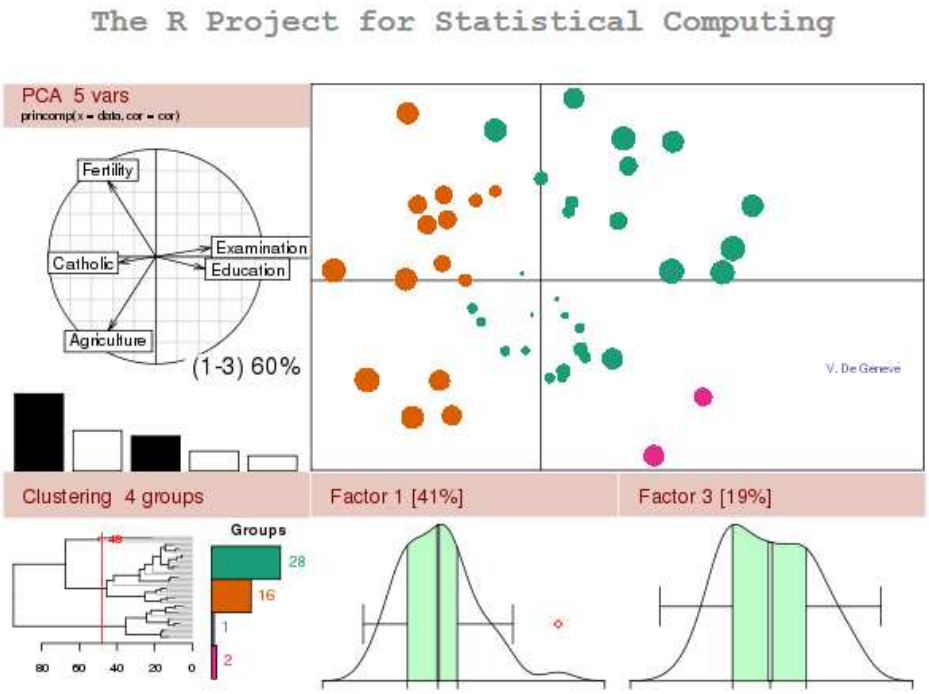
Statistiques descriptives avec Rcommander



Page d'accueil du site http://www.r-project.org/

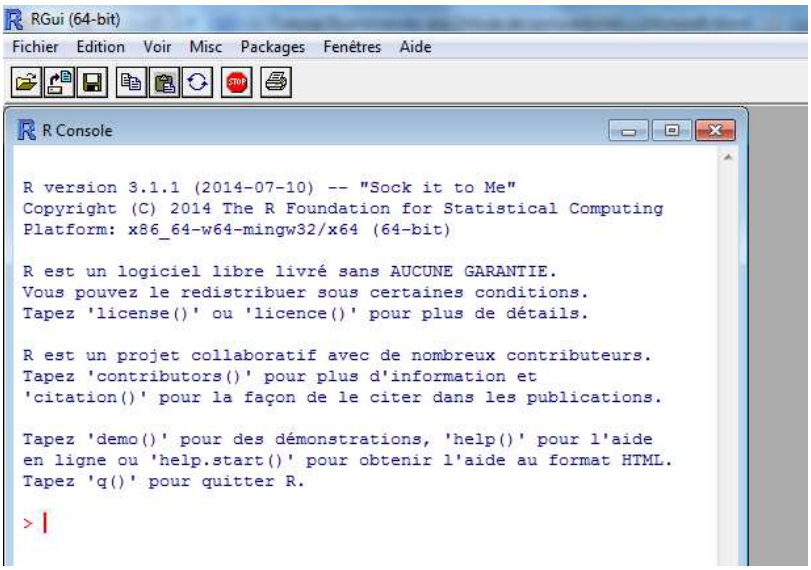
Sommaire

Introduction et installation

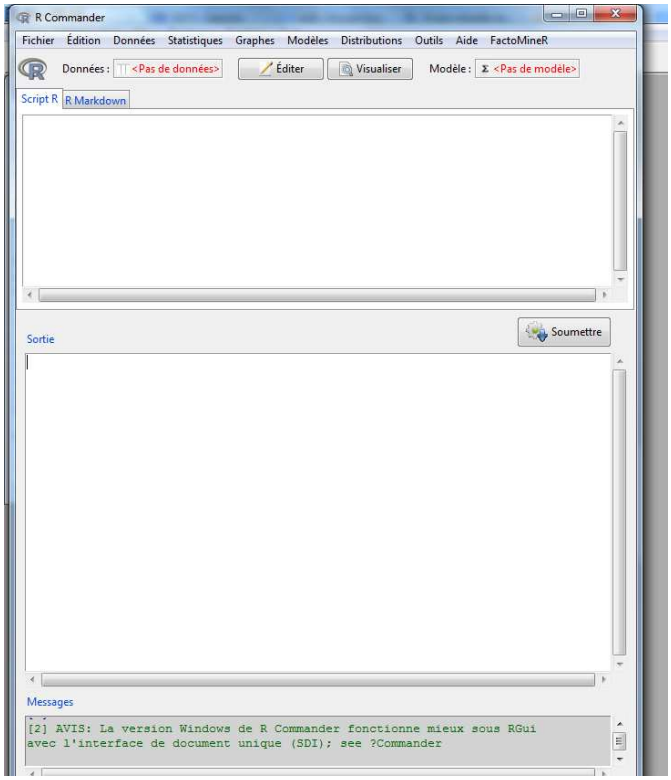
R est un langage et un environnement dédié aux statistiques et aux graphiques associés. Il est collaboratif, libre (GNU Public licence) et s'installe aussi bien sous Linux, Windows ou MacOS.

- Utilisé avec son interface RGui, R est un langage de programmation orienté objet dynamique,

```
> a=matrix(0,3,2)

> a
     [,1] [,2]
[1,]    0    0
[2,]    0    0
[3,]    0    0

> is.matrix(a)
[1] TRUE

> a+2
     [,1] [,2]
[1,]    2    2
[2,]    2    2
[3,]    2    2
```



- Utilisé avec l'interface Rcommander (Rcmdr), R devient un logiciel « clique bouton » type tableur.



On peut étendre la version de base de R grâce à des *packages* téléchargeables sur le site R cran project (Rcommander est lui-même un package de R). Il est aussi possible de programmer ses propres fonctions, soit directement en R (programmation matricielle type Matlab), soit en C/Fortran pour plus de performances.

Dans le cadre des statistiques descriptives de ce cours, nous avons besoin des packages Rcmdr et FactoMineR.

Pour ajouter ces packages à la version de base de R :

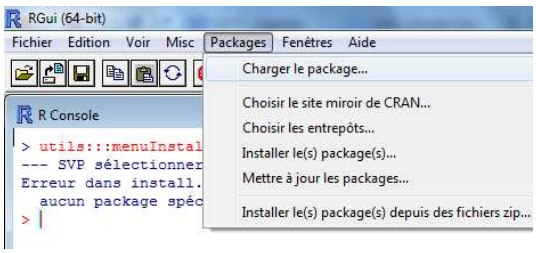
- aller dans le menu RGui :
  Packages > Installer le(s) package(s).
- Choisir un site de téléchargement sur C
  mirror.
- Sélectionner le packages. Ils sont rangés
  ordre alphabétique

ou bien taper l'instruction suivante dans la console :

```
install.packages("nom du package",dependencies = TRUE)
```

Pour charger ces packages dans une session :

- aller dans le menu RGui :
  Packages > Charger le package
- sélectionner le package



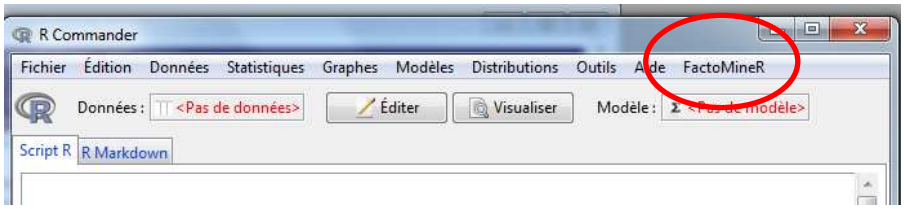
ou bien taper l'instruction suivante dans la console :

```
library(nom du package)
```

Pour intégrer le package FactoMineR dans Rcommander, taper l'instruction suivante dans la console.

```
source("http://factominer.free.fr/install-facto.r")
```

Recharger le package Rcommander. L'onglet FactoMineR apparait dans le menu.

