L3 Pro "Biotechnologies végétales et création variétale"
TP 1 : graphiques avec R

Marc Bailly-Bechet

Automne 2015

1 Introduction

L'objectif de ce TP est de vous faire utiliser le logiciel R pour procéder à l'exploration préliminaire d'un jeu de données. Le but sera essentiellement d'utiliser les commandes graphiques pour visualiser des données – la formulation d'hypothèses et les tests intervenant souvent après coup dans la pratique. Dans la suite du TP, les phrases en italique sont les questions auxquelles vous devez répondre.

Il vous est rappelé que, pour obtenir l'aide sur une fonction `machin`, il vous suffit, dans une console R, 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. N'hésitez pas à retourner voir les exemples du premier TP si vous ne vous souvenez plus comment faire un calcul !

1.1 Re-prise en main

Vous allez commencer par quelques commandes simples pour vous rappeler du premier TP :

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^e élément du vecteur `v` créé précédemment.

5. Multipliez la première valeur de votre vecteur par 2, la deuxième par trois, la troisième par quatre, etc... (il existe plusieurs méthodes pour réaliser cette opération)

2 Lecture et filtrage des données

Les données que vous aller 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 d'apprendre à réaliser des représentations graphiques, et à le sutiliser pour comprendre comment les différentes variables sont reliées entre elles.

Les données se présentent sous la forme d'un `data.frame` à 7 colonnes :

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

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.

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 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 revoir les bases de la manipulation d'un `data.frame` comme `baby` – une petite application de ce que vous avez vu dans le premier TP :

1. Affichez uniquement la colonne `age` du data.frame.
2. Affichez le poids au début de la grossesse de la $4^{\text{ème}}$ mère.
3. Affichez le poids de tous les bébés.
4. Affichez un vecteur `TRUE FALSE` indiquant si la mère a plus ou moins de 35 ans.
5. Combinez les deux questions précédentes pour, en un seule commande, affichez le poids des bébés dont les mères avaient plus de 35 ans.
6. Affichez le poids des bébés dans les cas ou il s'agit d'une première grossesse uniquement (indice : ce peut être un peu plus facile qu'avant).

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 sur 2 modalités uniquement, et l'âge d'un être humain ne peut pas être 900 ans [1]. Une façon simple de vérifier que les variables sont correctes consiste à tracer un 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 abcisse 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. C'est une très mauvaise idée : si on ne faisait pas atention, on aurait pu commettre des erreurs dans l'analyse. Pour éviter cela, sous 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 R (tandis que les valeurs absurdes le seraient ; imaginez que vous calculiez la moyenne des âges des mères, les points proches de 100 la feraient augmenter, exactement comme l'entrée de Bill Gates dans cette salle ferait augmenter dramatiquement le revenu moyen par personne). Pour remplacer les points absurdes par des `NA`, il faut indiquer ces points à R. Il existe des manières semi-automatiques de faire ces changements ; ici nous allons les faire à la main, pour manipuler quelques outils simples.

Tapez 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$^{\text{ème}}$ point du vecteur `baby$age` est aberrant, il vous faut taper :

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

Pour remplacer toutes les valeurs des ages supérieurs à 80 and par des `NA`, on peut employer l'indexation vectorielle :

```
baby$age[baby$age>80] <- 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

1. Yoda n'est pas humain

extrême de l'expérience, peut-être inattendu. Essayez d'identifier valeurs extrêmes et points aberrants : comment les distinguer ?

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

3.1 Variable continue

Il est souvent intéressant, avant de commencer une étude statistique, de traçer la distribution des variables. Cela permet de regarder notamment si les valeurs sont groupées autour de la moyenne ou très étalées, si elles se répartissent en plusieurs groupes... Plusieurs fonctions existent pour cela ; ici on n'abordera que le tracé des histogrammes, et facultativement, la densité.

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

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

On peut passer des options à une fonction graphique, comme à d'autres fonctions, en les ajoutant dans les parenthèses, séparées par des virgules. Une des options de `hist` est `freq`. Cette option peut prendre deux valeurs, `TRUE` ou `FALSE`. *Essayez par exemple* :

