

Travail Pratique 2

Les graphiques avec R

STT1039
BIOLOGIE QUANTITATIVE

Réalisé par
Marc Pépino



Département des sciences de l'environnement

1. Objectifs du TP2

Le travail pratique 2 (TP 2) a pour objectif général de familiariser les étudiants aux fonctions graphiques de R. À la fin du TP 2, l'étudiant doit savoir : ouvrir et organiser les fenêtres graphiques, représenter graphiquement la distribution des valeurs d'une variable et la relation entre deux variables, maîtriser les principaux paramètres graphiques lui permettant de réaliser un graphique « sur – mesure », sauvegarder ses graphiques à l'extérieur du logiciel R.

Note : Tout ce qui est mis en rouge et en gras dans les exemples (encadrés) sont des noms donnés par l'utilisateur. Ces noms sont à modifier en conséquence si les exemples sont utilisés avec d'autres bases de données.

Les graphiques présentés dans les exemples seront pour la plupart réalisés à partir de la base de données *TP2_Exemples.csv* disponible sur le portail du cours :

```
># Importation de la base de données
> tab <- read.csv("D:/bureau/TP2_Exemples.csv",sep=";",dec=".",h=T)
>
> #Visualisation de la structure du tableau de données
> str(tab)
'data.frame': 100 obs. of 3 variables:
 $ x: num 183 202 190 198 193 ...
 $ y: num 611 574 612 673 706 ...
 $ f: Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
```

OPTIONNEL :

```
#Une base de données similaire peut être obtenue à partir du code suivant:
#Construction des variables
x <- c(rnorm(60,200,10),rnorm(40,210,7))
f <- c(rep("A",60),rep("B",40))
y <- rep(NA,100)
y[1:60] <- 55 + 3*x[1:60] + rnorm(60,0,50)
y[61:100] <- 60 + 3*x[61:100] + rnorm(40,0,60)

#Construction du tableau de données
tab = as.data.frame(x)
tab$y=y
tab$f=as.factor(f)
```

Pour plus de détails sur les graphiques avec R, vous pouvez consulter **Paradis (2005)** dont le lien est donné dans la bibliographie.

2. Organisation des fenêtres graphiques

Pour ouvrir une fenêtre graphique, vous pouvez utiliser la fonction **x11()** (utiliser *quartz()* sur MacOS et *X11()* sur Linux). A chaque fois que vous taperez *x11()* dans la console, une nouvelle fenêtre graphique s'ouvrira. Seule la dernière fenêtre ouverte sera active (les graphiques se feront dans cette fenêtre) :

```
#Exemple 1
#Ouvrir une fenêtre graphique
x11()
#Faire le graphique de y en fonction de x
plot(tab$x,tab$y)
```

Si vous souhaitez diviser votre fenêtre graphique active pour faire plusieurs graphiques dans la même fenêtre, vous pouvez utiliser la fonction **layout()**. Le principal argument à placer dans cette fonction est une matrice à valeurs entières qui indique le nombre et l'emplacement des graphiques qui seront réalisés dans la fenêtre active.

```
#Exemple 2
#Diviser la fenêtre graphique en deux sur la hauteur
layout(c(1:2))
#Faire un premier graphique
plot(tab$x,tab$y)
#Faire un second graphique
plot(tab$y,tab$x)
```

Pour visualiser l'emplacement des graphiques, vous pouvez utiliser la fonction **layout.show()** en mettant le nombre de figures à visualiser en argument de cette fonction (voir d'autres exemples dans l'aide en ligne).

```
#Exemple 3
#Diviser la fenêtre graphique en six
M <- matrix(1:6,3,2)
layout(M)
#Visualiser la disposition des six graphiques
layout.show(6)
```

Il est également possible de diviser les fenêtres graphiques en utilisant la fonction **par()**. Cette fonction contrôle de nombreux paramètres graphiques dont une description sommaire est donnée à la section 3.3 *Les paramètres graphiques*.

3. Fonctions et paramètres graphiques

3.1. Les fonctions graphiques principales

Le tableau suivant reprend les principales fonctions qui vous permettront de visualiser les relations entre une ou plusieurs variables ainsi que la distribution des données.

