Année 2018 – 2019

Méthodes Statistiques pour l'Ingénieur – ENTPE
TP Maison : Découverte de R et analyse de données démographiques

Pour les 5 prochains TPs de MSI, vous formerez des binômes à conserver d'une séance à l'autre. Un compte-rendu sous format Word, OpenOffice ou LateX/ps-pdf sera rendu lors de la cinquième séance, qui sera notée. Il comprendra des réponses précises aux questions posées, illustrées par des graphiques et accompagnées de copies du code R correspondant (dans le corps du texte ou en annexe). Il est attendu que vous utilisiez le logiciel R comme support pour argumenter vos réponses. Le barème de notation des travaux pratiques est le suivant :

1. Compréhension du sujet, réponses aux questions posées et recul : 12 points.
2. Qualité du code R et des graphiques fournis : 5 points.
3. Qualité de la rédaction, orthographe : 3 points (retrait supplémentaire de points en cas d'expression/orthographe déplorables).

Il est vivement recommandé de s'appuyer sur les fiches d'aide sous R fournies, mais également la documentation internet. Le site www.duclert.org contient toutes les fonctions utiles pour répondre aux questions.

Ce TP à réaliser à la maison vise à vous permettre de prendre en main le logiciel R et vous familiariser avec son environnement. La correction du *Préambule* et de la *Partie I* sont intégralement fournies. Il est attendu que vous rendiez un rapport succinct et illustré répondant aux questions de la Partie II. Ce rapport doit être remis de façon électronique à vos intervenants de TD, à la rentrée de janvier (**8 janvier dernier délai)**. Des éléments de réponses vous sont fournis, la qualité de synthèse, de présentation et de lisibilité des graphiques constituera le principal critère de notation de ce TP.

Préambule (15 min)

Téléchargez le fichier envoyé « MSI_TPhome.csv », l'installer dans un répertoire de travail de votre choix. Ouvrir le logiciel R et changer le répertoire de travail (setwd(« Path »)) pour ne pas avoir à taper les chemins d'accès complets (vérifiez avec « getwd() »). Naviguez entre les différents menus pour vous approprier l'outil. Observez la richesse des contributions des utilisateurs (apports des packages, écriture fonctionnelle pour la pratique des statistiques…).

Partie I : Prise en main de l'outil (45 min)

R est un langage orienté objet et les différentes structures de données sont stockées dans la mémoire vive de l'ordinateur. Par exemple, en écrivant « *msi = 15* » on crée une variable msi, à laquelle on donne la valeur 15. On accède ensuite à l'objet msi en tapant son nom et la commande « ls() » donne la liste de tous les objets actuellement stockés en mémoire. Vous pouvez à tout moment accéder à l'aide sur une fonction en tapant « help(nom_fonction) ». La fonction « str » donne le type/nature de l'objet.

1. Créer un vecteur composé de 10 entiers. (Aide : la commande est « c » pour construire un vecteur). Multiplier tous les éléments par 10. Vérifier le type de votre vecteur avec la fonction « str ». Testez les fonctions qui donnent la longueur d'un vecteur, sa moyenne, sa variance, qui inversent l'ordre des éléments (« rev »), qui trient en ordre ascendant (« sort », « order » …) etc.

2. Créer deux matrices carrées de taille 2*2 composées des éléments de votre choix. Fusionner ces deux matrices en ligne ou en colonne (aide : « rbind » ou « cbind »). Apprendre la syntaxe pour le produit de matrice.

3. Charger le célèbre jeu de données des Iris de Fisher (aide : « data(iris) »). Observer la puissance du type de données de R « dataframe » en testant par exemple la fonction « summary », puis « plot », et apprenez à accéder à une colonne avec la touche « $ ».

4. Tester les fonctions graphiques de R (aide : « plot ») et voir comment changer les paramètres graphiques, en affichant de différentes façons les espèces des Iris de Fisher. Entraînez-vous à améliorer la présentation d'un graphique :
   i. Titres : main, xlab, ylab
   ii. Couleurs : col
   iii. Fonction « par ». Exécuter par(), que constatez-vous ?
   iv. Voir comment faire une légende : legend()
   v. Associer du texte à des valeurs ou faire apparaître du texte sur le graphe (text())

Partie II : Analyse de données démographiques par pays, pour l'année 2000 (1h)

Tout d'abord, mettre dans un objet le fichier « MSI_ TPhome.csv » à l'aide des fonctions « read.table » ou « read.csv ». Le but de ce TP est de comparer pour 169 pays les variables suivantes (pour l'année 2000) :

- Carbon Dioxide Emissions per capita: HDRO calculations based on data from Boden, Marland and Andres (2009).
- Forest area: UN (2011). ["ResourceSTAT: FAO"]. FAO. Rome.
- GDP per capita in PPP terms (constant 2005 international $): World Bank (2011).
- "World Development Indicators 2011." Washington, D.C.: World Bank. http://data.worldbank.org. Accessed May 15, 2011.
- Health index: HDRO calculations
- Life expectancy at birth: UNDESA (2011).
- Population, total both sexes (thousands): UNDESA (2011).
- Under-five mortality rate: UNICEF (2011).

