Éléments de Data Mining avec Tanagra

Vincent ISOZ, 2013-10-21 (V3.0 Revision 6)
{oUUID 1.679}

TABLE DES MATIÈRES

Introduction

TANAGRA est un logiciel gratuit d'exploration de données (DataMining) destiné à l'enseignement et à la recherche et à l'enseignement créé en 2003. Il implémente une série de méthodes de fouille de données issues du domaine de la statistique exploratoire, de l'analyse de données, de l'apprentissage automatique et des bases de données.

Par rapport à ses concurrents TANAGRA a selon moi quatre avantages majeurs:

1. L'interface est extrêmement simple et sobre et nécessite quasiment zéro effort pour comprendre la logique.

2. Les méthodes statistiques et leurs résultats respectifs sont clairement nommés selon l'usage par les spécialistes de la statistique.

3. La documentation est bien fournie aussi bien en anglais qu'en français avec des annexes accessibles à ceux qui ont des connaissances universitaires en mathématiques.

4. La rapidité de traitement d'une grosse masse de données qui en terme de performance vaut largement la concurrence gratuite (KNIME, Orange, R, RapidMiner, SIPINA, WEKA) ou payant (Oracle, MS SQL Server, SPSS, Statistica) d'après les tests effectués par l'auteur du logiciel (le logiciel est sobre et codé selon les règles de l'art ce qui accapare moins de mémoire).

5. Sa gratuité et le fait que le code source soit disponible à tous.

C'est un projet ouvert au sens qu'il est possible à tout chercheur d'accéder au code, d'ajouter ses propres algorithmes et de diffuser, toujours gratuitement, le logiciel modifié.

Tanagra est diffusé depuis décembre 2003. Il est compilé pour la plate-forme WIN32 mais il est possible de le faire fonctionner sous d'autres systèmes (par ex. avec WINE sous linux).

Précisions sur la licence de TANAGRA (voir le détail de la licence lors de l'installation). Le logiciel TANAGRA est développé à titre personnel par Ricco Rakotomalala. Il en a la propriété exclusive. Un logiciel est une oeuvre de l'esprit au sens du code de la propriété intellectuelle (Article L.112-2), exactement comme les ouvrages. Ricco Rakotomalala s'engage à rendre la version complète de TANAGRA indéfiniment gratuite sans aucune restriction. Le code source sera toujours librement accessible en ligne. Si une entité quelconque introduit des contraintes quant à l'accès au logiciel (ex. nécessité de s'enregistrer pour télécharger ; versions volontairement bridées avec des promesses de fonctionnalités étendues sur une variante améliorée payante ; code source non publié ; incorporation dans un package commercial ; ou que sais-je encore...), vous êtes face à une distribution illicite.

L'utilisation du logiciel est totalement libre, dans quelque contexte que ce soit, y compris dans le cadre d'une activité commerciale. Si vous souhaitez citer TANAGRA dans vos travaux de recherche, voici la référence à utiliser : **Ricco Rakotomalala, "TANAGRA : un logiciel gratuit pour l'enseignement et la recherche", in Actes de EGC'2005, RNTI-E-3, vol. 2, pp.697-702, 2005.**

Logiciels de Data Mining

Tanagra est certes très complet pour la majorité des besoins mais il ne peut convenir cependant qu'à des situations où:

1. il n'est pas nécessaire d'avoir des résultats en temps réel sur des serveurs de bases de données

2. l'utilisation de scripts d'automatisation de post ou prétraitement n'est pas nécessaire (pas de macros par exemple)

3. Il n'y pas de support technique pour répondre aux questions (du moins à ma connaissance)

et c'est aussi le cas pour d'autres logiciels gratuits de Data Mining comme S-Plus de Insight, Alice de Isoft, Predic de Neuralware, R (version gratuite de S-Plus), Weka et RapidMiner (sauf changement entre le moment où ces lignes ont été écrites et le moment où vous les lisez).

Cependant en matière de quantités de techniques, d'ergonomie et de rapidité d'enseignement, Tanagra est selon mon expérience personnelle loin devant pour l'enseignement en entreprise et à l'université.

Sinon, pour avoir testé sur un jeu d'un peu plus de 1.1 million de données que j'utilise dans le cade des mes formations (traitements effectués souvent en moins de dix secondes), nous pouvons très probablement sans problèmes utiliser Tanagra pour faire des analyses sur des bases de données de l'ordre de la dizaine de millions de données (par extrapolation au pouce...).

Sinon, les logiciels payants les plus connus en ce tout début de 21ème siècle seraient: SPSS Clementine, SAS Enterpise Miner, Statistica Data Miner, S-Plus Insightful Miner, Matlab et KXen ou RapidMiner si l'on fait appel aux services de consulting et de déploiement + installation.

Avertissements

Le but de ce support a pour but de mettre en pratique les démonstrations mathématiques théoriques effectuées lors des cours de statistiques et de méthodes numériques.

Le contenu du présent support est élaboré par un processus de développement par lequel des experts de la gestion de projets parviennent à un consensus. Ce processus qui rassemble des participants bénévoles recherche également les points de vue de personnes intéressées par le sujet de cet ouvrage. En tant que responsable du présent support, j'assure l'administration du processus et je fixe les règles qui permettent de promouvoir l'équité dans l'approche d'un consensus. Je me charge également de rédiger les textes, parfois de les tester/évaluer ou de vérifier indépendamment l'exactitude/solidité ou l'exhaustivité des informations présentées.

Je décline toute responsabilité en cas de dommages corporels, matériels ou autres de quelque nature que ce soit, particuliers, indirects, accessoires ou compensatoires, résultant de la publication, de l'application ou de la confiance accordée au contenu du présent support. Je n'émets aucune garantie expresse ou implicite quant à l'exactitude ou à l'exhaustivité de toute information publiée dans le présent support, et ne garantit aucunement que les informations contenues dans cet ouvrage satisfassent un quelconque objectif ou besoin spécifique du lecteur. Je ne garantis pas non plus les performances de produits ou de services d'un fabricant ou d'un vendeur par la seule vertu du contenu du présent support.

En publiant des textes, il n'est pas dans l'intention principale du présent support de fournir des services de spécialistes ou autres au nom de toute personne physique ou morale ni pour mon compte, ni d'effectuer toute tâche devant être accomplie par toute personne physique ou morale au bénéfice d'un tiers. Toute personne utilisant le présent support devrait s'appuyer sur son propre jugement indépendant ou, lorsque cela s'avère approprié, faire appel aux conseils d'un spécialiste compétent afin de déterminer comment exercer une prudence raisonnable en toute circonstance. Les informations et les normes concernant le sujet couvert par le présent support peuvent être disponibles auprès d'autres sources que le lecteur pourra souhaiter consulter en quête de points de vue ou d'informations supplémentaires qui ne seraient pas couverts par le contenu du présent site Internet.

Je ne dispose (malheureusement...) d'aucun pouvoir dans le but de faire respecter la conformité au contenu du présent ouvrage, et je ne m'engage nullement à surveiller ni à faire respecter une telle conformité. Je n'exerce (à ce jour...) aucune activité de certification, de test ni d'inspection de produits, de conceptions ou d'installations à fins de santé ou de sécurité des personnes et des biens. Toute certification ou autre déclaration de conformité en matière d'informations ayant trait à la santé ou à la sécurité des personnes et des biens, mentionnée dans le présent support, ne peut aucunement être attribuée au contenu du présent support et demeure sous l'unique responsabilité de l'organisme de certification ou du déclarant concerné.

Objectifs

J'ai tenté de mettre les exemples dans l'ordre de difficulté croissant et j'espère avoir atteint cet objectif pédagogique. Les premiers exemples sont vraiment élémentaires (ils ne dépassent pas le niveau du BAC) et faisables avec un simple tableur mais ils permettent au moins de se faire la main sur les manipulations courantes du logiciel.

Actuellement seulement **9 composants de Data Mining sur les 180 disponibles dans le logiciel** sont présentés dans ce support (sachant que 170 sont vraiment des techniques de fouilles de données). Je rédige un exemple à peu tous les 3 mois... depuis le **30 Avril 2011** sachant que je me limite à présenter uniquement les techniques pour lesquelles la démonstration mathématique détaillée et pédagogique (soit une trentaine à ce jour) se trouve sur déjà sur mon site www.sciences.ch (ou que j'ai déjà rédigée mais pas encore eu le temps de publier en ligne sur le site). Bien évidemment, si des lecteurs (étudiants / professeurs / passionnés) veulement m'aider à rédiger les démonstrations mathématiques... **toute contribution/aide est la bienvenue pour compléter les démonstrations mathématiques détaillées manquantes**!

Voici ci-dessous la liste des techniques et composants disponibles sur Tanagra. Celles qui sont précédées d'un ✓ ont été étudiées dans les détails dans le cours théorique et elles sont (ou seront) détaillées dans le présent support (pour les autres, il me manque les démonstrations mathématiques à un niveau de rigueur pouvant être considéré comme satisfaisant):

Data visualisation

- ✓ Correlation scatterplot
- ✓ Export dataset
- ✓ Scatterplot
- ✓ Scatterplot with label
- ✓ View multiple scatterplot

Statistics

- ANOVA Randomized Blocks
- Bartlett's test
- ✓ Brown - Forsythe's test
- Box's M Test
- ✓ Fisher's test
- ✓ Group characterization
- ✓ Group exploration
- Hotelling's T2
- Hotelling's T2 Heteroscedastic
- ✓ Levene's test
- ✓ Linear correlation
- ✓ More Univariate cont stat
- ✓ Normality Test
- ✓ One-way ANOVA
- One-way MANOVA

- ✓ Paired T-Test
- Paired V-Test
- ✓ Partial Correlation
- ✓ Semi-partial Correlation
- ✓ T-Test
- ✓ T-Test Unequal Variance
- ✓ Univariate continuous stat
- ✓ Univariate discrete stat
- ✓ Univariate Outlier Detection
- Welch ANOVA

Nonparametric statistics

- Ansari-Bradley Scale Test
- Categorical r
- Cochran's Q-test
- ✓ Contingency Chi-Square
- ✓ Friedman's ANOVA by Ranks
- FYTH 1-way ANOVA (Fisher-Yates-Terry-Hoeffding)
- Goodman Kruskal Gamma
- Goodman-Kruskal Lambda
- Goodman-Kruskal Tau
- Kendall Tau-b
- Kendall Tau-c
- ✓ Kendall's Concordance W
- Kendall's tau
- Klotz Scale Test
- ✓ Kruskal-Wallis 1-way ANOVA
- ✓ K-S 2-sample test
- ✓ Mann-Whitney Comparison
- ✓ Median test
- ✓ Mood Runs Test
- Mood Scale Test
- Partial Theil U
- ✓ Sign Test
- Sommers d
- Theil U
- Van der Waerden 1-way ANOVA
- ✓ Wald-Wolfowitz Runs Test
- ✓ Wilcoxon Signed Ranks Test

Instance selection

- ✓ Continuous select examples
- ✓ Discrete select examples
- ✓ Recover examples

- ✓ Rule-based selection
- ✓ Sampling
- ✓ Select first examples
- Stratified sampling

Feature construction

- ✓ 0_1_Binarize
- ✓ Binary binning
- ✓ Cont to disc
- ✓ Disc to cont
- ✓ EqFreq Disc
- ✓ EqWidth Disc
- ✓ Formula
- MDLPC (Minimum Description Length Principle Cut)
- Residual Scores
- ✓ Standardize
- ✓ Trend

Feature selection

- Backward-logit
- CFS filtering (Correlation Feature Selection)
- ✓ Define status
- FCBF filtering (Fast Correlation Based Filter)
- Feature ranking
- Fisher filtering
- Forward-logit
- MIFS filtering (Metamaterial Isoindex Filtering Selection)
- MODTree filtering (Multivalued Oblivious Decision Tree)
- ReliefF
- Remove constant
- Runs filtering
- Stepdisc

Regression

- ✓ Backward Elimination Reg
- C-RT Regression tree
- ✓ DfBetas
- Espilon SVR
- ✓ Forward Entry Regression
- ✓ Multiple linear regression
- Nu SVR
- Outlier Detection
- ✓ Regression Assessment

✓ Regression tree
Simultaneous Regression

Factorial analysis

AFDM
Bootstrap Eigenvalues
Canonical Discriminant Analysis
✓ Correspondence Analysis
Discriminant Correspondence Analysis
Factor rotation
Harris Component Analysis
✓ Multiple Correspondence Analysis
NIPALS (Nonlinear Iterative Partial Least Squares)
Parallel Analysis
✓ Principal Component Analysis
Principal Factor Analysis

PLS

PLS Conf. Interval (Partial Least Squares Confidence)
PLS Factorial
✓ PLS Regression
PLS Selection
PLSR (exécute PLS Factorial et le PLS Regression en même temps)

Clustering

CT
CTP (Clustering Tree Post-prunning)
EM-Clustering (Expectation-Maximization clustering)
EM-Selection
✓ HAC (Hierarchical Clustering)
✓ K-Means
Kohonen-SOM
LVQ (Learning Vector Quantized)
Neighborhood Graph
VARCLUS
VARHCA (Variable Hierarchical Clustering Analysis)
VARKMeans

SPV (Support Vector) Learning

✓ Binary logistic regression
C4.5
C-PLS
C-RT
CS-CRT (Cost Sensitive Classification Regression Tree)

CS-MC4 (Cost Sensitive Missclassification Cost Matrix)
C-SVC
Decision List
ID3
K-NN (K Nearest Neighbor)
Linear discriminant analysis
Log-Reg TRIRLS
Multilayer perceptron
Multinomial Logistic Regression
Naive bayes
Naive bayes continuous
PLS-DA (Discriminant Analysis)
PLS-LDA (Linear Discriminant Analysis)
Prototype-NN
Radial basis function
Rnd Tree
Rule Induction
SVM

Meta SPV (Support Vector) Learning

Arcing [Arc-x4]
Bagging
Boosting
Cost Sensitive Bagging
Cost Sensitive Learning
MultiCost
Supervised Learning

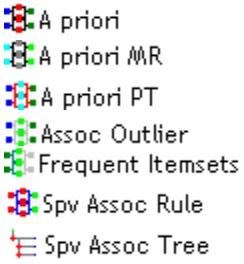
SPV (Support Vector) Learning assessment

Bias-variance decomposition
Bootstrap
Cross-validation
Hosmer Lemeshow Test
Leave-One-Out
Logistic Regression Residuals
Test
Train-test

Scoring

Lift curve
Posterior Prob
Precision-Recall curve
Reliability Diagram
Roc curve
Scoring

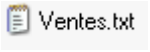
Association



Exercice 1.: Import et visualisation des données *.txt

Tanagra V1.4.36

A partir du fichier texte suivant se trouvant dans votre dossier d'exercice:



Contenant des données séparées par des tabulations (Tanagra impose les tabulations!):

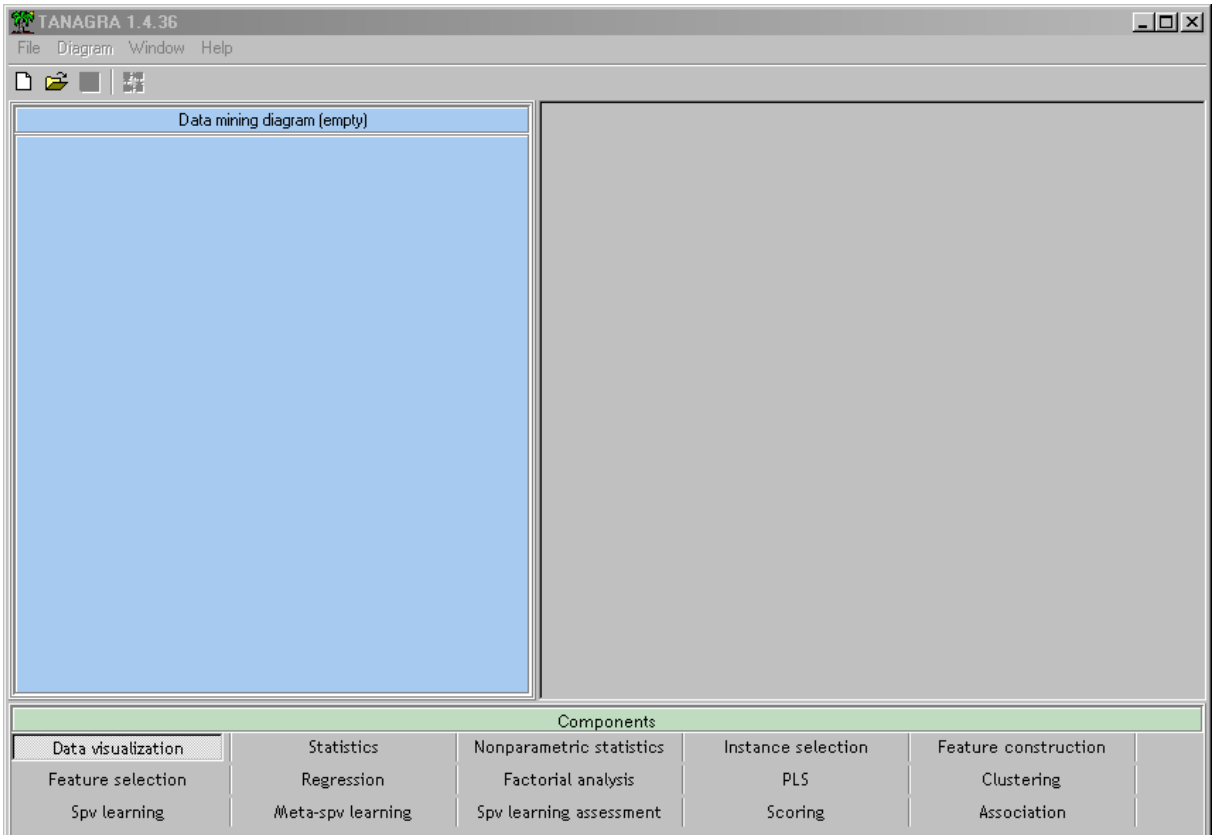
Ventes.txt - Bloc-notes

Fichier Edition Format Affichage ?

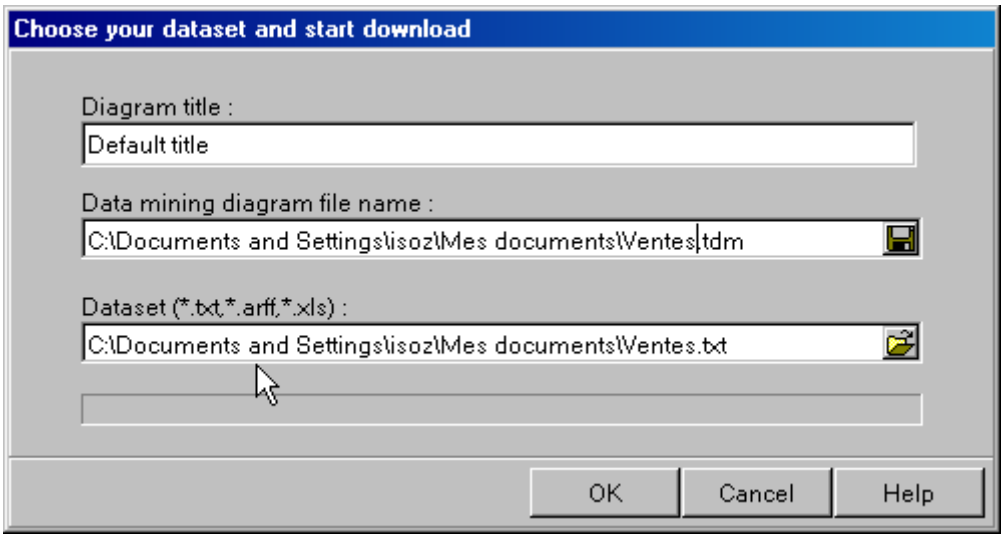
N° Client	Activité	N° de Commande	Date de commande		Article
100	Assurances	1	03.01.2000	Compaq Presario 100	12
123	Machines/Outils	2	03.01.2000	IBM 500 2	2299	0.00%
109	Éducation	3	03.01.2000	AST Intel 150	5	2690
104	Éducation	4	03.01.2000	AST Intel 200	3	3190
117	Banques 5	04.01.2000	Compaq Presario 100	13	1650
103	Assurances	6	04.01.2000	AST Intel 150	2	2690
104	Éducation	7	04.01.2000	AST Intel 200	2	3190
111	Alimentaire	8	04.01.2000	IBM 500 4	2299	0.00%
113	Construction	9	04.01.2000	Compaq Presario 100	4
116	Pharmaceutique	10	04.01.2000	IBM 500 2	2299	0.00%
110	Distribution	11	05.01.2000	AST Intel 200	6	3190
112	Machines/Outils	12	05.01.2000	Compaq Presario 100	6
123	Machines/Outils	13	05.01.2000	IBM 500 6	2299	1.50%
113	Construction	14	05.01.2000	AST Intel 150	3	2690
115	Distribution	15	05.01.2000	Compaq Presario 100	8
124	Éducation	16	05.01.2000	AST Intel 200	8	3190
124	Éducation	17	05.01.2000	Compaq Presario 100	11
106	Construction	18	05.01.2000	AST Intel 200	11	3190
101	Construction	19	05.01.2000	Compaq Presario 100	14
116	Pharmaceutique	20	06.01.2000	IBM 500 7	2299	1.50%
112	Machines/Outils	21	06.01.2000	AST Intel 150	6	2690
125	Construction	22	06.01.2000	Compaq Presario 100	23
100	Assurances	23	06.01.2000	IBM 500 3	2299	0.00%
125	Construction	24	06.01.2000	AST Intel 200	2	3190
104	Éducation	25	07.01.2000	AST Intel 150	12	2690
126	Machines/Outils	26	07.01.2000	AST Intel 150	24	2690
121	Pharmaceutique	27	07.01.2000	IBM 500 8	2299	1.50%
114	Distribution	28	07.01.2000	AST Intel 200	9	3190
103	Assurances	29	07.01.2000	Compaq Presario 100	6
125	Construction	30	07.01.2000	AST Intel 200	4	3190
120	Banques 31	10.01.2000	AST Intel 150	2	2690	0.00%
111	Alimentaire	32	10.01.2000	Compaq Presario 100	16
118	Éducation	33	10.01.2000	IBM 500 3	2299	0.00%
127	Alimentaire	34	10.01.2000	IBM 500 7	2299	1.50%
101	Construction	35	11.01.2000	AST Intel 200	6	3190
118	Éducation	36	11.01.2000	Compaq Presario 100	5
119	Distribution	37	11.01.2000	AST Intel 200	23	3190
106	Construction	38	11.01.2000	IBM 500 4	2299	0.00%
121	Pharmaceutique	39	11.01.2000	AST Intel 150	7	2690
104	Éducation	40	12.01.2000	Compaq Presario 100	12
101	Construction	41	12.01.2000	AST Intel 200	1	3190
106	Construction	42	12.01.2000	AST Intel 150	9	2690
113	Construction	43	12.01.2000	IBM 500 6	2299	1.50%
127	Alimentaire	44	12.01.2000	AST Intel 200	2	3190
100	Assurances	45	12.01.2000	AST Intel 200	4	3190
109	Éducation	46	13.01.2000	AST Intel 150	1	2690

Effectuez les opérations nécessaires pour visualisez les données contenues dans ce fichier directement depuis Tanagra.

Ouvrons Tanagra:

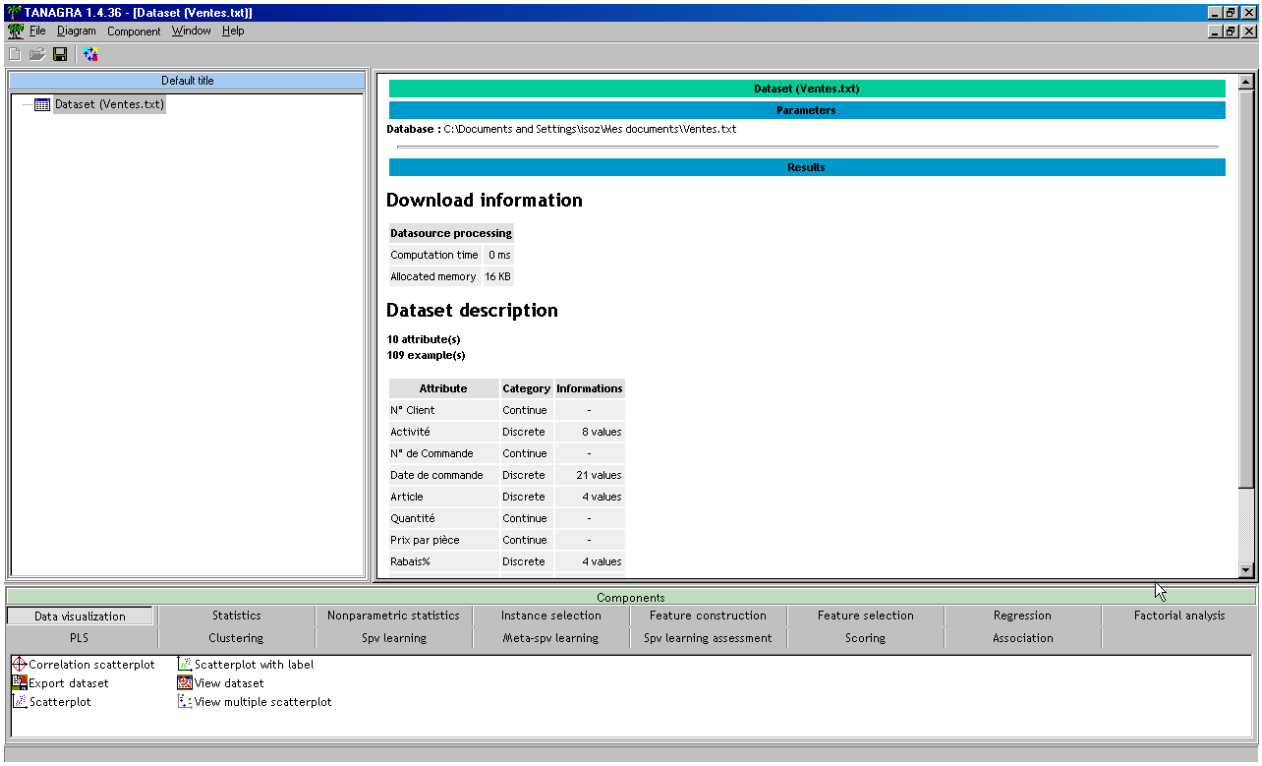


Allez dans le menu **File/New…**:

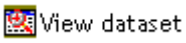


Puis entrez un nom pour le diagramme (par exemple **VisualisationDonnees**) ensuite un nom et un chemin pour le fichier Tanagra (*.tdm: **Tanagra Diagram**) et enfin allez cherchez la source de données dans le champ **Dataset** comme visible sur la capture ci-dessus.

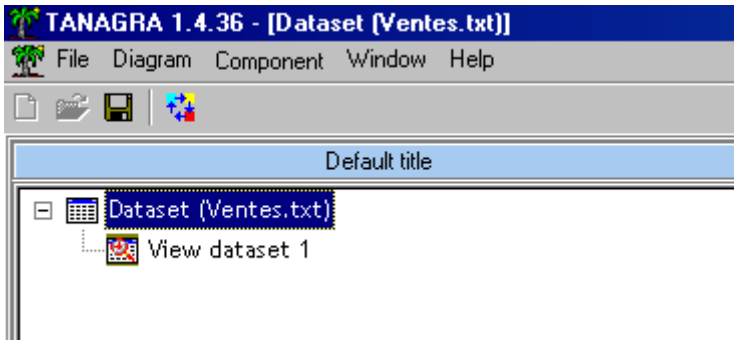
Validez par **OK** et vous aurez alors:



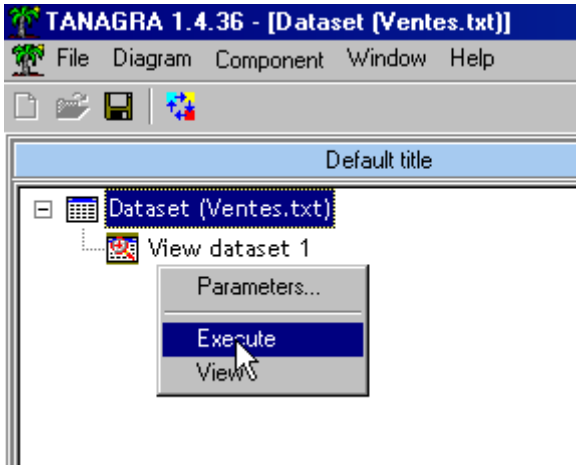
Depuis la catégorie des composants **Components** se trouvant dans la partie inférieure du logiciel, glissez l'opérateur nommé **View dataset** de la catégorie **Data visualization**:



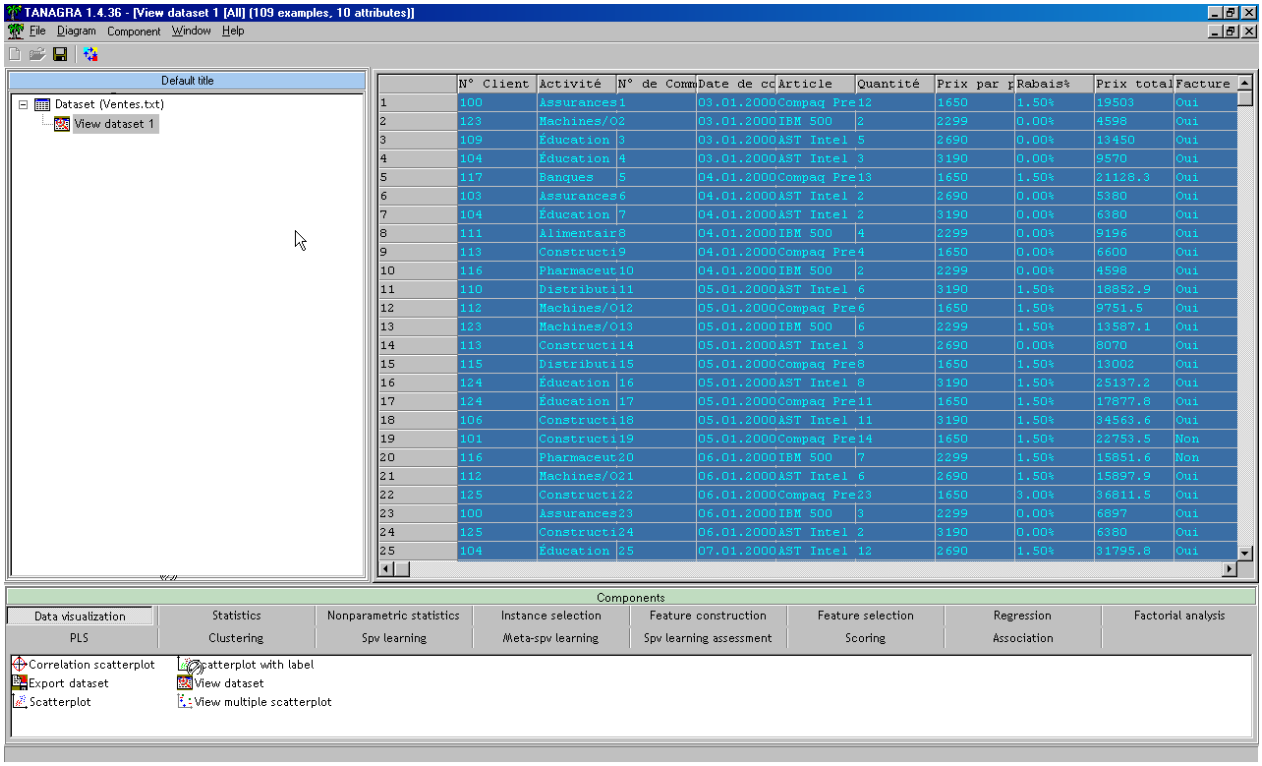
sur le **Dataset** afin d'obtenir:



Ensuite faites un clic droit sur l'opérateur **View dataset 1**:



et cliquez sur **Execute**. Refaites la même manipulation ensuite puis cliquez sur **View**. Vous aurez alors un visuel des données du fichier:



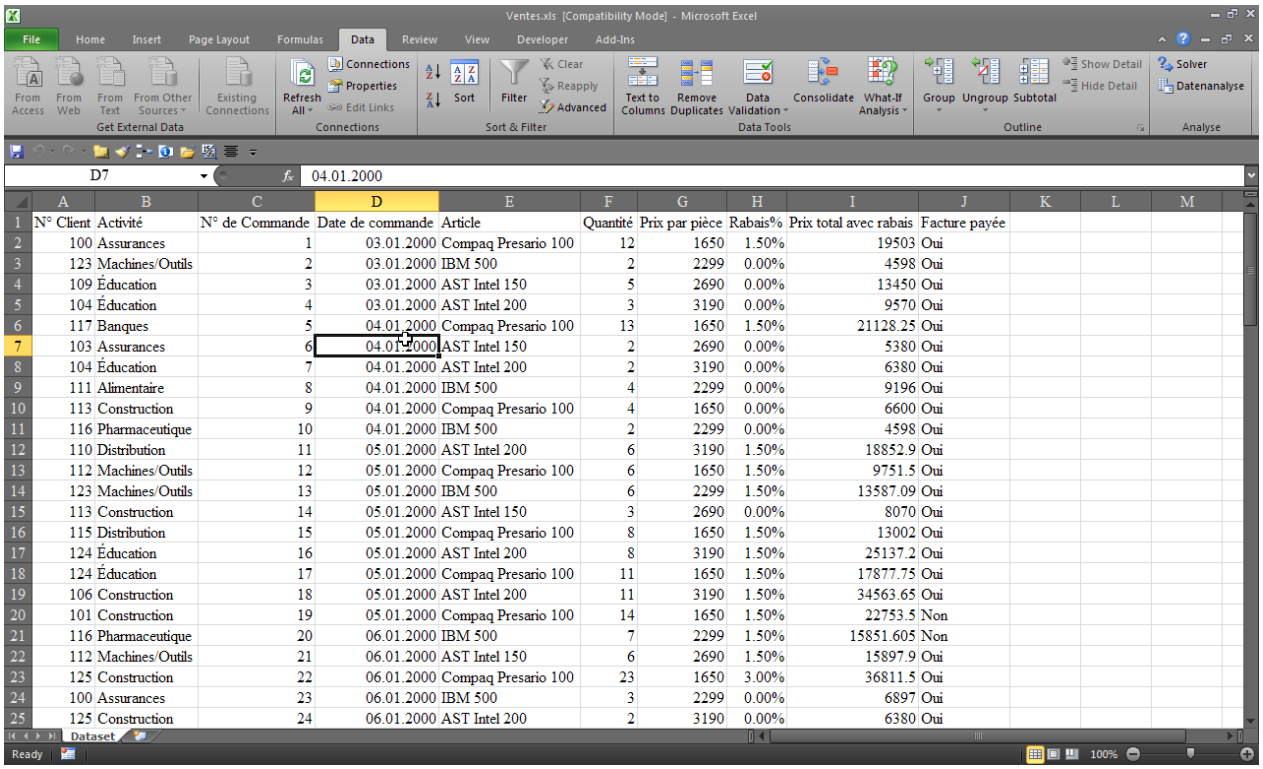
	N° Client	Activité	N° de Comm	Date de cc	Article	Quantité	Prix par p	Rabais%	Prix total	Facture
1	100	Assurances	1	03.01.2000	Compaq Pre	12	1650	1.50%	19503	Oui
2	123	Machines/O	2	03.01.2000	IBM 500	2	2299	0.00%	4598	Oui
3	109	Éducation	3	03.01.2000	AST Intel	5	2690	0.00%	13450	Oui
4	104	Éducation	4	03.01.2000	AST Intel	3	3190	0.00%	9570	Oui
5	117	Banques	5	04.01.2000	Compaq Pre	13	1650	1.50%	21128.3	Oui
6	103	Assurances	6	04.01.2000	AST Intel	2	2690	0.00%	5380	Oui
7	104	Éducation	7	04.01.2000	AST Intel	2	3190	0.00%	6380	Oui
8	111	Alimentair	8	04.01.2000	IBM 500	4	2299	0.00%	9196	Oui
9	113	Constructi	9	04.01.2000	Compaq Pre	4	1650	0.00%	6600	Oui
10	116	Pharmaceut	10	04.01.2000	IBM 500	2	2299	0.00%	4598	Oui
11	110	Distributi	11	05.01.2000	AST Intel	6	3190	1.50%	18852.9	Oui
12	112	Machines/O	12	05.01.2000	Compaq Pre	6	1650	1.50%	9751.5	Oui
13	123	Machines/O	13	05.01.2000	IBM 500	6	2299	1.50%	13587.1	Oui
14	113	Constructi	14	05.01.2000	AST Intel	3	2690	0.00%	8070	Oui
15	115	Distributi	15	05.01.2000	Compaq Pre	8	1650	1.50%	13002	Oui
16	124	Éducation	16	05.01.2000	AST Intel	8	3190	1.50%	25137.2	Oui
17	124	Éducation	17	05.01.2000	Compaq Pre	11	1650	1.50%	17877.8	Oui
18	106	Constructi	18	05.01.2000	AST Intel	11	3190	1.50%	34563.6	Oui
19	101	Constructi	19	05.01.2000	Compaq Pre	14	1650	1.50%	22753.5	Non
20	116	Pharmaceut	20	06.01.2000	IBM 500	7	2299	1.50%	15851.6	Non
21	112	Machines/O	21	06.01.2000	AST Intel	6	2690	1.50%	15897.9	Oui
22	125	Constructi	22	06.01.2000	Compaq Pre	23	1650	3.00%	36811.5	Oui
23	100	Assurances	23	06.01.2000	IBM 500	3	2299	0.00%	6897	Oui
24	125	Constructi	24	06.01.2000	AST Intel	2	3190	0.00%	6380	Oui
25	104	Éducation	25	07.01.2000	AST Intel	12	2690	1.50%	31795.8	Oui

Exercice 2.: Import et visualisation des données *.xls

Tanagra V1.4.36

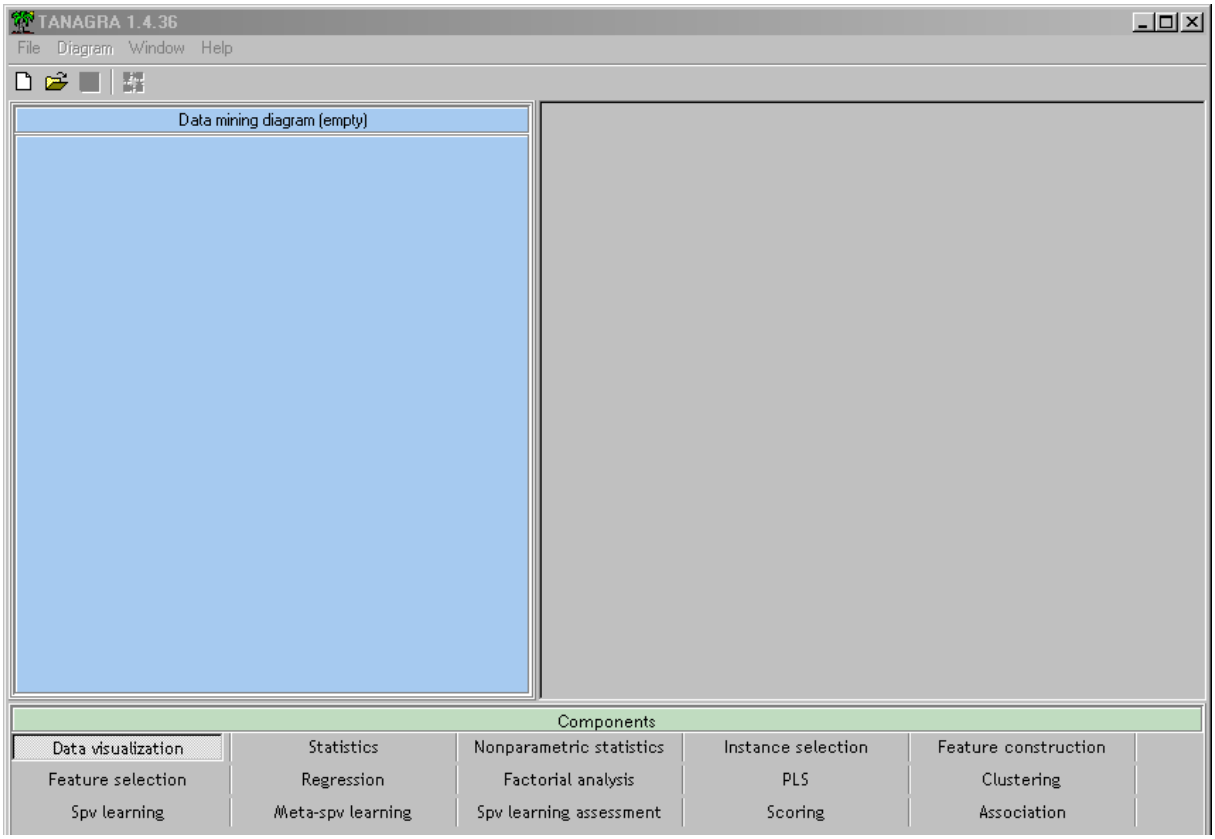
A partir du fichier texte suivant se trouvant dans votre dossier d'exercice:

Contenant les mêmes données que le fichier *.txt précédent:

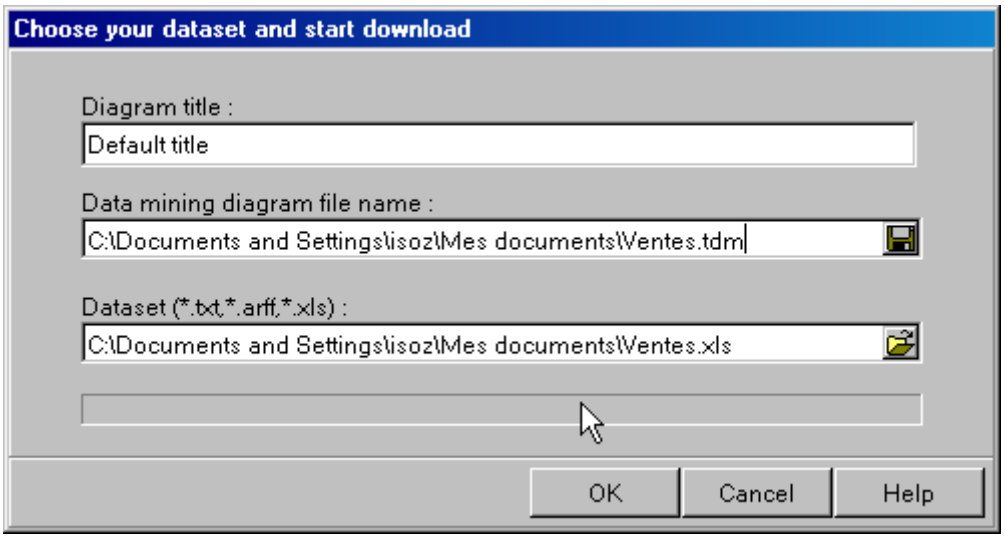


	A	B	C	D	E	F	G	H	I	J
1	N° Client	Activité	N° de Commande	Date de commande	Article	Quantité	Prix par pièce	Rabais%	Prix total avec rabais	Facture payée
2	100	Assurances	1	03.01.2000	Compaq Presario 100	12	1650	1.50%	19503	Oui
3	123	Machines/Outils	2	03.01.2000	IBM 500	2	2299	0.00%	4598	Oui
4	109	Éducation	3	03.01.2000	AST Intel 150	5	2690	0.00%	13450	Oui
5	104	Éducation	4	03.01.2000	AST Intel 200	3	3190	0.00%	9570	Oui
6	117	Banques	5	04.01.2000	Compaq Presario 100	13	1650	1.50%	21128.25	Oui
7	103	Assurances	6	04.01.2000	AST Intel 150	2	2690	0.00%	5380	Oui
8	104	Éducation	7	04.01.2000	AST Intel 200	2	3190	0.00%	6380	Oui
9	111	Alimentaire	8	04.01.2000	IBM 500	4	2299	0.00%	9196	Oui
10	113	Construction	9	04.01.2000	Compaq Presario 100	4	1650	0.00%	6600	Oui
11	116	Pharmaceutique	10	04.01.2000	IBM 500	2	2299	0.00%	4598	Oui
12	110	Distribution	11	05.01.2000	AST Intel 200	6	3190	1.50%	18852.9	Oui
13	112	Machines/Outils	12	05.01.2000	Compaq Presario 100	6	1650	1.50%	9751.5	Oui
14	123	Machines/Outils	13	05.01.2000	IBM 500	6	2299	1.50%	13587.09	Oui
15	113	Construction	14	05.01.2000	AST Intel 150	3	2690	0.00%	8070	Oui
16	115	Distribution	15	05.01.2000	Compaq Presario 100	8	1650	1.50%	13002	Oui
17	124	Éducation	16	05.01.2000	AST Intel 200	8	3190	1.50%	25137.2	Oui
18	124	Éducation	17	05.01.2000	Compaq Presario 100	11	1650	1.50%	17877.75	Oui
19	106	Construction	18	05.01.2000	AST Intel 200	11	3190	1.50%	34563.65	Oui
20	101	Construction	19	05.01.2000	Compaq Presario 100	14	1650	1.50%	22753.5	Non
21	116	Pharmaceutique	20	06.01.2000	IBM 500	7	2299	1.50%	15851.605	Non
22	112	Machines/Outils	21	06.01.2000	AST Intel 150	6	2690	1.50%	15897.9	Oui
23	125	Construction	22	06.01.2000	Compaq Presario 100	23	1650	3.00%	36811.5	Oui
24	100	Assurances	23	06.01.2000	IBM 500	3	2299	0.00%	6897	Oui
25	125	Construction	24	06.01.2000	AST Intel 200	2	3190	0.00%	6380	Oui

Ouvrons Tanagra:

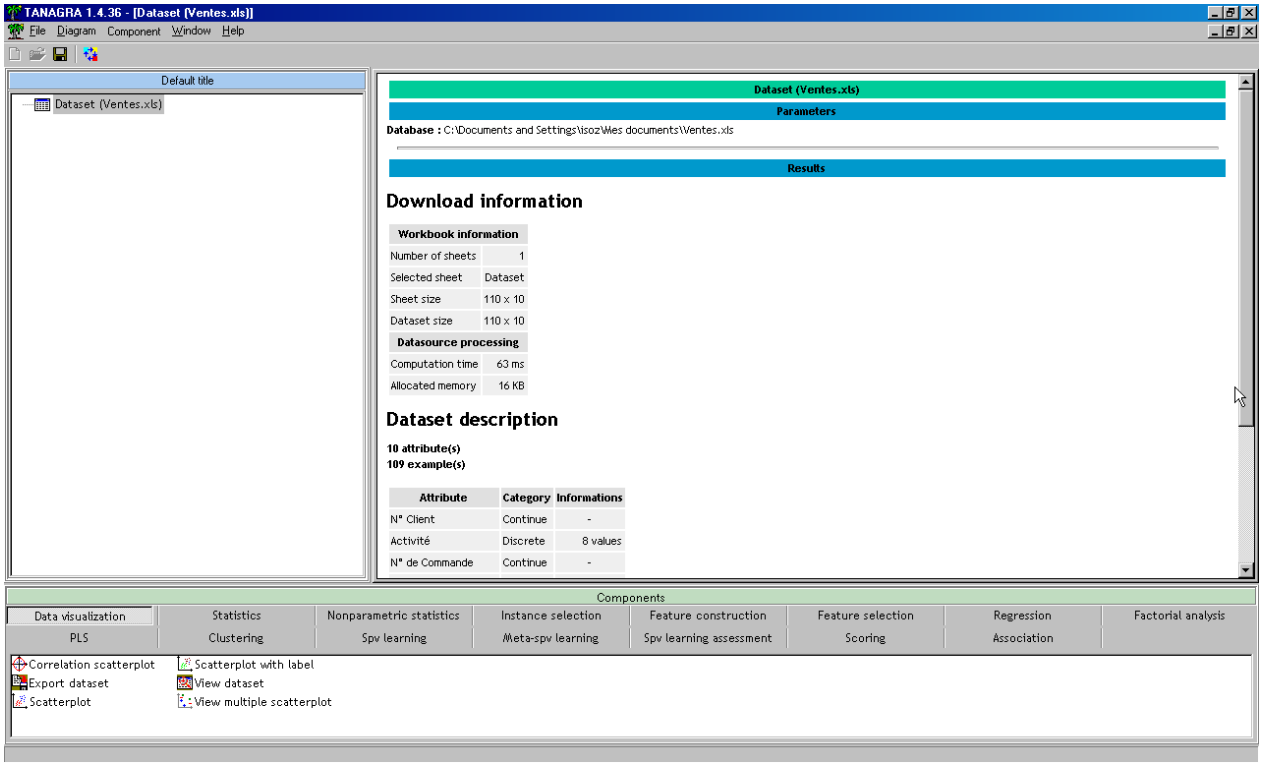


Allez dans le menu **File/New…**

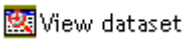


Puis entrez un nom pour le diagramme (par exemple **VisualisationDonnees**) ensuite un nom et un chemin pour le fichier Tanagra (*.tdm: **Tanagra Diagram**) et enfin allez cherchez la source de données dans le champ **Dataset** comme visible sur la capture ci-dessus.

