Exercices de la session 1

Vincent Guillemot

Manipulation des vecteurs, facteurs et matrices

Exercice 1 : Vecteurs

Soit k un entier entre 1 et 100. Déterminez les valeurs de k pour lesquelles $\sin(k) > 0$.

Indices : `which`, `:`.

Exercice 2 : Facteurs

Définissez un facteur `fac <- factor(c("a","b","b","b","a","b","a","a"))`. Calculez le nombre de `"a"` et de `"b"` dans `fac` utilisant la fonction `length` et des opérateurs binaires.

Que permet de faire la fonction `table` ? Appliquez la à `fac`.

Indices : `length`, `which`, `==`

Exercice 3 : Combiner des vecteurs

1. Créez un vecteur `a` contenant tous les entiers de 1 à 100.
2. Créez un vecteur `b` contenant tous les entiers pairs de 2 à 100.
3. Uniquement en utilisant `a` et `b`, créez un vecteur contenant les entiers **impairs** entre 1 et 100.

Indice: `seq`.

Exercice 4 : Matrices

1. Exécuter la commande `a <- rep(0:1, 50)`. Qu'a-t-on fait ?
2. Utilisez `a` pour construire une matrice `A` à 10 lignes et 10 colonnes.
3. Utilisez la fonction `t` sur cette matrice pour créer une matrice `B`. Que s'est-il passé ?
4. Que se passe-t-il après l'opération `M <- A+B` ?
5. Les commandes `A[1:5, ]` et `B[, 1:5]` permettent de récuperer respectivement les 5 premières lignes de `A` et les 5 dernières colonnes de `B`. Inspirez-vous de ces commandes pour récupérez les lignes de 1 de `A` et les colonnes de 0 de `B`.
6. Extrayez les 2 de la matrice `M`.

Listes et tableaux de données

Exercice 5 : `list` et `data.frame`

1. Créez une liste `x` contenant une variable aléatoire gaussienne de taille 10 appelée `a` et un vecteur contenant uniquement des 1 de taille 10 également. On peut accéder aux deux éléments de cette liste avec les commandes `x[[i]]` ou `x$nom_de_la_variable`. *Indice : `rnorm`.*

2. Créez un objet `y` qui est la transformation de cette liste en `data.frame`. On peut maintenant parcourir les éléments de chaque objet comme pour une matrice avec la commande `y[i,j]` !
3. Créez deux objets `z1` et `z2` contenant respectivement les 3 premières et les 3 dernières lignes de `y`. Quelle est la classe de ces deux objets ?
4. Rajoutez à la liste `x` un vecteur `alphabet` contenant les lettres de l'alphabet.
5. Essayez de transformer de nouveau `x` en `data.frame`. Que se passe-t-il ?

Fonctions

Exercice 6 : Création de fonction

1. Exécutez les commandes `data(iris)` puis `str(iris)`. Nous venons de charger en mémoire l'un des nombreux jeux de données distribués avec R ! Profitez de l'aide sur ce jeu de données pour en apprendre un peu plus sur les fleurs (`?iris`) !
2. Créez la fonction `f` suivante et décryptez la :

```
f <- function(i) c(moy = mean(iris[,i]), et = sd(iris[,i]) )
```

Remarque : pour exécuter plusieurs commandes au sein d'une même fonction, il faut utiliser des accolades `{...}`. Par exemple

```
f <- function(i) {
  moy <- mean(iris[,i])
  et <- sd(iris[,i])
  return( c(moy = moy, et = et) )
}
```

Exercice 7 : Fonction de Gauss

La fonction de Gauss, caractérisée par sa forme *en cloche*, est une fonction très utilisée en Statistique. Son expression est la suivante

$$\begin{array}{rccl} g & : & \mathbb{R} & \to & \mathbb{R}^+ \\ & & x & \mapsto & g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \end{array}$$

Elle dépend de deux paramètres : μ et σ qui caractérisent respectivement sa *position* et son *étendue*. De plus, le facteur multiplicatif $1/\sigma\sqrt{2\pi}$ garantit que l'intégrale de g sur $\mathbb{R}$ soit égale à 1. La fonction g est utilisée comme densité de probabilité d'une variable aléatoire gaussienne de moyenne μ et d'écart-type σ.