A vous de faire parler les données !

Proposez une analyse statistique personnelle de ces éléments partiels de la structure démographique et économique du monde en 2000. Répondez en particulier aux questions suivantes :

1. Renommer les colonnes du jeu de données (fonction names()). Regarder le nombre d'individus, le type de chaque variable. Commencer une analyse unidimensionnelle variable par variable.

2. Construire un diagramme en bâtons de la population et un histogramme des hectares de forêt par pays. Que constatez-vous ?

3. Elaborer une boîte à moustaches de l'espérance de vie et du taux de mortalité des moins de 5 ans. Donnez une interprétation des différentes barres composant les boîtes à moustache.

4. Construire une variable richesse en trois catégories : pays riche – pays à revenus moyens – pays pauvre. Expliquer les classes. Les tracer en camembert (diagrammes à secteurs).

5. Centrer-réduire la variable donnant la population totale. Quelles sont les caractéristiques d'une variable centrée-réduite ? les vérifier.

6. Tracer le nuage de points de la variable indice de santé et Carbon Dioxide Emissions. Lien ? Corrélation ? Idem pour indice de santé et espérance de vie.

7. Déterminer les 12 pays qui émettent le plus de CO2. Quel est leur GDP moyen? Déterminer leur émission moyenne de CO2.

Éléments de correction TP maison : Découverte de R

Préambule :

Le fichier MSI_TP1.csv a été fourni. Celui-ci contient des données démographiques qui sont utilisées dans la Partir II de ce TP.

Pour ne pas avoir à écrire le chemin complet d'accès à l'emplacement de vos fichiers, il est d'usage d'indiquer au logiciel R dans quel répertoire travailler. Il faut donc changer le « répertoire courant » de R, pour ne pas avoir à taper les chemins d'accès complets. Le réglage se fait au travers d'une interface dans l'onglet Fichier>Changer le répertoire courant… ou en ligne de commande par :

```
> setwd("C:/Users /Desktop")
> getwd()
[1] "C:/Users /Desktop"
```

La fonction setwd(arg) (comme *set working directory*) prend comme argument le chemin d'accès au dossier entre guillemets comme il s'agit d'une chaîne de caractère, et la commande getwd() (comme *get working directory*) renvoie le répertoire courant. La navigation entre les différents menus permet une meilleure appropriation de l'outil avec notamment dans l'onglet Aide, deux liens vers le site de R et du CRAN qui détiennent toutes les informations sur les fonctionnalités de R et les nombreux packages utilisables.

Partie I : Quelques fonctions de base

 # Introduction

Création de variable

En écrivant :

```
msi=15
```

On crée la variable « msi » à laquelle on donne la valeur 15. Pour voir la valeur d'une variable, il suffit d'écrire « msi » :

```
msi
[1] 15
```

Liste des variables et Aide

Afin de visualiser l'ensemble des variables déjà crées et en mémoire, il faut utiliser la commande ls() :

```
> ls()
[1] "msi"
```

Enfin, à tout moment il est possible d'accéder à l'aide sur une fonction, afin de voir les différents arguments possible et son fonctionnement par la commande help(nom_fonction).

 # Question 1 : quelques éléments sur les vecteurs et matrices

La création des vecteurs se fait par la commande c() :

```
vect1<-c(2,-5,4.1,8,-2.6,3*2,4,8,5-3,1)
vect2=c(3,5,8,23,1,0,8,1,4,6)
```

Deux vecteurs ont été générés vect1 et vect2. A noter que = ou <- sont équivalents.

```
> vect1
[1] 2.0 -5.0 4.1 8.0 -2.6 6.0 4.0 8.0 2.0 1.0
```

Multiplication de vecteur :

```
> vect3<-10*vect2
> vect3
[1] 30 50 80 230 10 0 80 10 40 60
```

Information sur un vecteur :

```
> str(vect3)
num [1 :10] 30 50 80 230 10 0 80 10 40 60
```