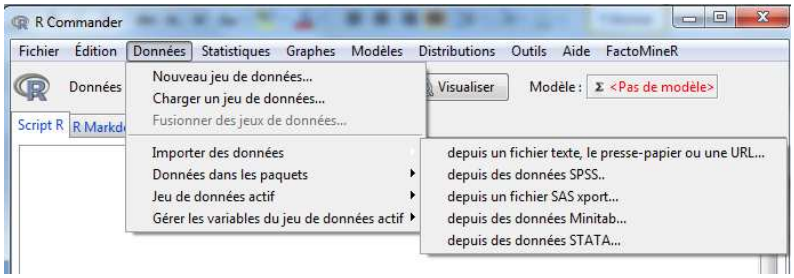


Utilisation de R commander

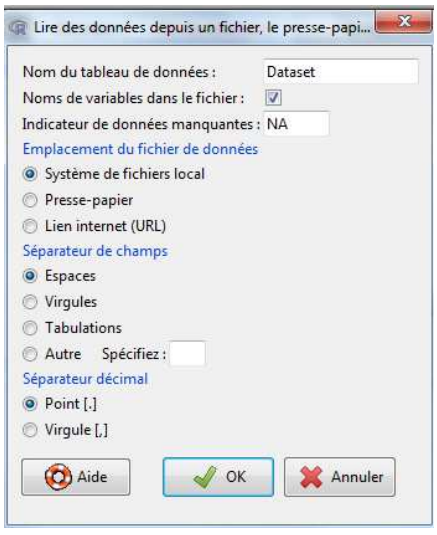
Importer des données

- Changer le répertoire courant

- Aller dans le menu : Données > Importer des données > depuis un fichier texte



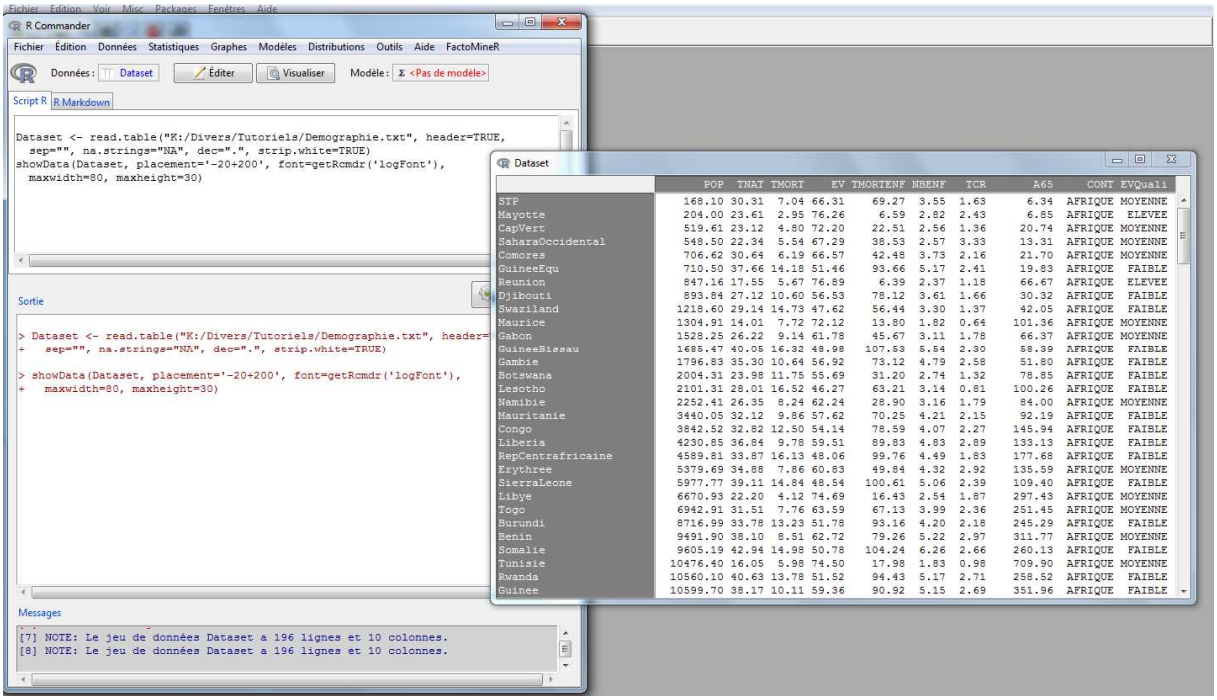
- Préciser le séparateur de champs, le système décimal …



Visualiser les données

Afin de savoir si les données ont été importées correctement, il faut les visualiser. Il suffit pour cela de cliquer sur Visualiser et le tableau s'ouvre dans une nouvelle fenêtre.

Vérifier que les noms des lignes et des colonnes en gris sont corrects.



Les différentes fenêtres de Rcommander

Il y a 3 fenêtres d'affichage dans Rcommander.

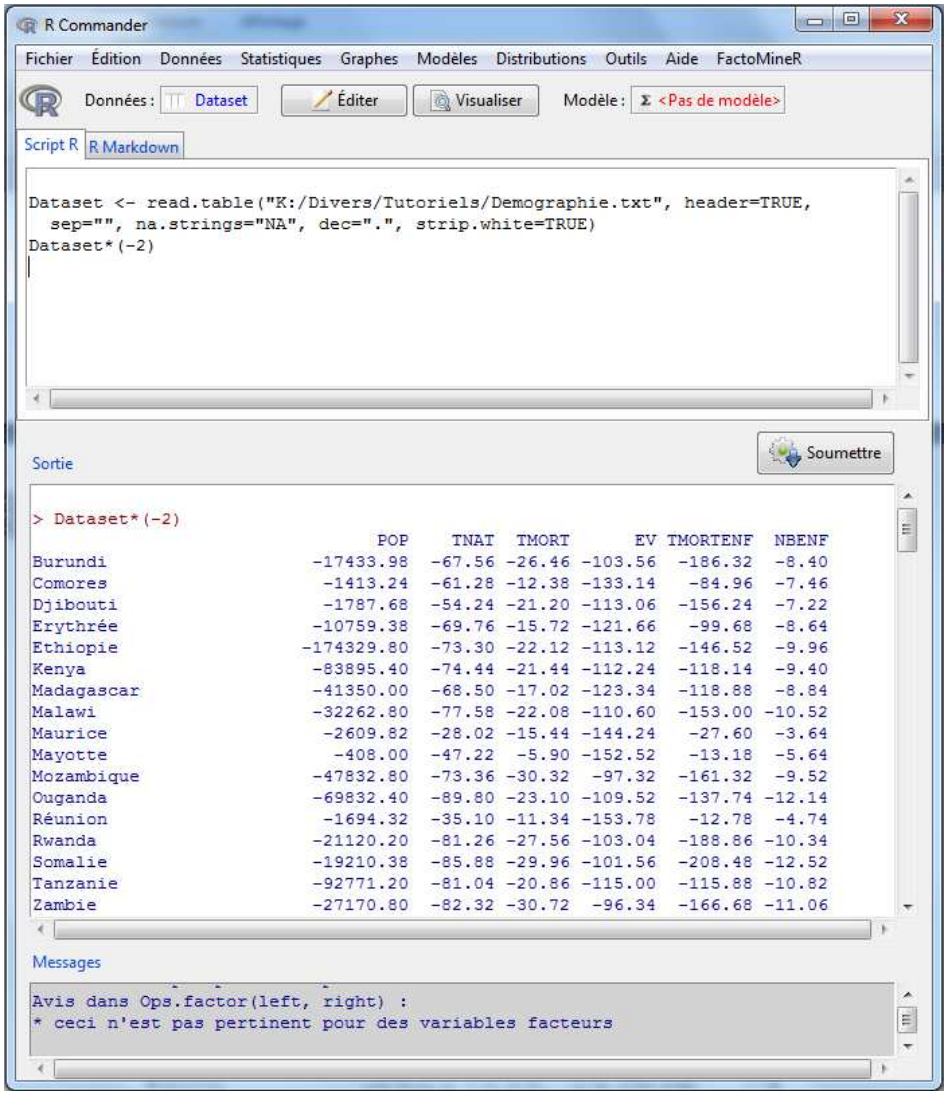
- La fenêtre Script dans laquelle s'affichent toutes les lignes de code générées par une action du menu. Il est possible de taper ses propres commandes dans cette fenêtre. Il faut alors cliquer sur « soumettre » pour qu'elles soient réalisées.
  Dans l'exemple ci-dessous, l'importation des données a généré la commande

  ```
  Dataset <- read.table("K:/Divers/Tutoriels/Demographie.txt",....
  ```

