
Mathématiques générales (été)
travaux pratiques 1

La page web du software gratuit R se trouve sur <http://www.r-project.org/>. Vous pouvez l'installer sur votre ordinateur personnel si vous le désirez.

Pour ceux qui sont familiers avec R, vous pouvez aller directement à l'analyse des données M&M's page 2.

Dans les ordinateurs de la salle des TP, l'application R se trouve dans le menu 'Démarrer → Applications CUI'.

Créer sur votre compte un Répertoire (Dossier) pour les TP nommé "Travaux Pratiques". Puis cliquer en haut à gauche sur 'Fichier → Changer le répertoire courant', puis Browser jusqu'à "Travaux Pratiques" ; vous acceptez avec OK. Vous pouvez maintenant créer un fichier avec 'Fichier → Nouveau script', que vous appellerez "tp1.r" au moment de le sauvegarder. Il sera alors sauvé sous "Travaux Pratiques\tp1.r". Vous pouvez taper dans ce fichier des commandes R, les copier les commandes et les coller dans la fenêtre R **Console**. Libre à vous de créer d'autres Directoires et Fichiers dans le futur pour analyser vos propres données et écrire vos propres fonctions. Vous êtes maintenant prêts à travailler avec R.

Vous avez accès en ligne à une aide (documentation) sur les fonctions R existantes. Par exemple, pour la fonction qui construit un histogramme, et quand vous connaissez le nom de la fonction, ici "hist", vous tapez :

```
> help(hist)
```

Si vous ne connaissez pas la fonction, vous tapez :

```
> help.search("histogram")
```

ou vous allez sur Google, et vous tapez "R histogram".

R permet de faire des calculs mathématiques et statistiques, de générer des nombres aléatoires, de faire des graphiques à partir de données. En voici quelques exemples rudimentaires que vous taperez dans la console R :

1. `> sqrt(0.64)` calcule la racine carrée (square root) de 0.64.
2. `> a=c(12,14,6,45.6,4)` crée un vecteur appelé "a" contenant cinq chiffres réels.
3. `> b=c("Rouge","Vert","Blanc")` crée un vecteur "b" de chaînes de caractères.
4. `> sqrt(a)` calcule la racine carrée de chaque élément de "a".
5. `> sqrt(b)` donne un message d'erreur. Pourquoi ?
6. `> length(a)` donne la longueur du vecteur "a". Quelle est la longueur de "b" ?
7. `> a[c(2,3)]` extrait les deuxième et troisième éléments de "a".
8. `> a[c(2 :length(a))]` extrait les éléments de "a" du deuxième au dernier.
9. `> (a>10)` teste si chaque élément de "a" est strictement supérieurs à 10 et retourne TRUE si vrai, ou FALSE sinon.

10. `> a[a>10]` retourne tous les éléments de "a" strictement supérieurs à 10.
11. `> s=c(0,0,1,0,1)` crée un vecteur où 0 = homme et 1 = femme associé au vecteur "a".
12. `> a[s==0]` est le vecteur des éléments de "a" correspondant à des hommes.
13. `> mean(a[s==1])` calcule la moyenne des éléments de "a" correspondant à des femmes.
14. `> d=(1 :12)` crée un vecteur de nombres entiers entre un à douze.
15. En utilisant la fonction `seq` (dont vous apprendrez à connaître les arguments en tapant `> help(seq)`), créer un vecteur d'entiers de 10 à 122 par pas de 2, c'est-à-dire : 2, 4, 6, ..., 122.

On est maintenant prêts à lire les données M&M's :

1. Télécharger le fichier "dataMM.txt" en allant sur le site web du cours
<http://www.unige.ch/math/mgene/>
 Le sauvegarder dans votre Répertoire "Travaux Pratiques". L'ouvrir avec l'éditeur de votre choix et regarder comment les données ont été rentrées : première ligne, nombre de colonnes, nombre de lignes, séparation des colonnes par une tabulation, nombre entier ou avec décimales, etc.
 Voici la description des données : "The data give the color counts and net weight (in grams) for a sample of 30 bags of M&Ms©. The advertised net weight is 47.9 grams."
2. Lire le fichier "dataMM.txt" dans R :
`> X=read.table("dataMM.txt", header=T)`
 L'argument `header=T` est utilisé (T pour TRUE au lieu de F pour FALSE) car la première ligne du fichier contient le nom des variables, à savoir : "Red", "Green", "Blue", "Orange", "Yellow", "Brown", "Weight".
3. Pour pouvoir accéder aux variables en tapant leur nom, taper :
`> attach(X)`
 Ainsi, en tapant
`> Yellow`
 vous obtenez le vecteur du nombre de M&M jaunes comptés dans 30 paquets pris au hasard.