1. Attribuez des valeurs aux paramètres μ (`mu`) et σ (`sigma`) : par exemple `mu <- 1` et `sigma <- 2`.
2. Ces paramètres étant fixés, calculez $g(1)$, $g(0)$ et $g(2)$.
3. Créez une fonction `g` (`g <- function(x, mu, sigma) {...}`) qui permet de calculer les valeurs de g.
4. Calculez son intégrale sur l'intervalle de votre choix en utilisant la fonction `integrate`.
5. La taille moyenne des hommes en France est de 1m72 avec un écart-type de 7.2cm[1]. Proposez une méthode permettant de calculer la probabilité qu'un homme mesure plus d'1m90.

[1] Ces données sont inspirées d'une étude faite en France dans les années 70.

Exercice 8 : La procédure `ifelse`

La fonction `ifelse` permet de transformer des vecteurs booléens (c'est à dire des vecteurs contenant uniquement des valeurs `TRUE` ou `FALSE`) en tout autre valeur : une valeur sera attribuée à `TRUE` et une autre valeur à `FALSE`.

Par exemple, `ifelse(c(TRUE, FALSE, TRUE, TRUE, FALSE), "C'est vrai !", "C'est faux !")`.

- A l'aide la fonction `ifelse`, générez un facteur qui contient 100 valeurs réparties en deux "niveaux". Les deux niveaux sont `"positif"` et `"négatif"` : `"positif"` quand $sin[k] > 0$ et `"négatif"` quand $sin[k] < 0$ ($k = 1, ..., 100$).

Exercice 9 : Les boucles `for`

1. Lisez l'aide sur la procédure permettant de réaliser des boucles indicées `for` (`help("for")`). *Remarque :* demander de l'aide sur cette procédure avec la syntaxe `?for` ne fonctionnera pas ! Pourquoi ?
2. Pour exemple, exécutez la boucle suivante `for (i in 1:10) print(i)`.
3. A l'aide d'une boucle, calculez la somme des entiers pairs compris entre 1 et 100.

Indices : `for`, `ifelse`, `:`, `%%`

Exercice 10 : `for`, `if` et `else`

1. Comme dans l'exercice précédent, lisez l'aide de la procédure conditionnelle `if` : (`help("if")`).
2. Utilisez les structures `if` et `else` pour créer un programme qui prend en entrée un réel x et qui lui associe $y = x^2$ si x est positif et $y = x^3$ si x est négatif.
3. Utilisez les structures `for`, `if` et `else` pour créer un programme qui imprime à l'écran, pour chaque entier relatif i compris entre -10 et 10, i^3 si $i <= 0$, ou i^2 si $i > 0$.

Exercice : Manipulation de données aléatoires

1. Créez un vecteur `x` de taille 100 qui est un échantillon $\mathcal{N}(0, 1)$ (de loi Gaussienne, moyenne nulle et écart-type unitaire). *Indice : `rnorm`.*
2. Trouvez toutes les valeurs de `x` qui sont supérieures à son 3ème quartile (quantile à 75 %), noté `q75`. *Indice : `?quantile`.*
3. Comptez-les en suivant les étapes suivantes :
   - calculez `sum(c(TRUE, FALSE))` ;
   - que remarquez-vous ?
   - utilisez `x > q75` et la fonction `sum`.
4. Répétez les étapes 1 à 3 pour un échantillon de loi uniforme de bornes -30 et 30.

Lire et sauvegarder des données

Exercice 11 : Lire les données d'un fichier

Il est possible de lire les données stockées dans des fichiers sous format `txt` grâce, entre autres, aux fonctions suivantes: `read.table()`, `read.csv()`, `read.csv2()` et `scan()`. Par ailleurs, la fonction `read.xls()` (resp. `write.xls()`) du package `gdata` fournit les outils pour lire (resp. écrire) des fichiers au format Excel.

Lorsque les données ont été sauvegardées sous le format propriétaire d'un logiciel statistique tiers, il est nécessaire de disposer d'outils permettant leur transfert vers le système. Par exemple les library `foreign` ou `gdata` offre ces outils pour une sélection des logiciels statistiques les plus courants (e.g. SAS, SPSS...). Par exemple, la fonction `read.spss()` prend en charge les données SPSS. La fonction `read.ssd()` prend en charge les tables permanentes SAS. Dans ce TP, nous nous limitons à la lecture des fichiers `.txt`, `.csv` et `.xls`.

Exercice 12 : Utilisation de la fonction `read.table()`

1. Importer dans une variable nommée `A` le jeu de données nommé `auto2004_original.txt`.
2. Importer dans une variable nommée `B` le jeu de données `auto2004_sans_nom.txt`.
3. Importer dans une variable nommée `C` le jeu de données `auto2004_virgule.txt`.
4. Importer dans une variable nommée `D` le jeu de données `auto_don_manquante.txt`.
5. Importer dans une variable nommée `E` le jeu de données `auto_don_manquante(99999).txt`.
6. Quel est le mode des objets créés par la fonction `read.table()` ?