Tableau 1. Fonctions graphiques principales (Extrait de Paradis 2005)

Fonction	Définition
plot(x,y) ou plot(y~x)	Diagramme de dispersion entre x (abscisse) et y (ordonnée)
pairs(M)	Dessiner toutes les relations entre les colonnes de M
hist(x)	Histogramme
boxplot(x) ou boxplot(y~f)	Diagramme à moustache
barplot(table(f))	Diagramme à bâtons
pie(table(f))	Camembert ou diagramme en pointe de tarte

Note : x et y sont des variables numériques, f est une variable catégorique, et M est une matrice ou un tableau de données.

```
#Exemple 4
#Exemple d'utilisation des fonctions du tableau 1
x11()
plot(tab$x,tab$y)
x11()
pairs(tab)
x11()
hist(tab$x)
x11()
boxplot(tab$x~tab$f)
x11()
pie(table(tab$f))
```

Plusieurs arguments peuvent être ajoutés à l'intérieur des fonctions graphiques principales. Il est possible de changer les valeurs par défaut de ces arguments pour modifier l'allure générale du graphique. Attention, certaines fonctions graphiques n'acceptent pas ces arguments ; consulter alors l'aide en ligne en tapant « ? » suivi du nom de la fonction).

Liste sommaire des arguments des fonctions principales :

add =	indique s'il faut superposer le nouveau graphique au graphique existant (TRUE ou FALSE)
axes =	indique s'il faut tracer les axes et le cadre autour du graphique (TRUE ou FALSE)
xaxt = 'n'	l'axe des x n'est pas tracé
yaxt = 'n'	l'axe des y n'est pas tracé
type=	indique le type de graphe, "p" : point, "l" : ligne, "o" : points connectés par des lignes, 'n' : rien n'est représenté

xlim=, ylim= indique les limites de l'axe des x et des y, par exemple : xlim=c(1,10)
xlab=, ylab= indique le nom des axes, par exemple ylab="axe des ordonnées"
main= indique le titre du graphique, par exemple main="Titre principal"
sub= indique le sous-titre du graphique, par exemple sub="sous-titre"

```
#Exemple 5
#Exemple d'utilisation de quelques arguments
x11()
plot(tab$x,tab$y,xlim=c(0,500),ylim=c(0,1000),xlab="x",ylab="y")
```

Il est pratique d'utiliser ces arguments en combinaison avec les fonctions graphiques secondaires. Nous en verrons quelques exemples dans la section suivante.

3.2. Les fonctions graphiques secondaires

Les fonctions graphiques secondaires sont utilisées pour modifier ou compléter un graphique déjà existant. Le tableau suivant résume les fonctions graphiques secondaires les plus courantes.

Tableau 2. Fonctions graphiques secondaires (Extrait de Paradis 2005)

Fonction	Définition
points(x,y)	Ajouter des points de coordonnées (x,y)
abline()	Tracer une ligne
segments(x1, y1, x2, y2)	Tracer des lignes entre les points (x1,y1) et (x2,y2)
text(x, y, labels=)	Ajouter du texte aux coordonnées (x,y)
mtext("texte",side=,line=)	Ajouter un texte dans la marge en bas (<i>side</i> =1), à gauche (2), en haut (3) ou à droite (4) ; « line » spécifie la distance par rapport au cadre du graphique
legend(x, y, legend)	Ajouter la légende aux coordonnées (x,y). « legend » indique ce qu'il faut mettre dans la légende (voir exemple 8)
title("Titre")	Ajouter un titre au graphique
axis(side=,vec=)	Ajouter l'axe en bas (<i>side</i> =1), à gauche (2), en haut (3) ou à droite (4). « vec » spécifie les graduations à tracer (optionnel)
polygon(x,y)	Tracer un polygone reliant les points de coordonnées (x,y)

```

#Exemple 6
#Faire le graphique sans point et sans axe en ajoutant le nom des axes
x11()
plot(tab$x,tab$y,type='n',xlab="x",ylab="y",axes=F)
#Ajouter les axes
axis(1)
axis(2)
#Ajouter le titre
title("Relation entre x et y")
#Ajouter les points
points(tab$x,tab$y,type="p")
#Ajouter du texte à gauche des points (pos=2)
text(tab$x,tab$y,tab$f,pos=2)

