Université Sorbonne Paris Nord, Institut Galilée
Année universitaire 2020–2021
mallein@math.univ-paris13.fr

TP 2 – Les packages, les graphiques

1 Packages disponible

Le language R étant le logiciel d'élection d'une très large communauté de chercheur en Statistique, de très nombreuses méthodes d'analyse de données sont implémentées en R, et de nouvelles ressources sont développées très régulièrement. Ces ressources ne sont pas ajouté au logiciel R lui-même mais restent disponible sous la forme de modules supplémentaires appelées packages. Il y a aujourd'hui plus de 16600 packages disponibles, permettant de réaliser rapidement les analyses les plus variées.

Il n'est bien sûr pas une bonne idée de télécharger l'ensemble de ces packages, mais plutôt d'ajouter un nouveau package correspondant à un besoin particulier au moment où ce besoin se fait sentir. Pour avoir accès à une liste de packages validés, on peut se rendre sur la page du CRAN (Comprehensive R Archive Network). En particulier, la page Task Views recense des packages en liens avec certaines tâches d'intérêt.

D'autres packages non-officiels peuvent également être trouvés sur GitHub. Il faut bien entendu faire attention à ne pas installer n'importe quoi, mais la plupart de ces packages sont testés par une large communauté, et peuvent être vérifiés "à la main" avant installation.

Exercice 1 (Installation d'un package R)**.** On prendra ici l'exemple du package `pacman` (pour *package manager*), disponible sur le site du CRAN.

1. Utiliser l'instruction `install.packages(pacman)`. Cette instruction installe le paquet pacman sur votre ordinateur.
2. Utiliser maintenant l'instruction `require(pacman)` ou `library(pacman)` pour rendre les fonctions du package disponible.

Une fois cette instruction exécutée, il devient possible d'utiliser pacman pour interagir de façon plus simple avec les packages. La fonction `p_load(mon_package)` permet de rentre le package `mon_package` disponible, en l'installant préalablement si nécessaire. La fonction `p_unload(mon_package)` permet de désactiver le package dans la session actuelle (le package reste installé). On peut utiliser `p_unload(all)` en fin de session pour désactiver tous les packages utilisés.

Attention, installer un paquet est une opération qui prend du temps. Vous pourrez tester *après le TP* la commande `p_load(shiny)` pour tester l'installation et la mise à disposition du package shiny, permettant de créer des interfaces graphiques lors de la manipulation de données.

2 Les graphiques

Le logiciel R est utilisé pour l'analyse et la visualisation des données. Il possède donc de très nombreuses fonctions graphiques, chacune possédant de très nombreuses options dont il serait fastidieux de faire la liste. On se contentera donc ici de donner quelques exemples d'utilisation des fonctions de base. On notera toutefois que le package `ggplot2` permet de réaliser des graphes encore plus esthétiques. De très nombreuses ressources sont disponibles en ligne pour explorer plus en avant les modules graphiques de R.

Exercice 2 (La fonction `plot`)**.** La fonction `plot` est une fonction de base de R. Elle est construite de façon à être extrêmement intuitive à l'utilisation, et permet de tracer de la même façon des séries de données, des relations entre données quantitatives, des relations avec des données qualitatives, des fonctions numériques. On illustrera ses propriétés sur les data frames `iris` et `lynx`.

1. Que contient `lynx` (on appelle ce type d'objets un `ts`, pour *time series*) ? Que renvoie `plot(lynx)` ? Et `plot(lynx,lw=2)` ? Et `plot(lynx,type="p")` ?
2. Que contient `iris` ? On pourra essayer la fonction `View`. Que renvoie `plot(iris$Sepal.Length)` ? Et `plot(iris$Sepal.Length,iris$Petal.Length)` ? Et `plot(iris)` ?
3. On testera ensuite les commandes `plot(iris$Species)`, `plot(iris$Species, iris$Petal.Length)` ainsi que `plot(iris$Petal.Length,iris$Species)`.