Le vecteur est de taille 1 ligne et 10 colonnes et contient des nombres (numeric) dont les valeurs sont données.
Longueur d'un vecteur :

```
> length(vect3)
[1] 10
```

Fonctions rep et seq : la fonction rep(a,n) réplique n fois une valeur v et la commande seq(a,b,p) va créer une séquence de valeur allant de a à b avec un pas p.

```
f=c(rep(12,4))
f2=c(seq(1,10,0.03)
```

Statistiques sur un vecteur : moyenne, écart-type, variance, racine carrée de la variance,…

```
> mean(vect3)
[1] 59
> sd(vect3)
[1] 66.40783
var(vect3)
sqrt(var(vect3))  # = sd(vect3)
```

Reverse, order, Tri etc.

```
rev(vect3)
vect3
vect4<-rev(vect3)
vect4
[1] 60 40 10 80 0 10 230 80 50 30
order(vect3)
[1] 6 5 8 1 9 2 10 3 7 4
vect5
[1] 6 5 8 1 9 2 10 3 7 4
vect5<-order(vect3)
vect3[vect5]
sort(vect3)
[1] 0 10 10 30 40 50 60 80 80 230
sort(vect3,decreasing=TRUE)
[1] 230 80 80 60 50 40 30 10 10 0
sort(vect3,decreasing=FALSE)
[1] 0 10 10 30 40 50 60 80 80 230
> ls()
[1] "f" "f2" "msi" "vect1" "vect2" "vect3" "vect4" "vect5"
```

#Question 2 : les matrices

Création de matrice

```
m1<- matrix(data=c(5,3,8,-6), nrow=2, ncol = 2)
m1
[,1] [,2]
[1,] 5 8
[2,] 3 -6
m2<- matrix(data=c(0,13,-1,2), nrow=2, ncol = 2)
[,1] [,2]
[1,] 0 -1
[2,] 13 2
```

La création se fait colonne par colonne à partir du vecteur entré en argument. Si l'on veut un remplissage ligne par ligne, il faut le spécifier en argument.

```
> matrix(data=c(0,13,-1,2), nrow=2, ncol = 2, byrow=TRUE)
[,1] [,2]
[1,] 0 13
[2,] -1 2
```

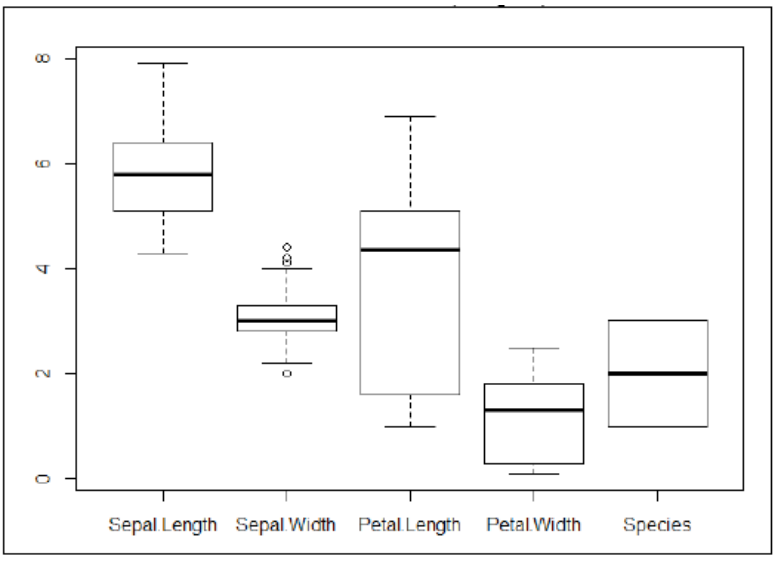
Concaténation de matrice :

```
f1<-rbind(m1,m2)
f2<-cbind(m1,m2)
```
Multiplication de matrices:
```
p1<-m1%*%m2
p2<-m2%*%m1
p1
[,1] [,2]
[1,] 104 11
[2,] -78 -15
p2
[,1] [,2]
[1,] -3 6
[2,] 71 92
```
Ou termes à termes
```
> m1*m2
[,1] [,2]
[1,] 0 -8
[2,] 39 -12
```