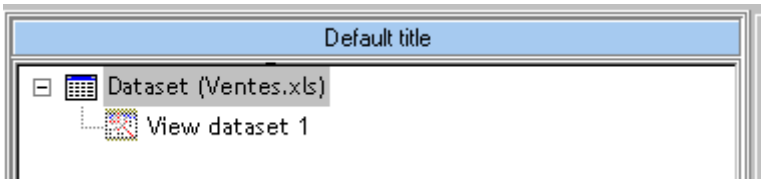
Validez par **OK** et vous aurez alors:



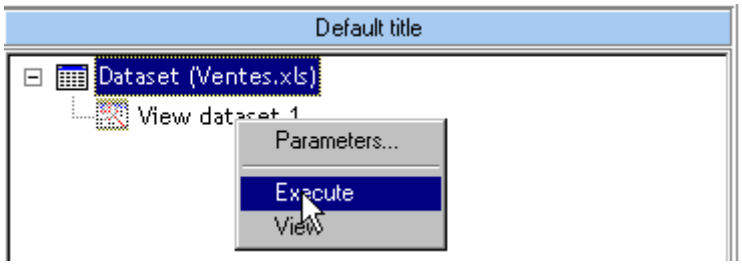
Depuis la catégorie des composants **Components** se trouvant dans la partie inférieure du logiciel, glissez l'opérateur nommé **View dataset** de la catégorie **Data visualization**:



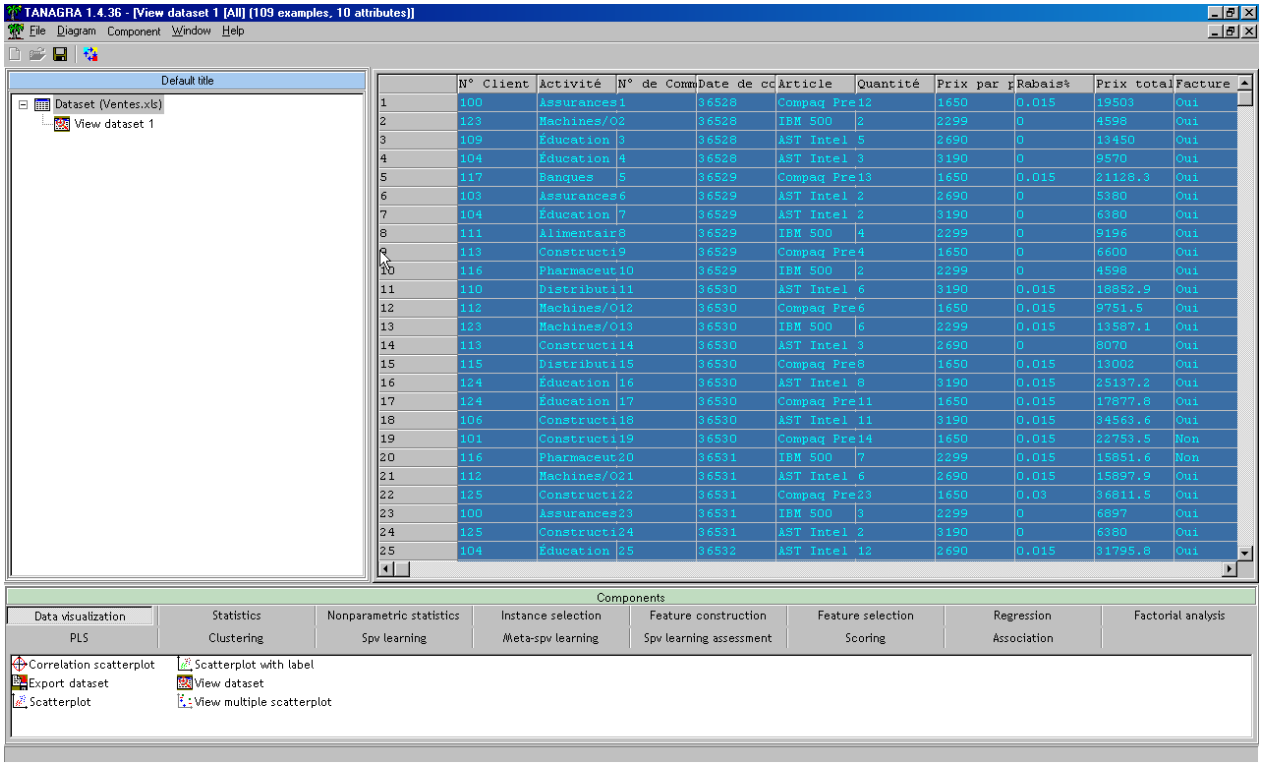
sur le **Dataset** afin d'obtenir:



Ensuite faites un clic droit sur l'opérateur **View dataset 1**:



et cliquez sur **Execute**. Refaites la même manipulation ensuite puis cliquez sur **View**. Vous aurez alors un visuel des données du fichier:



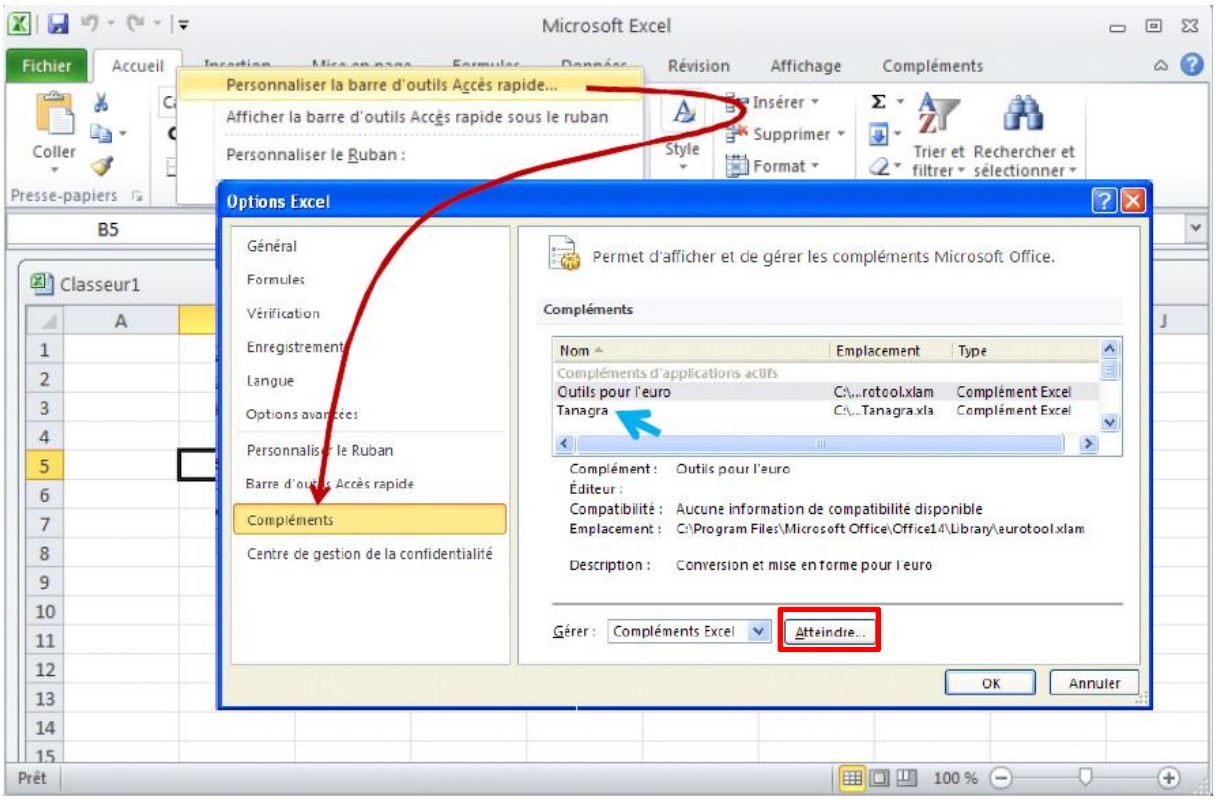
	N° Client	Activité	N° de Comm	Date de cc	Article	Quantité	Prix par p	Rabais%	Prix total	Facture
1	100	Assurances	1	36528	Compaq Pre	12	1650	0.015	19503	Oui
2	123	Machines/O	2	36528	IBM 500	2	2299	0	4598	Oui
3	109	Éducation	3	36528	AST Intel	5	2690	0	13450	Oui
4	104	Éducation	4	36528	AST Intel	3	3190	0	9570	Oui
5	117	Banques	5	36529	Compaq Pre	13	1650	0.015	21128.3	Oui
6	103	Assurances	6	36529	AST Intel	2	2690	0	5380	Oui
7	104	Éducation	7	36529	AST Intel	2	3190	0	6380	Oui
8	111	Alimentair	8	36529	IBM 500	4	2299	0	9196	Oui
9	113	Constructi	9	36529	Compaq Pre	4	1650	0	6600	Oui
10	116	Pharmaceut	10	36529	IBM 500	2	2299	0	4598	Oui
11	110	Distributi	11	36530	AST Intel	6	3190	0.015	18852.9	Oui
12	112	Machines/O	12	36530	Compaq Pre	6	1650	0.015	9751.5	Oui
13	123	Machines/O	13	36530	IBM 500	6	2299	0.015	13587.1	Oui
14	113	Constructi	14	36530	AST Intel	3	2690	0	8070	Oui
15	115	Distributi	15	36530	Compaq Pre	8	1650	0.015	13002	Oui
16	124	Éducation	16	36530	AST Intel	8	3190	0.015	25137.2	Oui
17	124	Éducation	17	36530	Compaq Pre	11	1650	0.015	17877.8	Oui
18	106	Constructi	18	36530	AST Intel	11	3190	0.015	34563.6	Oui
19	101	Constructi	19	36530	Compaq Pre	14	1650	0.015	22753.5	Non
20	116	Pharmaceut	20	36531	IBM 500	7	2299	0.015	15851.6	Non
21	112	Machines/O	21	36531	AST Intel	6	2690	0.015	15897.9	Oui
22	125	Constructi	22	36531	Compaq Pre	23	1650	0.03	36811.5	Oui
23	100	Assurances	23	36531	IBM 500	3	2299	0	6897	Oui
24	125	Constructi	24	36531	AST Intel	2	3190	0	6380	Oui
25	104	Éducation	25	36532	AST Intel	12	2690	0.015	31795.8	Oui

Exercice 3.: Installation de l'add-in MS Excel

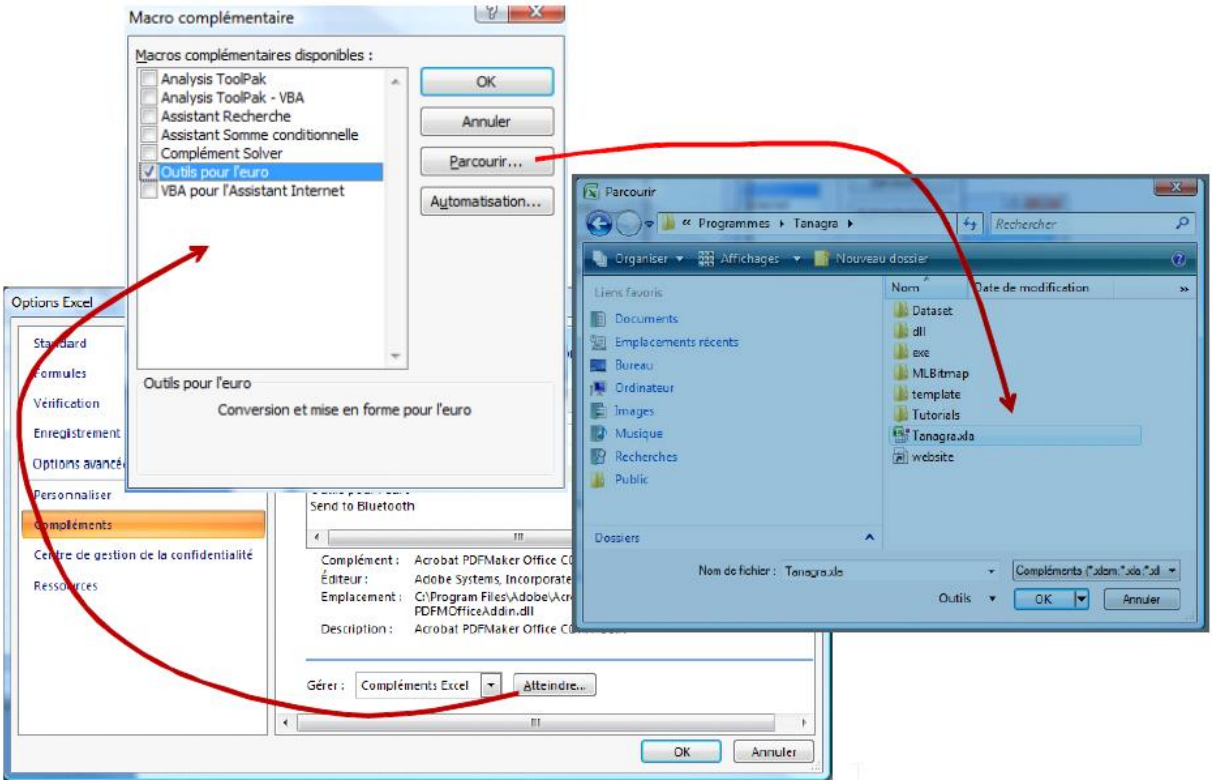
Tanagra V1.4.36

La macro complémentaire (« add-in » en anglais) *tanagra.xla* participe grandement à la diffusion du logiciel Tanagra. Le principe est simple, il s'agit d'intégrer un menu Tanagra dans Excel. Ainsi l'utilisateur peut lancer les calculs statistiques sans avoir à quitter le tableur. Pour simple qu'elle soit, cette fonctionnalité facilite le travail du data miner. Le tableur est un des outils les plus utilisés pour la préparation des données.

Nous ouvrons dans MS Excel 2010 pour aller faire un clic droit sur les rubans et en sélectionnant dans le menu contextuel qui apparaît l'option **Personnaliser la barre d'outils Accès rapide**:



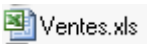
Dans la boîte de dialogue qui apparaît, nous cliquons sur la partie gauche sur **Compléments** et sur la partie droite sur **Atteindre**:



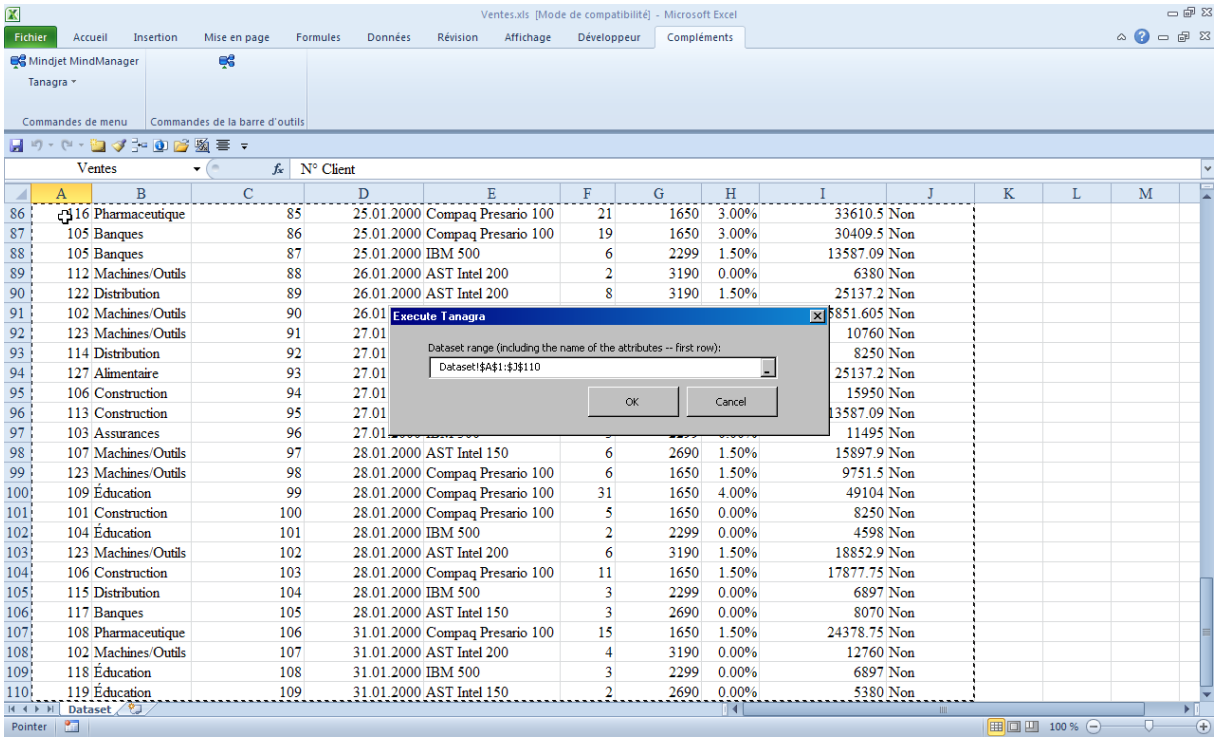
Viens alors la boîte de dialogue des Macros complémentaires. Il faut cliquer sur le bouton **Parcourir** et allez chercher *Tanagra.xla* sur le chemin C:\Programmes\Tangra.

Il faut ensuite valider trois fois par **OK** pour voir l'add-in Tanagra apparaître dans le ruban **Compléments**:

Pour voir comment cet add-in fonctionne, nous ouvrons le fichier:

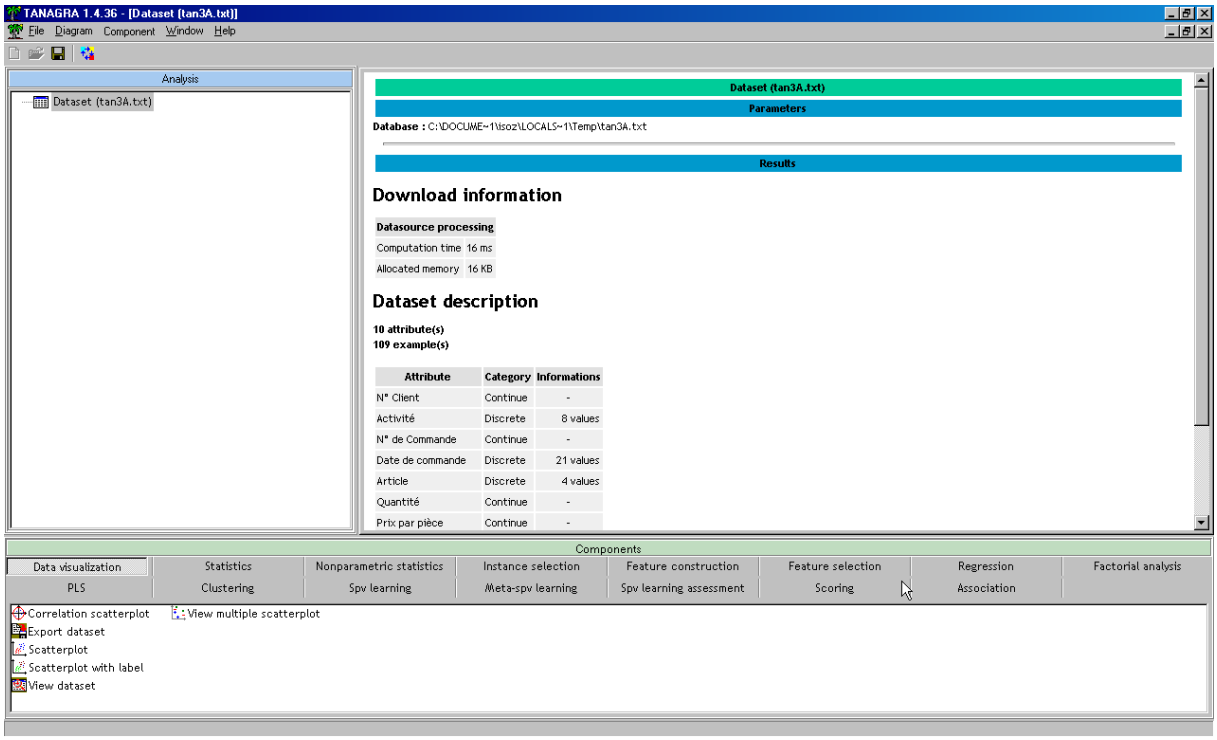


et nous cliquons sur **Execute Tanagra**:



	A	B	C	D	E	F	G	H	I	J
86	116	Pharmaceutique	85	25.01.2000	Compaq Presario 100	21	1650	3.00%	33610.5	Non
87	105	Banques	86	25.01.2000	Compaq Presario 100	19	1650	3.00%	30409.5	Non
88	105	Banques	87	25.01.2000	IBM 500	6	2299	1.50%	13587.09	Non
89	112	Machines/Outils	88	26.01.2000	AST Intel 200	2	3190	0.00%	6380	Non
90	122	Distribution	89	26.01.2000	AST Intel 200	8	3190	1.50%	25137.2	Non
91	102	Machines/Outils	90	26.01					5851.605	Non
92	123	Machines/Outils	91	27.01					10760	Non
93	114	Distribution	92	27.01					8250	Non
94	127	Alimentaire	93	27.01					25137.2	Non
95	106	Construction	94	27.01					15950	Non
96	113	Construction	95	27.01					13587.09	Non
97	103	Assurances	96	27.01					11495	Non
98	107	Machines/Outils	97	28.01.2000	AST Intel 150	6	2690	1.50%	15897.9	Non
99	123	Machines/Outils	98	28.01.2000	Compaq Presario 100	6	1650	1.50%	9751.5	Non
100	109	Éducation	99	28.01.2000	Compaq Presario 100	31	1650	4.00%	49104	Non
101	101	Construction	100	28.01.2000	Compaq Presario 100	5	1650	0.00%	8250	Non
102	104	Éducation	101	28.01.2000	IBM 500	2	2299	0.00%	4598	Non
103	123	Machines/Outils	102	28.01.2000	AST Intel 200	6	3190	1.50%	18852.9	Non
104	106	Construction	103	28.01.2000	Compaq Presario 100	11	1650	1.50%	17877.75	Non
105	115	Distribution	104	28.01.2000	IBM 500	3	2299	0.00%	6897	Non
106	117	Banques	105	28.01.2000	AST Intel 150	3	2690	0.00%	8070	Non
107	108	Pharmaceutique	106	31.01.2000	Compaq Presario 100	15	1650	1.50%	24378.75	Non
108	102	Machines/Outils	107	31.01.2000	AST Intel 200	4	3190	0.00%	12760	Non
109	118	Éducation	108	31.01.2000	IBM 500	3	2299	0.00%	6897	Non
110	119	Éducation	109	31.01.2000	AST Intel 150	2	2690	0.00%	5380	Non

et nous sélectionnons la plage du tableau. Nous validons par **OK** ce qui va faire ouvrir Tanagra avec le datamart chargé:



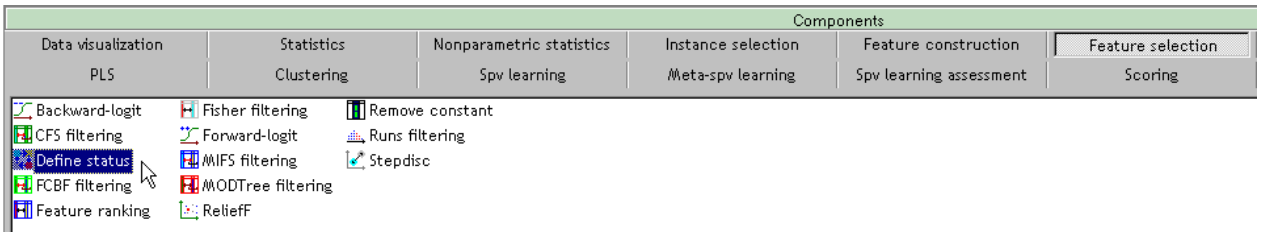
et ensuite y'a plus qu'à…

Exercice 4.: Statistiques élémentaires univariées continues

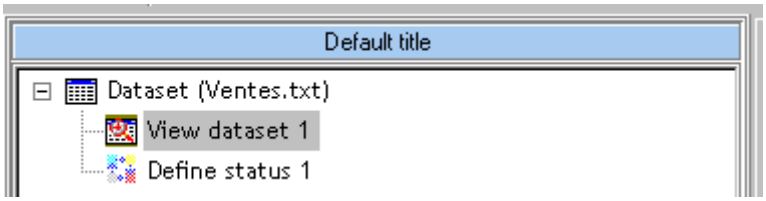
Tanagra V1.4.36

Toujours à partir du même fichier *Ventes.xls* nous souhaiterions générer de petites statistiques univariées continues élémentaires.

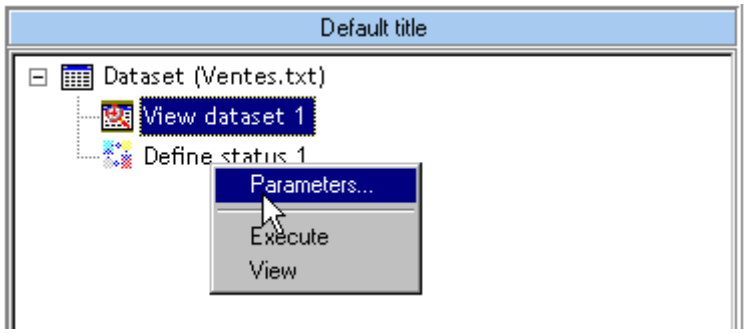
Pour cela nous rajoutons d'abord un sélecteur **Define status** de l'onglet **Feature Selection** ou de la barre de menu du logiciel (cela dépend de la version...):



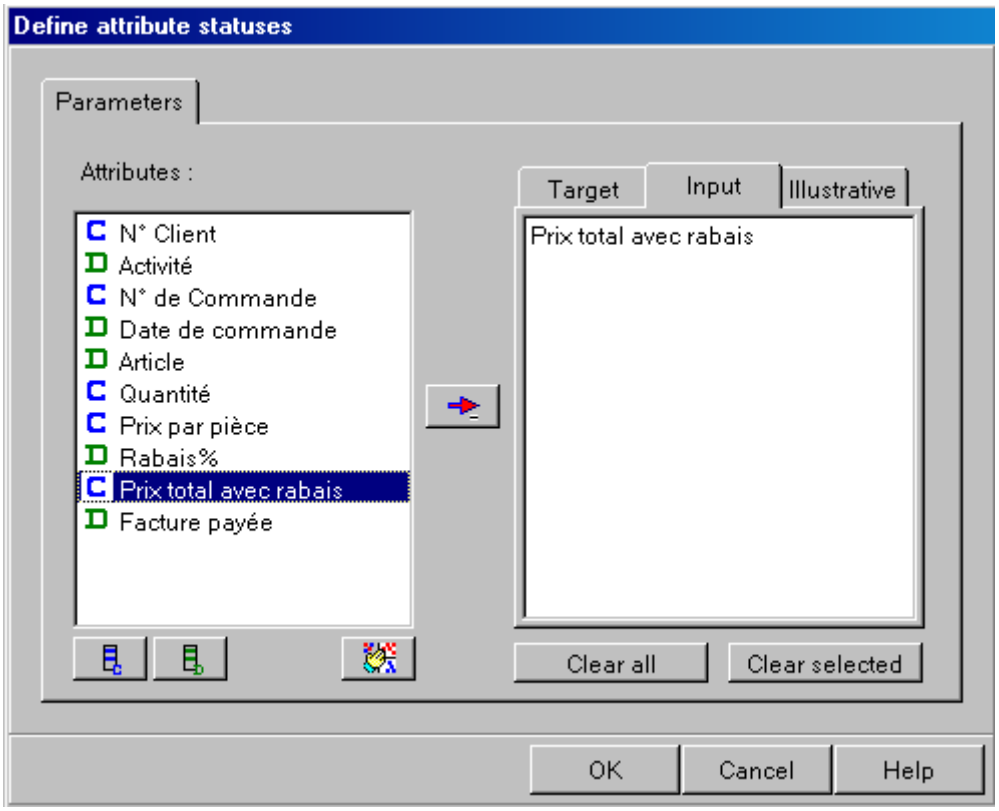
Ce qui nous donne:



Nous faisons un clic droit sur cet sélecteur pour aller dans les paramètres:

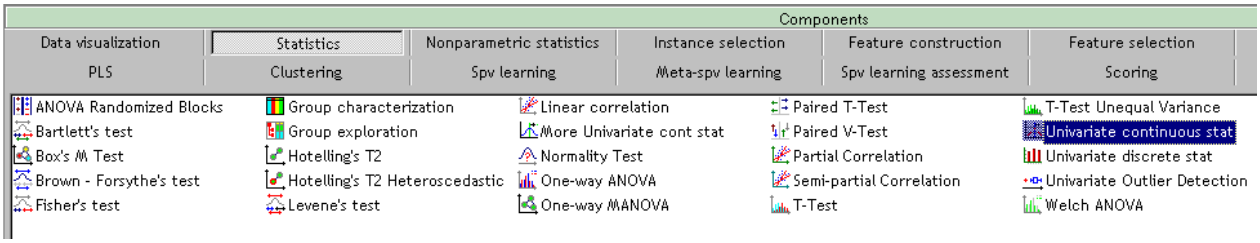


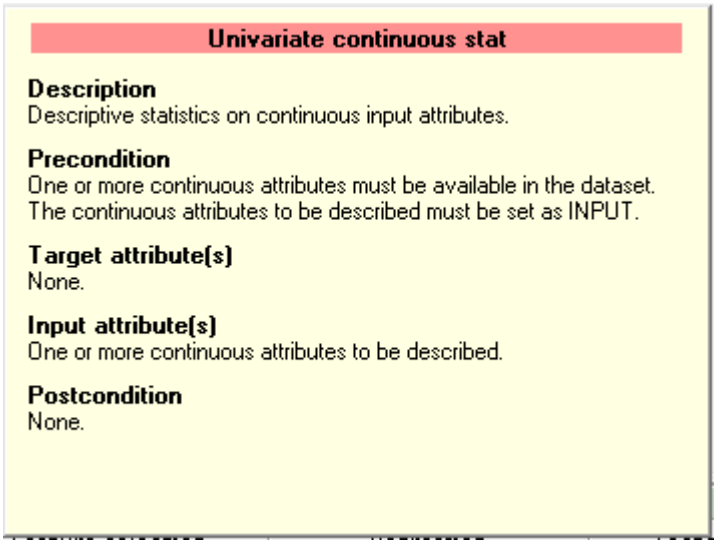
ce qui fait apparaître la boîte de dialogue suivante:



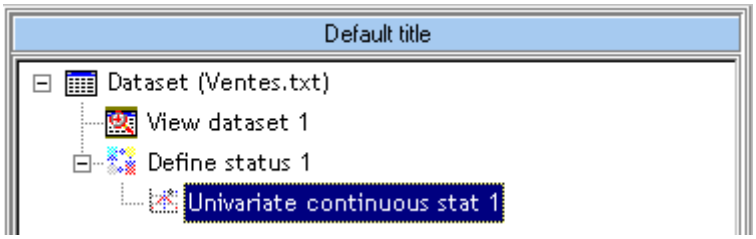
où nous avons sélectionné la variable continue (d'où le **C** en bleu à l'opposé des variables discrètes) et nous validons par **OK**.

Ensuite nous rajoutons un opérateur **Univariate continuous stat** de l'onglet **Statistics**:

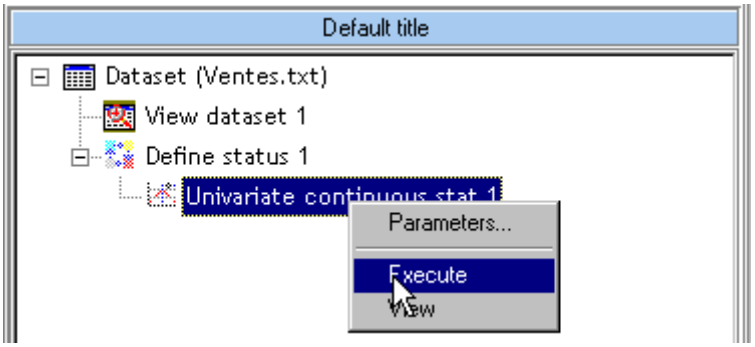




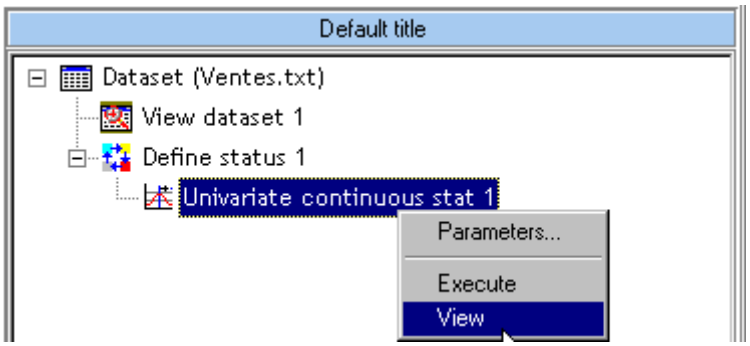
pour avoir:



et nous faisons un clic droit **Execute**:



et ensuite un clic droit **View**:



ce qui nous donne les statistiques élémentaires univariées suivantes:

Univariate continuous stat 1

Parameters

Attributes : 1
Examples : 109

Results

Attribute	Min	Max	Average	Std-dev	Std-dev/avg
Prix total avec rabais	2690	85219.2	18784.5448	15122.2731	0.8050

Computation time : 0 ms.
Created at 29.04.2011 21:43:42

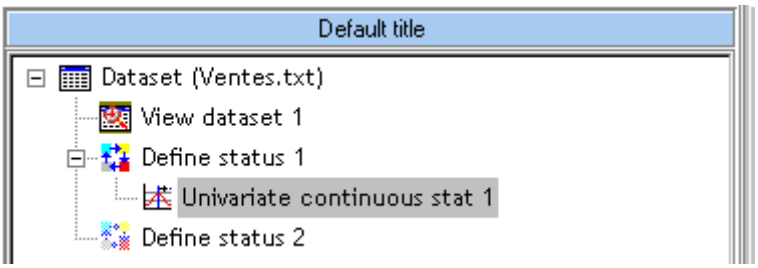
C'est suffisamment simple pour ne nécessiter aucune explication particulière.

Exercice 5.: Statistiques élémentaires univariées discrètes

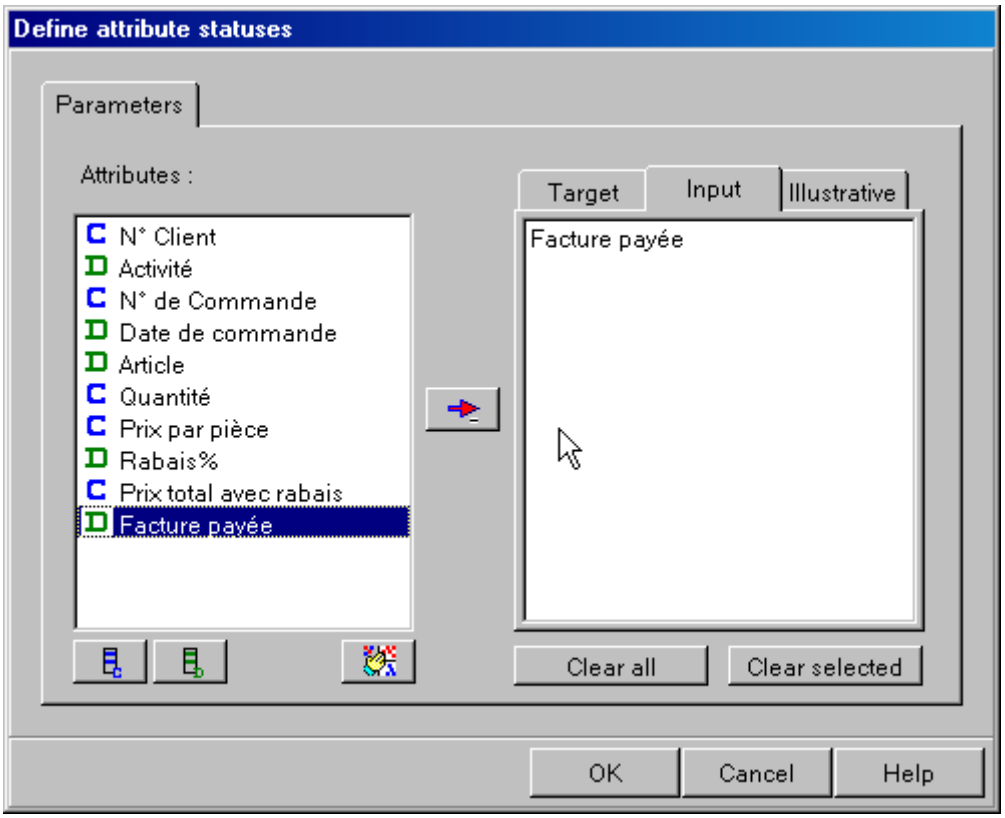
Tanagra V1.4.36

Toujours à partir du même fichier *Ventes.xls* nous souhaiterions générer de petites statistiques univariées continues discrètes.

Pour cela nous rajoutons encore un sélecteur **Define** *status* de l'onglet **Feature Selection** afin d'obtenir:

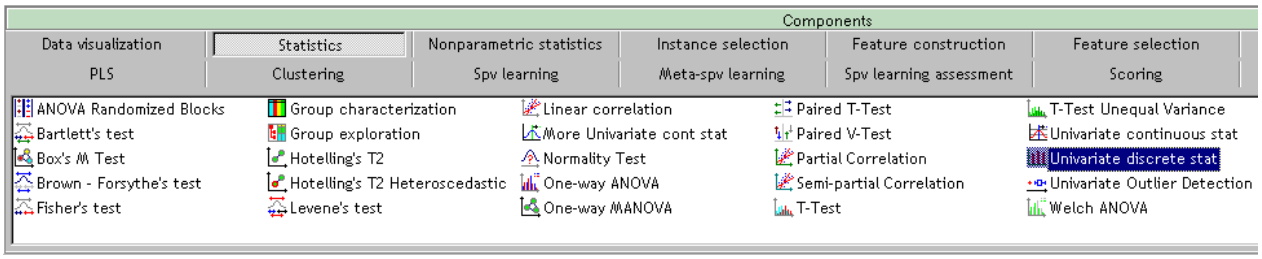


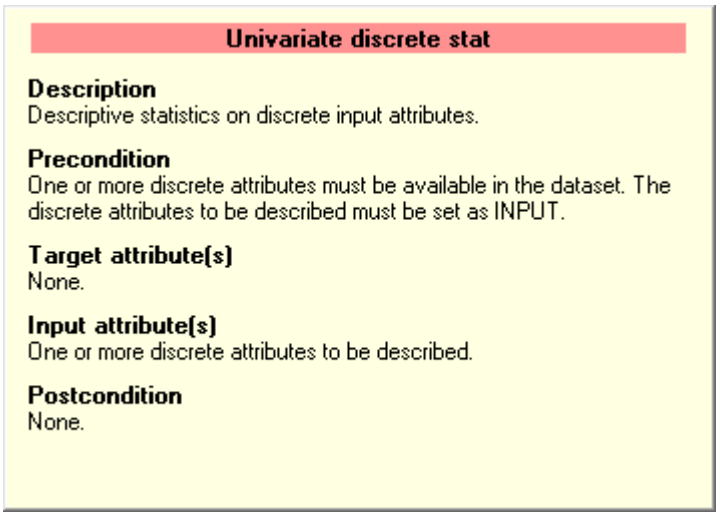
et nous allons dans ses paramètres pour choisir la variable discrète *Facture Payée*:



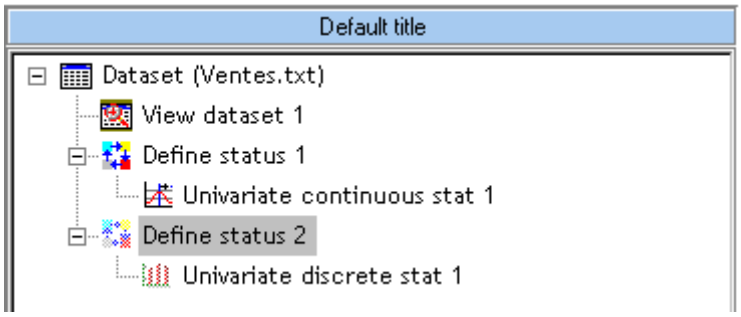
Nous validons par **OK**.

Ensuite nous rajoutons un opérateur **Univariate discrete stat** de l'onglet **Statistics**:





Pour avoir:



et nous procédons comme avant en exécutant et en affichant les données:

Univariate discrete stat 1

Parameters

Attributes : 1
Examples : 109

Results

Attribute	Gini	Distribution			
		Values	Count	Percent	Histogram
Facture payée	0.4929	Oui	61	55.96 %	
		Non	48	44.04 %	

Computation time : 0 ms.
Created at 29.04.2011 21:55:16

C'est suffisamment simple aussi pour ne nécessiter aucune explication particulière.

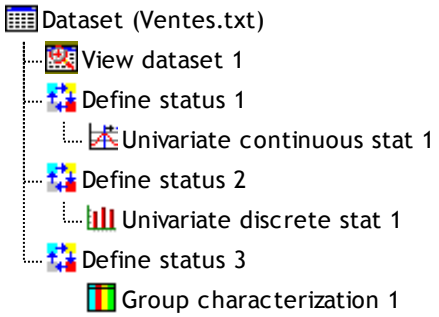
Concernant l'indice de Gini, nous avons déjà étudié comme calculer ce dernier dans le cours de statistiques théorique sur plusieurs pages (que je ne souhaite pas reproduire ici).

Exercice 6.: Statistiques univariées continues multiples

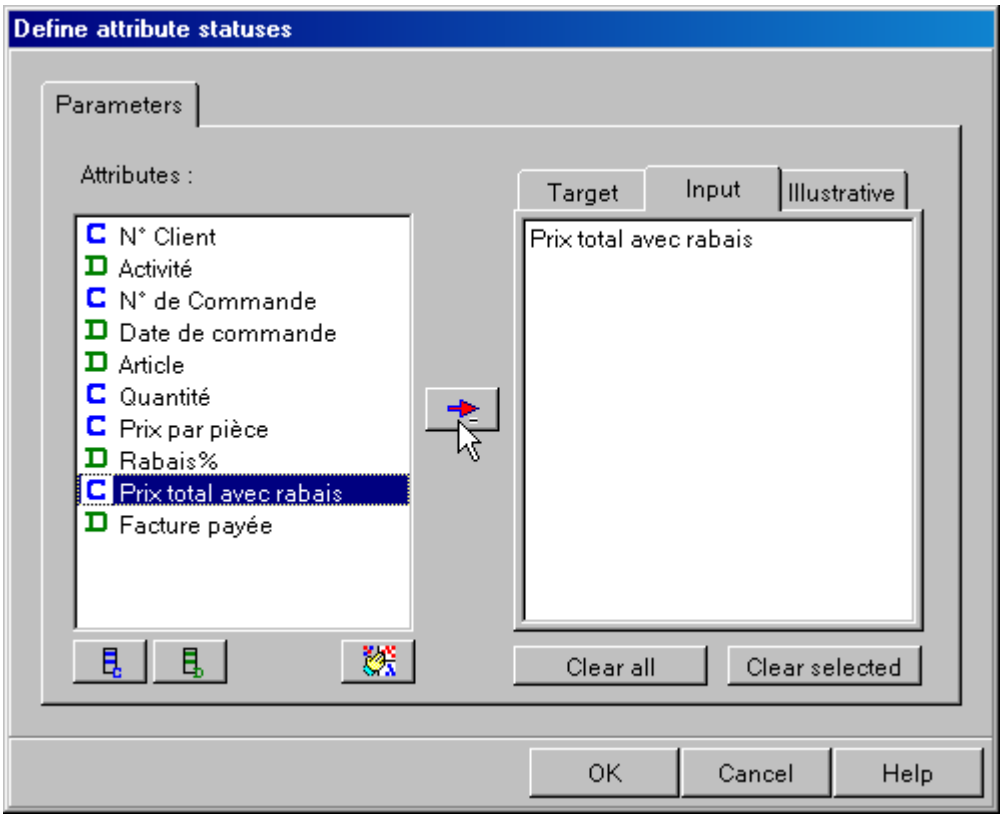
Tanagra V1.4.36

Avoir un peu plus d'indicateurs statistiques concernant la colonne *Prix total avec rabais* de notre fichier *Ventes.txt*.

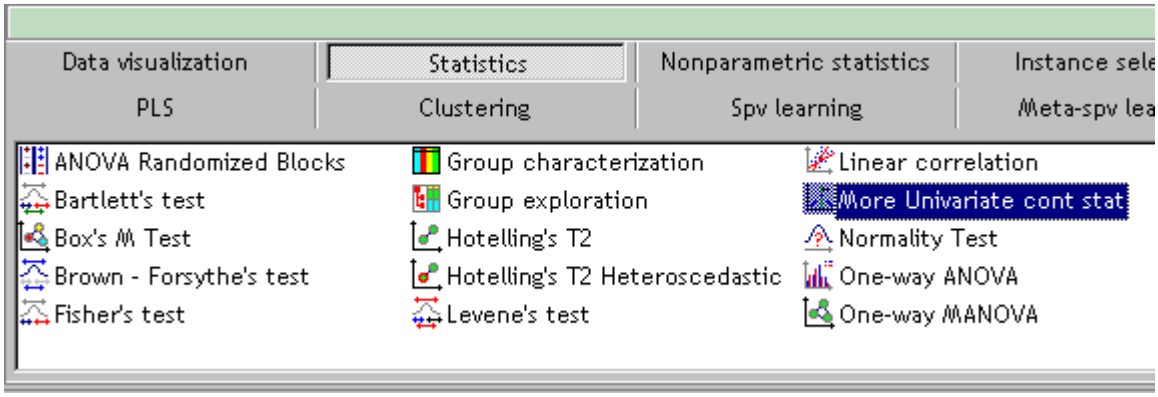
Nous repartons de la configuration suivante:

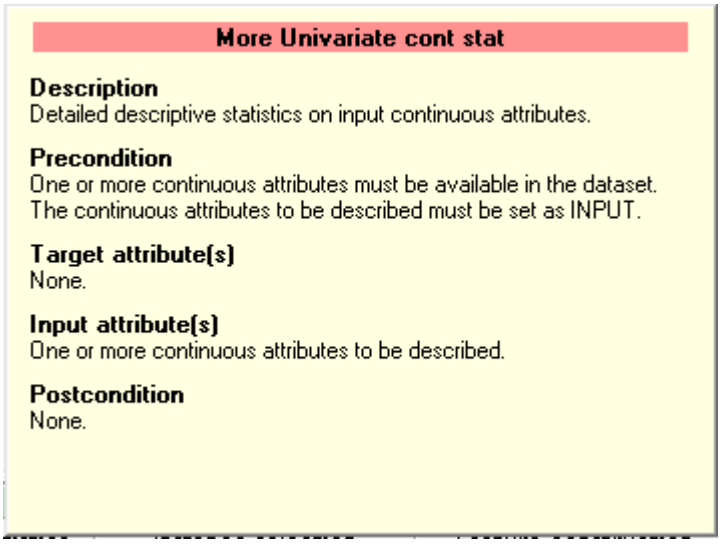


et nous allons ajouter *Prix total avec rabais* dans l'**Input**:



Pour y ajouter un autre sélecteur **Define status** et nous ajoutons l'opérateur **More Univariate cont stat** de l'onglet **Statistics**:





Nous exécutons et affichons cet opérateur pour obtenir:

More Univariate cont stat 1

Parameters

Attributes : 1
Examples : 109

Results

Attribute	Stats		Histogram			
	Statistics		Values	Count	Percent	Histogram
Prix total avec rabais	Average	18784.5448	x_<_10942.9203	37	33.94%	
	Median	13587.0898	10942.9203_=<_x_<_19195.8406	34	31.19%	
	Std dev. [Coef of variation]	15122.2731 [0.8050]	19195.8406_=<_x_<_27448.7609	21	19.27%	
	MAD [MAD/STDDEV]	10310.8494 [0.6818]	27448.7609_=<_x_<_35701.6813	8	7.34%	
	Min * Max [Full range]	2690.00 * 85219.20 [82529.20]	35701.6813_=<_x_<_43954.6016	2	1.83%	
	1st * 3rd quartile [Range]	9196.00 * 22753.50 [13557.50]	43954.6016_=<_x_<_52207.5219	1	0.92%	
	Skewness (std-dev)	2.2035 (0.2315)	52207.5219_=<_x_<_60460.4422	1	0.92%	
	Kurtosis (std-dev)	5.6165 (0.4590)	60460.4422_=<_x_<_68713.3625	3	2.75%	
			68713.3625_=<_x_<_76966.2828	1	0.92%	
			x>=_76966.2828	1	0.92%	

Computation time : 0 ms.
Created at 01.05.2011 13:23:24

où MAD est la *Median Absolute Deviation* défini par:

$$MAD = Median\left(\left|X_i - Median\left(X\right)\right|\right)$$

donc il s'agit de la médiane des écarts absolus à la médiane de la variable aléatoire *X*.

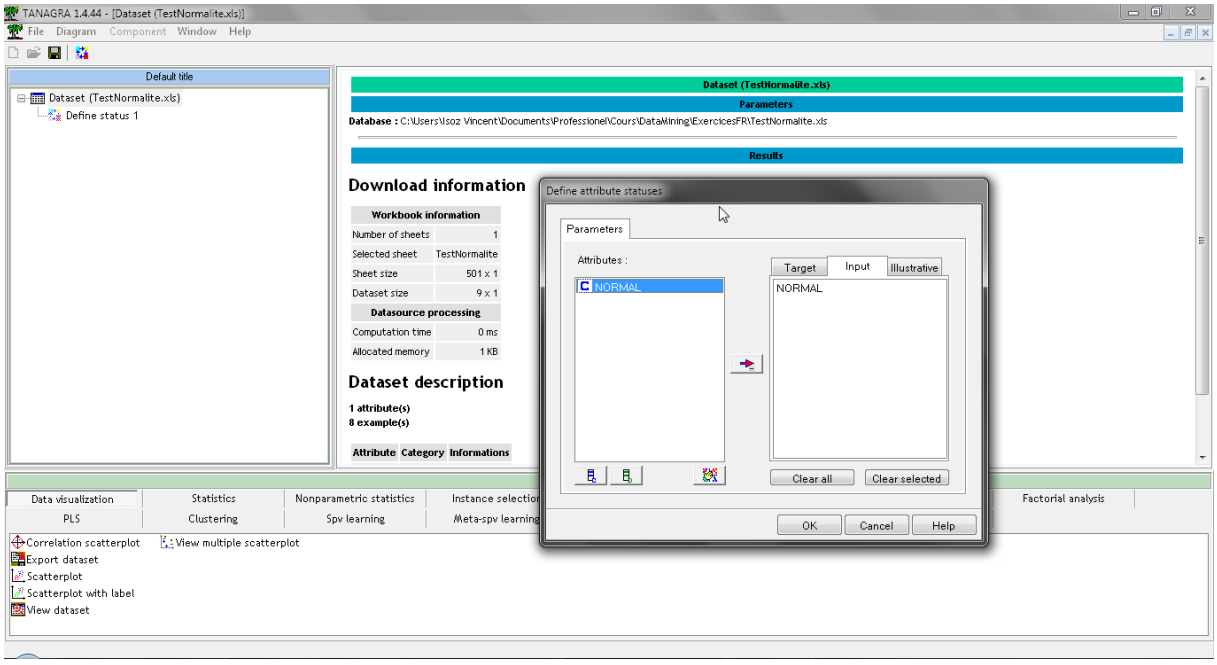
Exercice 7.: Test de Normalité

Tanagra V1.4.36

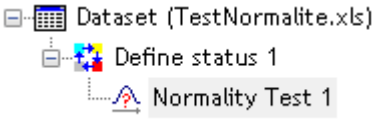
Pour le test de normalité (cas particulier d'application des tests de Shapiro-Wilk et d'Anderson-Darling démontré en cours), nous allons utiliser un échantillon de données différents car Tanagra, au même titre que certains autres logiciels de statistiques, refuse d'exécuter la statistique lorsqu'il y a moins de 8 individus. Nous allons donc nous baser sur l'échantillon suivant et le lecteur pourra vérifier si cela correspond bien évidemment par lui-même si cela correspond aux calculs faits à la main pendant le cours théorique (ce qui bien évidemment est le cas!):

	A
1	NORMAL
2	0.254740371
3	-1.225673714
4	-1.282637641
5	-0.142906629
6	0.531747446
7	1.649268597
8	0.296265625
9	0.596445489

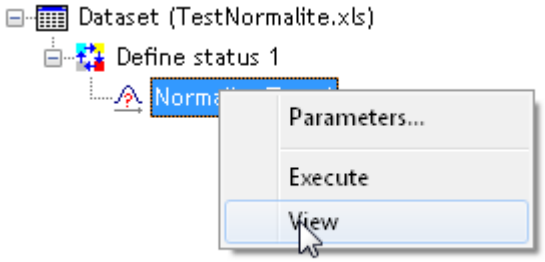
Nous souhaitons donc comparer si ces données suivent une loi Normale d'espérance et écart-type estimé sur l'échantillon. Pour cela, nous chargeons bien évidemment le fichier *.xls comme déjà vu plusieurs fois plus haut et nous avons alors:



Et nous prenons dans l'opérateur **Normality Test** de l'onglet **Statistics**:



Nous faisons un clic droit **View**:



pour avoir au final:

Normality Test 1

Parameters

Attributes : 1
Examples : 8

Results

Attribute	Mu ; Sigma	Shapiro-Wilk (p-value)	Lilliefors D = max[D-,D+] (p-value)	Anderson-Darling (p-value)	d'Agostino (p-value)
NORMAL	0,0847 ; 0,9726	0,921941 (0,4458)	0,1944 = max [0,1944,0,1744] (p >= 0.20)	0,367186 (p >= 0.10)	-0,2097 ^ 2 + 0,1072 ^ 2 = 0,0555 (0,9726)

Computation time : 0 ms.
Created at 25/10/2012 14:43:37

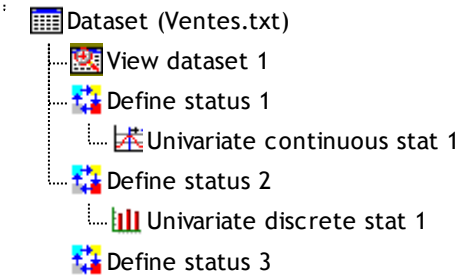
et donc outre le test d'Agostino et de Lilliefors que nous n'avons pas démontré en cours, nous retrouvons bien les valeurs pour les tests de Shapiro-Wilk ou d'Anderson-Darling.

Exercice 8.: Caractérisation de groupes

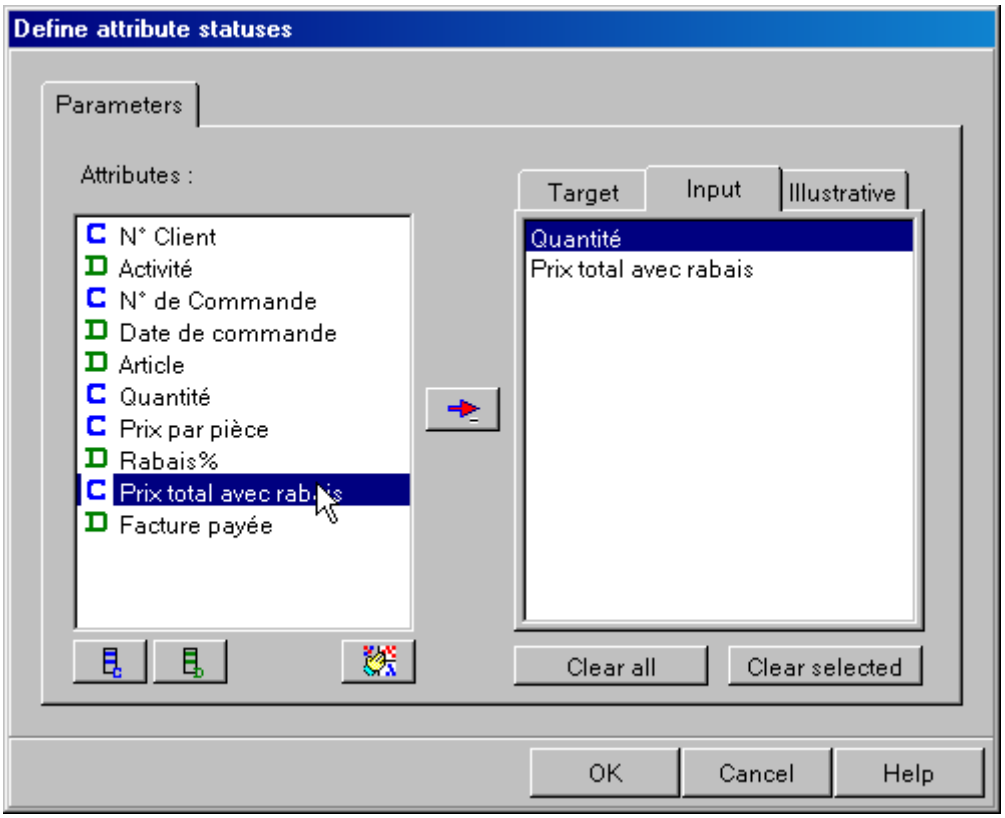
Tanagra V1.4.36

Toujours à partir du même fichier *Ventes.xls* nous souhaiterions caractériser la population à partir de l'état des factures payées.

