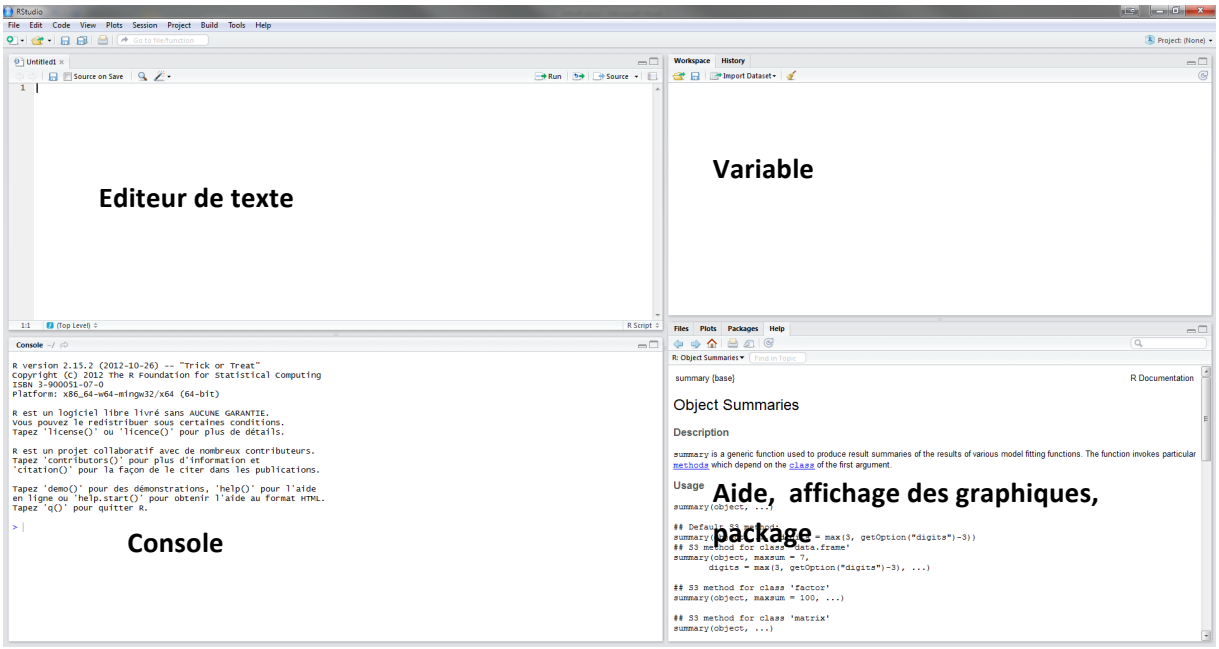
Introduction au logiciel statistique R

R est un logiciel de programmation libre dédié aux statistiques. Vous pouvez le télécharger gratuitement à cette adresse http://cran.univ-lyon1.fr/

Plusieurs interfaces graphiques existent pour R. Nous utiliserons RStudio lui aussi libre (http://www.rstudio.com/ide/download/desktop)



Variables « simples »

R est un logiciel basé sur l'utilisation de commandes qu'il faut taper à l'écran. Ces commandes permettent de faire des opérations sur des variables. Une variable permet de stocker une information comme un nombre ou une série de caractères.

La commande :

```
x <- 5
```

permet de créer une variable appelée x et qui contient la valeur 5.

La commande :

```
x
```

permet d'afficher la valeur contenue dans la variable x (ici 5).

La commande :

```
x <- "blabla"
```

permet de créer un variable contenant la chaine de caractère « blabla » dans la variable x. Dans cet exemple, la variable x ayant déjà été créée la valeur 5 qu'elle contient sera simplement remplacée la valeur « blabla ».

Attention : R est sensible à la casse pour lui, les lettres majuscules ne sont pas identiques aux lettres minuscules. Ainsi la commande x n'est pas équivalente à la commande X.

Exercice :
Créez une variable nommée y qui contient votre prénom. Faite afficher votre prénom à l'écran. Remplacez votre prénom dans la variable y par votre âge. Faite afficher votre âge à l'écran. Tapez la commande Y (en majuscule) et lisez le message.

