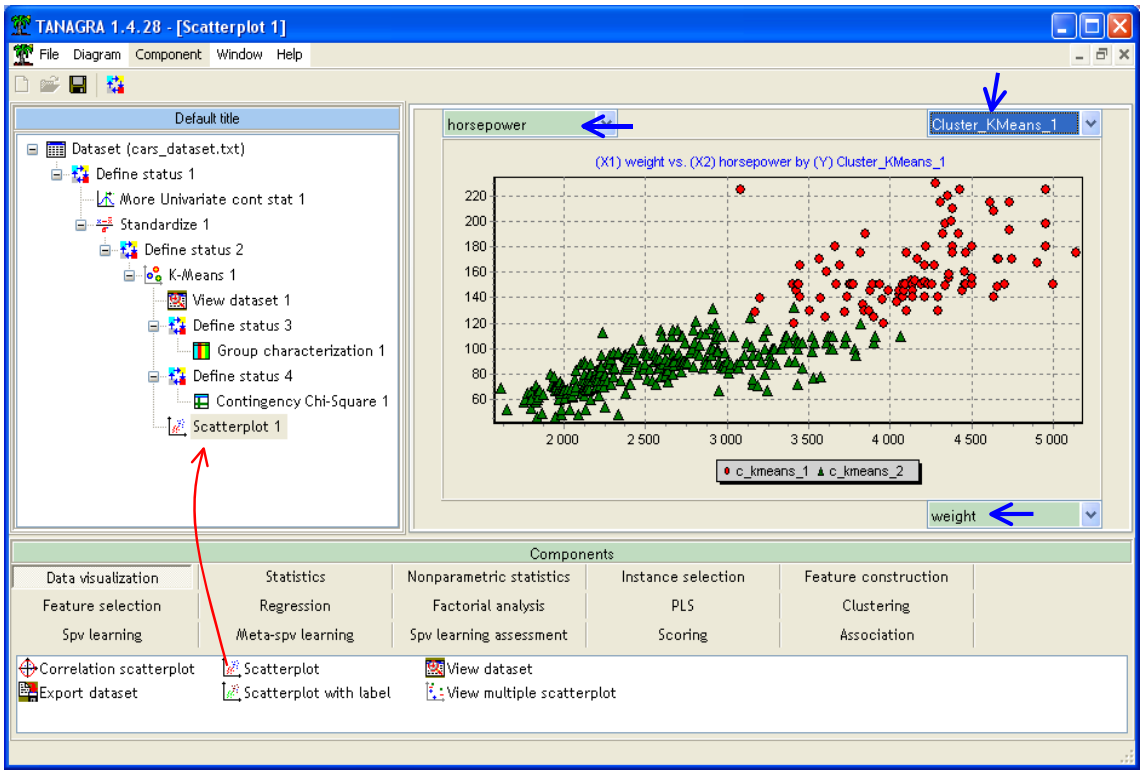
The results are of course consistent with those of the GROUP CHARACTERIZATION component. We have here more information about the strength of the association. Some statistical indicators such as the "Cramer's v" and so on are available. We can check if the association is statistically significant.

We can also display the results in the row or column percentage.



3.5.4 Scatter plot

Another way to highlight the results is the graphical representation. The scatter plot is a very useful tool in this context[4]. We can position the groups according two variables simultaneously. Thus we can check if there are interactions between variables.



[4] http://en.wikipedia.org/wiki/Scatter_plot

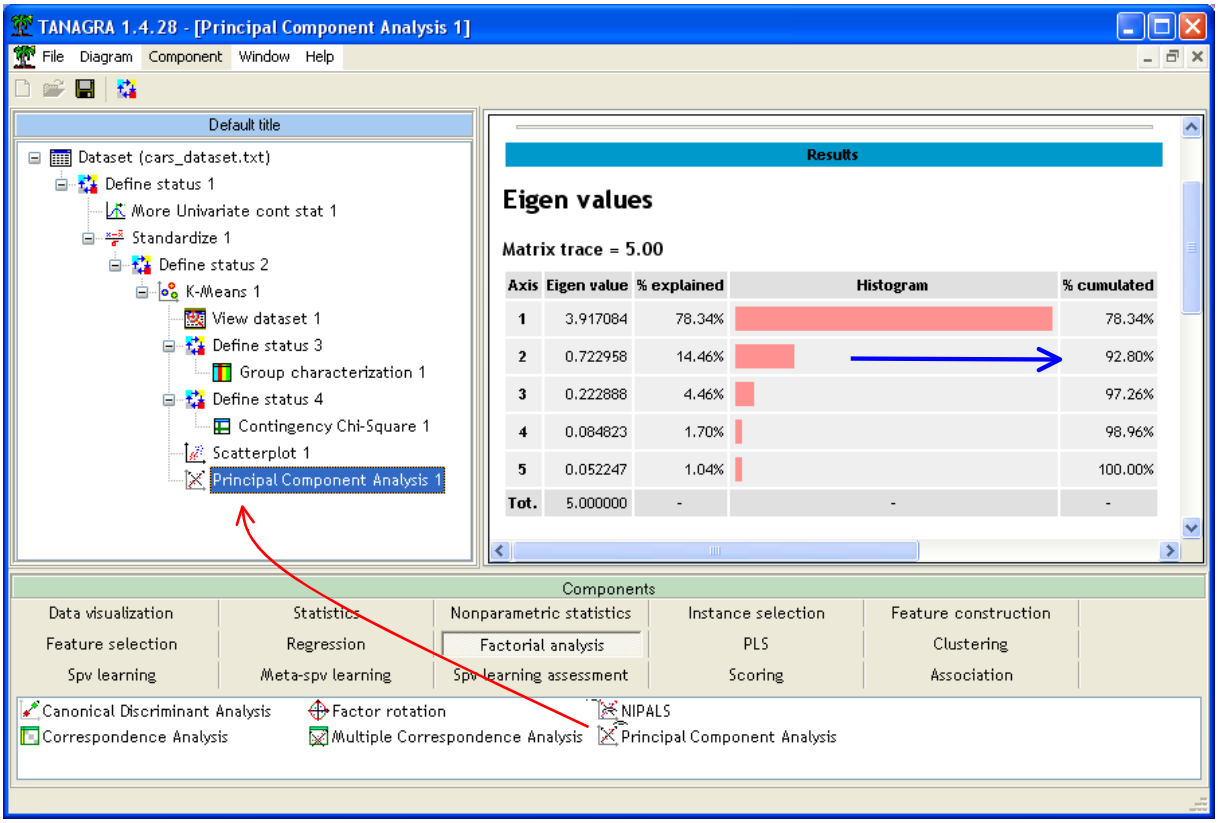
We add the SCATTERPLOT component (DATA VISUALIZATION tab). We click on the VIEW menu. We set WEIGHT on the horizontal axis, HORSEPOWER on the vertical axis. We use the cluster membership to colorize the points.

3.5.5 Graphical representation using principal component analysis

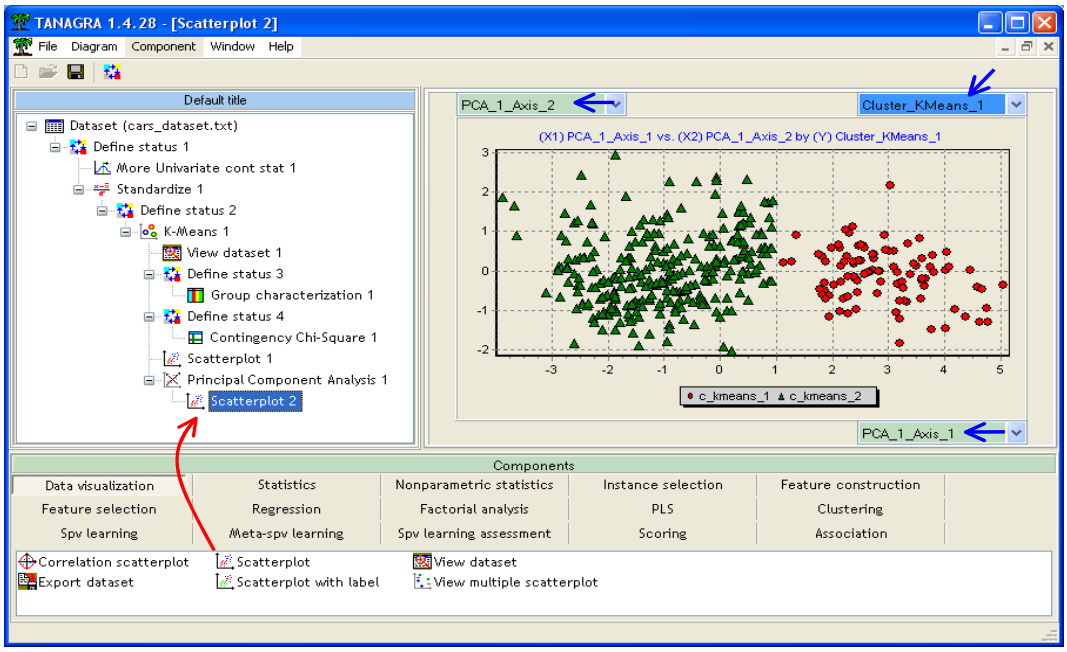
In order to take in consideration the interactions between more than two variables, we can use a principal component analysis (PCA) and set a graphical representation in the first two factors. If these axes are relevant, the relative localization of the groups in this representation space is quite faithful of their localization in the original space.

We add the PRINCIPAL COMPONENT ANALYSIS component (FACTORIAL ANALYSIS tab) after the K-MEANS 1 component. Thus they use the same active variables. We click on the VIEW menu.

The first two factors account 92.8% of the variation into the dataset. On the first factor, we have an opposition between the cars (1) with low consumption (MPG), not very fast (ACCELERATION), and (2) those which are powerful and heavy (HORSEPOWER, WEIGHT).



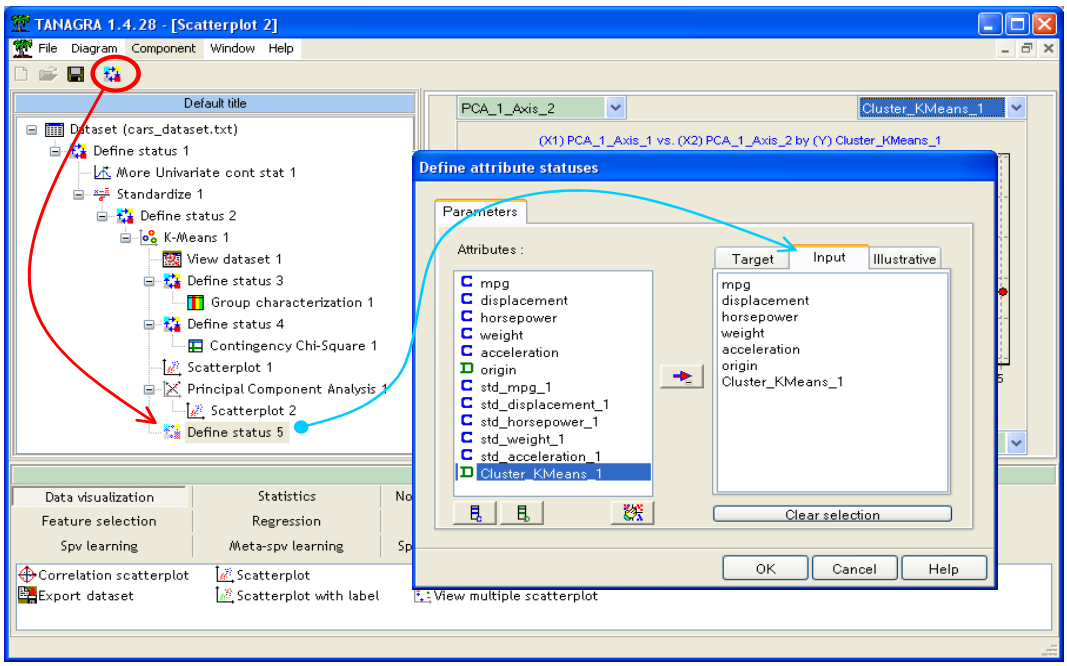
When we create a scatter plot and set to the horizontal axis the first factor (PCA_1_AXIS_1), to the vertical axis the second factor (PCA_1_AXIS_2), we note that the clusters are really distinct.



3.6 Exporting the dataset including the CLUSTER column

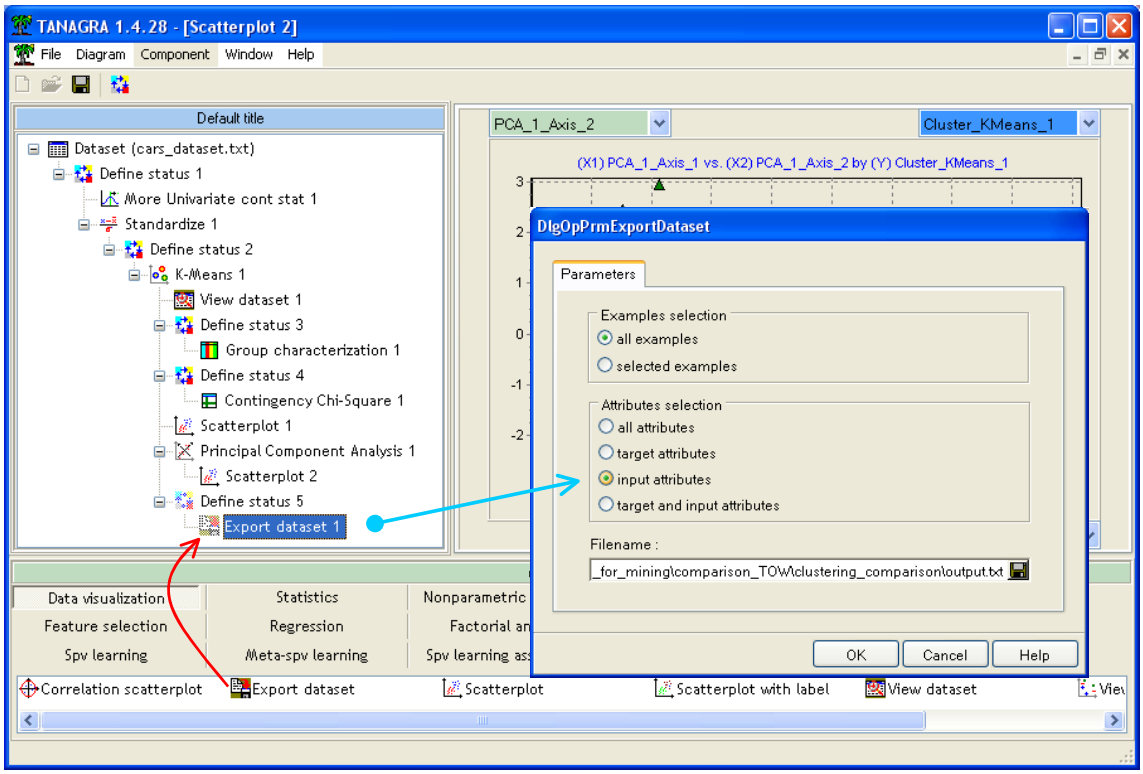
Last step of our analysis, we want to export the dataset with the additional column which indicates the cluster membership of each individual. TANAGRA can create a data file in the text file format with tab separator. We can handle it with the majority of tools (spreadsheet, data mining tools, etc.)[5].

We must before specify the columns to export using the DEFINE STATUS component. We set as INPUT the original variables (MPG…ORIGIN) and the computed column (CLUSTER_K_MEANS_1).

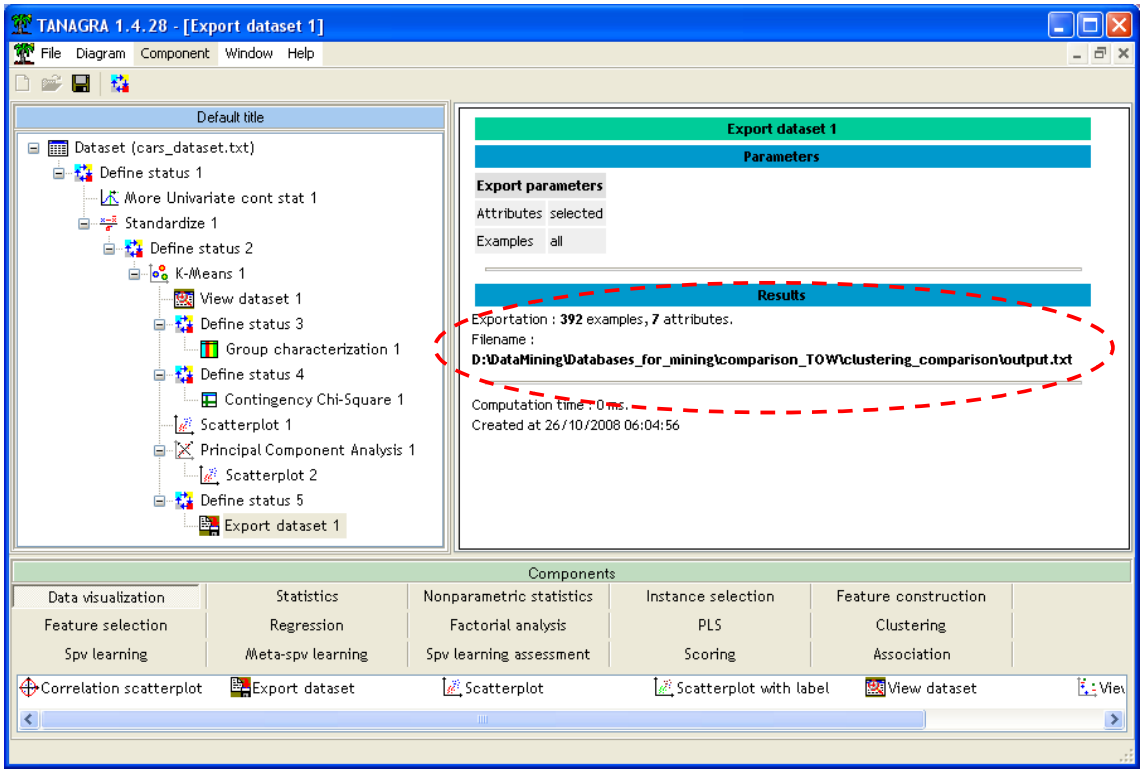


[5] TANAGRA can export also in the XLS (EXCEL) and ARFF (WEKA) file format.