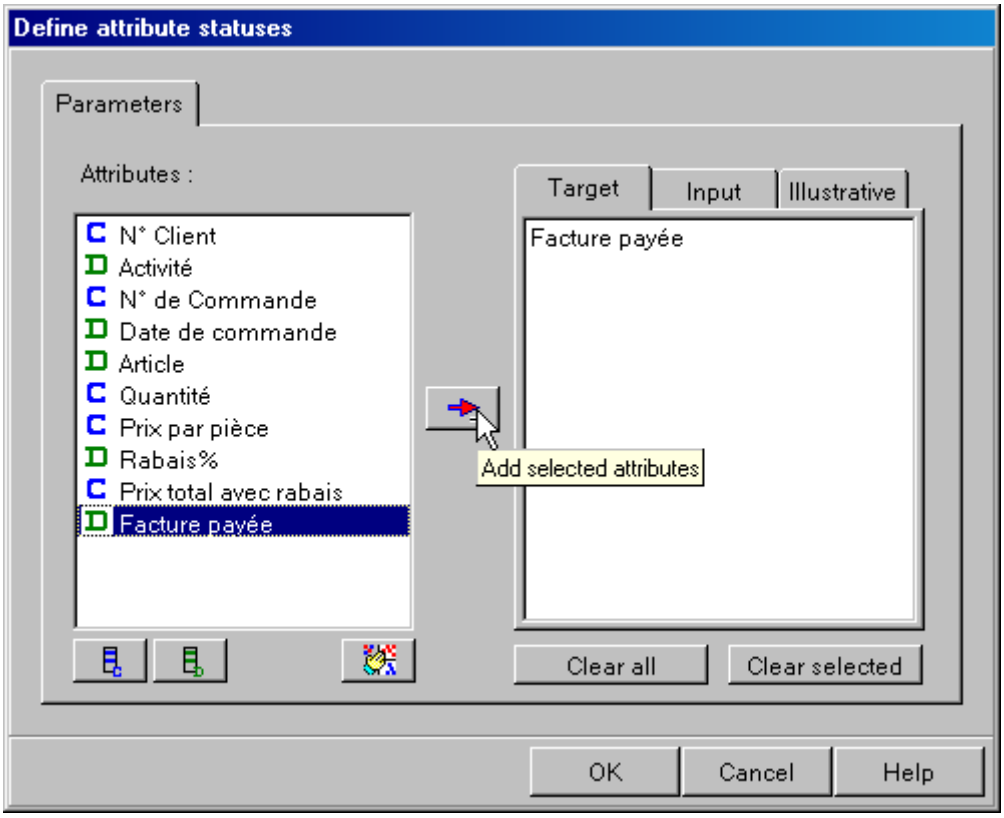
Pour cela nous rajoutons encore un sélecteur **Define status** de l'onglet **Feature Selection** afin d'obtenir:



et nous allons mettre des variables **Input** en entrée qui nous sembleraient être subjectivement facteurs d'influence des factures payées ou non:

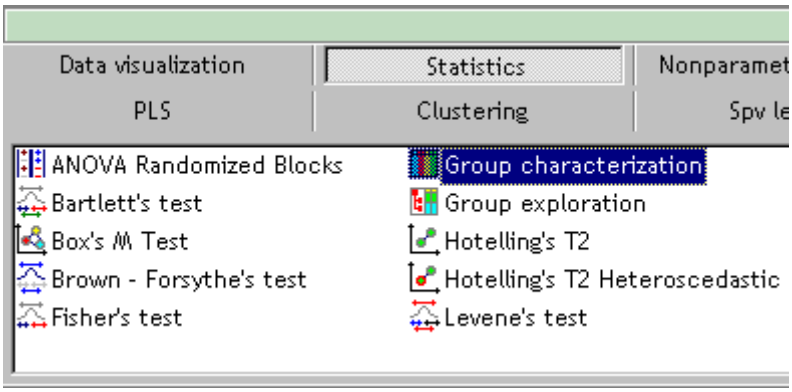


et la variable discrète qui nous intéresse en **Target**:



Nous validons par **OK**.

Nous y ajoutons un opérateur du type **Group characterization** depuis l'onglet **Statistics**:



Group characterization

Description
Comparative descriptive statistics in order to characterize groups defined by discrete attributes. The aim of this component is to show if there are differences between groups according to the various statistical indicators such as average, proportion, etc. The groups are defined by the discrete TARGET attributes. The descriptives statistics are computed on discrete or continuous INPUT variables. This component can be used for instance in order to depict groups computed by a clustering algorithm.

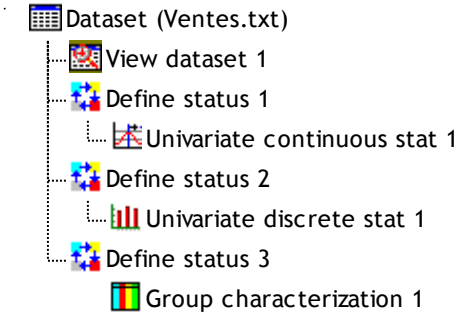
Precondition
The TARGET and INPUT attributes must be specified.

Target attribute(s)
One or more (if you want characterize several groups) discrete attributes.

Input attribute(s)
One or more continuous and/or discrete attributes.

Postcondition
none.

pour avoir:



et enfin nous exécutons cet opérteur et affichons les résultats comme précédemment:

Group characterization 1

Parameters

Normalization : 0

Results

Description of "Facture payée"

Facture payée=Oui			
Examples		[56.0 %] 61	
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Prix total avec rabais	-1.73	16545.60 (11350.26)	18784.54 (15122.27)
Quantité	-1.91	7.08 (4.98)	8.11 (6.31)
Discrete attributes : [Recall] Accuracy			

Facture payée=Non			
Examples		[44.0 %] 48	
Att - Desc	Test value	Group	Overral
Continuous attributes : Mean (StdDev)			
Quantité	1.91	9.42 (7.54)	8.11 (6.31)
Prix total avec rabais	1.73	21629.87 (18607.76)	18784.54 (15122.27)
Discrete attributes : [Recall] Accuracy			

Computation time : 0 ms.
Created at 29.04.2011 22:14:18

Nous observons donc qu'en moyenne, le prix total avec rabais est moins élévé (16'546.60) pour ceux qui payent les factures que pour ceux qui ne les paient pas. Il en est de même pour les quantités. Le résultat est peu surprenant…

Concernant la *Test value* (TV), il s'agit d'un indicateur permettant de comparer pour une variable continue la moyenne et pour une variable discrète la proportion.

Dans le cas d'une variable continue cette valeur provient simplement d'un test Z de la moyenne:

$$Z = \frac{\mu_{overall} - \mu}{\frac{\sigma_{overall}}{\sqrt{n}}}$$

Mais avec le facteur de correctiond la population démontré en cours, ce qui fait que la dernière relation devient:

$$Z = \frac{\mu_{overall} - \mu_{groupe}}{fcp \cdot \frac{\sigma_{overall}}{\sqrt{n}}} = \frac{\mu_{overall} - \mu_{groupe}}{\sqrt{\frac{N-n}{N-1}} \frac{\sigma_{overall}}{\sqrt{n}}}$$

Dans le cas d'un variable discrète, le test se fait sur la base des proportions vues aussi dans le cours théorique:

$$Z = \frac{\hat{p} - p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}}$$

et encore une fois avec le facteur de correction:

$$Z = \frac{\hat{p} - p}{\sqrt{\frac{N-n}{N-1}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}}$$

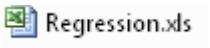
Si au lieu de travailler avec les proportions, nous voulons travailler avec le comptage, un simple transformation nous amène à:

$$Z = \frac{\hat{p} - p}{\sqrt{\frac{N-n}{N-1}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}}$$

Exercice 9.: Régression linéaire simple ou multiple

Tanagra V1.4.38

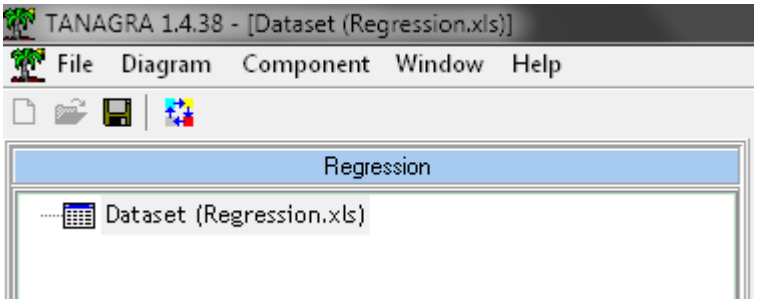
Nous allons prendre ce fichier qui MS Excel qui nous est connu mais qu'il a fallu restructurer pour Tanagra (voir cours sur MS Excel):



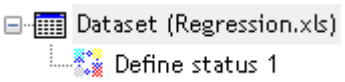
contenant:

	A	B	C	D	E
1	Mois	Coûts	Coût de A	Coût de B	Coût de C
2	1	44439	515	541	928
3	2	43936	929	692	711
4	3	44464	800	710	824
5	4	41533	979	675	758
6	5	46343	1165	1147	635
7	6	44922	651	939	901
8	7	43203	847	755	580
9	8	43000	942	908	589
10	9	40967	630	738	682
11	10	48582	1113	1175	1050
12	11	45003	1086	1075	984
13	12	44303	843	640	828
14	13	42070	500	752	708
15	14	44353	813	989	804
16	15	45968	1190	823	904
17	16	47781	1200	1108	1120
18	17	43202	731	590	1065
19	18	44074	1089	607	1132
20	19	44610	786	513	839

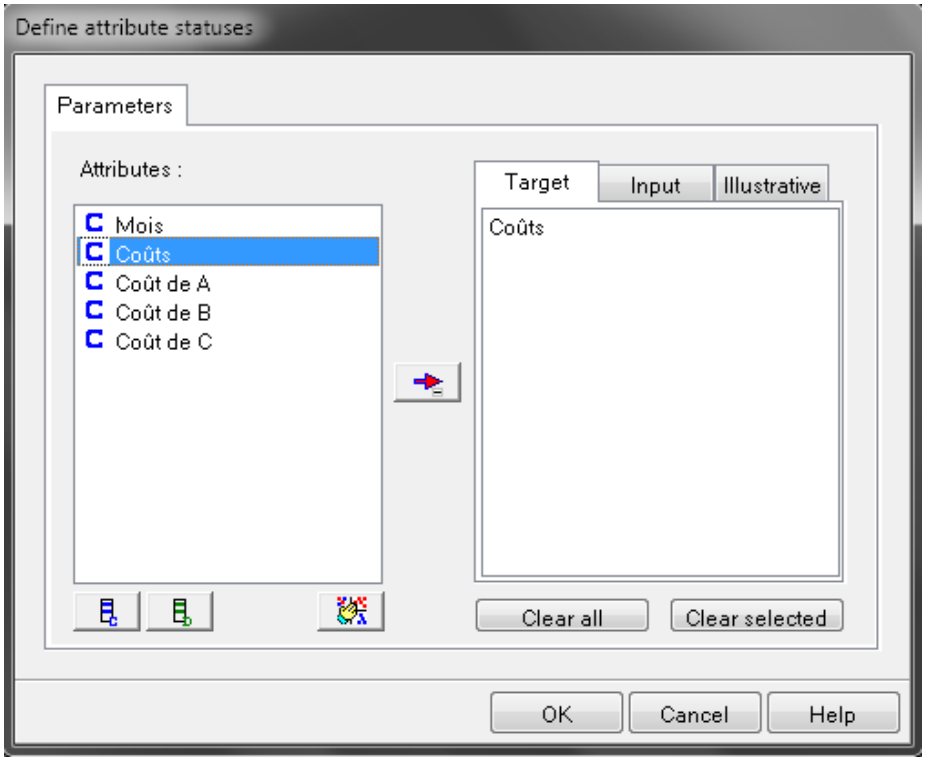
Nous l'importons dans Tanagra en utilisant la même procédure que les exercices précédents:



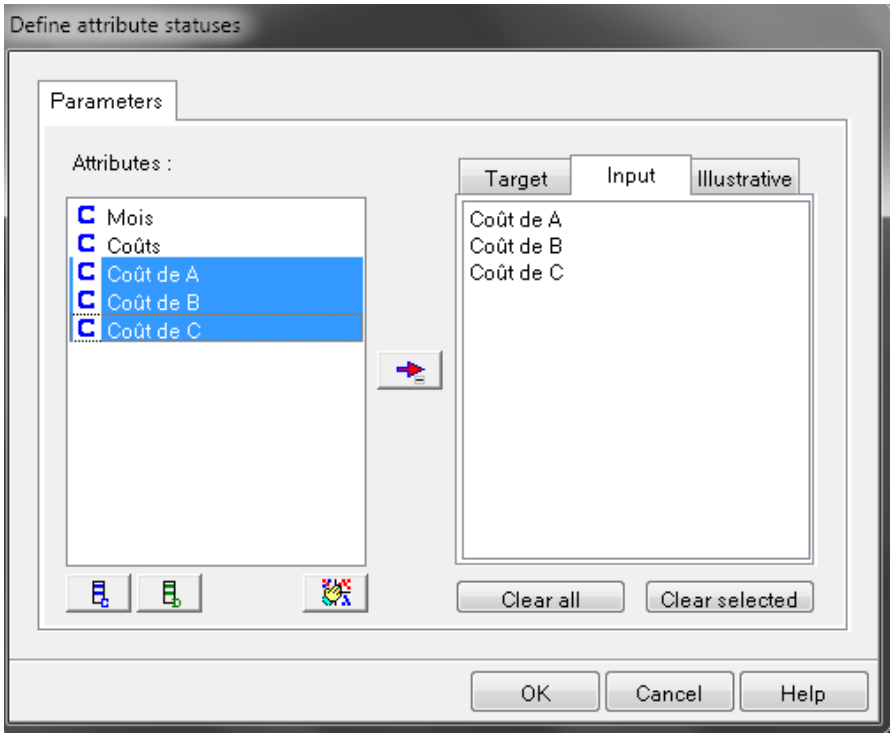
Nous y ajoutons un sélecteur de type **Define status** comme pour les exemples précédents:



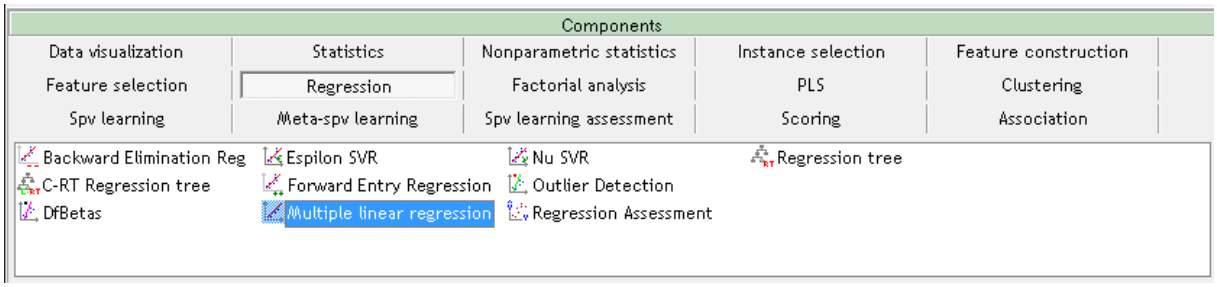
mais avec la variable d'intérêt dans **Target**:



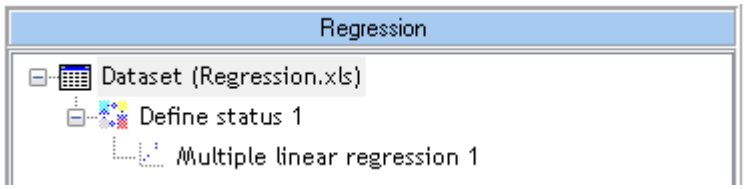
et dans les **Input**:

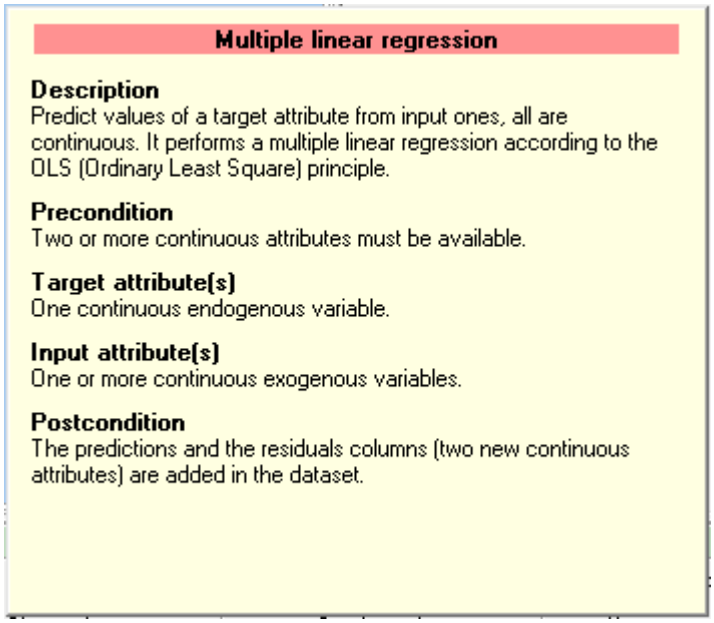


Ajoutons ensuite l'opérateur **Multiple linear regression**:

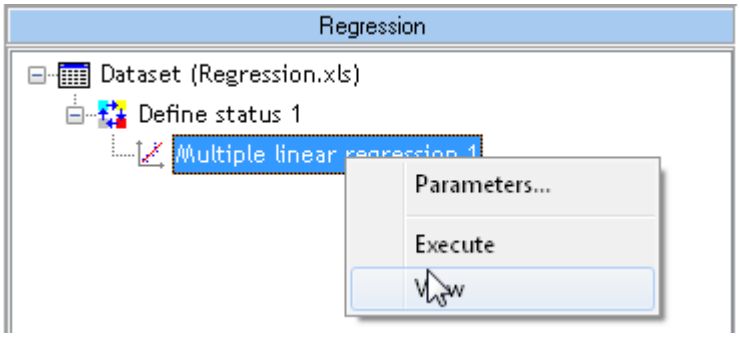


sous le **Define status 1**:





Nous lançons la régression en cliquant sur **View**:



Pour obtenir:

Multiple linear regression 1

Parameters

Regression parameters

Include intercept	yes

Results

Global results

Endogenous attribute	**Coûts**
Examples	19
R^2	0.645450
Adjusted-R^2	0.574539
Sigma error	1252.763898
F-Test (3,15)	9.1024 (0.001127)

Analysis of variance

Source	xSS	d.f.	xMS	F	p-value
Regression	42856229.8868	3	14285409.9623	9.1024	0.0011
Residual	23541260.7448	15	1569417.3830		
Total	66397490.6316	18			

Coefficients

Attribute	Coef.	std	t(15)	p-value
Intercept	35102.900449	1837.226911	19.106459	0.000000
Coût de A	2.065953	1.664982	1.240826	0.233727
Coût de B	4.176356	1.681253	2.484074	0.025288
Coût de C	4.790641	1.789316	2.677359	0.017223

Residuals analysis

Att. name: Err_Pred_lmreg_1

Full statistics

Statistics	
Average	0.0000
Median	236.1985
Std dev. [Coef of variation]	1143.6117 [-1424007051.9366]
MAD [MAD/STDDEV]	975.0982 [0.8526]
Min * Max [Full range]	-2042.81 * 1721.44 [3764.26]
1st * 3rd quartile [Range]	-977.19 * 818.98 [1796.18]
Skewness (std-dev)	-0.3246 (0.5238)
Kurtosis (std-dev)	-1.0173 (1.0143)

Histogram

Values	Count	Percent	Histogram
x_<_-1666.3889	2	10.53%	
-1666.3889_=<_x_<_-1289.9633	1	5.26%	
-1289.9633_=<_x_<_-913.5376	2	10.53%	
-913.5376_=<_x_<_-537.1119	2	10.53%	
-537.1119_=<_x_<_-160.6863	1	5.26%	
-160.6863_=<_x_<_215.7394	1	5.26%	
215.7394_=<_x_<_592.1650	2	10.53%	
592.1650_=<_x_<_968.5907	4	21.05%	
968.5907_=<_x_<_1345.0164	2	10.53%	
x>=_1345.0164	2	10.53%	

Nous obtenons donc toutes les valeurs vues dans le cours théorique.

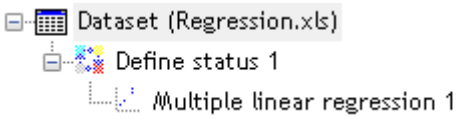
Exercice 10.: Test de Normalité des résidus de la régression linéaire

Tanagra V1.4.48

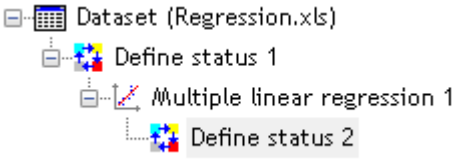
Nous allons reprendre les mêmes données que l'exemple sur la régression linéaire simple ou multiple précédemment:

	A	B	C	D	E
1	Mois	Coûts	Coût de A	Coût de B	Coût de C
2	1	44439	515	541	928
3	2	43936	929	692	711
4	3	44464	800	710	824
5	4	41533	979	675	758
6	5	46343	1165	1147	635
7	6	44922	651	939	901
8	7	43203	847	755	580
9	8	43000	942	908	589
10	9	40967	630	738	682
11	10	48582	1113	1175	1050
12	11	45003	1086	1075	984
13	12	44303	843	640	828
14	13	42070	500	752	708
15	14	44353	813	989	804
16	15	45968	1190	823	904
17	16	47781	1200	1108	1120
18	17	43202	731	590	1065
19	18	44074	1089	607	1132
20	19	44610	786	513	839

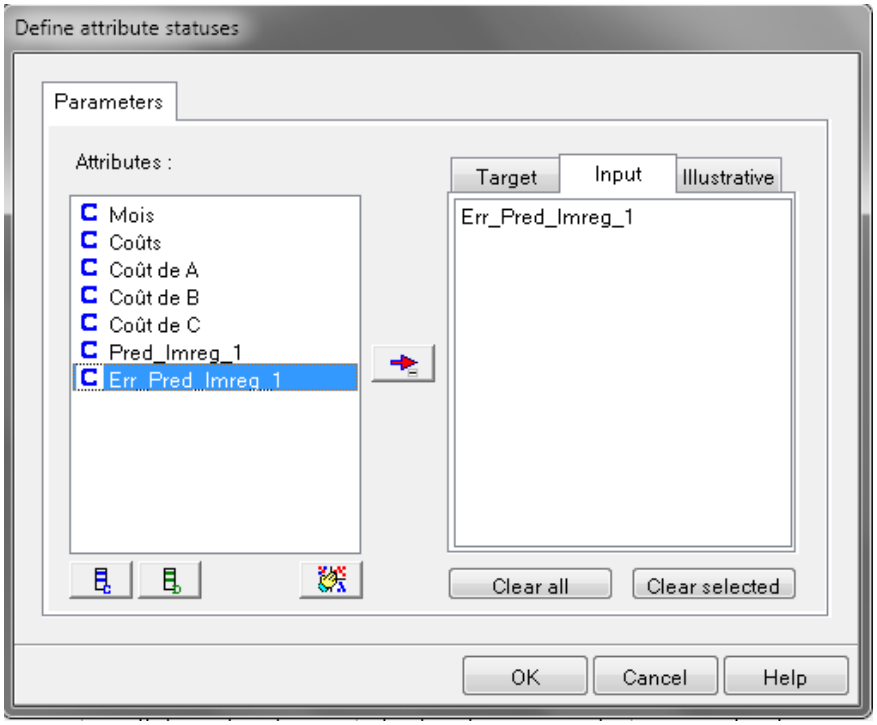
en laissant l'opérateur mis précédemment:



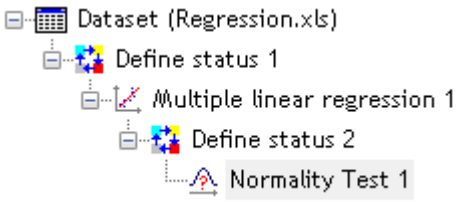
Mais nous allons rajout le sélecteur **Define Status**:



Et dans les paramètres, nous allons mettre en tant que *Input* la variable créée par le composant **Multiple linear regression** et qui est **Err_Pred_Imreg_1**:



Et nous ajoutons l'opérateur **Normality Test** du groupe **Statistics**:



et nous obtenons en l'exécutant:

Normality Test 1

Parameters

Attributes : 1
Examples : 19

Results

Attribute	Mu ; Sigma	Shapiro-Wilk (p-value)	Lilliefors D = max[D-,D+] (p-value)	Anderson-Darling (p-value)	d'Agostino (p-value)
Err_Pred_lmreg_1	0,0000 ; 1143,6117	0,951817 *(0,4241)*	0,1504 = max [0,1504,0,0708] *(p >= 0.20)*	0,343593 *(p >= 0.10)*	-0,6530 ^ 2 + -1,2458 ^ 2 = 1,9783 *(0,3719)*

Nous ne rejettons donc pas l'hypothèse nulle comme quoi les résidus sont normalement distribués.

Exercice 11.: Régression linéaire ascendante (Forward Entry Regression)

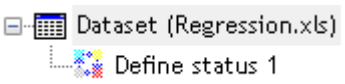
Tanagra V1.4.38

Nous allons reprendre les mêmes données que l'exemple sur la régression linéaire simple ou multiple précédemment:

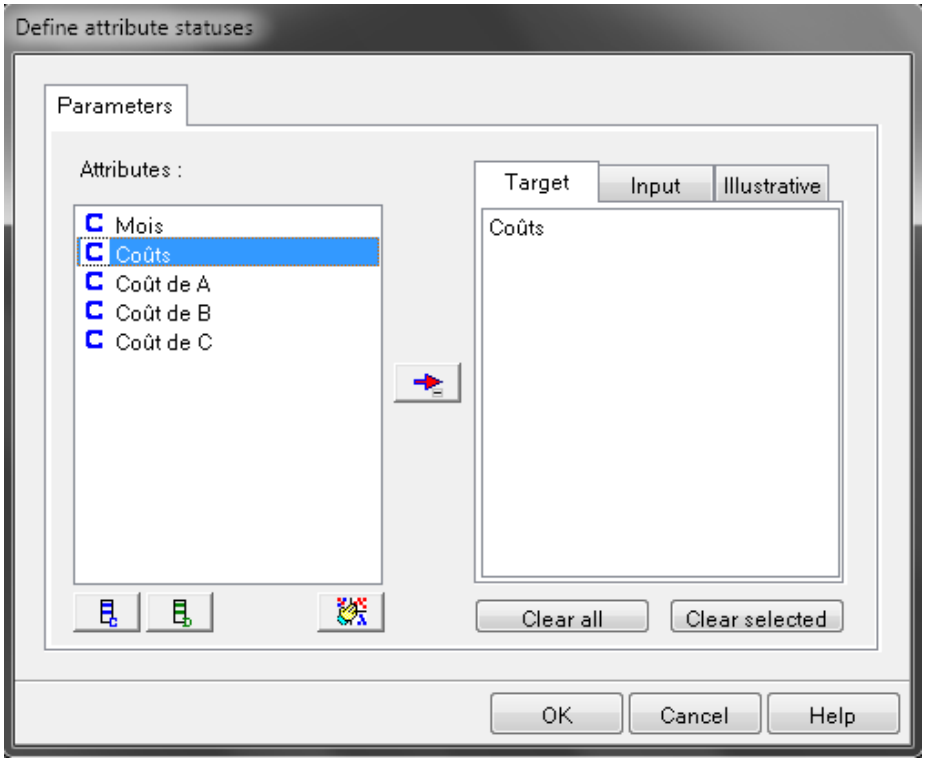
	A	B	C	D	E
1	Mois	Coûts	Coût de A	Coût de B	Coût de C
2	1	44439	515	541	928
3	2	43936	929	692	711
4	3	44464	800	710	824
5	4	41533	979	675	758
6	5	46343	1165	1147	635
7	6	44922	651	939	901
8	7	43203	847	755	580
9	8	43000	942	908	589
10	9	40967	630	738	682
11	10	48582	1113	1175	1050
12	11	45003	1086	1075	984
13	12	44303	843	640	828
14	13	42070	500	752	708
15	14	44353	813	989	804
16	15	45968	1190	823	904
17	16	47781	1200	1108	1120
18	17	43202	731	590	1065
19	18	44074	1089	607	1132
20	19	44610	786	513	839

pour effectuer une régression linéaire ascendante (Forward Entry Selection) et comparer les résultats par rapport à ceux obtenus à la mains dans MS Excel et ceux obtenus aussi dans Minitab 15 dans le cours théorique.

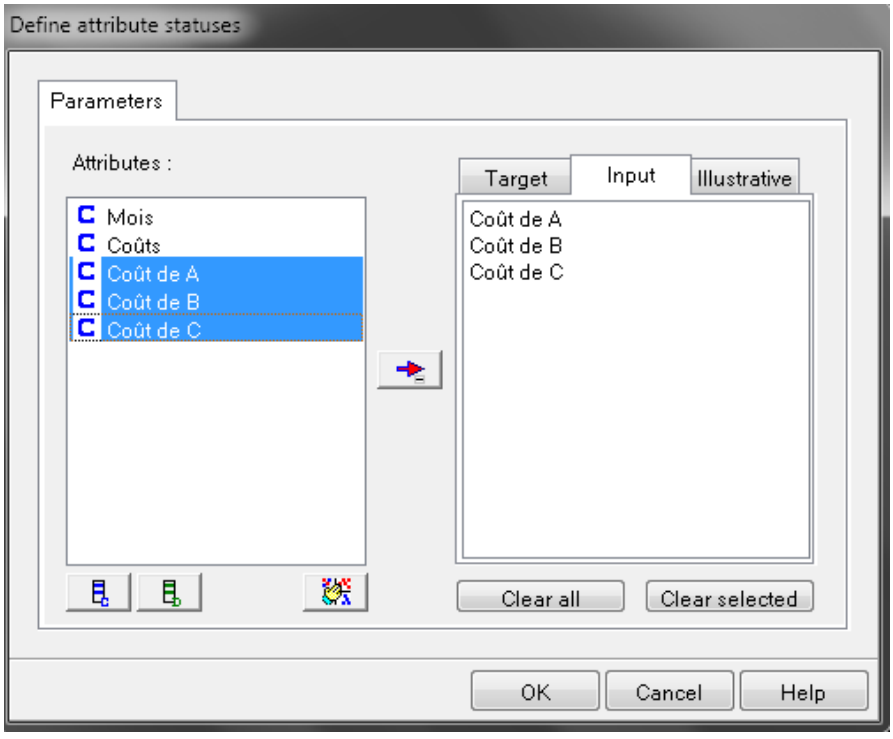
Nous y ajoutons un sélecteur de type **Define status** comme pour les exemples précédents:



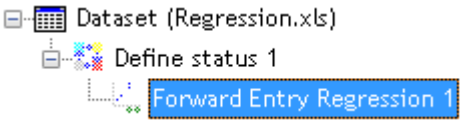
avec la variable d'intérêt dans **Target**:



et dans les **Input**:

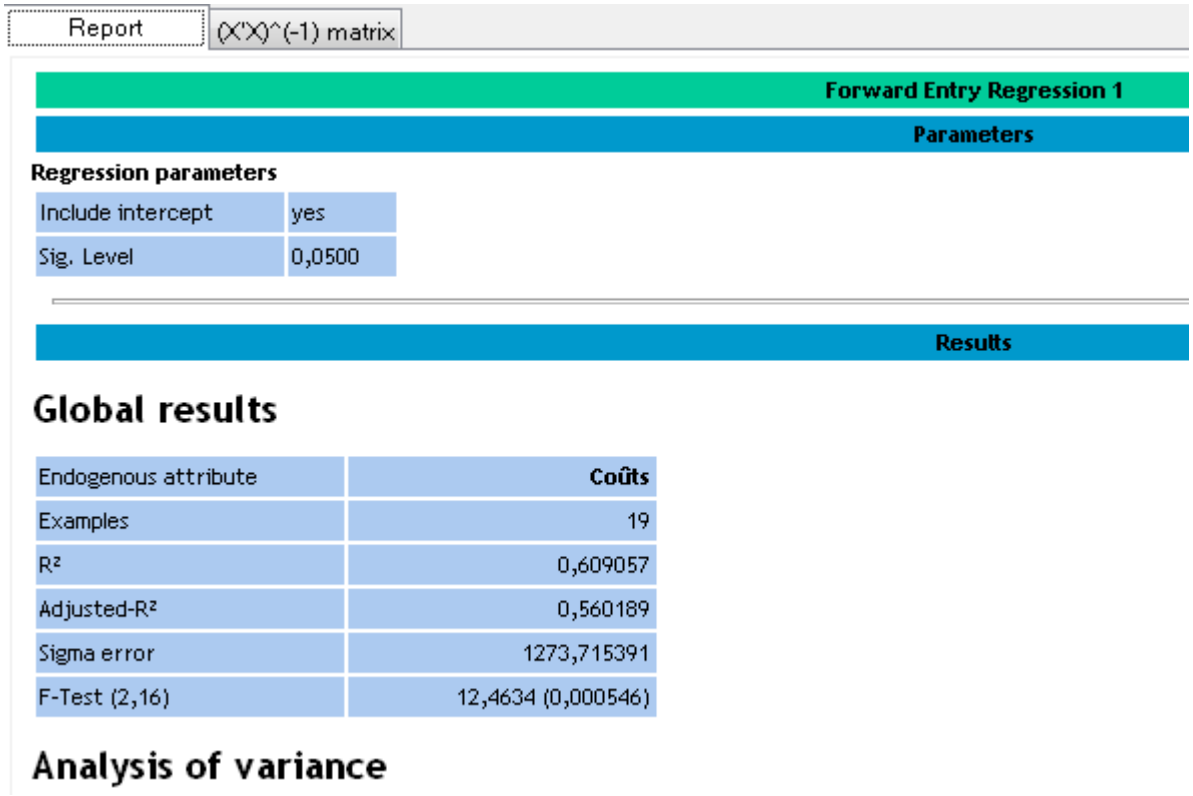


Ajoutons ensuite l'opérateur **Forward Entry Regression** du groupe **Regression**:



Nous pouvons dans les paramètres de cet opérateur (comme pour Minitab) donner le niveau de seuil de rejet des coefficients que nous allons laisser à 5%:

En exécutant cet opérateur nous voyons que nous retrouvons bien que les coefficient *C* et *B* comme pour les calculs faits dans MS Excel et avec Minitab mais à la différence que nous avons certaines informations en plus qui sont fort sympathiques d'abord dans le premier onglet **Report**:



Source	xSS	d.f.	xMS	F	p-value
Regression	40439876,2896	2	20219938,1448	12,4634	0,0005
Residual	25957614,3420	16	1622350,8964		
Total	66397490,6316	18			

Coefficients

Attribute	Coef.	std	t(16)	p-value
Intercept	35475,302554	1842,860853	19,250125	0,000000
Coût de B	5,320968	1,429095	3,723312	0,001849
Coût de C	5,417138	1,745312	3,103823	0,006825

Forward Selection Process

partial corr. F (p-value)	Step 1	Step 2	Step 3
d.f.	17	16	15
r(Y,Xj*/Xj1,Xj2...)	Coût de B : 0,6113	Coût de C : 0,6130	-
R^2	0,3737	0,6091	-
Coût de A	0,6007 9,60 (0,0065)	0,4042 3,13 (0,0962)	0,3051 1,54 (0,2337)
Coût de B	0,6113 10,14 (0,0054)	-	-
Coût de C	0,5199 6,30 (0,0225)	0,6130 9,63 (0,0068)	-

Residuals analysis

Att. name	Full statistics		Histogram			
	Statistics		Values	Count	Percent	Histogram
Err_Pred_fwdReg_1	Average	0,0000	x_<_-1730,6923	1	5,26%	
	Median	342,8704	-1730,6923_=<_x_<_-1331,7196	2	10,53%	
	Std dev. [Coef of variation]	1200,8704 [-249217432,2252]	-1331,7196_=<_x_<_-932,7469	2	10,53%	
	MAD [MAD/STDDEV]	1068,2069 [0,8895]	-932,7469_=<_x_<_-533,7742	2	10,53%	
	Min * Max [Full range]	-2129,67 * 1860,06 [3989,73]	-533,7742_=<_x_<_-134,8015	2	10,53%	
	1st * 3rd quartile [Range]	-1181,93 * 1057,95 [2239,88]	-134,8015_=<_x_<_264,1713	0	0,00%	
	Skewness (std-dev)	-0,2377 (0,5238)	264,1713_=<_x_<_663,1440	2	10,53%	
	Kurtosis (std-dev)	-1,3143 (1,0143)	663,1440_=<_x_<_1062,1167	4	21,05%	
			1062,1167_=<_x_<_1461,0894	3	15,79%	
			x>=_1461,0894	1	5,26%	

et dans le deuxième onglet nous avons la matrice d'information (pourquoi pas...):

Report | (X'X)^(-1) matrix

(X'X)^(-1)	Coût de B	Coût de C	intercept
Coût de B	1,2588608E-6	-8,94972E-8	-0,00094325199
Coût de C	-8,94972E-8	1,8775918E-6	-0,0015128489
intercept	-0,00094325199	-0,0015128489	2,0933425

Exercice 12.: Régression linéaire descendante (Backward Entry Selection)

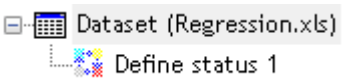
Tanagra V1.4.38

Nous allons reprendre les mêmes données que précédemment:

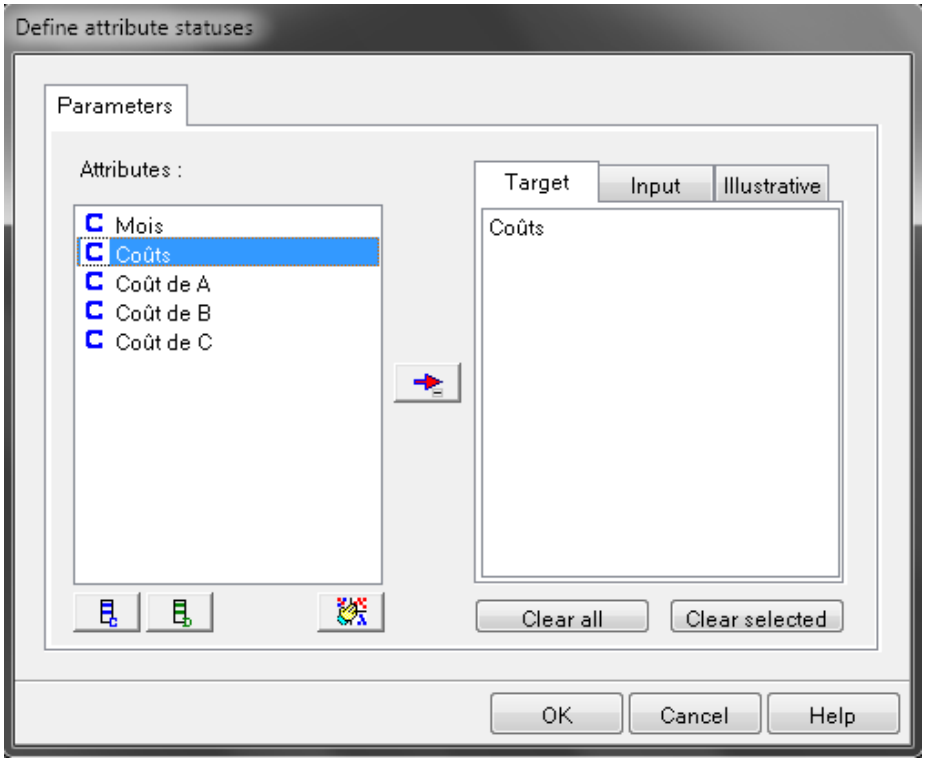
	A	B	C	D	E
1	Mois	Coûts	Coût de A	Coût de B	Coût de C
2	1	44439	515	541	928
3	2	43936	929	692	711
4	3	44464	800	710	824
5	4	41533	979	675	758
6	5	46343	1165	1147	635
7	6	44922	651	939	901
8	7	43203	847	755	580
9	8	43000	942	908	589
10	9	40967	630	738	682
11	10	48582	1113	1175	1050
12	11	45003	1086	1075	984
13	12	44303	843	640	828
14	13	42070	500	752	708
15	14	44353	813	989	804
16	15	45968	1190	823	904
17	16	47781	1200	1108	1120
18	17	43202	731	590	1065
19	18	44074	1089	607	1132
20	19	44610	786	513	839

pour effectuer une régression linéaire descendante (Backward Entry Selection) et comparer les résultats par rapport à ceux obtenus à la mains dans MS Excel et ceux obtenus aussi dans Minitab 15 dans le cours théorique.

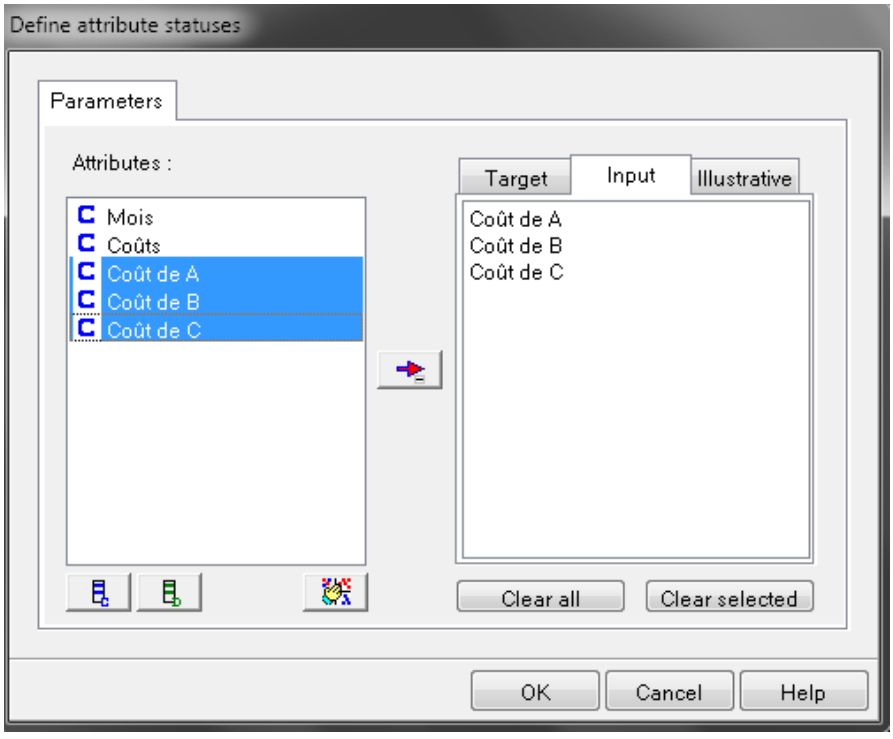
Nous y ajoutons un sélecteur de type **Define status** comme pour les exemples précédents:



avec la variable d'intérêt dans **Target**:



et dans les **Input**:



Ajoutons ensuite l'opérateur **Backward Elimination Reg** du groupe **Regression**:

Nous pouvons dans les paramètres de cet opérateur (comme pour Minitab) donner le niveau de seuil de rejet des coefficients que nous allons laisser à 5%:

En exécutant cet opérateur nous voyons que nous retrouvons bien que les coefficient *C* et *B* comme pour les calculs faits dans MS Excel et avec Minitab mais à la différence que nous avons certaines informations en plus qui sont fort sympathiques d'abord dans le premier onglet **Report**:

Report | (X'X)^(-1) matrix

Backward Elimination Reg 1

Parameters

Regression parameters

Include intercept	yes
Sig. Level	0,0500

Results

Global results

Endogenous attribute	**Coûts**
Examples	19
R^2	0,609057
Adjusted-R^2	0,560189
Sigma error	1273,715391
F-Test (2,16)	12,4634 (0,000546)

Analysis of variance

Source	xSS	d.f.	xMS	F	p-value
Regression	40439876,2896	2	20219938,1448	12,4634	0,0005
Residual	25957614,3420	16	1622350,8964		
Total	66397490,6316	18			

Coefficients

Attribute	Coef.	std	t(16)	p-value
Intercept	35475,302554	1842,860853	19,250125	0,000000
Coût de B	5,320968	1,429095	3,723312	0,001849
Coût de C	5,417138	1,745312	3,103823	0,006825

Backward Elimination Process

t (p-value)	Step 1	Step 2
#var (d.f.)	3 (15)	2 (16)
Adj.R² (R²)	0,575 (0,645)	0,560 (0,609)
Removed	Coût de A	-
Coût de A	1,24 (0,2337)	-
Coût de B	2,48 (0,0253)	3,72 (0,0018)
Coût de C	2,68 (0,0172)	3,10 (0,0068)

Residuals analysis

Att. name: Err_Pred_bwReg_1

Full statistics

Statistics	
Average	0,0000
Median	342,8704
Std dev. [Coef of variation]	1200,8704 [-249217432,2252]
MAD [MAD/STDDEV]	1068,2069 [0,8895]
Min * Max [Full range]	-2129,67 * 1860,06 [3989,73]
1st * 3rd quartile [Range]	-1181,93 * 1057,95 [2239,88]
Skewness (std-dev)	-0,2377 (0,5238)
Kurtosis (std-dev)	-1,3143 (1,0143)

Histogram

Values	Count	Percent	Histogram
x_<_-1730,6923	1	5,26%	
-1730,6923_=<_x_<_-1331,7196	2	10,53%	
-1331,7196_=<_x_<_-932,7469	2	10,53%	
-932,7469_=<_x_<_-533,7742	2	10,53%	
-533,7742_=<_x_<_-134,8015	2	10,53%	
-134,8015_=<_x_<_264,1713	0	0,00%	
264,1713_=<_x_<_663,1440	2	10,53%	
663,1440_=<_x_<_1062,1167	4	21,05%	
1062,1167_=<_x_<_1461,0894	3	15,79%	
x>=_1461,0894	1	5,26%	

et dans le deuxième onglet nous avons encore une fois la matrice d'information:

Report | (X'X)^(-1) matrix

(X'X)^(-1)	Coût de B	Coût de C	intercept
Coût de B	1,2588608E-6	-8,94972E-8	-0,00094325199
Coût de C	-8,94972E-8	1,8775918E-6	-0,0015128489
intercept	-0,00094325199	-0,0015128489	2,0933425

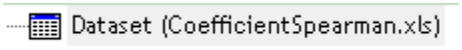
Exercice 13.: Coefficient de corrélation de Spearman (Spearman rho)

Tanagra V1.4.48

Nous allons partir ici des mêmes données que celles utilisées dans le cours théorique pour encore une fois vérifier que nous retombons sur la même chose ou pas:

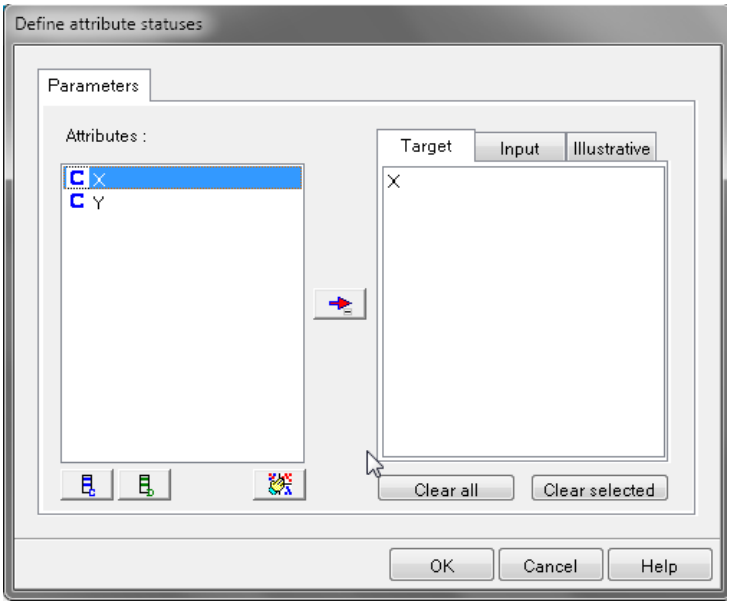
	A	B
1	*X*	*Y*
2	5.7	8.1
3	3.2	5.5
4	8.4	3.4
5	4.1	7.9
6	6.9	4.6
7	5.3	1.6
8	1.7	8.5
9	3.2	7.1
10	2.5	8.7
11	7.4	5.7

Nous ouvrons ce fichier dans Tanagra comme à l'habitude:

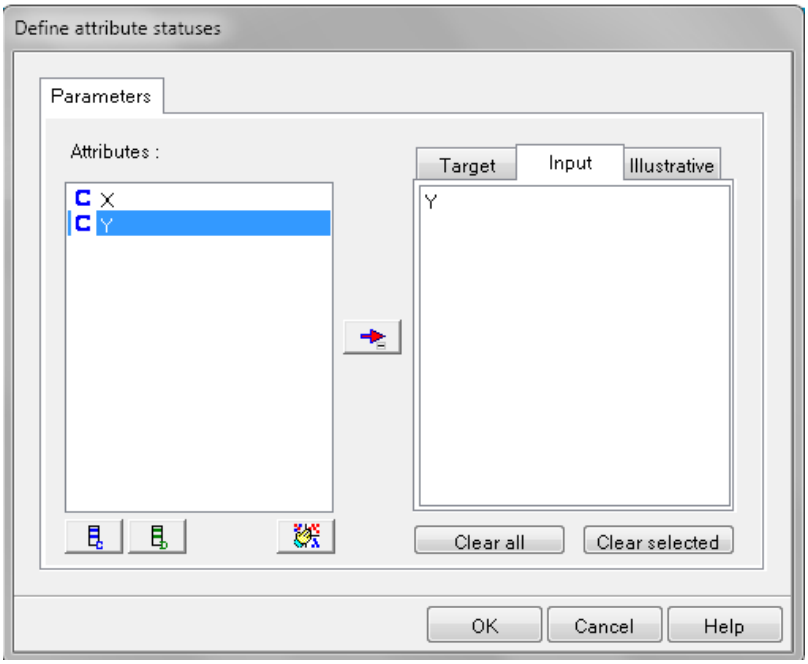


et nous lui mettons le sélecteur **Define Status**:

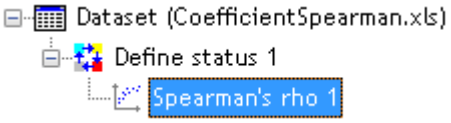
avec en **Input** le champ *X* (en réalité peut importe lequel comme nous l'avons vu dans le cours théorique):



et en **Input** la variable restante:



Nous ajoutons ensuite l'opérateur **Spearman's rho** du groupe **Nonparametric statistics**:



et nous l'exécutons sans autre pour obtenir:

Spearman's rho 1

Parameters

Cross-tab parameters	
Sort results	non
Input list	Target (Y) and input (X)

Results

Y	X	r	r²	t	Pr(> |t|)
X	Y	-0,6383	0,4074	-2,3453	0,0470

Computation time : 0 ms.
Created at 07/09/2013 23:53:28

Ce qui outre le test *t* que nous n'avons pas démontré dans le cours théorique, est parfaitement conforme aux calculs faits à la main.

Exercice 14.: Régression logistique binaire (SPV)

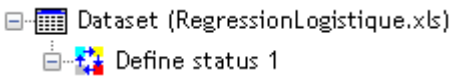
Tanagra V1.4.44

Ici encore nous allons vérifier si les calculs faits à la main lors de la démonstration du principe de la régression logistique correspondent avec MS Excel et Minitab.