```

3.3. Les paramètres graphiques

En plus des fonctions graphiques principales et secondaires, la qualité des représentations graphiques peut être améliorée grâce à l'utilisation des paramètres graphiques. Ces derniers ont tous des valeurs par défaut que vous pouvez consulter en tapant « ?par ». Vous pouvez modifier de façon permanente ces paramètres dans la fonction **par()** ou utiliser ces paramètres comme argument dans les fonctions graphiques principales ou secondaires (attention, pas tous les paramètres peuvent être utilisés dans les fonctions graphiques).

Tableau 3. Paramètres graphiques (Extrait de Paradis 2005)

Paramètre	Définition
bty	Forme du cadre autour du graphique ("o","l","7","u"); bty='n' supprime le cadre
cex	Taille des caractères (défaut : <i>cex</i> = 1). Il peut être utilisé pour modifier la taille des symboles, des axes (<i>cex.axis</i>), des titres des axes (<i>cex.lab</i>), du titre (<i>cex.main</i>) ou du sous-titre (<i>cex.sub</i>).
col	Couleur des symboles. Il y a aussi <i>col.axis</i> , <i>col.lab</i> , <i>col.main</i> , <i>col.sub</i> . Il faut indiquer le nom de la couleur en anglais entre guillemet (exemple : <i>col</i> ="red")
font	Style du texte (1 : normal ; 2 : gras ; 3 : italique ; 4 : gras et italique)
las	Orientation des annotations des axes (0 : parallèles aux axes, 1 : horizontales, 2 : perpendiculaires aux axes, 3 : verticales)
lty	Style des lignes (1 : continue, 2 : tirets, 3 : points, 4 : points et tirets alternés, 5 : tirets longs, 6 : tirets courts et longs alternés)
lwd	Largeur des lignes (défaut : <i>lty</i> = 1)
mar	Espace entre les axes et le bord de la figure. Il s'écrit de la forme <i>mar</i> = c(bas, gauche, haut, droite) ; défaut : <i>mar</i> = c(5.1, 4.1, 4.1, 2.1). Note : il peut être utile de l'utiliser en combinaison avec le paramètre <i>oma</i> qui contrôle les l'espace des marges extérieures.
mfrow	Diviser la fenêtre graphique. Utiliser <i>mfrow</i> = c(nr, nc) pour diviser la fenêtre graphique en nr lignes et nc colonnes. Exemple : <i>mfrow</i> = c(2,1) permet de faire deux graphiques l'un en dessous de l'autre.
pch	Type des symboles (entiers de 1 à 25 ou n'importe quel caractère entre guillemets).

```
#Exemple 7
#Exemple d'utilisation des paramètres mfrow et mar
#Partitionner la fenêtre et modifier les marges
x11()
par(mfrow=c(2,1),mar=c(4.1,4.1,1.1,2.1))
#Ajouter les données pour "A" sur le premier graphique puis pour "B" sur le deuxième
plot(tab$x,tab$y,type='n',xlab="",ylab="y",bty='n')
points(tab[tab$f=="A",]$x,tab[tab$f=="A",]$y)
mtext("Catégorie : A")
plot(tab$x,tab$y,type='n',xlab="x",ylab="y",bty='n')
points(tab[tab$f=="B",]$x,tab[tab$f=="B",]$y)
mtext("Catégorie : B")
```

```
#Exemple 8
#Exemple d'utilisation des paramètres graphiques combinés aux fonctions graphiques
x11()
#Mettre en page le graphique
plot(tab$x,tab$y,type='n',xlab="x",ylab="y",bty='n', yaxt='n')
axis(2,las=2)
#Ajouter les données pour "A" et "B" en changeant de symbole
points(tab[tab$f=="A",]$x,tab[tab$f=="A",]$y,pch=20,col="blue",cex=2)
points(tab[tab$f=="B",]$x,tab[tab$f=="B",]$y,pch=20,col="red",cex=2)
#Ajouter la légende
legend(locator(1),pch=20,pt.cex=2,col=c("blue","red"),c("A","B"))
```

