





# L2 Biostatistiques-Bioinformatique



## TP : exploration d'un jeu de données

### 1 Introduction à


L'objectif de ce TP est de vous faire utiliser le logiciel  pour procéder à l'exploration préliminaire d'un jeu de données. Le but sera essentiellement d'apprendre à utiliser les commandes graphiques pour visualiser des données – la formulation d'hypothèses et les tests intervenant souvent après coup dans la pratique.

Il vous est rappelé que, pour obtenir l'aide sur une fonction `machin`, il vous suffit, dans une console , de taper `help(machin)`. Ceci vous sera très utile au cours de ce TP, si vous voulez par exemple savoir quelles options passer à une fonction pour qu'elle fasse un "joli" graphique. Vous disposez également d'un glossaire  précisant les principales fonctions que vous pourrez utiliser au cours de ce TP. Finalement, votre cours  est toujours disponible sur Spiral.

Dans la suite du TP, les phrases en italique sont les questions auxquelles vous devez répondre.

*Commencez par lancer une console  depuis votre environnement de travail, en double-cliquant sur l'icône  sous Windows<sup>1</sup>.*

#### 1.1 Prise en main de

Vous allez commencer par quelques commandes simples qui vous aideront à vous familiariser au langage  :

1. Créez deux variables `x` et `y` et affectez leur les valeurs 2 et 3 respectivement.
2. Créez la variable `z` qui soit la somme de `x` et `y`.
3. Créez un vecteur `v` de type numérique et de longueur 10. Initialisez ce vecteur avec les valeurs de 1 à 10.
4. Affichez le 5<sup>e</sup> élément du vecteur `v` créé précédemment.

### 2 Lecture et filtrage des données

Les données que vous allez étudier concernent le poids à la naissance de bébés américains de sexe masculin. Pour expliquer les variations de cette variable, d'autres ont été enregistrées, concernant la mère de l'enfant : taille, poids, âge, etc... Votre but lors de ce TP est de parvenir à comprendre comment les différentes variables sont reliées entre elles, et ce à l'aide de représentations graphiques appropriées.

Les données se présentent sous la forme d'un `data.frame` à 8 colonnes :

`bwt` : le poids du bébé à la naissance, en kgs.

---

1. ou en tapant "R" dans un terminal (Applications ⇒ Accessoires ⇒ Terminal) sous Ubuntu

**weight** : le poids de la mère au début de la grossesse, en kgs.

**height** : la taille de la mère, en cm.

**age** : l'âge de la mère, en années.

**gestation** : la durée de la grossesse, en jours.

**parity** : T si c'est la première grossesse de la mère, F dans le cas contraire.

**smoke** : le fait que la mère fume ou non : F si elle ne fume pas, T si elle fume.

**tension** : la tension artérielle moyenne de la mère au cours de la grossesse.

## 2.1 Lecture des données

Le tableau de données est disponible sur un serveur Web. On peut le lire directement depuis le serveur, si l'on en connaît l'adresse, par la commande :

```
baby<-read.table("http://pbil.univ-lyon1.fr/R/donnees/TP_bioinfo_L2_baby.txt",header=TRUE)
```

On rappelle que l'option `header=TRUE` indique à  $\mathbb{R}$  que la première ligne du fichier contient les noms des variables. *Vous pouvez vous faire une idée du contenu du tableau de données avec les commandes :*

```
names(baby)
dim(baby)
head(baby)
```

## 2.2 Manipulation d'un data.frame

Vous allez tout d'abord vous familiariser avec la manipulation d'un `data.frame` comme `baby` :

1. Affichez uniquement la colonne `age` du `data.frame`.
2. Affichez le poids de la 4<sup>ème</sup> mère au début de la grossesse.
3. Affichez le poids des bébés dans les cas où il s'agit d'une première grossesse uniquement.