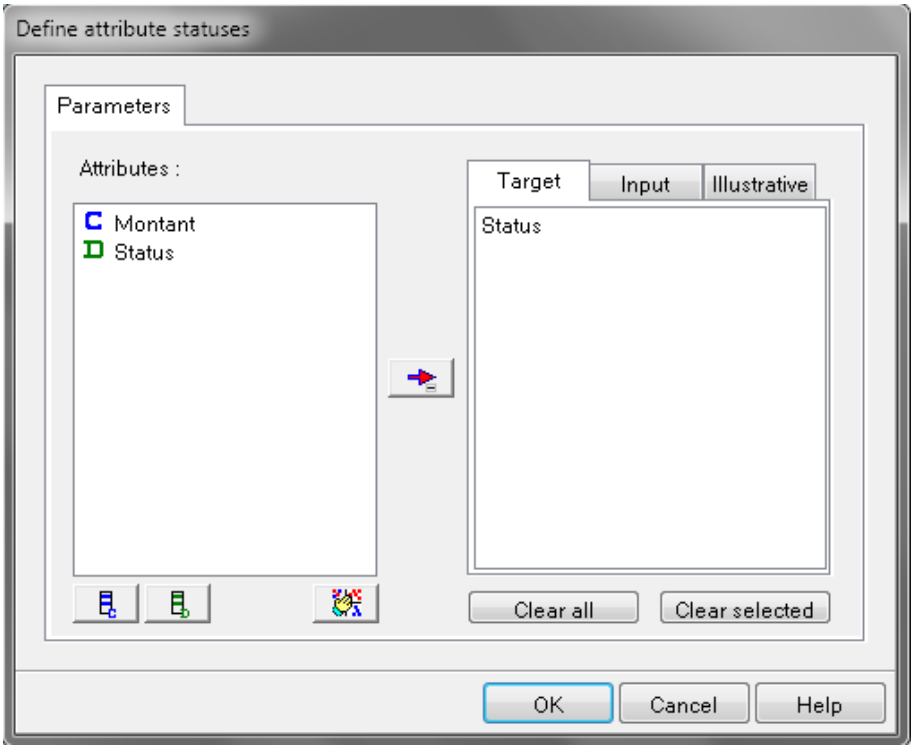
Nous partons donc de la liste des crédits suivante de 137 lignes (fichier *RegressionLogistique.xls*):

	A	B
1	**Montant**	**Status**
2	27200	Oui
3	27200	Oui
4	27200	Oui
5	27200	Oui
6	27200	Oui
7	27200	Oui
8	27200	Oui
9	27200	Oui
10	27200	Oui
11	27200	Non
12	27200	Oui
13	27200	Oui
14	27200	Oui
15	27200	Oui
16	27200	Oui
17	27200	Oui
18	27200	Oui
19	27200	Oui
20	27200	Oui
21	27200	Oui
22	27200	Oui
23	27200	Oui

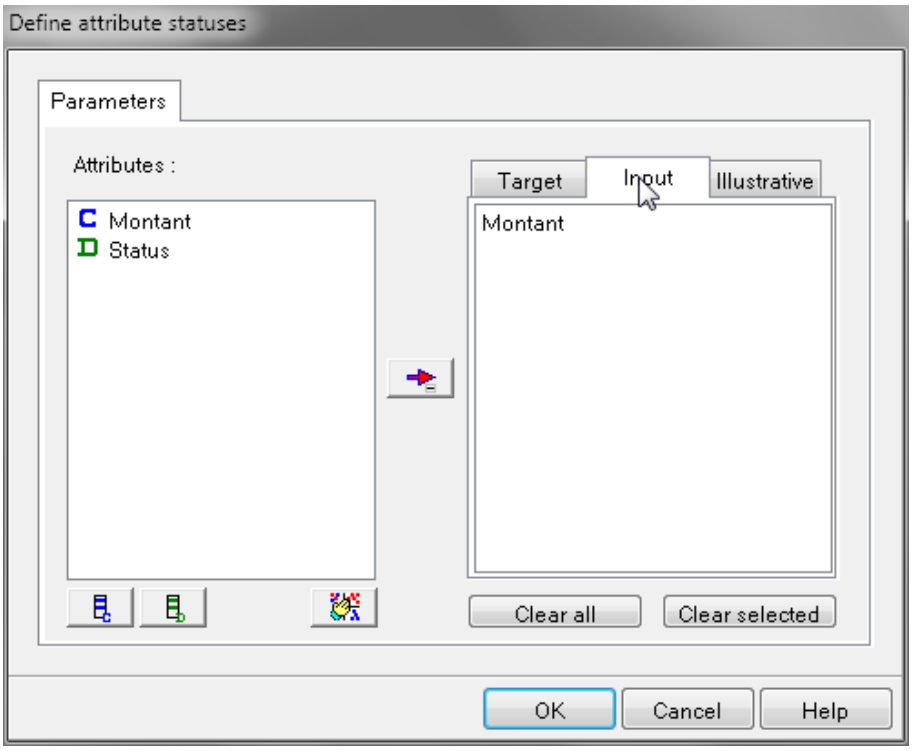
Nous l'importons dans Tanagra comme à l'habitude et y mettons un sélecteur **Define status**:



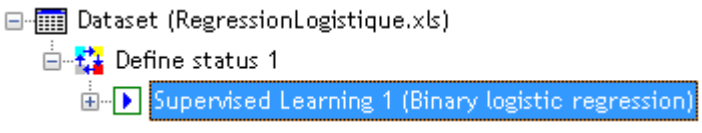
Dans les paramètres de celui-ci nous mettons le champ **Status** en *Target* (qui doit absolument être une variable discrète binaire textuelle):



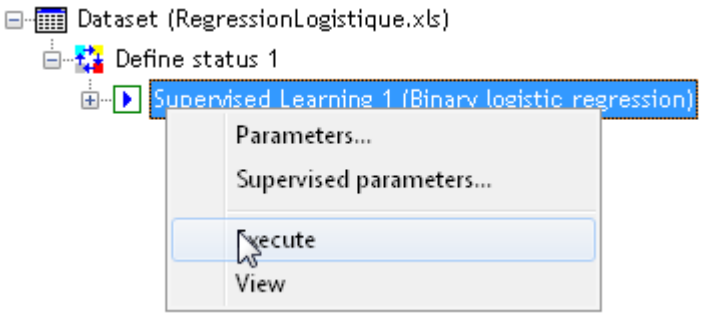
et le *Montant* en **Input**:



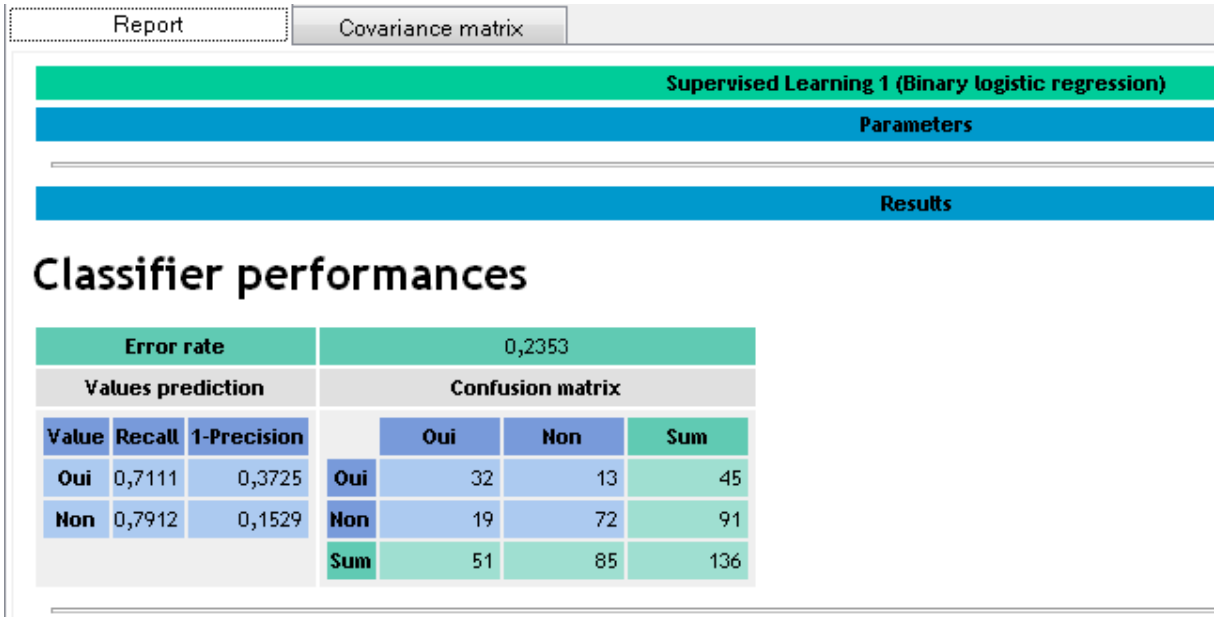
N'oubliez pas d'exécuter ce composant! Ensuite nous rajoutons l'opérateur **Binary logistic regression** du groupe **Spv**:



sans y changer les paramètres du composant nous l'exécutons de suite:



Il vient alors après avoir fait un **View** après l'exécution (les informations sont plus pertinentes que celles renvoyées par Minitab):



Avant d'aller plus loin nous voyons dans la matrice de confusion que sur les 91 bon débiteurs (correspondant ici au statut: *Non*) qu'il y avait dans la liste d'origine, le modèle en prédit 19 comme étant mauvais débiteurs et 72 comme étant bons. La même lecture est valable pour les 45 mauvais débiteurs. Si évidemment le modèle était parfait, la matrice de confusion serait diagonale.

Continuons avec les captures d'écran de l'onglet **Report**:

Classifier characteristics

Data description

Target attribute	Status (2 values)
# descriptors	1

Adjustement quality

Predicted attribute		Status
Positive value		Oui
Number of examples		136
Model Fit Statistics		
Criterion	Intercept	Model
AIC	174,666	127,101
SC	177,579	132,926
-2LL	172,666	123,101
Model Chi² test (LR)		
Chi-2		49,5656
d.f.		1
P(>Chi-2)		0,0000
R²-like		
McFadden's R²		0,2871
Cox and Snell's R²		0,3054
Nagelkerke's R²		0,4248

Ici il n'y a pas grand chose à dire puisque nous n'avons pas encore étudié ces indicateurs dans le cours théorique mais celui du khi-2 est cependant un classique dont l'interprétation ne souffre d'aucun doute sur la conclusion du modèle.

Enfin, toujours dans le même onglet **Report** et pour finir:

Attributes in the equation

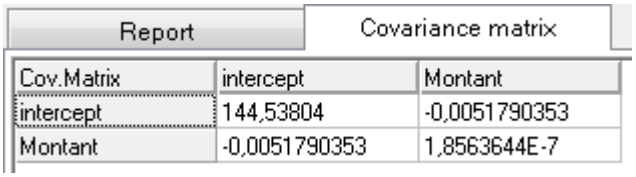
Attribute	Coef.	Std-dev	Wald	Signif
constant	61,318317	12,0224	26,0135	0,0000
Montant	-0,002211	0,0004	26,3346	0,0000

Odds ratios and 95% confidence intervals

Attribute	Coef.	Low	High
Montant	0,9978	0,9969	0,9986

Nous voyons que contrairement à Minitab et à Excel les signes des coefficients sont inversés mais c'est juste une convention dans le choix de distribuer les signe "-" présent dans l'exponentielle du modèle logistique à l'intérieur de la parenthèse.

Et nous avons dans le deuxième onglet la **Covariance matrix**:



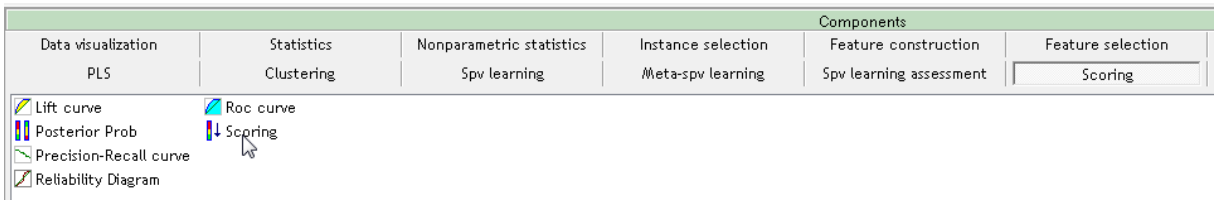
Cov.Matrix	intercept	Montant
intercept	144,53804	-0,0051790353
Montant	-0,0051790353	1,8563644E-7

Exercice 15.: Lift Curve et ROC Curve (sur régression logistique binaire)

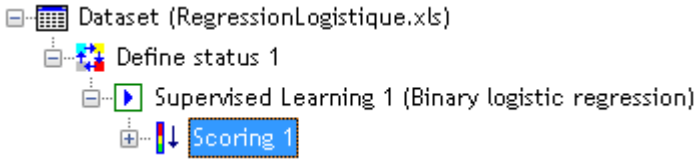
Tanagra V1.4.44

Le but va être ici de vérifier que nous retrouvons la même forme de Lift Curve et ROC Curve (Receiver Operating Characteristic) que celles obtenues à la main avec MS Excel dans le cours théorique pour la régression logistique (mais le principe est toujours le même).

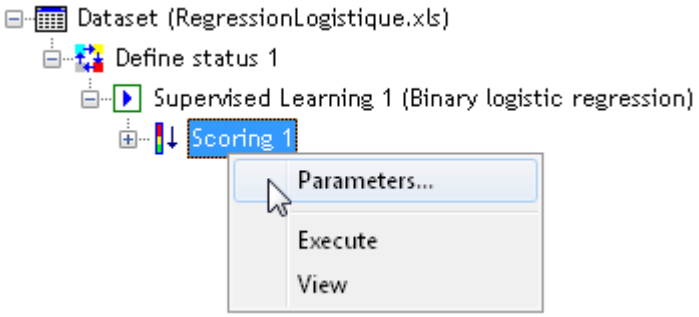
Pour cela nous ajoutons d'abord le composant **Scoring** du groupe **Scoring**:



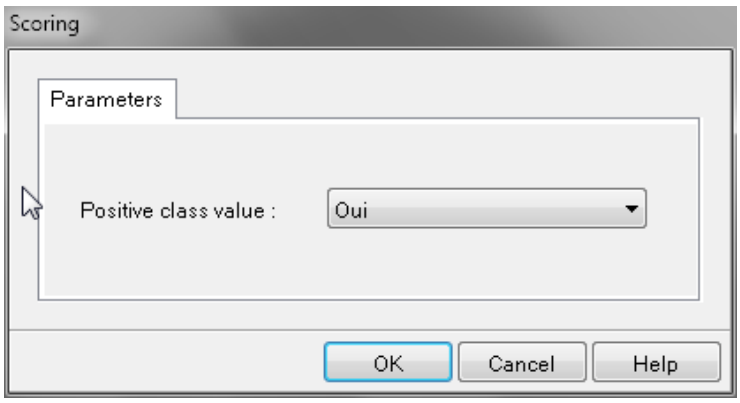
Afin d'avoir:



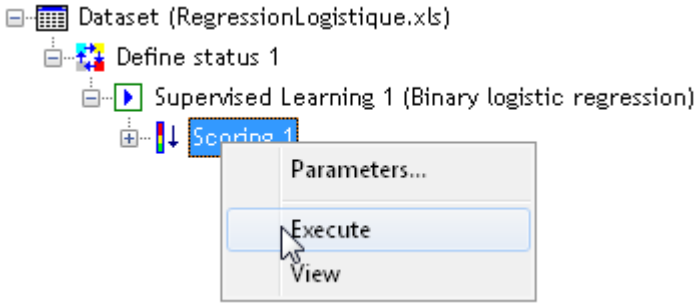
et dans les paramètres de ce composant:



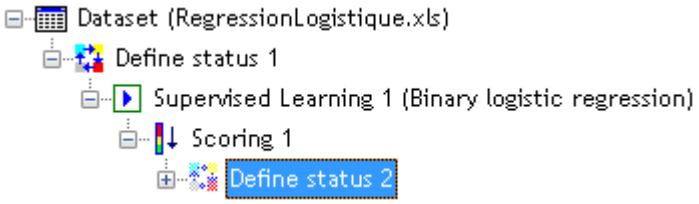
nous disons que nous allons nous intéresser aux débiteurs à risque:



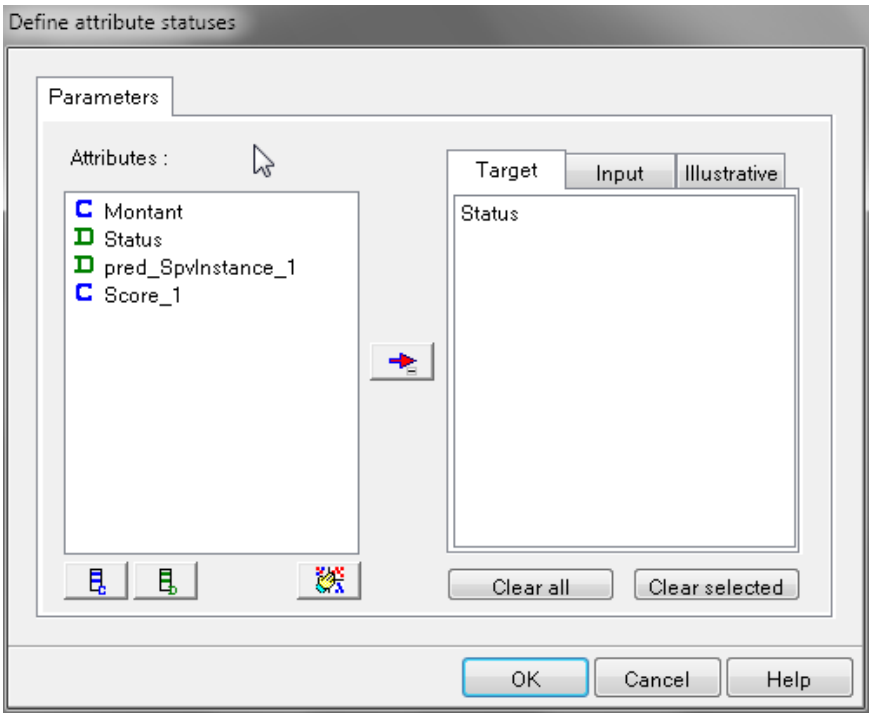
Vous n'oubliez pas ensuite d'exécuter ce composant:



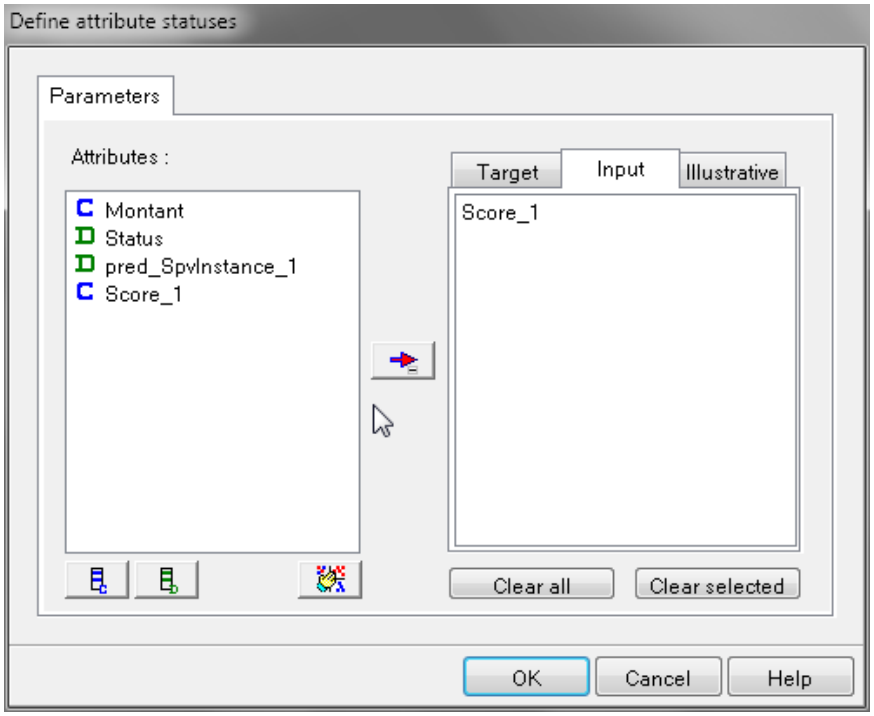
Une fois ceci fait, il ne sert à rien dans l'état présent. Il faut lui ajouter un sélecteur **Define satut**:



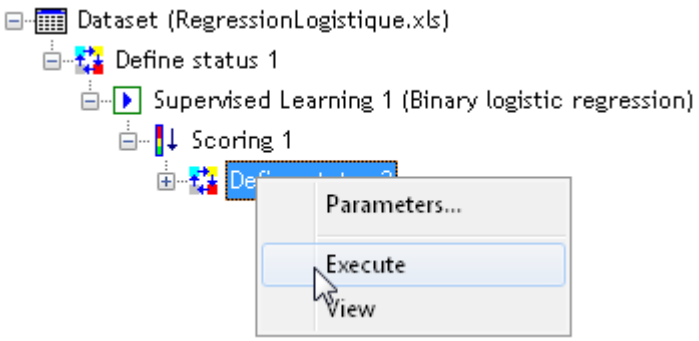
Avec *Status* comme champ dans l'onglet **Target**:



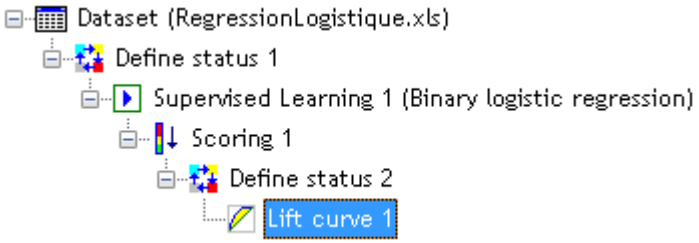
et **Score_1** dans **Input**:



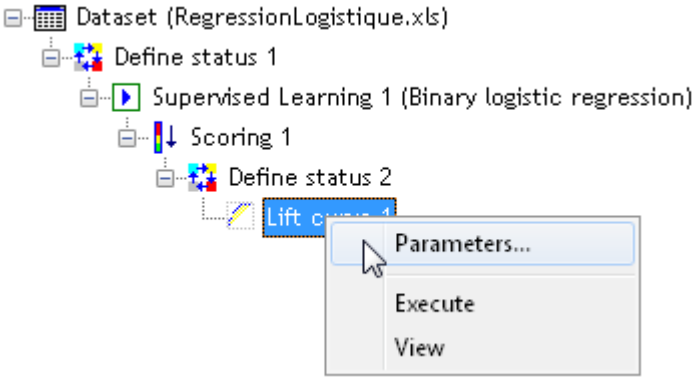
Vous n'oubliez pas ensuite d'exécuter aussi ce composant:



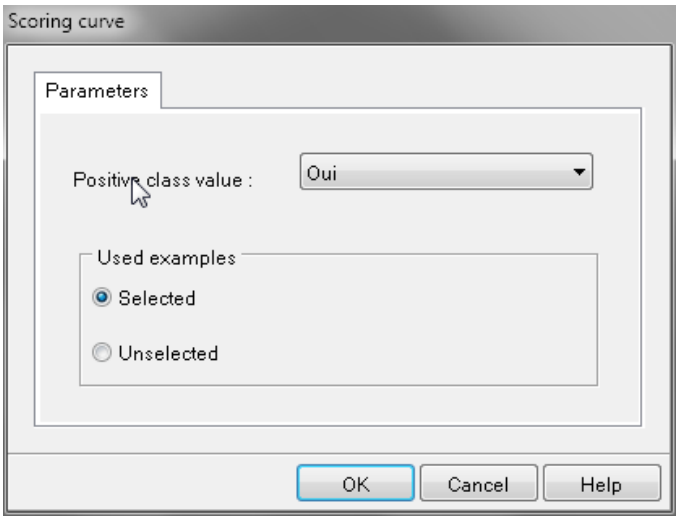
Enfin, nous rajoutons l'opérateur **Lift curve** du groupe **Scoring**:



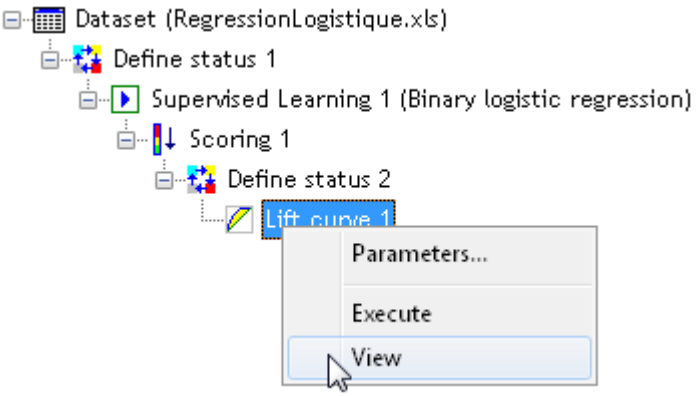
Dans ses paramètres:



nous prenons:



et nous affichons le contenu:



Pour obtenir un rapport en deux onglets dont le premier contient:

Lift curve 1

Parameters

Positive class value : Oui
Used examples : Selected

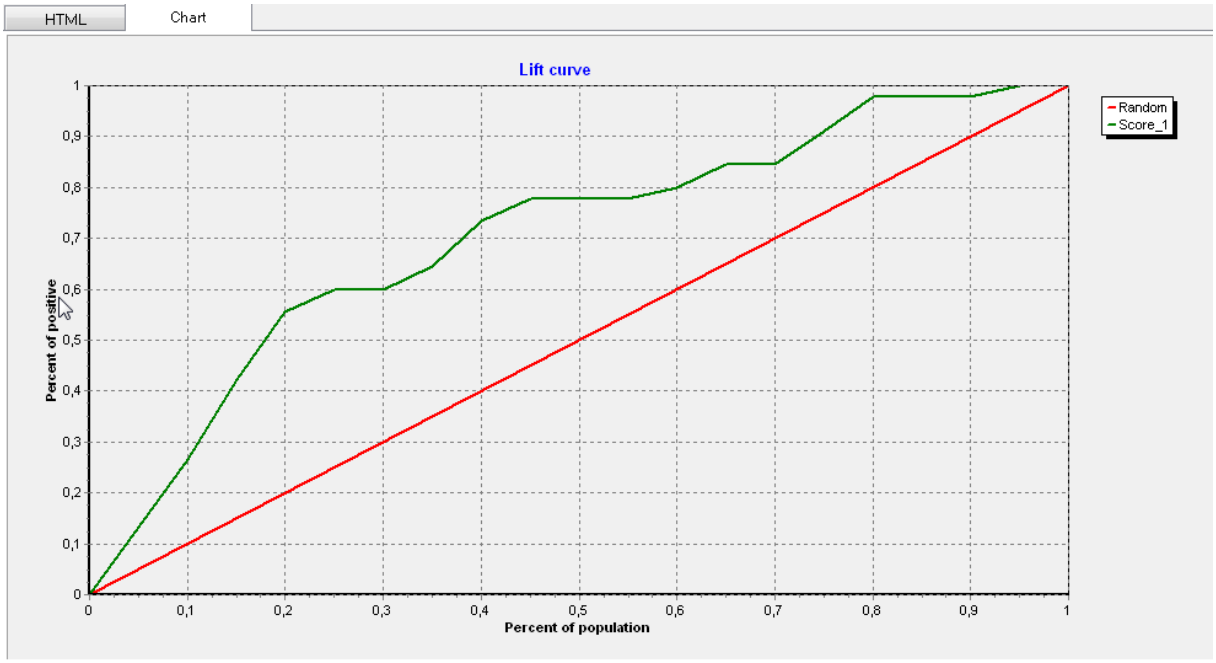
Results

LIFT Curve

Sample size : 136
Positive examples : 45

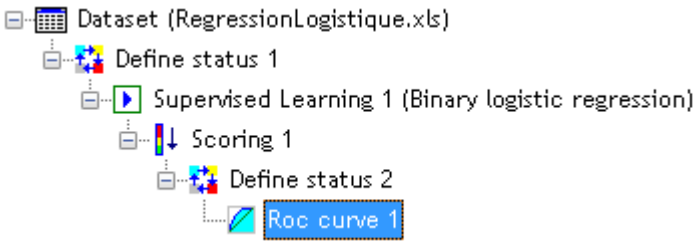
Score Attribute	Score_1	
Target size (%)	Score	TP-Rate
0	0,7646	0,0000
5	0,7646	0,1333
10	0,7646	0,2667
15	0,7646	0,4222
20	0,7646	0,5556
25	0,5182	0,6000
30	0,5182	0,6000
35	0,5182	0,6444
40	0,2220	0,7333
45	0,2220	0,7778
50	0,2220	0,7778
55	0,2220	0,7778
60	0,2220	0,8000
65	0,1862	0,8444
70	0,1862	0,8444
75	0,1862	0,9111
80	0,1862	0,9778
85	0,0082	0,9778
90	0,0082	0,9778
95	0,0082	1,0000
100	0,0082	1,0000

et le deuxième onglet contient simplement un tracé de la colonne **TP-Rate** (*TP*=True Positive) en fonction de la **Target-Size**:

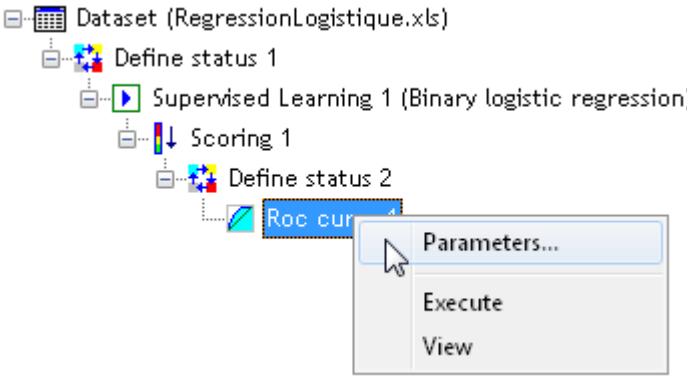


Nous pouvons observer qu'aussi bien le TP-Rate que la courbe Lift sont erronées par rapport au calcul à la main et Minitab+SPSS! Après étude du code source de Tanagra de ma part il semblait qu'il y ait une erreur de codage car ce que fait ci-dessus Tanagra c'est qu'il ne nous montre que des multiples du ratio 1/45. Je pense que cette erreur vient du fait qu'à la base le développeur n'a peut-être pas pensé que l'on pourrait avoir des très nombreux doublons dans la population d'origine. Donc pour l'instant utilisez Minitab/SPSS ou autre...

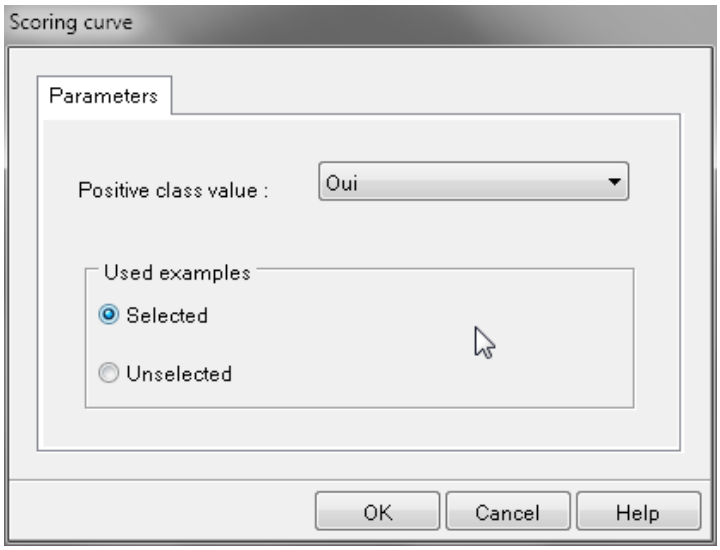
Enfin, nous rajoutons le composant **Roc curve** du groupe **Scoring**:



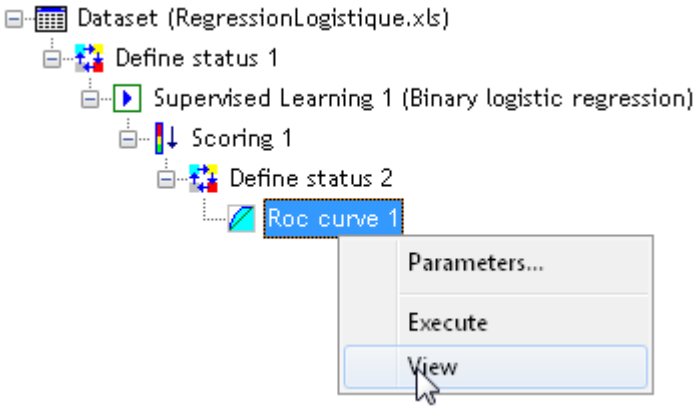
Dans ses paramètres:



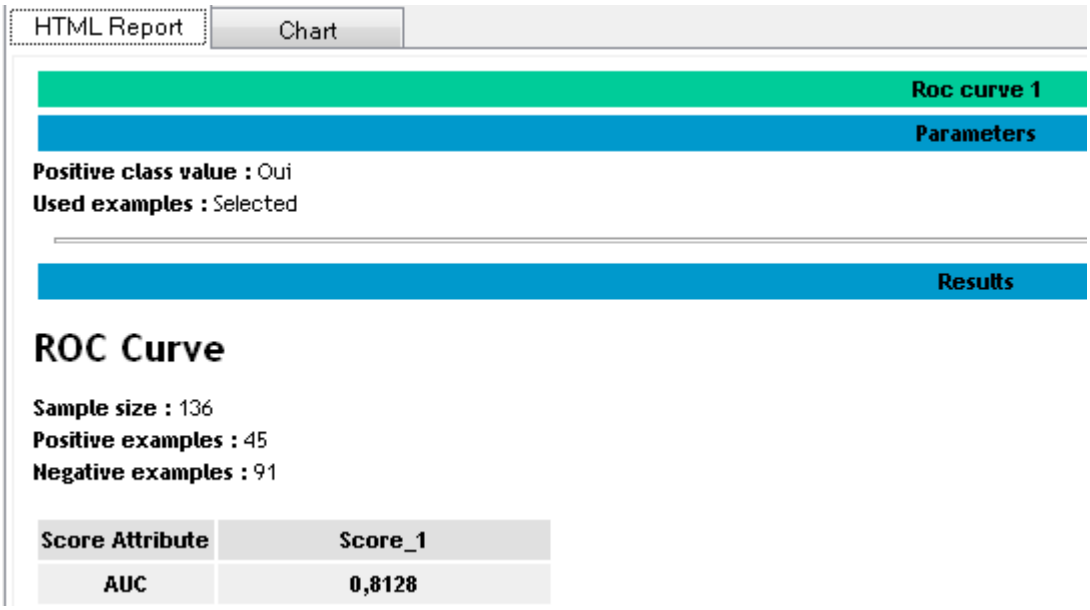
Nous prenons:



Nous validons et faisons un **View**:

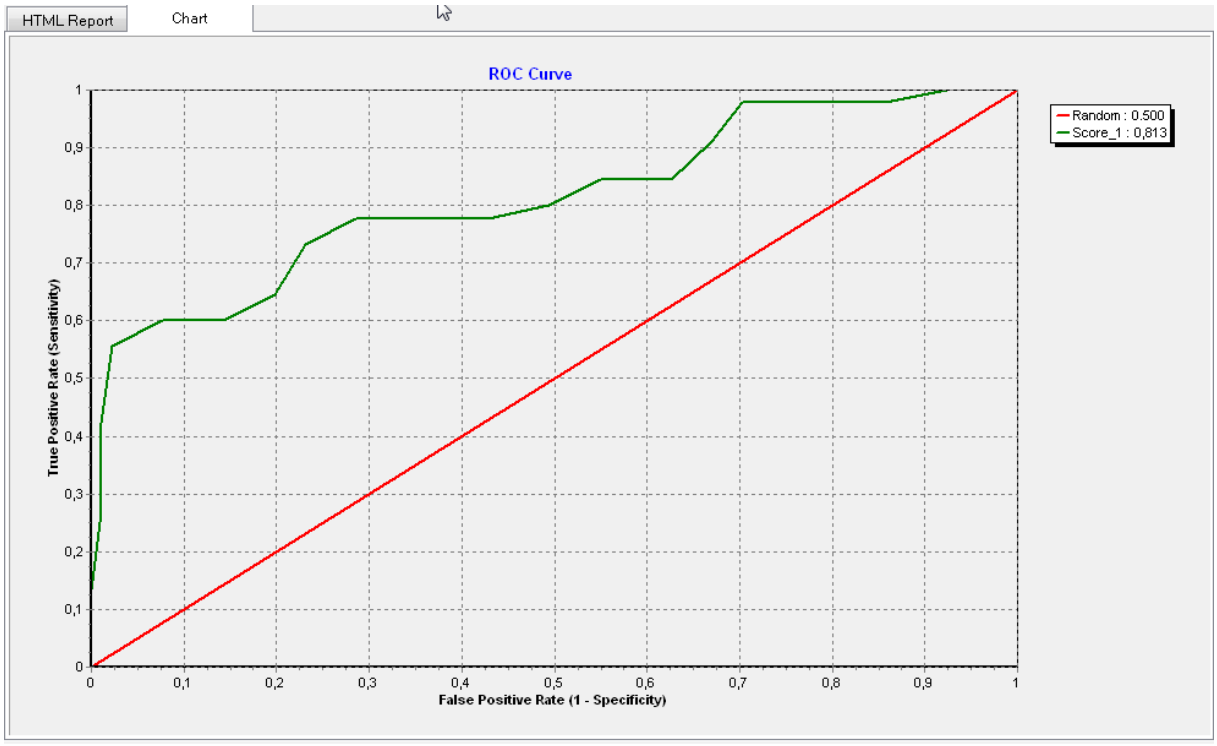


Pour obtenir au final un rapport en deux onglets le premier contenant:



Target size (%)	Score	FP-Rate	TP-Rate
0	0,7646	0,0000	0,0000
5	0,7646	0,0000	0,1333
10	0,7646	0,0110	0,2667
15	0,7646	0,0110	0,4222
20	0,7646	0,0220	0,5556
25	0,5182	0,0769	0,6000
30	0,5182	0,1429	0,6000
35	0,5182	0,1978	0,6444
40	0,2220	0,2308	0,7333
45	0,2220	0,2857	0,7778
50	0,2220	0,3626	0,7778
55	0,2220	0,4286	0,7778
60	0,2220	0,4945	0,8000
65	0,1862	0,5495	0,8444
70	0,1862	0,6264	0,8444
75	0,1862	0,6703	0,9111
80	0,1862	0,7033	0,9778
85	0,0082	0,7802	0,9778
90	0,0082	0,8571	0,9778
95	0,0082	0,9231	1,0000
100	0,0082	1,0000	1,0000

et le deuxième onglet:



Nous pouvons observer qu'aussi bien le TP-Rate que le FP-Rate que la courbe ROC sont erronées par rapport au calcul à la main et Minitab+SPSS! Après étude du code source de Tanagra de ma part il semblait qu'il y ait une erreur de codage car ce que fait ci-dessus Tanagra c'est qu'il ne nous montre encore une fois que des multiples du ratio 1/45. Je pense que cette erreur vient du fait qu'à la base le développeur n'a peut-être pas pensé que l'on pourrait avoir des très nombreux doublons dans la population d'origine. Donc pour l'instant utilisez Minitab/SPSS ou autre...

Exercice 16.: Test-T homoscédatique

Tanagra V1.4.44

Nous allons ici vérifier si nous retombons sur le même résultat que celui obtenu en cours lors de l'étude théorique et la démonstration mathématique du test-t de comparaison des moyennes deux échantillons non appariés.

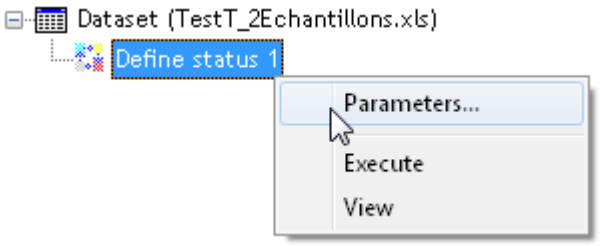
Nous allons travailler avec le tableau contenant les données du cours théorique:

	A	B
1	Pipeline1	Pipeline2
2	163	167
3	150	157
4	171	149
5	155	145
6	186	135
7	145	157
8	154	135
9	173	167
10	152	154
11	150	165
12	143	170
13	138	165
14	166	154
15	193	176
16	158	155
17	175	157
18	167	134
19	150	156
20	158	147

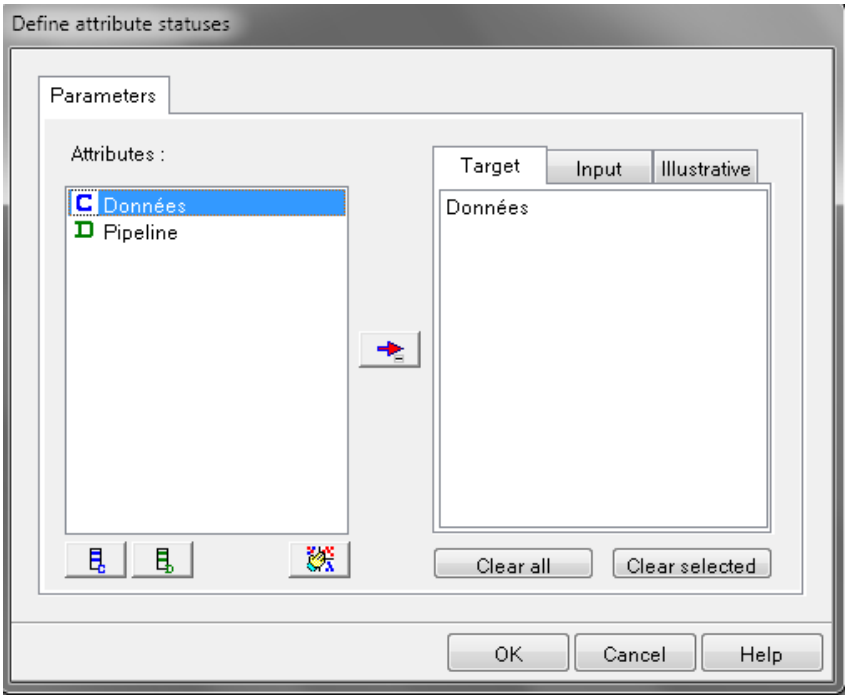
que nous allons devoir redisposer de la manière suivant pour Tanagra (ce qui est la structure conforme à du Data Mining):

	A	B
1	**Données**	**Pipeline**
2	163	Pipeline 1
3	150	Pipeline 1
4	171	Pipeline 1
5	155	Pipeline 1
6	186	Pipeline 1
7	145	Pipeline 1
8	154	Pipeline 1
9	173	Pipeline 1
10	152	Pipeline 1
11	150	Pipeline 1
12	143	Pipeline 1
13	138	Pipeline 1
14	166	Pipeline 1
15	193	Pipeline 1
16	158	Pipeline 1
17	175	Pipeline 1
18	167	Pipeline 1
19	150	Pipeline 1
20	158	Pipeline 1
21	167	Pipeline 2
22	157	Pipeline 2
23	149	Pipeline 2
24	145	Pipeline 2
25	135	Pipeline 2
26	157	Pipeline 2

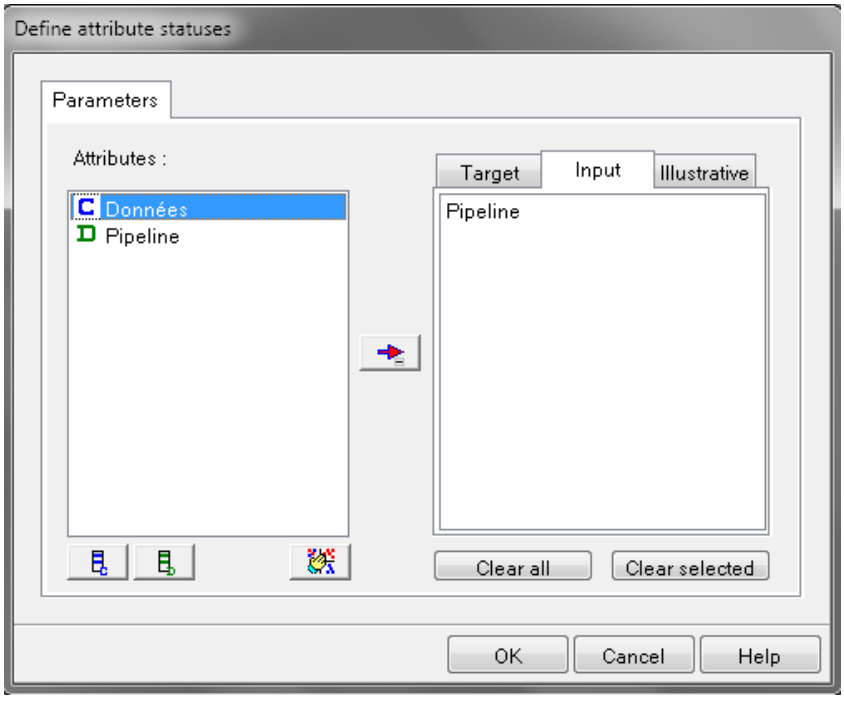
Nous l'importons dans Tanagra et y ajoutons un sélecteur **Define status**:



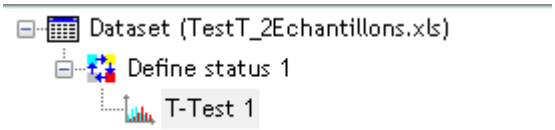
Pour y mettre comme **Target** les données:



et comme **Input** les catégories:



Nous ajoutons le composant **T-Test**:



et en affichons le contenu pour avoir:

T-Test 1

Parameters

Parameters	
Sort results	no

Results

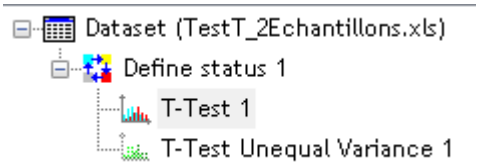
Attribute_Y	Attribute_X	Description				Statistical test	
		Value	Examples	Average	Std-dev	T	5,3684 / 4,3301 = 1,239791
Données	Pipeline	Pipeline 1	19	160,3684	14,5343	d.f.	36,00
		Pipeline 2	19	155,0000	12,0416	p-value	0,223074
		All	38	157,6842	13,4428		

Nous voyons que les sorties correspondent à ce que nous avons calculé dans le cours théorique. Il manque cependant l'intervalle de confiance qui est important dans la pratique. C'est dommage...

Exercice 17.: Test-T hétéroscédastique

Tanagra V1.4.44

Nous continuons l'exemple d'avant en ajoutant l'opérateur **T-Test Unequal Variance**:



et nous affichons le résultat:

T-Test Unequal Variance 1

Parameters

Parameters	
Sort results	no

Results

Attribute_Y	Attribute_X	Description				Statistical test	
		Value	Examples	Average	Std-dev	T	5,3684 / 4,3301 = 1,239791
Données	Pipeline	Pipeline 1	19	160,3684	14,5343	d.f.	34,80
		Pipeline 2	19	155,0000	12,0416	p-value	0,223301
		All	38	157,6842	13,4428		

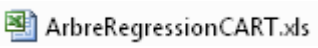
Là encore il manque l'intervalle de confiance mais ce qui est sympathique que les d.f. ne sont pas arrondis et que nous tombons exactement sur les degrés de libertés obtenus avec l'équation de Welch–Satterthwaite démontrée en cours.

Exercice 18.: Clustering CART (arbres de régression)

Tanagra V1.4.38

Nous allons ici vérifier si nous retombons sur le même résultat que celui obtenu en cours lors de l'étude théorique et la démonstration mathématique du principe de fonctionnement des arbres de régression.

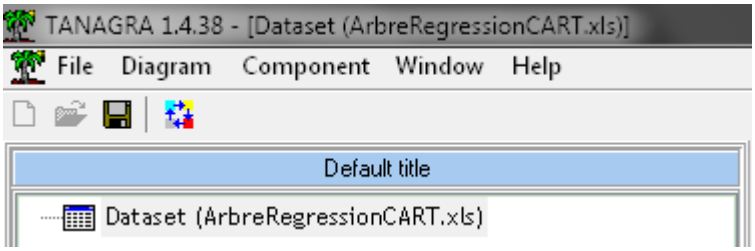
Nous allons travailler avec le fichier suivant:



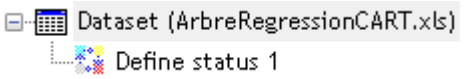
qui contient les mêmes données que celles vues dans le cours théorique:

	A	B	C
1	Revenus	Surface	Propriétaire
2	60	18.4	1
3	85.5	16.8	1
4	64.8	21.6	1
5	61.5	20.8	1
6	87	23.6	1
7	110.1	19.2	1
8	108	17.6	1
9	82.8	22.4	1
10	69	20	1
11	93	20.8	1
12	51	22	1
13	81	20	1
14	75	19.6	2
15	52.8	20.8	2
16	64.8	17.2	2
17	43.2	20.4	2
18	84	17.6	2
19	49.2	17.6	2
20	59.4	16	2
21	66	18.4	2
22	47.4	16.4	2
23	33	18.8	2
24	51	14	2
25	63	14.8	2

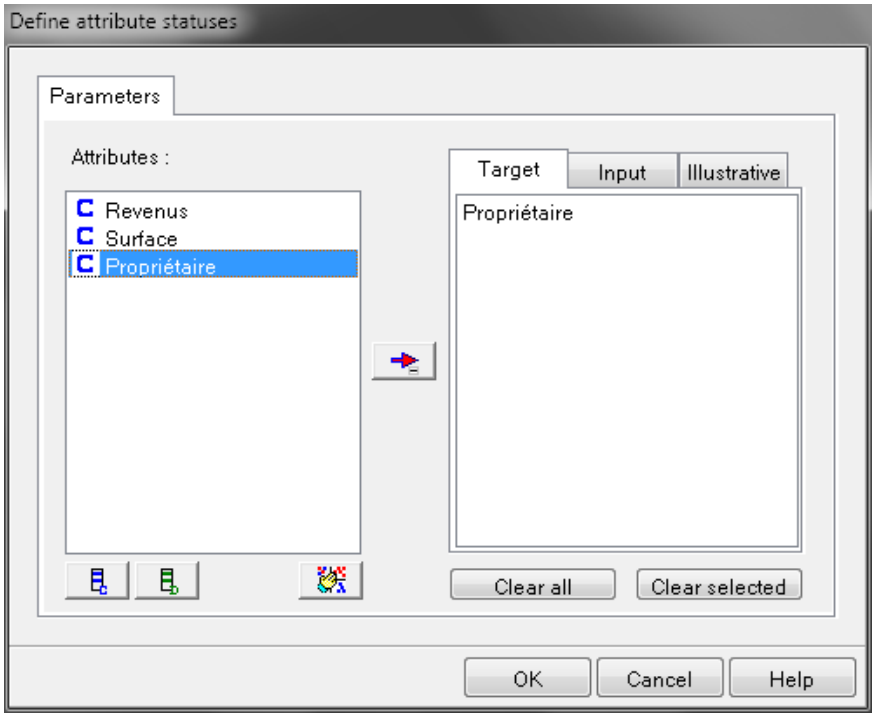
Nous l'importons dans Tanagra en utilisant la même procédure que les exercices précédents:



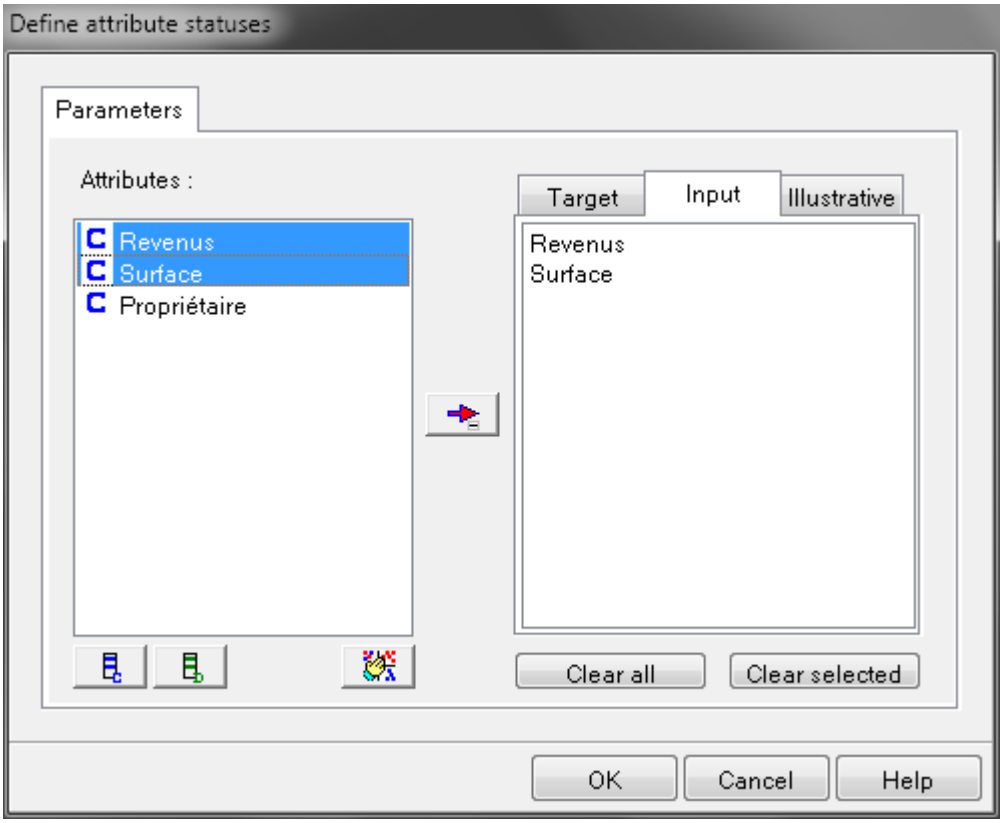
Nous y ajoutons un sélecteur de type **Define status** comme pour les exemples précédents:



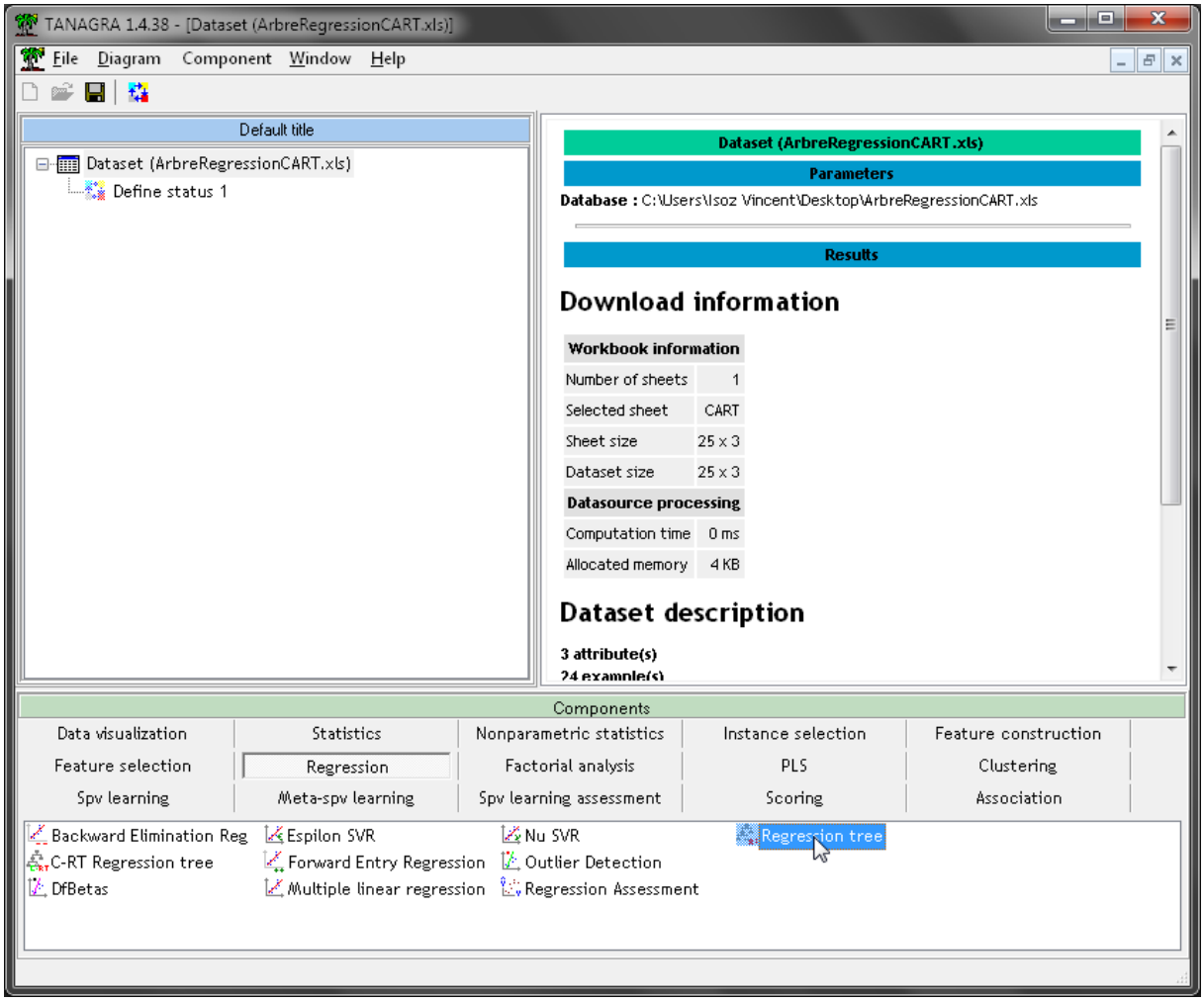
mais avec la variable d'intérêt dans **Target**:



et dans les **Input**:



Ajoutons ensuite l'opérateur **Regression tree**:



Regression tree

Description
Predict values of a continuous target attribute with a regression tree, input(s) can be continuous or discrete. The used algorithm is the univariate version of the Clustering Tree (CTP -- See "Clustering" tab). The learning method includes a post pruning process. Detailed results about the pruning sequence can be depicted. The best tree on the pruning set and the selected tree are underlined.