Remarque : Pour le positionnement de la légende vous pouvez utiliser la fonction **locator()** au lieu de préciser les coordonnées x et y. Dans l'exemple 8, il faut cliquer avec souris à l'endroit désiré pour l'emplacement de la légende. Une autre option souvent pratique est d'utiliser l'une des quatre options suivantes : « *topleft* », « *topright* », « *bottomleft* », ou « *bottomright* ».

4. Faire un histogramme « sur-mesure »

Un histogramme se construit avec la fonction **hist()**. Les arguments contenus dans cette fonction permettent de modifier l'allure générale de l'histogramme. Combinés aux fonctions graphiques secondaires et aux paramètres graphiques, il est possible de construire un histogramme « sur-mesure ». Voici une liste des principaux arguments servant à modifier l'allure de l'histogramme (une liste complète se trouve dans l'aide en ligne « ?hist »).

Liste des principaux arguments de la fonction **hist()** :

x	vecteur de données sur lequel est fait l'histogramme
breaks	limites ou nombre de classes. Il y a de nombreuses options dont écrire un vecteur indiquant les limites de chacune des classes, écrire un nombre indiquant le nombre de classe, écrire des fonctions pour déterminer les classes (les fonctions déjà implémentées sont "Sturges", "Scot" et "FD")
freq	histogramme de fréquence (<i>freq</i> = TRUE) ou de probabilité (<i>freq</i> = FALSE)

density	densité des hachures dans les barres de l'histogramme (exemple : <i>density</i> = 20)
angle	inclinaison des hachures dans les barres de l'histogramme (exemple : <i>angle</i> = 45)
col	couleur de remplissage des barres (exemple : <i>col</i> = "grey")
border	couleur de la bordure des barres (si <i>border</i> = NA, aucune bordure n'est tracé)
plot	représentation de l'histogramme (<i>plot</i> = TRUE) ou résumé des statistiques descriptives de l'histogramme (<i>plot</i> = FALSE)

```

> #Exemple 9
> #Récupérer les statistiques descriptives de l'histogramme
> h<-hist(tab$x)
> h
$breaks
[1] 180 185 190 195 200 205 210 215 220 225 230

$counts
[1] 3 3 4 17 27 20 12 10 3 1

$sintensities
[1] 0.006 0.006 0.008 0.034 0.054 0.040 0.024 0.020 0.006 0.002

$density
[1] 0.006 0.006 0.008 0.034 0.054 0.040 0.024 0.020 0.006 0.002

$mids
[1] 182.5 187.5 192.5 197.5 202.5 207.5 212.5 217.5 222.5 227.5

$xname
[1] "tab$x"

$equidist
[1] TRUE

attr("class")
[1] "histogram"

```

Interpretation : « *breaks* » correspond aux bornes de classes, « *counts* » à l'effectif pour chacune des classes, « *density* » à la fréquence relative et « *mids* » au centre de classe.

Remarque : les fréquences cumulées s'obtiennent à l'aide de la fonction *cumsum()*.

```
#Exemple 10
#Faire deux histogrammes sur le même graphique
x11()
#Premier histogramme
ha=hist(tab[tab$=="A"],$x,col=rgb(1,0,0,alpha=0.8),border=NA,xlab="x",
ylab="Fréquence",main="",xlim=c(160,260),breaks=seq(160,260,5))
#Deuxième histogramme
hb=hist(tab[tab$=="B"],$x,col=rgb(1,0,0,alpha=0.8),border=NA,
breaks=seq(160,260,5),add=T)
#Légende
legend(locator(1),pch=22,pt.cex=3,pt.bg=c("red","blue"),c("A","B"),bty='n')
```

Remarque : S'assurer de prendre les mêmes intervalles de classe pour les deux histogrammes. L'utilisation de la fonction **rgb()** permet de choisir toutes les nuances de couleur et de mettre les couleurs en transparence grâce à l'argument *alpha*.

