

# Formulaire R pour les analyses uni et bivariées

## I. Charger des données dans R

Pour charger un dataframe dans une variable nommée ici `data` en utilisant la console R (les paramètres ci-dessous sont adaptés au format du fichier d'exemple, pas nécessairement transposables) :

```
data <- read.table(file=mon_url, header=TRUE, stringsAsFactors = FALSE, dec=".") ;  
# stringsAsFactors=FALSE permet de conserver les chaînes de caractères telles quelles  
# header=TRUE explique que la première ligne contient le nom des colonnes  
# dec="." Précise que le séparateur décimal est le point (et non la virgule)
```

Le nom du fichier (ici symbolisé par `mon_url`) sera renseigné entre guillemets et avec des slashes et non des antislashes. N'oubliez pas l'extension du fichier, même si elle est masquée par Windows « .txt ». Remplacez naturellement l'emplacement du fichier par celui qui convient, sans oublier les guillemets, et en mettant des slashes à la place des antislashes. Exemple :

```
data <- read.table(file="C:/travaux/ENSEIGNEMENTS/2009-11-30/data.txt", header=TRUE) ;
```

## II. Description univariée

Fonctions génériques

```
print( ma_var ) ; # valeurs natives  
summary( ma_var ) ; # description d'une variable, quali ou quanti  
print(table( ma_var )) ; # tableau de contingence
```

Graphiques pour une variable qualitative :

```
pie(table( ma_var_quali )) ; # camembert  
plot(table( ma_var_quali )) ; # diagramme en barres (échelle X qualitative)  
mosaicplot(table( ma_var_quali )) ; # mosaïque
```

Graphiques pour une variable quantitative discrète :

```
plot(table( ma_var_quanti )) ; # barres (échelle X quantitative)  
plot(ecdf(ma_var_quanti ), do.points=TRUE, verticals=TRUE) ; # effectifs cumulés
```

Graphiques pour une variable quantitative continue :

```
hist(ma_var_quanti, freq=FALSE, col="pink") ; lines(density(ma_var_quanti)) ;  
# histogramme et estimation de la densité de probabilité  
boxplot(ma_var_quanti, col="pink" ) ; # boxplot = boîte à moustaches  
plot(ecdf(ma_var_quanti ), do.points=TRUE, verticals=TRUE) ; # estimation fonction répartition
```

Calcul automatique avec R (sur données natives) :

```
mean(ma_var_quanti) # moyenne  
sd(ma_var_quanti) # estimateur non biaisé de l'écart type  
var(ma_var_quanti) # estimateur non biaisé de la variance  
median(ma_var_quanti) # médiane
```

Pas de formule clefs en main pour le mode... qui en fait est très peu utilisé

Pour la suite, le tilde (~) signifie « en fonction de ».

Pour écrire le tilde : [AltGr]+[2] puis [espace]

### III. Croiser deux variables qualitatives

Description graphique

```
mosaicplot( table( mavarquali1, mavarquali2 ) ); # graphique en mosaïque
```

Description en console

```
table(mavarquali1, mavarquali2); # affiche le tableau de contingence de deux variables qualitatives
```

Test bivarié

```
chisq.test(mavarquali1, mavarquali2); # réalise automatiquement le test du  $\chi^2$ . La correction de Yates  
# est réalisée dans tous les cas (elle a un effet négligeable)
```

```
fisher.test(mavarquali1, mavarquali2); # réalise automatiquement le test exact de Fisher
```

### IV. Croiser une variable qualitative et une variable quantitative (comparer des moyennes)

Description graphique

```
boxplot( formula=mavarquanti ~ mavarquali ); # boxplot
```

Test bivarié

```
t.test( formula=mavarquanti ~ mavarquali ); # T de Student. Pour données non appariées.  
# Dans son paramétrage par défaut,  
# ce test propose une approximation de Welch
```

```
wilcox.test( formula=mavarquanti ~ mavarquali ); # T de Wilcoxon pour données non appariées.  
summary(aov(formula=mavarquanti ~ mavarquali)); # Analyse de la variance à un facteur
```

### V. Croiser deux variables quantitatives

Description graphique

```
plot( x=mavarquanti1, y=mavarquanti2 ); # nuage de points, scatterplot  
library(hexbin); # charger tout d'abord la bibliothèque Hexbin  
plot(hexbin(x=mavarquanti1, y=mavarquanti2)); # hexbin : nuage de points amélioré
```

Description en console

```
cov(mavarquanti1, mavarquanti2); # covariance  
cor(mavarquanti1, mavarquanti2); # coefficient de corrélation de Pearson  
cor(mavarquanti1, mavarquanti2, method="spearman"); # coef. de cor. sur les rangs de Spearman
```

Test bivarié

```
cor.test(mavarquanti1, mavarquanti2); # test de nullité du coefficient de corrélation de Pearson  
cor.test(mavarquanti1, mavarquanti2, method="spearman"); # test sur les rangs de Spearman  
summary(glm(formula=mavarquanti1 ~ mavarquanti2)); # régression linéaire simple  
# pour la régression, multiple, ajouter des var explicatives avec +  
# pour une régression logistique, ajouter family="binomial"
```