Université d'Evry Val d'Essonne **11-13 janvier 2017**
Introduction au logiciel R

Vecteurs

Exercice 1

Commandes `c()`, `seq()`, `rep()`, `paste()` et leurs options

1. Créer un vecteur contenant la suite des entiers de 1 à 12 de deux manières différentes.
2. Créer le vecteur `c(0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0)` de trois manières différentes.
3. Créer un vecteur contenant tous les multiples de 2 compris entre 1 et 50.
4. Créer un vecteur contenant 3 fois chacun des 10 chiffres.
5. Créer un vecteur contenant une fois la lettre A, deux fois la lettre B, etc., 26 fois la lettre Z. Quelle est la longueur de cette suite ? (Utiliser la chaîne `LETTERS` prédéfinie).
6. Créer le vecteur `c("individu 1", "individu 2", ..., "individu 100")`.

Exercice 2

Commandes `sample`, `length`, `sort`, `rev`, `sum`, `table`, etc.

1. Générer une séquence d'ADN de n bases. Compter le nombre d'occurrences de chaque lettre (d'abord sans puis avec la fonction `table`). Renvoyer les indices de la séquence où l'on trouve la lettre `"t"`.

2. Créer un vecteur contenant les 100 premiers entiers échantillonnés aléatoirement. Renvoyer l'emplacement de la valeur minimale et de la valeur maximale. À partir de ce vecteur, créer les vecteurs `x` et `y` des 100 premiers entiers ordonnés dans l'ordre croissant et décroissant. Concatenez `x` et `y`, enlever le seul nombre apparaissant deux fois de suite en le repérant à l'aide de la commande `diff`.

Exercice 3

On mesure le taux d'insuline de deux groupes d'individus. Le premier groupe comprend des individus atteints de diabète de type 1 et le deuxième groupe des individus normaux. On observe les valeurs suivantes :

```
grp1 <- c(14.40 , 13.70 , 14.20 , 17.30 , 13.90 , 13.60 , 15.40 , 10.80 , 12.20 , 13.60)
grp2 <- c(14.00 , 15.90 , 16.90 , 14.10 , 13.80 , 20.30 , 16.00 , 15.30 , 16.10 , 15.90)
```

1. En utilisant `R`, calculez la moyenne, la médiane, la variance et l'écart type pour chaque groupe.
2. Représentez les données sous forme de boîtes à moustaches.

Facteurs

Exercice 4

On s'intéresse au rendement de champs d'orge traité à différente dose d'engrais et appartenant à différentes variété :

```
    variete <- c("victory", "victory", "victory", "victory", "Golden.rain", "Golden.rain", "Golden.rain", "Golden.rain",
"Marvellous", "Marvellous", "Marvellous", "Marvellous", "victory", "victory", "victory", "victory", "Golden.rain",
"Golden.rain", "Golden.rain", "Golden.rain", "Marvellous", "Marvellous", "Marvellous", "Marvellous", "victory",
"victory", "victory", "victory", "Golden.rain", "Golden.rain", "Golden.rain", "Golden.rain", "Marvellous", "Marvellous",
"Marvellous", "Marvellous", "victory", "victory", "victory", "victory", "Golden.rain", "Golden.rain", "Golden.rain",
"Golden.rain", "Marvellous", "Marvellous", "Marvellous", "Marvellous", "victory", "victory", "victory", "victory",
"Golden.rain", "Golden.rain", "Golden.rain", "Golden.rain", "Marvellous", "Marvellous", "Marvellous", "Marvellous",
"victory", "victory", "victory", "victory", "Golden.rain", "Golden.rain", "Golden.rain", "Golden.rain", "Marvellous",
"Marvellous", "Marvellous", "Marvellous")
engrais <- c("0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt",
"0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt",
"0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt",
"0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt",
"0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt",
"0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt",
"0.2cwt", "0.4cwt", "0.6cwt", "0.0cwt", "0.2cwt", "0.4cwt", "0.6cwt")
rendement <- c(111, 130, 157, 174, 117, 114, 161, 141, 105, 140, 118, 156, 61, 91, 97, 100, 70, 108, 126, 149,
96, 124, 121, 144, 68, 64, 112, 86, 60, 102, 89, 96, 89, 129, 132, 124, 74, 89, 81, 122, 64, 103, 132, 133,
70, 89, 104, 117, 62, 90, 100, 116, 80, 82, 94, 126, 63, 70, 109, 99, 53, 74, 118, 113, 89, 82, 86, 104, 97,
99, 119, 121)
```

1. Tracer la répartition empirique des rendements à l'aide de la commande `boxplot`, en découpant par variété, par dose d'engrais reçu puis par couple variété/dose.
2. Calculer la moyenne par variété, par dose d'engrais reçu puis par couple variété/dose. Toujours selon ces mêmes découpages, faites un résumé numérique.
3. (a) Combien y a-t-il de champs au total ? de champ de chaque variété ? Par dose d'engrais ? Par couple (variété,engrais) ?
   (b) Même question en ne conservant que les champs dont le rendement est supérieur au rendement moyen par groupe.
   (c) Même question en ne conservant que les champs dont le rendement est supérieur au rendement moyen total.
   (d) Quelle est la meilleure combinaison (engrais,variété) en terme de rendement ? La moins bonne ?