5. Sauvegarder les graphiques

Il existe de nombreuses façon de sauvegarder un graphique et cela en plusieurs format possible (exemple : pdf, eps, TIFF, Jpeg...). Il est conseillé de sauvegarder en format eps ou pdf pour conserver une bonne qualité graphique. Lorsque la fenêtre graphique est active, il est possible de sauvegarder le graphique à partir de la barre de menu :

Fichier -> Sauver sous

Alternativement, un **clic droit** avec la souris sur le graphique permet de copier ou sauvegarder le graphique. Lorsque le graphique est copié à partir de la fenêtre graphique pour l'insérer ensuite dans un document, le copier comme vectoriel donne une bonne résolution.

Enfin, il est possible aussi d'exporter le graphique à partir de la console à l'aide des fonctions **pdf()** pour les format pdf ou **postscript()** pour les format eps. Le code commence avec la fonction **pdf()** ou **postscript()**, continue avec le code pour réaliser le graphique et se termine en ajoutant **dev.off()** pour fermer le fichier. Attention : ne pas utiliser **x11()** entre **pdf()** et **dev.off()**.

```
#Exemple 11
#Exportation du graphique en pdf
pdf(file="D:/bureau/graphique.pdf",paper="letter",width=7,height=7)
#Réaliser le graphique
par(mfrow=c(2,1),mar=c(4.1,4.1,1.1,2.1))
hist(tab[tab$=="A"],$x,col="black",border="white",xlab="",ylab="Fréquence",
main="Catégorie A",breaks=seq(160,260,5))
hist(tab[tab$=="B"],$x,col="black",border="white",xlab="x",ylab="Fréquence",
main="Catégorie B",breaks=seq(160,260,5))
#fermeture du fichier
dev.off()
```

6. Conseils pratiques à retenir

Consulter le cours « Présentation des données et explorations graphiques », la **webographie** et les exercices pour plus d'exemples !

La bibliothèque *ggplot2* est très populaire pour visualiser rapidement des bases de données complexes (voir un exemple dans l'exercice 5).

Tableau 1 : fonctions principales (p. 4). Utiliser la fonction requise pour générer le type de graphique souhaité.

Liste des arguments des fonctions principales (p. 4-5). Utiliser ces arguments pour modifier l'allure générale du graphique (ex : limite et noms des axes).

Tableau 2 : fonctions secondaires (p. 5). Utiliser ces fonctions pour ajouter des couches d'informations à un graphique existant.

Tableau 3 : paramètres graphiques (p. 6). Ces paramètres graphiques sont utilisables comme argument dans les fonctions principales ou secondaires pour modifier le type de représentation (ex : taille et couleur des symboles, épaisseur des lignes, texte, etc.)

Astuce. Il est souvent pratique de créer un graphique (ex : fonction '*plot*') sans ajouter de symbole en utilisant l'argument « type = 'n' ». On obtient ainsi un graphique vierge sur lequel peuvent être ajoutées des couches d'information à l'aide des fonctions secondaires.

7. Exercices

Répondez aux exercices dans un fichier texte « Nom_Prenom_TP2.txt ». Au début du fichier, identifiez-vous par votre nom et prénom, ajouter la date et le numéro du TP, le tout en utilisant des commentaires. Utilisez des commentaires pour structurer vos réponses.

Importez la base de données *TP2_Exercices.csv* disponible sur votre portail étudiant. **Optionnel** : vous pouvez également utiliser le code mentionné à la fin de l'exercice 5 pour refaire les exercices 1 à 4 avec une nouvelle base de données.

Exercice 1. Représenter y en fonction de x . Renommer l'axe des x en « Longueur » et l'axe des y en « Poids ». Enlever le cadre autour du graphique. Choisir deux symboles distincts pour représenter les données correspondant aux catégories A et B de la variable f . Ajouter la légende. **Optionnel** : ajouter une courbe de lissage pour chacune des catégories A et B.