We add the EXPORT DATASET component (DATA VISUALIZATION tab) into the diagram. In the settings dialog box (PARAMETERS menu), we specify that only the INPUT attributes must be exported. We can also define the directory and the file name. Then we validate and click on the VIEW menu.



A new data file (OUTPUT.TXT) with 392 observations and 7 variables is created.



4 K-Means with R

In this section, we duplicate the steps above using the R software (http://www.r-project.org/)[6].

4.1 Data importation and descriptive statistics

We set the following instructions in order to import the dataset and compute the descriptive statistics.

```
#importation des données
setwd("D:/DataMining/Databases_for_mining/comparison_TOW/clustering_comparison")
voitures <- read.table(file="cars_dataset.txt",header=T,dec=".")
#description et statistiques descriptives
summary(voitures)
```

We obtain…

```
> summary(voitures)
      mpg         displacement     horsepower        weight      acceleration        origin
 Min.   : 9.00   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00   american:245
 1st Qu.:17.00   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st Qu.:14.00   european: 68
 Median :23.00   Median :151.0   Median : 93.5   Median :2804   Median :16.00   japanese: 79
 Mean   :23.49   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean   :15.68
 3rd Qu.:29.00   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.00
 Max.   :47.00   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :25.00
```

4.2 Standardizing the variables

To transform the variable, we create first a call back function "**centrage_reduction(.)**" which standardizes one variable. Then we call the "**apply(.)**" function. The new data frame is "**voitures.cr**".

```
#préparer la fonction de standardisation d'une colonne
centrage_reduction <- function(x){
  return((x-mean(x))/sqrt(var(x)))
}
#appliquer pour produire le tableau des données centrées et réduites
voitures.cr <- apply(voitures[,1:5],2,centrage_reduction)
#vérification
apply(voitures.cr,2,mean)
apply(voitures.cr,2,var)
```

The mean of the new variables is 0; their variance is 1.

```
> apply(voitures.cr,2,mean)
          mpg  displacement    horsepower        weight  acceleration
-2.400263e-16 -6.008637e-17 -1.767797e-16 -7.163479e-18  1.160824e-16
> apply(voitures.cr,2,var)
         mpg displacement   horsepower       weight acceleration
           1            1            1            1            1
```

4.3 K-Means with the standardized variables

We can now launch the K-Means algorithm on these new variables. We ask a partitioning into two groups. We limit the number of iterations to 40.

[6] Unfortunately, the comments into the source code are in French. I apologize. I hope the instructions remain understandable.

```
#K-Means en 2 groupes
nb.classes <- 2
voitures.kmeans <- kmeans(voitures.cr,centers=nb.classes,iter.max=40)
print(voitures.kmeans)
```

R supplies among others: the number of examples in each cluster; the conditional mean according to the active variables; the cluster membership of each instance.

Note: It seems that we get the same groups than Tanagra. We should compare the 2 partitions to be sure. In some situations, we obtain a different partition of one data mining tool to the other. Indeed, since the approach relies on a heuristic, the initialization of the algorithm can influence the final result.

```
> print(voitures.kmeans)
K-means clustering with 2 clusters of sizes 100, 292

Cluster means:
        mpg displacement horsepower     weight acceleration
1 -1.1208246    1.4517603  1.4595718  1.3553242   -1.0180683
2  0.3838440   -0.4971782 -0.4998534 -0.4641521    0.3486535

Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [47] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [93] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[139] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[185] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[231] 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2
[277] 1 1 2 1 1 2 2 1 2 2 2 2 1 1 2 1 1 2 2 2 2 2 2 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 1 1 2 1 1 2 1
[323] 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[369] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 146.9415 684.7405

Available components:
[1] "cluster"  "centers"  "withinss" "size"
```

4.4 Interpretation of clusters

For the interpretation of the groups, we compute the **conditional mean** for the continuous variables.

```
#récupération des groupes d'appartenance
groupe <- as.factor(voitures.kmeans$cluster)
#calculer les barycentres des classes
#dans l'espace des variables actives initiales (numéro 1 à 5)
centres <- NULL
for (k in 1:nb.classes){
  ligne <- colMeans(voitures[groupe==k,1:5,drop=FALSE])
  centres <- rbind(centres,ligne)
}
numero <- seq(from=1,to=nb.classes)
rownames(centres) <- paste("clus_",numero,sep="")
print(centres)
```

We obtain...

```
> print(centres)
           mpg displacement horsepower   weight acceleration
clus_1 14.7500     346.3300   160.65000 4128.800     12.87000
clus_2 26.4863     142.3853    85.22945 2583.332     16.64384
```

We can compare these results to those of Tanagra: CLUS_1 of R is identical to C_KMEANS_2 of Tanagra.

In order to create a **cross tabulation** between the clusters and the ORIGIN categorical illustrative variable:

```
#croisement des clusters avec la variable illustrative catégorielle
print(table(voitures$origin,groupe))
```

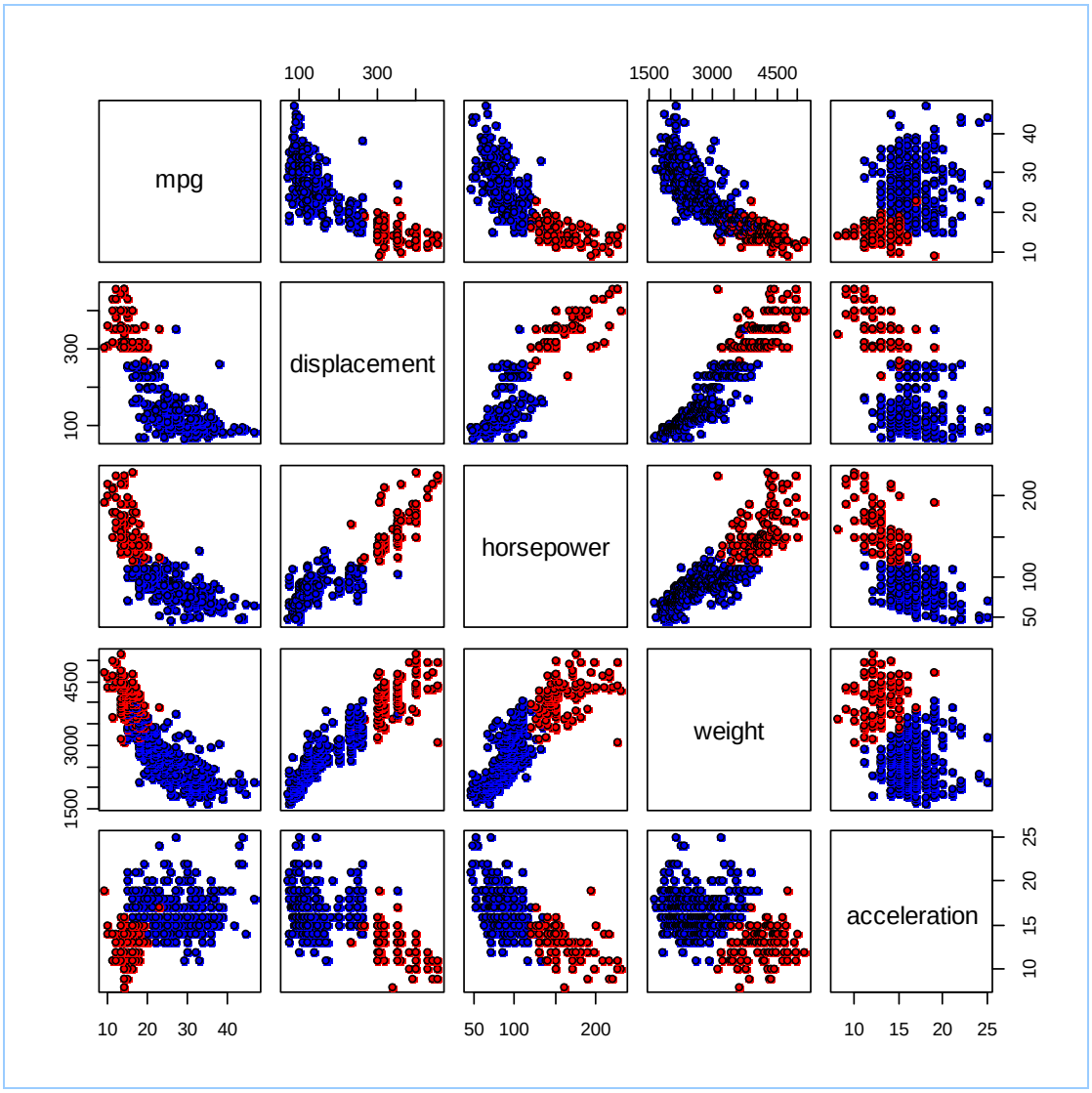
R supplies the following table.

```
> #croisement des clusters avec la variable illustrative catégorielle
> print(table(voitures$origin,groupe))
          groupe
             1   2
  american 100 145
  european   0  68
  japanese   0  79
```

We use the following instructions in order to create the **scatter plot** according each pair of variables.

```
#graphique des variables 2 à 2 avec groupe d'appaternance
pairs(voitures[,1:5],pch=21,bg=c("red","blue")[groupe])
```

We note that most of the variables are highly correlated. The groups are clearly discernable whatever the pairs of variables used.



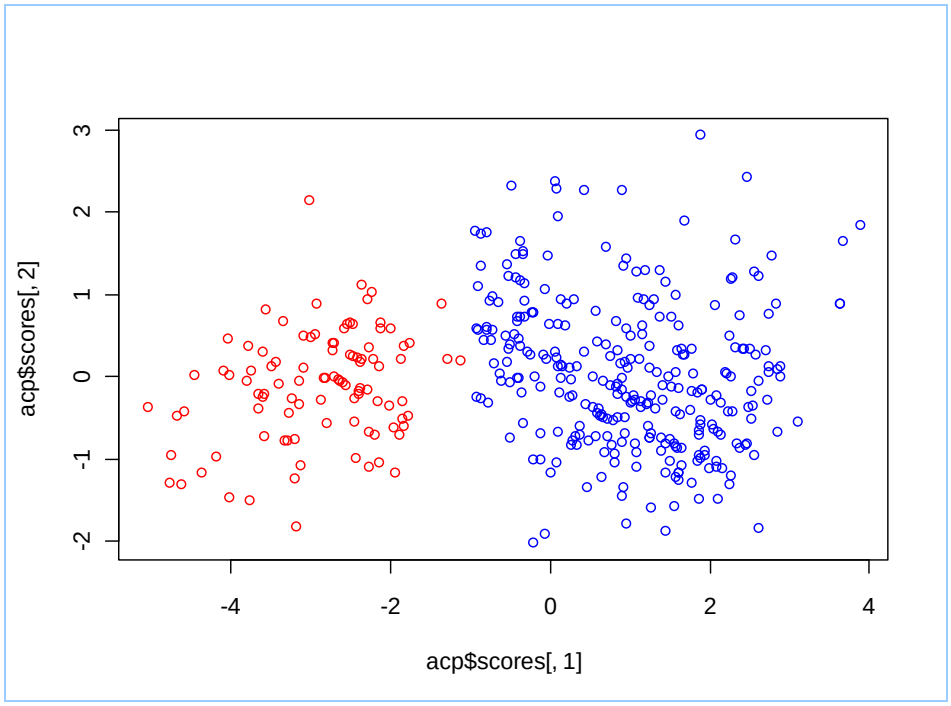
Last, we implement a PCA (**Principal Component Analysis**) for a multivariate characterization. We use the "**princomp(.)**" procedure.

```
#ACP sur les données centrées réduites
acp <- princomp(voitures.cr,cor=T,scores=T)
print(acp)
#pour obtenir les valeurs propres
print(acp$sdev^2)
#pour obtenir les corrélations sur le premier axe
print(acp$loadings[,1]*acp$sdev[1])
#graphique dans le premier plan factoriel, avec mise en évidence des groupes
plot(acp$scores[,1],acp$scores[,2],type="p",pch=21,col=c("red","blue")[groupe])
```

With some adjustments, we obtain the same results as Tanagra.

```
> #pour obtenir les valeurs propres
> print(acp$sdev^2)
    Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
3.91708377 0.72295773 0.22288784 0.08482332 0.05224734
> #pour obtenir les corrélations sur le premier axe
> print(acp$loadings[,1]*acp$sdev[1])
         mpg displacement   horsepower       weight acceleration
   0.8779948   -0.9588019   -0.9599118   -0.9343510    0.6576210
```

Then we create the scatter plot in the two first factors representation space.



4.5 Exporting the dataset including the CLUSTER column

Last, we export both the original dataset and the K-Means computed column.

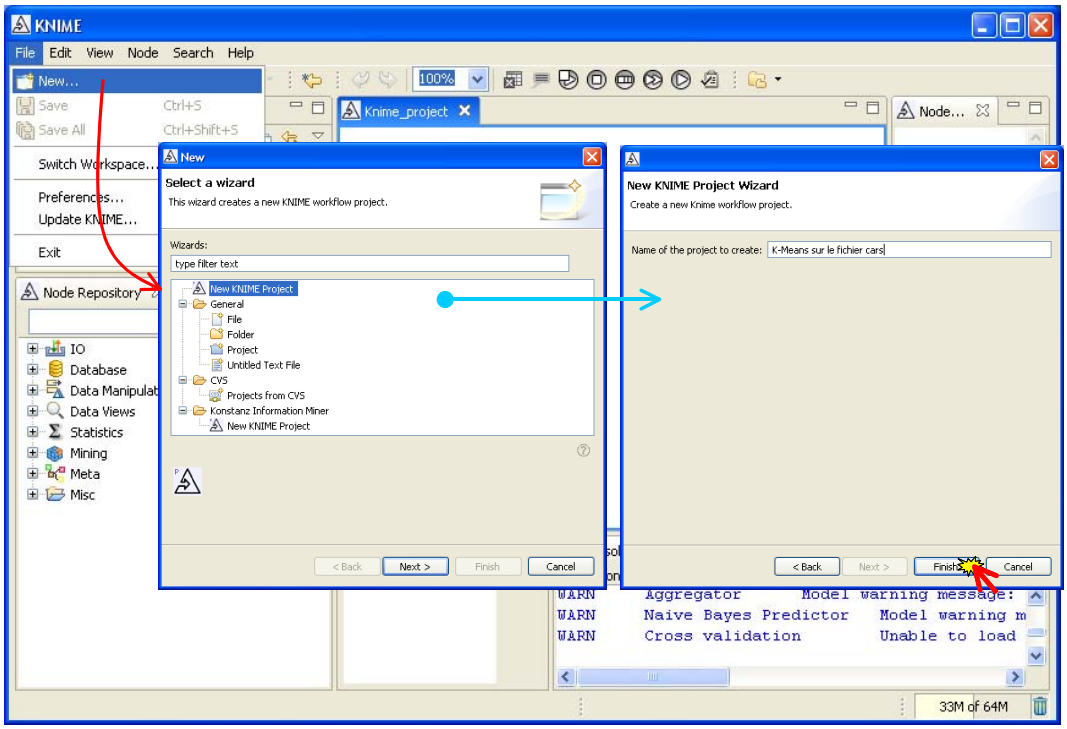
```
#exportation des données avec le cluster d'appartenance
voitures.export <- cbind(voitures,groupe)
write.table(voitures.export,file="export_r.txt",sep="\t",dec=".",row.names=F)
```

5 K-Means with KNIME

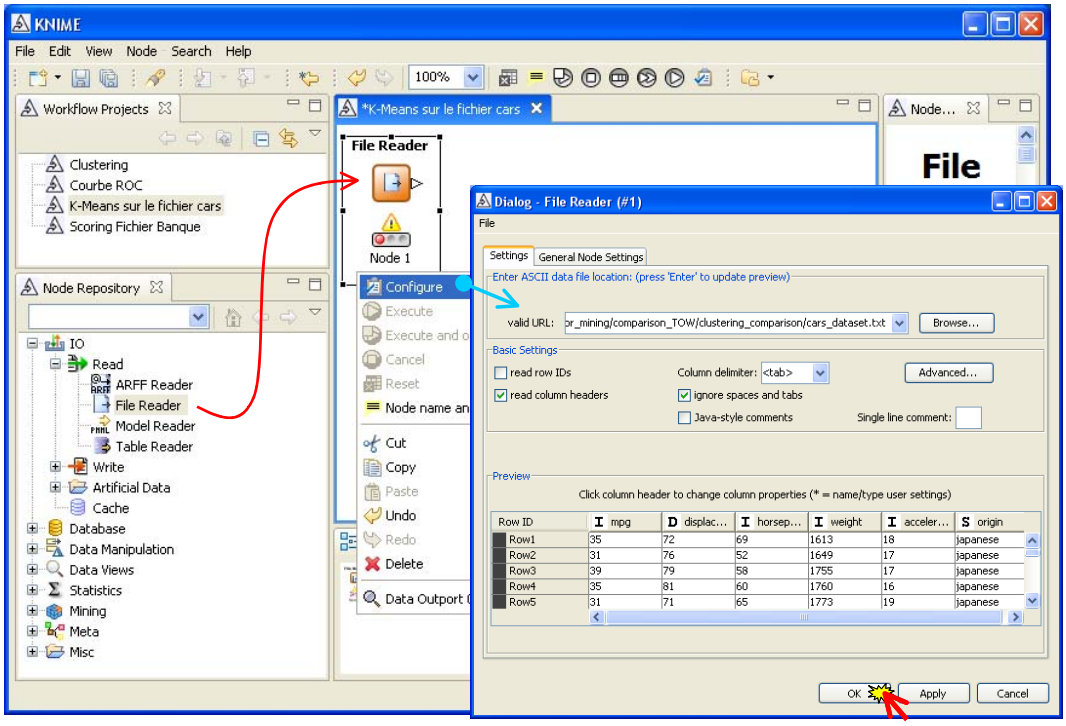
In this section, we duplicate the steps above using the Knime software (http://www.knime.org/).

5.1 Creating a workflow and importing the dataset

We create a new workflow by clicking on the FILE / NEW menu. We choose the "New Knime Project" item.