  Comme son nom l'indique la commande `read.table` permet d'importer un tableau de données. `Dataset` est le nom du tableau importé.

- La fenêtre « Sortie » affiche les résultats des commandes du script. Sur l'exemple ci-dessous le résultat de `Dataset*(-2)` s'affiche en bleu.

- La fenêtre « Messages » affiche des informations du type erreurs ou avertissement. Dans l'exemple, nous sommes prévenus qu'il n'est pas pertinent de multiplier par deux le tableau de données puisque celui-ci contient des variables qualitatives (facteurs)


R Commander
Fichier Édition Données Statistiques Graphes Modèles Distributions Outils Aide FactoMineR
Données : Dataset
Éditer
Visualiser
Modèle : <Pas de modèle>
Script R
R Markdown
Dataset <- read.table("K:/Divers/Tutoriels/Demographie.txt", header=TRUE,
sep="", na.strings="NA", dec=".", strip.white=TRUE)
Dataset*(-2)
Soumettre
Sortie
> Dataset*(-2)
POP TNAT TMORT EV TMORTENF NBENF
Burundi -17433.98 -67.56 -26.46 -103.56 -186.32 -8.40
Comores -1413.24 -61.28 -12.38 -133.14 -84.96 -7.46
Djibouti -1787.68 -54.24 -21.20 -113.06 -156.24 -7.22
Erythrée -10759.38 -69.76 -15.72 -121.66 -99.68 -8.64
Ethiopie -174329.80 -73.30 -22.12 -113.12 -146.52 -9.96
Kenya -83895.40 -74.44 -21.44 -112.24 -118.14 -9.40
Madagascar -41350.00 -68.50 -17.02 -123.34 -118.88 -8.84
Malawi -32262.80 -77.58 -22.08 -110.60 -153.00 -10.52
Maurice -2609.82 -28.02 -15.44 -144.24 -27.60 -3.64
Mayotte -408.00 -47.22 -5.90 -152.52 -13.18 -5.64
Mozambique -47832.80 -73.36 -30.32 -97.32 -161.32 -9.52
Ouganda -69832.40 -89.80 -23.10 -109.52 -137.74 -12.14
Réunion -1694.32 -35.10 -11.34 -153.78 -12.78 -4.74
Rwanda -21120.20 -81.26 -27.56 -103.04 -188.86 -10.34
Somalie -19210.38 -85.88 -29.96 -101.56 -208.48 -12.52
Tanzanie -92771.20 -81.04 -20.86 -115.00 -115.88 -10.82
Zambie -27170.80 -82.32 -30.72 -96.34 -166.68 -11.06
Messages
Avis dans Ops.factor(left, right) :
* ceci n'est pas pertinent pour des variables facteurs

Comparaison de k échantillons

L'objectif est de déterminer si k échantillons proviennent de la même population ou s'il y a une différence significative entre eux. Les k échantillons sont les mesures d'une même variable quantitative observées pour différentes modalités d'une variable qualitative (facteur). Par exemple, est-ce que le taux de mortalité est le même suivant le continent ?

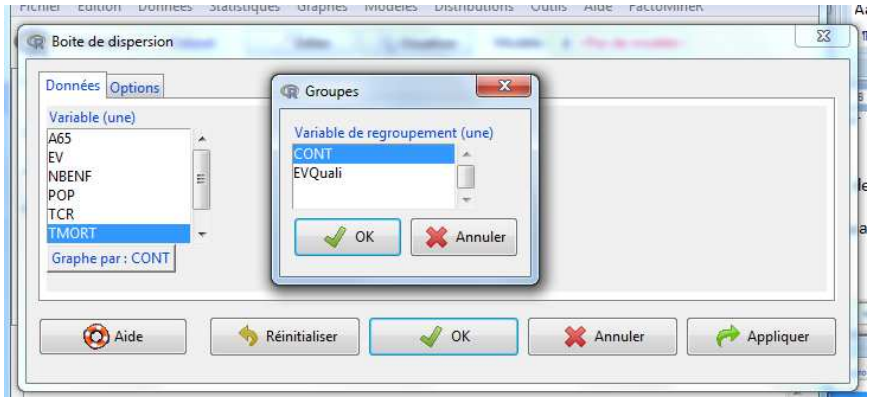
Boxplots conditionnelles

Un premier aperçu graphique permet d'avoir une idée sur le comportement de la variable quantitative en fonction du facteur.

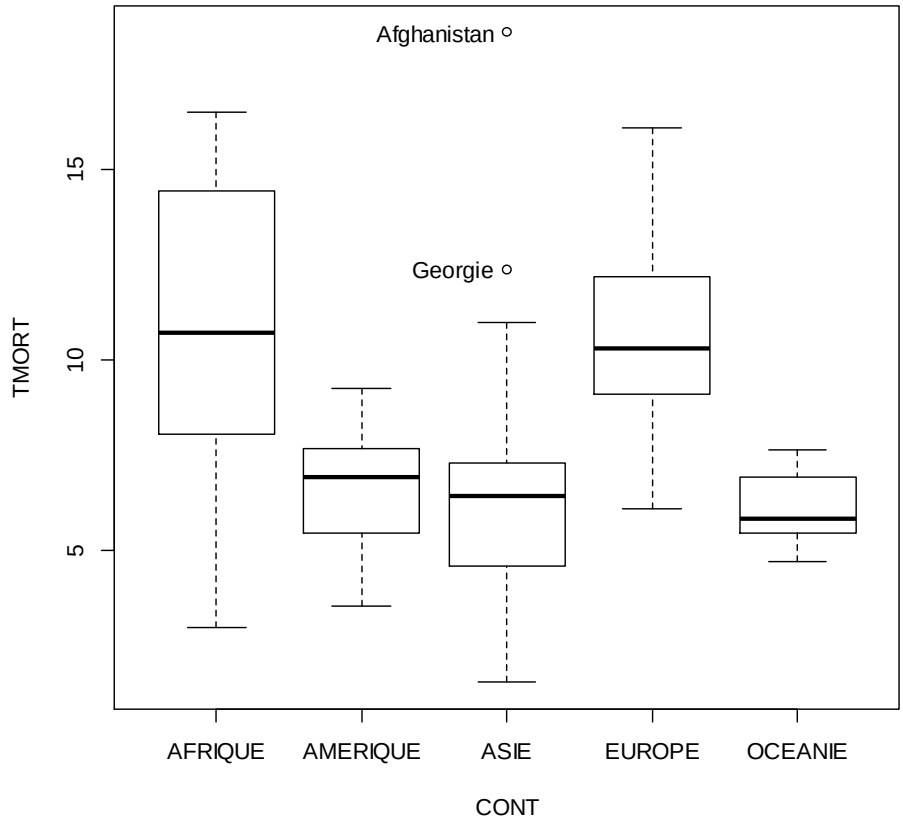
- Aller dans le menu : Graphes > Boites de dispersion

R reconnait automatiquement les variables quantitatives et les variables qualitatives.