Exercice 13 : Utilisation de la fonction `read.xls()`

Importer dans une variable nommée `F` le jeu de données `bordeaux.xls`.

Exercice 14 : Enregistrer des données

Créer la matrice suivante :

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

1. Sauver `A` sous le nom de `matrice.txt` à une adresse valide. Que remarquez vous?
2. Ajouter des arguments à la commande précédente pour retirer des noms aux lignes et aux colonnes du fichier créé.
3. Sauvegarder la data.frame `A` au format `.txt` sous le nom `auto.txt`.
4. Sauver les objets présents en mémoire à l'adresse `chemin/Donnees.Rdata`.
5. Sauver les objets disponibles en mémoire vive à l'adresse `chemin/` grâce à la fonction `setwd()`.
6. Sauver l'ensemble des données au format `.Rdata` grâce à la fonction `save`.

Exercice 15 : Test d'égalité de deux moyennes

1. Importer dans une variable nommée `I` le jeu de données nommé `faillite.csv`.
2. Pour chacune des variables explicatives, on désire savoir s'il existe une différence significative entre les moyennes observées dans chacun des groupes (faillite/non-faillite). Réaliser le test approprié.
3. Stocker les 4 p-valeurs dans un vecteur.

La corrélation: rappels et calcul *à la main*

Exercice 16 : Définition du coefficient de corrélation

Dans cet exercice, nous allons nous intéresser à la simulation d'un jeu de données aléatoire et au calcul des corrélations entre les variables constituant ce jeu de données.

Par définition, un coefficient de corrélation se calcule entre deux variables aléatoires X et Y :

$$\rho_{XY} = \frac{E((X - \bar{X})(Y - \bar{Y}))}{\sqrt{V(X)V(Y)}},$$

où $\bar{X}$ et $\bar{Y}$ représentent les espérances de X et Y, $V(X)$ et $V(Y)$ représentent leur variance.

L'estimateur classique du coefficient de corrélation sur un échantillon (x_i, y_i) de taille n est le suivant :

$$cor(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Exercice 17 : Lien entre coefficient de corrélation et coefficient dans un modèle linéaire

Dans la formalisation classique du modèle linéaire entre deux variables, on suppose que X est une variable aléatoire, et que Y est la combinaison linéaire de X et d'un *bruit* ϵ :

$$Y = a + bX + \epsilon,$$

où a représente l'intercept, et b le coefficient de la droite de régression. Le bruit ϵ est le plus souvent supposé gaussien, de moyenne nulle, et d'écart-type σ.

Ce modèle étant posé, on peut montrer que

$$b = \rho_{XY} \sqrt{\frac{V(Y)}{V(X)}}.$$

Exercice 18 : Modèle linéaire : lire un résultat

Le poids du cerveau et le poids total du corps ont été mesurés chez 62 mammifères (Allison and Cicchetti 1976). Un modèle linéaire sans intercept (Weisberg 1980) a été appliqué sur ces données. Le résultat, après application de la commande appropriée dans R, est le suivant[2] :

```
Call:
lm(formula = y ~ x + 0, data = tab2)

Residuals:
    Min      1Q  Median      3Q     Max
-219.88    0.45    4.39   78.52 1234.28

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
```

[2] Pour les besoins de l'exercice, nous avons aléatoirement sélectionné 10 individus, cela permet d'avoir une p-valeur "intéressante" à calculer.

```
x   1.3825     0.8157   1.695    0.124

Residual standard error: 421.2 on 9 degrees of freedom
Multiple R-squared:  0.242, Adjusted R-squared:  0.1577
F-statistic: 2.873 on 1 and 9 DF,  p-value: 0.1243
```

Le but de l'exercice n'est pas pour l'instant d'apprendre à calculer un modèle linéaire en R, mais de se concentrer sur les lignes qui contiennent le coefficient de la droite de régression linéaire ainsi que sur la ligne contenant le nombre de degrés de liberté, est de comprendre les liens qui existent entre ces différents chiffres.

```
Coefficients:
  Estimate Std. Error t value Pr(>|t|)
x   1.3825     0.8157   1.695    0.124

  ...

Residual standard error: 421.2 on 9 degrees of freedom
```

On voit la valeur du coefficient : proche de 1.4, ainsi que son erreur type (*standard error*), proche de 0.8.