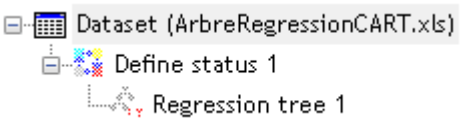
Precondition
One continuous target attribute is needed. The input attributes can be continuous or discrete.

Target attribute(s)
The continuous class attribute.

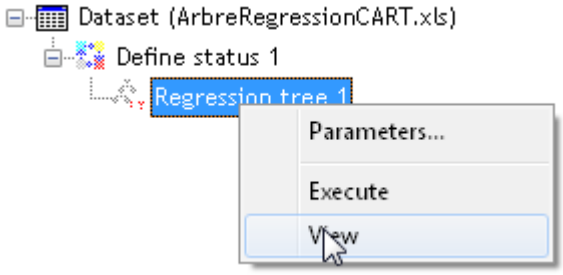
Input attribute(s)
One or more continuous and/or discrete input attributes.

Postcondition
The predictions and the residuals columns (two new continuous attributes) are added in the dataset.

ce qui donnera:



on fait un clic droit sur l'opération pour choisir **View**:



et on admire le résultat dans la fenêtre de sortie:

Regression tree 1

Parameters

Tree Parameters	
Rnd generator	1
Max Number of Clusters	50
Distance normalization	0
Min. size for split	5
Min. size of leaves	2
Max. depth	10
Goodness threshold	0.00
Pruning set size	33 %
Delta	0.0010
Show all tree sequence	0

Results

Global results

Endogenous attribute	**Propriétaire**
Examples	24
R^2	0.3403

Trees sequence (# 3) -- Within-Groups Sum of Squares

N°	# Leaves	WSS (growing set)	WSS (pruning set)
3	1	1.0000	1.0000
2	2	0.5455	1.6033
1	4	0.3750	1.2500

Tree description

Number of nodes	7
Number of leaves	4

Tree

- Surface < 18.0000 then **avg(Propriétaire) = 2.0000** (std-dev = 0.0000, with 5 examples [31.25%])
- Surface >= 18.0000
  - Surface < 19.8000 then **avg(Propriétaire) = 1.5000** (std-dev = 0.5774, with 4 examples [25.00%])
  - Surface >= 19.8000
    - Revenus < 57.1500 then **avg(Propriétaire) = 1.5000** (std-dev = 0.7071, with 2 examples [12.50%])
    - Revenus >= 57.1500 then **avg(Propriétaire) = 1.0000** (std-dev = 0.0000, with 5 examples [31.25%])

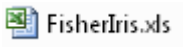
Computation time : 0 ms.
Created at 08.02.2012 12:17:42

Nous voyons que nous obtenons la même chose que dans le cours théorique à la différence que l'arbre s'arrête plus tôt.

Exercice 19.: K-NN (K nearest neighbors)

Tanagra V1.4.48

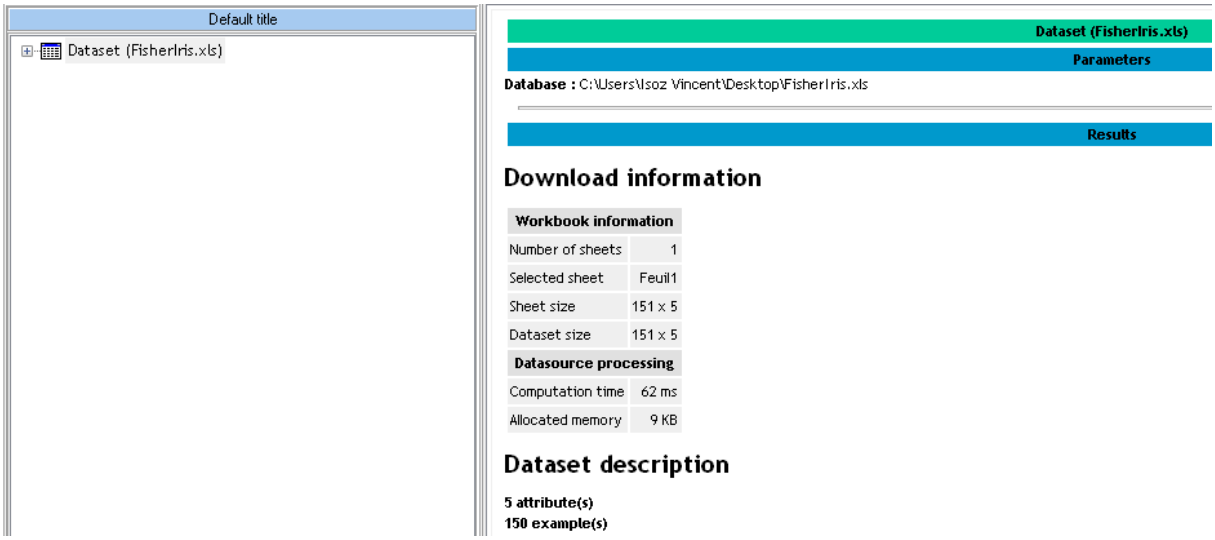
Nous avons vu en cours l'approche des k plus proches voisins. Nous allons appliquer ici ce qui a été présenté en cours avec le fichier Excel des fleurs d'Iris



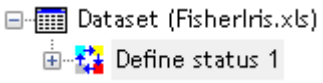
dont le contenu est:

	A	B	C	D	E
1	Sepal length	Sepal width	Petal length	Petal width	Species
2	7.9	3.8	6.4	2	*I. virginica*
3	7.7	3.8	6.7	2.2	*I. virginica*
4	7.7	2.6	6.9	2.3	*I. virginica*
5	7.7	2.8	6.7	2	*I. virginica*
6	7.7	3	6.1	2.3	*I. virginica*
7	7.6	3	6.6	2.1	*I. virginica*
8	7.4	2.8	6.1	1.9	*I. virginica*
9	7.3	2.9	6.3	1.8	*I. virginica*
10	7.2	3.6	6.1	2.5	*I. virginica*
11	7.2	3.2	6	1.8	*I. virginica*
12	7.2	3	5.8	1.6	*I. virginica*
13	7.1	3	5.9	2.1	*I. virginica*
14	7	3.2	4.7	1.4	*I. versicolor*
15	6.9	3.1	4.9	1.5	*I. versicolor*
16	6.9	3.2	5.7	2.3	*I. virginica*
17	6.9	3.1	5.4	2.1	*I. virginica*
18	6.9	3.1	5.1	2.3	*I. virginica*
19	6.8	2.8	4.8	1.4	*I. versicolor*
20	6.8	3	5.5	2.1	*I. virginica*
21	6.8	3.2	5.9	2.3	*I. virginica*
22	6.7	3.1	4.4	1.4	*I. versicolor*

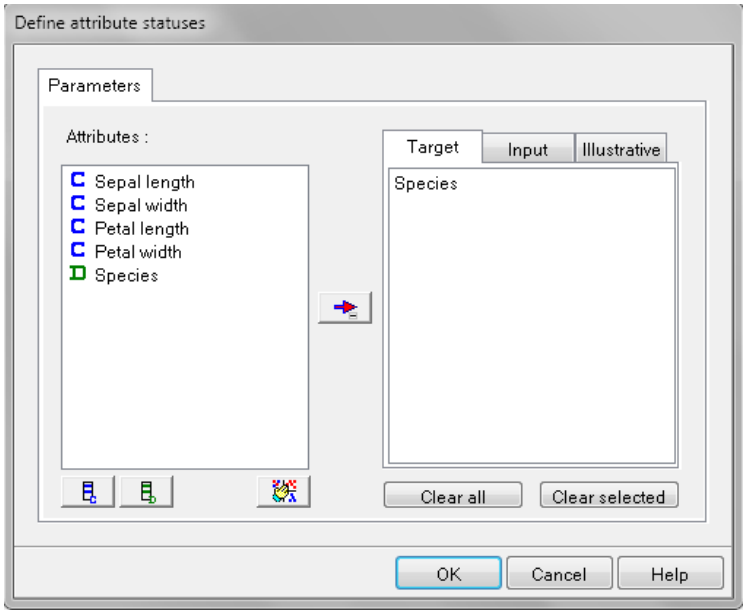
Ensuite, nous l'ouvrons dans Tanagra selon la méthode habituelle:



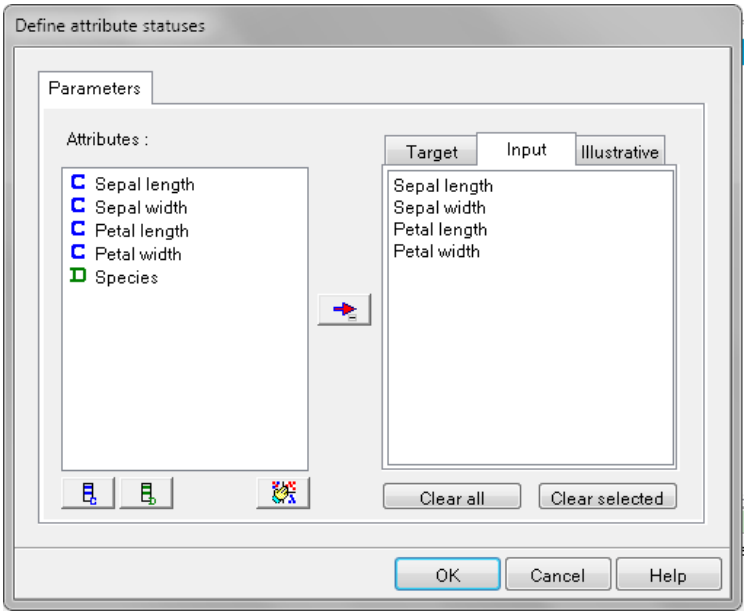
Ensuite, nous ajoutons le sélecteur *Define Status*:



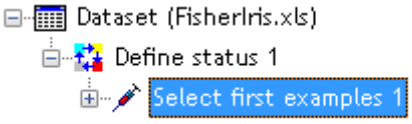
avec en *Target*:



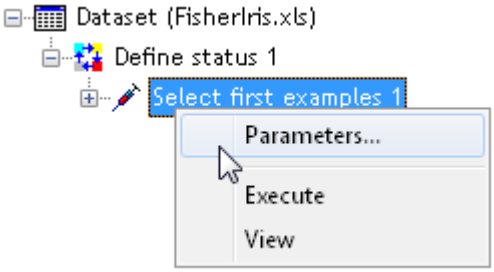
et en *Input*:



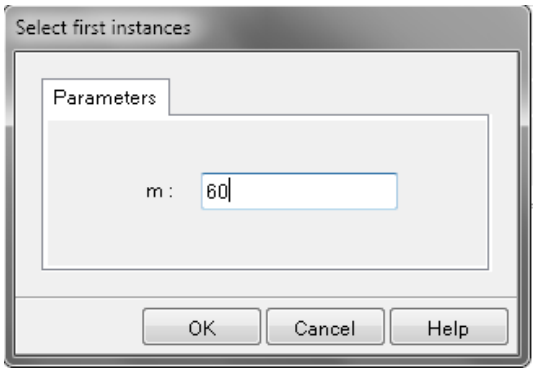
Ensuite nous rajoutons le sélecteur *Select first examples* du groupe *Instance selection*:



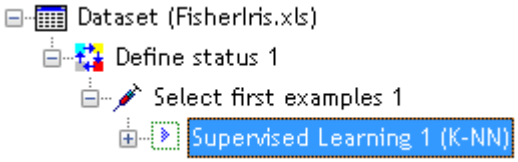
et dans les paramètres du sélecteur:



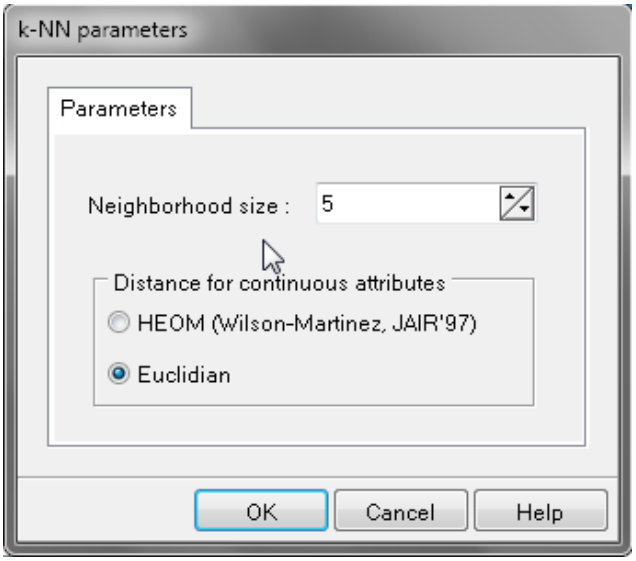
Nous prenons les 60 premières lignes du fichier comme données d'entraînement (choix un peu arbitraire):



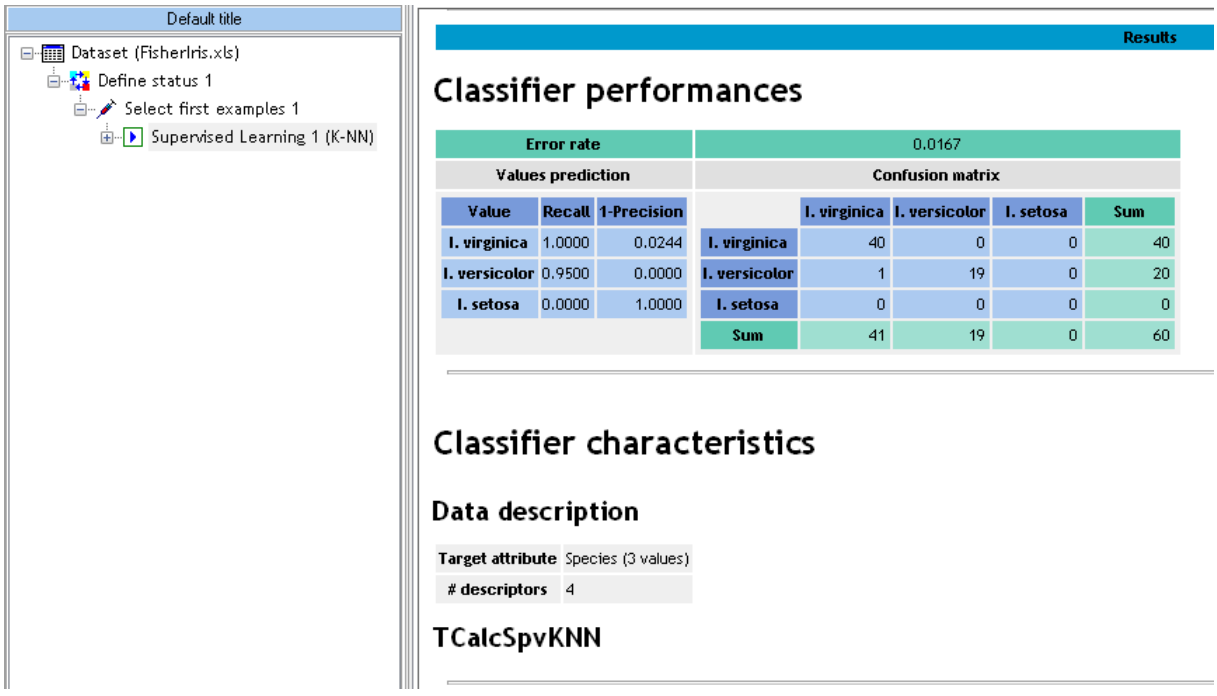
Ensuite, nous rajoutons l'opérateur *K-NN* du groupe *Spv Learning*:



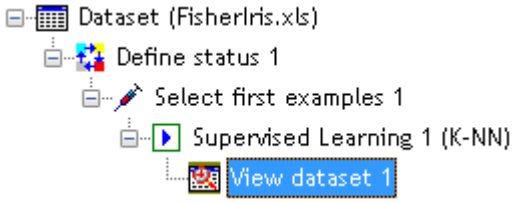
Ensuite, nous choisissons le type de distance et le nombre de *k* voisins pour l'apprentissage:



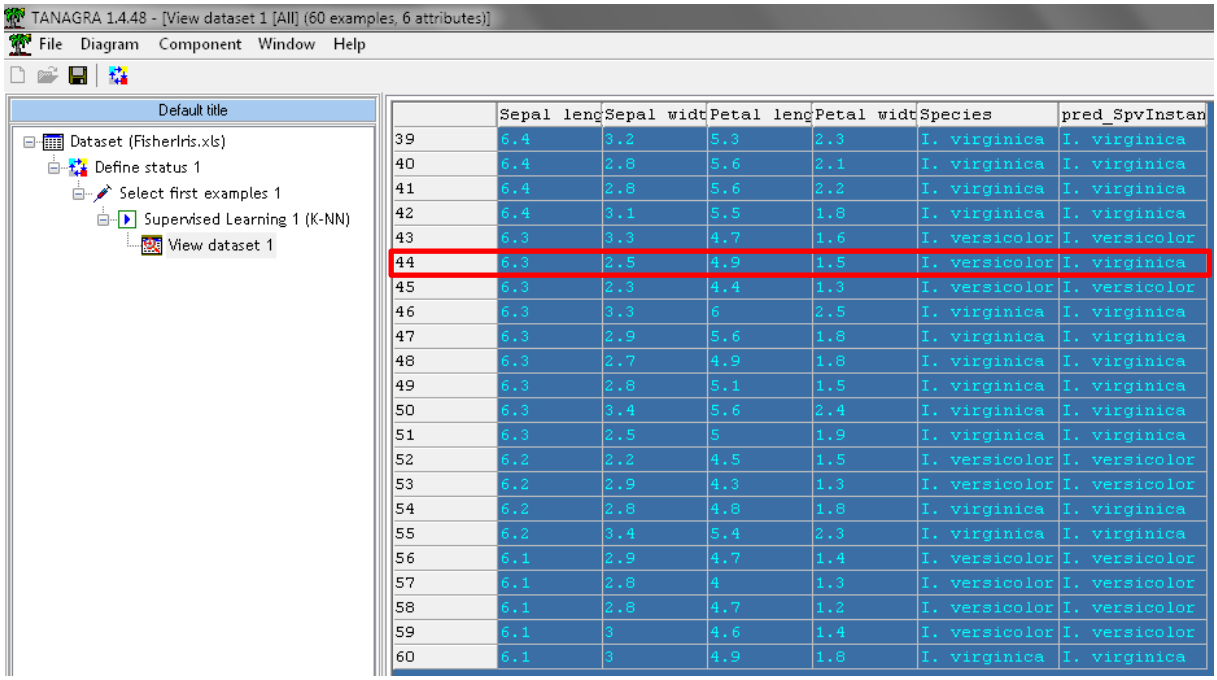
Nous exécutons l'opérateur et nous avons alors:



Nous voyons que le classificateur est très bon. Pour avoir le détail, nous ajoutons l'opérateur *View Data Set* du groupe *Data visualization*:

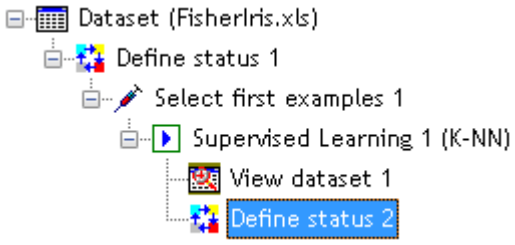


et nous l'exécutons pour avoir les détails des prédictions (nous avons mis en évidence l'un deux ceux qui est mal prédit):

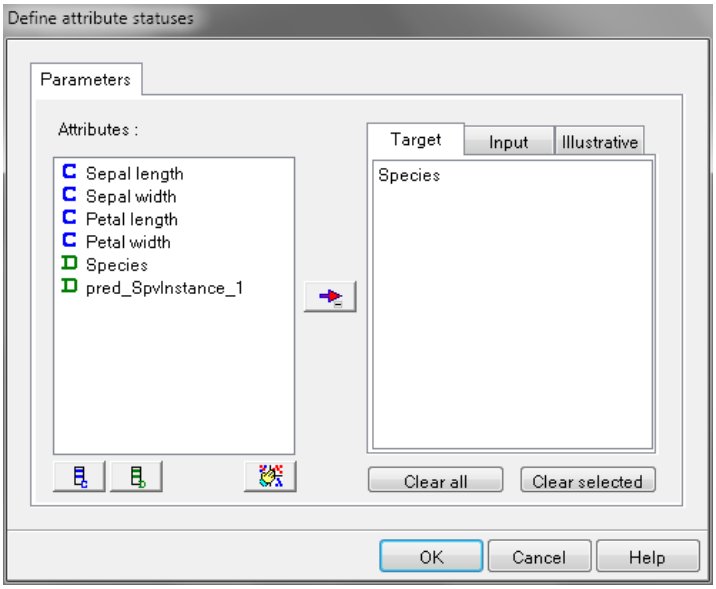


	Sepal leng	Sepal widt	Petal leng	Petal widt	Species	pred_SpvInstan
39	6.4	3.2	5.3	2.3	I. virginica	I. virginica
40	6.4	2.8	5.6	2.1	I. virginica	I. virginica
41	6.4	2.8	5.6	2.2	I. virginica	I. virginica
42	6.4	3.1	5.5	1.8	I. virginica	I. virginica
43	6.3	3.3	4.7	1.6	I. versicolor	I. versicolor
44	6.3	2.5	4.9	1.5	I. versicolor	I. virginica
45	6.3	2.3	4.4	1.3	I. versicolor	I. versicolor
46	6.3	3.3	6	2.5	I. virginica	I. virginica
47	6.3	2.9	5.6	1.8	I. virginica	I. virginica
48	6.3	2.7	4.9	1.8	I. virginica	I. virginica
49	6.3	2.8	5.1	1.5	I. virginica	I. virginica
50	6.3	3.4	5.6	2.4	I. virginica	I. virginica
51	6.3	2.5	5	1.9	I. virginica	I. virginica
52	6.2	2.2	4.5	1.5	I. versicolor	I. versicolor
53	6.2	2.9	4.3	1.3	I. versicolor	I. versicolor
54	6.2	2.8	4.8	1.8	I. virginica	I. virginica
55	6.2	3.4	5.4	2.3	I. virginica	I. virginica
56	6.1	2.9	4.7	1.4	I. versicolor	I. versicolor
57	6.1	2.8	4	1.3	I. versicolor	I. versicolor
58	6.1	2.8	4.7	1.2	I. versicolor	I. versicolor
59	6.1	3	4.6	1.4	I. versicolor	I. versicolor
60	6.1	3	4.9	1.8	I. virginica	I. virginica

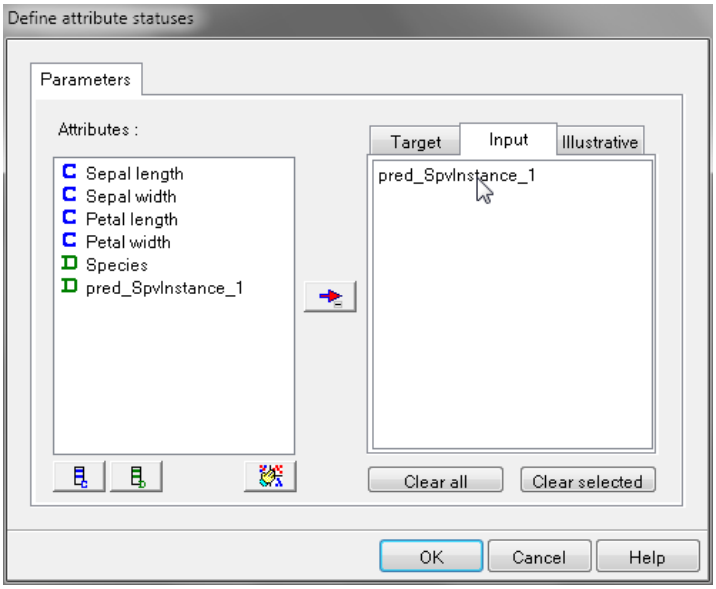
Maintenant injectons pour y mettre un jeu de test, nous remettons un opérateur de sélection *Define Status*:



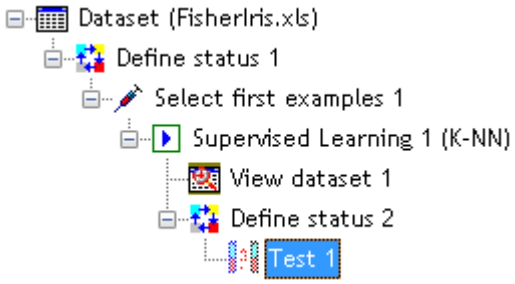
avec en *Target*:



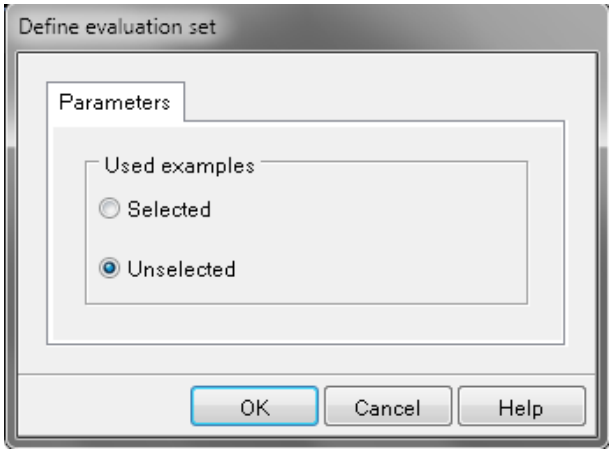
et en *Input*:



Ensuite, nous rajoutons l'opérateur *Test* du groupe *Spv learning assessment* (nous aurions pu faire la même chose pour la régression logistique mais ayant l'équation explicite c'était moins utile alors que là c'est très utile!):

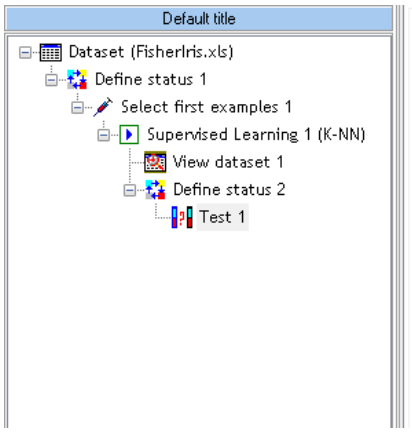


et dans ses paramètres, nous avons:



Nous prenons *Unselected* ce qui prendra les 150-60=90 données restantes.

Et nous exécutons pour obtenir:



Test 1

Parameters

Evaluation set : **unselected** examples

Results

pred_SpvInstance_1

Error rate	0.5778

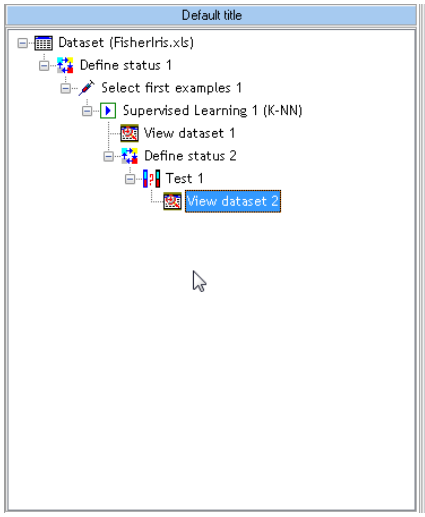
Values prediction

Value	Recall	1-Precision
I. virginica	0.9000	0.1000
I. versicolor	0.9667	0.6375
I. setosa	0.0000	1.0000

Confusion matrix

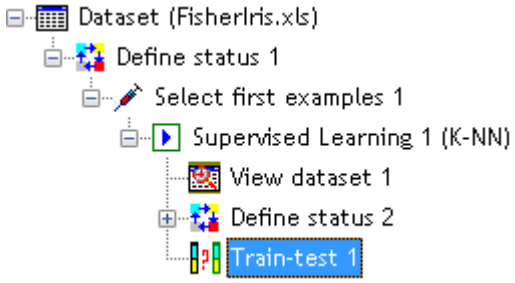
	I. virginica	I. versicolor	I. setosa	Sum
I. virginica	9	1	0	10
I. versicolor	1	29	0	30
I. setosa	0	50	0	50
Sum	10	80	0	90

Et nous rajoutons un composant *View Dataset* pour voir comment les données de test (ou données nouvelles) sont classées:

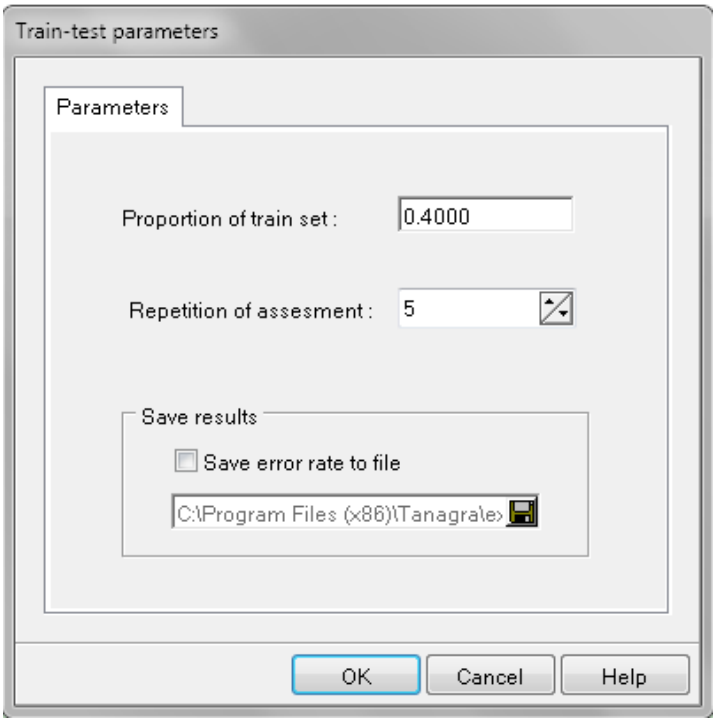


	Sepal leng	Sepal widt	Petal leng	Petal widt	Species	pred_SpvInstance_1
1	7,9	3,8	6,4	2	I. virginica	I. virginica
2	7,7	3,8	6,7	2,2	I. virginica	I. virginica
3	7,7	2,6	6,9	2,3	I. virginica	I. virginica
4	7,7	2,8	6,7	2	I. virginica	I. virginica
5	7,7	3	6,1	2,3	I. virginica	I. virginica
6	7,6	3	6,6	2,1	I. virginica	I. virginica
7	7,4	2,8	6,1	1,9	I. virginica	I. virginica
8	7,3	2,9	6,3	1,8	I. virginica	I. virginica
9	7,2	3,6	6,1	2,5	I. virginica	I. virginica
10	7,2	3,2	6	1,8	I. virginica	I. virginica
11	7,2	3	5,8	1,6	I. virginica	I. virginica
12	7,1	3	5,9	2,1	I. virginica	I. virginica
13	7	3,2	4,7	1,4	I. versicolor	I. versicolor
14	6,9	3,1	4,9	1,5	I. versicolor	I. versicolor
15	6,9	3,2	5,7	2,3	I. virginica	I. virginica
16	6,9	3,1	5,4	2,1	I. virginica	I. virginica
17	6,9	3,1	5,1	2,3	I. virginica	I. virginica
18	6,8	2,8	4,8	1,4	I. versicolor	I. versicolor
19	6,8	3	5,5	2,1	I. virginica	I. virginica
20	6,8	3,2	5,9	2,3	I. virginica	I. virginica
21	6,7	3,1	4,4	1,4	I. versicolor	I. versicolor
22	6,7	3	5	1,7	I. versicolor	I. versicolor

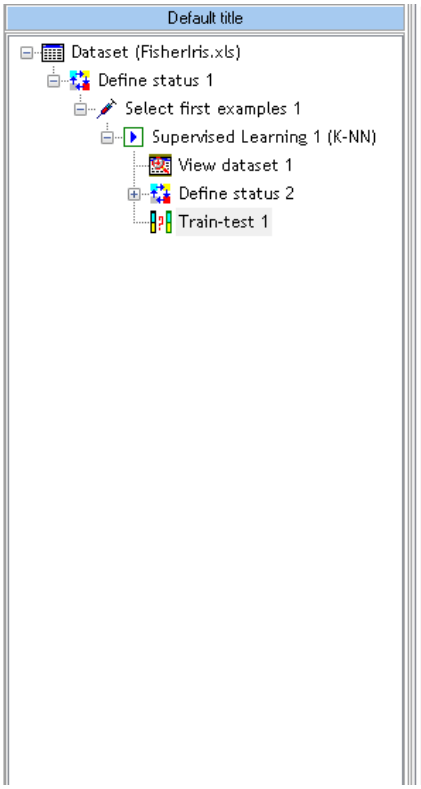
Nous pouvons aussi rajouter un opérateur *Train Test* du groupe *Spv learning assessment*:



et dans les paramètres de cet opérateur, nous prenons:



et en exécutant l'opérateur, nous obtenons:



Train-test 1

Parameters

Train-test parameters	
Train proportion	0.40
Trials	5

Results

Dataset size : 150

Tests error rate

Trial	Train size	Test size	Error rate
1	60	90	0.0667
2	60	90	0.0333
3	60	90	0.0222
4	60	90	0.0222
5	60	90	0.0556

Overral test error rate

Error rate	0.0400

Values prediction

Value	Recall	1-Precision
I. virginica	0.9463	0.0662
I. versicolor	0.9329	0.0544
I. setosa	1.0000	0.0000

Confusion matrix

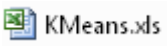
	I. virginica	I. versicolor	I. setosa	Sum
I. virginica	141	8	0	149
I. versicolor	10	139	0	149
I. setosa	0	0	152	152
Sum	151	147	152	450

Exercice 20.: Classificaiton K-Means (nuée dynamique)

Tanagra V1.4.44

Nous allons ici vérifier la technique de clustering que nous avons étudié dans le cours théorique de Méthodes Numériques avec MS Excel et Minitab pour voir si nous retrouvons les mêmes résultats.

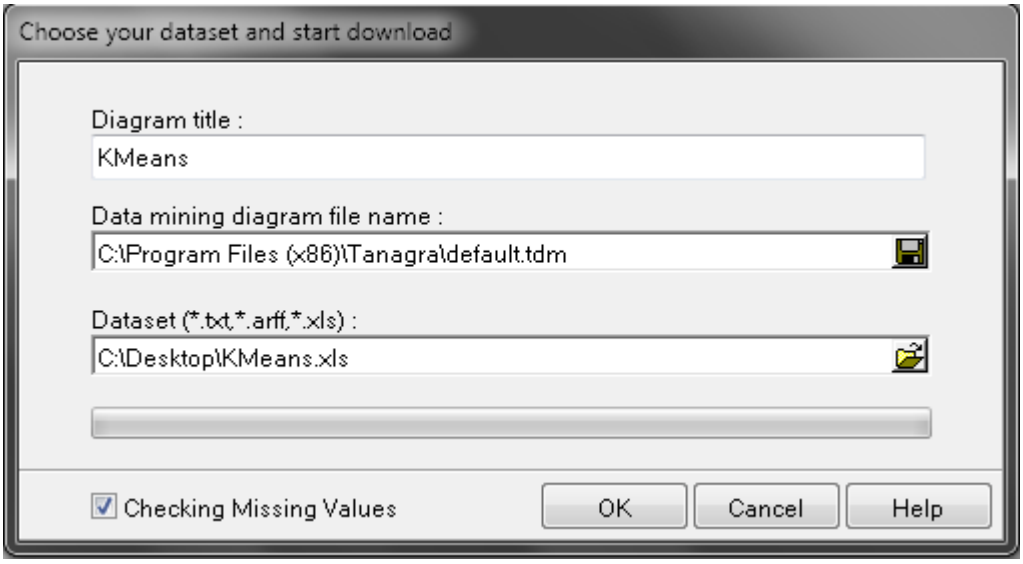
D'abord ouvrez le fichier:



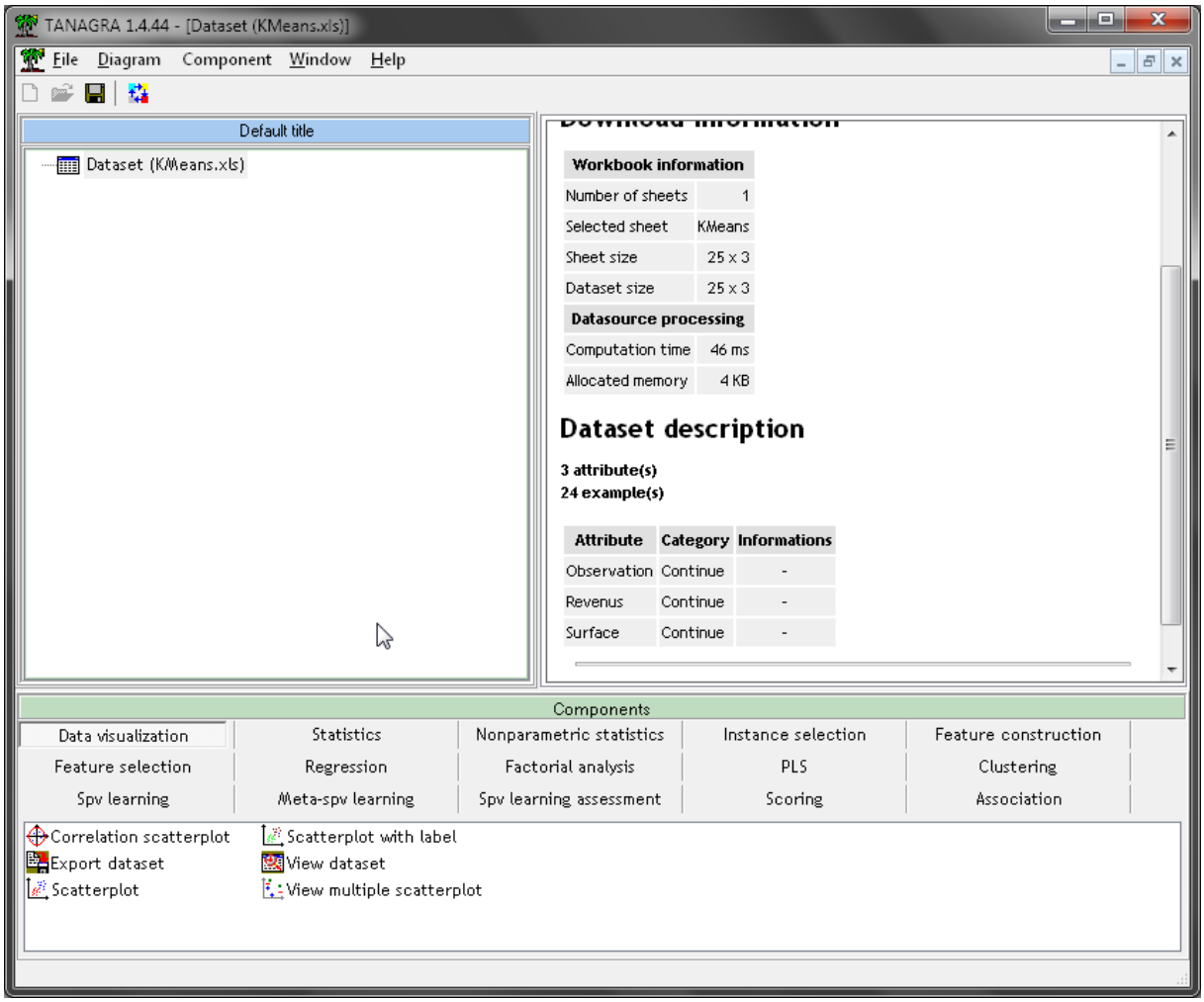
pour vérifier qu'il contient bien les données utilisées lors du cours théorique:

	A	B	C
1	Observation	Revenus	Surface
2	1	60	18.4
3	2	85.5	16.8
4	3	64.8	21.6
5	4	61.5	20.8
6	5	87	23.6
7	6	110.1	19.2
8	7	108	17.6
9	8	82.8	22.4
10	9	69	20
11	10	93	20.8
12	11	51	22
13	12	81	20
14	13	75	19.6
15	14	52.8	20.8
16	15	64.8	17.2
17	16	43.2	20.4
18	17	84	17.6
19	18	49.2	17.6
20	19	59.4	16
21	20	66	18.4
22	21	47.4	16.4
23	22	33	18.8
24	23	51	14
25	24	63	14.8

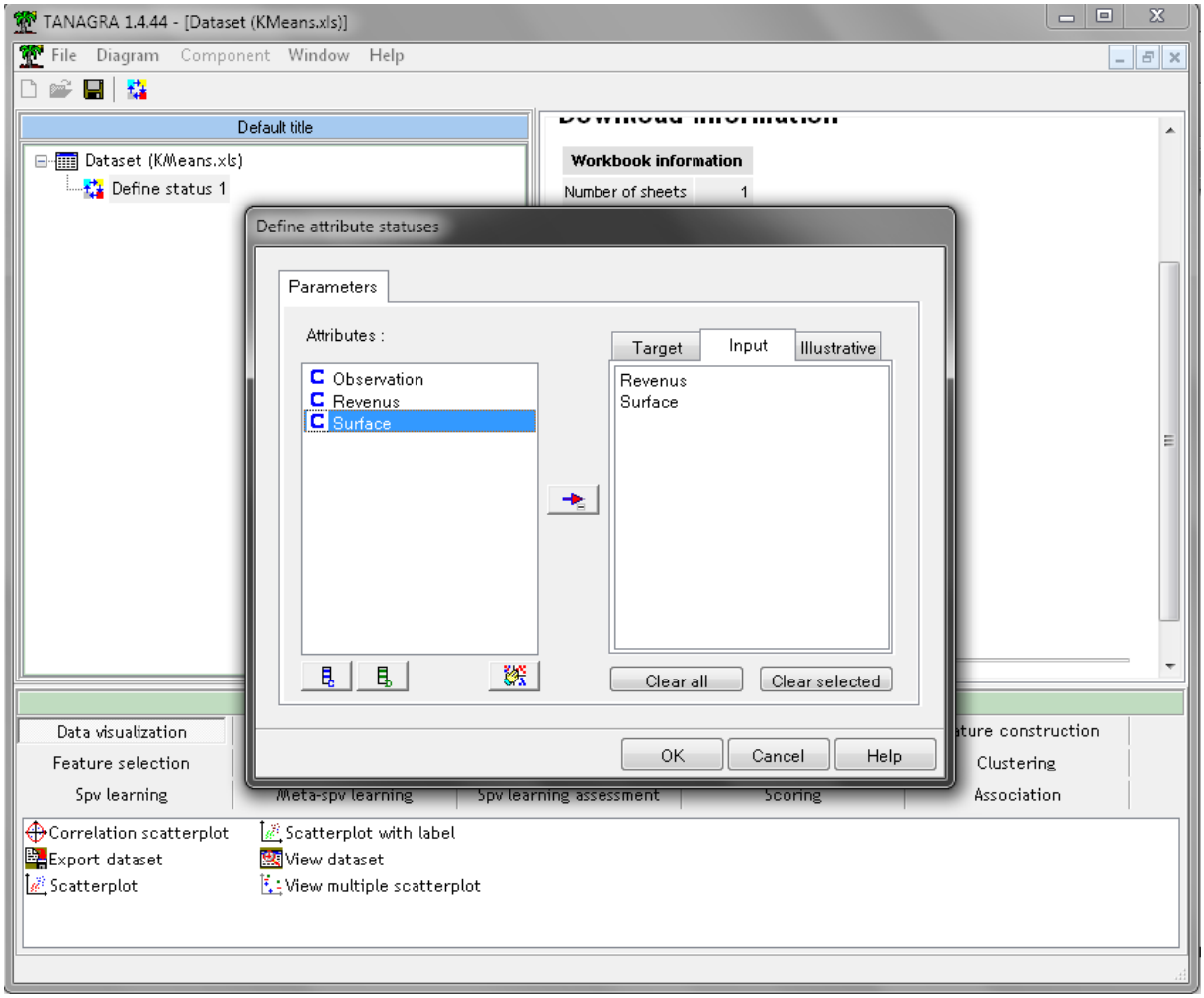
Ensuite, nous ouvrons Tanagra et création un nouveau projet basé sur ce fichier MS Excel:



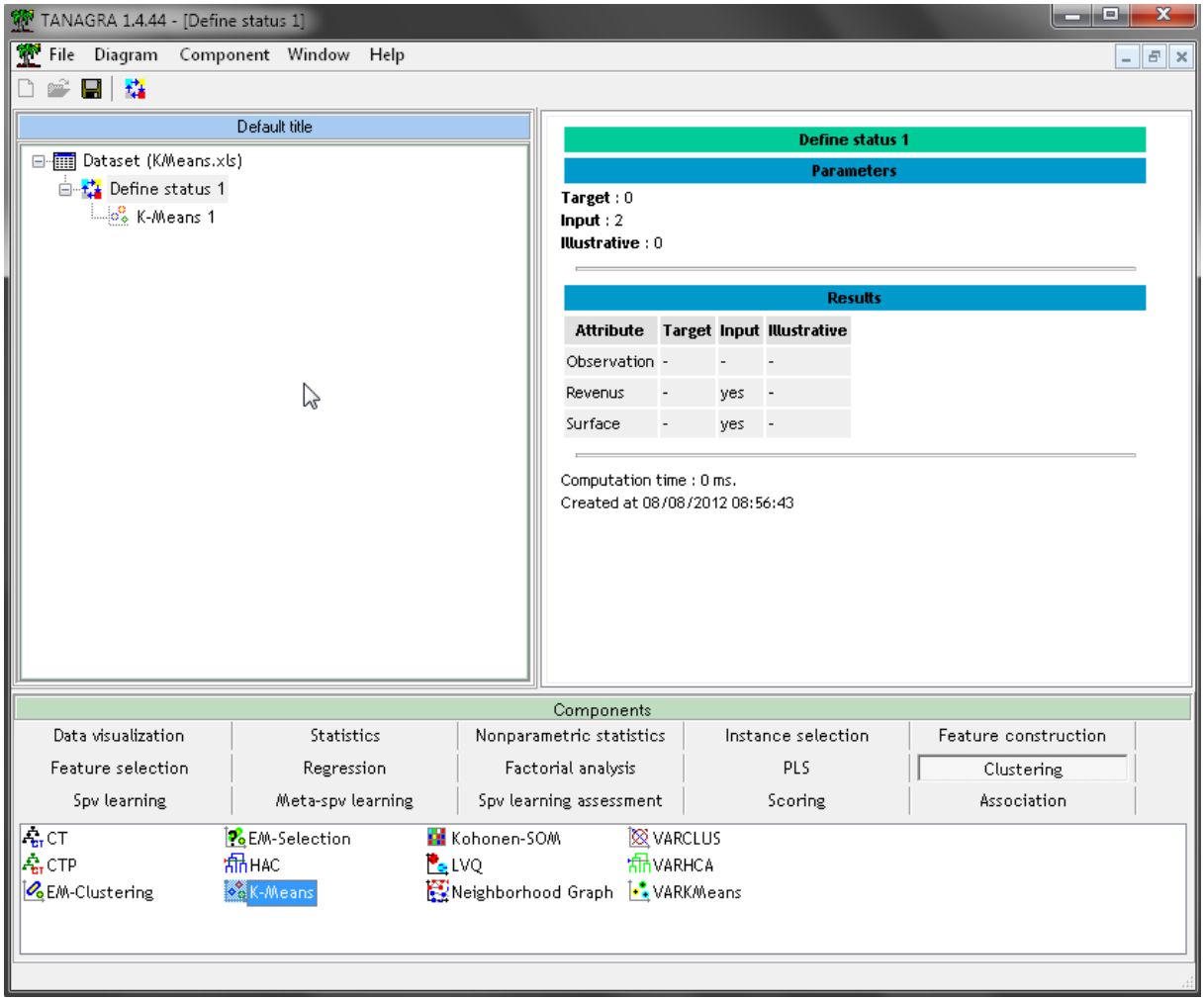
Ce qui donnera:



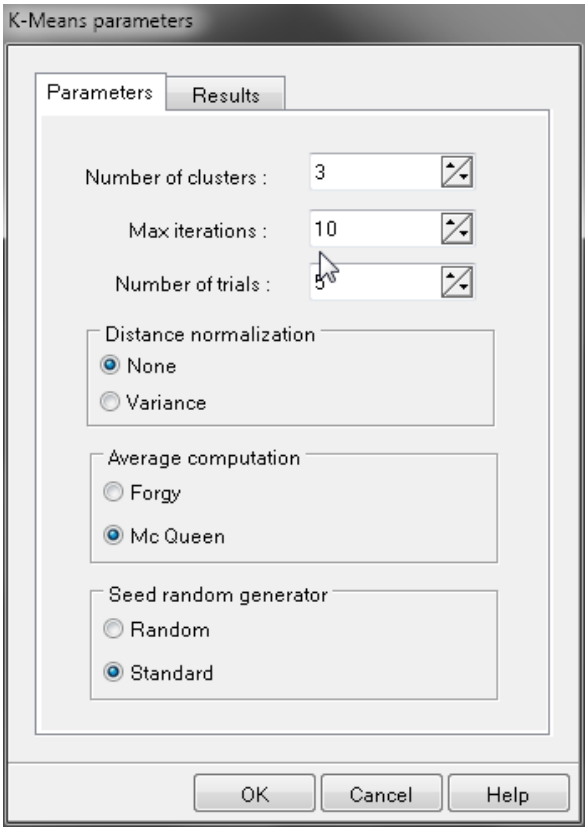
Nous voulons faire un K-Means sur les revenus et la Surface donc nous prenons le sélecteur **Define status** où nous mettons en **Input** les deux variables à clusteriser:



Ensuite, nous rajoutons le composant **K-Means** du groupe **Clustering**:



et dans les paramètres du composant nous mettons:



Nous exécutons et visualisons le composant et obtenons:

K-Means 1

Parameters

K-Means parameters	
Clusters	3
Max Iteration	10
Trials	5
Distance normalization	none
Average computation	McQueen
Seed random generator	Standard

Results

Global evaluation

Within Sum of Squares	1517,0291
Total Sum of Squares	9146,2960
R-Square	0,8341

Cluster size and WSS

Clusters	3		
Cluster	**Description**	**Size**	**WSS**
cluster n°1	c_kmeans_1	9	229,4400
cluster n°2	c_kmeans_2	8	960,1149
cluster n°3	c_kmeans_3	7	327,4743

R-Square for each attempt

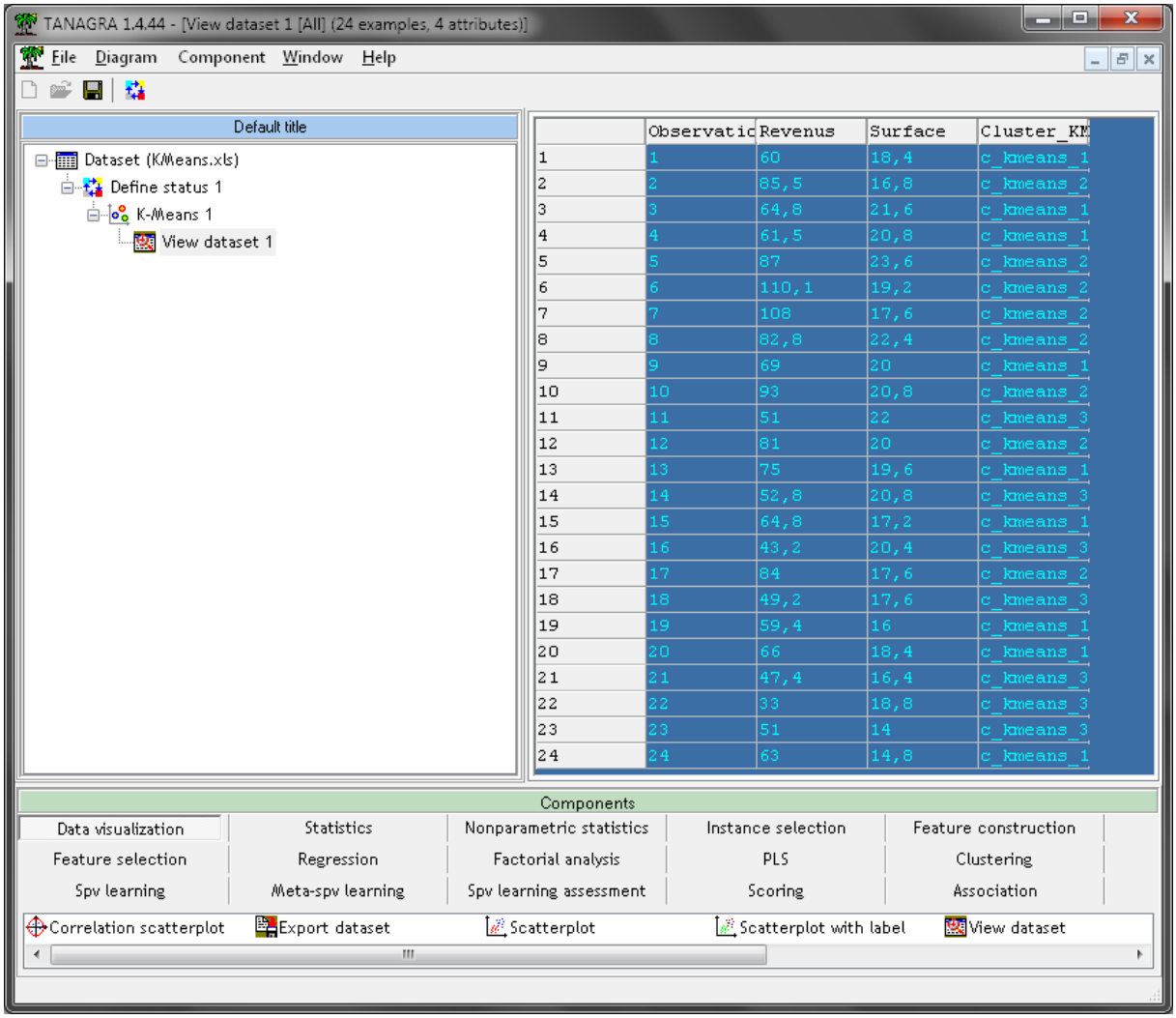
Number of trials	5
Trial	**R-square**
1	0,812924
2	0,795216
3	0,807683
4	0,834137
5	0,834137

Cluster centroids

Attribute	Cluster n°1	Cluster n°2	Cluster n°3
Revenus	64,833334	91,425000	46,800000
Surface	18,533333	19,750000	18,571428

Use GROUP CHARACTERIZATION for detailed comparisons

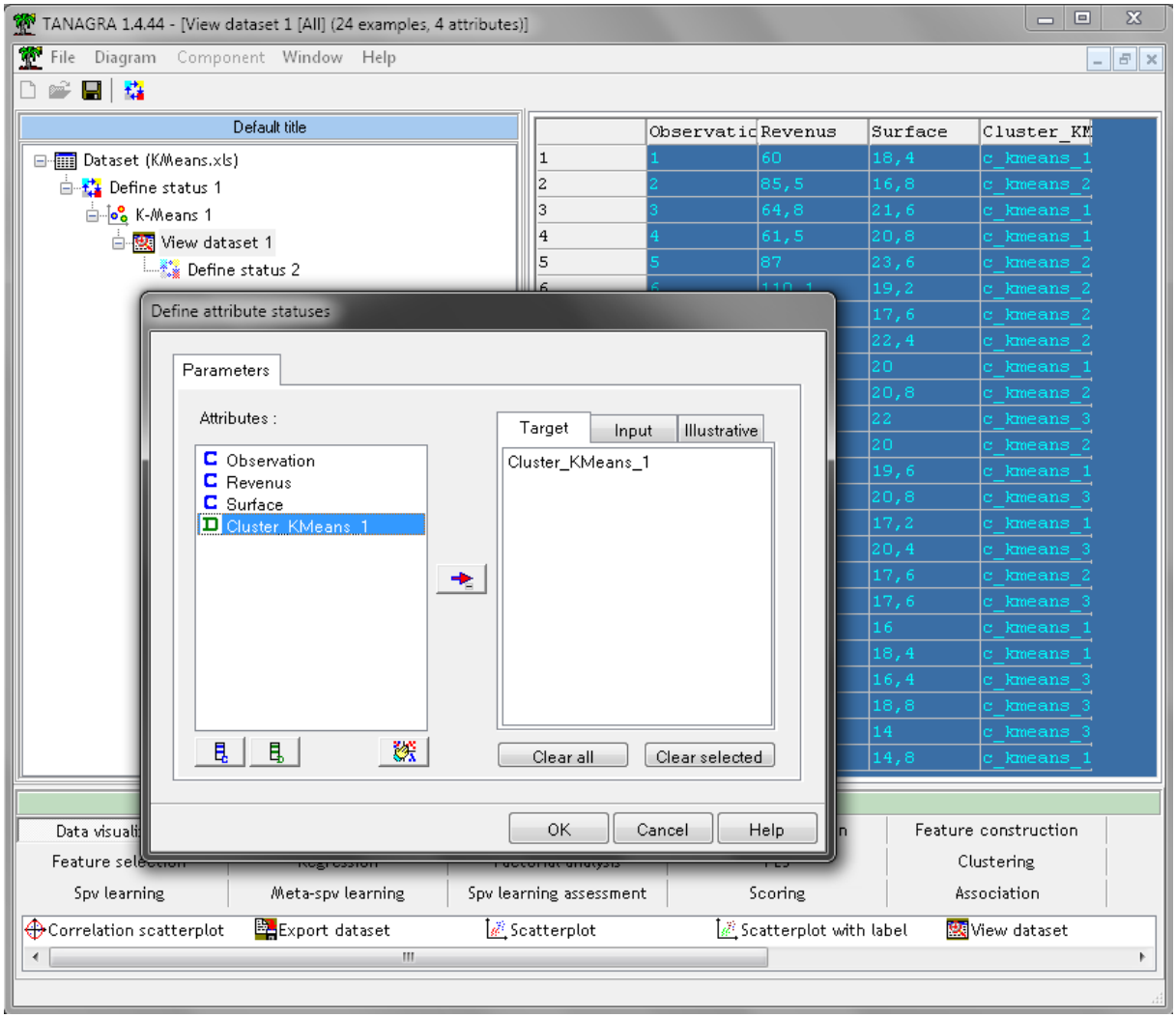
Et nous retrouvons bien les résultats obtenus avec MS Excel et Minitab. Cependant nous souhaiterions un peu plus de détails avec Minitab. Pour cela, nous rajoutons un composant **View dataset** que nous exécutons et visualisons:



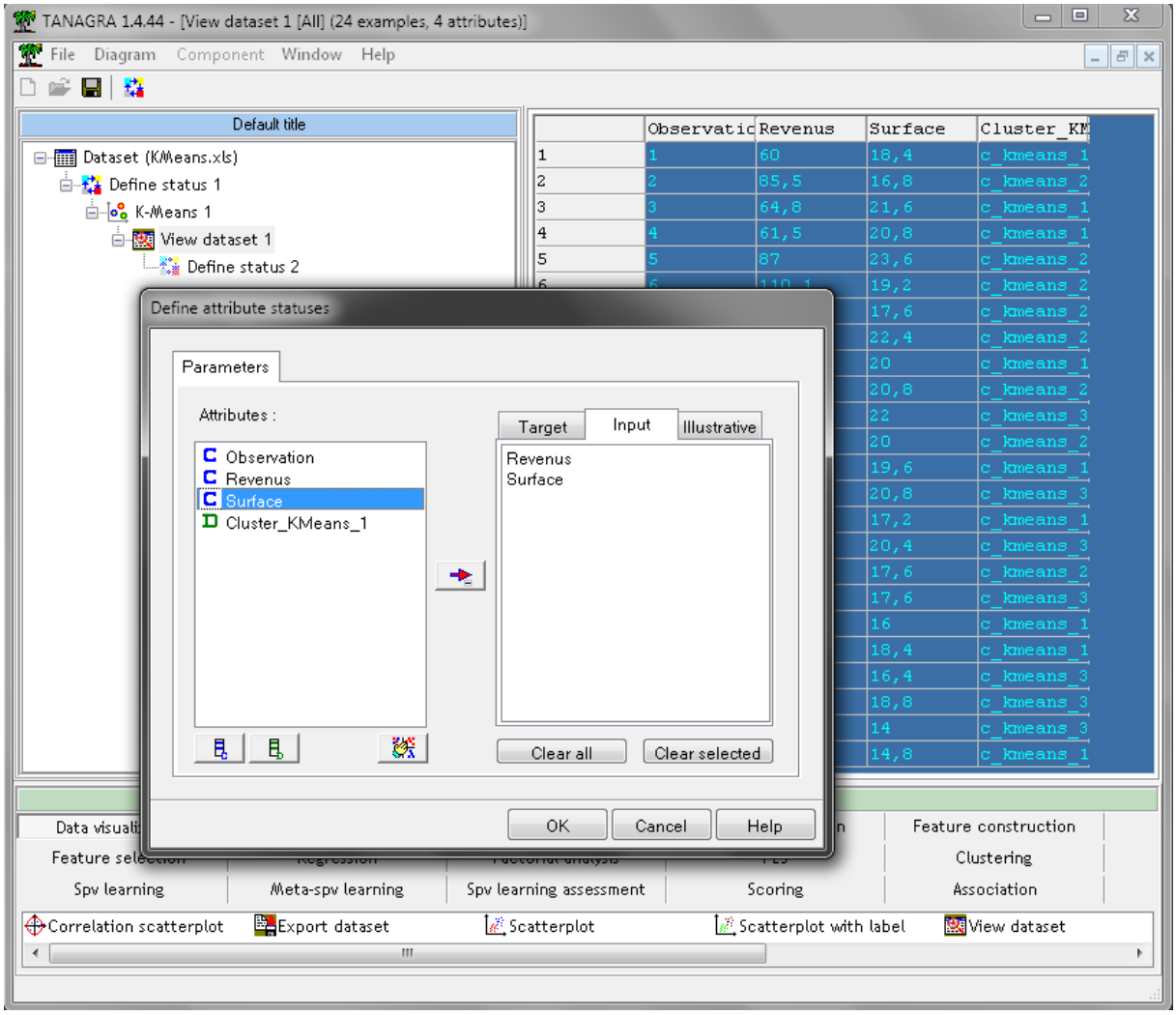
	Observatio	Revenus	Surface	Cluster_KM
1	1	60	18,4	c_kmeans_1
2	2	85,5	16,8	c_kmeans_2
3	3	64,8	21,6	c_kmeans_1
4	4	61,5	20,8	c_kmeans_1
5	5	87	23,6	c_kmeans_2
6	6	110,1	19,2	c_kmeans_2
7	7	108	17,6	c_kmeans_2
8	8	82,8	22,4	c_kmeans_2
9	9	69	20	c_kmeans_1
10	10	93	20,8	c_kmeans_2
11	11	51	22	c_kmeans_3
12	12	81	20	c_kmeans_2
13	13	75	19,6	c_kmeans_1
14	14	52,8	20,8	c_kmeans_3
15	15	64,8	17,2	c_kmeans_1
16	16	43,2	20,4	c_kmeans_3
17	17	84	17,6	c_kmeans_2
18	18	49,2	17,6	c_kmeans_3
19	19	59,4	16	c_kmeans_1
20	20	66	18,4	c_kmeans_1
21	21	47,4	16,4	c_kmeans_3
22	22	33	18,8	c_kmeans_3
23	23	51	14	c_kmeans_3
24	24	63	14,8	c_kmeans_1

Nous avons alors sur la droite exactement le même tableau que celui obtenu avec MS Excel ou Minitab pour montrer quels individus appartiennent à quel Cluster.

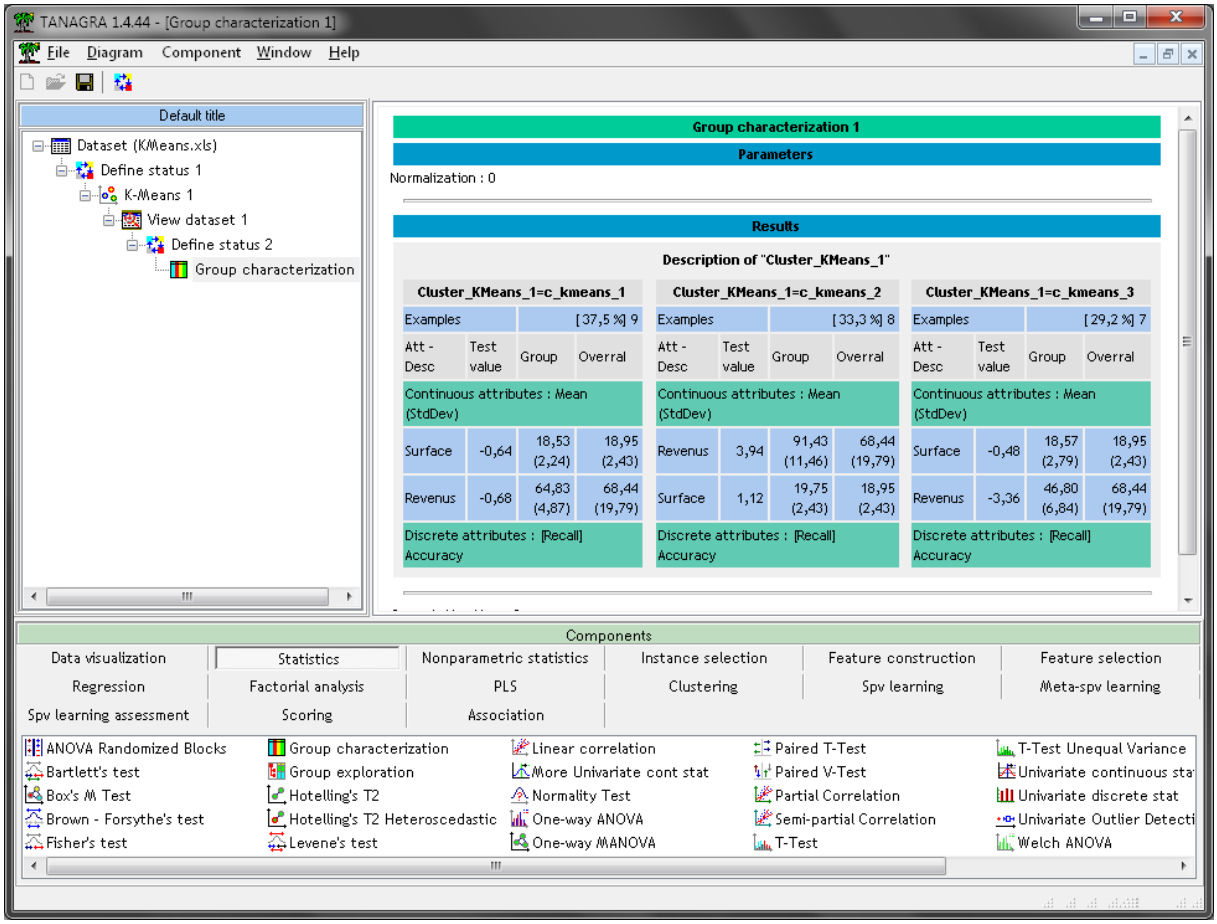
Maintenant regardons les caractéristiques de groupes (c'est à partir de Maintenant que le logiciel est bien plus efficace que les autres). Nous ajoutons un composant **Define status** avec en **Target** les clusters:



et en **Input** les variables:

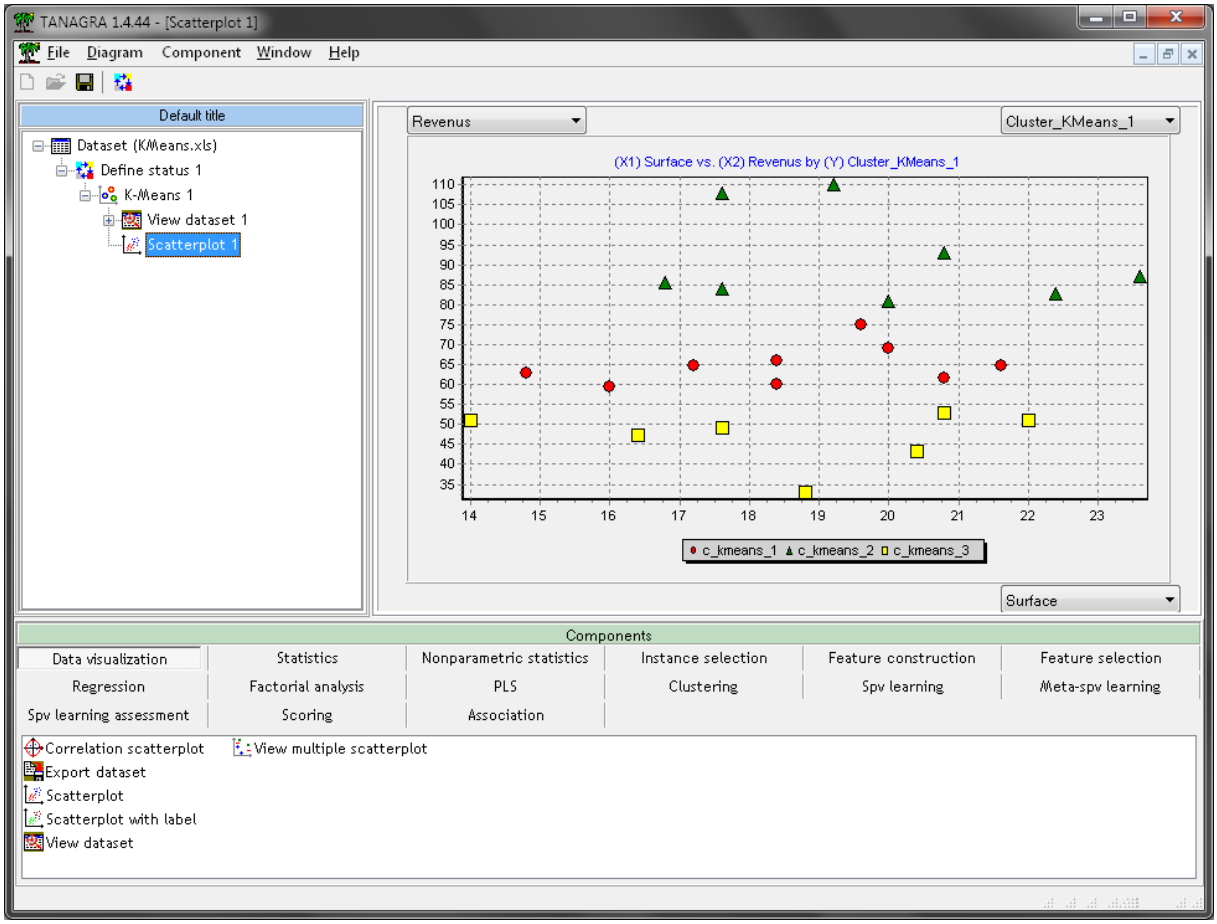


Et nous y ajoutons le composant **Group characterization**:

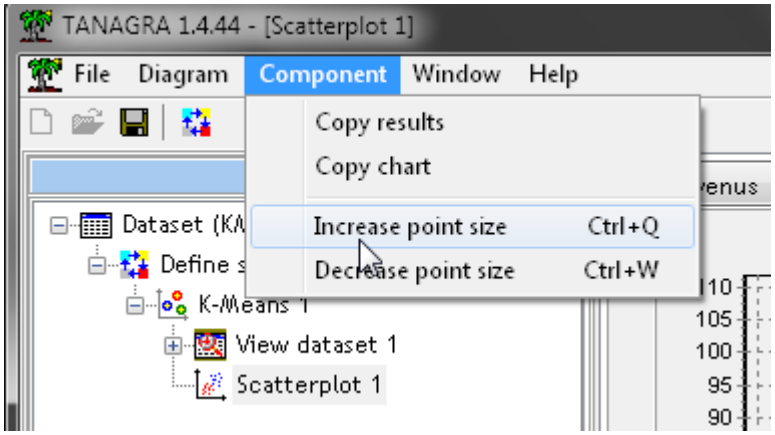


Au vu des résultats, nous nous rendons compte qu'il aurait été peut-être plus malin de laisser la colonne *Propriétaire* dans le fichier d'origine afin d'avoir une caractérisation utilisant ce group pouvant peut-être aider à la conclusion...

Pour finir, ajoutons un opérateur **Scatterplot**:



et nous voyons bien comment sont composés les 3 clusters. Si jamais pour grossir les points il faut aller dans le menu **Component**:



Exercice 21.: Clustering ID-3 (Iterative Dichotomiser 3)

Tanagra V1.4.48

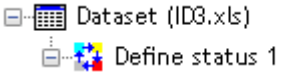
Nous allons ici vérifier la technique de clustering ID-3 que nous avons étudié dans le cours théorique de Méthodes Numériques et calculé à la main.

Nous allons donc travailler avec le fichier suivant et donc avec les mêmes données que dans le cours théorique:

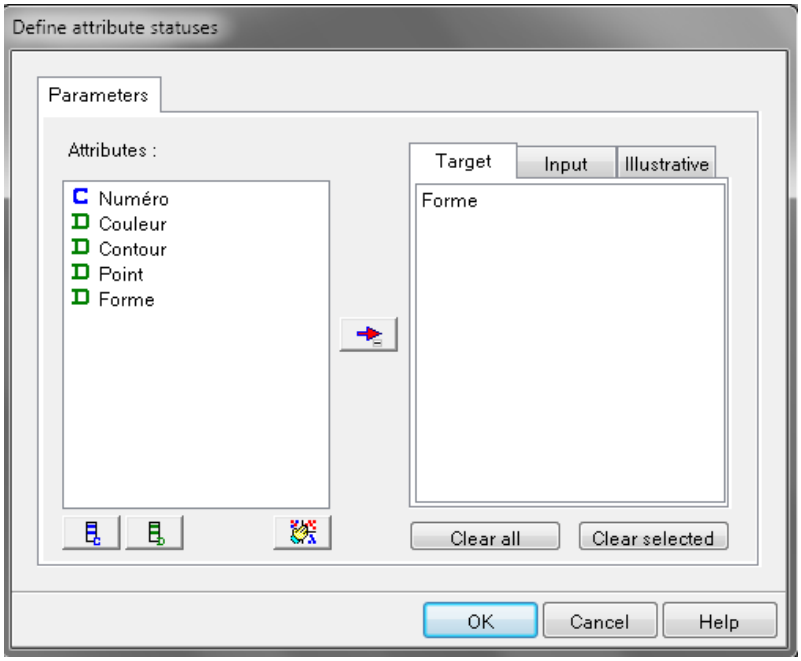
	A	B	C	D	E
1	**Numéro**	**Couleur**	**Contour**	**Point**	**Forme**
2	1	Vert	Pointillé	Non	Triangle
3	2	Vert	Pointillé	Oui	Triangle
4	3	Jaune	Pointillé	Non	Carré
5	4	Rouge	Pointillé	Non	Carré
6	5	Rouge	Plein	Non	Carré
7	6	Rouge	Plein	Oui	Triangle
8	7	Vert	Plein	Non	Carré
9	8	Vert	Pointillé	Non	Triangle
10	9	Jaune	Plein	Oui	Carré
11	10	Rouge	Plein	Non	Carré
12	11	Vert	Plein	Oui	Carré
13	12	Jaune	Pointillé	Oui	Carré
14	13	Jaune	Plein	Non	Carré
15	14	Rouge	Pointillé	Oui	Triangle

Nous importons cette liste comme à l'habitude dans Tanagra (la méthode étant toujours la même).

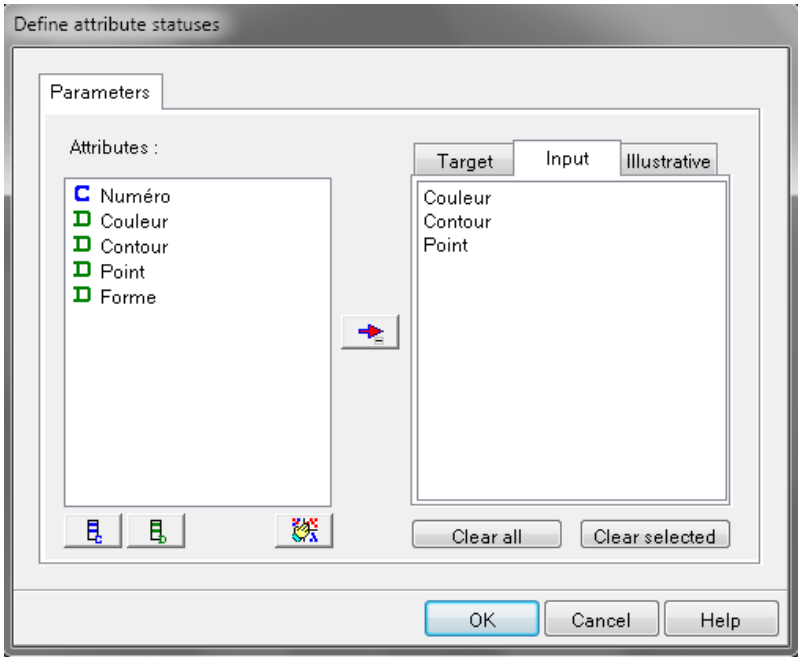
Nous mettons le sélecteur **Define status**:



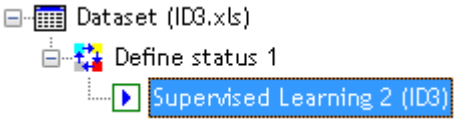
avec comme **Target** la colonne *Formes* (car c'est ce que nous voulons deviner):



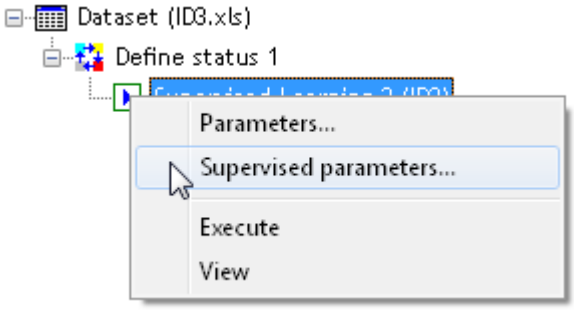
et comme **Input** les trois autres champs (peut importe l'ordre d'insertion):



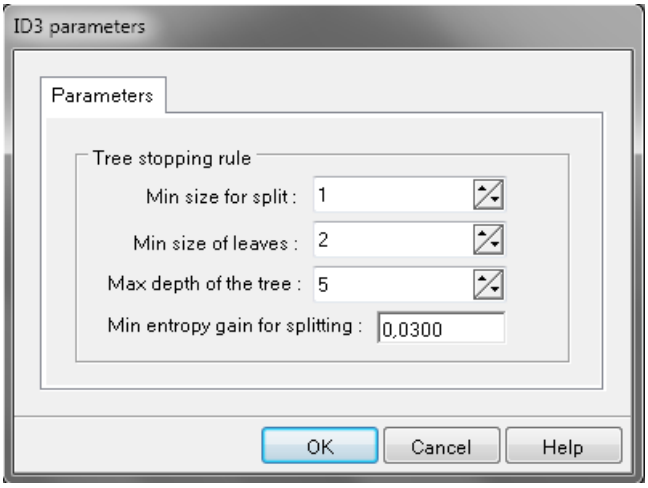
Ensuite, nous ajoutons l'opérateur **ID3** du groupe **SPV Learning**:



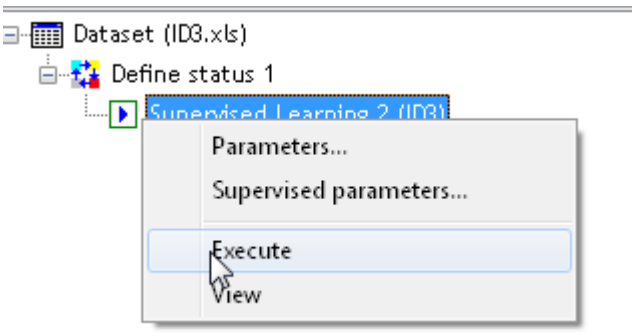
Nous allons dans les options ***Supervised parameters...***:



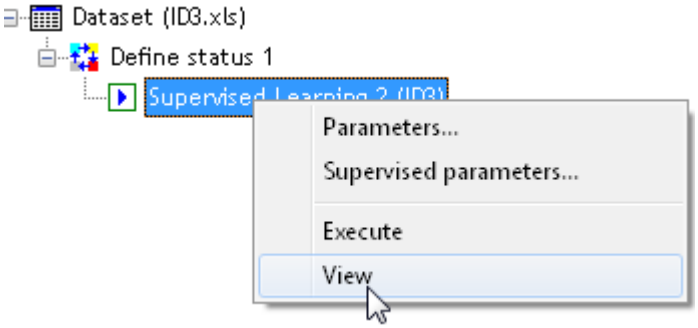
pour mettre:



Ensuite, nous exécutons le composant en cliquant sur **Excecute** comme à l'habitude:



et en faisons un **View**:



Pour obtenir exactement les résultats correspondant à ceux calculés à la main:

Supervised Learning 2 (ID3)

Parameters

ID3 parameters	
Size before split	1
Size after split	2
Max depth of leaves	5
Goodness of split thresold	0,0300

Results

Classifier performances

Error rate	0,0000

Values prediction

Value	Recall	1-Precision
Triangle	1,0000	0,0000
Carré	1,0000	0,0000

Confusion matrix

	Triangle	Carré	Sum
Triangle	5	0	5
Carré	0	9	9
Sum	5	9	14

Classifier characteristics

Data description

Target attribute	Forme (2 values)
# descriptors	3

Tree description

Number of nodes	8
Number of leaves	5

Decision tree

- Couleur in [Vert]
  - Contour in [Pointillé] then Forme = **Triangle** (100,00 % of 3 examples)
  - Contour in [Plein] then Forme = **Carré** (100,00 % of 2 examples)
- Couleur in [Jaune] then Forme = **Carré** (100,00 % of 4 examples)
- Couleur in [Rouge]
  - Point in [Non] then Forme = **Carré** (100,00 % of 3 examples)
  - Point in [Oui] then Forme = **Triangle** (100,00 % of 2 examples)

Dommage qu'il n'y ait pas de diagramme cependant... cela aiderait à la compréhension.

Exercice 22.: HAC (Hierarchical Ascendant Clustering)

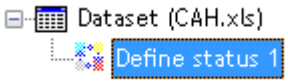
Tanagra V1.4.48

Nous allons ici vérifier la technique de clustering HAC que nous avons étudié dans le cours théorique de Méthodes Numériques et calculé à la main.

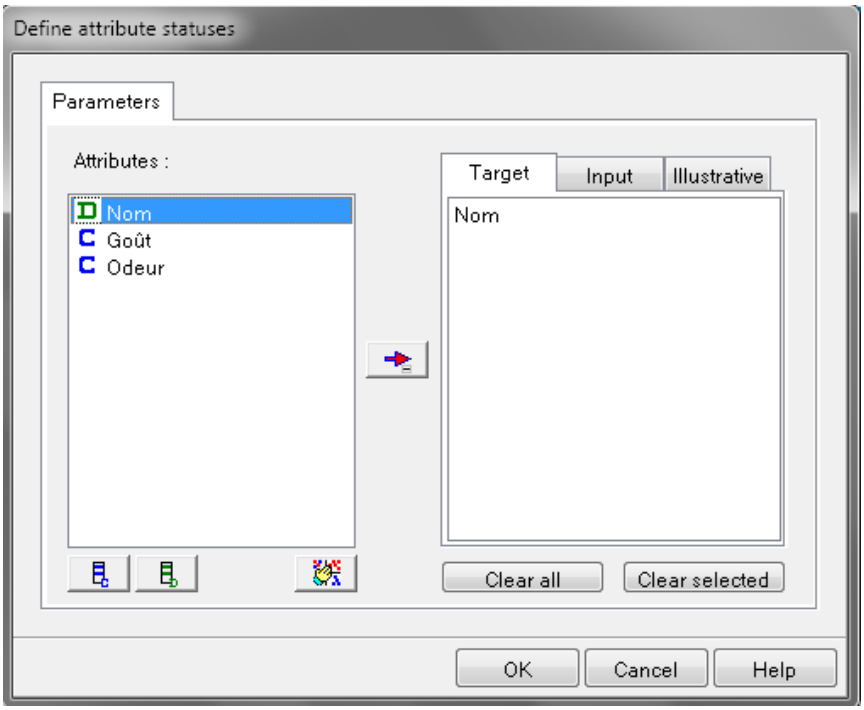
Nous partons de la liste suivante:

	A	B	C
1	**Nom**	**Goût**	**Odeur**
2	A	2	5
3	B	7	8
4	C	3	3
5	D	8	9
6	E	4	5

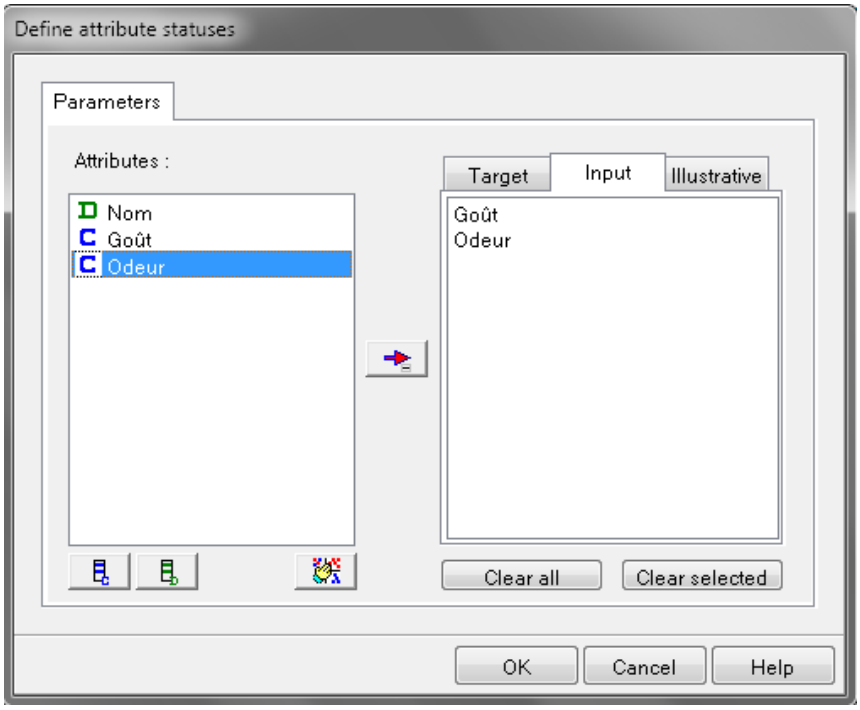
Nous l'importons dans Tanagra comme à l'habitude et lui ajoutons le sélecteur **Define status**:



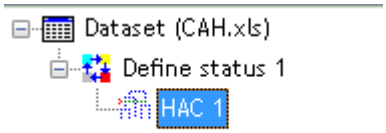
avec le champ *Nom* dans les **Target**:



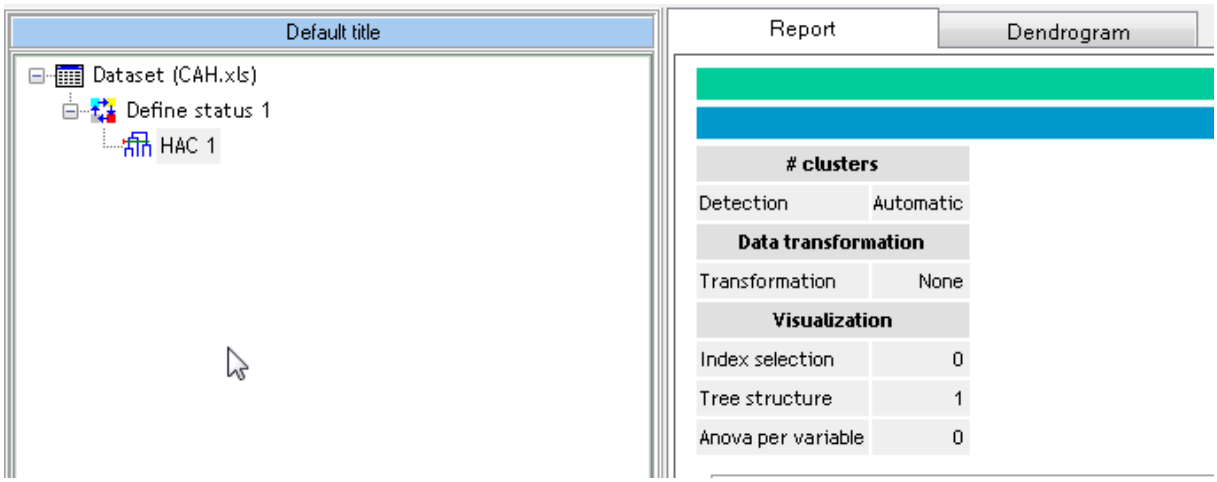
et dans **Input** le reste:



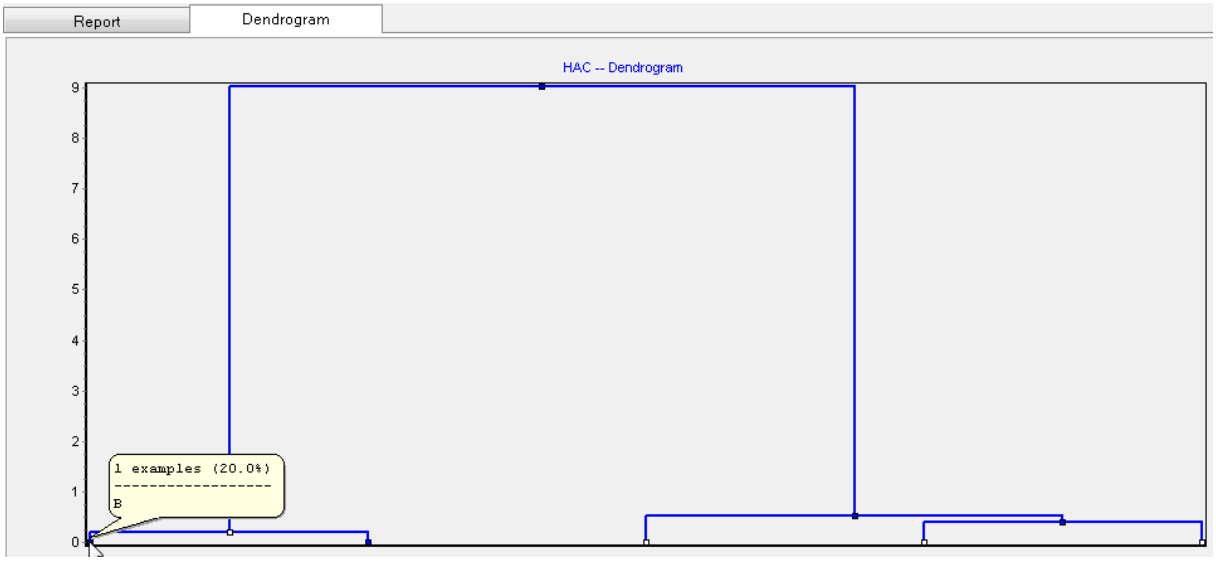
Ensuite, nous ajoutons le composant **HAC** du groupe **Clustering**:



Et nous exécutons le composant pour avoir:



Nous cliquons sur l'onglet **Dendrogram** et apparaît alors le même diagramme que celui obtenu avec les calculs manuels à l'exception des valeurs de l'axe vertical (la différence venant juste d'une convention):



Si l'on reste appuyé avec le bouton gauche de la souris sur chaque point, nous retrouvons les nom des lignes de la liste d'origine.

Exercice 23.: Classification naïve bayésienne

Tanagra V1.4.48

Comme dans le cours théorique, nous partons des données suivantes:

	A	B	C	D	E
1	**Exemple**	**Couleur**	**Type**	**Origine**	**Volé**
2	1	Rouge	Sports	Domestique	Oui
3	2	Rouge	Sports	Domestique	Non
4	3	Rouge	Sports	Domestique	Oui
5	4	Jaune	Sports	Domestique	Non
6	5	Jaune	Sports	Importé	Oui
7	6	Jaune	SUV	Importé	Non
8	7	Jaune	SUV	Importé	Oui
9	8	Jaune	SUV	Domestique	Non
10	9	Rouge	SUV	Importé	Non
11	10	Rouge	Sports	Importé	Oui

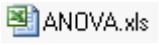
Exceptionnellement nous allons faire l'analyse avec RapidMiner car la sortie de Tanagra n'est pas agréable du tout et l'interprétation pour l'usage pratique peu adaptée.

Donc nous ouvrons RapidMiner:

Exercice 24.: ANOVA à un facteur

Tanagra V1.4.36

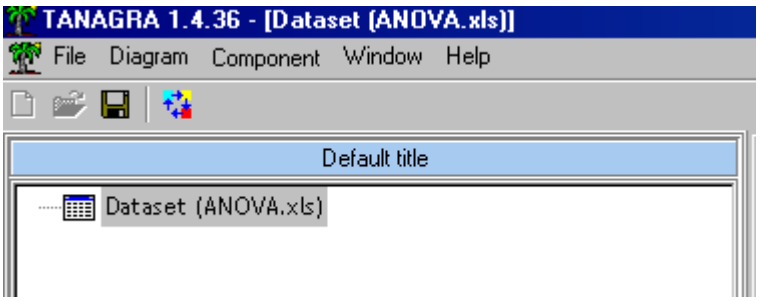
Nous allons prendre ce fichier qui MS Excel qui nous est connu mais qu'il a fallu restructurer pour Tanagra (voir cours sur MS Excel):



contenant:

	A	B
1	**Pièces**	**Equipe**
2	78	Equipe 1
3	88	Equipe 1
4	90	Equipe 1
5	77	Equipe 1
6	85	Equipe 1
7	88	Equipe 1
8	79	Equipe 1
9	77	Equipe 2
10	75	Equipe 2
11	80	Equipe 2
12	83	Equipe 2
13	87	Equipe 2
14	90	Equipe 2
15	85	Equipe 2
16	88	Equipe 3
17	86	Equipe 3
18	79	Equipe 3
19	93	Equipe 3
20	79	Equipe 3
21	83	Equipe 3
22	79	Equipe 3

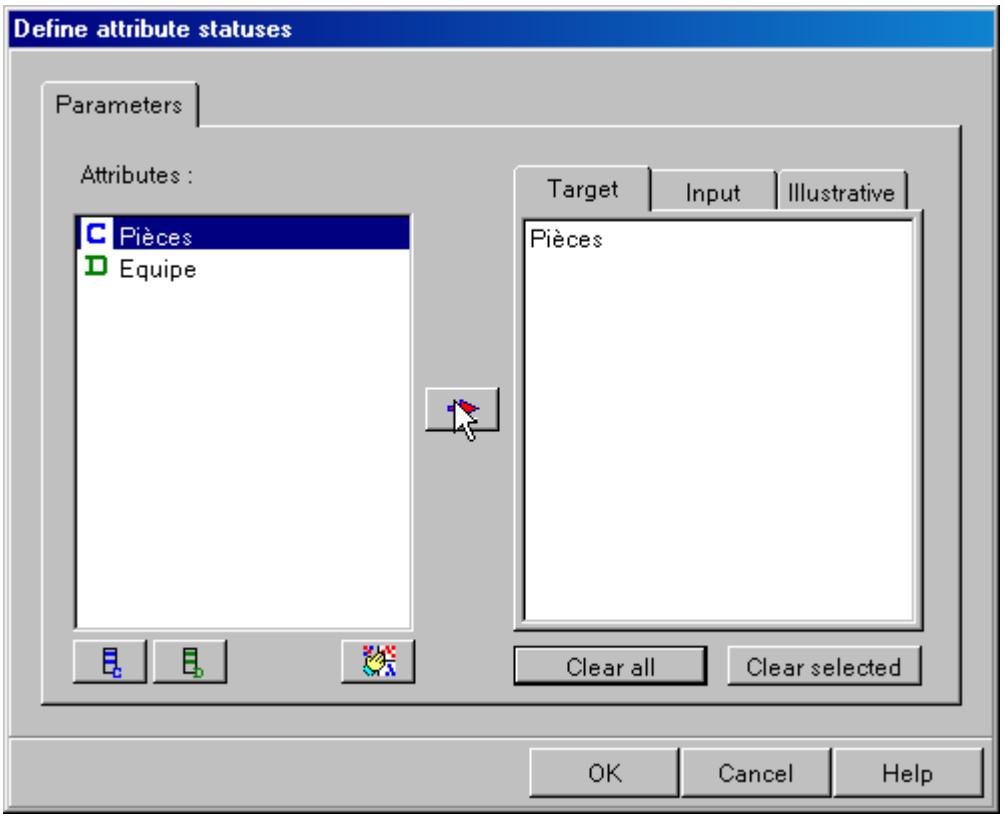
Nous l'importons dans Tanagra en utilisant la même procédure que les exercices précédents:



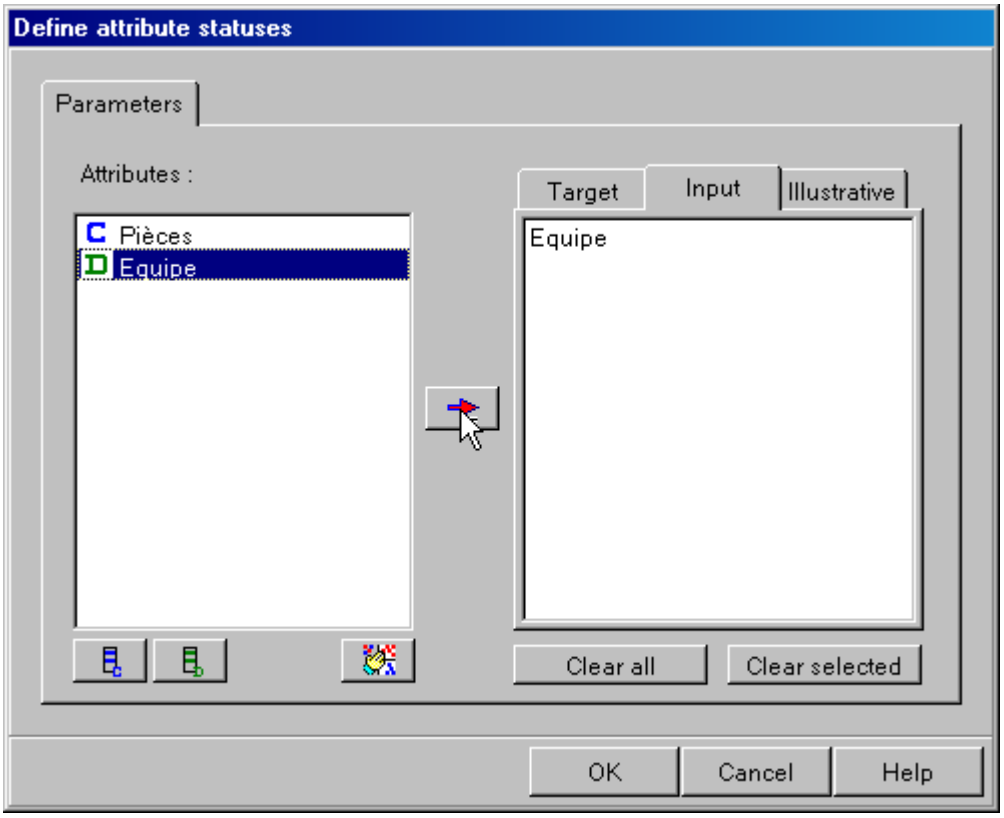
Nous y ajoutons un sélecteur de type **Define status** comme pour les exemples précédents:

Dataset (ANOVA.xls)
Define status 1

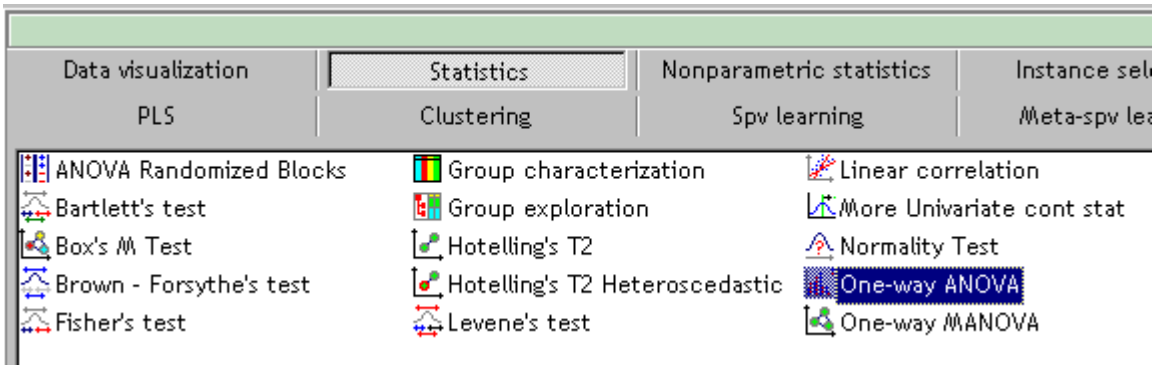
mais avec la variable d'intérêt dans **Target**:



et la variable de classement dans **Input**:



On ajoute ensuite l'opérateur **One-way ANOVA**:



One-way ANOVA

Description
One way analysis of variance: compare the average of continuous TARGET attributes according to groups defined by INPUT attributes. If several TARGET and INPUT attributes are defined, the component computes the ANOVA of each pair of TARGET-INPUT attributes.

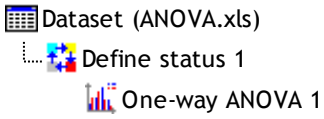
Precondition
At least one or more discrete and continuous attributes must be available together. The TARGET and INPUT attributes must be defined.

Target attribute(s)
The continuous dependent variables(s).

Input attribute(s)
The discrete factor(s).

Postcondition
None.

afin d'avoir:



et on exécute et affichons le résultats comme dans les exemples précédents pour avoir au final:

One-way ANOVA 1

Parameters

Parameters	
Sort results	no

Results

Attribute_Y	Attribute_X	Description				Statistical test		
		Value	Examples	Average	Std-dev	Variance decomposition		
						Source	Sum of square	d.f.
Pièces	Equipe	Equipe 1	7	83.5714	5.4423	BSS	8.0000	2
		Equipe 2	7	82.4286	5.4116	WSS	530.2857	18
		Equipe 3	7	83.8571	5.4292	TSS	538.2857	20
		All	21	83.2857	5.1879	Significance level		
						Statistics	Value	Proba
						Fisher's F	0.135776	0.873924

Computation time : 0 ms.
Created at 30.04.2011 11:03:58

Nous retrouvons exactement les mêmes chiffres que dans les autres cours donc il nous avons les mêmes conclusions.

Exercice 25.: ANOVA de Friedman par les rangs

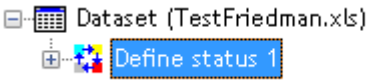
Tanagra V1.4.48

À nouveau le but ici va être de vérifier (comparer) les calculs faits à la main dans le cours théorique ainsi qu'avec Minitab 15.