R permet de faire des opérations.
La commande :

```
2+3
```

effectue l'opération et affiche son résultat.
`2^3` (2 puissance 3), `sqrt(3)` (racine carrée de 3), `log(3)` (logarithme de 3), `exp (3)` (exponentielle de 3) etc…

Les opérations peuvent faire intervenir des variables.
Les commandes

```
x <- 2
y <- 2*3
z <- y-x
```

permettent de créer x qi contient 2 puis y qui contient 2×3=6 puis z qui contient y-x soit 6-2=4.

Exercice :
Créez x contenant 2×3+4 puis y contenant 50 et enfin z contenant x au carré plus y. Affichez les valeurs de x, y et z.

Vecteurs ou colonnes de valeurs

Une même variable peut également contenir une série de valeurs.
La commande :

```
x <- c(8,12,6,5,30)
```

permet de créer une variable nommée x qui se présente sous la forme d'un vecteur contenant la colonne de chiffre 8, 12, 6, 5, 30. Les coordonnée du vecteur sont indexées, ici les indexe vont de 1 à 5.
La commande :

```
x[2]
```

permet d'afficher la deuxième coordonnée du vecteur x.
La commande :

```
x[2:4]
```

permet d'afficher les coordonnées de la deuxième à la quatrième.

Exercice :
Créer un variable z sous la forme d'un vecteur dont les coordonnées sont 3, 6, 2, 8, 5. Affichez les valeurs contenues dans z. Changez la valeur 6 (deuxième valeur) par la valeur 10. Faite de nouveau afficher z pour vérifier que cette valeur a été bien changée. Même question en créant une variable n contenant les caractères : « Georges », « Pierre » et « Patricia ». Puis remplacez « Pierre » par « Laurent ». Vérifiez à chaque fois le contenu de n en l'affichant.

R peut faire des opérations sur les valeurs contenu dans les vecteurs.
Les commandes :

```
y <- c(10,5,2)
y+2
```

permettent de créer y comme un vecteur contenant (10, 5, 2) puis de calculer et d'afficher la somme des valeurs contenues dans y et 2.

R peut calculer des grandeurs de statistique descriptive.
Les commandes :

```
mean(y)
var(y)
sd(y)
median(y)
min(y)
max(y)
```

permettent de calculer respectivement la moyenne, la variance, l'écart type, la médiane, le minimum et le maximum de la série de valeur contenant dans la variable y lorsque cette dernière a été créée comme un vecteur.

Exercice :
Créez la variable y comme un vecteur dont les coordonnées contiennent les nombres 10, 14, 8, 16. Affichez successivement le carré de y, la moyenne, la variance, l'écart type, la médiane, le minimum et le maximum de y.

Matrice ou tableau de valeurs

R peut traiter des variables sous forme de matrices.
La commande :

```
m <- matrix(2,ncol=3,nrow=4)
```

permet de créer la variable m comme une matrice de 3 colonnes et 4 lignes contenant la valeur 2.
Les commandes :

```
x <- 2
m <- matrix(x,ncol=3,nrow=4)
```

permettent de créer x contenant 2 puis m de 3 colonnes et 4 lignes contenant la valeur de x.
Les commandes :

```
x <- c(10,5,20)
m <- matrix(x,ncol =3,nrow=4)
```

permettent de créer x comme un vecteur de 3 coordonnées puis m une matrice formée de 4 vecteurs x mis côte à côte.
Les commandes :

```
m[2,3]
m[2,]
m[,3]
```

permettent d'afficher respectivement la valeur de la matrice x située sur sa deuxième ligne et sa troisième colonne, les valeurs situées sur sa deuxième ligne, les valeurs situées sur sa troisième colonne.

Exercice :
Créez la variable m sous la forme d'une matrice de 4 lignes et 3 colonnes contenant la valeur 4. Affichez toute les valeurs de m. Changez la valeur de m située sur la ligne 3 et la colonne 2 pour lui attribuer la valeur 5. Affichez m et vérifiez que la valeur a effectivement changé sur la ligne 3 et la colonne 2.
Calculez la moyenne de toutes les valeurs contenues dans m. Même question pour la variance et l'écart type.