Matrices, listes , tableaux de données

Exercice 5

1. Charger les valeurs numériques des données iris à l'aide de la commande

   ```
   data(iris)
   ```

2. Donner la dimension de la matrice ainsi construite. Trouver la plus grande valeur observée. Donner le numéro de ligne et de colonne correspondant.
3. Calculer la moyenne en ligne et en colonne, d'abord avec les commandes `rowSums`, `colSums` et `nrow`, `ncol`, puis à l'aide de la commande `apply`. Quel individu à la plus grande longueur de Sépale ? Largeur de Pétale
4. Représenter le graphe des paires de variables à l'aide de la commande `pairs`.

Exercice 6

1. Charger les valeurs numériques de données d'expression de gènes pour différents types de cancers à l'aide de la commande :

   ```
   microarray <-
   as.matrix(read.table("http://statweb.stanford.edu/ tibs/ElemStatLearn/datasets/14cancer.xtrain"))
   ```

2. Calculer la covariance entre les échantillons. Représenter le résultat sous forme d'image. Transformer la covariance en corrélation et représenter à nouveau cette image. Représenter ensuite le résultat de la fonction `heatmap`.

Exercice 7

On utilise un programme permettant de calculer le nombre d'occurences des 4 nucléotides "A", "T", "G" et "C" dans une séquence d'ADN. Celui-ci renvoie une liste comportant 4 éléments, chacun étant un vecteurs décrivant les indices des occurrences des lettres correspondantes.

1. (a) Considérons la séquence "AATTCCTCCCGTGACGAAATATA". Créer l'objet `R` correspondant à l'exécution du programme ci-dessus.
   (b) Déterminer le nombre d'occurences de chaque lettre dans la séquence à partir de cette liste.
2. (a) On dispose maintenant de 3 chaînes "ATTCG", "CCGT" et "GCGAGG". Créer une liste comprenant 3 entrées, chacune étant une liste comme celle décrite aux deux questions précédentes.
   (b) Déterminer la longueur de chaque séquence à partir de cette liste
   (c) Déterminer le nombre d'occurences de chaque nucléotide dans chacune des listes. Renvoyer le résultat sous forme de matrice 3 x 4 (on pourra s'aider de la fonction `sapply`).

Exercice 8

1. Charger le tableau de données `diamonds` de la librairie `ggplot2` (commande : `library(ggplot2); data(diamonds)`). Vérifier qu'il s'agit bien d'un `data.frame`. Déterminer les noms des variables considérées et leur nature. Faire un résumé numérique.
2. À l'aide de la commande `subset`, extraire les entrées du tableau telles que
   — les diamands soient de qualité `Premium`
   — le carat soit supérieur à 3
   — le volume (approximatif) soit supérieur à $500mm^3$
   — la qualité soit idéale, le prix inférieur à 1000 et le carat supérieur à .5. Déterminer la répartition des coleurs pour ce sous-ensemble
3. Déterminer le prix moyen par classe de qualité. Même question par intervalle de carat (créer une variable factorielle composée de 6 intervalles à l'aide la fonction `cut`).
4. Tracer le volume en fonction du prix, le carat en fonction du prix. Représenter les boxplot de carat, prix et profondeur par classe de qualité et par couleur.
5. Pour chaque triplet (cut,color,clarity), renvoyer le prix moyen.

Statistiques descriptives

Exercice 9

1. Créer un tableau à 24 lignes et 3 colonnes en lisant le fichier `chromosomes.txt` avec la fonction `read.table`. Chaque ligne représente un chromosome humain (22 autosome, 2 chromosomes sexuels) et les colonnes sont respectivement leur noms, nombre de gènes, et longueur en bases.
2. Représenter Le nombre de gènes en fonction du nombre de bases.
3. Ajouter une colonne supplémentaire au tableau qui spécifie pour chaque chromosome s'il est autosome ou pas.
4. Calculer le nombre total de paires de bases d'un génome humain (pour un homme, puis pour une femme).
5. Exporter le tableau ainsi créé dans un fichier `chromosomes2.txt`

Exercice 10

1. Charger le jeu de données `hdpg` du package `ade4` et lire son descriptif.
2. Nous considérerons le tableau `hdpg\$ind` qui décrit l'échantillon des 1066 individus de l'étude.
3. Combien de populations différentes participent à l'étude ?
4. Dresser les tableaux des effectifs des variables population, région et sexe.
5. Transformer ces tableaux en tableaux de fréquences.
6. Représenter vos tableaux de fréquence par des diagrammes en bâton, et par des camemberts.
7. Représenter les fréquences cumulées.
8. Commenter les représentations.