- Sélectionner : Graphes par :



Le graphique des boxplots s'affiche dans la fenêtre de R (et non Rcommander)



Analyse de la variance (ANOVA)

L'ANOVA est un test statistique (paramétrique) permettant de décider s'il y a une différence significative entre les moyennes $\mu_1,\ldots,\mu_p$, des sous-populations définies par les p modalités du facteur,

$$\begin{cases} H_0 : \mu_1=\ldots=\mu_p \\ H_1 : \exists i,j,\ \mu_i \neq \mu_j \end{cases}$$

Conditions d'application

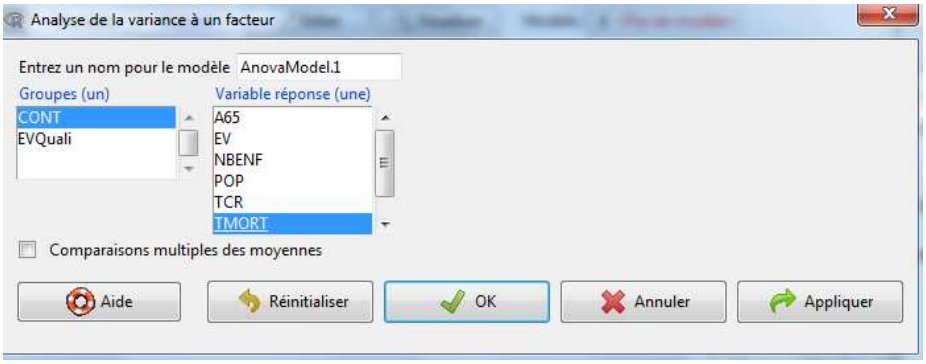
Les variables aléatoires sur chaque sous-population sont gaussiennes de variance constante :

$$X_i \sim N(\mu_i,\sigma^2).$$

Toutefois le test est robuste à la non-normalité des variables surtout quand les k échantillons sont grands.

Consulter le cours pour la définition de l'ANOVA.

- Aller dans le menu : Statistiques > Moyennes > ANOVA un facteur



Cela génère plusieurs lignes de code qui donnent en sortie :

- Les ddl, la statistique du test, la p-valeur
- Un résumé des moyennes et écart-types par sous-population

Ici la p-valeur est inférieure à 2e-16 donc H_0 est très largement rejetée. On considère donc qu'il y a une différence signicative du taux de mortalité suivant le continent.

```
> summary(AnovaModel.1)
             Df Sum Sq Mean Sq F value Pr(>F)
CONT          4  927.5  231.88   29.53 <2e-16 ***
Residuals   191 1499.9    7.85
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> with(Dataset, numSummary(TMORT, groups=CONT, statistics=c("mean", "sd")))
               mean        sd data:n
AFRIQUE  10.902182 3.7770553     55
AMERIQUE  6.545897 1.4565058     39
ASIE      6.415800 2.9152016     50
EUROPE   10.580750 2.3958547     40
OCEANIE   6.067500 0.8835581     12
```

Remarque : l'instruction `aov` (pour analysis of variance) génère un objet auquel on a donné le nom AnovaModel.2 (modifiable). Pour connaitre les attributs d'un objet, il suffit de soumettre l'instruction

```
attributes(AnovaModel.2)
```

dans la fenêtre de script.

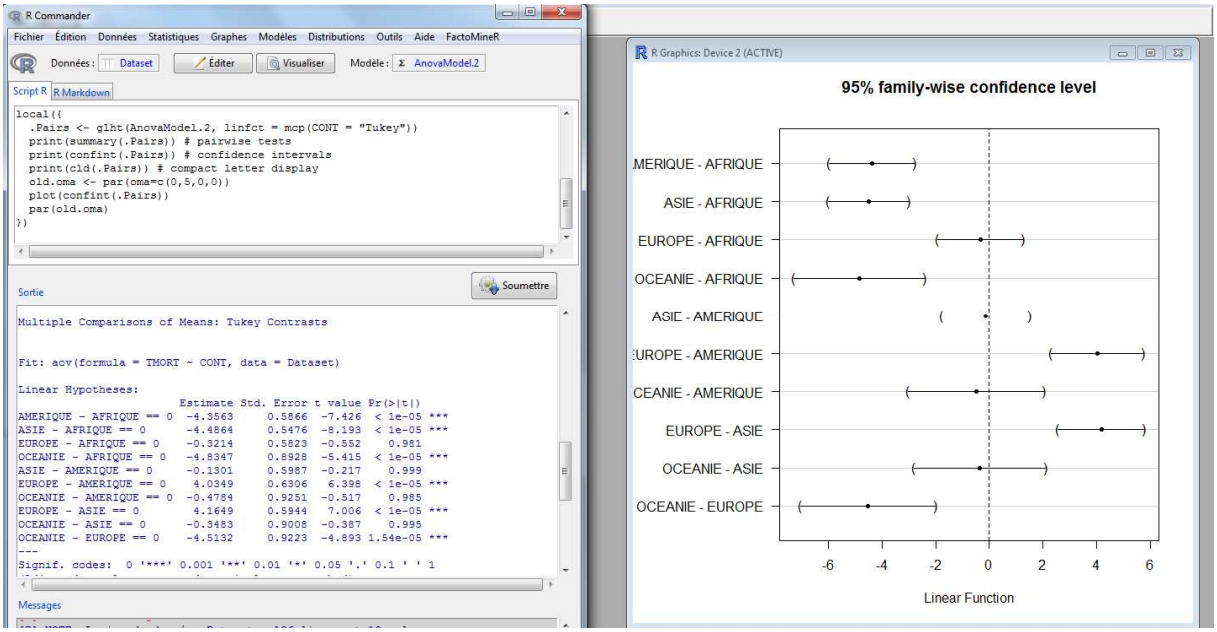
Script R | R Markdown

```
AnovaModel.2 <- aov(TMORT ~ CONT, data=Dataset)
summary(AnovaModel.2)
with(Dataset, numSummary(TMORT, groups=CONT, statistics=c("mean", "sd")))
attributes(AnovaModel.2)
```

Sortie — Soumettre

```
> with(Dataset, numSummary(TMORT, groups=CONT, statistics=c("mean", "sd")))
              mean        sd data:n
AFRIQUE 10.902182 3.7770553     55
AMERNOR  7.685000 0.1909188      2
AMERSUD  6.484324 1.4704658     37
ASIE     6.415800 2.9152016     50
EUROPE  10.580750 2.3958547     40
OCEANIE  6.067500 0.8835581     12

> attributes(AnovaModel.2)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "contrasts"     "xlevels"       "call"          "terms"
[13] "model"

$class
[1] "aov" "lm"
```

Dans le cas où l'hypothèse alternative est acceptée, il peut être intéressant de savoir entre quelle(s) modalité(s) les moyennes sont significativement différentes. On procède alors à un test de comparaison de moyennes deux à deux. Il faut pour cela cocher « comparaisons multiples des moyennes » dans la fenêtre « Analyse de la variance à un facteur ».