```
hist(baby$bwt,freq=TRUE)
```

Une autre option est `breaks=n`, où l'on remplace `n` par le nombre de son choix.

Essayez de comprendre le fonctionnement et le rôle des options ***`freq`*** *et* ***`breaks`***

Étudiez la répartition des différentes variables de l'étude. Sont-elles toutes distribuées normalement, c-à-d. regroupées autour de leur moyenne de manière symétrique ? Si oui, qu'en concluez-vous ? Si non, avez-vous une hypothèse explicative ?

3.2 FACULTATIF – densité d'une variable continue

Une autre manière d'étudier la distribution d'une variable continue est de tracer la densité de cette variable. Cette densité est une sorte d'histogramme continu, où l'ordonnée est une mesure de probabilité (comme avec l'option `freq=FALSE` de la commande `hist`). On peut la tracer avec la commande :

```
plot(density(baby$bwt))
```

Pour comparer les deux, on peut :

1. *Tracer l'histogramme de la variable, avec* ***`freq=FALSE`****, comme précedemment.*
2. *Ajouter la densité en se servant de la commande* ***`lines`*** *au lieu de* ***`plot`****. La commande* ***`lines`*** *ajoute une courbe à un graphe déjà existant. Pour plus de clarté, on peut ajouter cette densité d'une couleur différente (option* `col`*). Que se passe-t-il si vous ajoutez la densité avec* ***`plot`*** *?*

3. *Pour bien voir le lien entre les deux, vous pouvez chercher à faire un histogramme avec des barres de plus en plus fines.*

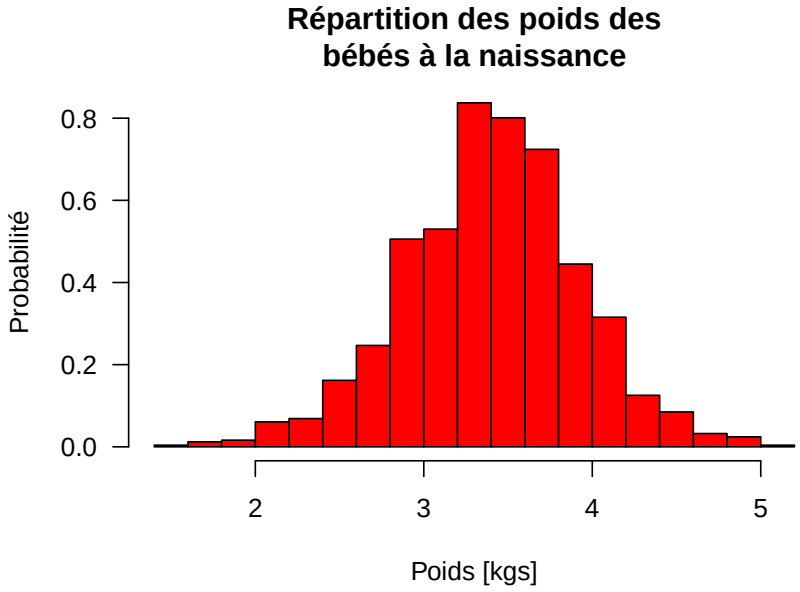
Un des paramètres essentiels du tracé de densité est l'option `adjust`. En diminuant sa valeur, on obtient une densité qui devient de moins en moins lisse. *Que se passe-t-il quand on change la valeur de cette option ?*

3.3 Autour d'un graphe : titre, unités, couleur...

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 graphiques de base, que l'on peut utiliser sur toutes les commandes, sur ce graphique :*

— *Ajoutez des descriptions des variables sur les 2 axes, avec les unités si possible (`xlab` et `ylab`).*
— *Ajoutez un titre (`main`).*
— *Coloriez votre histogramme en rouge (`col`).*
— *Ecrivez les labels de l'axe des ordonnées dans le bon sens (`las`)*

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



3.4 Variable discrète

Quand on veut étudier la répartition d'une variable discrète [2], on ne peut pas employer les fonctions précédentes. On va utiliser une fonction prévue pour ce type de variable. Dans notre jeu de données, deux variables sont discrètes : la variable `parity` et la variable `smoke`.

Le fait que la variable soit discrète crée une catégorisation naturelle : on va représenter le nombre d'individus ayant chaque valeur (par exemple la taille), sans avoir besoin de regrouper les classes, ou de définir à la main le nombre d'intervalles que l'on veut. La fonction `table`

2. Les variables discrètes n'ont rien à voir avec les variables cachées

joue ce rôle : elle compte le nombre d'individus ayant chaque valeur. *Vous pouvez ensuite représenter ce nombre avec la fonction* `barplot` *:*