Les commandes :

```
rm(x,y)
rm(list=ls())
```

permettent respectivement de supprimer des variables particulières (ici x et y), et d'effacer la totalité des variables.

Lecture d'un fichier

La commande :

```
resEnq <- read.csv("D:/Simon/enqueteAgriVire.csv", sep=";", dec=",")
```

permet de lire le fichier enqueteAgriVire.csv qui est situé dans le répertoire D:/Simon et de charger son contenu dans la variable resEnq. Cette commande spécifie également que le fichier contient un tableau valeurs séparées par des points virgules (;). Elle indique également que les décimales sont écrites à l'aide de point (.) et non d'une virgule.

Exercice :

Lisez le fichier enqueteAgriVire.csv et chargez son contenu dans une variable nommée resEnq. Affichez resEnq puis resEnq de la première ligne et première colonne. Que contient la variable resEnq ?

La commande :

```
save.image("C:/Documents and Settings/rougier.simon /Bureau/n/resultatEnquete.RData")
```

permet de sauvegarder l'ensemble des variables du projet R dans le répertoire D:/Simon dans un fichier RData

La commande :

```
str(resEnq)
```

permet de voir la structure d'un objet ouvert dans R

La commande :

```
summary(resEnq)
```

permet de récapituler les informations d'un objet. Si il s'agit d'une matrice elle permet d'obtenir des statistiques sur les colonnes.

Opérations sur des matrices contenant des noms

La commande :

```
mean(resEnq$Aire_ha)
```

permet d'obtenir la moyenne de la colonne Aire_ha. Comme cette colonne est la troisième de notre matrice *la commande :*

```
mean(resEnq[,3])
```

permet d'obtenir le même résultat

La commande :

```
mean(resEnq$Aire_ha[resEnq$Usage=="Culture"])
```

permet d'obtenir la moyenne de l'aire des parcelles (Aire_ha) où l'usage de ces parcelles est culture

Exercice :

Calculer la moyenne du maximum de pente (MaxSlope) pour l'ensemble des parcelles.
Calculer la moyenne et l'écart type de l'aire des parcelles des prairies temporaires.

Statistiques inférentielles

Les moyennes

La commande :

```
t.test(resEnq$Aire_ha, conf.level=0.95)
```

permet de réaliser un test de Student. Pour rappel celui-ci permet de comparer les moyennes d'échantillons pour savoir si leur différence est significative. Ici nous avons qu'un seul échantillon donc cette commande nous permet de calculer des intervalles de confiance ici avec un risque de 5%. Les valeur de cet intervalle sont écrit après la ligne " 95 percent confidence interval: "

Exercice :

Calculer l'intervalle de confiance du maximum de pente pour l'ensemble des parcelles avec un risque de 1%
Calculer l'intervalle de confiance de l'aire des parcelles de culture avec un risque de 5%
Calculer l'intervalle de confiance de l'aire des parcelles de prairies temporaires avec un risque de 1%

La commande :

```
t.test(resEnq$Aire_ha[resEnq$Usage=="Prairie
naturelle"],resEnq$Aire_ha[resEnq$Usage=="Prairie temporaire"])
```

permet de réaliser un test de Student pour comparer l'aire des parcelles de prairies naturelles à celles des parcelles de prairies temporaires. Plusieurs valeurs sont à expliquer :
"t" correspond au résultat du test que l'on reportait dans la table pour vérifier si on accepte ou rejette l'hypothèse nulle (H0)
"df" correspond au nombre de degré de liberté
"p-value" est la probabilité que H0 soit vraie. C'est à dire que pour que l'on rejette H0 il faut que notre seuil de risque soit supérieur à la valeur p
"alternative hypothesis" reprend l'hypothèse alternative H1

Exercice :
Réaliser un test de Student entre les parcelles de cultures et celles de prairies naturelles. La différence est elle significative avec un risque de 5% ? De 10 % ?
Réaliser un test de Student entre les parcelles de cultures et celles de prairies temporaires. La différence est elle significative avec un risque de 5% ? De 10 % ?