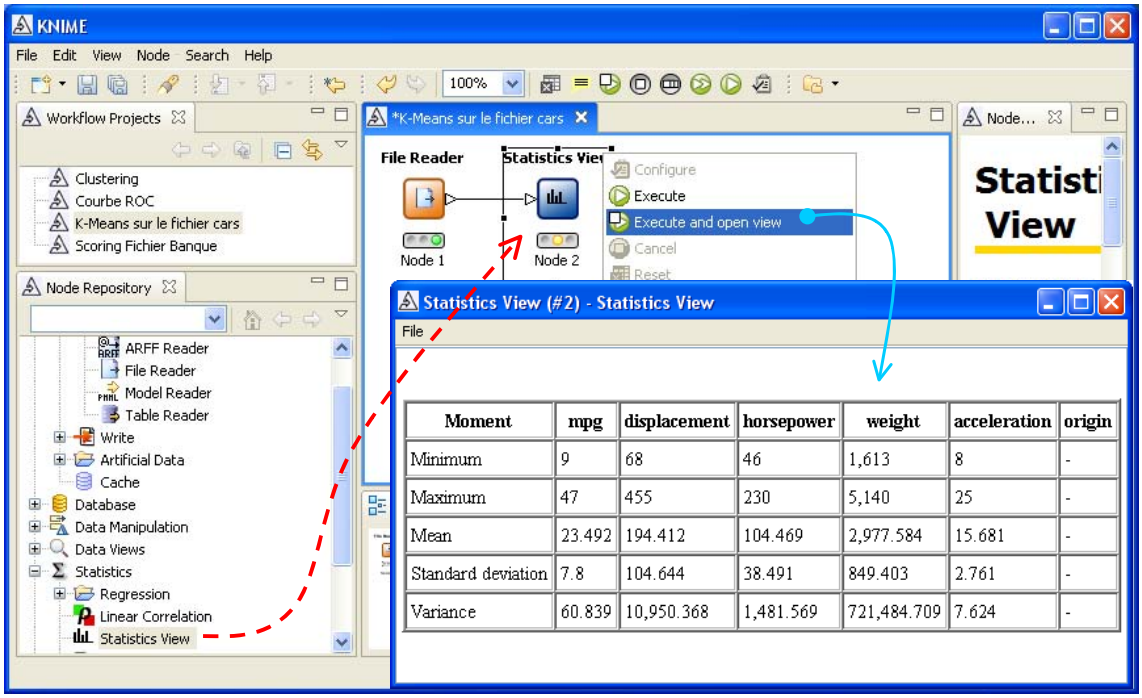


Then, we load the dataset using the FILE READER component.



5.2 Descriptive statistics

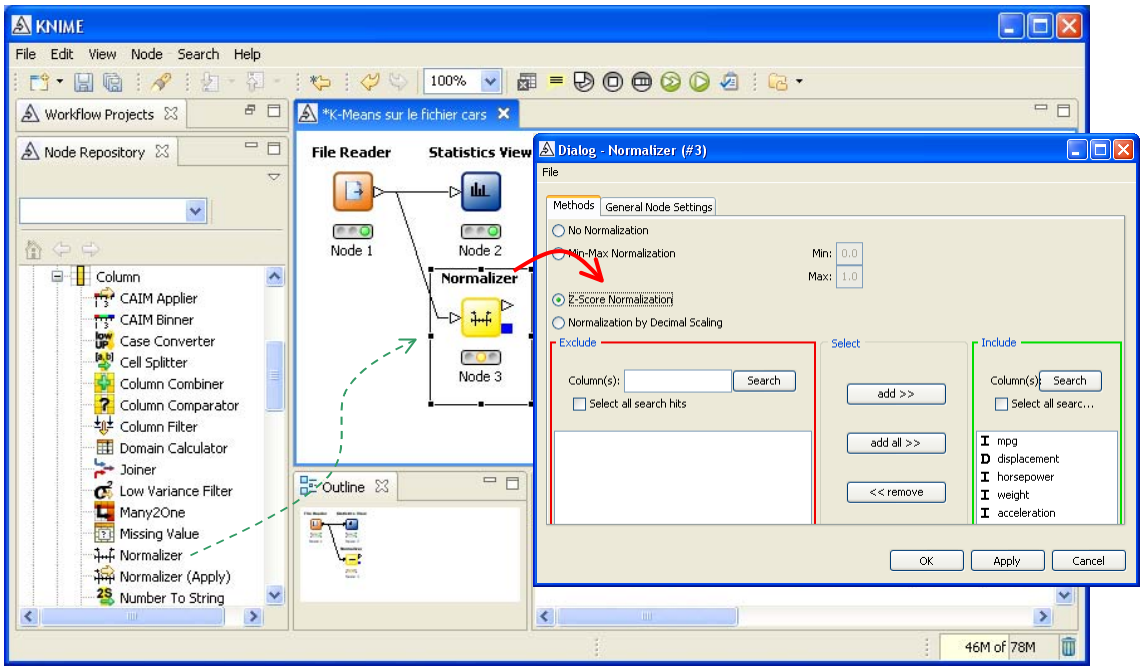
We use the STATISTICS VIEW component for the computation of the descriptive statistics indicators. We connect the FILE READER component to this last one. Then we click on the EXECUTE AND OPEN VIEW menu. The results are displayed in a new window.



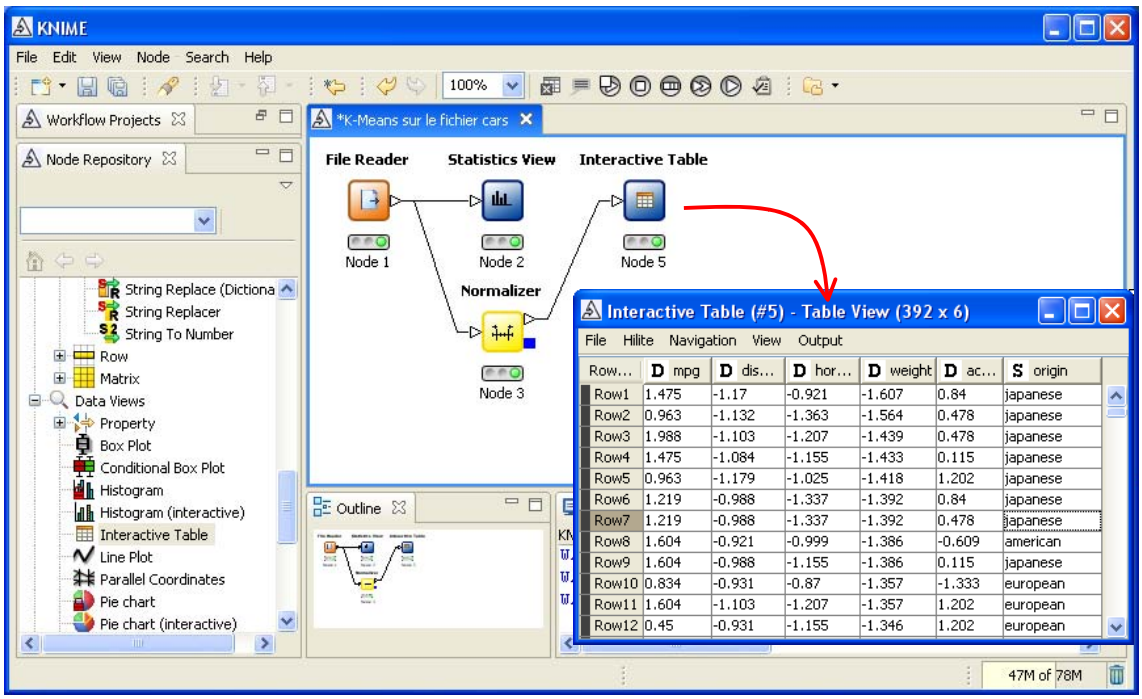
Moment	mpg	displacement	horsepower	weight	acceleration	origin
Minimum	9	68	46	1,613	8	-
Maximum	47	455	230	5,140	25	-
Mean	23.492	194.412	104.469	2,977.584	15.681	-
Standard deviation	7.8	104.644	38.491	849.403	2.761	-
Variance	60.839	10,950.368	1,481.569	721,484.709	7.624	-

5.3 Standardizing the variables

The NORMALIZER component allows to standardize the variables. We can implement different kind of normalization. We select the appropriate settings by clicking on the CONFIGURE menu.

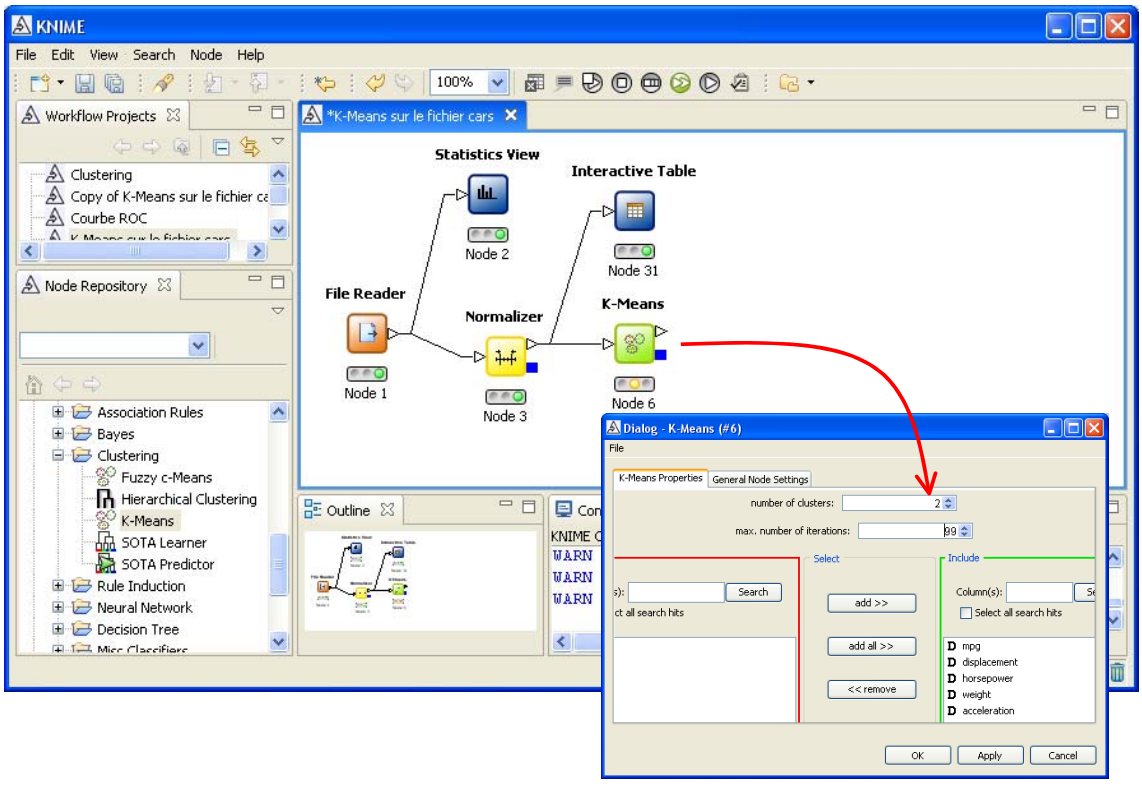


We can visualize the dataset with the INTERACTIVE TABLE component. Only the continuous variables are transformed of course.

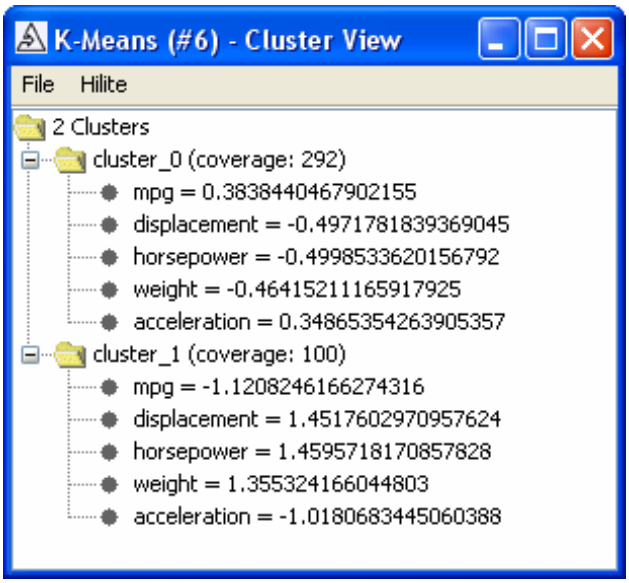


5.4 K-Means

We can launch the K-Means procedure. We add the K-Means component into the workflow. We click on the CONFIGURE menu in order to set the appropriate parameters.



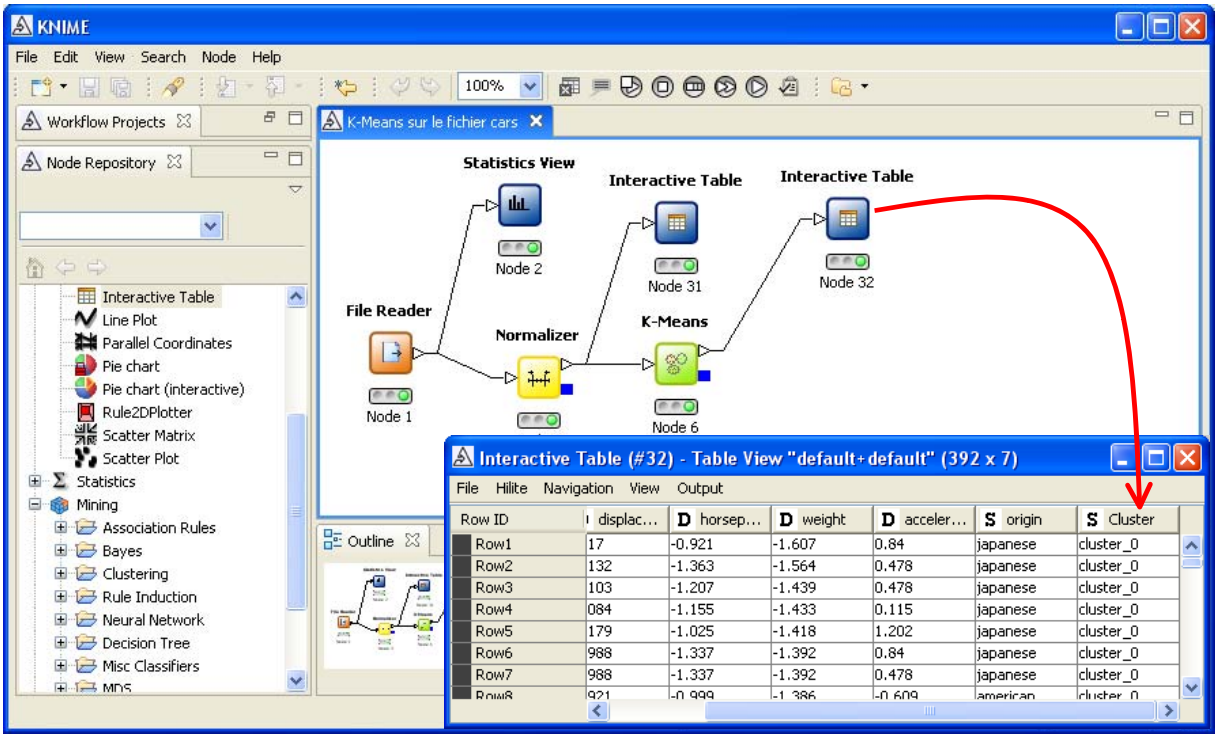
By clicking on the EXECUTE AND OPEN VIEW menu, we obtain the results in a new window. There are 2 groups with respectively 292 and 100 instances. The conditional means are also displayed, but computed on the standardized variables. This is not really useful for the interpretation.



5.5 Interpretation of groups

5.5.1 Group membership

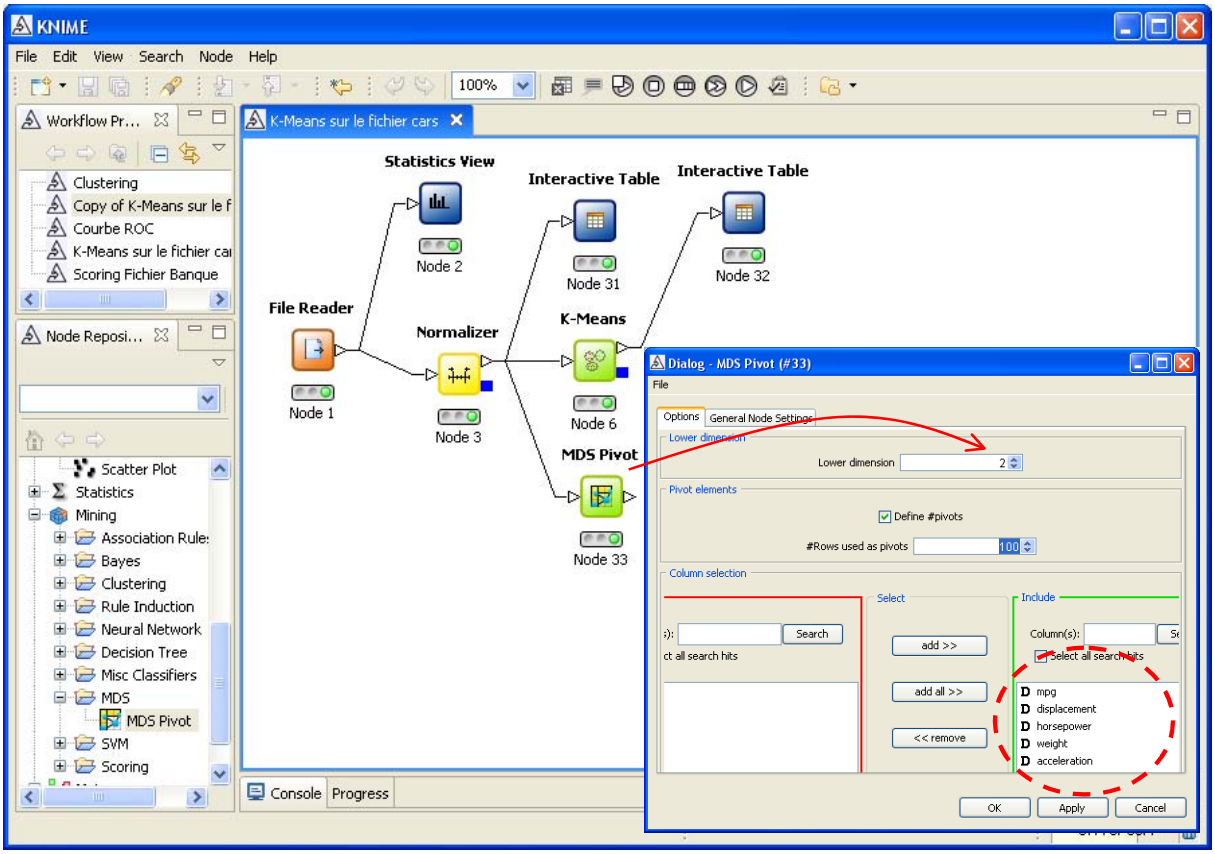
The INTERACTIVE TABLE component allows to visualize the cluster membership of each individual.