On peut maintenant commencer à obtenir des statistiques. Taper chaque commande suivante (en essayant de comprendre ce que vous tapez) dans le fichier "tp1.r", copier la avec la souris, puis coller la dans la fenêtre de la Console R :

1. `> dim(X)` donne deux chiffres. Le premier chiffre donne le nombre de paquets ; c'est la taille n de l'échantillon. Le deuxième chiffre donne le nombre de variables mesurées, soit $p = 7$: 6 couleurs et un poids.
2. `> Xcolor=X[, -7]` permet d'isoler la matrice des comptages des couleurs en enlevant la septième colonne. Vérifier en affichant "Xcolor".
3. `table(cut(Green, seq(min(Green)-.5, max(Green)+0.5, by=1)))` et `table(cut(Green, seq(min(Green)-.5, max(Green)+0.5, by=1)))/length(Green)` permettent de calculer la table des fréquences pour le nombre de Vert.
4. `> mean(Xcolor[,1])` permet de faire la moyenne du nombre de Rouge (la colonne 1).
5. `> mean(Xcolor[,2])` permet de faire la moyenne du nombre de Vert (la colonne 2), etc.

6. Plus astucieux : `> Xcoloravg=apply(Xcolor,2,mean)` permet de faire la moyenne des nombres contenus dans "Xcolor" colonne par colonne. Ainsi `> Xcoloravg[1]` est le même chiffre obtenu avec `> mean(Xcolor[,1])` ci-dessus.
7. `> pie(Xcoloravg)` représente ces six moyennes dans un graphique en camembert (pie chart). La distribution des couleurs semble-t-elle uniforme ?
8. En faisant la moyenne nous avons perdu des informations.
`> boxplot(Xcolor)` permet de mieux visualiser la distribution de chaque couleur par rapport aux autres. Le trait horizontal noir au milieu de chacune des six boîtes représente la médiane. Ces médianes peuvent être obtenues avec
`> Xcolormed=apply(Xcolor,2,median)`
 Est-il normal que les médianes ne soient pas égales aux moyennes calculées auparavant ?
9. Pour les données relatives aux poids de ces paquets, on peut calculer des mesures de centralité :
 - (a) `> Xweight=X[,7]` extrait les données poids (colonne 7), ou alors utiliser la variable "Weight".
 - (b) `> mean(Xweight)` calcule la moyenne du poids des trente paquets. C'est un estimateur du poids des paquets vendus sur le marché.
 - (c) `> median(Xweight)` calcule la médiane empirique. En utilisant `> sort(Xweight)` qui ordonne les valeurs de "Xweight" par ordre croissant, retrouver la valeur de la médiane (n est pair).
10. On peut calculer des mesures de dispersion :
 - (a) `> sd(Xweight)` calcule l'écart type empirique du poids des paquets. Retrouver le résultat avec la fonction `var`.
 - (b) `> mad(Xweight)` calcule l'écart absolu médian (median absolute deviation).
 - (c) En utilisant la fonction `quantile`, calculer l'étendue interquartile.
11. On peut dessiner un histogramme pour visualiser la distribution d'une variable aléatoire continue à partir d'un échantillon, par exemple :
`> hist(Weight)`
`> lines(rep(47.9,2), c(0,10), col=2, lty=2)` permet de visualiser la valeur annoncée par le fabricant. Le poids du paquet est-il des fois inférieur au poids annoncé ? Estimer la probabilité d'acheter un paquet dont le poids est inférieur au poids annoncé de 47.9g (vous pourrez utiliser un vecteur d'indicateurs `> Weightsmaller=(Weight<47.9)`).

Sauver le fichier "tp1.r" avant de quitter R avec

`> q()`.