Les proportions

Les commandes :

```
TotParcelles <- length(resEnq$Usage)
nbCulture <- length(resEnq$Usage[resEnq$Usage=="Culture"])
```

permettent de calculer le nombre d'observations est de les stocker dans une variables. La première compte l'ensemble des observations, la deuxième seulement celles où l'usage est culture

La commande :

```
prop.test(nbCulture, TotParcelles)
```

permet de calculer l'intervalle de confiance sur des fréquences, ici sur la proportion des parcelles de culture

Exercice :
Calculer l'intervalle de confiance de la proportion des parcelles de prairies naturelles et temporaires avec un intervalle de confiance de 5 et de 1%

Khi2

La commande :

```
tabContin <- table(resEnq$Commune,resEnq$Usage)
```

permet réaliser un tableau de contingence entre les colonnes usage et commune et de le stocker dans la variable tabContin

La commande :

```
prop.table(tabContin)
```

permet de calculer le tableau des fréquences relatives

La commande :

```
prop.table(tabContin, margin=1)
```

permet de calculer le tableau des fréquences conditionnelles relatives calculé en lignes

La commande :

```
prop.table(tabContin, margin=2)
```

permet de calculer le tableau des fréquences conditionnelles relatives calculé en colonnes

La commande :

chisq.test(tabContin)

permet de réaliser un test du Khi2 permettant de vérifier l'indépendance des variables

La commande :

chi2 <- chisq.test(tabContin)

permet de réaliser un test du Khi2 et de le stocker dans une variable chi2 qui permet d'obtenir plusieurs résultats qui peuvent être utiles

La commande :

chi2$statistic

permet d'obtenir la valeur de la statistique de ce test

La commande :

chi2$parameter

permet d'obtenir le nombre de degré de liberté

La commande :

chi2$p.value

permet d'obtenir la valeur de p

La commande :

chi2$observed

permet d'obtenir le tableau de contingence observé

La commande :

chi2$expected

permet d'obtenir le tableau de contingence théorique

La commande :

chi2$residuals

permet d'obtenir le tableau de contingence des résidus (les écart entre le tableau observé et celui théorique)

Régression linéaire

La commande :

plot (resEnq$Aire_ha,resEnq$MeanSlope)

permet de réaliser un graphique sous forme de nuage de point entre les variables Aire_ha et MeanSlope

La commande :

lin <- lm(resEnq$Aire_ha~resEnq$MeanSlope)

permet de réaliser une régression linéaire et de stocker le résultat dans une variable lin pour obtenir plus de résultats

La commande :

lin$coefficients

le vecteur des coefficients de la droite

La commande :

lin$fitted.values

les valeurs calculées par la régression d'après les variables explicatives

La commande :

lin$residuals

le vecteur des résiduels (c'est à dire, les valeurs effectives moins les valeurs calculées)

Exercice :

Réaliser un nuage de point et calculer les coefficients de la droite de régression entre les variables Aire_ha et MaxSlope puis entre Max_Slope et MeanSlope

Covariance et corrélation

La commande :

cov (resEnq$Aire_ha,resEnq$MeanSlope)

permet de calculer la covariance entre Aire_ha et MeanSlope

La commande :

cor (resEnq$Aire_ha,resEnq$MeanSlope)

permet de calculer le coefficient de corrélation entre Aire_ha et MeanSlope

Exercice :
Calculer la covariance et le coefficient de corrélation entre les variables Aire_ha et MaxSlope puis entre Max_Slope et MeanSlope

La commande :

```
cor (resEnq[,3:5])
```

permet de calculer le coefficient de corrélation entre toutes les paires de variables

La commande :

```
cor.test (resEnq$Aire_ha,resEnq$MeanSlope)
```

permet de réaliser un test de correlation entre Aire_ha et MeanSlope

Exercice :
Réaliser un test de corrélation entre les variables Aire_ha et MaxSlope. Ces variables sont elles corrélées ?
Entre Max_Slope et MeanSlope