Statistiques avec le logiciel R - TD3

Objectifs

— Importer une table de données au format texte.
— Connaitre l'environnement graphique de R

1 Importer des données dans R

Généralement, nous ne souhaitons pas recopier un tableau de données mais l'importer directement à partir d'un format texte. La fonction `read.table` permet d'importer facilement des données dans **R**. Vous pourrez faire appel à l'aide pour obtenir plus d'informations. On notera cependant ces arguments importants :

— `file`, le premier argument, une chaîne de caractère définissant le fichier à importer (son extension incluse)
— `header`, la première ligne donne-t-elle le nom des colonnes ?
— `sep`, le caractère qui sépare les colonnes. Il peut s'agir de l'espace `' '`, d'une virgule `','`, d'un point virgule `';'`, d'une tabulation `'\t'` ou de tout autres caractères.
— `dec`, le symbole utilisé pour signifier les décimales. Attention, en français il s'agit de la virgule `','`, alors que le point est utilisé en anglais.
— `skip`, le nombre de lignes qu'il faut sauter avant de lire la première ligne du tableau.

2 Les graphiques sous R

Il existe un grand nombre de fonctions qui permettent de créer des graphiques avec R. Dans cette section, nous verrons comment utiliser les plus basiques. Les fonctions plus spécifiques aux statistiques descriptives seront étudiées plus tard.

2.1 La fonction `plot`

La fonction `plot(x,y)` est la fonction graphique de base sous **R**. Elle prends deux arguments principaux `x` et `y` qui sont générallement de type numérique et doivent impérativement être de même taille et trace un petit cercle à chaque point de coordonnées (x_i, y_i). Le vecteur `x` est donc le vecteur des abscisses et le vecteur `y` le vecteur des ordonnées. Le cercle en statistique est préféré au point car il représente l'erreur de mesure sur `x` et `y`.

Dans l'exemple traité dans ce TD, nous utilisons les données sur le prix du pain du TD 2.

```
> annees=1992:2011
> pain=c(2.15,2.23,2.30,2.34,2.39,2.42,2.45,2.50,2.56,
+ 2.63,2.73,2.84,2.95,3.00,3.07,3.18,3.32,3.35,3.35,3.42)
> plot(annees,pain)
```

On remarque d'abord que les échelles sont automatiquement choisies pour que l'ensemble des points "rentrent" dans la figure et que les axes sont nommés par le nom des variables `x` et `y`, ce qui donne un aspect "fini" à la figure. Bien entendu, il est possible de changer tous ces paramètres.

— `xlim` et `ylim` permettent de changer les échelles horizontales et verticales. Ce sont deux vecteurs de taille 2, contenant la valeur minimale et maximale de l'axe.
— `xlab` et `xlab` permettent de changer le nom des axes. `xlab` est une chaîne de caractères, c'est le nom de l'axe des abscisses.
— `main` est une chaîne de caractères pour le titre du graphe
— `cex` permet de modifier le style de points (cercles, carrés, étoiles, ...) `?points` pour plus de détails.
— `type` donne le type de graphique si `type='p'` les coordonnées (x_i, y_i) sont représentés par un symbol discret. Si `type='l'` les points (x_i, y_i) sont reliés par une droite.
— `col` permet de modifier la couleur du graphique.

On propose alors de représenter la figure précédente par le graphique

```
> plot(annees,pain,type='l', col='red',
+ xlab='Année', ylab='Prix du pain',
+ main='Evolution du prix du pain')
```

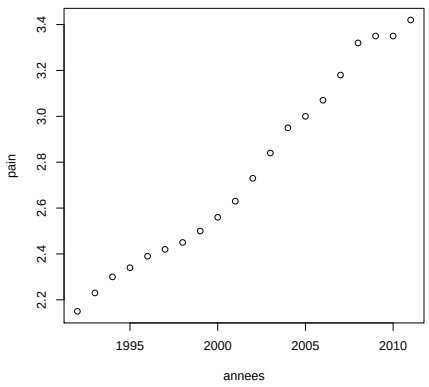


FIGURE 1 – Figure produite par l'instruction `plot(annees,prix)`

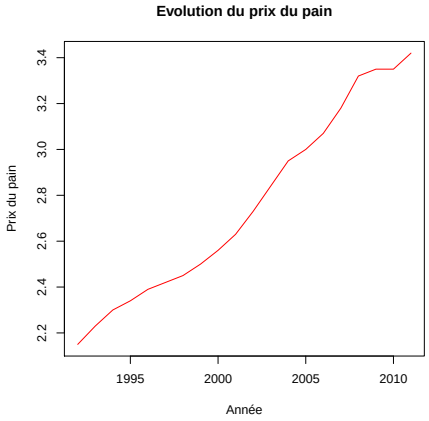


FIGURE 2 – Figure produite par l'instruction `plot(annees,pain,type='l', col='red', xlab='Année', ylab='Prix du pain', main='Evolution du prix du pain')`