Exercice 2. Représenter y en fonction de f à l'aide d'une boîte à moustache. Appeler l'axe des x « Sexe » et l'axe des y « Poids ». **Optionnel** : ajouter entre parenthèse le nombre de données pour chacune des deux catégories au dessus de leur boîte respective (c'est-à-dire trouver combien la catégorie A contient de valeur, combien la catégorie B contient de valeur et écrire ces deux nombres au dessus de chacune des boîtes).

Exercice 3. Représenter l'histogramme de x . Appeler l'axe des x « Longueur » et l'axe des y « Fréquence ». Ne pas mettre de titre au graphique. **Optionnel** : au-dessus de chaque barre de l'histogramme, indiquer le nombre de données qu'elle contient.

Exercice 4. Représenter dans une même fenêtre graphique deux histogrammes, l'un en dessous de l'autre. Le premier représente x pour la catégorie A, le deuxième représente x pour la catégorie B. Appeler l'axe des x « Longueur » et l'axe des y « Fréquence ». S'assurer que les deux graphiques aient les mêmes limites de classe. Ajouter sur chacun des histogrammes une ligne verticale en tirets et à épaisseur double pour $x = 200$. Ajouter un titre pour identifier les catégories A et B. Indiquer le nombre de données contenues dans chacun des histogrammes en sous-titre et de la forme « $n = xx$ ». Exporter le graphique en format pdf sur le bureau.

Exercice 5. Importer la base de données « TP1_Exercices.xlsx » se trouvant sur votre portail étudiant (convertir d'abord le fichier en format .csv). Ces données sont aussi disponibles à l'adresse suivante : <http://datadryad.org/resource/doi:10.5061/dryad.p2k02>. À l'aide de la bibliothèque *ggplot2*, représenter les relations longueur – poids pour chacune des familles d'omble de fontaine à la fin de l'expérience. Ajouter une courbe de lissage sur chacun des graphiques générés.

Code pour créer une base de données similaire à celle du fichier *TP2_Exercices.csv* :

```
Construction des variables
x <- c(rnorm(60,200,10),rnorm(40,210,7))
f <- c(rep("A",60),rep("B",40))
y <- rep(NA,100)
y[1:60] <- 55 + 3*x[1:60] + rnorm(60,0,50)
y[61:100] <- 60 + 3*x[61:100] + rnorm(40,0,60)

#Construction du tableau de données
tab = as.data.frame(x)
tab$y=y
tab$f=as.factor(f)

#Visualisation de la structure du tableau de données
str(tab)
```

Bibliographie

- Crawley, M. J. 2013. The R book (2^{ème} éd.). John Wiley and Sons, Chichester, England.
[Disponible en ligne à la bibliothèque de l'UQTR \(voir la Webthèque sur le portail de cours\)](#)
- Legendre, P. 2009. Introduction au langage statistique R.
http://biol09.biol.umontreal.ca/BIO6077/Introduction_R.pdf.
- Millot, G. 2018. Comprendre et réaliser les tests statistiques à l'aide de R : manuel pour les débutants (4^{ème} éd.). De Boeck, Bruxelles, Belgique.
- Murrell, P. 2011. R graphics (2^{ème} éd.). Chapman & Hall/CRC The R series, Boca Raton, Floride.
- Paradis, E. 2005. R pour les débutants.
http://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf.
- Short, T. 2004. R Reference Card.
<http://cran.r-project.org/doc/contrib/Short-refcard.pdf>.

Liens utiles :

- Site officiel de R.
<http://www.r-project.org>
- Ateliers R du CSBQ.
<https://qcbs.ca/wiki/r>
- Quick R.
<http://www.statmethods.net/>
- Apprentissage en ligne du logiciel R.
<https://www.datacamp.com/>
- Beginners statistics introduction.
http://influentialpoints.com/Training/basic_statistics_introduction.htm
- R Graphics Cookbook.
<http://www.cookbook-r.com/Graphs/>
- Liste des bases de données disponibles sous R.
<https://vincentarelbundock.github.io/Rdatasets/datasets.html>