## 2.3 Filtrage des données

La toute première étape d'une analyse de données consiste généralement à vérifier que les données dont on dispose correspondent bien à la réalité. On ne sait pas encore quels sont les résultats qui vont en sortir, mais on peut être certains des supports de certaines variables : le sexe doit être codé en binaire, et l'âge d'un être humain ne peut pas être 900 ans<sup>2</sup>. Une façon de vérifier que les variables sont correctes consiste à tracer un simple graphe d'une seule variable : les données aberrantes y apparaîtront clairement. Par exemple, vérifions que l'âge des mères est raisonnable, en utilisant la commande graphique `plot` :

```
plot(baby$age)
```

Sur cette figure, on a l'âge en ordonnée, et la position en abscisse est simplement la position du point dans le jeu de données – ce qui n'a aucune importance pour nous, les points n'étant pas ordonnés. On remarque immédiatement deux points à près de 100 ans, un âge possible pour des êtres humains mais irréaliste pour des femmes venant d'accoucher. Ces points sont probablement des données manquantes, c'est-à-dire des femmes dont l'âge était indéterminé (ou simplement illisible sur le questionnaire). Plutôt que de laisser une case blanche, cette absence de données a ici été notée par un âge très élevé. Sous  $\mathbb{R}$ , on note de manière standard par `NA` les données manquantes (`NA`=Not Available, indisponible en anglais). On va donc remplacer les points absurdes par des `NA`, qui auront l'avantage de ne pas être pris en compte par  $\mathbb{R}$  (tandis que les valeurs absurdes le seraient ; imaginez que vous calculiez la moyenne des âges des

---

2. Yoda n'est pas humain

mères, les points proches de 100 la feraient augmenter). Pour remplacer les points absurdes par des NA, il faut indiquer ces points à `Q`. Il existe des manières semi-automatiques de faire ces changements ; ici nous allons les faire à la main, pour manipuler quelques outils simples.

*Essayez de taper les commandes suivantes et déterminez ce qu'elles renvoient comme réponse :*

```
baby$age>80
which(baby$age>80)
```

Une fois que vous avez identifié les points aberrants sur une colonne, vous pouvez remplacer la valeur en mémoire par un NA. Pour cela, si le 400<sup>ème</sup> point du vecteur `baby$age` est aberrant, il vous faut taper :

```
baby$age[400] <- NA
```

*De la même manière que précédemment, trouvez les données aberrantes dans ce tableau, sur cette colonne et les autres, et corrigez-les. Attention : un point ne peut pas être "un peu" aberrant : soit sa valeur est clairement du domaine de l'impossible, soit il s'agit d'un point extrême de l'expérience, peut-être inattendu.*

*Pour vérifier si toutes les variables ont bien été filtrées, vous pouvez utiliser la commande :*

```
summary(baby)
```

Celle-ci vous indique (entre autres) les valeurs minimales et maximales de chaque variable, et vous permet donc de déceler les valeurs aberrantes sans refaire le graphe à chaque fois.

### 3 Distribution d'une variable

Dans de nombreux cas, on doit supposer, pour pratiquer des tests sur des variables, que celles-ci sont distribuées normalement. Cette hypothèse, qui peut être vérifiée elle aussi de manière formelle par un test (test de Shapiro), peut également être abordée sous un angle graphique, en traçant la distribution des variables. Plusieurs fonctions existent pour cela ; ici on n'abordera que le tracé des histogrammes.

Si la variable est continue, on peut tracer un histogramme, comme ceci :

```
hist(baby$bwt)
```

*En lisant l'aide de cette fonction, essayez de comprendre le fonctionnement et le rôle des options `freq=FALSE` et `breaks`.*

*Étudiez la répartition des différentes variables de l'étude. Sont-elles toutes distribuées normalement ? Si oui, qu'en concluez-vous ? Si non, avez-vous une hypothèse explicative ?*

