TP : Introduction à R

Objectif :
— prendre en main les concepts de base du logiciel R

1 Introduction

R [1] est un logiciel de statistiques gratuit et open-source, disponible sous Linux, Windows et OS X.

Pour lancer R, tapez **R** en ligne de commande. Il est également possible d'écrire des scripts et de les interpréter avec R. Pour obtenir de l'aide en ligne sur une commande, il suffit de la faire précéder du symbole ?. Les flèches haut et bas permettent de naviguer dans l'historique de commandes. Les commentaires sont précédés par le symbole #. La touche tab peut être utilisée pour la complétion des commandes et des noms de variables.

2 Premiers pas

Le moyen le plus rapide d'entrer un faible nombre de données dans R est la fonction c (*combine* operator). Ainsi la commande suivante permet d'affecter un ensemble de réels à la variable v. Pour visualiser le contenu d'une variable, il suffit de taper son nom. Dans le cas présent le [1] qui est affiché indique que la variable est de type vecteur. Calculer la moyenne, médiane, variance ... de ces valeurs s'obtient simplement en tapant le nom de la fonction correspondante en passant le nom de la variable en paramètre. Notez que le symbole = est utilisé pour l'affectation mais il est également possible d'utiliser <-.

Modifier le contenu d'un vecteur peut se faire en indiquant l'indice de l'élément à modifier entre crochets. De nombreuses opérations sont possibles sur les vecteurs comme déterminer la somme des éléments dont la valeur est supérieure à 10. Les autres types de données fréquemment manipulés sont les matrices, les listes et les data frames.

```
> v = c(12, .4, 5, 2, 50, 8, 3, 1, 4, .25) # data
> v
 [1] 12.00  0.40  5.00  2.00 50.00  8.00  3.00  1.00  4.00  0.25
> mean(v)
[1] 8.565
> v[2]=7
> v
 [1] 12.00  7.00  5.00  2.00 50.00  8.00  3.00  1.00  4.00  0.25
> sum(v[v>10])
[1] 62
```

Question 1. Cherchez comment calculer le nième percentile d'un vecteur. Calculez le 90e percentile de v.

Toutes ces commandes peuvent être enregistrées dans un fichier qui peut être interprété en spécifiant son chemin avec la commande **setwd** et lancé avec la commande **source**.

3 Graphiques

Question 2. Testez les commandes **boxplot** et **barplot** pour représenter graphiquement les données d'un vecteur. Il est possible de mettre plusieurs vecteurs en paramètre. Dans ce cas, chacun des vecteurs sera représenté.

Plutôt que d'afficher le graphique à l'écran, il est possible de l'enregistrer directement dans un fichier. Ainsi la commande **pdf(file="test.pdf")** permet de rediriger tous les affichages de figures dans le fichier test.pdf. Dans ce cas, il est nécessaire d'utiliser la fonction dev.off() afin de vider le buffer d'impression.

1. http://www.r-project.org

4 Importation de données à partir de fichiers

R offre la possibilité de charger les données à partir de nombreux formats de fichiers (texte, feuilles de calcul, autres logiciels de statistiques). Pour charger les données d'un fichier texte, celui-ci doit respecter un certain format : la première ligne doit indiquer le nom des variables de chaque colonne et les éléments des colonnes doivent être séparés par des caractères d'espacement (espace par défaut).

Dans l'exemple ci-dessous, la fonction **read.table** permet de charger les données du fichier data.txt [2] dans la variable **data**, qui est de type data frame. Ce fichier contient les données de l'expérience 3 décrite dans l'article suivant :

Géry Casiez, Nicolas Roussel, Romuald Vanbelleghem, and Frédéric Giraud. 2011. Surfpad: riding towards targets on a squeeze film effect [3]*. In Proceedings of the 2011 annual conference on Human factors in computing systems (CHI '11). ACM, New York, NY, USA, 2491-2500.*

Le plan expérimental est détaillé dans l'article. Dans ce fichier la première colonne correspond aux numéros des participants, les colonnes qui suivent jusque *density* aux variables indépendantes et les colonnes suivantes aux variables dépendantes. La fonction **names** permet d'afficher les noms des colonnes.