4. On définie la fonction `absolue` renvoyant la valeur absolue d'un nombre passé en entrée. Que renvoie `plot(absolue)` ? Et `plot(absolue,-5,5)` ?
5. On définie une fonction `h` renvoyant 1 si $x < 1$, $2x - 1$ si $1 \leq x \leq 2$ et $3x^2/4$ si $x \geq 2$ avec une structure `if ... else ...`. Que renvoie `plot(h)` ? Reprendre la fonction en utilisant des opérations booléennes, qu'observez-vous ?

N'hésitez pas à consulter `help(plot)`, qui vous donne la documentation essentielle sur cette fonction. N'hésitez pas non plus à consulter les nombreuses aides en ligne proposant de très nombreux exemples d'utilisation.

Exercice 3 (La fonction `boxplot`)**.** La fonction `boxplot` permet de dessiner des boîtes à moustache de façon simple. Testez les commandes suivantes.

1. `boxplot(iris$Sepal.Width)`, `boxplot(iris)`, `boxplot(iris$Petal.Length~iris$Species)`.
2. `boxplot(lynx, horizontal=TRUE)`

De la même façon, n'hésitez pas à consulter l'aide sur `boxplot` pour tester d'autres commandes.

Exercice 4 (La fonction `hist`)**.** La fonction `hist` permet de dessiner des histogrammes. On l'utilise pour visualiser la distribution d'une quantité.

1. Dessiner l'histogramme de la longueur des pétales d'iris du data.frame `iris`.
2. Faites en sorte que cet histogramme ait 25 classes.
3. Dessiner l'histogramme de la longueur des pétales d'iris de genre `setosa`. Même question pour les deux autres genres d'iris.
4. Modifier la couleur et le remplissage des barres des histogrammes déjà tracés.

On rappellera que `help(hist)` permet d'avoir de nombreuses informations sur les options de l'histogramme.

Exercice 5 (La fonction `curve`)**.** La fonction `curve` permet de tracer une fonction numérique de la même façon que `plot`. Elle permet toutefois de tracer une fonction de plusieurs variables en précisant la valeur de `x` le paramètre d'intérêt.

1. Tester les instructions `curve(exp)`, `curve(exp,0,5)`, `curve(exp,0,100)`.
2. On définit la fonction $u : (x, y) \mapsto (x - y)/(x^2 + y^2)$. Quel est l'effet de l'instruction `curve(u(x,1),0,2)` ? Et `curve(u(2,x),-1,1)` ?
3. Utiliser la fonction `dnorm` pour tracer la densité de la gaussienne de moyenne 3 et variance 2 sur l'intervalle $[-10, 10]$. Faites en sorte que la courbe soit rouge.

N'oubliez pas l'existence de la documentation sur la fonction `curve`.

Exercice 6 (Organisation de graphes)**.** On met ici en avant quelques commandes permettant de construire des figures plus complexes, permettant entre autre de tracer plusieurs courbes sur le même graphe, et plusieurs graphes sur la même figure.

1. Utiliser la commande `plot(iris$Petal.Width, iris$Petal.Length, main="Analyse des iris", sub="Taille des pétales", xlab="Largeur des pétales", ylab="Longueur des pétales")`. Reprendre les précédents graphes et ajouter des titres et labels personnalisés. Utiliser le bouton export pour enregistrer un graphe.
2. Pour ajouter des tracers à une figure, on utilise l'instruction `add=TRUE`.
   (a) Tracer l'histogramme des longueurs de sépales du data.frame `iris`, en utilisant l'option `prob=TRUE` permettant de tracer la densité plutôt que la fréquence.
   (b) Utiliser `curve(..., add=TRUE)` pour surimposer sur la figure le graphe de la loi normale de moyenne et variance égales à celles des longueurs de sépales, multiplié par le nombre de données considérées.
   (c) Utiliser la commande `title("Mon Super Plot")` pour changer le titre.
3. Il est également possible de tracer plusieurs graphes sur une seule figure. Pour ce faire on peut utiliser l'instruction `layout`, l'instruction `mfrow`, etc. Attention, ces instructions ne fonctionnent pas ensemble, on fera attention de choisir une méthode ou l'autre.
   (a) Exécuter l'instruction `def.par = par(no.readonly=TRUE)`, permettant de sauvegarder les paramètres par défaut. À n'import quel moment, l'instruction `par(def.par)` permet de revenir aux paramètres par défaut en cas de problème.