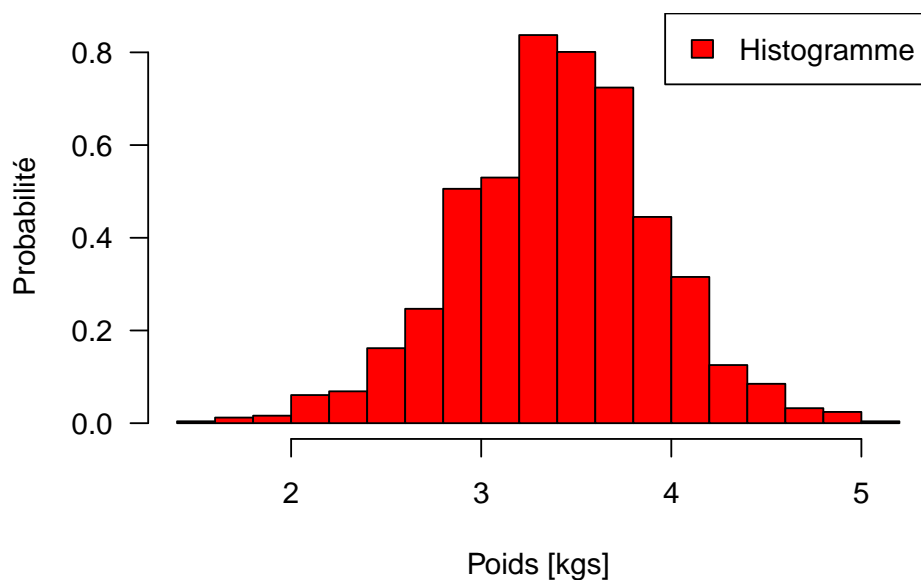
#### 3.1 Autour d'un graphe : légende, titre, unités...

Le graphe précédent est rudimentaire, et il est impossible à quelqu'un qui n'aurait pas lu toute cette fiche de savoir ce qu'il représente. À l'aide des options de la commande `hist` (qui sont les mêmes, pour certaines, que celles de la commande `plot` vue en cours), sur ce graphique :

- Ajoutez des descriptions des variables sur les 2 axes, avec les unités si possible.
- Ajoutez un titre.
- Coloriez votre histogramme en rouge
- (Facultatif) En utilisant la fonction `legend`, ajoutez une légende.

Le graphe final peut ressembler à cela (ou pas, vos choix graphiques sont personnels, tant qu'ils sont lisibles) :

## Répartition des poids des bébés à la naissance



## 4 Variables quantitatives

Une fois que l'on a filtré et vérifié graphiquement la normalité des variables, on peut commencer à étudier les relations que l'on peut trouver entre elles. Dans cette étude, le but est d'expliquer quels sont les facteurs importants qui influencent le poids du bébé à la naissance, facteur dont on sait qu'il est très corrélé à la bonne santé générale du bébé et en particulier aux problèmes de mort subite du nourrisson. Une idée simple est que le poids de la mère doit être un déterminant du poids du bébé, en particulier car il reflète l'alimentation. *Pour observer cela, tracez le poids du bébé en fonction du poids de la mère :*

```
plot(x=baby$weight,y=baby$bwt)
```

*Ajoutez une droite de régression à ce graphique (commande `abline`, prenez exemple sur le cours pour la syntaxe). Vous pouvez également changer le type de points utilisés, avec l'option `pch`, ou changer la couleur avec l'option `col`. N'oubliez pas titres et légendes. Que pensez-vous de notre hypothèse, au vu du graphique ?*

## 5 Variables qualitatives

Il arrive souvent de vouloir comparer des groupes d'individus, où l'appartenance est définie par une variable qualitative à plusieurs modalités : le sexe, le fait de fumer ou non, ou encore une division en groupes comme {"pas du tout sportif", "peu sportif", "sportif", "très sportif"}. Même si la comparaison réelle se fait à l'aide d'un test, une bonne visualisation des données peut permettre de comprendre leur structure et ainsi de poser les questions appropriées lors d'un test. Nous allons voir deux manières de faire ce type de représentation.