Test de Kruskal-Wallis

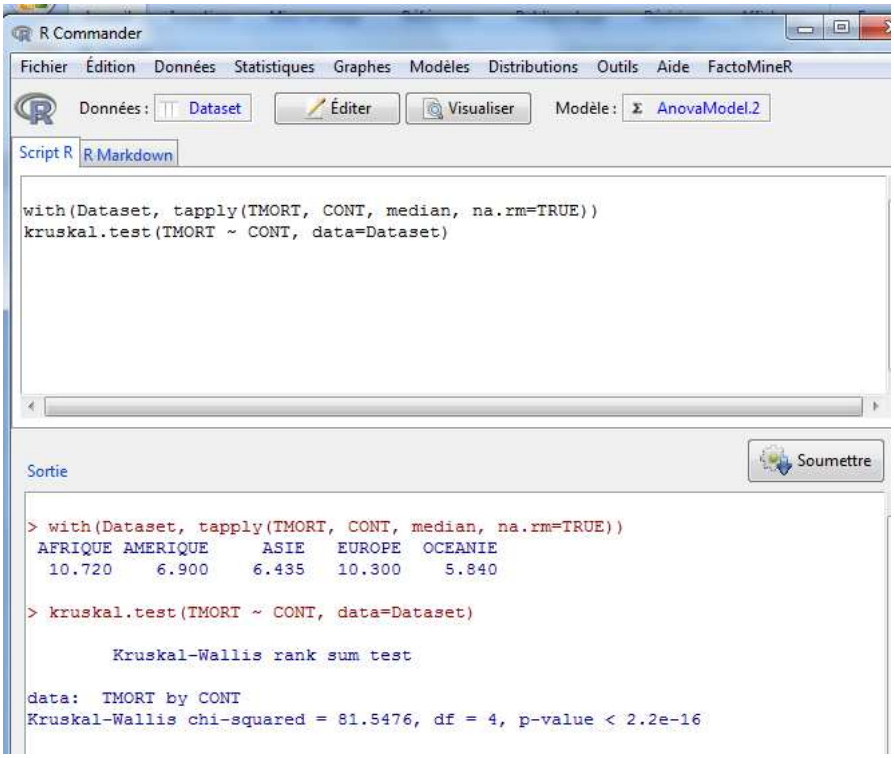
Le test de Kruskal-Wallis est un test statistique (non paramétrique) permettant de décider s'il y a une différence significative entre les lois $L(X_1)$,…, $L(X_p)$ des sous-populations définies par les *p* modalités du facteur,

$$\begin{cases} H_0 : L(X_1)=\ldots=L(X_p) \\ H_1 : \exists i,j,\ L(X_i)\neq L(X_j) \end{cases}$$

La statistique du test (variable de décision) se calcule à partir des rangs attribués aux observations suite au classement des valeurs par ordre croissant (http://www-irma.u-strasbg.fr/~fbertran/enseignement/DUS2_2011/DUS2_CoursNonPara_2.pdf)

Conditions d'application
Aucune (mise à part l'indépendance des échantillons)

- Aller dans le menu : Statistiques > Tests non paramétriques > Test de Kruskal Wallis



Quel test choisir ?

Si les conditions d'application sont vérifiées il est préférable d'utiliser le test de l'ANOVA car plus puissant que celui de Kruskal-Wallis.

En revanche lorsque les échantillons sont petits et que les conditions ne peuvent pas être vérifiées, on utilisera le test de Kruskal-Wallis. Par exemple, lorsqu'on souhaite comparer les performances de plusieurs algorithmes, la mesure de performance (CPU, précision, …) suit

une loi très éloignée de la loi normale (exponentielle, Weibull,…). La moyenne n'est alors pas un indicateur très pertinent car non robuste aux valeurs extrême. On utilisera donc le test de Kruskal-Wallis.

Régression linéaire

L'objectif est de construire un modèle prédictif permettant de modéliser une variable quantitative continue, Y , en fonction d'autres variables quantitatives, $X_1,\ldots,X_p$ sous la forme

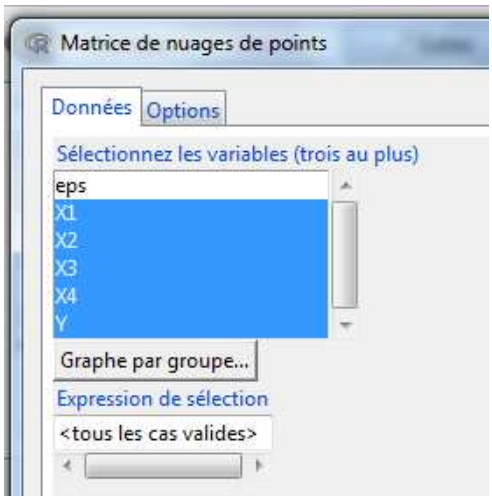
$$Y=a_0+a_1X_1+\ldots+a_pX_p+\varepsilon$$

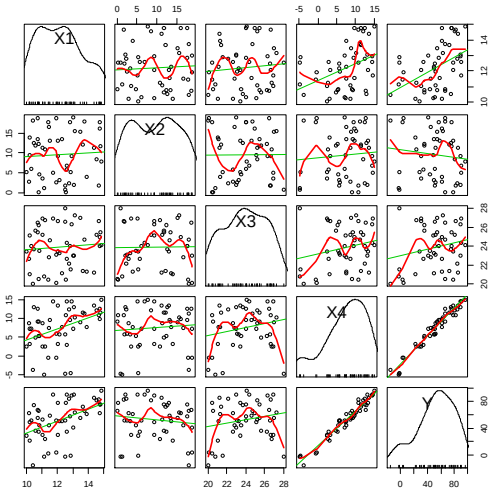
où $\varepsilon \sim N(0,\sigma^2)$. La variable Y est appelée la réponse et $X_1,\ldots,X_p$ les variables explicatives

Nuages de points

Un premier aperçu graphique permet d'avoir une idée sur un lien entre la réponse Y et les variables explicatives prises une à une.

- Aller dans le menu Graphes > Matrice de nuages de points

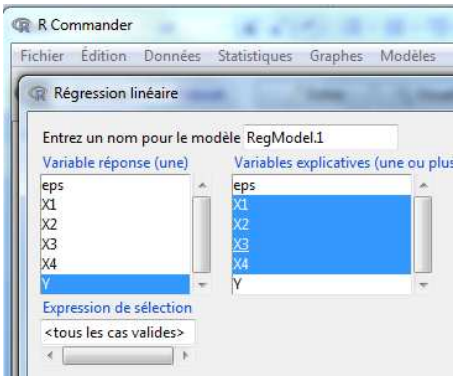


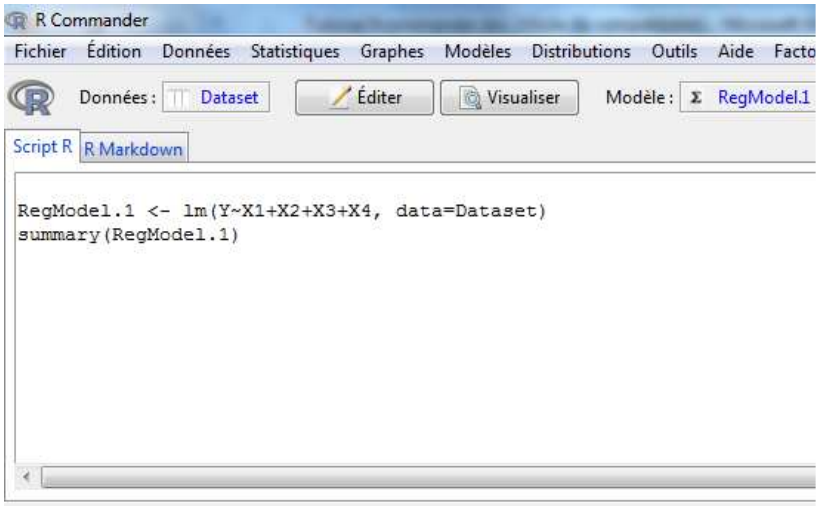


L'instruction pour générer ce type de graphique est `scatterplotMatrix`.