```
table(baby$smoke)
```

```
FALSE  TRUE
  742   484
```

```
barplot(table(baby$smoke))
```

Essayez d'appliquer cette méthode sur les différentes variables. Dans quels cas cela fonctionne-t-il mal ? Pourquoi ?

4 Relation entre variables quantitatives

Une fois que l'on a filtré et observé les variables une à une, 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)
```

Vous pouvez 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 ? Faites la même analyse (formulation d'hypothèse, réalisation graphique et conclusion) sur la question du lien entre durée de la grossesse et le poids du bébé à la naissance.*

4.1 FACULTATIF – Cartes de densité

De la même manière que l'on a pu regarder l'histogramme d'une variable quantitative avec `hist`, on peut être amené à vouloir étudier la répartition combinée de deux variables quantitatives. C'est le cas lorsque le graphe de l'une en fonction de l'autre contient beaucoup de points qui se superposent, ce qui ne permet pas de juger de la relation entre les variables par la forme du nuage. On dispose ici de relativement peu de points pour ce genre d'application, donc le résultat graphique ne sera pas forcément parfait, mais on va néanmoins faire une exemple. On doit tout d'abord charger une bibliothèque de fonctions qui est désactivée par défaut :

```
library(MASS)
```

Ensuite, on va utiliser la fonction `kde2d` pour calculer la densité combinée des deux variables. Cette fonction n'est qu'un intermédiaire ; il faut ensuite utiliser une fonction graphique particulière pour tracer le résultat. *Essayez :*

```
baby_filter<-na.omit(baby)
x<-kde2d(x=baby_filter$height,y=baby_filter$weight)
image(x)
contour(x)
```

Le rôle de la commande `na.omit` est de retirer les lignes contenant des valeurs manquantes du jeu de données, sans quoi la fonction `kde2d` ne fonctionne pas. *Vous pouvez augmenter la "résolution" avec l'option* `n` *de* `kde2d`*, qui est fixé par défaut à 25. Augmenter cette valeur augmente le nombre de points sur le graphe final. Pour obtenir une meilleure image, vous pouvez également choisir des bornes plus serrées que celles par défaut, avec les options* `xlim` *et* `ylim` *dans* `kd2ed`*.*

5 Relation impliquant des 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"}. 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. 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 α, on doit :

1. Tracer avec `plot` le graphe de Y en fonction de X, uniquement pour les points ayant une valeur donnée de α (voir la partie 1.1), d'une couleur particulière (option `col`). . Pour sélectionner les points correspondants à une valeur de α et uniquement ceux-ci, on peut utiliser les indexations conditionelles. Par exemple, pour regarder le poids du bébé en fonction de l'âge de la mère *et* du fait que ce soit la première grossesse ou non, on va commencer par *tracer le graphe du poids du bébé en fonction de l'âge de la mère, uniquement pour les premières grossesses* :

   ```
   ### Les lignes précédées de ### ne servent à rien et sont des commentaires
   ### Le code est sur 3 lignes pour rentrer sur la feuille de TP.
   ### Pour vous pas besoin  d'aller à la ligne!
   plot(baby$age[baby$parity==T],baby$bwt[baby$parity==T],pch=20,
   col="blue", main="Relation poids de l'enfant -\nâge de la mère",
   xlab="Age (ans)",ylab="Poids (kgs)",las=1)
   ```

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 α, dans une autre couleur. *Pour continuer l'exemple* :