#Questions 3 et 4 : les iris de Fisher

Le jeu de données est chargé en mémoire par
```
Data(iris)
```
On peut le vérifier en tapant
```
str(iris)
```
On voit que l'on a un objet de type « data.frame », on a 169 observations (ou individus) de 5 variables : longueur et largeur des pétales, longueur et largeur des sépales, espèce (trois type d'espèce au total).
On peut résumer le jeu de données avec la commande
```
summary(iris)
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100 setosa :50
1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300 versicolor:50
Median :5.800 Median :3.000 Median :4.350 Median :1.300 virginica :50
Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199
3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800
Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500
```

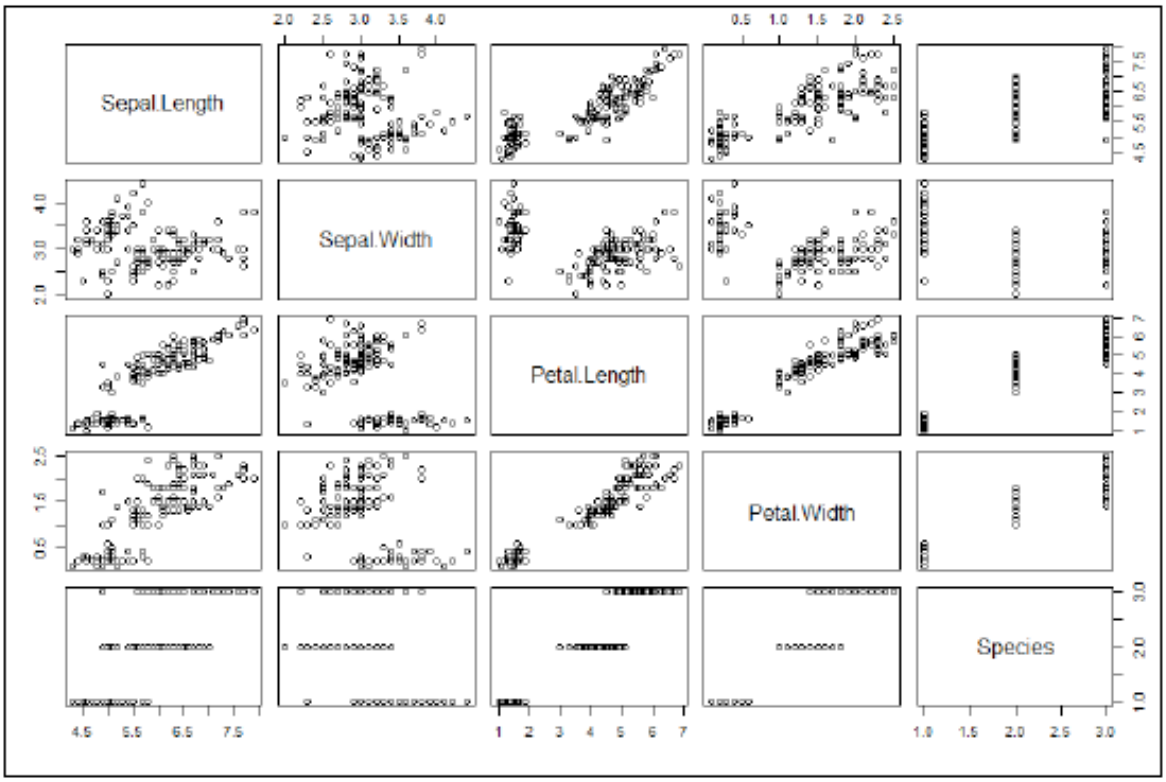
Qui peut s'illustrer à l'aide d'une boîte à moustache (boxplot) :



Avec le type dataframe, on accède à une colonne avec le symbole « $ ».

Par exemple on affiche le vecteur ‘longueur des pétales ‘ par la commande iris$Petal.Length.
Si vous ne souhaitez plus utiliser le dollar et accéder directement aux colonnes, vous pouvez taper :
attach(iris)

On peut tracer tous les types de graphiques possibles 2 à 2 en tapant :



plot(iris)

Mais aussi faire des choses plus raffinées en traçant (par exemple) les longueurs et largeurs des pétales tout en affichant chaque point dans une couleur qui correspond à son espèce (3 couleurs au total) :
plot(iris$Petal.Length,iris$Petal.Width, main= 'titre du graphique', col=as.factor(iris$Species), type='p', pch=15)
ou
plot(iris$Petal.Length,iris$Petal.Width, main= 'titre du graphique', col=as.factor(iris$Species), xlab='Longeur des pétales', ylab='Largeur des pétales')
/ !\ Il faut bien vérifier que le type de la variable indiquant les couleurs est de type factor