Construction du modèle

- Aller dans le menu Statistiques > Ajustement de modèles > Régression linéaire





La commande est `lm` pour Linear Model. L'objet construit ici s'appelle `RegModel.1`. Pour avoir les résultats l'instruction est `summary(`*`objet`*`)`. …

```
Sortie
Call:
lm(formula = Y ~ X1 + X2 + X3 + X4, data = Dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-2.53539 -0.61223  0.04626  0.50989  1.96800

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.52779    1.80794   1.951   0.0573 .
X1           1.83838    0.09939  18.496   <2e-16 ***
X2          -0.95994    0.02211 -43.423   <2e-16 ***
X3          -0.07916    0.05969  -1.326   0.1915
X4           5.02062    0.02741 183.174   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9148 on 45 degrees of freedom
Multiple R-squared:  0.999, Adjusted R-squared:  0.9989
F-statistic: 1.149e+04 on 4 and 45 DF,  p-value: < 2.2e-16
```

La colonne *Estimate* donne les coefficients de la régression. *Intercept* est la constante. La dernière colonne donne les résultats des tests de Student.

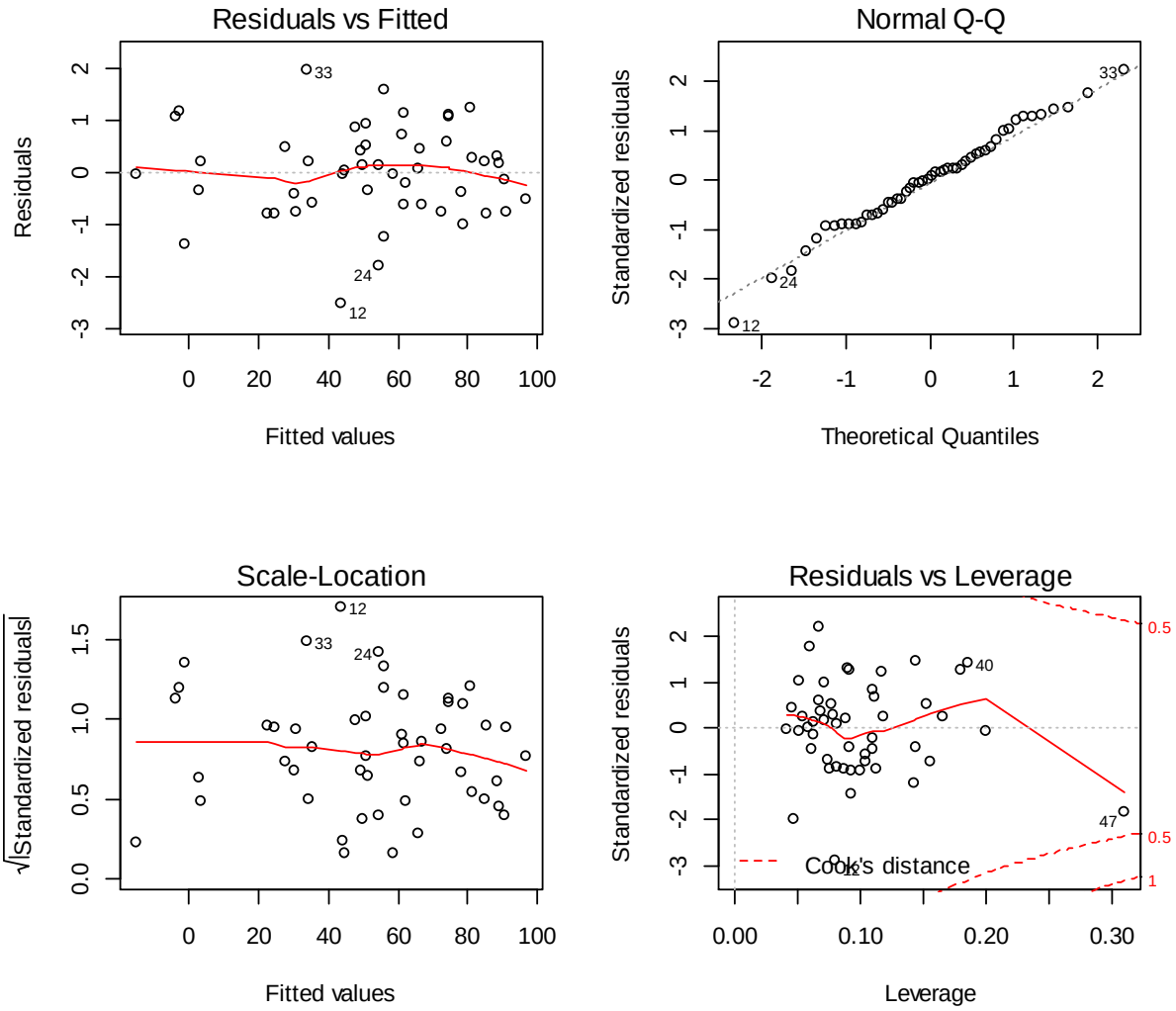
Le dernier paragraphe donne des indicateurs globaux comme le test de Fisher ou le coefficient de détermination R^2.

Les attributs de l'objet sont :

```
> attributes(RegModel.1)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"


> RegModel.1$fitted.values
         1          2          3          4          5          6
34.2269844 74.5898285 49.8471962 90.4507429 65.9990440 50.7005028
         7          8          9         10         11         12
66.5587577 55.8231396  2.8483800 -15.2184365 22.7905635 43.8619834
        13         14         15         16         17         18
24.5876048 91.0335951 78.9014837 61.8653966 62.4222364 58.6127271
```

- Aller dans le menu Modèles pour avoir des informations sur le modèle construit et notamment des graphiques sur les résidus.

lm(Y ~ X1 + X2 + X3 + X4)



Remarque 1 : Il existe des algorithmes de sélection de modèle (`stepwise`) disponible dans le menu Modèles. Cependant, il est conseillé de faire la sélection soi-même.

Remarque 2 : On peut ajouter d'autres termes dans le modèle comme par exemple :

```
RegModel.2 <- lm(Y~X1+X2+X3+X4+X1*X2+I(X1^2), data=Dataset)

summary(RegModel.2)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.183309  10.205409   2.272   0.0282 *
X1          -1.347651   1.634057  -0.825   0.4141
X2          -0.984469   0.206556  -4.766 2.17e-05 ***
X3          -0.079027   0.058512  -1.351   0.1839
X4           5.012756   0.027176 184.458  < 2e-16 ***
I(X1^2)      0.127948   0.066299   1.930   0.0602 .
X1:X2        0.001719   0.016821   0.102   0.9191
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8967 on 43 degrees of freedom
Multiple R-squared:  0.9991, Adjusted R-squared:  0.999
F-statistic:  7972 on 6 and 43 DF,  p-value: < 2.2e-16
```

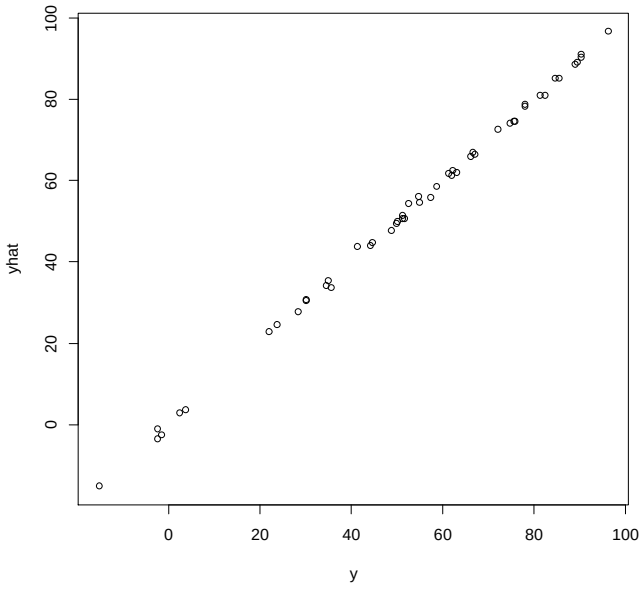
Prévision

- La prévision ne se fait pas directement. Il faut taper l'instruction :

```
predict.lm(model,newdata,interval='confidence')
```

