TP 1 : Statistiques descriptives

Le logiciel `R` est un logiciel de statistiques largement utilisé. Il est gratuit et vous pourrez donc l'installer sans aucun problème sur votre ordinateur — si vous en possédez un. Vous pourrez le télécharger, consulter des manuels et bien d'autres choses en suivant ce lien. Je vous recommande particulièrement ce manuel pour débuter.

Le logiciel a une interface graphique mais sur les ordinateurs de la fac, vous n'y aurez pas accès malheureusement... Tout se passera donc dans un terminal. Pour lancer le logiciel il suffit de taper (dans un terminal donc) `R`. Si vous ne connaissez pas le rôle d'une fonction, vous pourrez lire l'aide en tapant `?nom_fonction_inconnue`.

Votre enseignant devrait maintenant vous faire une rapide présentation des commandes `R` servant à manipuler les données. N'hésitez pas à poser des questions si vous êtes perdu...

1 Importation et préparation des données

Nous allons pour ce premier TP utiliser les données du chapitre 1 du cours sur le poids des bébés. Pour cela, il nous faut importer nos données via la fonction `read.table` et la stocker dans un objet (ici l'objet `data`).

```
data <- read.table("chemin_du_fichier_de_donnees", header = TRUE)
```

L'option `header = TRUE` indique que la première ligne de notre fichier contient le nom des variables. Essayer d'importer les données `babies23.data`.

Vous pourrez alors visualiser les données en tapant `data`. Afin de comprendre le sens de ces variables, il est indispensable de lire le fichier `babies.readme.txt`. On remarque qu'il y a deux variables nommées `wt`. Comment `R` a t il géré ce problème ? Que se passe-t-il si vous tapez par exemple `wt` ? Qu'en concluez vous ? Et si vous tapez `data$wt` ?

Il est souvent utile « d'attacher » toutes les variables de sorte que l'appel à leur nom sera suffisant au lieu de faire `data$wt`. Pour cela tapez `attach(data)`, puis essayer de taper `wt`. Pratique non ?

Nous allons nous intéresser plus particulièrement aux poids du bébé, de la mère, du père ainsi que le statut fumeur ou non de la mère. Je vous conseille donc d'enregistrer ces variables dans de nouveaux objets ayant des noms plus « parlants » ; par exemple

```
poids.bebe <- wt
poids.mere <- wt.1
poids.pere <- dwt
fume <- smoke
```

La variable `fume` est un *facteur*, elle n'est pas vraiment numérique puisqu'on aurait tout autant mettre des 'A', 'B', 'C', ... plutôt que des nombres. En particulier, la valeur '9' indique une valeur manquante. Le logiciel `R` code les valeurs manquantes par des `NA` (not available). La fonction `replace` va nous aider à changer ces '9' en 'NA' :

```
fume <- replace(fume, fume == 9, NA)
```

Notez que l'appel à `replace` ne modifie en rien l'objet, il faut donc le « réécrire » dans le même objet (ou un autre) pour qu'il y ait un effet. Recodez également les valeurs manquantes pour les trois autres variables `poids.bebe`, `poids.mere` et `poids.pere`.

Pour notre étude, on s'intéresse uniquement aux mères fumant pendant leur grossesses, de sorte qu'on aimerait que la variable `fume` soit une variable binaire valant '1' si elle fumait pendant la grossesse et '0' sinon. Utilisez la fonction `replace` et l'opérateur logique `|` afin de coder cette variable proprement.

Enfin la dernière étape consiste à dire que la variable `fume` est un facteur. Pour cela faire

```
fume <- factor(fume)
```

Si vous êtes arrivé ici, vos données devraient être prêtes pour l'analyse!!!