5.5.2 Descriptive statistics and graphical representation

Some preliminary manipulations are necessary before the calculations of the conditional descriptive statistics and the graphical representation.

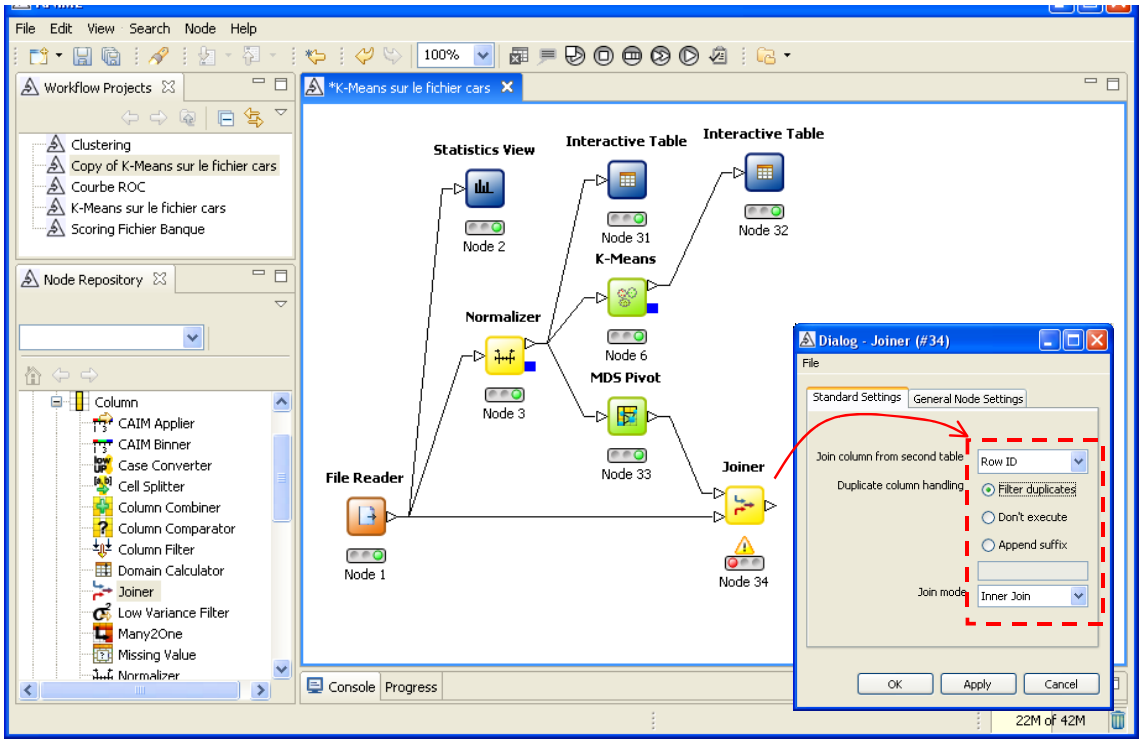
The PCA is not available under Knime. But it can perform a Multidimensional Scaling (MDS)[7]. We obtain the same factors when we launch this method on a similarity matrix (distance matrix) computed using a Euclidian distance[8]. We must thus compute this distance matrix using the PIVOT TABLE component. We set the appropriate parameters in order to compute the distance from the standardized variables. Only two latent variables are computed.



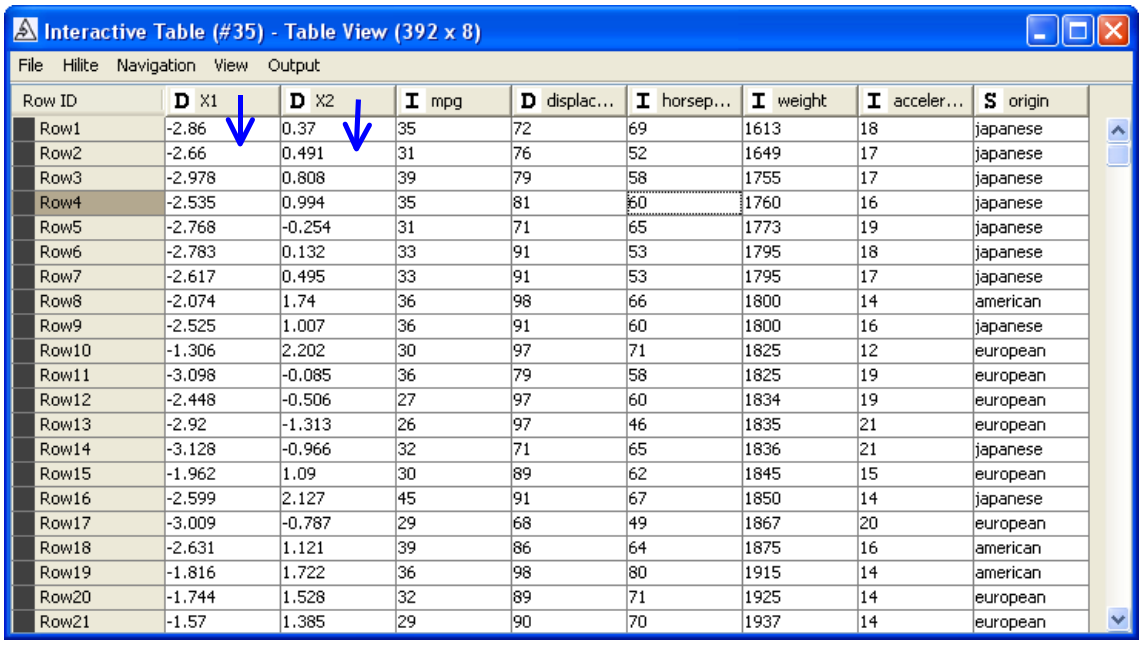
Two new columns are generated and available for the subsequent procedures. But, we must join them to the original dataset with the JOINER component. The connection settings are very important here. We must set them with caution.

[7] http://en.wikipedia.org/wiki/Multidimensional_scaling

[8] http://www.mathpsyc.uni-bonn.de/doc/delbeke/delbeke.htm



We insert the INTERACTIVE TABLE component in order to check the merging operation.

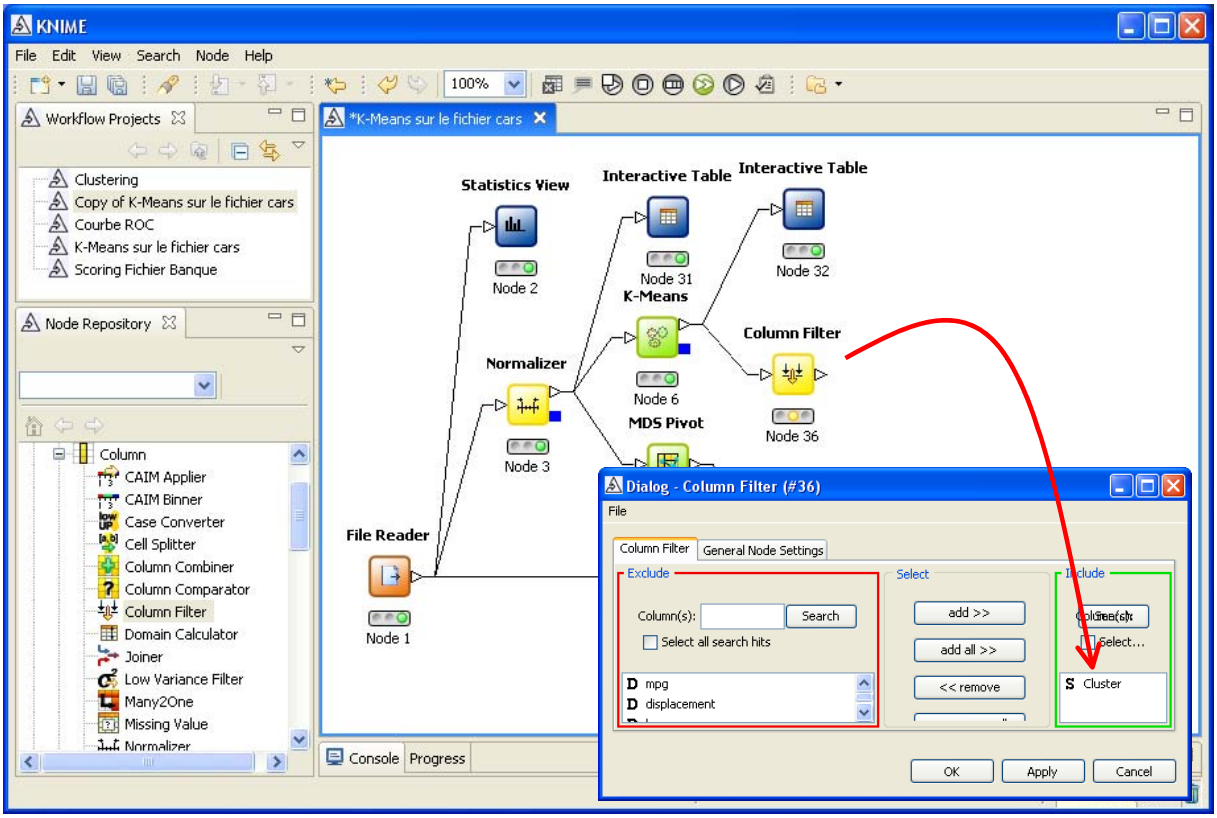


Row ID	D X1	D X2	I mpg	D displac...	I horsep...	I weight	I acceler...	S origin
Row1	-2.86	0.37	35	72	69	1613	18	japanese
Row2	-2.66	0.491	31	76	52	1649	17	japanese
Row3	-2.978	0.808	39	79	58	1755	17	japanese
Row4	-2.535	0.994	35	81	60	1760	16	japanese
Row5	-2.768	-0.254	31	71	65	1773	19	japanese
Row6	-2.783	0.132	33	91	53	1795	18	japanese
Row7	-2.617	0.495	33	91	53	1795	17	japanese
Row8	-2.074	1.74	36	98	66	1800	14	american
Row9	-2.525	1.007	36	91	60	1800	16	japanese
Row10	-1.306	2.202	30	97	71	1825	12	european
Row11	-3.098	-0.085	36	79	58	1825	19	european
Row12	-2.448	-0.506	27	97	60	1834	19	european
Row13	-2.92	-1.313	26	97	46	1835	21	european
Row14	-3.128	-0.966	32	71	65	1836	21	japanese
Row15	-1.962	1.09	30	89	62	1845	15	european
Row16	-2.599	2.127	45	91	67	1850	14	japanese
Row17	-3.009	-0.787	29	68	49	1867	20	european
Row18	-2.631	1.121	39	86	64	1875	16	american
Row19	-1.816	1.722	36	98	80	1915	14	american
Row20	-1.744	1.528	32	89	71	1925	14	european
Row21	-1.57	1.385	29	90	70	1937	14	european

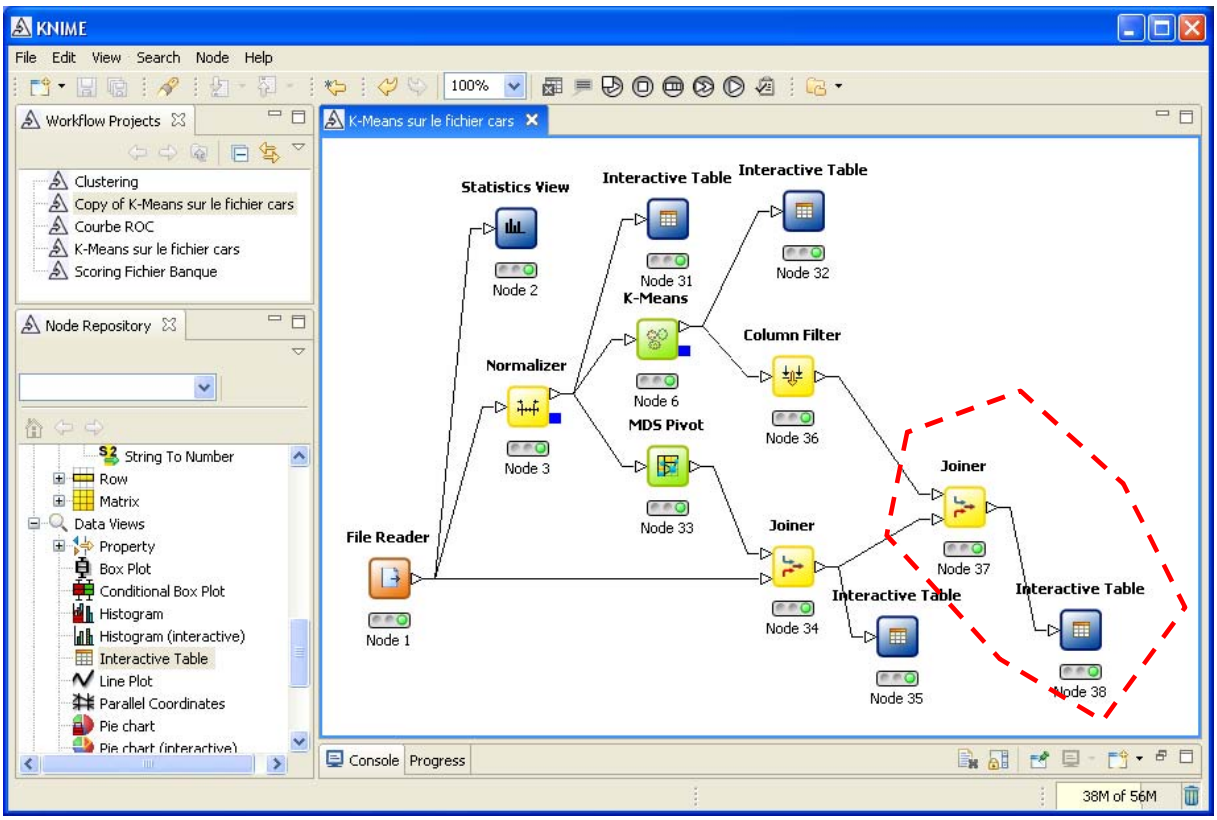
We have the original variables and the two additional columns supplied by the MDS component.

Now, we must merge this dataset to the additional column, the cluster membership, supplied by the K-Means component. We perform the operation into two steps:

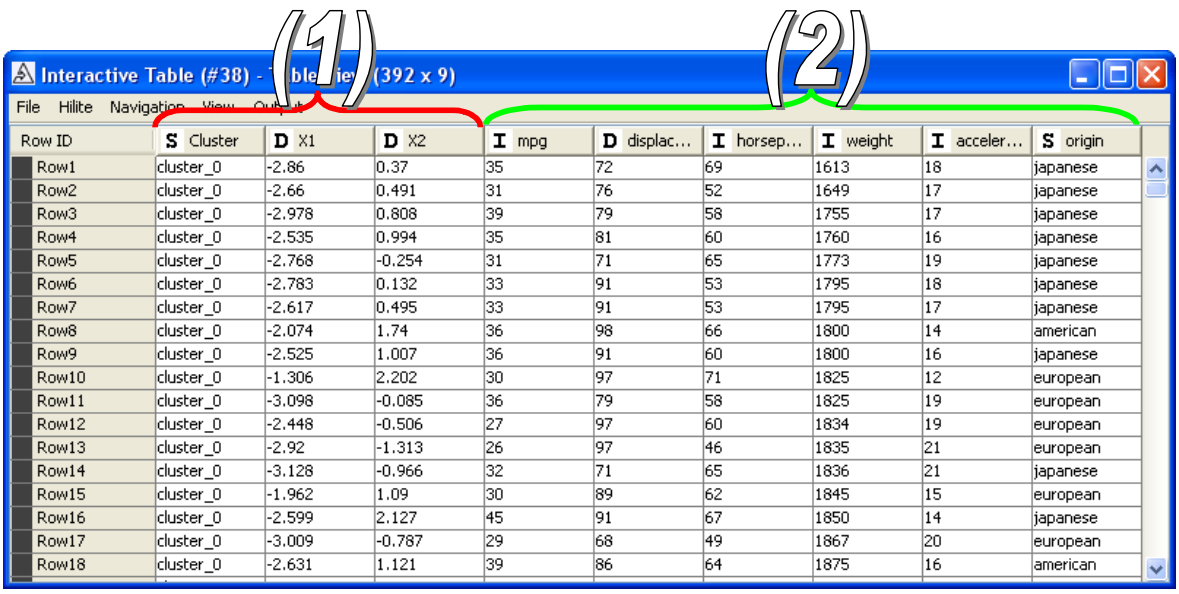
(1) With the COLUMN FILTER component, we select the cluster membership column from the K-Means component.



(2) With the JOINER component, we merge this column to the dataset.



We add the INTERACTIVE TABLE component in order to visualize the resulting dataset. The first group of variables is supplied by the various components; the second group comes from the original data file.