### 5.1 Usage des couleurs à bon escient

Une manière utile d'employer les couleurs sur un graphe est de représenter, sur le graphe de deux variables quantitatives (comme le poids du bébé et la durée de la grossesse), une information qualitative supplémentaire en colorant les points d'une façon qui dépend de cette variable. Par exemple, si on veut

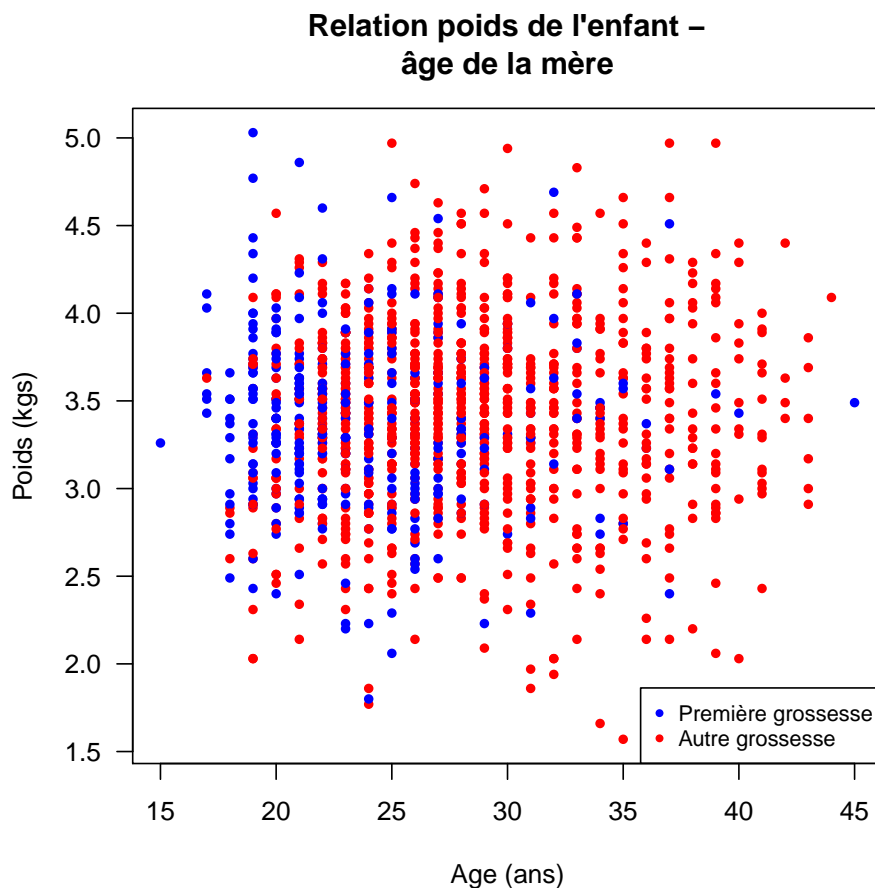
faire un graphe du poids d'une variable quantitative Y en fonction d'une autre variable quantitative X et d'une variable qualitative  $\alpha$ , on doit :

1. Tracer avec `plot` le graphe de Y en fonction de X, uniquement pour les points ayant une valeur donnée de  $\alpha$  (voir la partie 1.1), d'une couleur particulière (option `col`).
2. Utiliser la fonction `points`, qui fonctionne comme la fonction `plot`, pour tracer le graphe de Y en fonction de X, uniquement pour les points ayant une autre valeur de  $\alpha$ , dans une autre couleur.

On ne peut pas employer deux fois la fonction `plot`, car celle-ci trace à chaque fois un nouveau graphe, tandis que `points`, comme `lines`, ajoute des points sur un graphe déjà existant. Pour sélectionner les points correspondant à une valeur de  $\alpha$  et uniquement ceux-ci, on peut utiliser les indexations conditionnelles. Par exemple, les valeurs du poids du bébé dans les cas où la mère fume sont sélectionnées avec :

```
baby$bwt[baby$smoke==1]
```

Tracez maintenant (sans oublier le double signe "="!) avec deux couleurs, le poids du bébé en fonction de l'âge de la mère et du fait que ce soit la première grossesse ou non. Le graphique devrait ressembler à ça (la légende en bas à gauche s'obtient avec la fonction `legend`) :