Et si l'on souhaite rajouter des nuages de points en plus sur un graphique :
plot(iris$Sepal.Length,iris$Sepal.Width,main= 'titre du graphique',col=iris$Species)
points(iris$Sepal.Length,iris$Petal.Width,main= 'titre du graphique',col=iris$Species,pch=17)

Partie II : quelques éléments de réponse

 # Environnement de travail

```
setwd('D:/ENTPE/TP_maison')
getwd()
[1] "D:/ENTPE/TP_maison"
```

 # Import des données

```
donnees = read.csv("MSI_TPhome.csv", header=TRUE, sep=";")
```

Le fichier possède une en-tête en 1ere ligne : header et le séparateur : sep des colonnes est la virgule

#Analyse à l'ouverture des données

```
dim(donnees)

summary(donnees)

str(donnees)

View(donnees)
```

#Renommer les colonnes

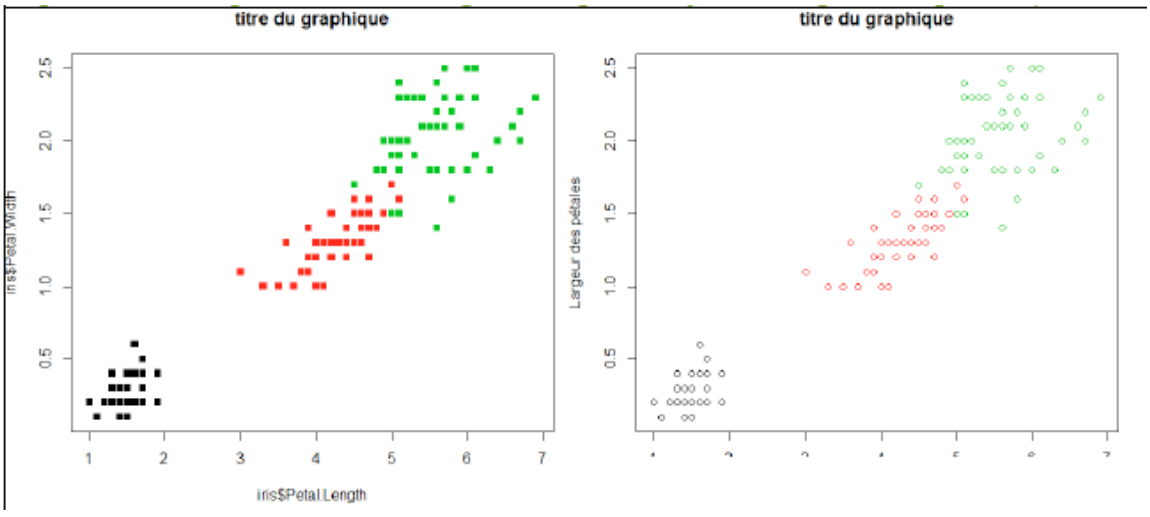
Pour afficher les noms de colonnes `names(donnees)` et les noms de lignes `row.names(donnees)`

`names(données) = c(…)` # bien penser au type des éléments contenus dans le vecteur c(…)

Si on veut changer le type d'une variable en chaîne de caractères (idem pour autres types) :
`donnees$pays=`**`as.character`**`(donnees$pays)`

 # Diagrammes en bâtons et histogramme

`barplot(…)`



hist(…)

 # Boîtes à moustache

boxplot(…)

 # La catégorie richesse et la fonction which (question 4)

Préalable :

On peut opérer des **requêtes conditionnelles / filtres** (vecteur Cond de type booléen TRUE/FALSE) pour ne conserver que certaines lignes d'une matrice : m[Cond,] ou seulement certaines colonnes : m[,Cond].

Cond = m[, j] > 0

m[Cond,] # conserve toute les lignes de m, dont l'élément de la colonne j est positif

La commande which permet de faire des requêtes sur des éléments particuliers d'une matrice par exemple : which(Cond) fournit l'index de positionnement des TRUE dans le vecteur booléen Cond.

La fonction rep(vect, each =) ou rep(vect, times =) permet de répéter des valeurs d'un vecteur

Réponse :
Pauvrete = rep(2,nrow(donnees)) ou : Pauvrete=rep(« moyen», nrow(donnees))
Donnees = cbind(donnees, Pauvrete)
Modifier cette dernière variable de donnees pour que les lignes correspondant aux pays riches portent la valeur 3 ou « riche » et les pays pauvres la valeur 1 ou « pauvre ».
Table(…)
Pie(…)

#Question 5
Calcul direct ou usage de **scale()**

#Question 6
plot(...)
cor(...)

#Question 7 :
Utiliser des requêtes conditionnelles