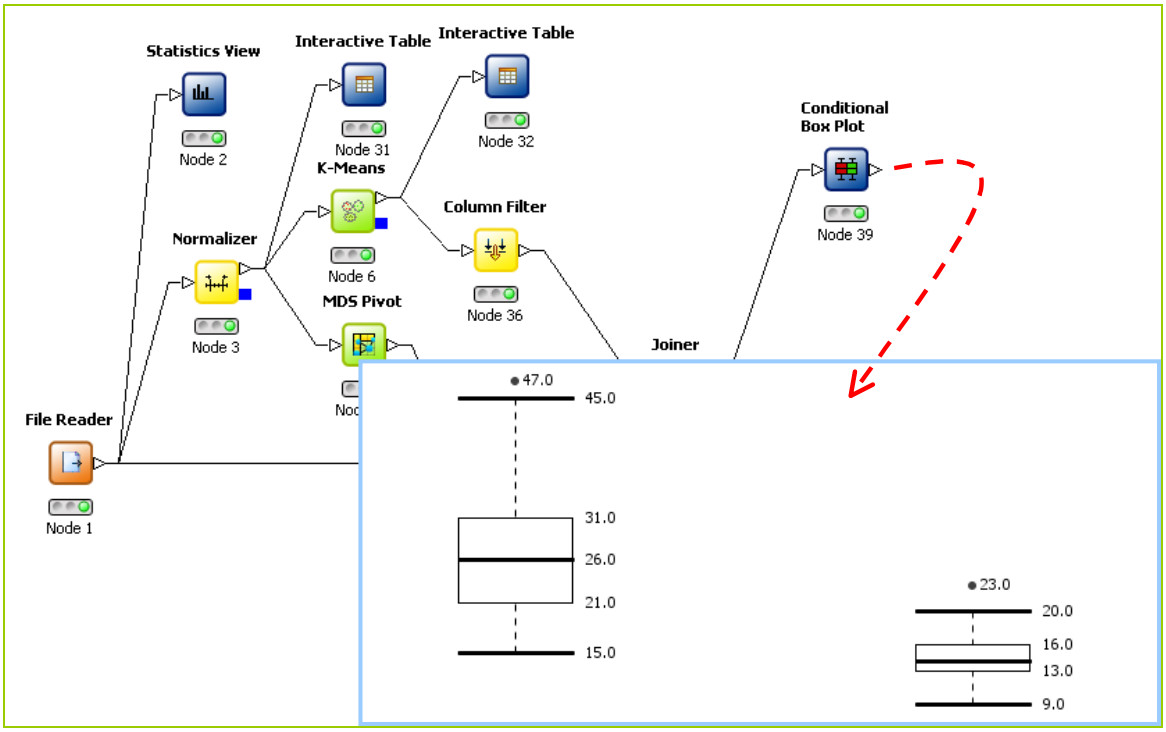


Row ID	S Cluster	D X1	D X2	I mpg	D displac...	I horsep...	I weight	I acceler...	S origin
Row1	cluster_0	-2.86	0.37	35	72	69	1613	18	japanese
Row2	cluster_0	-2.66	0.491	31	76	52	1649	17	japanese
Row3	cluster_0	-2.978	0.808	39	79	58	1755	17	japanese
Row4	cluster_0	-2.535	0.994	35	81	60	1760	16	japanese
Row5	cluster_0	-2.768	-0.254	31	71	65	1773	19	japanese
Row6	cluster_0	-2.783	0.132	33	91	53	1795	18	japanese
Row7	cluster_0	-2.617	0.495	33	91	53	1795	17	japanese
Row8	cluster_0	-2.074	1.74	36	98	66	1800	14	american
Row9	cluster_0	-2.525	1.007	36	91	60	1800	16	japanese
Row10	cluster_0	-1.306	2.202	30	97	71	1825	12	european
Row11	cluster_0	-3.098	-0.085	36	79	58	1825	19	european
Row12	cluster_0	-2.448	-0.506	27	97	60	1834	19	european
Row13	cluster_0	-2.92	-1.313	26	97	46	1835	21	european
Row14	cluster_0	-3.128	-0.966	32	71	65	1836	21	japanese
Row15	cluster_0	-1.962	1.09	30	89	62	1845	15	european
Row16	cluster_0	-2.599	2.127	45	91	67	1850	14	japanese
Row17	cluster_0	-3.009	-0.787	29	68	49	1867	20	european
Row18	cluster_0	-2.631	1.121	39	86	64	1875	16	american

5.5.2.1 Descriptive statistics

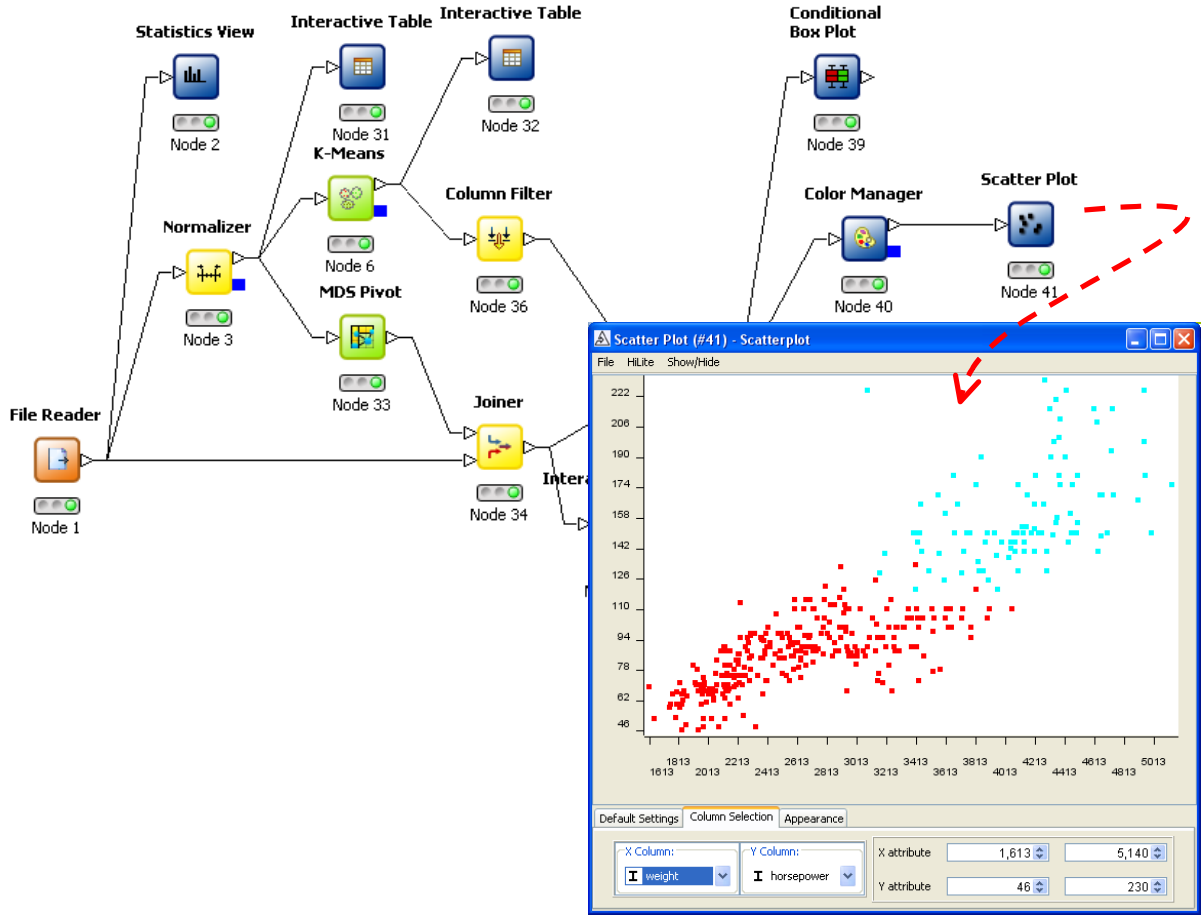
Knime offers a very interesting tool: the conditional boxplot. We can visualize more information about the characteristics of the distributions: central tendency measures, the shape of the distribution, the outliers, etc. The drawback is that we must insert one component for each variable.

We add the CONDITIONAL BOXPLOT component into the workflow. We set the appropriate settings (CONFIGURE menu). Then we click on the EXECUTE AND OPEN VIEW menu.



5.5.2.2 Scatter plot

We want to visualize the clusters in the representation space defined by the pair of variables. We must first specify the illustrative variable (CLUSTER) using the COLOR MANAGER component. We add after the SCATTERPLOT component in order to create the graphical representation.

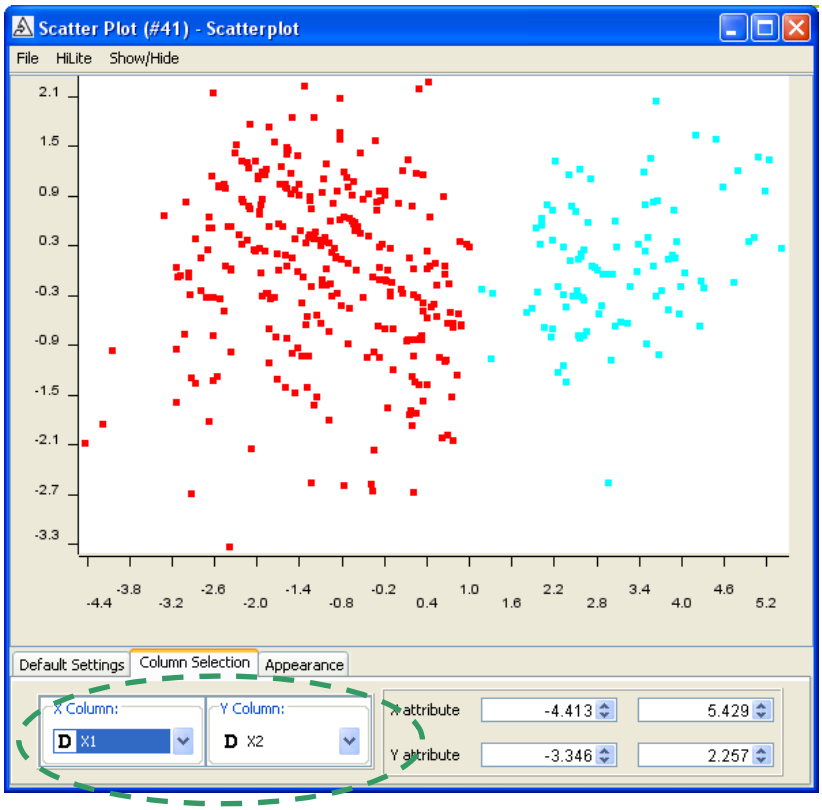


We clearly distinguish the two groups. We note that it is possible, as in Tanagra, to interactively modify the variables on the horizontal axis and the vertical axis.

5.5.2.3 Scatter plot in the latent variables representation space (MDS)

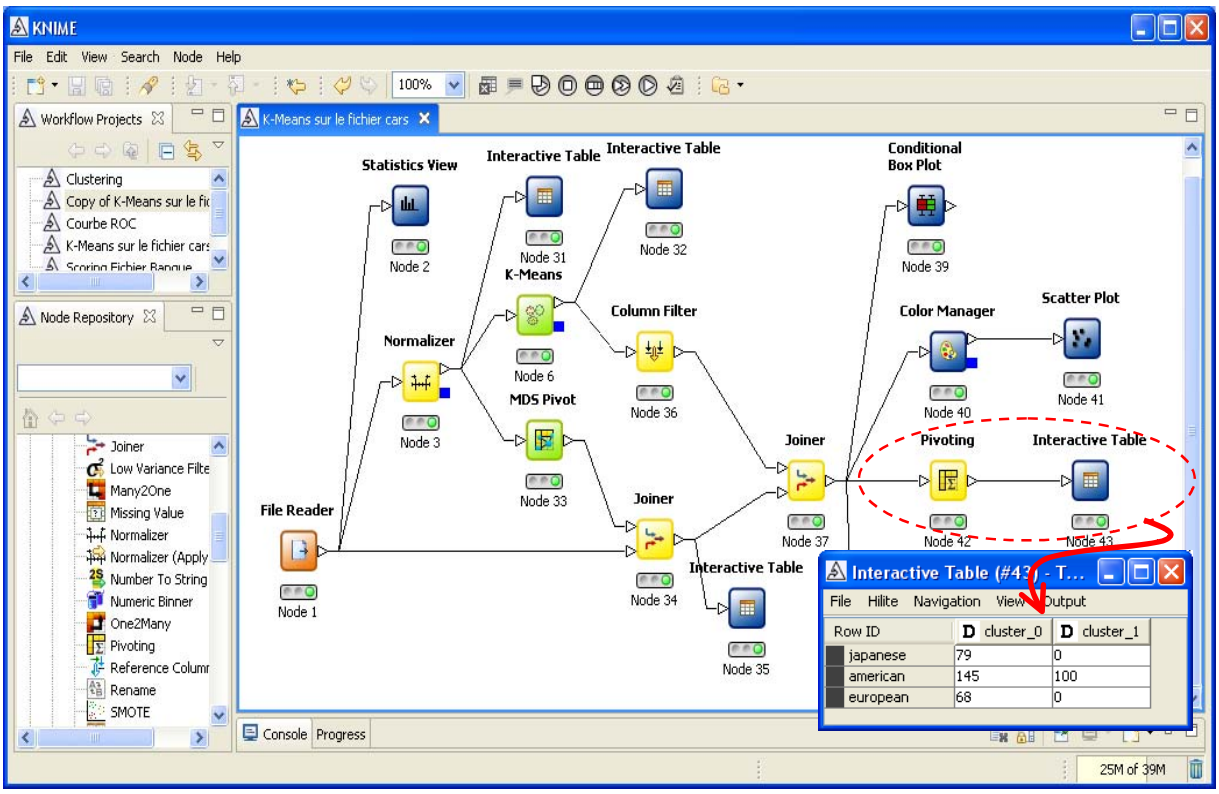
With the tool above (SCATTERPLOT), we can also create the scatter plot in the representation space defined by the MDS component. We set the appropriate columns into the X and Y axes.

The result is very similar to those obtained by the PCA component under R or Tanagra. The two groups are clearly discernable according the first factor.



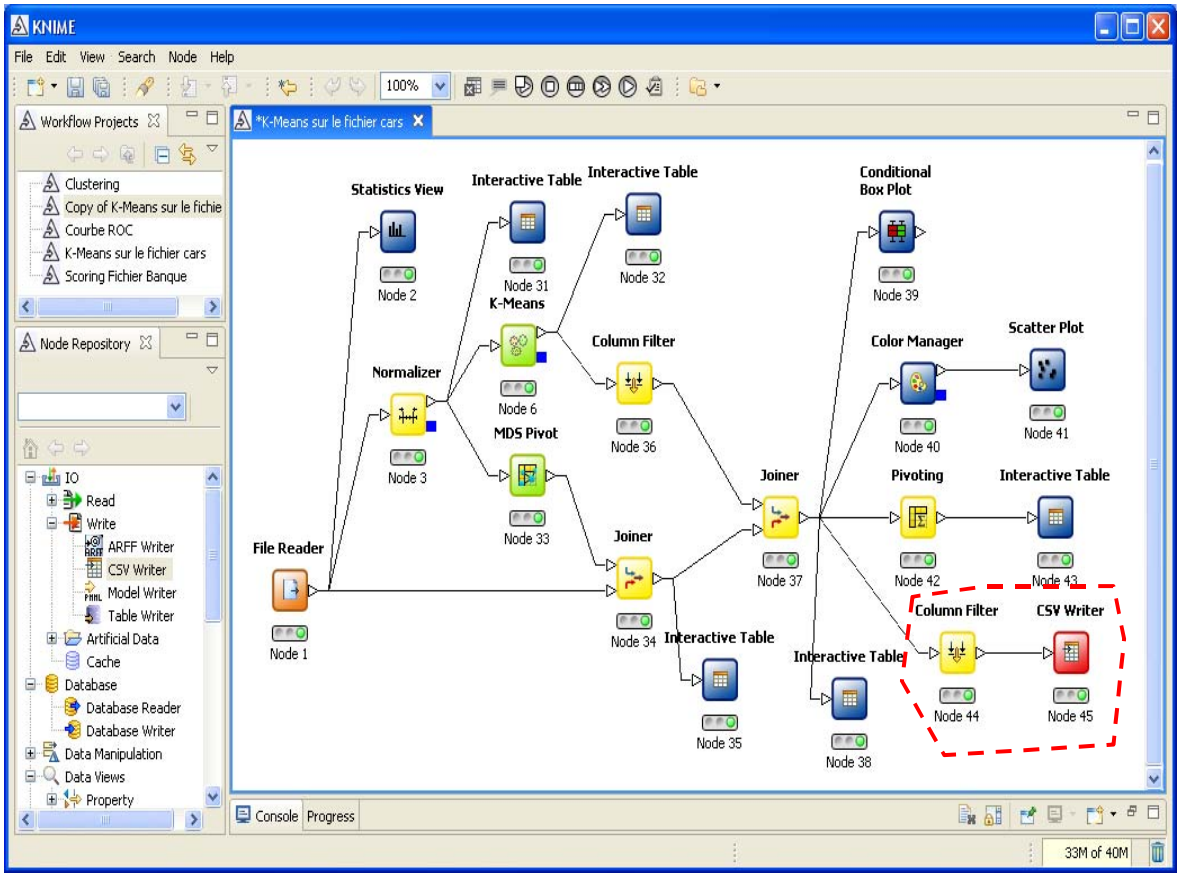
5.5.2.4 Cross tabulation with the ORIGIN variable

The PIVOTING component allows to create a cross tabulation between the CLUSTER and the ORIGIN columns. We use the INTERACTIVE TABLE component in order to visualize the table.



5.6 Exportation of the dataset

Last, we want to export the dataset with the cluster membership column. In the first time, we must filter the dataset in order to select the columns that we want to export. We use the COLUMN FILTER component. In the second time, we use the CSV WRITER component in order to create the data file. The resulting file is in the CSV format. We select ";" as the column separator.



The dataset can be imported easily into a spreadsheet or other data mining tools.

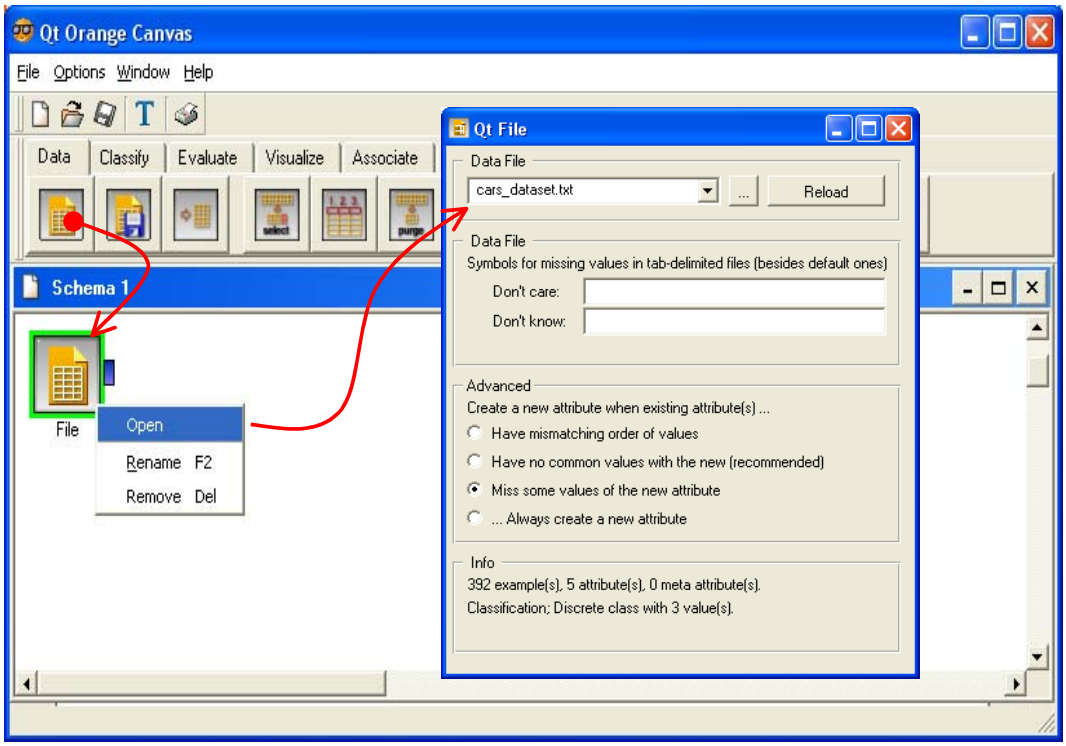
Knime is without any doubt a very performing tool. The analysis capabilities are very large. But the definition of the appropriate succession of components is sometimes difficult. We need a little training to get the correct sequence of operations.