2 Analyse descriptive

Il est toujours bon de représenter graphiquement nos données avant de commencer. La fonction générique sous `R` est la fonction `plot`. Afin de faire 4 figures en une seule, on peut dire au logiciel de diviser la région graphique en plusieurs sous régions. Cela se fait via la commande

```
par(mfrow = c(1, 4))
```

ici nous divisons donc la région graphique en un « tableau » ayant 1 ligne et 4 colonnes. Mais nous aurions également pu faire 2 lignes et 2 colonnes en précisant `par(mfrow = c(2, 2))`. On affichera nos données sous forme graphique en faisant alors

```
plot(poids.bebe)
plot(poids.mere)
plot(poids.pere)
plot(fume)
```

Commentez le graphique obtenu. Notez que vous pouvez redimensionner votre graphique à votre convenance ; vous pouvez également l'enregistrer en pdf tant que la fenêtre graphique est ouverte (d'autres formats sont également disponibles) en faisant

```
dev.copy2pdf(file = "nom_de_mon_graphique.pdf")
```

Pour obtenir un résumé numérique rapide d'une variable on utilise la fonction `summary`. Appliquez cette fonctions aux quatre variables qui nous intéressent et interprétez les sorties. Notez que cette fonction fait appel à plusieurs fonctions : `min`, `max`, `mean`, `median` et `quantile`. Essayer d'utiliser ces fonctions, vous aurez peut-être besoin de passer l'option `na.rm = TRUE` pour ignorer les valeurs manquantes.

Nous allons maintenant faire un histogramme de la taille des bébés. Tapez les commandes suivantes et interprétez.

```
par(mfrow = c(1, 3))
hist(poids.bebe)
hist(poids.bebe, freq = FALSE)
hist(poids.bebe, breaks = seq(50, 200, by = 10), freq = FALSE)
```

A quoi sert l'option `freq = FALSE` ? Et l'option `breaks` ?

Nous allons maintenant regarder le poids des bébés selon le statut fumeur ou non de la mère. Essayez de comprendre la commande suivante et interprétez

```
summary(poids.bebe[fume == 0])
summary(poids.bebe[fume == 1])
```

Faites de même pour ces autres commandes.

```
hist(poids.bebe[fume == 0], freq = FALSE, col = "lightgrey",
     xlab = "Poids (onces)", main = "")
hist(poids.bebe[fume == 1], density = 5, freq = FALSE, add = TRUE, col = "blue")
legend("topleft", c("Non fumeur", "Fumeur"), col = c("lightgrey", "blue"),
       lty = 1, bty = "n", inset = 0.05)
```

Selon vous à quoi pourraient servir les deux graphiques suivants ? Vous aurez besoin de comprendre à quoi sert la fonction `qqline` en allant voir l'aide, i.e., tapez `?qqline`.

```
par(mfrow = c(1, 2))
qqnorm(poids.bebe[fume == 0])
qqline(poids.bebe[fume == 0], col = 2, lty = 2)
qqnorm(poids.bebe[fume == 1])
qqline(poids.bebe[fume == 1], col = 2, lty = 2)
```

Quelles conclusions pouvez vous en tirer ? À partir du graphique précédent, justifiez les commandes suivantes :

```
hist(poids.bebe[fume == 1], freq = FALSE, xlab = "Poids (fumeur)",
     xlim = c(40, 200), main = "")
mu <- mean(poids.bebe[fume == 1], na.rm = TRUE)
sigma <- sd(poids.bebe[fume == 1], na.rm = TRUE)
lines(seq(40, 200, by = 1), dnorm(seq(40, 200, by = 1), mu, sigma), col = 2)
```

Utilisez l'approximation normale pour le poids des bébés dont les mères fument, pour estimer le quantile à 85%, 95% et 99%. Comparer avec les quantiles empiriques. Quel est l'intérêt de cette approximation normale ?

Rappelez le nom et le but du graphique suivant ?

```
qqplot(poids.bebe[fume == 0], poids.bebe[fume == 1],
       ylim = range(poids.bebe), xlim = range(poids.bebe),
       xlab = "Poids (Non fumeur)", ylab = "Poids (fumeur)")
abline(0, 1)
```

Vous avez fini votre première session sous `R`. Pour quitter vous pouvez fermer le terminal ou encore taper `q()`. Il vous sera alors demandé si vous voulez enregistrer votre travail. Par défaut, votre travail sera enregistrer dans un fichier caché `.RData`.