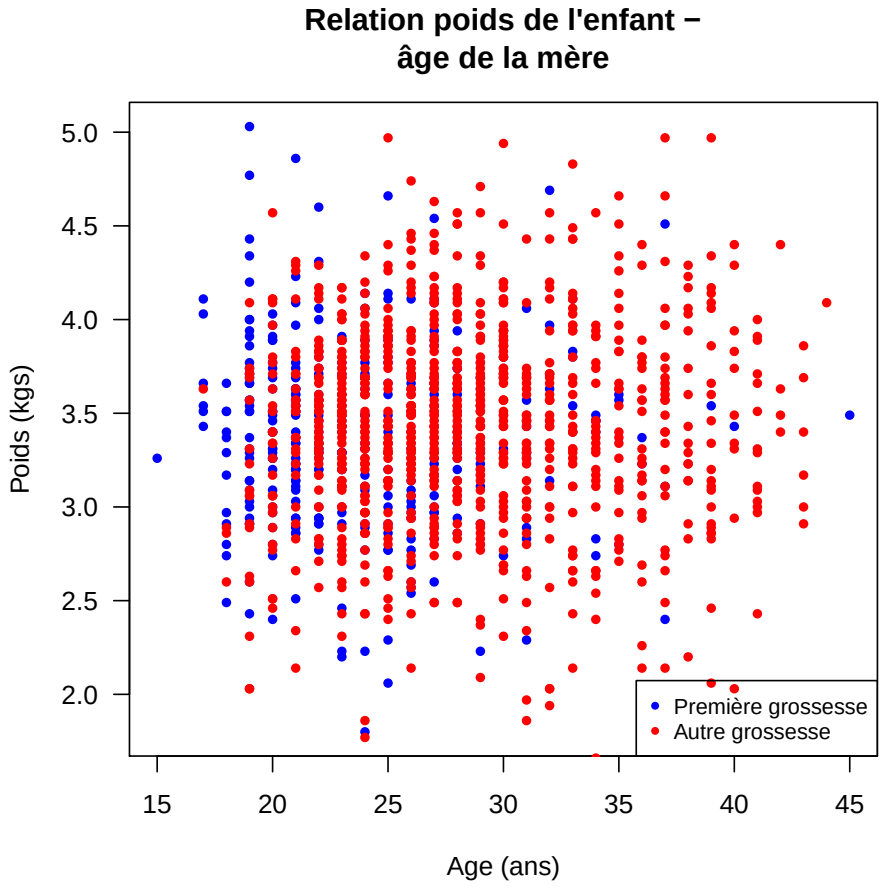
```
points(baby$age[baby$parity==F],baby$bwt[baby$parity==F],pch=20,col="red")
```

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.

Dans ce genre de cas, il est difficile *a posteriori* de savoir qui est qui. Il faut donc ajouter une légende sur le graphe. On peut le faire comme cela :

```
legend(x="bottomright",legend=c("Première grossesse","Autre grossesse"),
col=c("blue","red"),pch=20)
```

Le graphique final devrait ressembler à ca :



Remarquez-vous quelque chose de spécial ?

FACULTATIF 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. Si on veut employer la variable `parity` pour définir les groupes, on peut faire, par exemple :

```
ifelse(baby$parity==T,"blue","red") -> couleur_parity
```

et ensuite dans le `plot` de toutes les données, passer comme option de couleur `col=couleur_parity`. Par exemple :

```
plot(baby$age,baby$bwt, main="Relation poids de l'enfant -\nâge de la mère",
col=couleur_parity,xlab="Age (ans)",ylab="Poids (kgs)",pch=20, las=1)
```

Tracez maintenant avec deux couleurs, à l'aide de la méthode de votre choix, 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 (vous verrez cela lors du dernier TP !). La fonction graphique à employer est la fonction `boxplot`. *Par exemple, pour traçer le poids de la mère en fonction de son nombre de grossesses antérieures, on tapera :*

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

Le symbole ~ signifie sous R "en fonction de", et est également employé dans les commandes pour réaliser des ANOVA. La boîte centrale pour chaque variable représente l'intervalle dans lequel 50% de l'échantillon est distribué ; les moustaches s'étendent par défaut à 1.5 fois cette distance, et cela peut être modifié (option `range`). 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.

À vous : avec cette méthode, 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.

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 ? Est-il particulièrement dépendant d'une autre variable de l'étude ? De plusieurs ? Ces différentes variables causales sont-elles elles-mêmes reliées entre elles ? Certains résultats vous paraissent-ils étonnants ?