Si l'argument newdata n'est pas renseigné, il fait la prévision aux points observés. Par exemple, taper dans la fenêtre de script :

```
y=Dataset[,1]
yhat=predict.lm(RegModel.1)
plot(y,yhat)
```



L'argument newdata doit être de la classe dataframe et le nom des colonnes doit correspondre aux noms des variables. Par exemple, taper dans la fenêtre de script :

```
newdata=matrix(runif(5*4),5,4)        # runif(n) génère n random
newdata
newdata=as.data.frame(newdata)
```

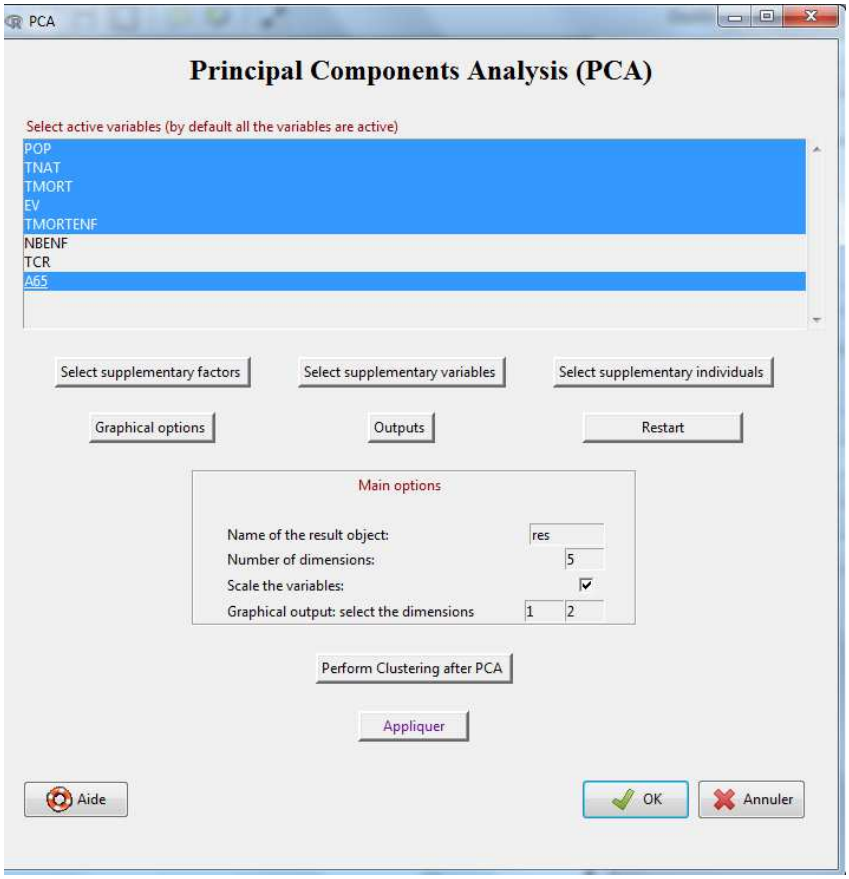
```
names(newdata)=c("X1","X2","X3","X4")
newdata
predict.lm(RegModel.1,newdata,interval="confidence")
```

```
> newdata
          X1        X2        X3        X4
1 0.44957673 0.0936427 0.8167446 0.8726096
2 0.31369145 0.8902276 0.2878036 0.1911907
3 0.08250587 0.9879956 0.1534465 0.4052449
4 0.03899360 0.9569071 0.8850801 0.9386386
5 0.80672764 0.8347900 0.3454870 0.7976580


> predict.lm(RegModel.1,newdata,interval="confidence")
       fit       lwr       upr
1 8.580776 5.0551951 12.106356
2 4.187016 0.6116504  7.762381
3 4.753476 1.1338284  8.373124
4 7.323378 3.7553070 10.891449
5 8.186896 4.6659598 11.707833
```

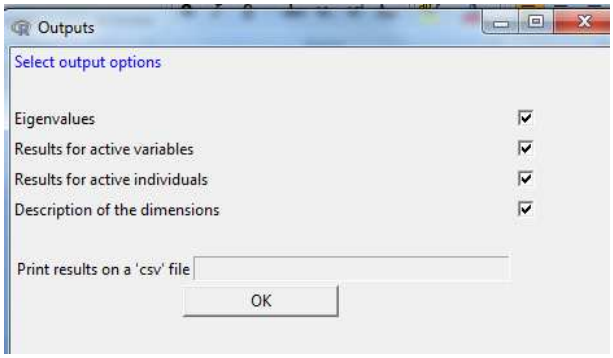
Analyse des composantes principales

L'objectif est représenter en dimension 2 un nuage de points décrits par *p* variables continues et d'interpréter le graphique.

- Aller dans le menu FactoMineR > Principal Component Analysis (PCA)



- Dans l'onglet Outputs sélectionner toutes les sorties



L'instruction est PCA et le résultat est un objet appelé `res` (modifiable dans la fenêtre main options) avec les attributs suivants

```
> attributes(res)
$names
[1] "eig"  "var"  "ind"  "svd"  "call"

$class
[1] "PCA"   "list "
```

`res$eig` contient les informations sur les valeurs propres, `res$var` les informations sur les variables et `res$ind` les informations sur les individus.

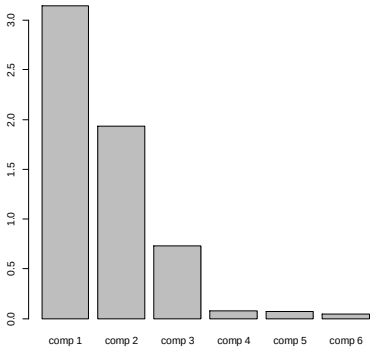
Analyse des valeurs propres

Dans `res$eig` , on a les composantes principales en ligne et les valeurs propres, la variance expliquée, la variance cumulée en colonne.

```
> res$eig
       eigenvalue percentage of variance cumulative percentage of variance
comp 1 3.14259386              52.3765643                            52.37656
comp 2 1.93054531              32.1757552                            84.55232
comp 3 0.72985811              12.1643018                            96.71662
comp 4 0.07880662               1.3134437                            98.03007
comp 5 0.07206824               1.2011373                            99.23120
comp 6 0.04612785               0.7687975                           100.00000
```

Si on souhaite tracer les valeurs propres, il suffit de taper l'instruction :

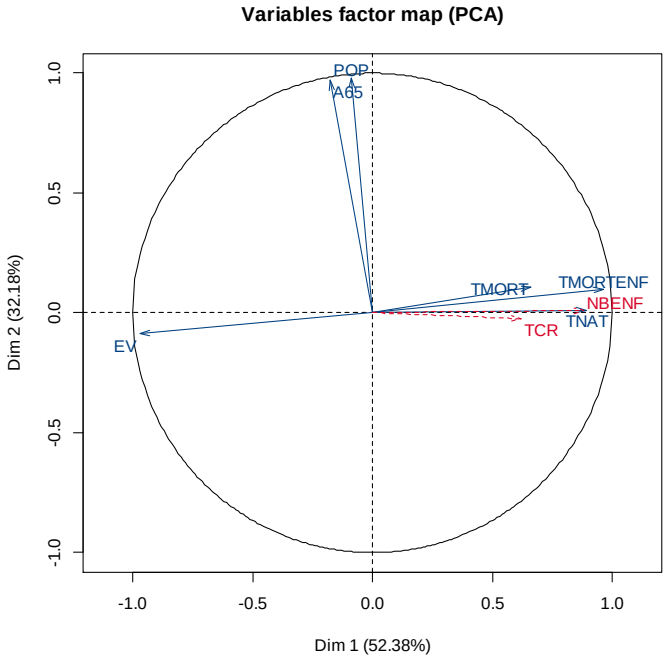
```
barplot(res$eig[,1],
     names.arg=row.names(res$eig))
```