6 K-Means with ORANGE

ORANGE is a nice Data Mining tool. It is above all very easy to use (http://www.ailab.si/orange/). A comprehensive description is available for each component. It describes the goal and the settings of the approach; sometimes a detailed example is supplied. We must think to press the F1 key when we need help.

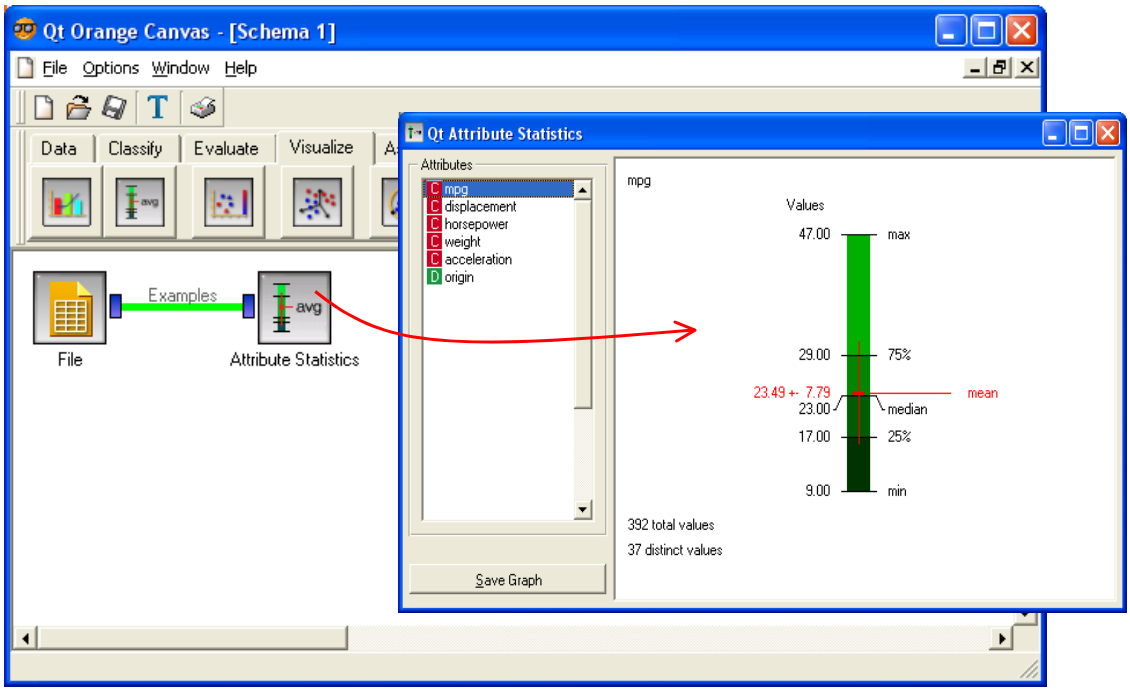
6.1 Creating a schema and importation of the dataset

An empty schema is available when we launch Orange. We add the FILE component (DATA tab). We set the appropriate settings by clicking on the OPEN menu. We select our data file (CARS_DATASET.TXT).



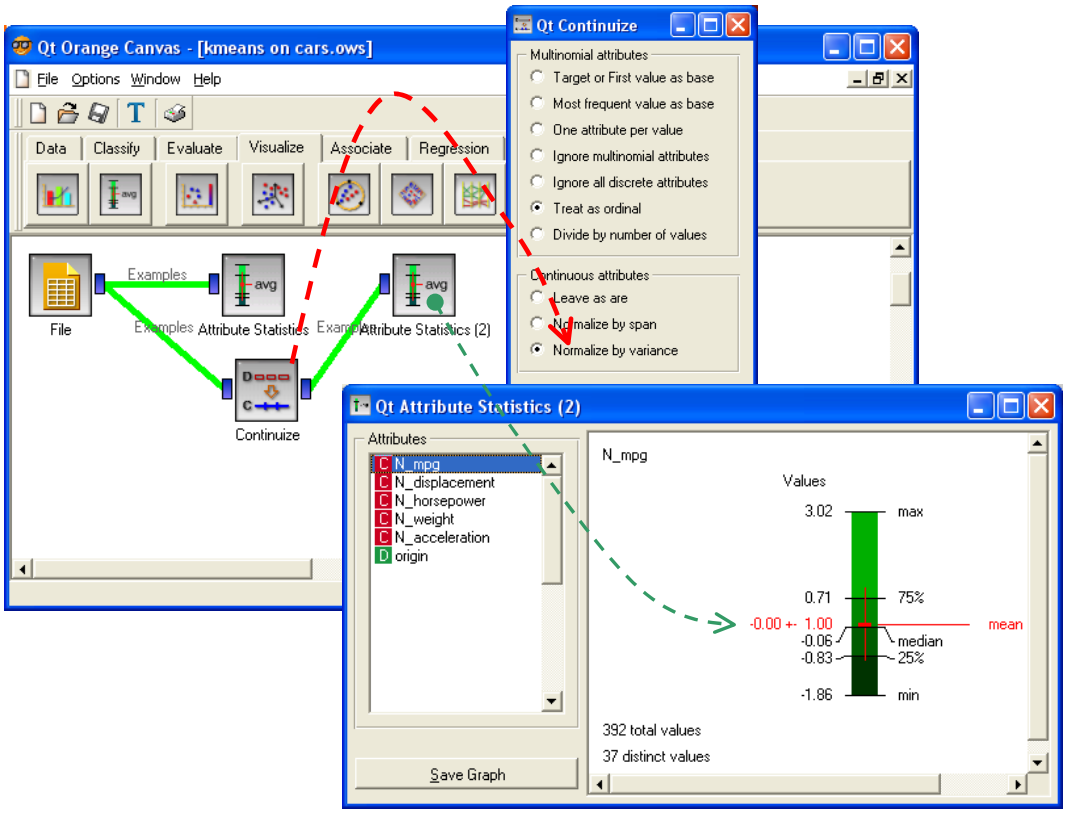
6.2 Descriptive statistics

Various descriptive statistics indicators are supplied by the ATTRIBUTES STATISTICS component. We can interactively select the variable in the left part of the visualization window. For categorical variable, we obtain the frequency table.



6.3 Standardizing the variables

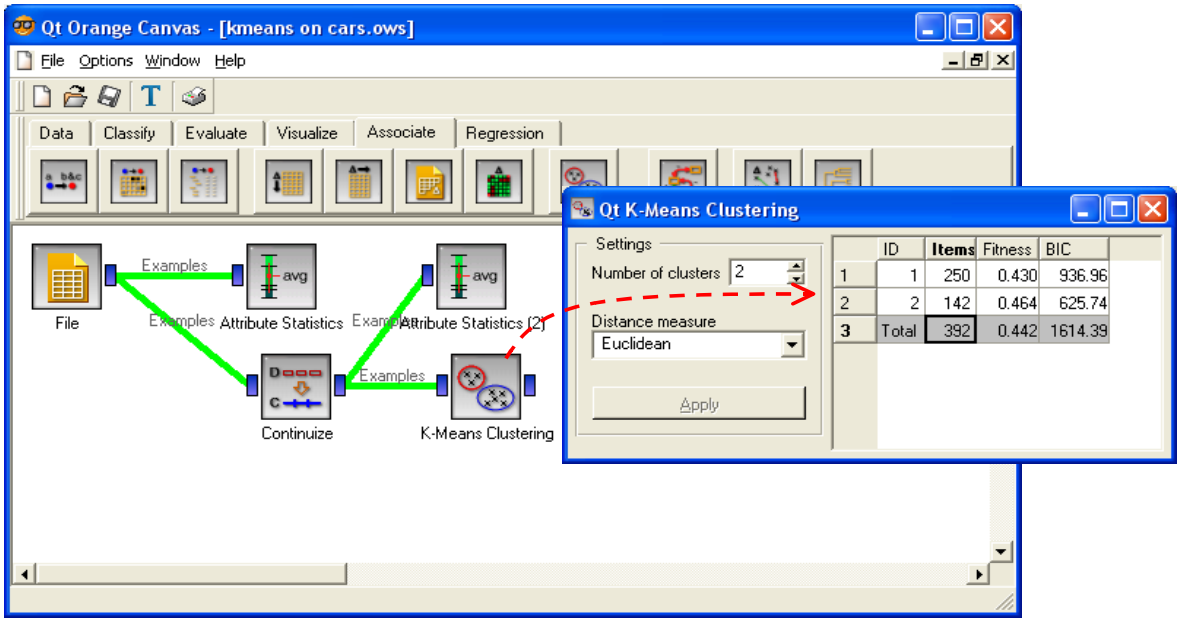
The CONTINUIZE component allows to standardize the variables. In the settings dialog box, we can set the right approach according to the type of the variable.



The ATTRIBUTE STATISTICS allows to check the transformation. All the continuous variables have now a mean = 0 and a standard deviation = 1.

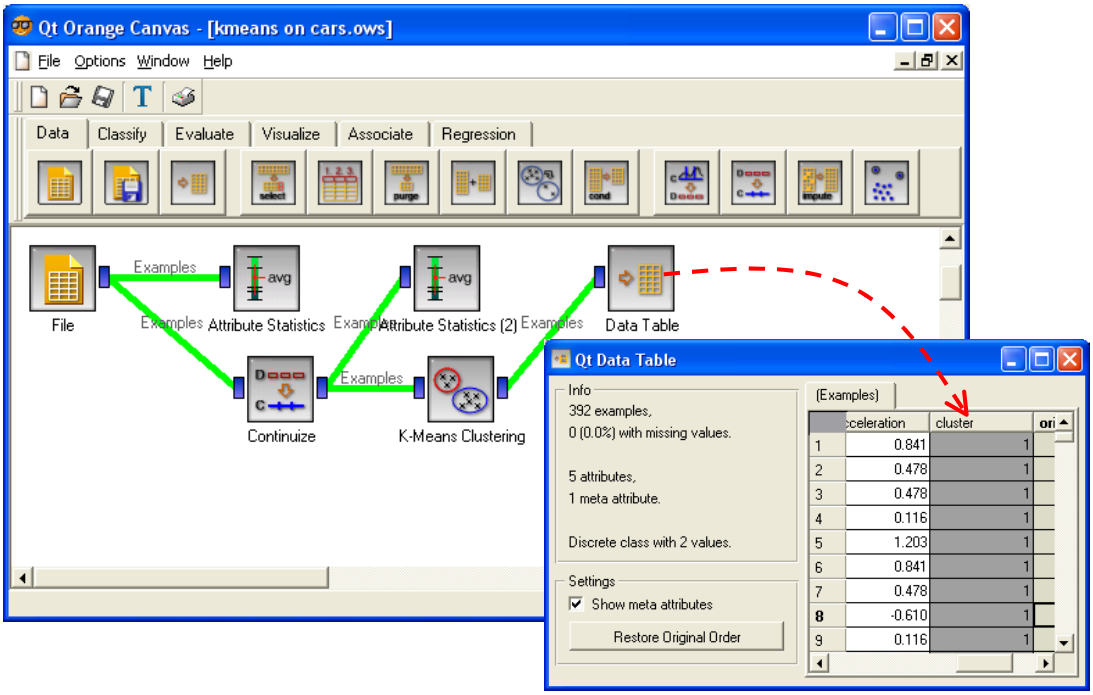
6.4 K-Means

The K-MEANS CLUSTERING component is available into the ASSOCIATE tab. We connect CONTINUIZE to this last one. Then we click on the OPEN menu: we set the appropriate parameters and we click on the APPLY button. Orange indicates the number of instance into each group (250 and 142). It supplies also some indicators of fitness for each group (see the help file for detailed description).



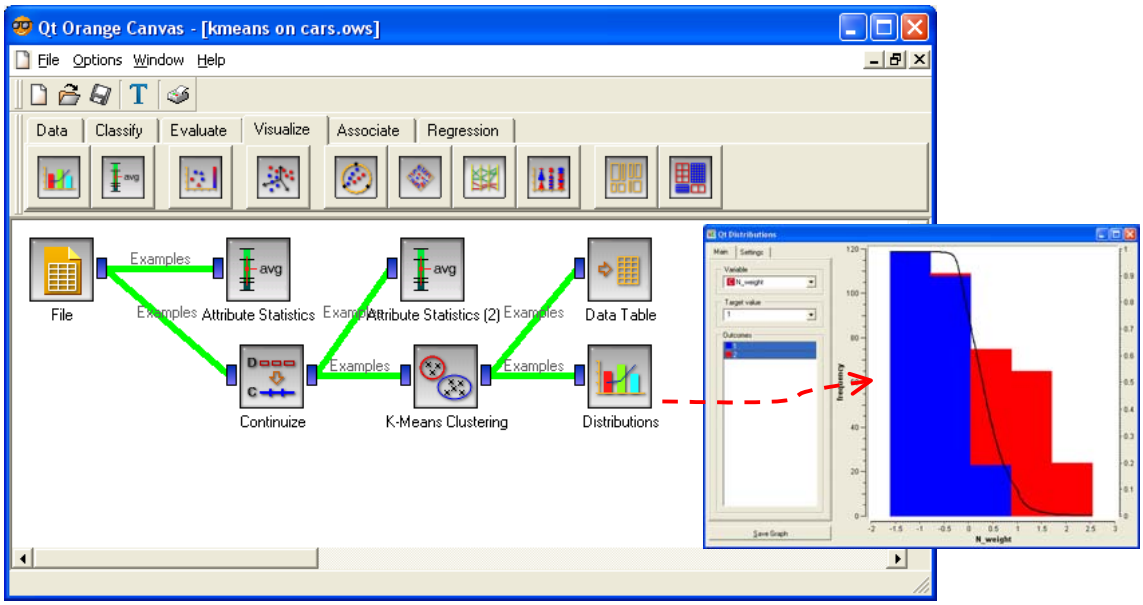
6.5 Interpretation of the partitioning

Cluster membership. Like the other tools, Orange creates a new column (CLUSTER) which describes the cluster membership of each individual.



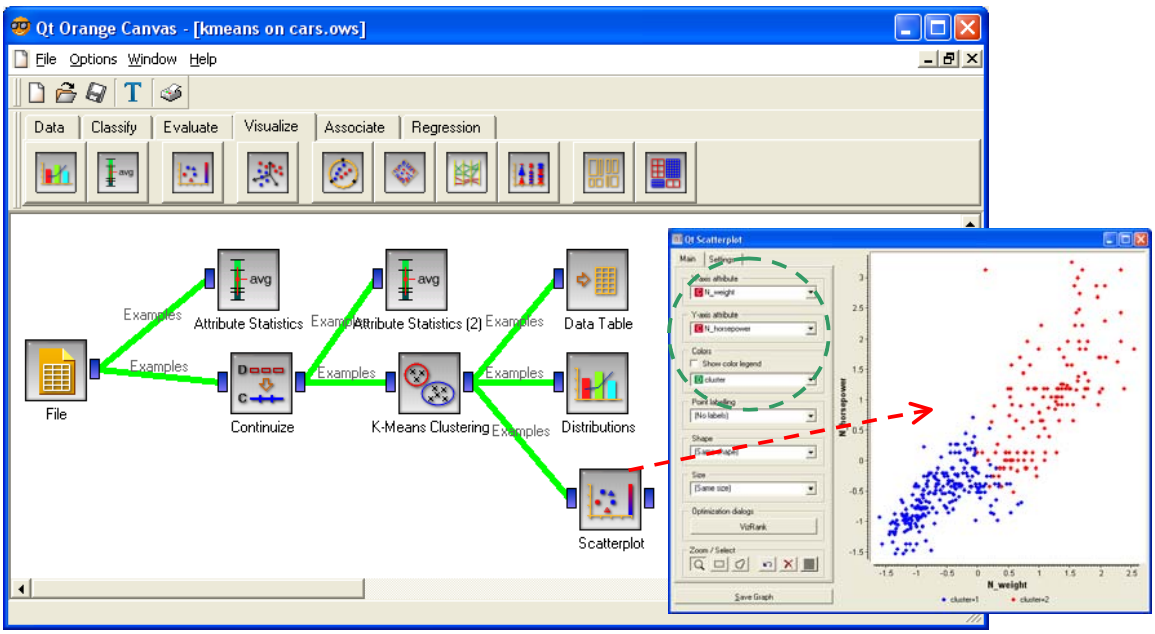
We can visualize this column with the DATA TABLE component.

Descriptive statistics. The DISTRIBUTIONS component allows to compute the histogram of variables according to the values of a categorical variable, the cluster membership in our case. Below, we have the histogram of WEIGHT variable.

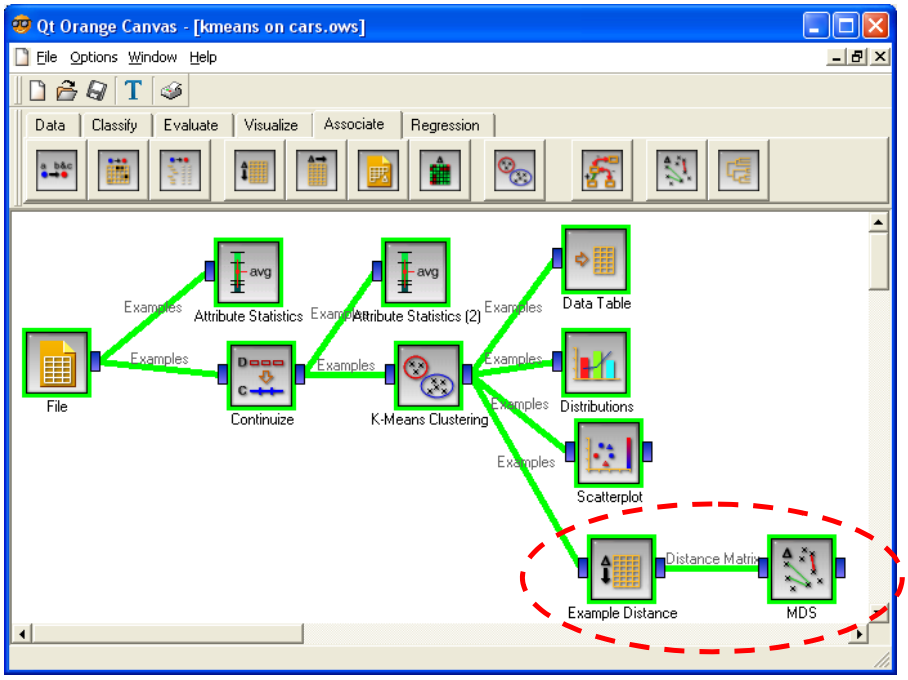


Note: The histograms are computed on the standardized variable in this part. A tool such as JOINER of KNIME is missing in order to recover the variables of the original data file in the subsequent part of the schema.