(b) Définissez la matrice $A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 2 \\ 3 & 0 & 2 \end{bmatrix}$. Utiliser la commande `layout(A)`. Comment les figures sont-elles maintenant positionnées ?

(c) Après être revenus aux paramètres par défaut, exécutez l'instruction `par(mfrow = c(3,1))`. Utiliser ces nouveaux paramètres pour tracer les histogrammes des longueurs de pétales d'iris des trois genres sur trois figures superposées, avec la même échelle. On pourra utiliser les paramètres `xlim= c(a,b)` et `ylim=c(c,d)` pour fixer les échelles.

La fonction `par` est utilisée pour changer de façon "permanente" (pour la session) la façon de dessiner les figure (couleur, taille, fonte des légendes, nombre et disposition des figures, etc.). Consulter la documentation pour plus d'informations.

3 Génération de variables aléatoires

Le langage de programmation R permet bien entendu de simuler de nombreuses variables aléatoires. On verra ici certaines de ces méthodes pour tester les différentes méthodes d'analyse et de visualisation des données.

Exercice 7 (Loi normale)**.**

1. Utiliser la fonction `rnorm` pour générer un vecteur de 1000 variables aléatoires de loi normale centrée réduite.
2. Tracer sur le même graphe l'histogramme de densité de la réalisation de 10000 variables de loi normale de moyenne 5 et variance 3, et la densité de la loi normale de moyenne 5 et variance 3.
3. On pose `mu` et `sigma` deux nombres en début d'un script.
   (a) Rappeler l'estimateur non-biaisé de la moyenne et la variance n-échantillon gaussien.
   (b) Construire une matrice 100 par 1000 de variables gaussiennes de moyenne `mu` et de variance `sigma`^{2}.
   (c) Construire 100 estimateurs de la moyenne et de la variance des 1000-échantillons composés des différentes lignes. Tracer les histogrammes des résultats obtenus. Qu'observe-t-on ?
   (d) Tracer le nuage de point correspondant aux couples (estimateur de la moyenne, estimateur de la variance). Observe-t-on une tendance ?
   (e) Calculer la corrélation entre le vecteur d'estimateurs de la moyenne et celui d'estimateurs de la variance. Que remarquez-vous ?
4. Tracer un nuage de points représentant 1000 réalisations d'un vecteur Gaussien de moyenne $10,5$ et de matrice de covariance $A = \begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix}$.
5. En traçant l'histogramme de 1000 réalisations de la variable X/Y, où X et Y sont deux lois normales centrées réduites indépendantes, ainsi que la fonction de densité de la loi de Cauchy, vérifier que $X/Y \sim \text{Cauchy}(0,1)$.

Exercice 8 (Loi binomiale)**.** 1. Utiliser la fonction `pbinom` pour tracer la fonction de répartition de la loi binomiale de paramètres 10 et $0,3$.

2. Réalisez l'histogramme de 1000 réalisations de la loi binomiale de paramètres 500 et $0,4$. En traçant sur le même graphe la fonction de densité d'une loi normale bien choisie, vérifiez le théorème central limite.
3. Réalisez l'histogramme de 1000 réalisations de la loi binomiale de paramètres 500 et $0,01$. En traçant sur le même graphe la fonction de densité d'une loi de Poisson (`dpois`) bien choisie, vérifier le théorème d'approximation binomiale-Poisson.