*Remarquez-vous quelque chose de spécial?*

Il existe une autre méthode pour tracer ce genre de graphe en une seule commande. Pour cela, il faut définir un vecteur de couleurs qui corresponde à la variable quantitative que l'on veut représenter : ce vecteur doit contenir une série de couleurs dans le même ordre que la variable qualitative, avec une couleur différente par modalité. La fonction `ifelse`, qui vérifie si une condition est réalisée ou non dans un vecteur, et renvoie une réponse ou un autre en fonction de cela, est appropriée pour faire ce type de manipulation. Si on veut employer la variable `smoke` pour définir les groupes, on peut faire, par exemple :

```
ifelse(baby$smoke==1,"red","green") -> couleur_smoke
```

et ensuite dans le `plot` de toutes les données, passer comme option de couleur `col=couleur_smoke`.  $\mathcal{R}$  associe alors à chaque point la couleur qui correspond à la valeur de `smoke` correspondante (pour que cela marche, il faut que les valeurs de `smoke` et les valeurs à tracer soient dans le même ordre, comme ici). *Tracez maintenant avec deux couleurs, le poids du bébé en fonction de la durée de la grossesse et du fait que la mère fume ou non. Des remarques ?*

## 5.2 Boxplot

Une autre façon de représenter des groupes de données, plus directe mais moins intuitive au niveau graphique, consiste à représenter la dispersion des données en fonction du groupe. Ceci est très inspiré au niveau théorique de l'ANOVA, où l'on teste comment la variance globale du jeu de données se répartit dans les différentes modalités du groupe. La fonction graphique à employer est la fonction `boxplot`. *Par exemple, pour tracer le poids de la mère en fonction de son nombre de grossesses antérieures, on tapera :*

```
boxplot(baby$weight~baby$parity)
```

Le symbole `~` signifie sous  $\mathcal{R}$  "en fonction de", et est également employé dans les commandes pour réaliser des ANOVA. Une option utile des `boxplot` est l'option `notch=TRUE` ; en l'ajoutant, des encoches vont se placer sur les boîtes tracées. Ces encoches correspondent à l'intervalle de confiance autour de la médiane de la variable étudiée ; si les encoches de deux boîtes ne se chevauchent pas, on peut considérer que les valeurs de la variable dans les deux groupes sont significativement différentes.

*Observez-vous une influence du fait que la mère fume sur le poids du bébé à la naissance ?*

Si on veut utiliser un `boxplot` et visualiser l'information de l'effectif dans chacun des groupes, ici par exemple combien de mères ont déjà été enceintes auparavant, on peut utiliser l'option `varwidth=TRUE`, qui rend la largeur de chaque boîte proportionnelle à l'effectif du groupe. Pour savoir exactement combien de mères appartiennent à chaque catégorie, on peut employer la commande :

```
table(baby$parity)
```

Attention, cette commande n'est utile qu'avec des variables qualitatives (essayez avec des variables continues et vous verrez le problème...).

## 6 Questions ouvertes

À l'aide de ce jeu de données, et des techniques que vous avez apprises durant le TP, que pouvez-vous dire sur le poids du bébé à la naissance ? En particulier, à l'aide de graphes de votre choix, essayez de démêler l'écheveau suivant :

- *Vérifiez séparément l'influence de l'âge de la mère, de la parité, du poids de la mère et du fait de fumer sur le poids du bébé.*
- *Vérifiez si les variables précédentes sont ou non indépendantes ; laquelle est le meilleur prédicteur du poids du bébé ?*
- *Essayez de synthétiser vos conclusions aux deux questions précédentes pour répondre à l'objet de l'étude, qui était de savoir si, "toutes choses étant égales par ailleurs", le fait de fumer affecte le poids du bébé à la naissance.*
- *Imaginons maintenant que l'on veuille se focaliser non pas sur le poids du bébé, mais sur la tension de la mère en fonction des autres caractéristiques. Appliquez la même démarche que ci-dessus et trouvez les déterminants de la tension.*