> Par définition, une valeur aléatoire de Student à n degrés de liberté est le quotient de deux variables aléatoires indépendantes : une variable aléatoire normale et une variable aléatoire du χ^2 à n degrés de liberté.

1. Divisez la valeur du coefficient par son erreur type. Que remarquez vous ?
2. La valeur de t ainsi calculée permet d'obtenir la p-valeur (notée ici `Pr(>|t|)`). Re-calculez la p-valeur en vous servant des données *(Indice: `?pt`)*.
3. Quelles sont les hypothèses qui ont été faites pour calculer cette p-valeur ?

Exercices de niveau avancé

Exercice 19 : Propriétés des estimateurs de la moyenne et de la variance (Husson and Pagès 2013)

Dans une population de taille $N = 5$, une variable X peut prendre uniquement les valeurs suivantes de façon équiprobable

$$-6, -3, 0, 3, 6$$

De façon générale, pour une variable aléatoire X pouvant prendre des valeurs X_i avec une probabilité p_i, les moyenne et variance théoriques se calculent de la façon suivante

$$\mu = \sum_i p_i X_i \text{ et } \sigma^2 = \sum_i p_i (X_i - \mu)^2.$$

1. Calculez la moyenne μ et la variance σ^2 (théoriques) de X dans cette population.
2. On effectue des prélèvements d'échantillons de taille $n = 2$ sans remise dans cette population. Enumérez tous les échantillons possibles. Pour chacun d'entre eux, calculez leur moyenne empirique et leur variance empirique. *Indices : utilisez la fonction `expand.grid` ainsi qu'une boucle `for`. Attention, vous aurez très probablement besoin de convertir le résultat de `expand.grid` en `matrix` !*
3. Vérifiez que la moyenne empirique est un estimateur sans biais de la moyenne μ. (Même chose pour la variance)

Exercice 20 : Système d'équations

Il y a une grande colline entre les deux villages A et B. Le train prends 2h pour aller de A à B, et 2h30 min de B à A. La vitesse du train est de 30 km/h quand il monte et de 40 km/h quand il descend. Calculez la distance entre A et le sommet de la colline S ainsi que la distance entre B et S.

Indice : On peut résoudre le système à deux équations et deux inconnues suivant

$$\begin{cases} x + y = 1 \\ x + 2y = 2 \end{cases}$$

à l'aide de la commande `solve(matrix(c(1,1,1,2),2,2), c(1, 2))`.

Exercice 21 : Equations du 2nd degré

Créez une fonction qui accepte trois arguments numériques `a`, `b` et `c` et qui permet de calculer les solutions de l'équation quadratique

$$ax^2 + bx + c = 0.$$

Utilisez-la pour résoudre les trois équations suivantes :

$$x^2 + 2x + 1 = 0$$
$$x^2 + 3x + 2 = 0$$
$$-x^2 + x - 2 = 0$$

Exercice 22 : Approximation de l'exponentielle

La fonction exponentielle peut s'approximer à l'aide de séries entières. Mathématiquement :

$$\lim_{n \to +\infty} \sum_{k=0}^{n} \frac{x^k}{k!} = e^x,$$

où $k! = 1 \times 2 \times ... \times k$ est la fonction *factorielle*.

Nous allons montrer graphiquement la qualité de cette approximation avec les étaês suivantes.

1. Initialisez deux vecteurs `a` et `b` à la valeur 1.
2. Initialisez un réel `x` à la valeur 0.2.
3. Initialisez un entier `n` à la valeur 10.
4. A l'aide d'une boucle `for`, faites en sorte que `a` et `b` soient tels que
$$a[1] = b[1] = 1 \text{ puis } a[k] = x^{k-1} \text{ et } b[k] = (k-1)! \text{ pour } k = 2, ..., n.$$
(Indice : utilisez la fonction `factorial`)
5. Divisez `a` par `b` et faites en la somme ! Vous obtenez ainsi une approximation à 10 termes de $e^{0.2}$.
6. Répétez l'opération pour obtenir toutes les approximations de $e^{0.2}$ à 2, 3, ..., 8 et 9 termes.
7. Faites un graphe contenant ces valeurs et observez la vitesse de convergence !

Références

Allison, T, and D. Cicchetti. 1976. "Sleep in Mammals: Ecological and Constitutional Correlates." *Science* 194 (4266): 732–34.

Husson, François, and Jérôme Pagès. 2013. *Statistiques Générales Pour Utilisateurs: Exercices et Corrigés*. Presses universitaires de Rennes.

Weisberg, Sanford. 1980. *Applied Linear Regression*. Wiley.