Pour calculer par exemple le temps moyen mis par le participant 2 pour la technique *SurfPad*, il faut dans un premier temps extraire ces données pour ensuite en calculer la moyenne, comme détaillé ci-dessous.

```
data = read.table("data.txt", header=TRUE, sep=",")
participant2SurfPad=subset(data,Participant==2 & Technique=="SurfPad")
mean(participant2SurfPad[,"Time"])
```

Nous souhaitons maintenant représenter graphiquement le temps moyen de réalisation de la tâche pour chaque technique, toutes conditions confondues. Pour cela, il est nécessaire de lister les différentes techniques, en utilisant la fonction **unique** et, pour chacune de ces techniques, calculer le temps moyen. Remarquez la notation avec $ pour accéder aux données d'une colonne.

```
techniques=unique(data$Technique)
```

R permet de définir des fonctions. La syntaxe générale est la suivante :

```
myfunction = function(arg1, arg2, ... ){
  statements
  return(object)
}
```

Question 3. Ecrivez une fonction qui calcule le temps moyen pour une technique donnée. Votre fonction prendra comme paramètres un data frame et une chaîne de caractères représentant la technique.

Question 4. Utilisez la fonction **sapply** pour appliquer la fonction précédemment écrite à un vecteur afin d'obtenir le temps moyen pour chaque technique.

Question 5. Représentez le temps moyen pour chaque technique sous forme de diagramme en barres.

Question 6. Vos calculs précédents incluent les essais pour lesquels il y a eu une erreur (indiqué par un 1 dans la colonne Err). Extrayez les données qui représentent des essais sans erreur. Celles-ci seront utilisées pour les questions suivantes.

Question 7. Sachant que l'intervalle de confiance à 95% peut être estimé par l'équation 1, où SD représente l'écart type et N le nombre d'échantillons, écrivez une fonction qui détermine l'intervalle de confiance à 95% d'une technique.

$$CI = 1.96 \frac{SD}{\sqrt{N}} \tag{1}$$

2. data.txt
3. http://dx.doi.org/10.1145/1978942.1979307

Barplot ne permet pas de représenter des moustaches sur un graphique. Il est pour cela nécessaire d'installer le package **gplots** qui met à disposition la fonction **barplot2**, prévue à cet effet. L'installation se fait de la façon suivante :

```
install.packages("gplots")
library("gplots")
```

Une fois la librairie installée, il faut la charger en utilisant la fonction **library**.

Question 8. Etudiez la documentation de barplot2 pour ajoutez l'affichage des intervalles de confiance à 95% sur le diagramme en barres.

5 Anova

L'analyse de variance des données de l'expérience en utilisant R est détaillée ci-dessous

```
library("reshape")
library("ez")

# Chargement des donnees
data = read.table("data.txt", header=TRUE, sep=",")

# On ne garde que ce qui nous interesse
filteredData = subset(data, (Err==0), select = c(Participant, Block, Technique,
                      A, W, density, Time))

# Aggregation des donnees pour ne conserver qu'une valeur par condition
attach(filteredData)
aggdata = aggregate(filteredData$Time, by=list(Participant,Block,Technique,W, density),
                    FUN=mean)
detach(filteredData)

# Reecriture des noms de colonnes
colnames(aggdata) = c("Participant","Block","Technique","W", "density", "Time")

# Conversion des donnees au format long
data.long = melt(aggdata, id = c("Participant","Block","Technique","W","density","Time"))

# On specifie les variables independantes
data.long$Block = factor(data.long$Block)
data.long$Technique = factor(data.long$Technique)
data.long$W = factor(data.long$W)
data.long$density = factor(data.long$density)

# L'ANOVA:
print(ezANOVA(data.long, dv=.(Time), wid=.(Participant), within=.(Technique,W,density)))

# Analyse post-hoc avec ajustement de Bonferroni
attach(data.long)
print(pairwise.t.test(Time, interaction(Technique), p.adj = "bonf"))
print(pairwise.t.test(Time, interaction(Technique, density), p.adj = "bonf"))
detach(data.long)
```

Question 9. Examinez les effets significatifs. L'effet significatif de *Technique* permet-il de conclure directement que *Surfpad* est meilleure que les deux autres techniques ?

Question 10. Pour comprendre l'interaction entre *Technique* et *Density*, représentez sur un diagramme en barres les techniques pour chaque densité, de façon à obtenir une figure similaire à la figure 8 de l'article. L'interaction entre *Technique* et *Density* existerait-elle encore si on supprimait les données de la technique *SemPoint* ? Vérifiez.