Université Paris Descartes
UFR de Mathématiques et Informatique
45, rue des Saints-Pères 75270 Paris cedex 06

Introduction aux probabilités - Licence MIA Parcours Informatique

Travaux Pratiques - $3^{\text{ème}}$ séance

1 Compléments sur R

1.1 Minimum, Maximum et Tri

Il est possible en R de calculer le minimum ou le maximum d'un vecteur à l'aide des commandes `min` et `max`. Par exemple,

```
> X<-runif(10)
> min(X)
> max(X)
```

La fonction `range` renvoie elle directement le minimum et le maximum.
Il est possible de trier le vecteur `X` à l'aide de la commande `sort`. Celui ci est alors trié en ordre ascendant par défaut.

```
> X<-c(3,5,1,4)
> sort(X)
[1] 1 3 4 5
> sort(X,decreasing=T)
[1] 5 4 3 1
```

1.2 Histogrammes

Les histogrammes permettent de visualiser la répartition d'un ensemble de valeurs réelles. Ils permettent de représenter par exemple un échantillons de tirages aléatoires suivant une certaine loi. La fonction `hist` permet de tracer un histogramme. Par exemple pour tracer l'histogramme d'un échantillon de taille 1000 de loi uniforme sur $[0, 1]$, on peut taper
`> x <- runif(1000); hist(x,10)`
Le deuxième argument donne le nombre d'intervalles de regroupement : dans cet exemple, les valeurs de x sont regroupées dans 10 intervalles de même taille et le diagramme affiche le nombre de valeurs à l'intérieur de chaque

intervalle. Pour obtenir un histogramme plus fin, on peut entrer :
`> x <- runif(1000); hist(x,100)`
On peut aussi utiliser la forme suivante (histogramme de densité) :
`> x <- runif(1000); hist(x,100,freq=FALSE)`
Ici l'aire totale des rectangles affichés vaut 1. C'est utile si l'on veut comparer l'histogramme avec la loi de probabilité théorique.

1.3 Graphiques

Le logiciel R permet aussi les sorties graphiques. La fonction `plot(vec1,vec2)` ouvre une fenêtre graphique et affiche les points d'abcisses `vec1` et d'ordonnées `vec2`

```
> x <- seq(-pi,pi,.01); y <- cos(x)+1; plot(x,y)
```

La fonction `points`, avec les mêmes arguments et les mêmes résultats que la fonction `plot`, permet d'afficher des points dans une fenêtre graphique déjà ouverte.

```
> plot(x,y)
> points(x,sin(x)+1)
```

On peut préciser, parmi les arguments des fonctions graphiques `plot` et `points`, la forme du point (`pch`), sa couleur (`col`), son type (`type`), sa taille (`cex`)...

- La forme du point peut être donnée par un argument explicite : `'+'`, `'-'`, `'&'`... ou par un nombre compris entre 0 et 25.
- Comme autres couleurs, on a notamment `'yellow'`, `'pink'`, `'gray'`, `'orange'`, `'green'`, `'chocolate'`, `'tomato'`,... La commande `colors()` renvoie les noms des différentes couleurs disponibles.
- Il y a huit types de tracé possibles :
  1. `'p'` pour afficher des points (type par défaut)
  2. `'l'` pour afficher des lignes
  3. `'b'` pour afficher des points reliés par des lignes sans superposition
  4. `'o'` pour afficher des points reliés par des lignes avec superposition
  5. `'h'` pour afficher des lignes verticales depuis l'axe des abscisses
  6. `'s'` et `'S'` pour afficher des fonctions en escalier
  7. `'n'` pour afficher les points de façon invisible
- Enfin `cex` définit simplement un coefficient multiplicateur pour la taille des points.

```
> plot(1:3,c(1,3,2))
> plot(1:3,c(1,3,2),col='red')
> plot(1:3,c(1,3,2),col='red',type='b')
> plot(1:3,c(1,3,2),col='red',type='b',pch='*')
> plot(1:3,c(1,3,2),col='red',type='b',pch='*',cex=3)
> points(1:3,3:1,col='blue',type='h')
```

Exercice 1. *Définir une fonction* `LGN` *qui à un entier $N \geq 1$, calcule une réalisation de $(X_1, \dots, X_N)$ variables gaussiennes indépendantes de moyenne 2 et de variance 1 puis trace l'évolution de la moyenne empirique de $(X_1, \dots, X_n)$ en fonction de n, pour n de 1 à N. Que remarquez-vous ?*

2 Approximations de la loi binomiale

Dans une entreprise viticole, le remplissage des bouteilles de vin se fait à l'aide d'une machine automatique. En régime normal chaque bouteille reçoit exactement 75 cL de vin, mais à chaque fois la machine a une probabilité $q = 0.03$ de se dérégler, et dans ce cas elle verse une quantité aléatoire de vin entre 0 et 100 cL, modélisée par une loi uniforme sur $[0, 100]$. S'il y a plus de 80 cL versés, la bouteille déborde (donc le niveau de vin reste à 80 cL). On considère que la bouteille est bien remplie si elle contient entre 74 et 76 cL de vin.

Exercice 2. *Soit X la variable aléatoire égale au niveau de vin dans une bouteille. Quelle est la loi de X ? Écrire une fonction* `Bouteilles` *à deux arguments n et q permettant de simuler un n-échantillon de cette loi.*

Exercice 3. *Quelle est la probabilité p qu'une bouteille soit mal remplie ? On note Y le nombre de bouteilles mal remplies dans une série de n bouteilles. Quelle est la loi de Y ? Écrire une fonction* `NombreDefect` *à trois arguments N, n et q qui simule N séries de n bouteilles et renvoie le nombre de bouteilles mal remplies dans chaque série.*

Exercice 4. *Lorsque $n = 100$, dire pourquoi on peut approcher la loi de Y par une loi de Poisson, et donner le paramètre λ de cette loi. Tracer sur un même graphique l'histogramme de densité du nombre de bouteilles défectueuses pour 1000 séries de 100 bouteilles, la loi de Y et la loi de Poisson de paramètre λ.*

Exercice 5. *Lorsque* $n = 10000$*, dire pourquoi on utilise plutôt le théorème de Moivre-Laplace : la loi de* $Y_c = \frac{Y-np}{\sqrt{np(1-p)}}$ *est approchée par une loi normale* $\mathcal{N}(0,1)$*. Tracer sur un graphique l'histogramme de fréquences de* Y *pour une simulation de* 1000 *séries de* 10000 *bouteilles, puis sur un autre graphique l'histogramme de densité de* Y_c *et la loi normale* $\mathcal{N}(0,1)$*.*