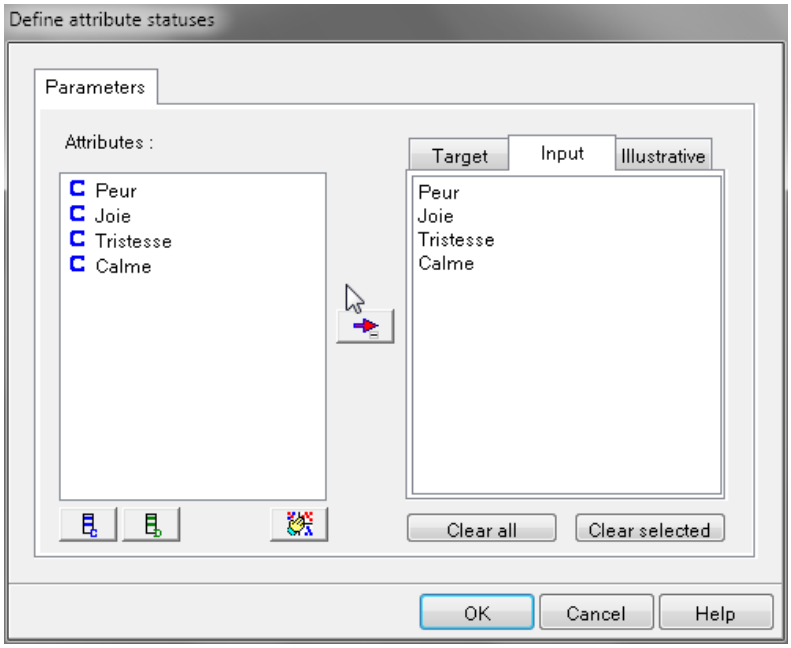
D'abord, nous partons du fichier Excel suivant pour Tanagra (remarquez la structure particulière par rapport à la présentation utilisée dans le cours théorique et Minitab):

	A	B	C	D
1	**Peur**	**Joie**	**Tristesse**	**Calme**
2	23.1	22.7	22.5	22.6
3	57.6	53.2	53.7	53.1
4	10.5	9.7	10.8	8.3
5	23.6	19.6	21.1	21.6
6	11.9	13.8	13.7	13.3
7	54.6	47.1	39.2	37
8	21	13.6	13.7	14.8
9	20.3	23.6	16.3	14.8

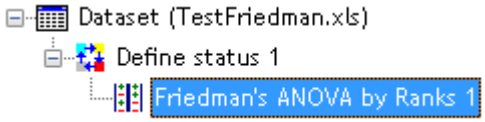
Nous importons comme à l'habitude dans Tanagra et mettons le sélecteur **Define Status**:



et dans les paramètres nous mettons uniquement tous les champs en **Input**:



Ensuite, nous ajoutons le composant **Friedman's ANOVA by rank** sans rien changer ni paramétrer:



Nous exécutons le tout et obtenons:

Friedman's ANOVA by Ranks 1

Parameters

Results

Results

RANKS		
Att.	Sum(Ranks)	Mean(Ranks)
Peur	27,0	3,3750
Joie	20,0	2,5000
Tristesse	19,0	2,3750
Calme	14,0	1,7500

Friedman Statistic	
Stat.	Value
Frideman Fr	6,45000
d.f.	3
p-value	0,09166

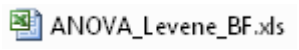
Soit les mêmes valeurs que dans le cours théorique et dans Minitab15.

Exercice 26.: Tests de Levene et Brown-Forsythe

Tanagra V1.4.36

Nous allons ici vérifier si nous retombons sur le même résultat que celui obtenu en cours lors de l'étude théorique et la démonstration mathématique des tests de Levene et de Brown-Forsythe.

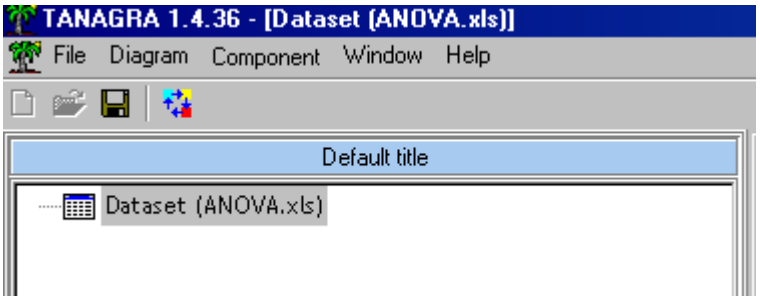
Nous allons prendre ce fichier qui MS Excel qui nous est connu mais qu'il a fallu restructurer pour Tanagra (voir cours sur MS Excel):



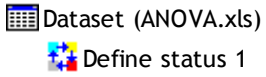
contenant:

	A	B
1	**Pièces**	**Equipe**
2	78	Equipe 1
3	88	Equipe 1
4	90	Equipe 1
5	77	Equipe 1
6	85	Equipe 1
7	88	Equipe 1
8	79	Equipe 1
9	77	Equipe 2
10	75	Equipe 2
11	80	Equipe 2
12	83	Equipe 2
13	87	Equipe 2
14	90	Equipe 2
15	85	Equipe 2

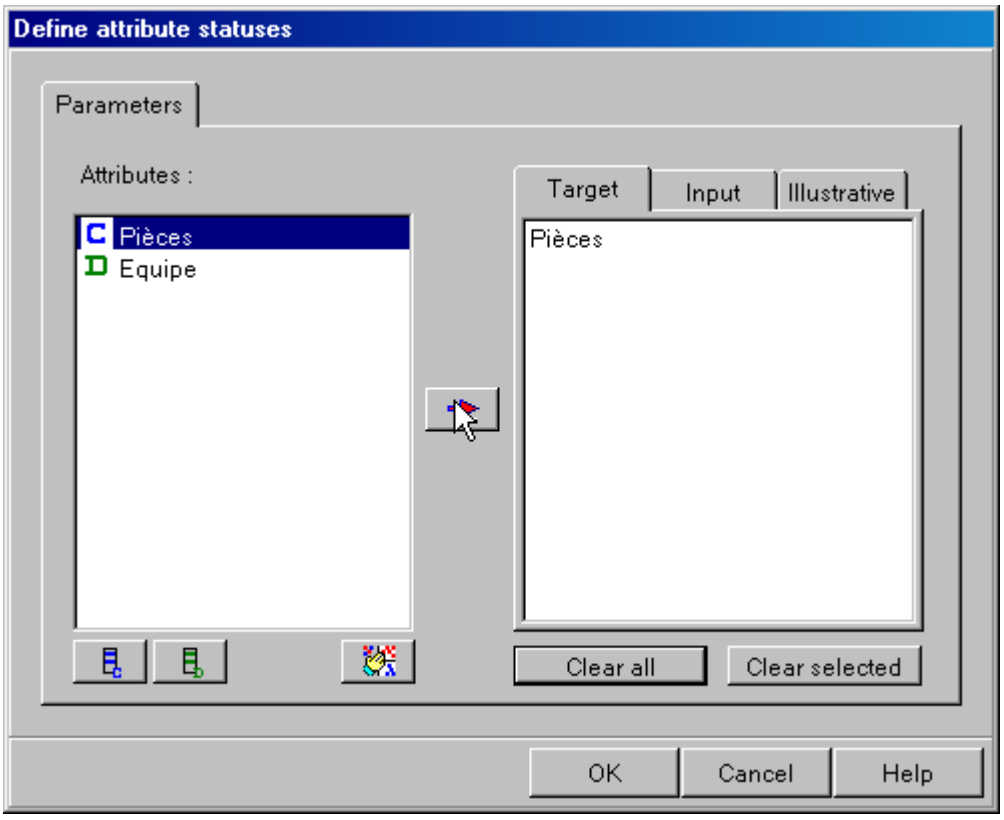
Nous l'importons dans Tanagra en utilisant la même procédure que les exercices précédents:



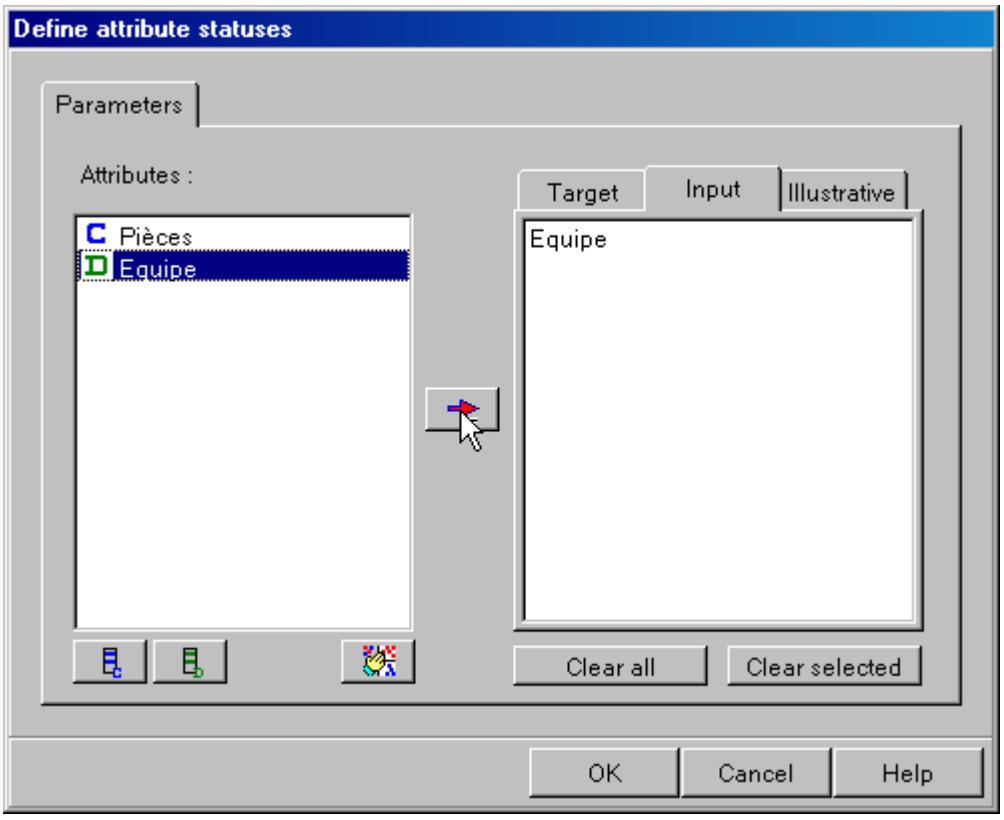
Nous y ajoutons un sélecteur de type **Define status** comme pour les exemples précédents:



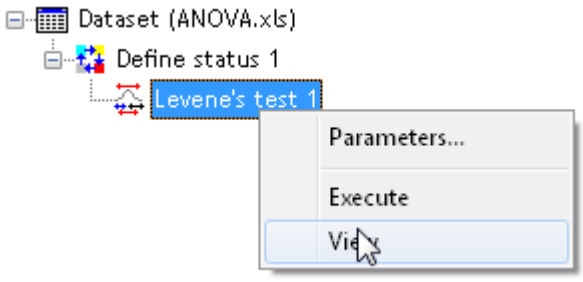
mais avec la variable d'intérêt dans **Target**:



et la variable de classement dans **Input**:



Ensuite, nous ajoutons l'opérateur **Levene's test** du groupe **Statistics** et la visualisons:



pour obtenir:

Levene's test 1

Parameters

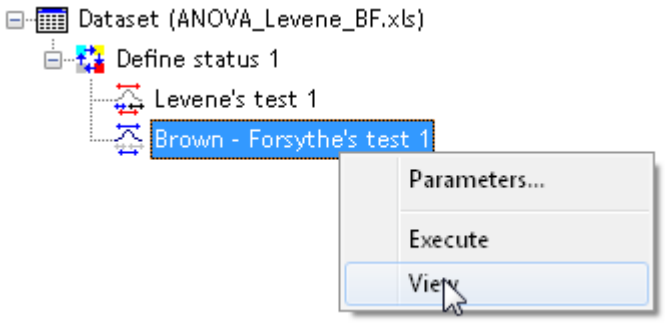
Parameters	
Sort results	no

Results

Attribute_Y	Attribute_X	Description				Statistical test	
		Value	Examples	Average	Std-dev	Test	
Pièces	Equipe	Equipe 1	7	83,5714	5,4423	Levene's W	0,062976
		Equipe 2	7	82,4286	5,4116	df	2/18
		Equipe 3	7	83,8571	5,4292	p-value	0,939172
		All	21	83,2857	5,1879		

Computation time : 0 ms.
Created at 16/08/2013 16:34:39

Nous obtenons la même chose que les calculs faits à la main! Et pour Brown-Forsythe nous effectuons pareil en ajoutant l'opérateur **Brown-Forsythe's test**:



et nous avons alors:

Brown - Forsythe's test 1

Parameters

Parameters	
Sort results	no

Results

Attribute_Y	Attribute_X	Description				Statistical test	
		Value	Examples	Average	Std-dev	Test	
Pièces	Equipe	Equipe 1	7	83,5714	5,4423	Brown & Forsythe's W	0,036036
		Equipe 2	7	82,4286	5,4116	df	1/12
		All	14	83,0000	5,2477	p-value	0,852614

Soit le mêmes résultats que ceux faits à la main et dans Minitab (mais avec moins de détails: sans les IC).

Exercice 27.: Analyse en Composantes Principales pure (ACP)

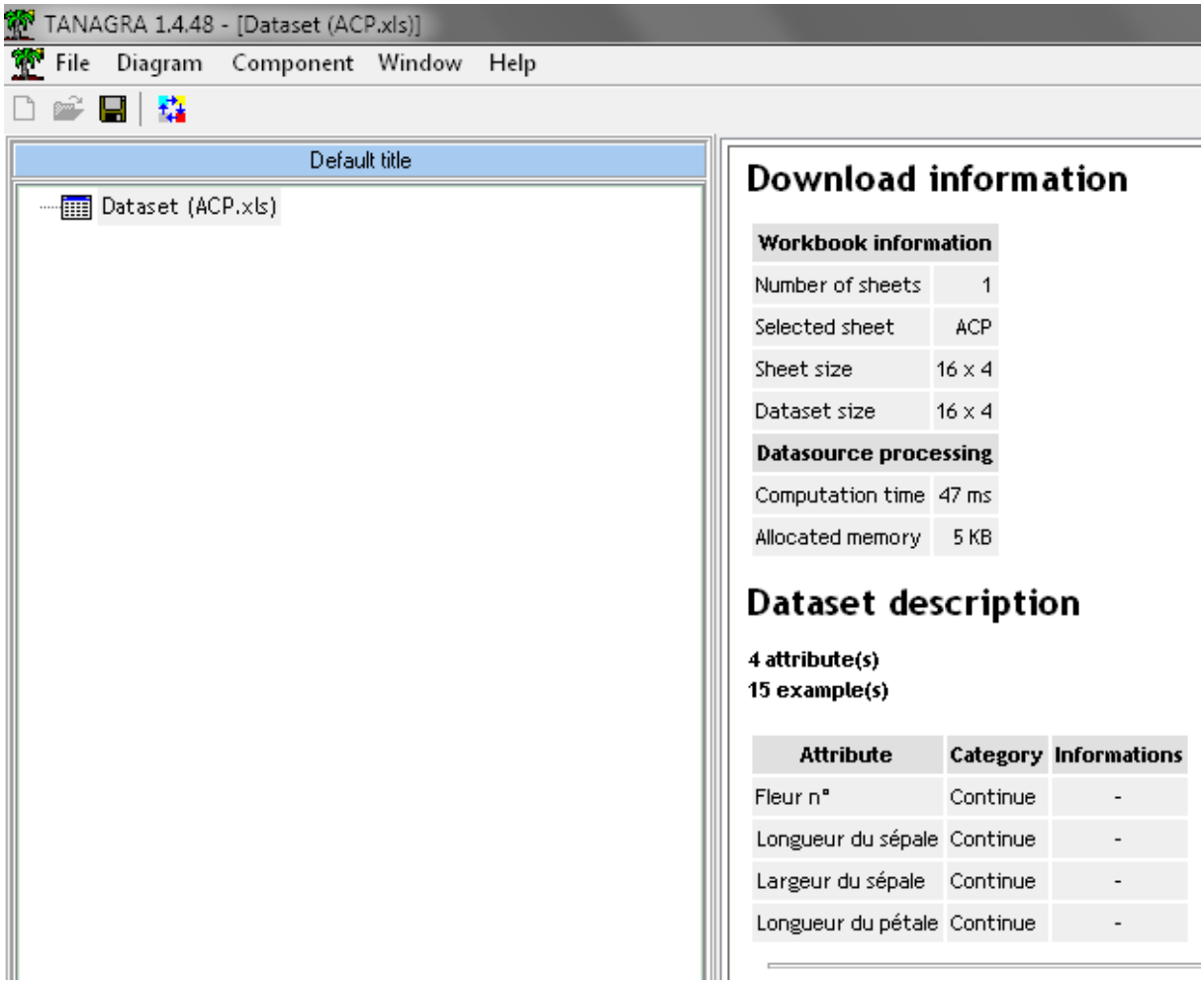
Tanagra V1.4.48

Le but va être ici de vérifier si nous retrouvons à nouveau les calculs fait à la main suite à la démonstration mathématique des concepts théoriques sous-jacents à l'A.C.P.

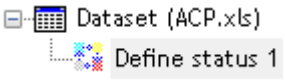
Donc nous allons prendre aussi les données d'Iris de Fisher:

	A	B	C	D
1	Fleur n°	Longueur du sépale	Largeur du sépale	Longueur du pétale
2	1	5.1	3.5	1.4
3	2	4.9	3	1.4
4	3	4.7	3.2	1.3
5	4	4.6	3.1	1.5
6	5	5	3.6	1.4
7	6	7	3.2	4.7
8	7	6.4	3.2	4.5
9	8	6.9	3.1	4.9
10	9	5.5	2.3	4
11	10	6.5	2.8	4.6
12	11	6.3	3.3	6
13	12	5.8	2.7	5.1
14	13	7.1	3	5.9
15	14	6.3	2.9	5.6
16	15	6.5	3	5.8

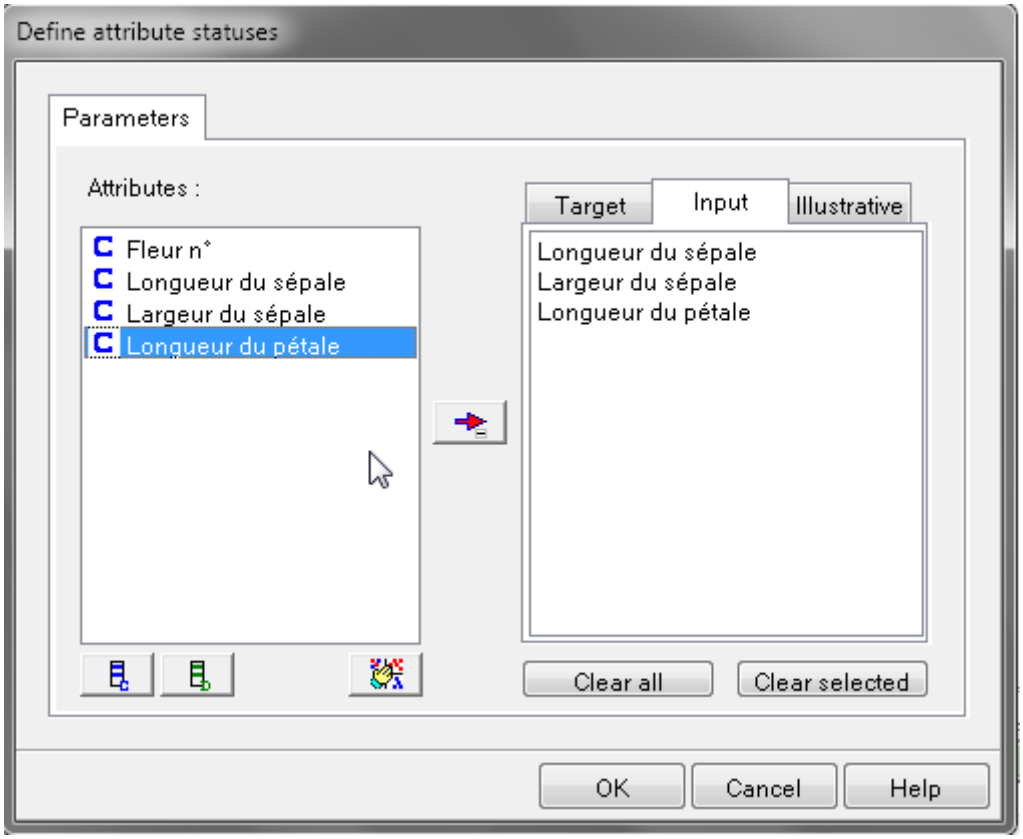
Comme à l'habitude, nous importons ces données dans Tanagra:



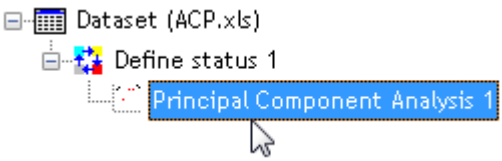
Nous ajoutons le sélecteur *Define Status*:



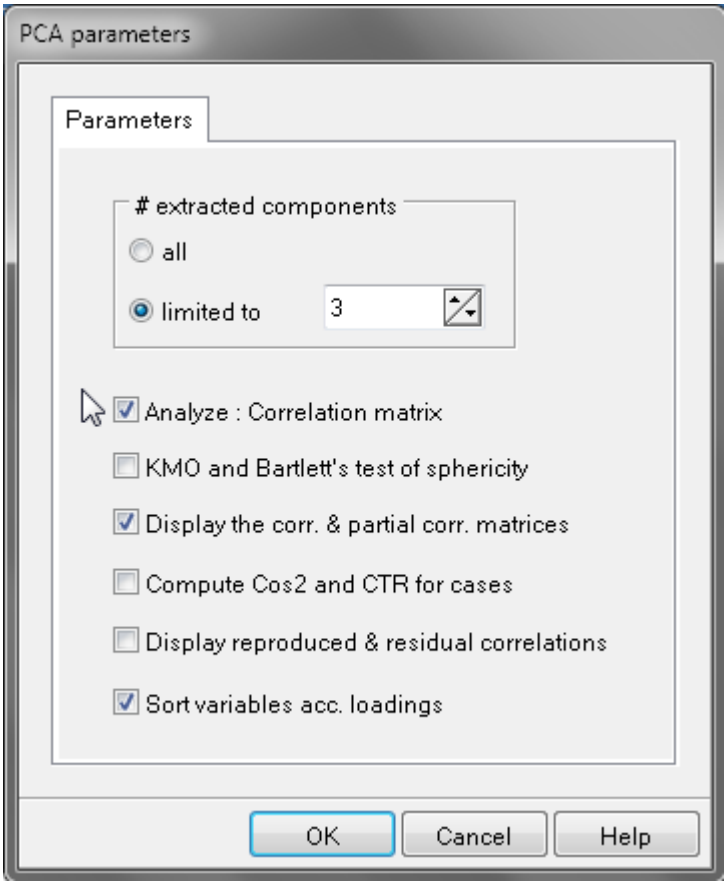
avec en *Input* les données suivantes:



Ensuite, nous ajoutons l'opérateur *Principal Component Analysis* du groupe *Factorial Analysis*:



et dans les paramètres de l'opérateur, nous prenons:



et nous exécutons le composant pour avoir pour avoir:

Principal Component Analysis 1

Parameters

Number of asked factors : 3
Compute COS2 and CTR : 0
Standardizing attributes : 1
Bartlett's test and MSA (KMO indices) : 0
Correlations and partial correlations : 1
Reproduced correlations : 0
Sort variables according to loadings : 1

Results

Eigen values

Matrix trace	**3,000000**
Average	**1,000000**

Donc nous retrouvons bien la trace de valeur 3.00 et la moyenne de 1.00. Ensuite pour la suite Tanagra donne:

Axis	Eigen value	Difference	Proportion (%)	Histogram	Cumulative (%)
1	2,030183	1,145628	67,67 %		67,67 %
2	0,884555	0,799293	29,49 %		97,16 %
3	0,085262	-	2,84 %		100,00 %
Tot.	3,000000	-	-	-	-

La aussi nous retrouvons les données calculées à la main. La suite donnée par Tanagra:

Significance of Principal Components

Global critical values	
Kaiser-Guttman	1
Karlis-Saporta-Spinaki	1,75593

Eigenvalue table - Test for significance

Eigenvalues - Significance		
Axis	Eigenvalue	Broken-stick critical values
1	2,030183	1,833333
2	0,884555	0,833333
3	0,085262	0,333333

n'a pas été étudiée en cours (exceptée la valeur numérique des trois valeurs propres bien évidemment!).

Ensuite Tanagra donne les saturations et les score de ce qui est normalement sujet de l'Analyse Factorielle sans rotation. Nous reviendrons là-dessus avec l'exemple que nous avions étudié dans le cours théorique pour l'Analyse Factorielle:

Factor Loadings [Communality Estimates]

Attribute	Axis_1		Axis_2		Axis_3	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)	Corr.	% (Tot. %)
Longueur du pétale	0,97128	*94 % (94 %)*	-0,11016	*1 % (96 %)*	-0,21088	*4 % (100 %)*
Longueur du sépale	0,91333	*83 % (83 %)*	-0,35768	*13 % (96 %)*	0,19467	*4 % (100 %)*
Largeur du sépale	-0,50261	*25 % (25 %)*	-0,86284	*74 % (100 %)*	-0,05377	*0 % (100 %)*
Var. Expl.	2,03018	*68 % (68 %)*	0,88455	*29 % (97 %)*	0,08526	*3 % (100 %)*

Factor Score Coefficients

Attribute	Mean	Std-dev	Axis_1	Axis_2	Axis_3
Longueur du sépale	5,9066667	0,8441696	0,6410021	-0,3803022	0,6666982
Largeur du sépale	3,0600000	0,3072458	-0,3527505	-0,9174154	-0,1841631
Longueur du pétale	3,8733333	1,8277369	0,6816768	-0,1171292	-0,7222170

Ensuite Tanagra donne la matrice des corrélations que nous avions calculée (bien évidemment obligatoirement) dans le cours théorique avec les mêmes valeurs:

Matrices

Correlations

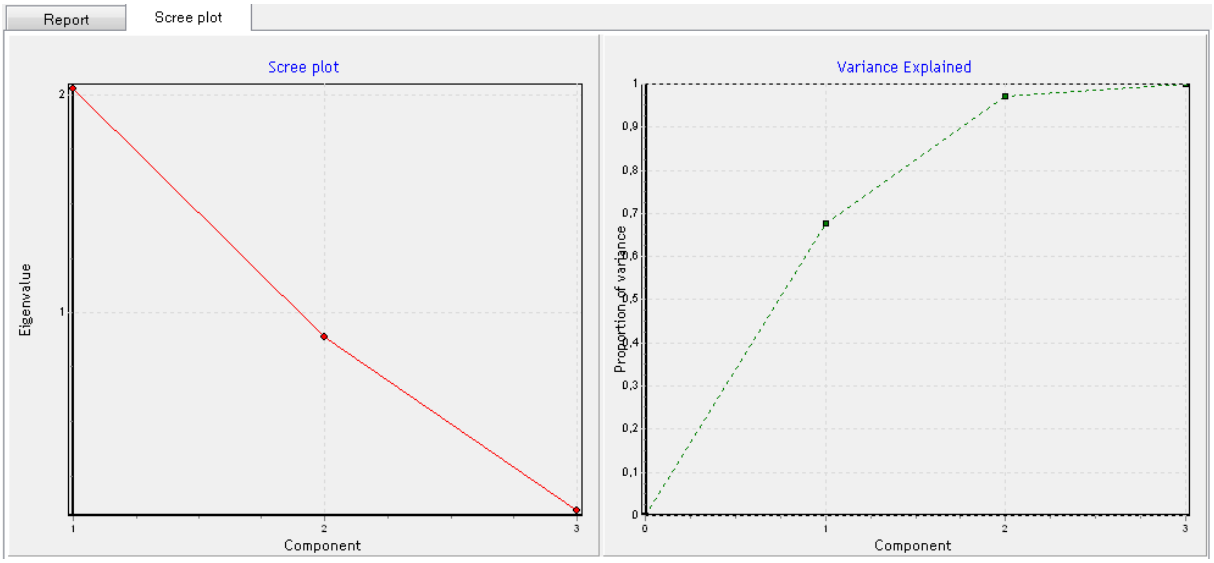
	Longueur du pétale	Longueur du sépale	Largeur du sépale
Longueur du pétale	1,00000	0,88545	-0,38179
Longueur du sépale	0,88545	1,00000	-0,16090
Largeur du sépale	-0,38179	-0,16090	1,00000

Ensuite Tanagra donne les corrélations partielles (mais cela n'est normalement pas directement liée à l'A.C.P.). donc nous ne l'avions pas calculé dans le cours théorique, nous le mettrons donc de côté:

Partial Correlations Controlling all other Variables

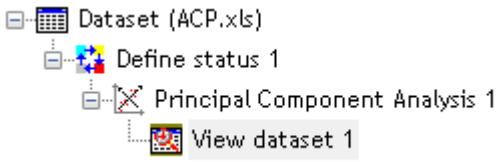
	Longueur du pétale	Longueur du sépale	Largeur du sépale
Longueur du pétale	1,00000	0,90332	-0,52175
Longueur du sépale	0,90332	1,00000	0,41243
Largeur du sépale	-0,52175	0,41243	1,00000

et il vient automatiquement les deux tracés triviaux suivants qui sont donnés par Tanagra:



Il est possible d'accéder directement aux données calculées, c'est-à-dire les projections dans le nouvel espace (calcul laborieux que nous n'avons pas fait dans le cours théorique). En effet, le composant ACP rajoute automatiquement une série de variables à l'ensemble de données. Il s'agit, pour chaque individu et pour chaque axe demandé, des projections sur les axes, des contributions et des cos².

Pour visualiser le tableau de données associé, nous plaçons dans le diagramme le composant *View Dataset* du groupe *Data visualization*

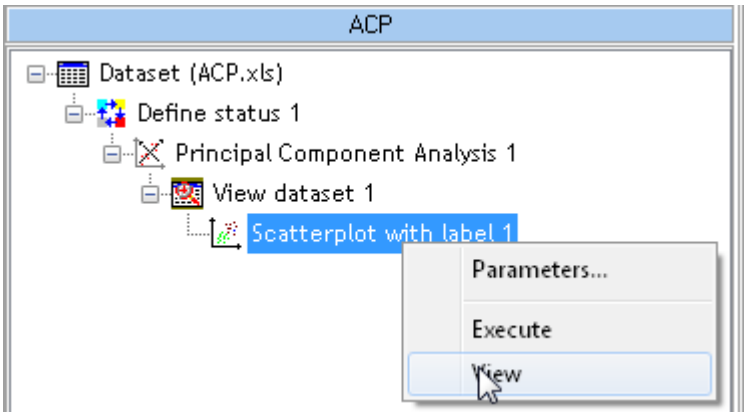


et nous double cliquons dessus pour obtenir:

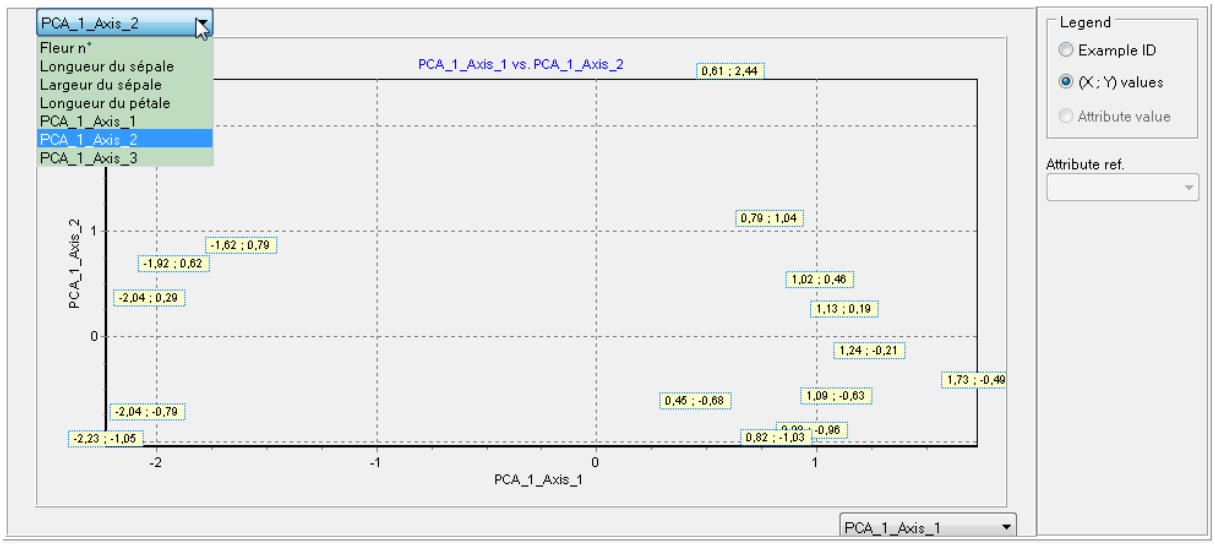
	Fleur n°	Longueur d	Largeur du	Longueur d	PCA_1_Axis	PCA_1_Axis	PCA_1_Axis
1	1	5,1	3,5	1,4	-2,04015	-0,791902	0,0765041
2	2	4,9	3	1,4	-1,61796	0,791166	0,218251
3	3	4,7	3,2	1,3	-2,03675	0,290488	-0,0200689
4	4	4,6	3,1	1,5	-1,92328	0,621316	-0,118134
5	5	5	3,6	1,4	-2,23089	-1,04544	-0,0624125
6	6	7	3,2	4,7	0,97778	-0,963559	0,452912
7	7	6,4	3,2	4,5	0,44759	-0,680439	0,0580801
8	8	6,9	3,1	4,9	1,09125	-0,632731	0,354847
9	9	5,5	2,3	4	0,611008	2,4444	0,0843202
10	10	6,5	2,8	4,6	1,02006	0,462475	0,337303
11	11	6,3	3,3	6	0,81629	-1,03011	-0,673551
12	12	5,8	2,7	5,1	0,789823	1,04438	-0,353166
13	13	7,1	3	5,9	1,73089	-0,488324	0,177598
14	14	6,3	2,9	5,6	1,12635	0,189898	-0,275734
15	15	6,5	3	5,8	1,238	-0,211612	-0,256749

La popularité de l'ACP repose en grande partie sur les représentations graphiques qu'elle propose. Elles nous permettent d'apprécier visuellement les proximités entre les observations.

Dans notre cas, nous projetons les observations dans le premier plan factoriel. Nous voulons associer les identifiants aux points. Nous utilisons pour cela le composant SCATTERPLOT WITH LABEL (onglet DATA VISUALIZATION) que nous plaçons en dessous de l'ACP. Nous le paramétrons de manière à avoir en abscisse le premier facteur, en ordonnée le second facteur.



Notons qu'il est très aisé avec Tanagra de passer d'un plan factoriel à un autre:



Il est possible de modifier la taille des étiquettes avec les raccourcis CTRL+Q et CTRL+W.

Nous voyons que nous retrouvons la même forme de graphique au niveau visuel que celle obtenue dans le cours théorique mais les données ne sont pas centrées réduites (du moins a priori). Le graphique a cependant exactement les mêmes valeurs que celui sorti par le logiciel Minitab.

Exercice 28.: Analyse Factorielle sans rotation (AF)

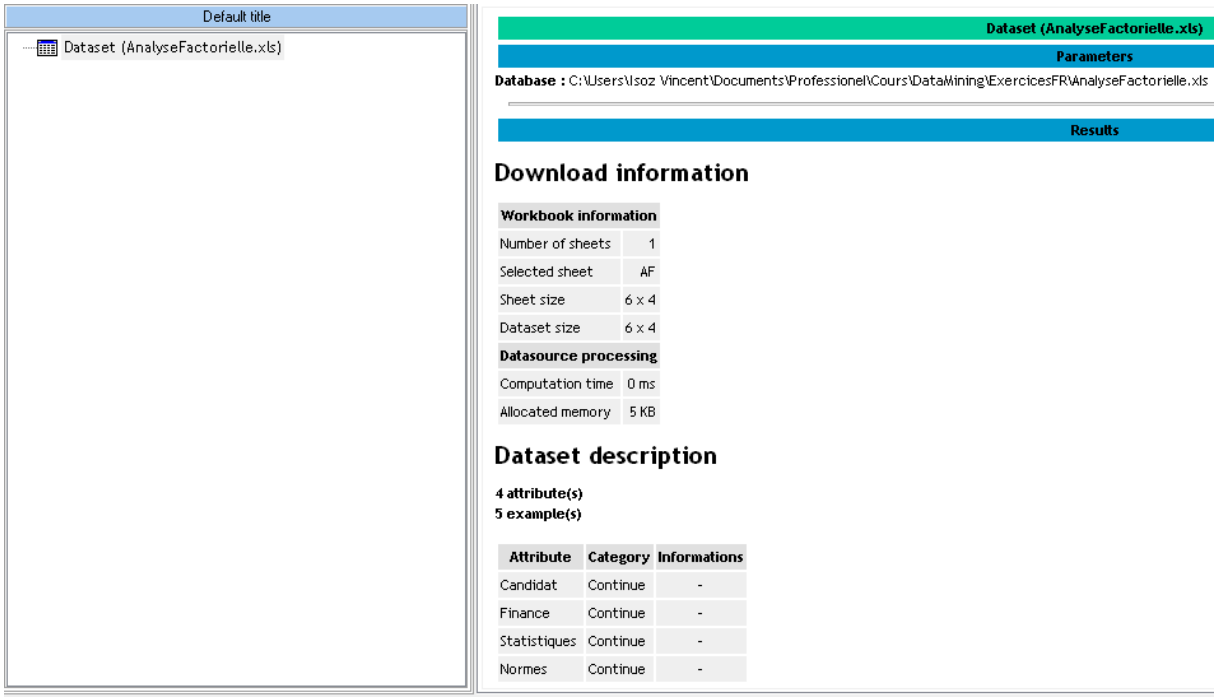
Tanagra V1.4.49

Voyons donc comment obtenir une analyse factorielle sans rotation et tout cela avec l'exemple qui nous a servi de bases pour les calculs à la main lors de la démonstration mathématique de la méthode.

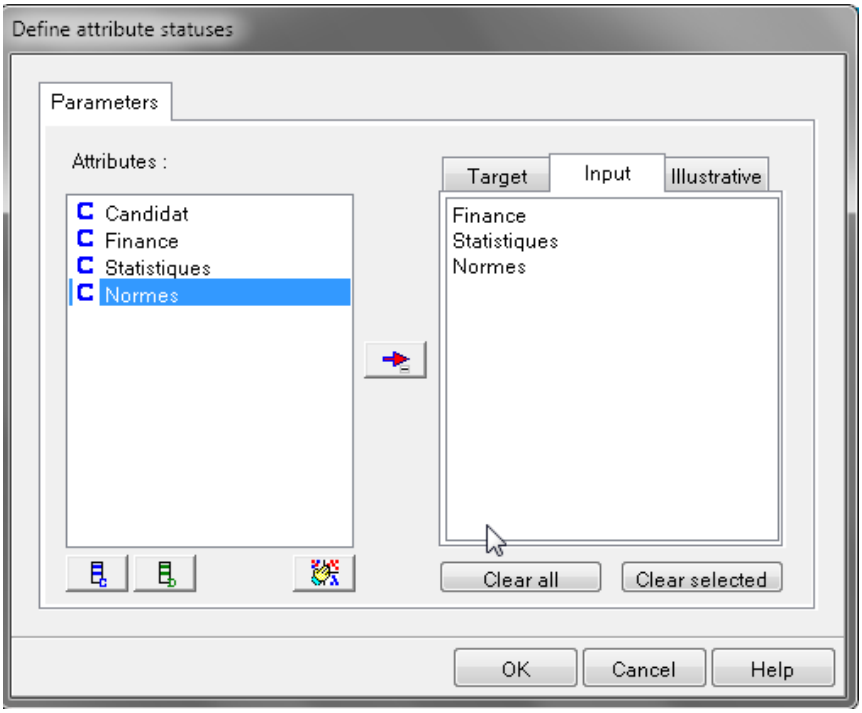
Nous importons donc comme à l'habitude les données suivantes:

	A	B	C	D
1	Candidat	Finance	Statistiques	Normes
2	1	3	6	5
3	2	7	3	3
4	3	10	9	8
5	4	3	9	7
6	5	10	6	5

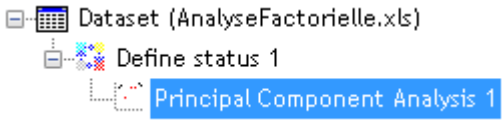
dans Tanagra:



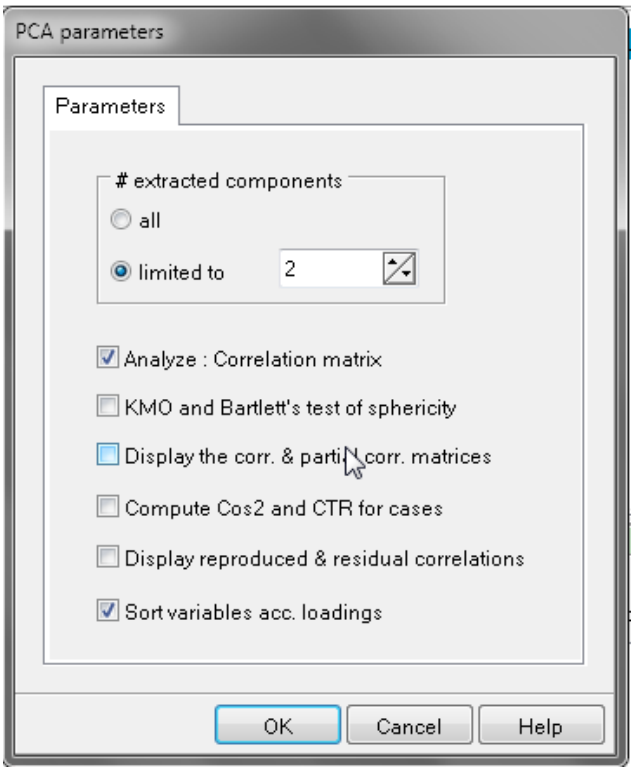
Nous ajoutons le composant de sélection *Define Status* et mettons en *Input* les trois variables:



et c'est maintenant qu'intervient une petite subtilité de Tanagra: Si nous voulons retrouver les valeurs calculées en cours à la main et conformes au modèle mathématique sans rotation, nous devons utiliser le composant *Principal Component Analysis*:



et pour être conforme à l'exemple particulier que nous avons vu dans le cours théorique, mettre les paramètres suivants:



En exécutant le composant, il vient dans un premier temps:

Principal Component Analysis 1

Parameters

Number of asked factors : 2
Compute COS2 and CTR : 0
Standardizing attributes : 1
Bartlett's test and MSA (KMO indices) : 0
Correlations and partial correlations : 0
Reproduced correlations : 0
Sort variables according to loadings : 1

Results

Eigen values

Matrix trace	**3,000000**
Average	**1,000000**

Axis	Eigen value	Difference	Proportion (%)	Histogram	Cumulative (%)
1	1,981463	0,973157	66,05 %		66,05 %
2	1,008306	0,998076	33,61 %		99,66 %
3	0,010231	-	0,34 %		100,00 %
Tot.	3,000000	-	-	-	-

valeurs des valeurs propres identiques à celles calculées en cours. Ensuite, nous avons tout en bas les deux tableaux qui nous intéressent:

Factor Loadings [Communality Estimates]

Attribute	Axis_1		Axis_2	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)
Normes	-0,99613	99 % (99 %)	-0,05139	0 % (99 %)
Statistiques	-0,99413	99 % (99 %)	0,08153	1 % (99 %)
Finance	-0,02987	0 % (0 %)	-0,99951	100 % (100 %)
Var. Expl.	1,98146	66 % (66 %)	1,00831	34 % (100 %)

où nous retrouvons bien les saturations calculées à la main dans le cours théorique au signe près (mises en évidence en rouge et nommées pour rappel en anglais "loadings").

Et le dernier tableau:

Factor Score Coefficients

Attribute	Mean	Std-dev	Axis_1	Axis_2
Finance	6,6000000	3,1368774	-0,0212208	-0,9953835
Statistiques	6,6000000	2,2449944	-0,7062359	0,0811929
Normes	5,6000000	1,7435596	-0,7076585	-0,0511807

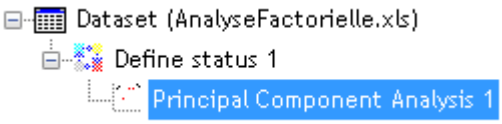
n'est pas contre pour les deux dernières colonnes pas conforme à ce que nous avons calculé manuellement dans le cours théorique ni conforme à ce que nous renvoie le logiciel Minitab.

Exercice 29.: Analyse Factorielle avec rotation VARIMAX

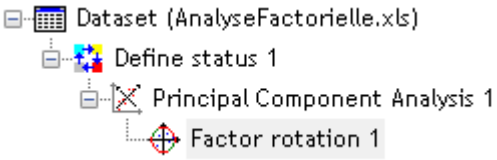
Tanagra V1.4.49

Le but va être ici de vérifier non pas les calculs faits à la main dans le cours théorique mais de vérifier que Tanagra redonne les mêmes résultats que Minitab ou que SAS pour les mêmes données que l'exemple précédent mais avec une rotation VARIMAX.

Donc nous reprenons l'état précédent où nous avions:



et nous ajoutons l'opérateur *Factor rotation* du groupe *Factor analysis*:



pour obtenir:

Factor rotation 1

Parameters

Factors rotation	
Method	VARIMAX
# factors	2
Reproduced correlations	0
Sort variables according to loadings	1

Results

Rotated Factor Loadings

Attribute	Axis_1		Axis_2	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)
Statistiques	0,99572	*99 % (99 %)*	-0,05900	*0 % (99 %)*
Normes	0,99471	*99 % (99 %)*	0,07393	*1 % (99 %)*
Finance	0,00723	*0 % (0 %)*	0,99993	*100 % (100 %)*
Var. Expl.	1,98096	*66 % (66 %)*	1,00881	*34 % (100 %)*

où nous retrouvons bien les résultats de SAS!

Tanagra donne en-dessous le tableau des saturations sans rotation (tableau obtenu lors de l'exercice précédent!):

vs. Unrotated Factor Loadings

Attribute	Axis_1		Axis_2	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)
Statistiques	-0,99413	*99 % (99 %)*	0,08153	*1 % (99 %)*
Normes	-0,99613	*99 % (99 %)*	-0,05139	*0 % (99 %)*
Finance	-0,02987	*0 % (0 %)*	-0,99951	*100 % (100 %)*
Var. Expl.	1,98146	*66 % (66 %)*	1,00831	*34 % (100 %)*

et le score des facteurs après rotation:

Factor Scores

Attribute	Mean	Std-dev	Axis_1	Axis_2
Finance	6,6000000	3,1368774	-0,0103745	0,9957173
Statistiques	6,6000000	2,2449944	0,7085425	-0,0697833
Normes	5,6000000	1,7435596	0,7057639	0,0626120

Exercice 30.: Régression (linéaire) des moindres carrés partiels (régression linéaire PLS univariée: PLS1)

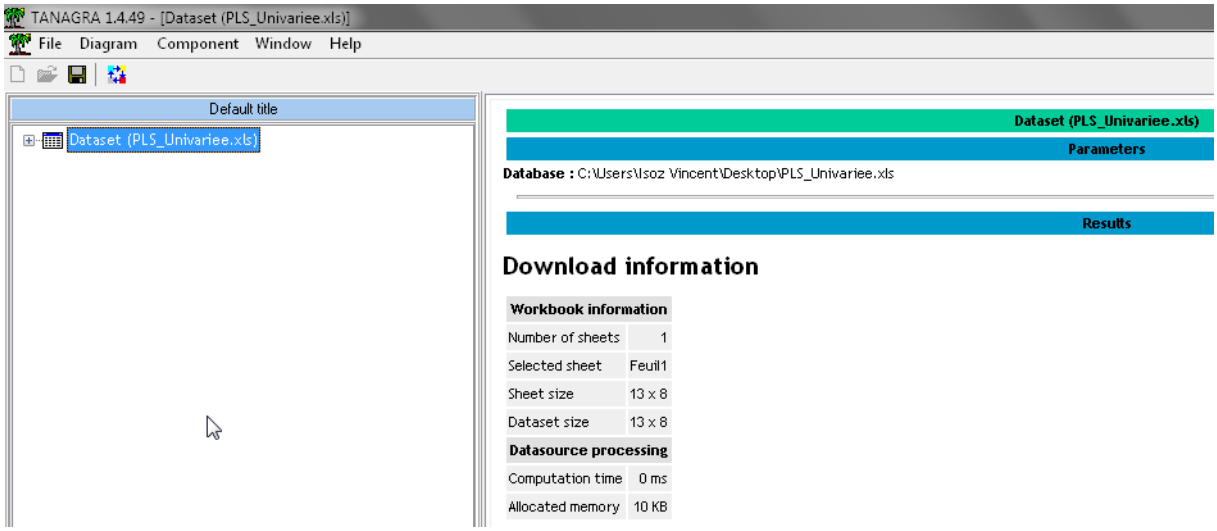
Tanagra V1.4.48

Le but va être ici de vérifier si nous obtenons ou pas les résultats des calculs vu dans le cours théorique lors de la lecture du l'ouvrage de M. Tenenhaus[1] sur la régression PLS univariée (PLS1), c'est-à-dire la régression sur des variables explicatives corrélées avec une unique variable à expliquer.