Analyse des variables

Dans `res$var`, on a toutes les informations concernant les variables :

```
> res$var
$coord
               Dim.1       Dim.2       Dim.3       Dim.4       Dim.5
POP       -0.09040872  0.97979890 -0.07929260 -0.07177437 -0.04486245
TNAT       0.89235087  0.01176960 -0.40505713  0.19211500 -0.01889551
TMORT      0.66043471  0.10878725  0.73796331  0.07495697  0.01149027
EV        -0.97180140 -0.08701228  0.01511576  0.10401249  0.17675860
TMORTENF   0.96211180  0.09869088 -0.12012875 -0.11593919  0.19150173
A65       -0.17860282  0.97018308  0.01581973  0.08287248  0.04062044

$cor
               Dim.1       Dim.2       Dim.3       Dim.4       Dim.5
POP       -0.09040872  0.97979890 -0.07929260 -0.07177437 -0.04486245
TNAT       0.89235087  0.01176960 -0.40505713  0.19211500 -0.01889551
TMORT      0.66043471  0.10878725  0.73796331  0.07495697  0.01149027
EV        -0.97180140 -0.08701228  0.01511576  0.10401249  0.17675860
TMORTENF   0.96211180  0.09869088 -0.12012875 -0.11593919  0.19150173
A65       -0.17860282  0.97018308  0.01581973  0.08287248  0.04062044

$cos2
               Dim.1        Dim.2        Dim.3        Dim.4        Dim.5
POP       0.008173736 0.9600058870 0.0062873167 0.005151560 0.0020126391
TNAT      0.796290068 0.0001385234 0.1640712790 0.036908174 0.0003570404
TMORT     0.436174012 0.0118346648 0.5445898489 0.005618548 0.0001320262
EV        0.944397963 0.0075711367 0.0002284862 0.010818598 0.0312436011
TMORTENF 0.925659114 0.0097398888 0.0144309157 0.013441897 0.0366729131
A65       0.031898966 0.9412552128 0.0002502639 0.006867848 0.0016500205

$contrib
               Dim.1        Dim.2       Dim.3      Dim.4       Dim.5
POP        0.2600952 49.727187455  0.86144370  6.536964  2.7926852
TNAT      25.3386248  0.007175351 22.47988707 46.833847  0.4954199
TMORT     13.8794267  0.613021860 74.61585219  7.129538  0.1831962
EV        30.0515436  0.392176070  0.03130557 13.728031 43.3528014
TMORTENF 29.4552575  0.504514904  1.97722208 17.056811 50.8863723
A65        1.0150521 48.755924361  0.03428939  8.714810  2.2895252
```



Le graphique des variables se trouve dans la fenêtre de R.

Il est possible de représenter d'autres dimensions. Il suffit de les choisir dans la fenêtre de PCA, le cadre *main options*, *graphical outputs*.

Il est possible d'ajouter des variables supplémentaires, c'est-à-dire des variables qui ne contribuent pas au calcul des composantes principales. Il suffit de cliquer sur l'onglet *select supplementary variables* dans la fenêtre PCA.

Analyse des individus

Dans `res$ind`, on a toutes les informations concernant les individus :

```
> res$ind
$coord
                          Dim.1        Dim.2       Dim.3        Dim.4
STP                  1.169681259 -0.291100448 -0.93682975 -0.156501658
Mayotte             -1.247246515 -0.594725919 -1.32742798  0.297015954
CapVert             -0.570205914 -0.488965258 -0.93053963  0.038955425
SaharaOccidental     0.024464025 -0.405469196 -0.79449999 -0.356112551
Comores              0.610740680 -0.369080415 -1.03876603  0.168205477
GuineeEqu            3.541645550  0.028086257  0.33730184 -0.008648613
Reunion             -1.296736562 -0.536215359 -0.37737470  0.125134601
```

...

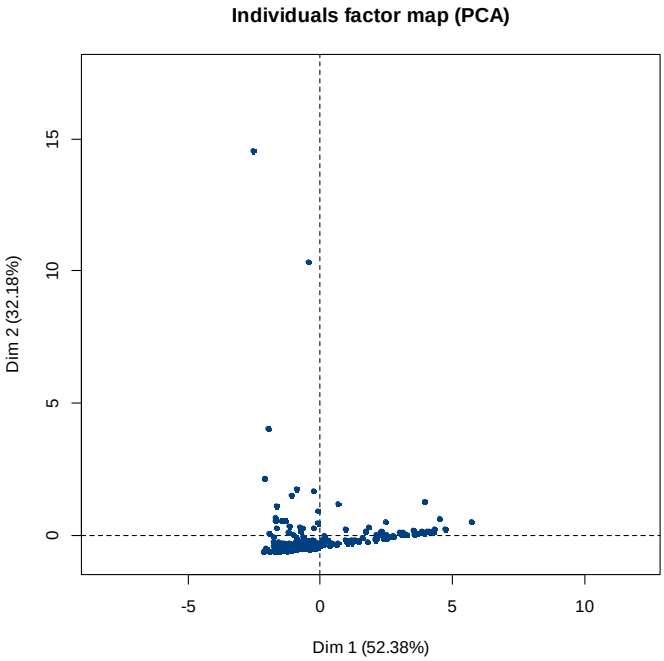
```
$cos2
                          Dim.1        Dim.2        Dim.3        Dim.4
STP                 5.117130e-01 3.169400e-02 0.3282560308 9.160712e-03
Mayotte             4.083779e-01 9.285203e-02 0.4625722951 2.315886e-02
CapVert             2.249502e-01 1.654166e-01 0.5990909775 1.049927e-03
SaharaOccidental    6.119990e-04 1.681166e-01 0.6454798006 1.296789e-01
Comores             2.300566e-01 8.401618e-02 0.6655127912 1.745023e-02
GuineeEqu           9.884147e-01 6.216082e-05 0.0089653333 5.894160e-06
```

...

```
$contrib
                          Dim.1        Dim.2        Dim.3        Dim.4
STP                 2.221216e-01 2.239493e-02 0.6135173394 1.585693e-01
Mayotte             2.525575e-01 9.347547e-02 1.2317637746 5.711376e-01
CapVert             5.278604e-02 6.318586e-02 0.6053063592 9.824650e-03
SaharaOccidental    9.716537e-05 4.344899e-02 0.4412587011 8.210238e-01
Comores             6.055770e-02 3.600028e-02 0.7542945405 1.831730e-01
GuineeEqu           2.036413e+00 2.084739e-04 0.0795321840 4.842550e-04
```

...

```
$dist
                  STP                  Mayotte                  CapVert
            1.6351377                1.9517364                1.2022326
      SaharaOccidental                  Comores                GuineeEqu
            0.9889004                1.2733258                3.5623410
              Reunion                 Djibouti                Swaziland
            1.4635425                2.1993607                3.0793481
               Mauri                     Gabon            GuineeBissau
```



Le graphique des individus se trouve dans la fenêtre de R.

Il est possible d'ajouter des individus supplémentaires, c'est-à-dire des individus qui ne contribuent pas au calcul des composantes principales. Il suffit de cliquer sur l'onglet *select supplementary factors* dans la fenêtre PCA.

Analyse factorielle des correspondances

Analyse discriminante