Scatter plot. The SCATTERPLOT component allows to visualize the instances according to simultaneously two variables.

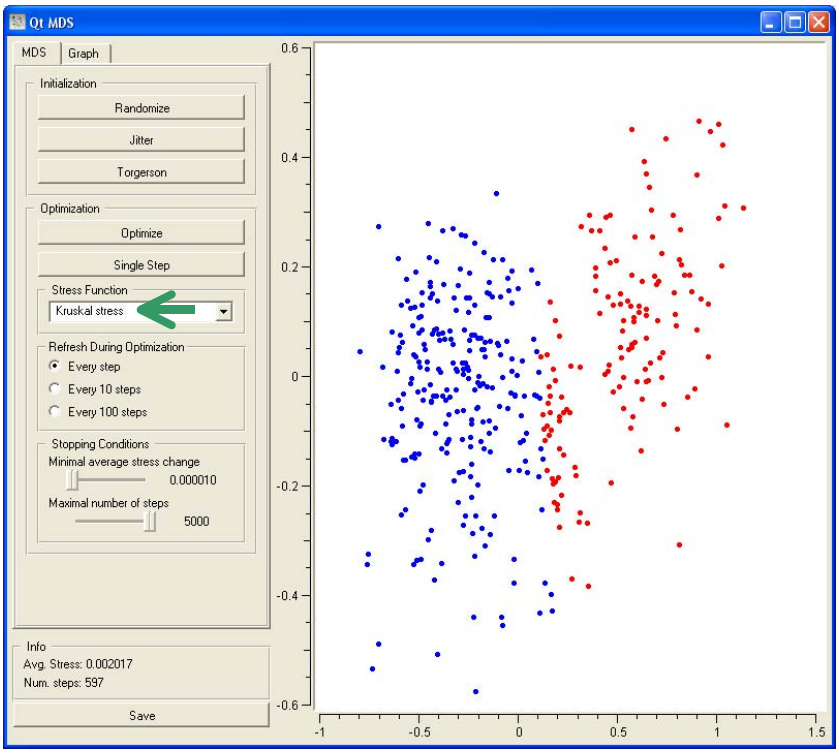


Projection in the representation space of MDS. The PCA is not available into Orange. Like Knime, we must compute first the distance matrix (the distance for each pair of instances). Then we perform a multidimensional scaling on this matrix. We define the following sequence of components in the schema. We click on the OPEN menu of the MDS component.



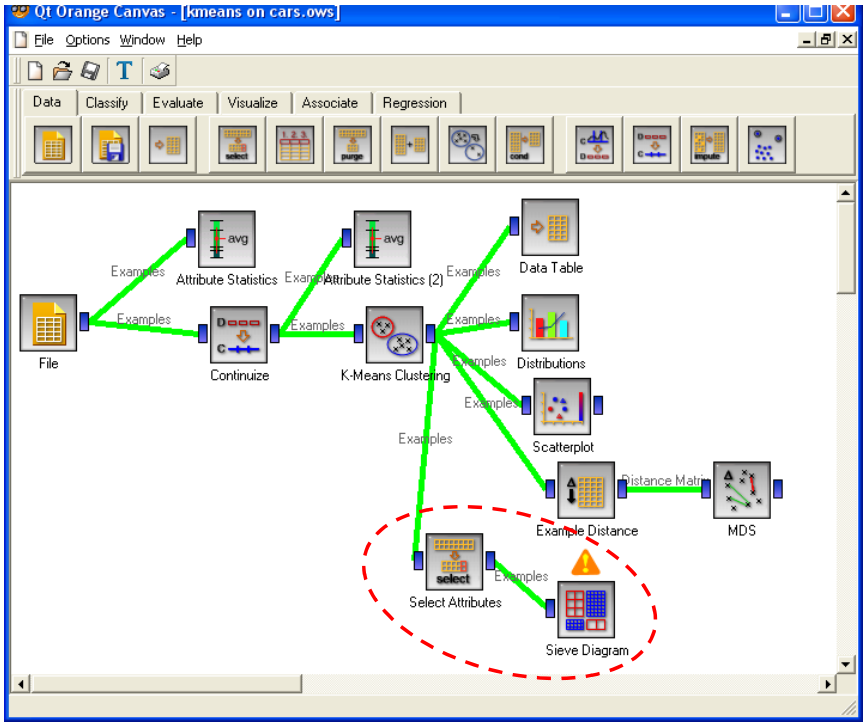
The tool is also interactive. We can define on the fly the illustrative variable which colorizes the points (GRAPH tab). We select the CLUSTER column here, but we can use any categorical variable. In the MDS tab, we select the STRESS function and we click on the OPTIMIZE button.

As we say above, the results are very similar to those of PCA. It is not really surprising.

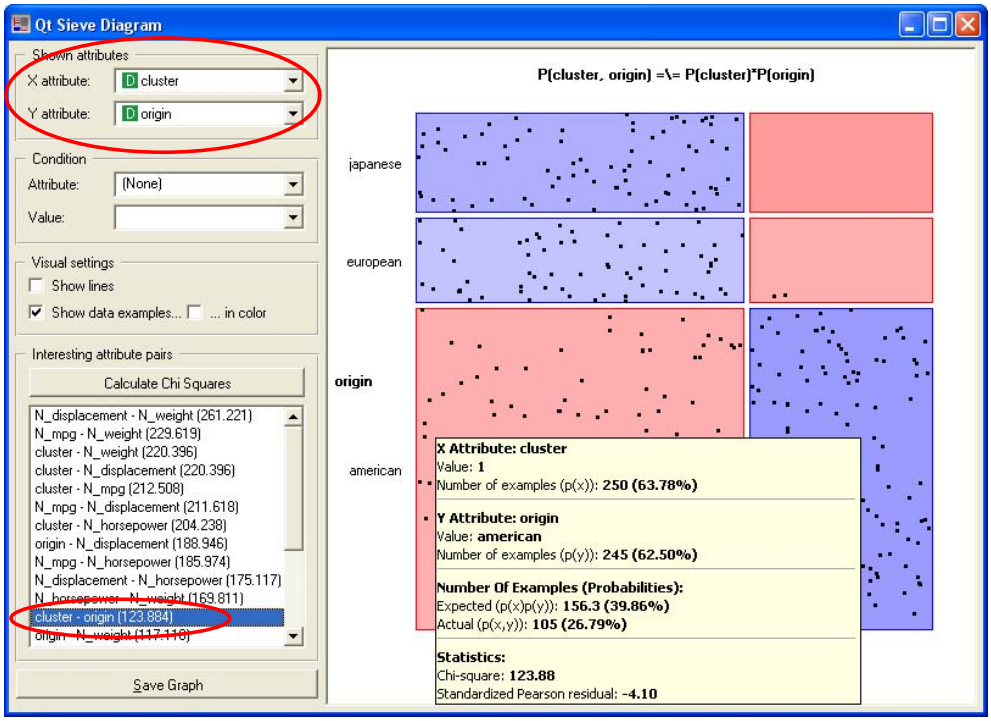


6.5.1 Cross-tabulation

The SIEVE DIAGRAM component allows to create a cross-tabulation between CLUSTER and ORIGIN. We must before use the SELECT ATTRIBUTES component in order to specify the used variables. We set all the variables as INPUT.



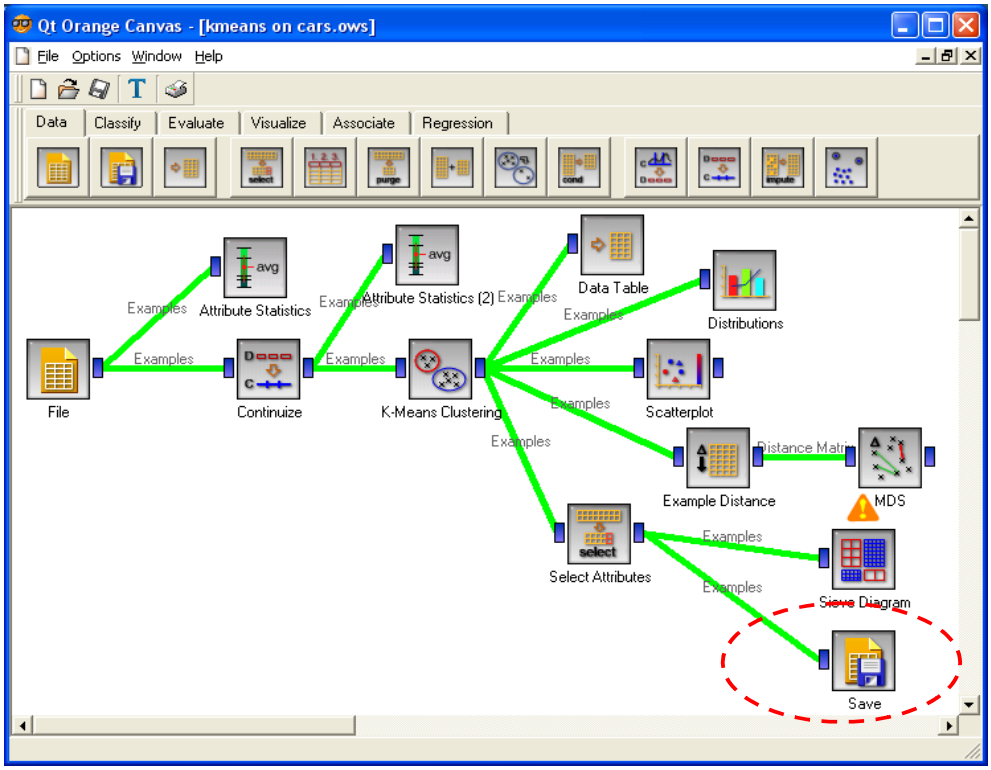
We click on the OPEN menu. The obtained visualization window seems mysterious. But if we consider with caution the results, we note that we can observe the desired information.



Into the selected field, we observe 105 instances. They correspond to the association between CLUSTER = 1 and ORIGIN = AMERICAN. Under the independence assumption, we should have 156.3. The CHI-Square statistic of the test for independence is 124.884.

6.6 Exporting the dataset including the CLUSTER column

Finally, we use the SAVE component in order to export the dataset. Orange exports the standardized variables. I did not know how to recover the original variables with the CLUSTER column.



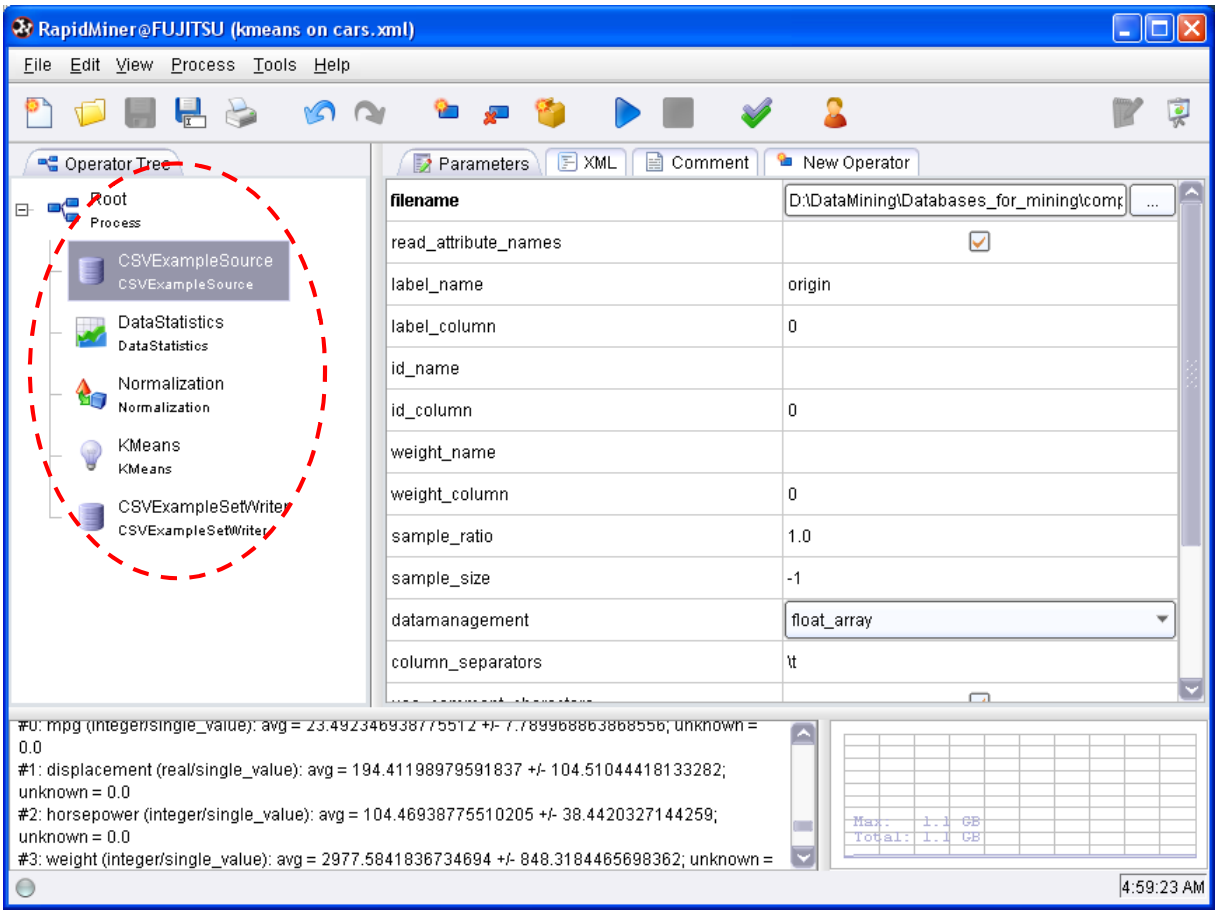
7 K-Means with RAPIDMINER

RAPIDMINER (http://rapid-i.com/content/blogcategory/38/69/) is the successor of YALE. Two versions are available, we use the free one i.e. the « Community Edition » version.

It is not possible to launch each component when it is inserted into the diagram. Each time you activate the PLAY button all the components of the diagram are executed. Fortunately, the computation is very fast. For this reason, unlike other tools in this tutorial, we adopt a different approach: we first define the whole diagram, and then we launch all the computations.

7.1 Specifying the diagram

Here is the whole diagram.

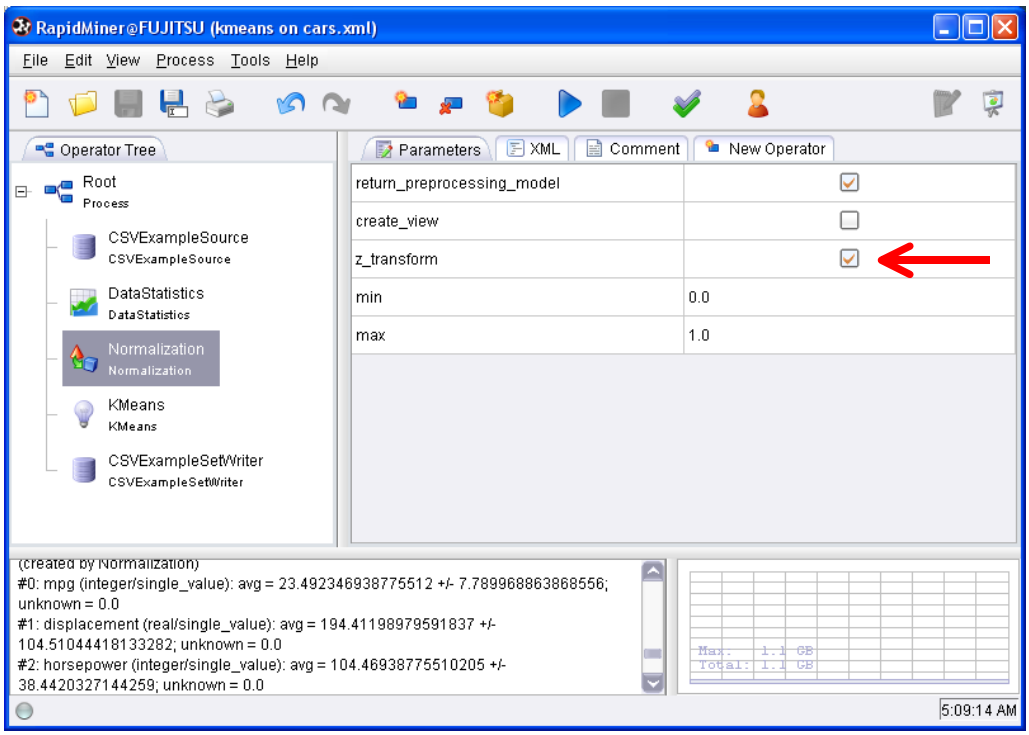


We observe the following tools.