Nous utiliserons donc les données suivantes:

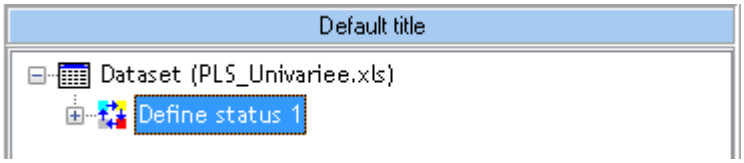
	A	B	C	D	E	F	G	H
1	Dist_Directe	Reformat	Naptha_The	Naptha_Cat	Polymere	Alkylat	Essence_Naturelle	Reponse
2	0.00	0.23	0.00	0.00	0.00	0.74	0.03	98.70
3	0.00	0.10	0.00	0.00	0.12	0.74	0.04	97.80
4	0.00	0.00	0.00	0.10	0.12	0.74	0.04	96.60
5	0.00	0.49	0.00	0.00	0.12	0.37	0.02	92.00
6	0.00	0.00	0.00	0.62	0.12	0.18	0.08	86.60
7	0.00	0.62	0.00	0.00	0.00	0.37	0.01	91.20
8	0.17	0.27	0.10	0.38	0.00	0.00	0.08	81.90
9	0.17	0.19	0.10	0.38	0.02	0.06	0.08	83.10
10	0.17	0.21	0.10	0.38	0.00	0.06	0.08	82.40
11	0.17	0.15	0.10	0.38	0.02	0.10	0.08	83.20
12	0.21	0.36	0.12	0.25	0.00	0.00	0.06	81.40
13	0.00	0.00	0.00	0.55	0.00	0.37	0.08	88.10

que nous importons comme à l'habitude dans Tanagra, ce qui donnera:

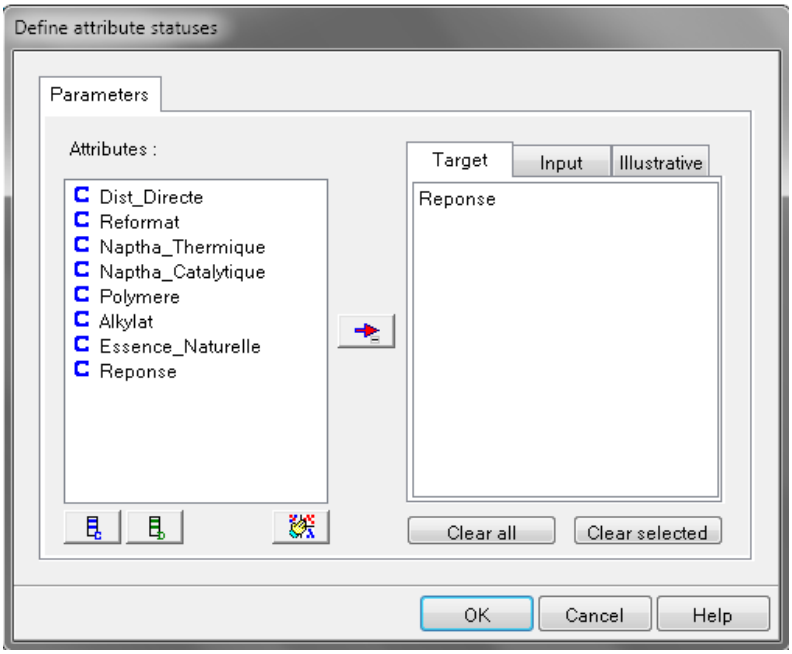


Ensuite, nous ajoutons le sélecteur *Define Status* comme à l'habitude:

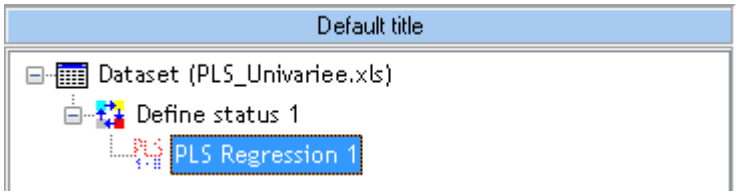
[1]Michel Tenenhaus, *Régression PLS, Édition Technip, ISBN 2-7108-0735-1*, Pages 75-83



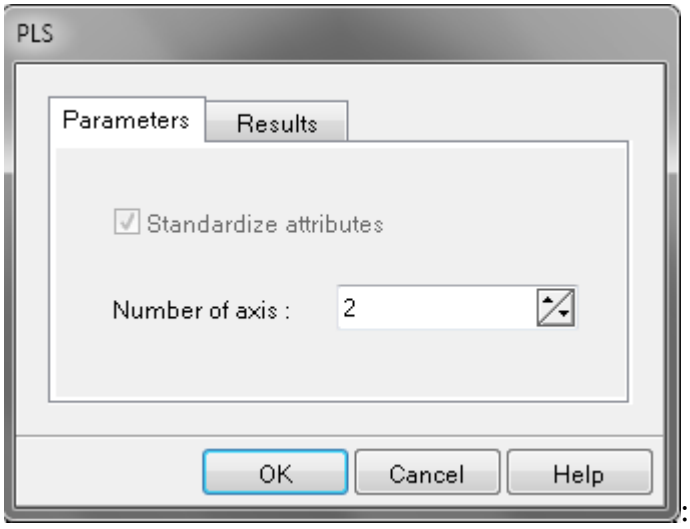
avec les paramètres d'entrée et de sortie suivants:



Ensuite, nous ajoutons le composant *PLS Regression*:

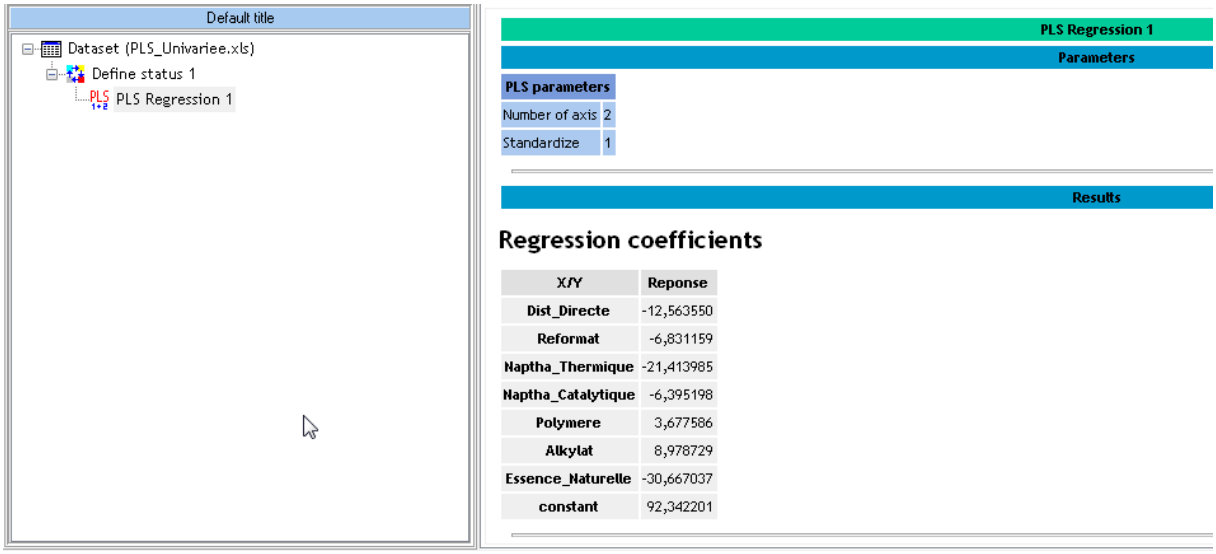


et allons dans les paramètres pour prendre que deux variables intermédiaire:



.

Nous avons alors:



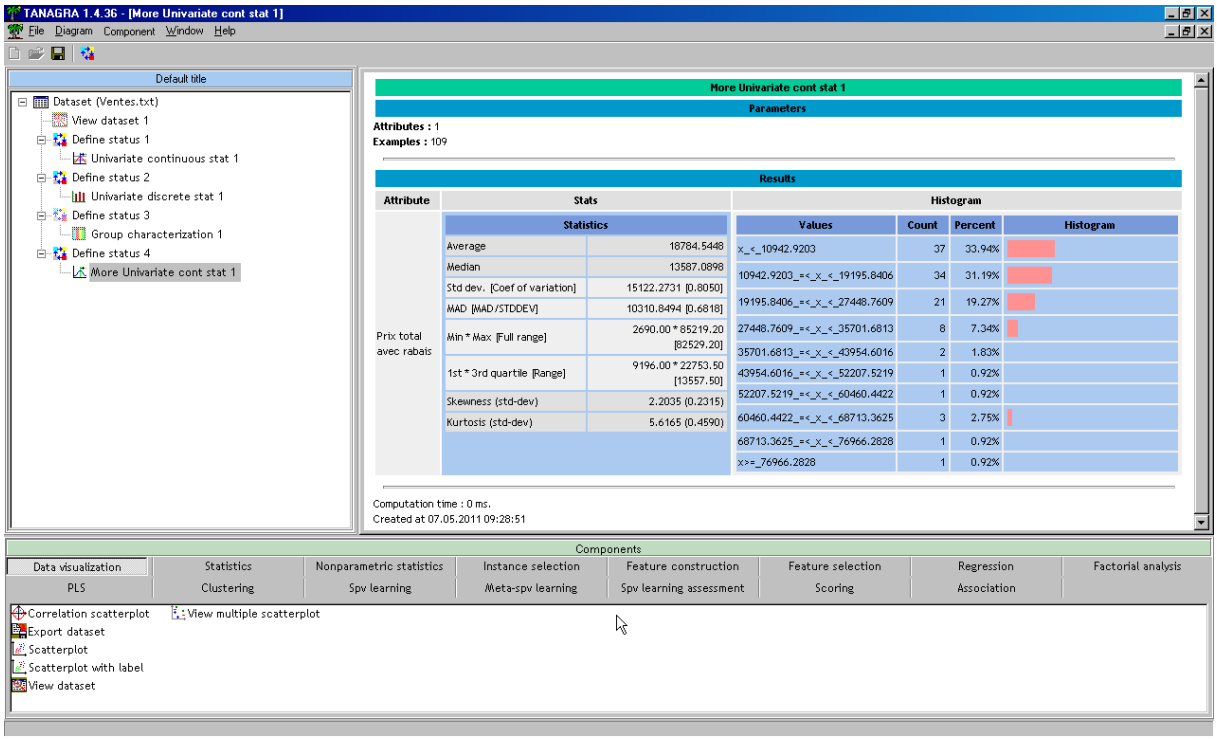
On retrouve bien les mêmes coefficients non normalisés que dans Minitab ou que ceux calculés à la main.

Exercice 31.: Export d'un résultat vers MS Excel

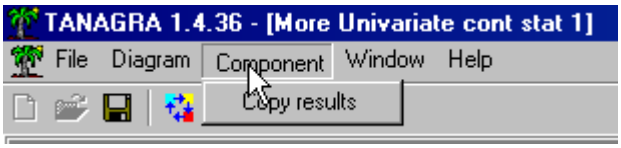
Tanagra V1.4.36

Nous souhaiterions montrer ici qu'il est possible rapidement d'exporter une analyse ainsi qu'un jeu de données traitées dans MS Excel.

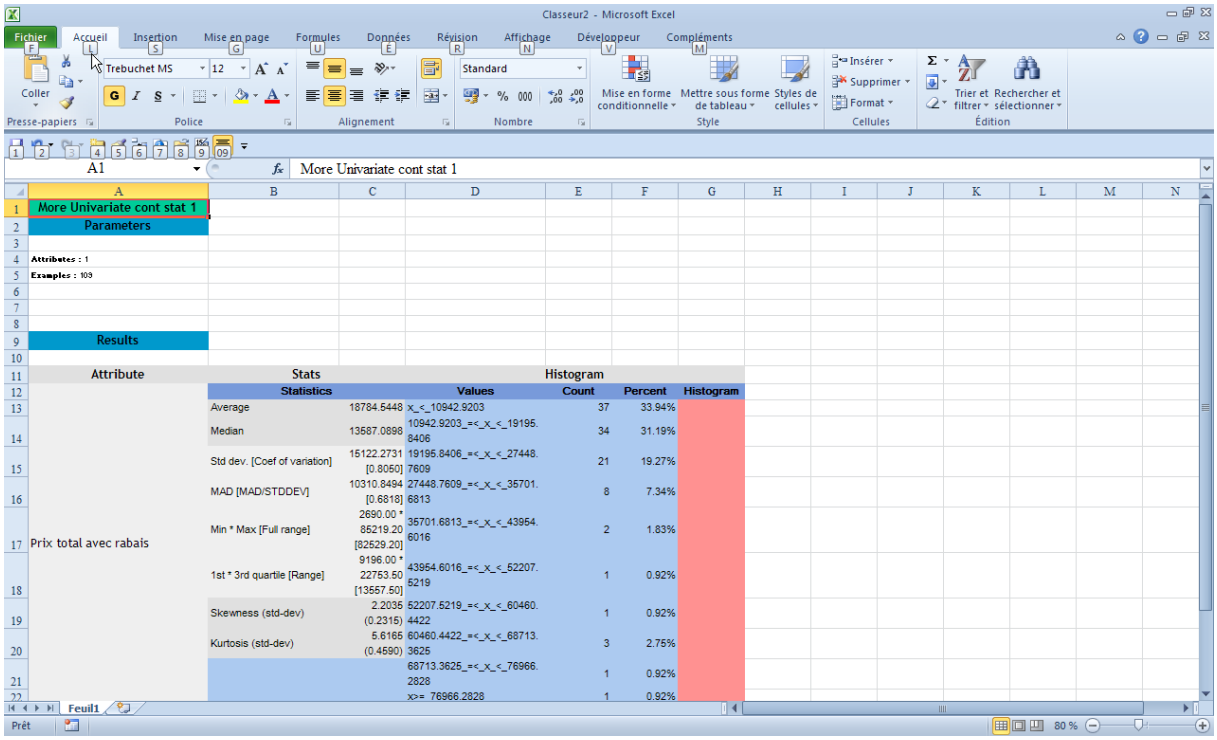
Pour commencer avec le premier cas reprenons l'exemple de l'exercice que nous avions fait sur l'**Exercice 6.: Statistiques univariées continues multiples**:



Nous allons dans le menu **Component** et nous cliquons sur **Copy results**:



et nous faisons un **Coller** dans MS Excel pour obtenir:



nous retrouvons donc bien les données textes mais par contre nous perdons les données visuelles comme les barre de données de l'histogramme. Heureusement cela peut être vite reproduit.

Maintenant, reprenons l'**Exercice 19.: K-NN** (K nearest neighbors)
Tanagra V1.4.48

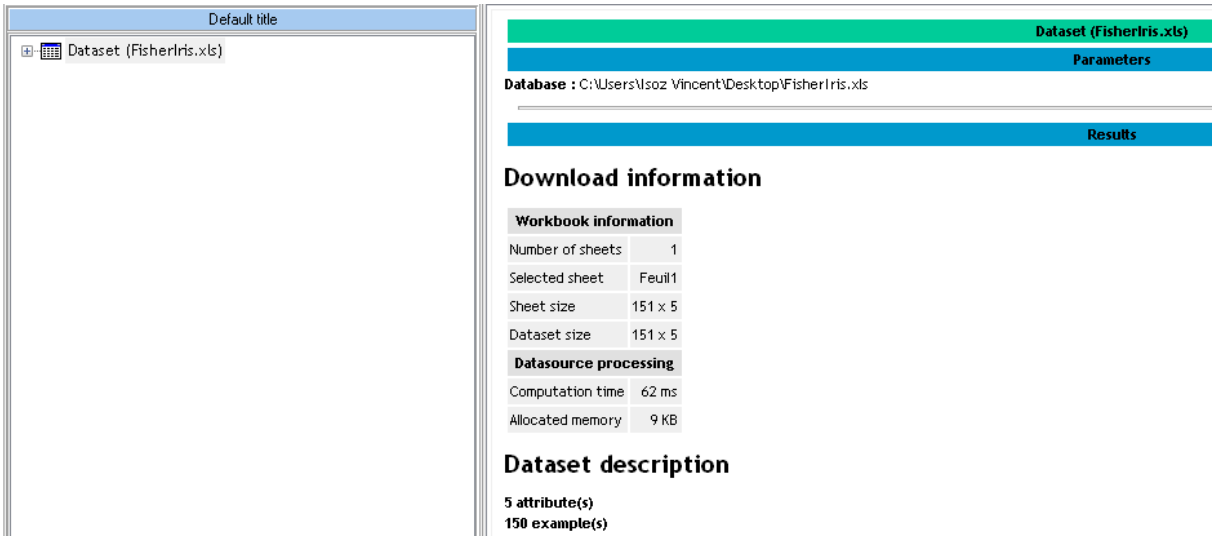
Nous avons vu en cours l'approche des k plus proches voisins. Nous allons appliquer ici ce qui a été présenté en cours avec le fichier Excel des fleurs d'Iris

FisherIris.xls

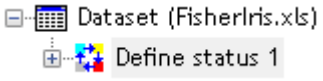
dont le contenu est:

	A	B	C	D	E
1	Sepal length	Sepal width	Petal length	Petal width	Species
2	7.9	3.8	6.4	2	*I. virginica*
3	7.7	3.8	6.7	2.2	*I. virginica*
4	7.7	2.6	6.9	2.3	*I. virginica*
5	7.7	2.8	6.7	2	*I. virginica*
6	7.7	3	6.1	2.3	*I. virginica*
7	7.6	3	6.6	2.1	*I. virginica*
8	7.4	2.8	6.1	1.9	*I. virginica*
9	7.3	2.9	6.3	1.8	*I. virginica*
10	7.2	3.6	6.1	2.5	*I. virginica*
11	7.2	3.2	6	1.8	*I. virginica*
12	7.2	3	5.8	1.6	*I. virginica*
13	7.1	3	5.9	2.1	*I. virginica*
14	7	3.2	4.7	1.4	*I. versicolor*
15	6.9	3.1	4.9	1.5	*I. versicolor*
16	6.9	3.2	5.7	2.3	*I. virginica*
17	6.9	3.1	5.4	2.1	*I. virginica*
18	6.9	3.1	5.1	2.3	*I. virginica*
19	6.8	2.8	4.8	1.4	*I. versicolor*
20	6.8	3	5.5	2.1	*I. virginica*
21	6.8	3.2	5.9	2.3	*I. virginica*
22	6.7	3.1	4.4	1.4	*I. versicolor*

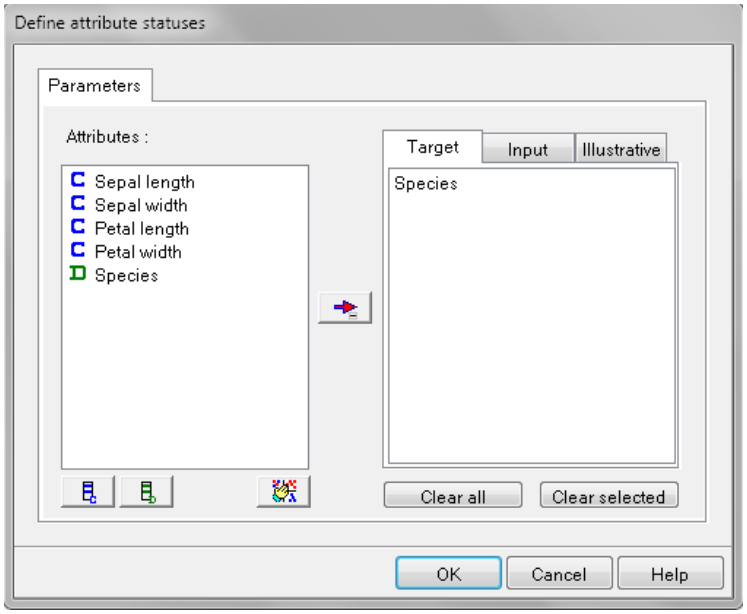
Ensuite, nous l'ouvrons dans Tanagra selon la méthode habituelle:



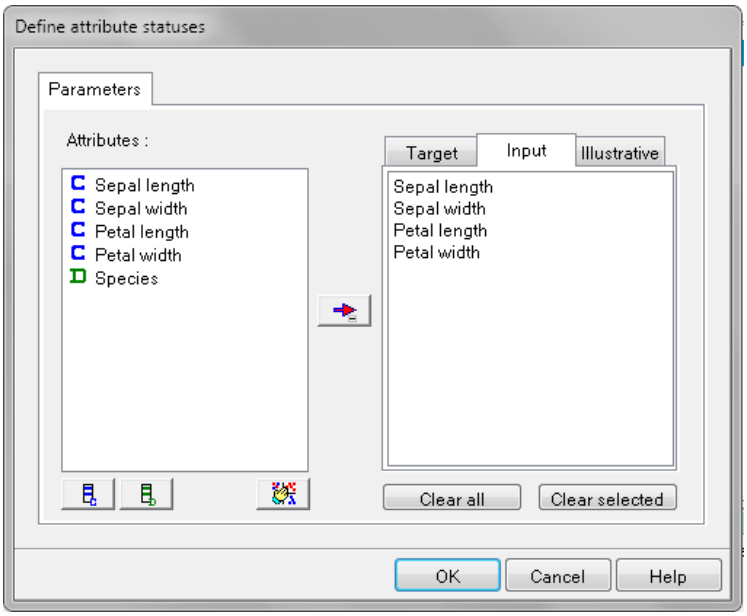
Ensuite, nous ajoutons le sélecteur *Define Status*:



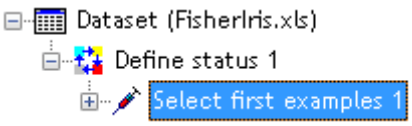
avec en *Target*:



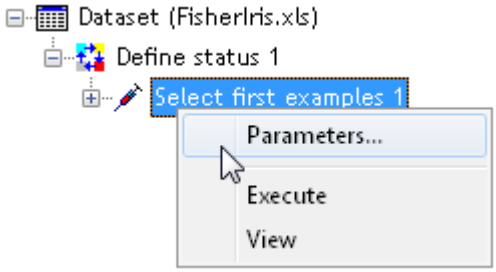
et en *Input*:



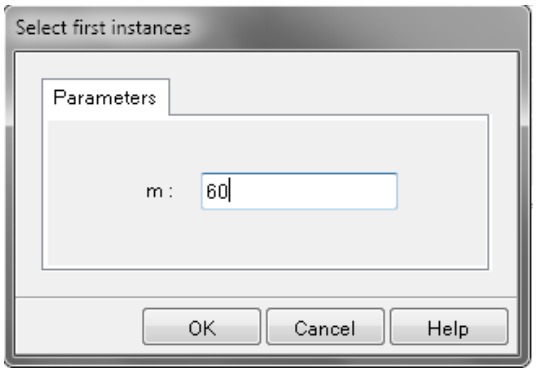
Ensuite nous rajoutons le sélecteur *Select first examples* du groupe *Instance selection*:



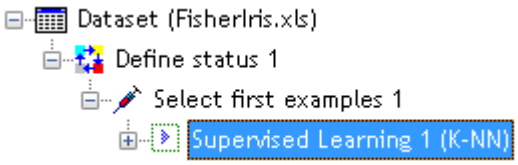
et dans les paramètres du sélecteur:



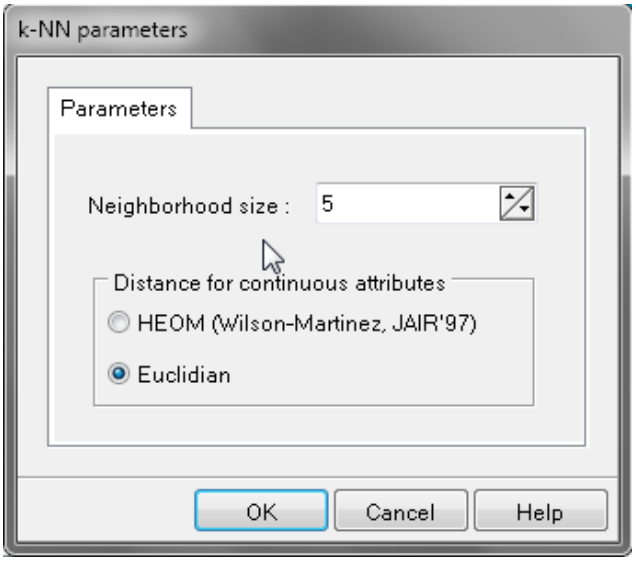
Nous prenons les 60 premières lignes du fichier comme données d'entraînement (choix un peu arbitraire):



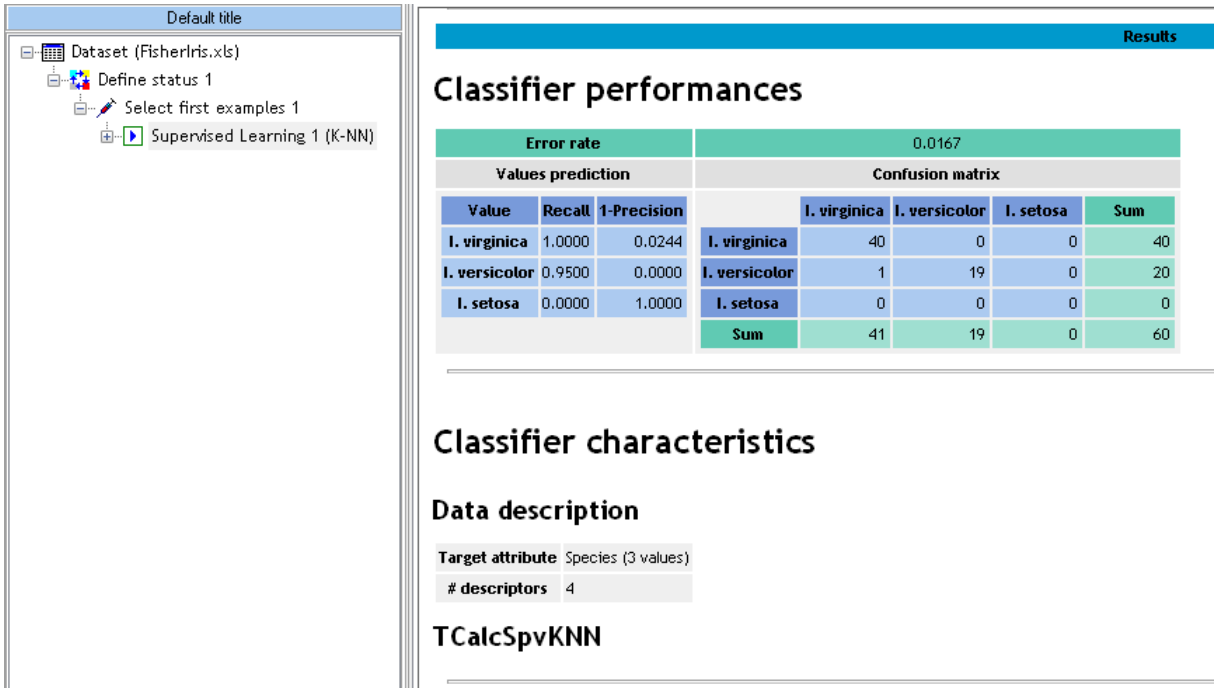
Ensuite, nous rajoutons l'opérateur *K-NN* du groupe *Spv Learning*:



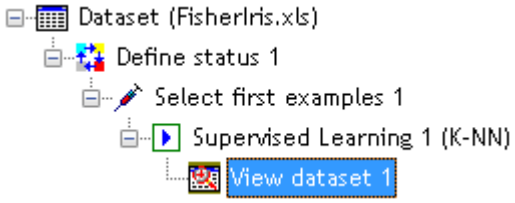
Ensuite, nous choisissons le type de distance et le nombre de *k* voisins pour l'apprentissage:



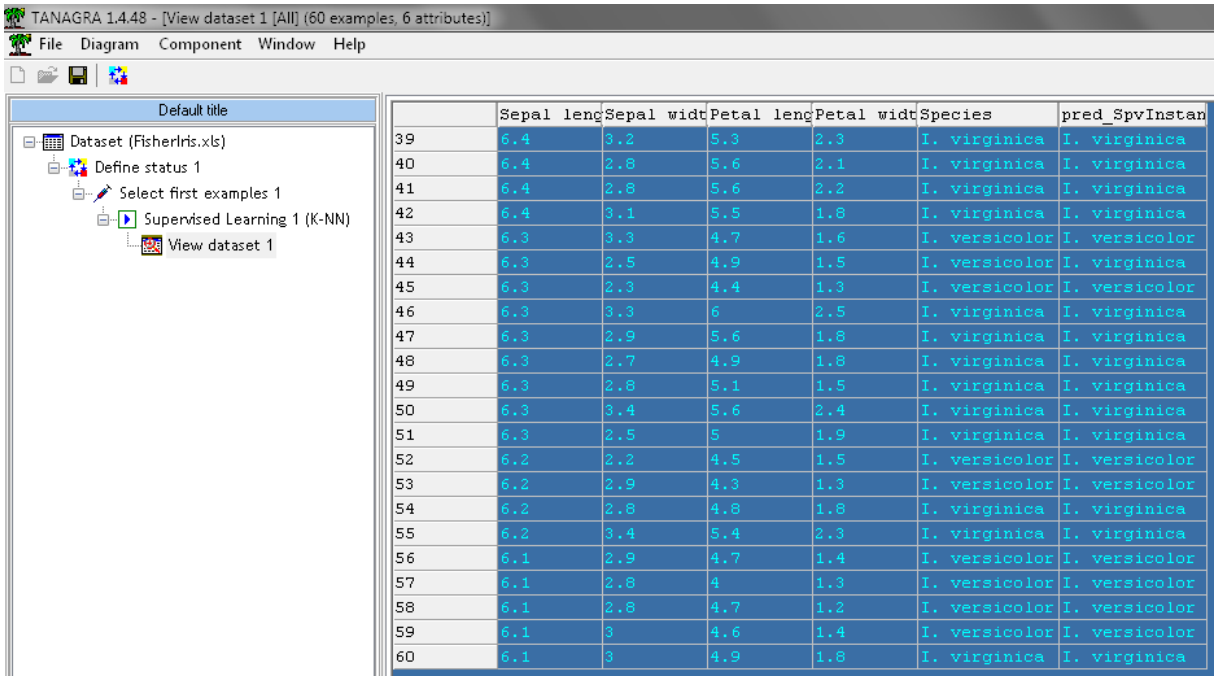
Nous exécutons l'opérateur et nous avons alors:



Nous voyons que le classificateur est très bon. Pour avoir le détail, nous ajoutons l'opérateur *View Data Set* du groupe *Data visualization*:

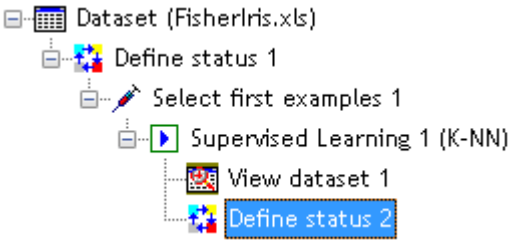


et nous l'exécutons pour avoir les détails des prédictions (nous avons mis en évidence l'un deux ceux qui est mal prédit):

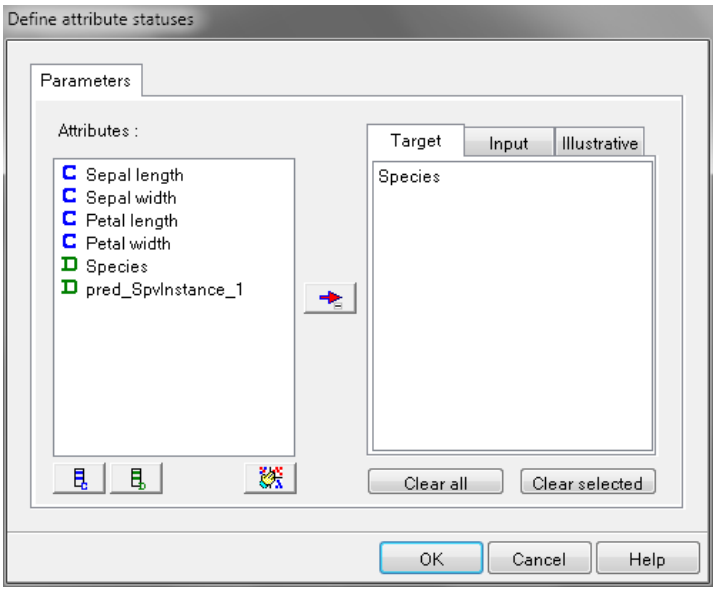


	Sepal leng	Sepal widt	Petal leng	Petal widt	Species	pred_SpvInstan
39	6.4	3.2	5.3	2.3	I. virginica	I. virginica
40	6.4	2.8	5.6	2.1	I. virginica	I. virginica
41	6.4	2.8	5.6	2.2	I. virginica	I. virginica
42	6.4	3.1	5.5	1.8	I. virginica	I. virginica
43	6.3	3.3	4.7	1.6	I. versicolor	I. versicolor
44	6.3	2.5	4.9	1.5	I. versicolor	I. virginica
45	6.3	2.3	4.4	1.3	I. versicolor	I. versicolor
46	6.3	3.3	6	2.5	I. virginica	I. virginica
47	6.3	2.9	5.6	1.8	I. virginica	I. virginica
48	6.3	2.7	4.9	1.8	I. virginica	I. virginica
49	6.3	2.8	5.1	1.5	I. virginica	I. virginica
50	6.3	3.4	5.6	2.4	I. virginica	I. virginica
51	6.3	2.5	5	1.9	I. virginica	I. virginica
52	6.2	2.2	4.5	1.5	I. versicolor	I. versicolor
53	6.2	2.9	4.3	1.3	I. versicolor	I. versicolor
54	6.2	2.8	4.8	1.8	I. virginica	I. virginica
55	6.2	3.4	5.4	2.3	I. virginica	I. virginica
56	6.1	2.9	4.7	1.4	I. versicolor	I. versicolor
57	6.1	2.8	4	1.3	I. versicolor	I. versicolor
58	6.1	2.8	4.7	1.2	I. versicolor	I. versicolor
59	6.1	3	4.6	1.4	I. versicolor	I. versicolor
60	6.1	3	4.9	1.8	I. virginica	I. virginica

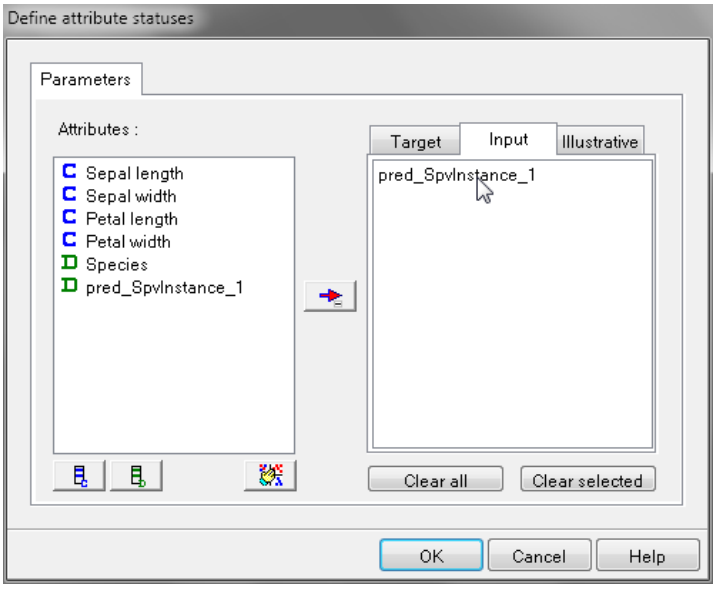
Maintenant injectons pour y mettre un jeu de test, nous remettons un opérateur de sélection *Define Status*:



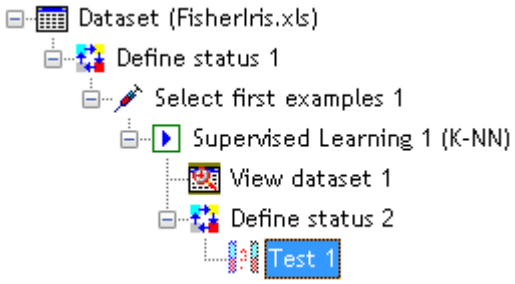
avec en *Target*:



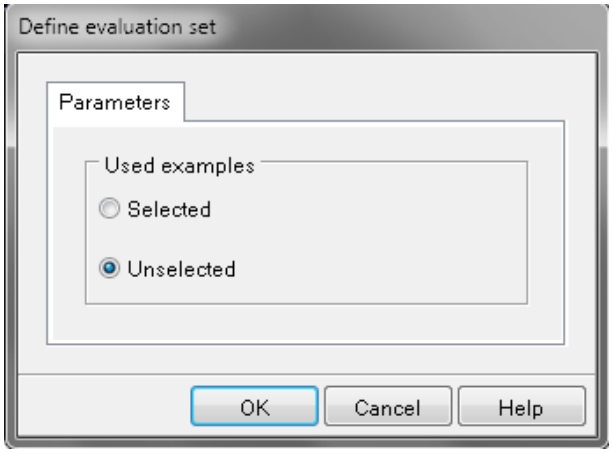
et en *Input*:



Ensuite, nous rajoutons l'opérateur *Test* du groupe *Spv learning assessment* (nous aurions pu faire la même chose pour la régression logistique mais ayant l'équation explicite c'était moins utile alors que là c'est très utile!):

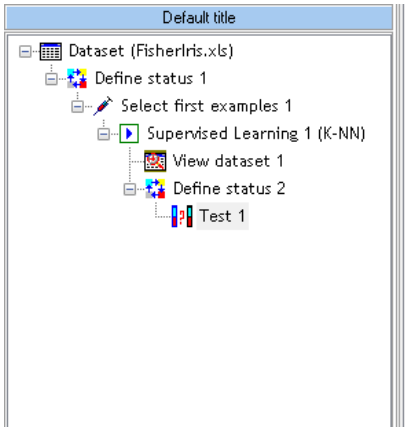


et dans ses paramètres, nous avons:



Nous prenons *Unselected* ce qui prendra les 150-60=90 données restantes.

Et nous exécutons pour obtenir:



Test 1

Parameters

Evaluation set : **unselected** examples

Results

pred_SpvInstance_1

Error rate	0.5778

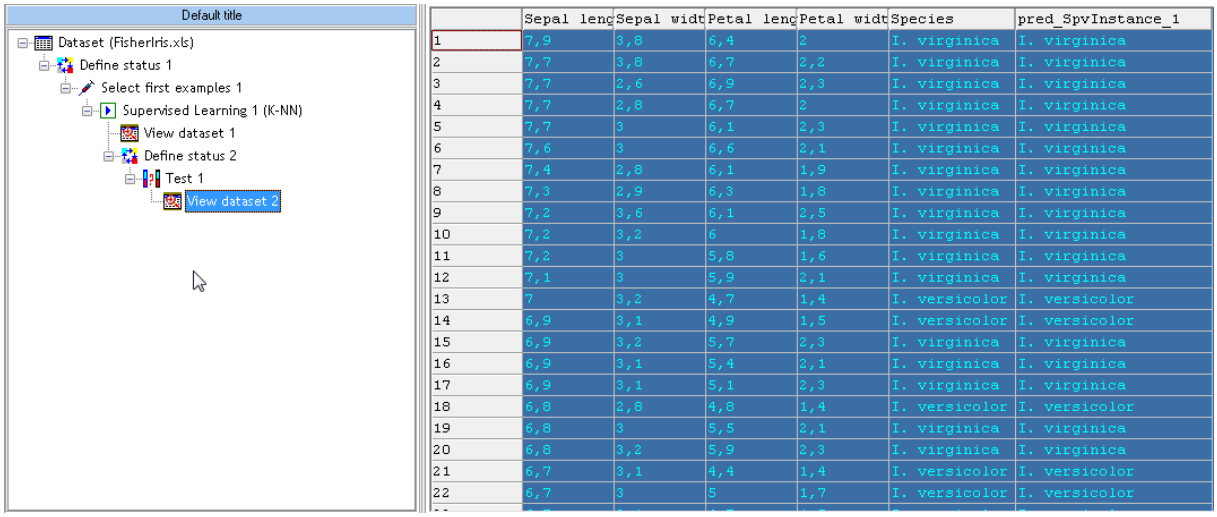
Values prediction

Value	Recall	1-Precision
I. virginica	0.9000	0.1000
I. versicolor	0.9667	0.6375
I. setosa	0.0000	1.0000

Confusion matrix

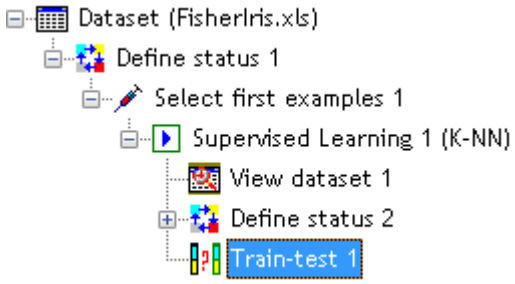
	I. virginica	I. versicolor	I. setosa	Sum
I. virginica	9	1	0	10
I. versicolor	1	29	0	30
I. setosa	0	50	0	50
Sum	10	80	0	90

Et nous rajoutons un composant *View Dataset* pour voir comment les données de test (ou données nouvelles) sont classées:

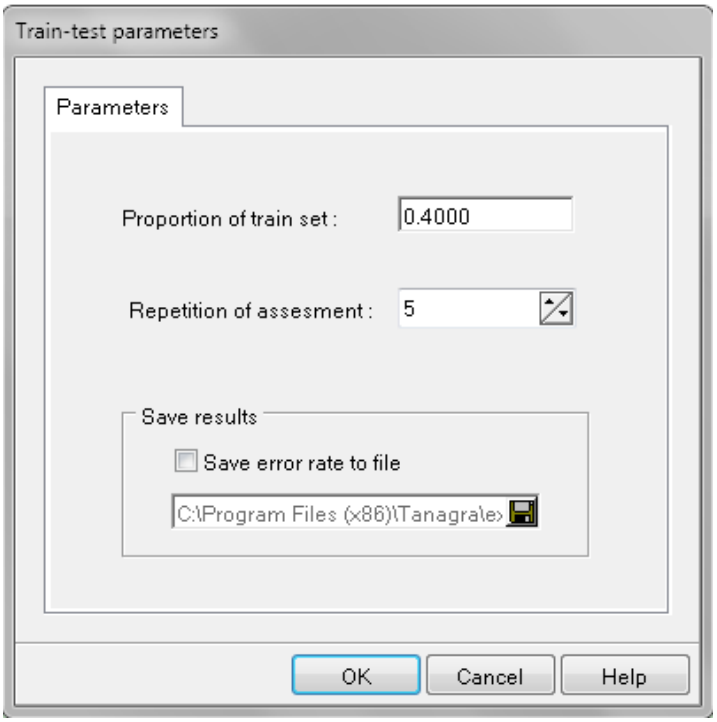


	Sepal leng	Sepal widt	Petal leng	Petal widt	Species	pred_SpvInstance_1
1	7,9	3,8	6,4	2	I. virginica	I. virginica
2	7,7	3,8	6,7	2,2	I. virginica	I. virginica
3	7,7	2,6	6,9	2,3	I. virginica	I. virginica
4	7,7	2,8	6,7	2	I. virginica	I. virginica
5	7,7	3	6,1	2,3	I. virginica	I. virginica
6	7,6	3	6,6	2,1	I. virginica	I. virginica
7	7,4	2,8	6,1	1,9	I. virginica	I. virginica
8	7,3	2,9	6,3	1,8	I. virginica	I. virginica
9	7,2	3,6	6,1	2,5	I. virginica	I. virginica
10	7,2	3,2	6	1,8	I. virginica	I. virginica
11	7,2	3	5,8	1,6	I. virginica	I. virginica
12	7,1	3	5,9	2,1	I. virginica	I. virginica
13	7	3,2	4,7	1,4	I. versicolor	I. versicolor
14	6,9	3,1	4,9	1,5	I. versicolor	I. versicolor
15	6,9	3,2	5,7	2,3	I. virginica	I. virginica
16	6,9	3,1	5,4	2,1	I. virginica	I. virginica
17	6,9	3,1	5,1	2,3	I. virginica	I. virginica
18	6,8	2,8	4,8	1,4	I. versicolor	I. versicolor
19	6,8	3	5,5	2,1	I. virginica	I. virginica
20	6,8	3,2	5,9	2,3	I. virginica	I. virginica
21	6,7	3,1	4,4	1,4	I. versicolor	I. versicolor
22	6,7	3	5	1,7	I. versicolor	I. versicolor

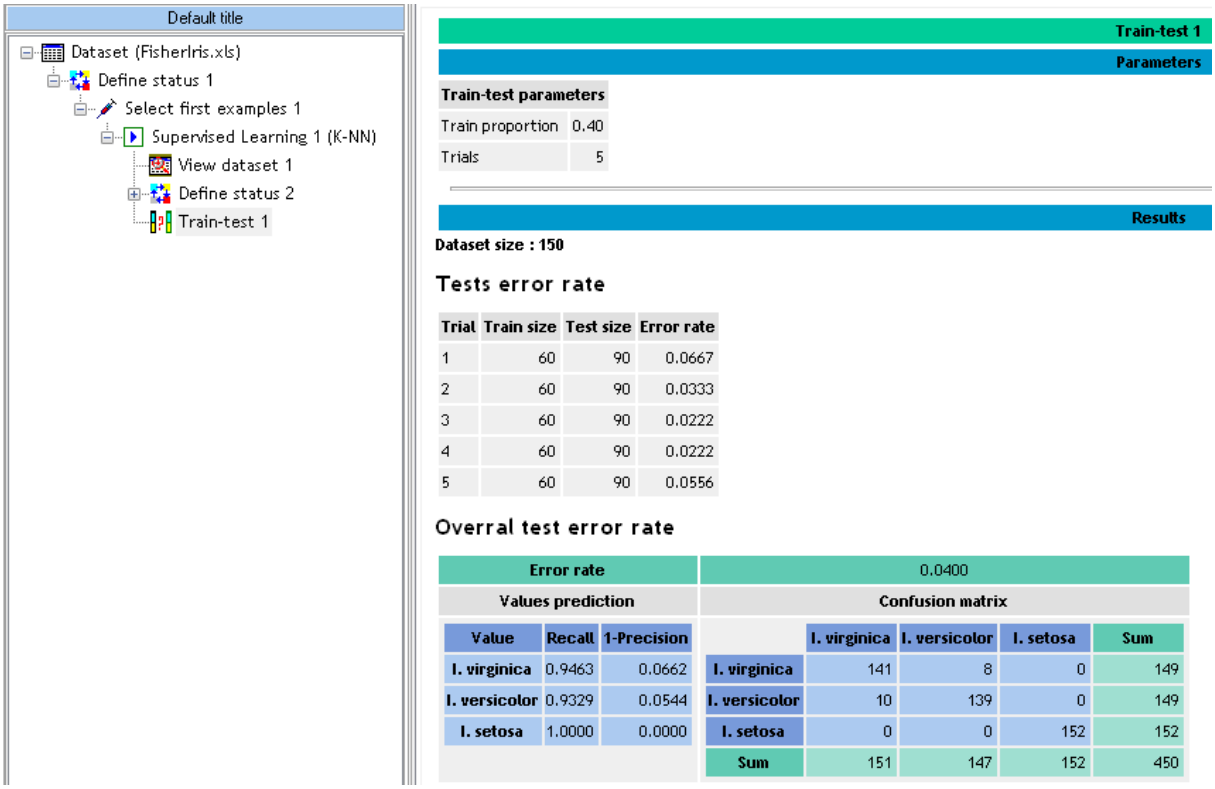
Nous pouvons aussi rajouter un opérateur *Train Test* du groupe *Spv learning assessment*:



et dans les paramètres de cet opérateur, nous prenons:



et en exécutant l'opérateur, nous obtenons:



Train-test 1

Parameters

Train-test parameters	
Train proportion	0.40
Trials	5

Results

Dataset size : 150

Tests error rate

Trial	Train size	Test size	Error rate
1	60	90	0.0667
2	60	90	0.0333
3	60	90	0.0222
4	60	90	0.0222
5	60	90	0.0556

Overral test error rate

Error rate	0.0400

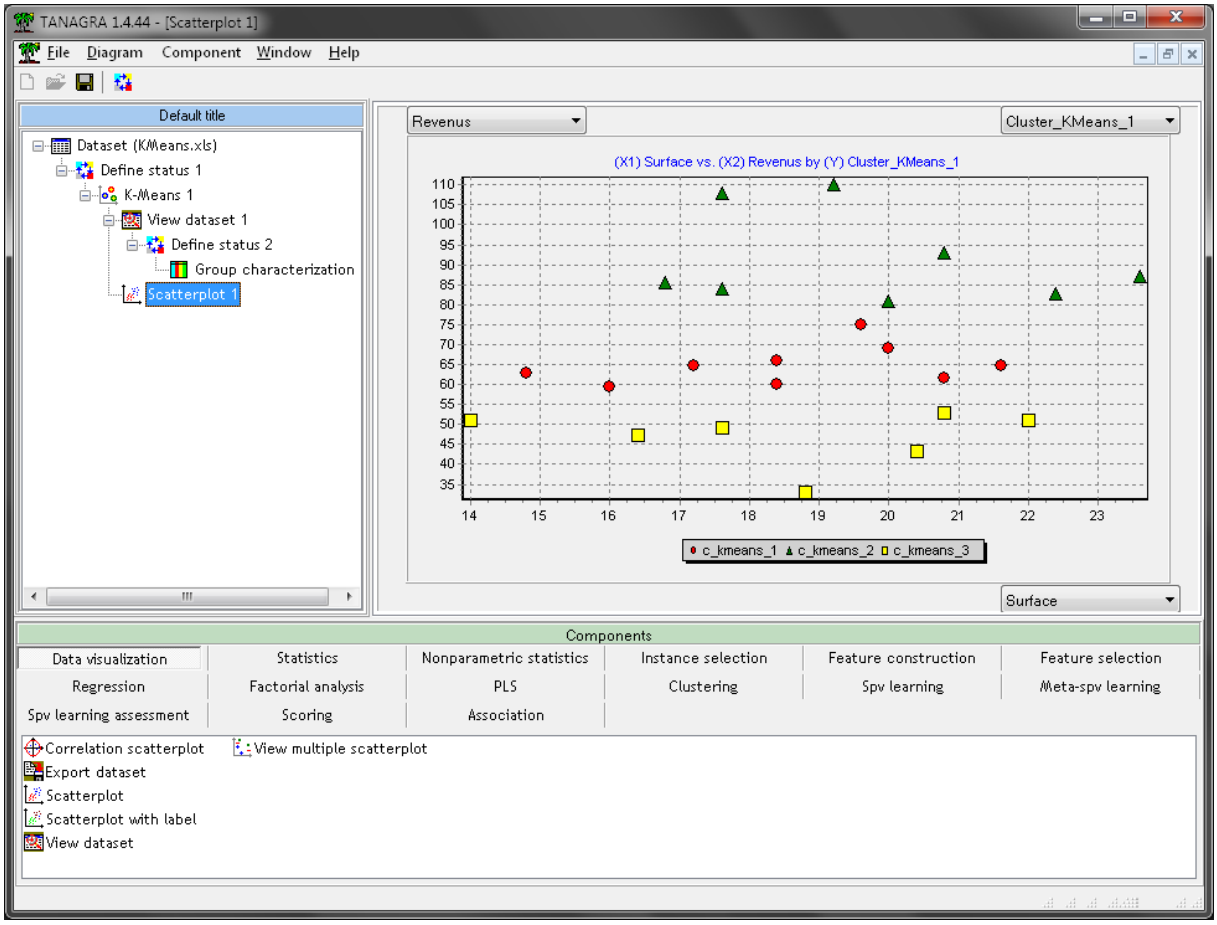
Values prediction

Value	Recall	1-Precision
I. virginica	0.9463	0.0662
I. versicolor	0.9329	0.0544
I. setosa	1.0000	0.0000

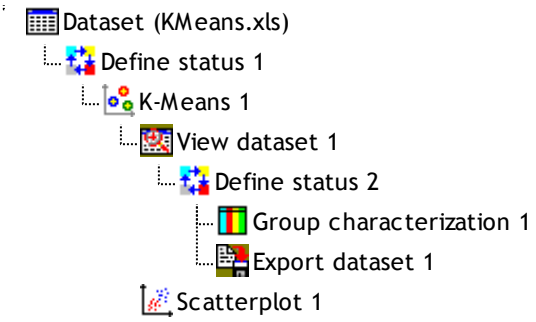
Confusion matrix

	I. virginica	I. versicolor	I. setosa	Sum
I. virginica	141	8	0	149
I. versicolor	10	139	0	149
I. setosa	0	0	152	152
Sum	151	147	152	450

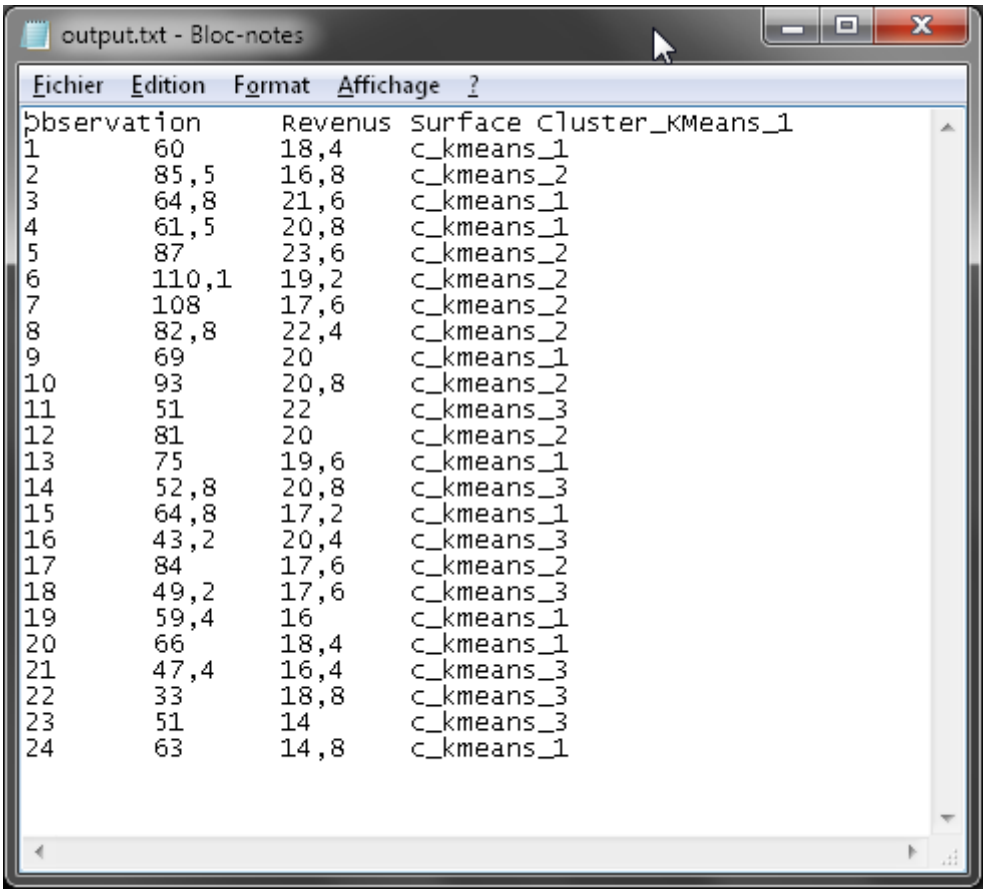
Exercice 20.: Classificaiton K-Means:



Nous mettons le composant **Export Dataset**du groupe **Data visulaziation** en prenant bien soir de la mette après un sélecteur **Define status**:



Une fois que nous l'exécutons en faisant un double clic dessus, nous obtenons un fichier *.txt dans le dossier du fichier Tanagra:


output.txt - Bloc-notes

Fichier Edition Format Affichage ?

Observation	Revenus	Surface	Cluster_KMeans_1
1	60	18,4	c_kmeans_1
2	85,5	16,8	c_kmeans_2
3	64,8	21,6	c_kmeans_1
4	61,5	20,8	c_kmeans_1
5	87	23,6	c_kmeans_2
6	110,1	19,2	c_kmeans_2
7	108	17,6	c_kmeans_2
8	82,8	22,4	c_kmeans_2
9	69	20	c_kmeans_1
10	93	20,8	c_kmeans_2
11	51	22	c_kmeans_3
12	81	20	c_kmeans_2
13	75	19,6	c_kmeans_1
14	52,8	20,8	c_kmeans_3
15	64,8	17,2	c_kmeans_1
16	43,2	20,4	c_kmeans_3
17	84	17,6	c_kmeans_2
18	49,2	17,6	c_kmeans_3
19	59,4	16	c_kmeans_1
20	66	18,4	c_kmeans_1
21	47,4	16,4	c_kmeans_3
22	33	18,8	c_kmeans_3
23	51	14	c_kmeans_3
24	63	14,8	c_kmeans_1