Exercice 11

Un sondage est réalisé auprès de 100 individus pour savoir où va leur préférence parmi un panel représentatif de marques de bière. Les résultats obtenus se trouvent dans le fichier `bieres.csv`.

1. Lire le fichier de données sous forme de `data.frame`.
2. Combien de marques sont considérées ? Quelles sont-elles ?
3. Compter les occurrences de chacune des marques de bières. Les représenter sous la forme de graphe en barres. Représenter cette distribution sous forme de camembert en choisissant les couleurs vous même. Utiliser une seule fenêtre graphique pour les deux figures.

Commandes utiles : `levels`, `nlevels`, `table`, `barplot`, `pie`, `par`.

Exercice 12

Pour étudier l'effet d'un somnifère, on mesure chez 20 patients le nombre d'heures de sommeil supplémentaires par rapport à la durée moyenne de leur nuit sans traitement. On obtient les résultats suivants :

patient	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
extra	0.7	-1.6	-0.2	-1.2	-0.1	3.4	3.7	0.8	0.0	2.0	1.9	0.8	1.1	0.1	-0.1	4.4	5.5	1.6	4.6	3.4

1. Saisir ces données dans un vecteur.
2. Faire un résumé numérique.
3. Tracer la fonction de répartition empirique puis l'histogramme normalisé des données dans la même fenêtre graphique.
4. Ces données sont en fait issues de deux groupes d'individus : apposer une variable indiquant le groupe associé à l'observation de la variable `extra` sachant que les 10 premiers individus sont issus du groupe 1 et les 10 suivants du groupe 2 (utiliser, par exemple, la commande `data.frame`). Faire un résumé statistique pour chaque groupe et tracer alors les boîtes à moustaches des observations selon les groupes. Qu'en pensez-vous ?

Exercice 13

Le coefficient de Gini permet de mesurer l'inégalité des revenus dans une population. Si tous les individus gagnent le même salaire le coefficient de Gini vaut 0 (situation égalitaire), alors que si un seul individu gagne tous le revenu disponible et les autres rien l'index de gini vaut 1. Les états-unis ont par exemple un coefficient de Gini de 0.47.

1. Charger le jeu de données `gini.Rdata`.
2. Sélectionner les lignes du tableau correspondant à l'année 2007.
3. Tracer l'histogramme des coefficients.
4. Tracer l'histogramme lissé des coefficients.
5. Tracer le boxplot des coefficients.
6. Tracer un diagramme des fréquences cumulées des coefficients.
7. Écrire une fonction R qui rende les pays de coefficient Gini d'index maximum et minimum.
8. Classer les pays par leur coefficient de Gini.
9. Calculer la moyenne, la variance, le coefficient d'asymétrie, le coefficient d'aplatissement pour la distribution des coefficients de gini. Commenter.
10. Combien de pays sont plus égalitaires que la France en europe.

Programmation

Exercice 14

1. Construire une fonction qui calcule la valeur de la fonction $f : x \mapsto sin(x)^2 + \sqrt{|x-3|}$
2. Tracer la courbe représentative de la fonction f sur le domaine $[-6, 3]$
3. Reprendre les mêmes questions pour la fonction :

$$g : x \mapsto \begin{cases} sin(x)^2 log(x) & \text{si } x > 0 \\ sin(x)^2 x & \text{si } x \leq 0 \end{cases}$$

Exercice 15

La formule du calcul de l'indice de masse corporelle (ICM) est la suivante :

$$ICM = \frac{poids(kg)}{taille(m)^2}$$

. l'ICM permet d'évaluer les risques liés à un surpoids chez l'adulte :

$ICM(kg/m^2)$	Classification	Risque
< 18.5	Poids insuffisant	Accru
18.5 à 24.9	Poids normal	Moindre
25 à 29.9	Surpoids	Accru
> 30	Obésité	Elevé

1. Créer une fonction qui prend en entrée le poids et la taille d'un individu et qui renvoie en sortie son ICM.
2. Calculer l'ICM d'une personne :
   — mesurant 1.64 m et pesant 64 kg
   — mesurant 1.61 m et pesant 56 kg
   — mesurant 1.72 m et pesant 102 kg
   — mesurant 1.65 m et pesant 51 kg
3. Créer une seconde fonction qui prend en argument le poids et la taille d'un individu et qui renvoie en sortie sa classification.
4. Quelle est la classification des 4 personnes de la question 2.