Université Grenoble Alpes
L2/STA301

Le but de ces TP est que vous soyez capable de mener une étude statistique sur des données expérimentales, qu'elles soient qualitatives, quantitatives discrètes ou continues. Il est important que vous compreniez les commandes et que vous sachiez les reproduire. Elles ne seront pas données lors des évaluations.

TP1 : Prise en main de R - Description d'une variable qualitative

Objectifs : apprendre les commandes de base du logiciel R et savoir calculer les statistiques descriptives standards pour une variable qualitative.

1 Premiers pas avec R et l'environnement R Studio :

RStudio est un logiciel libre et collaboratif de statistique. Nous utiliserons dans les TP l'interface Rstudio de ce logiciel, que l'on peut télécharger gratuitement depuis le site

`www.rstudio.com`,

en ayant au préalable installé le logiciel R lui même, disponible sur le site

`www.r-project.org`.

Attention, dans RStudio, une majuscule et une minuscule n'ont pas le même sens.

RStudio est séparé en 4 fenêtres graphiques :

- la console (en bas à gauche) : permet d'executer les instructions avec la touche Entrée ↩
- le script (en haut à gauche) : permet d'écrire l'ensemble des commandes (ou instructions) que l'on veut exécuter et de pouvoir les sauvegarder dans un fichier. Une instruction s'exécute dans la fenêtre en bas à gauche en appuyant sur Ctrl+Entrée. Pour ne pas perdre de temps, vous pouvez copier-coller les commandes qui sont données dans les énoncés de TP dans votre script.
- la fenêtre Files/Plots/Packages/Help (en bas à droite) : permet entre autres de visualiser les graphiques ou l'aide.
- la fenêtre Workspace/History (en haut à droite) : permet de voir l'ensemble des objets en mémoire et historique de toutes les instructions réalisées.

Exercice 1

1. Créer un répertoire STA301 dans votre dossier "documents", et un sous-répertoire TPSTA301 dans lequel vous rangerez tous vos documents de Tps : fiches de Tps .pdf, scripts .R, graphiques .pdf, ... Enregistrer l'énoncé du TP1 dans ce répertoire.

   Ce répertoire servira pour toutes les séances de TP. On créera un 'script' au début de chaque séance de TP.

   Nous vous suggérons de créer une arborescence qui vous permettra d'organiser votre travail de façon claire. Par exemple, en créant les répertoires "données" dans lequel vous placerez les fichiers de données, un répertoire "scripts" dans lequel vous sauvegarderez les résultats de vos TPs.

2. Pour se simplifier la vie, nous allons travailler dans R studio dans le répertoire de travail TPSTA301. Pour cela, aller dans l'onglet session de la barre supérieure du menu, choisir Set Working Directory (Répertoire de Travail) et sélectionner le répertoire TPSTA301 nouvellement créé.
3. Dans RStudio, créer un nouveau projet en cliquant sur File/New Project/ puis à partir d'un répertoire existant (Existing Directory) et en sélectionnant le répertoire existant (Existing Directory) TPSTA301.

   Ensuite créer un nouveau script (File/New File/R script) et le sauvegarder dans le répertoire TPSTA301. Vérifier que le script est sauvegardé en fichier ".R".

Exercice 2

Dans R, une fonction s'écrit toujours avec des parenthèses, dans lesquelles les paramètres de la fonction sont précisés. Pour connaitre l'utilité et les paramètres d'une fonction, vous pouvez vous reporter à l'aide (cf. question 7).
Les fonctions de bases sont listées dans le document "Commandes-R-1.pdf" que vous trouverez sur Alfresco. N'hésitez pas à vous y reporter lorsque vous cherchez une fonction. Google est aussi une bonne aide.

1. Dans le script, créer la suite de données $(1, 2, 3, 4, 5)$ avec l'instruction :
   `c(1,2,3,4,5)`
   `c(.)` est la concaténation qui "range" les valeurs ou caractères tapés les uns à la suite des autres dans un vecteur.

   Exécuter la ligne en cliquant sur "Run" ou avec Ctrl+R.
2. Le précédent vecteur n'a pas de nom. Donner lui un nom :

   ```
   x<-c(1,2,3,4,5)
   ```

   La flèche permet d'assigner des valeurs à l'objet x créé. RStudio ne retourne rien. Pour vérifier que le vecteur x contient les valeurs, exécuter l'instruction

   `x`

   Prenez l'habitude de vérifier les objets que vous créez, utilisez.
   A la place de `<-`, on peut aussi utiliser `=`.
3. Créer le vecteur y contenant les valeurs $(2, 4, 6, 8, 10)$.
4. Prenez l'habitude de sauvegarder votre script au fur et à mesure ! (cliquer sur le script et appuyer sur Ctrl+S).
5. Vérifier que les vecteurs x et y ont la même longueur (le même nombre de valeurs)
   `length(x)`
   `length(y)`
6. Tracer sur un graphique les points définis par les deux vecteurs (x, y) :
   `plot(x,y)`
7. Personnaliser votre graphique :

