T. D. nº 3
Révision des statistiques descriptives sur une série de mesures et compléments

Exercice 1 Jeu de données `Europe`, page 147 du livre « Initiation à la statistique avec R ».
Le but de cet exercice est de réviser les commandes qui permettent de calculer les résumés numériques et de tracer une boîte à moustaches, découvertes dans le T.D. 2.

1. Afficher les six premières lignes de ce jeu de données qui est disponible dans la bibliothèque `BioStatR`.
2. De quoi est constitué ce jeu de données ? C'est-à-dire :
   — Combien d'unités statistiques sont présentes dans ce jeu de données ?
   — Combien de variables composent ce jeu de données ?
   — Quelle est la nature de chacune de ces variables ?
3. Quelle est la classe et la taille de ce jeu de données ?
4. Donner la moyenne, la valeur minimale, la valeur maximale, la médiane, le premier quartile, le troisième quartile et la ou les classes modale(s) de la variable `Duree`.
5. Donner la variance, l'écart-type corrigé, l'étendue, l'étendue interquartile, le MAD de la variable `Duree` .
6. Donner le coefficient de variation.
7. Tracer la boîte à moustaches de la variable `Duree` en mettant un label pour l'axe vertical qui est « Durée en heures ». Représenter sur cette même boîte la moyenne.
8. Sauvegarder la boîte à moustaches au format .pdf en utilisant la fonction `pdf()`.

Exercice 2 Fonction factor, pages 143 à 145 du livre « Initiation à la statistique avec R ».
Dans cet exercice, vous allez découvrir comment fonctionne la fonction `factor` *que vous devez connaître pour la suite.*

Sur trois variétés de pommes notées 1, 2 et 3, la jutosité de chaque pomme est relevée. La jutosité est un indice compris entre 0 et 10. Il y a quatre pommes par variété qui ont été testées. La variété 1 est la Golden Delicious, la variété 2 est la pomme Calville et la variété 3 est la Belle de Boskoop. Vraisemblablement, la

question que vous pourriez vous poser serait : « quelle est la variété de pomme la plus juteuse ? » Vous ne chercherez pas à répondre à cette question ici. En effet, il s'agit d'une application d'une technique statistique connue sous le nom d'analyse de la variance que vous ne connaissez pas encore. Le but de cet exercice est de vous montrer comment vous servir de la fonction `factor()`. Les résultats obtenus sont inscrits dans le tableau suivant :

Variété de pomme	Jutosité	Variété de pomme	Jutosité
1	4	2	7
1	6	2	6
1	3	3	8
1	5	3	6
2	7	3	5
2	8	3	6

1. Rentrer les données sous `R` en introduisant deux variables :
   — une première variable que vous noterez `Variete` et
   — une seconde variable que vous noterez `Jutosite`.
   À l'issue de cette opération, construiser un `data.frame` dont le nom est `Pommes`.
2. Donner la structure du jeu de données `Pommes` que vous venez de construire à la question précédente. Que constatez-vous ?
   Il faut donc transformer la variable `Variete` en un `factor`.
3. Pour transformer un vecteur de type numérique ou entier, vous pouvez utiliser la fonction `factor()`. Ainsi pour transformer la variable `Variete` qui est pour l'instant de mode `numeric`, vous tapez la ligne de commande suivante :
   ```
   > Variete<-factor(Variete)
   ```
   puis
   ```
   > Pommes<-data.frame(Variete,Jutosite)
   > rm(Variete)
   > rm(Jutosite)
   ```
   Quelle est la nature du jeu de données `Pommes` ? Quels sont les modes des deux variables qui constituent le jeu de données `Pommes` ?
   Remarque : `rm()` pour « remove ».
4. Vous auriez pu procéder autrement. Cette seconde façon est beaucoup plus rapide et vous êtes invité à vous en servir dès que vous savez qu'une variable dans votre jeu de données est un facteur.
   Taper les lignes de commandes suivantes :
   ```
   > Variete<-factor(c(rep(1,4),rep(2,4),rep(3,4)))
   > Jutosite<-c(4,6,3,5,7,8,7,6,8,6,5,6)
   > Pommes<-data.frame(Variete,Jutosite)
   ```
   Qu'obtenez-vous ? Avez-vous le même résultat qu'auparavant, c'est-à-dire la même structure pour le jeu de données `Pommes` ?
5. Il vous est conseillé, au moins dans les premiers temps de votre apprentissage de la statistique, de ne pas utiliser des nombres pour les niveaux de votre facteur, mais plutôt des lettres. Pour cela, vous utiliserez l'option `labels` dans

la fonction `factor`.
Vous allez donner un label aux valeurs numériques `1`, `2` et `3`, à savoir `1` devient `V1`, `2` devient `V2` et `3` devient `V3`, `V` pour `Variete`. Pour cela, taper les lignes de commande suivantes :

```
> Variete<-factor(c(rep(1,4),rep(2,4),rep(3,4)),labels=c("V1","V2","V3"))
> Jutosite<-c(4,6,3,5,7,8,7,6,8,6,5,6)
> Pommes<-data.frame(Variete,Jutosite)
```

Qu'obtenez-vous ? Il y a quelque chose qui a changé. Pouvez-vous dire quoi ?

6. Enfin, il existe une fonction `as.factor()` qui permet d'arriver au même résultat.
   Taper les lignes de commande suivantes :

```
> Variete<-as.factor(c(rep(1,4),rep(2,4),rep(3,4)))
> Jutosite<-c(4,6,3,5,7,8,7,6,8,6,5,6)
> Pommes<-data.frame(Variete,Jutosite)
```

   Vérifier bien que vous obtenez le même résultat qui est attendu.
7. Calculer les moyennes pour chacun des groupes défini par la variable `Variete` en utilisant la fonction `tapply` :

```
> tapply(Jutosite,Variete,mean)
```

   Procéder de même pour obtenir l'écart-type, les quantiles ou appliquer la fonction `summary` à chacun des groupes défini par le facteur `Variete`.

Exercice 3 Comment grouper des données ?, pages 145 et 146 du livre « Initiation à la statistique avec R ».
Vous allez vous intéresser ici à la variable masse du jeu de données `Mesures`. *Dans les rappels de cours, vous avez vu comment grouper ces données automatiquement à l'aide de l'une des trois règles dues à Sturges, Scott et Freedman-Diaconis. Vous allez voir comment grouper des données suivant différents critères.*

a) Groupez les données en 5 classes à l'aide de l'option `breaks=5` de la fonction `hist`. Que se passe-t-il si vous cherchez à en obtenir seulement 4 ?

b) Groupez les données en utilisant les classes suivantes $[0;5]$, $]5;10]$, $]10;15]$, $]15;20]$ et $]20;50]$ à l'aide de l'option `breaks=c(0,5,10,15,20,50)` de la fonction `hist`.

c) Comparez le résultat obtenu avec :

```
> brk <- c(0,5,10,15,20,50)
> table(cut(Mesures$masse, brk))
> data.frame(table(cut(Mesures$masse, brk)))
```

d) Si vous cherchez à créer des groupes dont les effectifs sont équilibrés, vous pouvez par exemple utiliser la fonction `cut2` de la bibliothèque `Hmisc`. Après avoir téléchargé et installé cette bibliothèque, commentez les lignes de code suivantes et en particulier le rôle des options `g` et `m`.

```
> library(Hmisc)
> brk <- c(0,5,10,15,20,50)
```

```
> res <- cut2(Mesures$masse, brk)
> table(res)
> table(cut2(Mesures$masse, g=10))
> table(cut2(Mesures$masse, m=50))
```