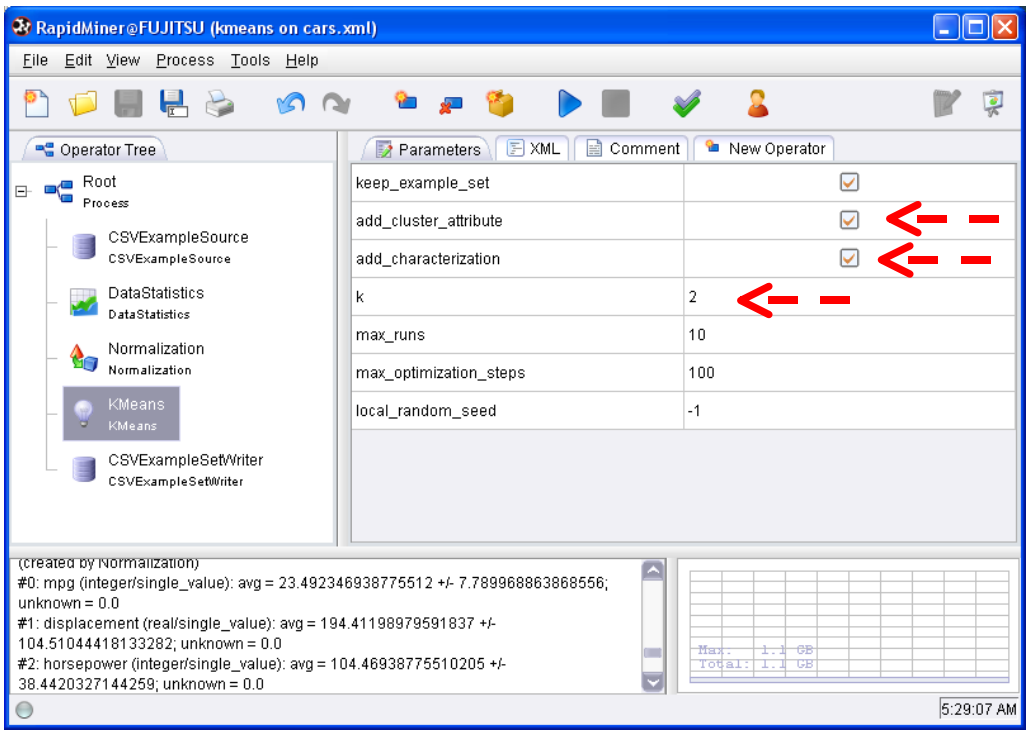
Accessing the data file. The CSVEXAMPLESOURCE component allows to access the dataset. The main parameters are: FILENAME specifies the file name; LABEL_NAME refers to the label of each instance, we use the ORIGIN variable in our tutorial, it is not really relevant but it allows to separate active and illustrative variables; COLUMN_SEPARATORS corresponds to the column separator, we set "\t" i.e. the tabulation character.

Descriptive statistics. DATASTATISTICS describes the dataset through descriptive statistics indicators.

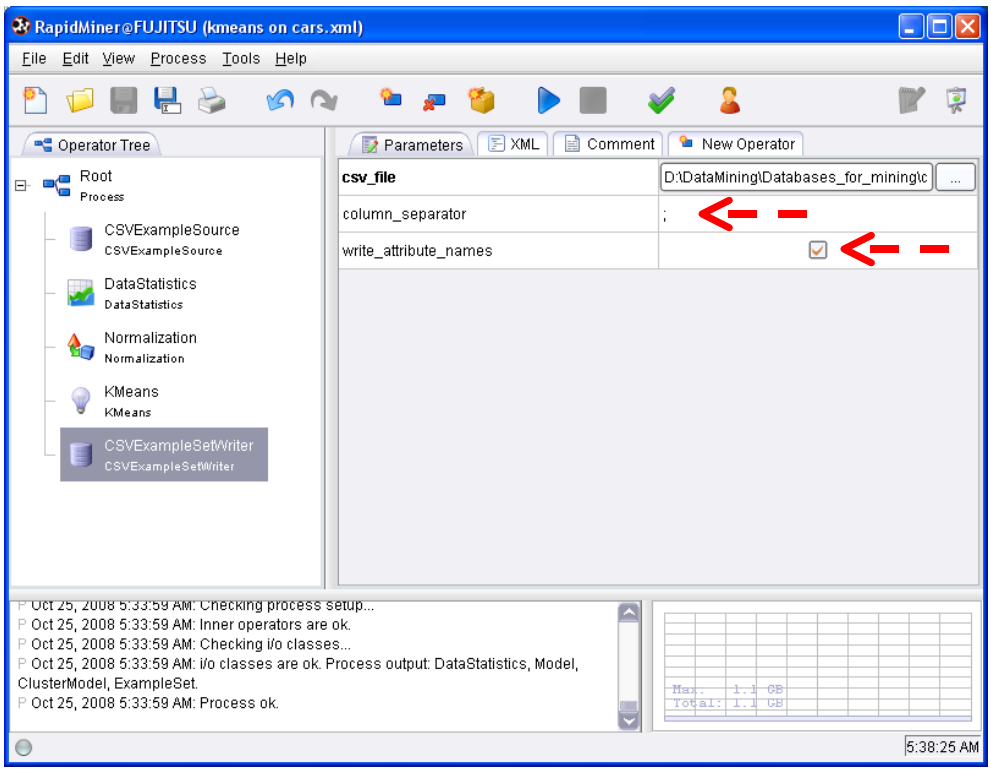
Standardization of the variables. The NORMALIZATION component is used for the standardization of the variables. Various formulas are available, we ask the Z transformation.



K-Means. KMEANS corresponds to the K-Means algorithm. We ask 2 groups (K); and we want to utilize the CLUSTER column in the subsequent part of the diagram (ADD_CLUSTER_ATTRIBUTE). Furthermore, we want that the clusters are characterized with comparative descriptive statistics indicators (ADD_CHARACTERIZATION). The other settings are related to the computation (MAX_RUNS and MAX_OPTIMIZATION_STEPS).

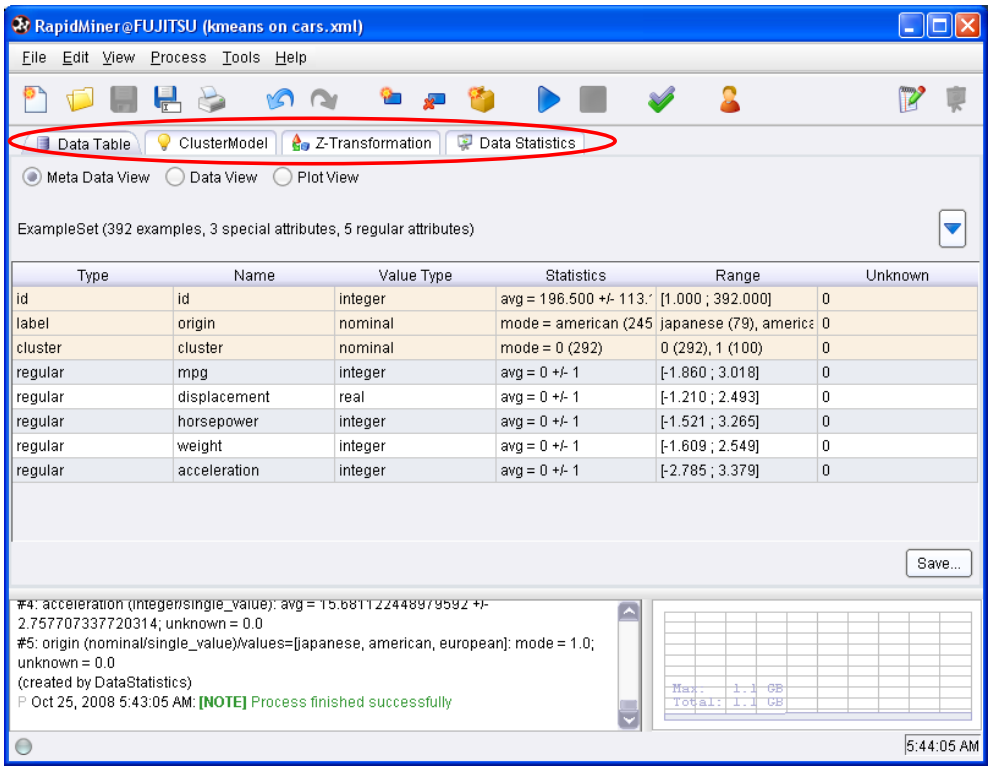


Exporting the dataset including the cluster column. Finally, we export the dataset using the CSVEXAMPLESETWRITER component. We set the data file name and the column separator character.

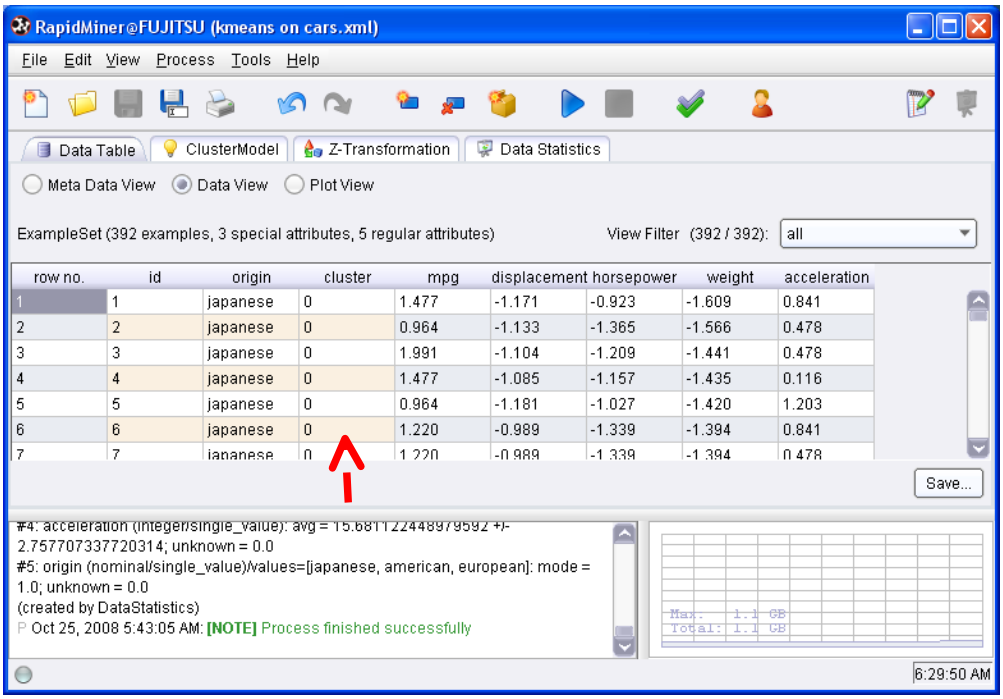


7.2 Examining the results

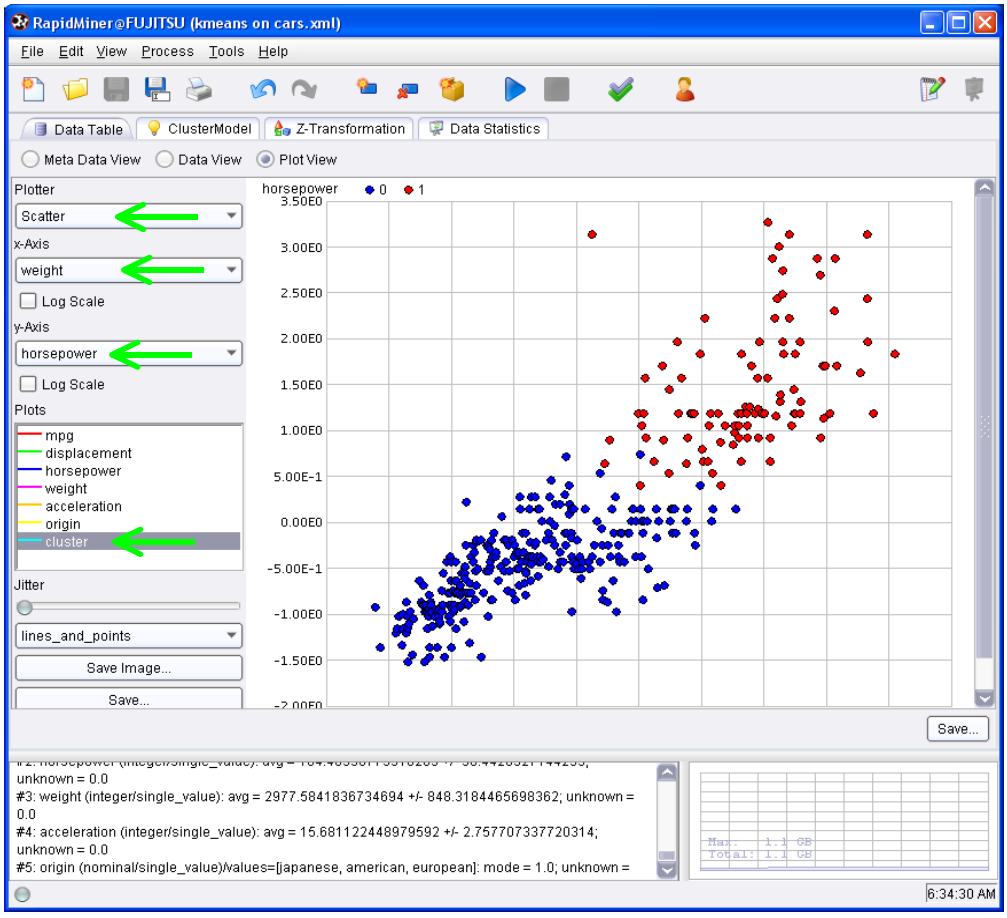
After we save the diagram, we click on the button. A window summarizes the results. We can select the results associated to each component by clicking on the appropriate tab.



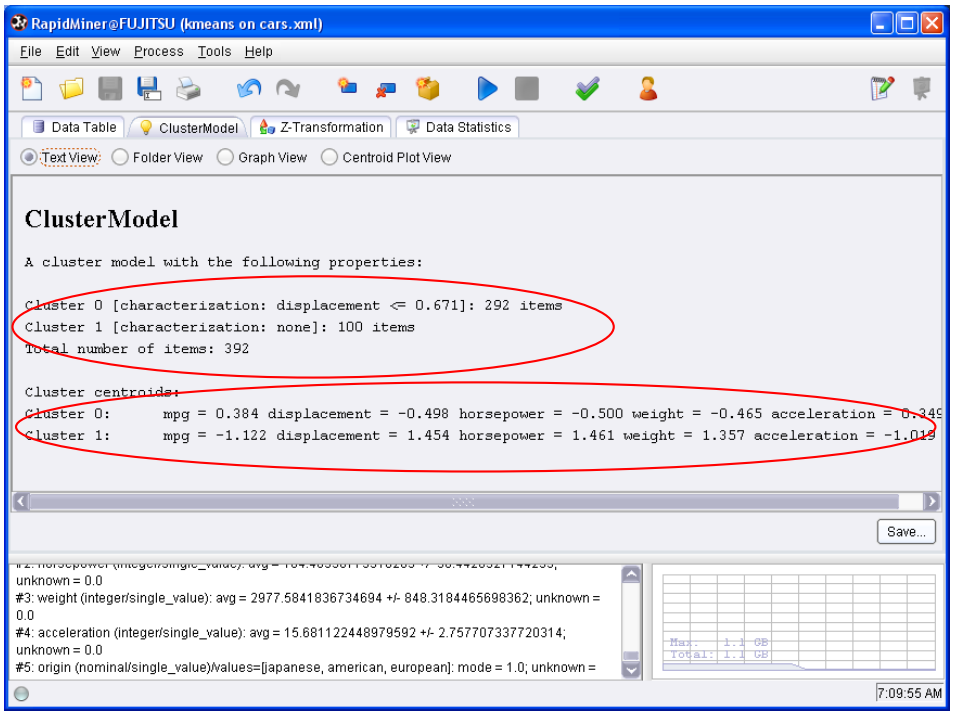
Description of the dataset. The DATA TABLE tab describes the dataset: META DATA VIEW gives the basic characteristics of the variables according their type; DATA VIEW displays the values of the variables, including the CLUSTER column.



PLOT VIEW is a graphical tool. We can create a scatter plot with the SCATTER option.

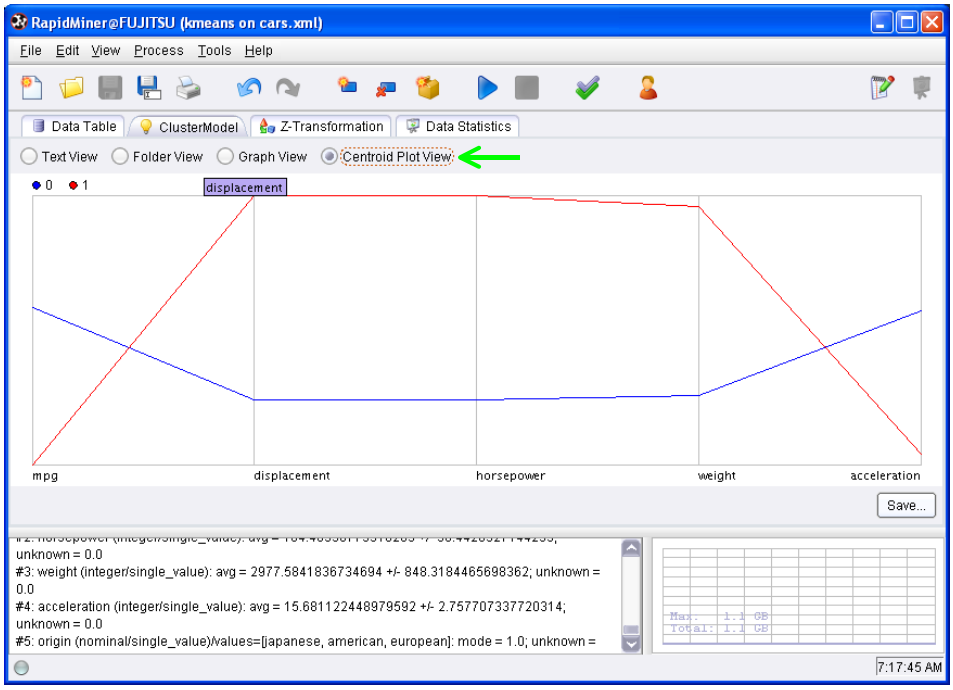


CLUSTERMODEL. This tab describes the results of the clustering process. TEXT VIEW option supplies the number of instances on each group (292 and 100). We obtain also the conditional mean according to the standardized variables.



The FOLDER VIEW and GRAPH VIEW options allow to visualize the cluster membership of each case.

CENTROID PLOT VIEW is a graphical representation of the conditional mean for each variable.



2 other tabs complete the results:

- **Z-TRANFORM.** It describes the parameters used for the standardization of the variables i.e. the mean and the standard deviation of each variable.
- **DATA STATISTICS**. It computes the descriptive statistics indicators.

8 Conclusion

In this tutorial, we show that almost the free tools can perform a K-means clustering algorithm. Even if some details are different, especially for the presentation of the results, we note that they supply comparable results. It is rather encouraging for the utilization of these tools.