   ```
   plot(x,y, type = "p", pch = 3) # change les symboles
   plot(x,y, type = "b") # ajoute une ligne
   plot(x,y, col = "red") # change la couleur
   ```

   On peut aussi rajouter des titres

   ```
   plot(x,y,main="y selon x", type="p",xlab="abscisse",ylab="ordonnée") # ajoute un titre
   (paramètre main) et des légendes sur chaque axe (parametres xlab et ylab)
   ```

   Créer votre propre graphique en changeant les couleurs, les symboles, les titres.
   Toutes les fonctions sont décrites dans l'aide. Par exemple pour plot, taper :
   `help(plot)`
   ou
   `?plot`

8. Sauver votre graphique comme un fichier pdf en cliquant sur "export"
9. Pour connaitre l'ensemble des objets en mémoire, taper :
   `ls()`
   ou regarder la fenêtre environnement (en haut à droite).

Exercice 3

1. Opérations basiques : comprendre les opérations suivantes
   `x/5`
   `x+5`
   `sum(x)`
   `cumsum(x)`
   `sqrt(x)`
   `x ^ 3`
2. Rajouter des valeurs à la suite du vecteur x
   `c(x,6)`
   Cette commande ne change pas x puisqu'on n'a pas utilisé la flèche ou `x`. Pour changer x, faire :

   ```
   x<-c(x,6)
   x
   ```

   Pour reprendre les valeurs d'origine de x (dont on se sert ensuite)
   `x<-c(1,2,3,4,5)`
   `z<- c(x,1,1,1,1,1)`
   `c(x,rep(1,5))`
   `c(x,seq(from=1, to=10, by=2))`
   `c(x, 6:15)`
   Créer la suite de valeurs $(1, 4, 7, 10, \ldots, 61, 2, 2, 2, \ldots, 2, 1, 2, 3, \ldots, 20)$ où le nombre 2 a été répété 20 fois au milieu de la suite.
3. Dans R, on peut faire des tests logiques sur les éléments d'un vecteur. Comprendre les instructions suivantes :
   `(y>4)`
   `(y!=4)`
   `y==4)`
   `(y>4)&(y<=6)`
4. Dans R, les crochets permettent d'aller chercher des éléments d'un vecteur. Par exemple, pour extraire les deuxième et quatrième valeurs de y :
   `y[c(2,4)]`
   Comprendre les instructions suivantes :
   `y[1:4]`
   `y[(y>4)]`

   Extraire les valeurs de y plus grandes strictement que 2 et plus petites ou égales à 8.
   Extraire les valeurs de z différentes de 1.
   Extraire les valeurs de x égales à 2.
5. Comprendre les opérations de base avec deux vecteurs :
   `x+y`
   `x*y`
   `x/y`

6. Créer une table (ou une matrice) avec les deux vecteurs x et y

   ```
   cbind(x,y) # matrice avec 5 lignes et 2 colonnes (cbind permet de coller des colonnes les unes à la suite des autres)
   rbind(x,y) # matrice avec 2 lignes et 5 colonnes (rbind permet de coller des lignes les unes à la suite des autres)
   ```

7. De la même manière que pour les vecteurs, les crochets permettent d'aller chercher des éléments d'une matrice ou d'un tableau. Il y a alors 2 paramètres à définir, le premier pour les lignes et le second pour les colonnes à sélectionner (les 2 séparés par une virgule). Si rien n'est préciser pour le premier paramètre, cela signifie que l'on prend toutes les lignes. Si rien n'est préciser pour le deuxième paramètre, cela signifie que l'on prend toutes les colonnes.

   ```
   M<-rbind(x,y,y,x) # matrice avec 4 lignes et 5 colonnes
   M # toute la matrice
   M[3,2] # élément de la 3ieme ligne 2ieme colonne
   M[ ,1] # élément de la premiere colonne
   ```

   Extraire les élements de la deuxieme et troisieme lignes.
   Extraire les éléments de la première et troisième colonnes.

2 Statistique pour une variable qualitative

Exercice 4

Dans l'ensemble des centres de transfusion sanguine de la région Rhône-Alpes on a observé sur un échantillon de n patients choisis au hasard parmi ceux ayant donné leur sang en 2013 les deux variables qualitatives : Groupe sanguin et Rhésus. Les effectifs observés sont donnés dans la table de contingence suivante :

Groupe / Facteur	O	A	B	AB
Rhésus +	40	38	6	1
Rhésus -	7	7	1	0

1. Quelles sont les variables d'intérêt ? Quels sont leurs types ?
2. Rentrer les effectifs des Rhésus + dans un vecteur

   ```
   Rp <-c(40,38,6,1)
   ```

   Faire de même avec les effectifs des Rhésus -, stockés dans un vecteur `Rm`.
3. Créer une table `S` rassemblant les deux vecteurs. Ajouter le nom des groupes sanguins en faisant

   ```
   colnames(S)<- c("O", "A", "B", "AB)
   ```

4. Quelle est la taille de la population ?
5. Quels sont les effectifs de chaque Rhesus ? Quelles sont les fréquences des deux Rhesus ? Quelle est la proportion de donneurs ayant un Rhesus négatif ?