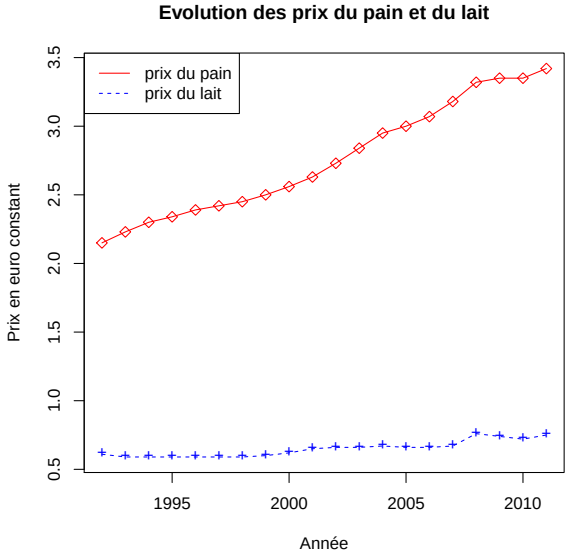
Remarque : La fonction `plot()` se comporte différemment selon la classe des objets `x` et `y` en entrée de la fonction. On distinguera notamment ces trois cas :

- `x` est `numeric` et `y` est `numeric` => diagramme de dispersion
- `x` est `factor` et `y` est `numeric` => boîtes à moustaches
- `x` est `factor` et `y` est `factor` => diagramme mozaïque

2.2 Les fonctions `points` et `lines`

Il est possible d'ajouter à un graphique existant une seconde courbe, ou de nouveaux points à l'aide des fonctions `lines` et `points`. On peut ainsi représenter plusieur courbes sur un même graphique, en changeant la couleur (avec `col`) ou le type de ligne (avec `lty`) si le graphique doit être imprimé en noire et blanc. Enfin la fonction `legend` permet de légender les différentes courbes. Etudier le code ci-dessous.

```
> lait=c(0.61,0.59,0.59,0.59,0.59,0.59,0.59,0.60,0.62,0.65,0.66,0.66,
+ 0.67,0.66,0.66,0.67,0.76,0.74,0.72,0.75)
> plot(annees,pain,type='l', col='red',
+      ylim=c(range(pain,lait)),
+             xlab='Année', ylab='Prix en euro constant',
+             main='Evolution des prix du pain et du lait')
>      lines(annees, lait, col='blue',lty=2)
> points(annees, lait, pch='+', col='blue')
> points(annees, pain, pch=5, col='red')
> legend('topleft',c('prix du pain', 'prix du lait'), col=c('red','blue'), lty=c(1,2))
```



2.3 Les paramètres de l'environnement graphique et la fonction `par`

La fonction `par` permet de moduler l'ensemble des paramètres graphiques. En un coup d'oeil sur l'aide de cette fonction vous verrez qu'il en existe beaucoup. Nous n'en ferons pas la liste exhaustive ici mais discuterons de deux points importants. L'appel à la fonction `par` se fait avant l'appel à la fonction `plot` et modifie les paramètres de l'environnement graphique de la session en cours jusqu'à ce qu'un nouvel appel à la fonction vienne mettre à jour ces paramètres.

— Un graphique est constitué de deux espaces, l'espace du graphique proprement dit dans lequel les points de coordonnées (x, y) sont tracés, et les marges. Ce sont les axes qui délimitent ces deux espaces, donc la graduation des axes, le nom des axes et le titre du graphiques sont dans les marges. Si l'on veut agrandir l'espace dans lequel est tracé notre graphique au profit des marges, il suffit de diminuer les marges avec l'argument `mar` ou `omar`.

— On peut facilement disposer plusieurs graphiques sur une même figure avec les arguments `mfrow` ou `mfcol`. Ces arguments prennent un vecteur d'entiers de longueur 2 `c(nrow,ncol)` précisant en combien de lignes et en combien de colonnes on souhaite séparer la fenêtre graphique. Par exemple, suite à l'instruction `par(mfrow=c(2,2))`, les quatres prochains graphiques appelés par exemple par la fonction `plot` seront placés dans un tableau à 2 lignes et 2 colonnes. Le tableau sera rempli par ligne. La fonction `par(mfrow=c(2,2))` opère exactement de la même façon à l'exception que le tableau sera rempli par colonne.

3 Exercices

Exercice 1

1. Copier le fichier `titanic.dat2.csv` dans votre répertoire de travail.
2. Ouvrir ce fichier avec un éditeur de texte type wordpad et repérer
   — la première ligne du tableau. Celle-ci donne-t-elle le nom des variables ?
   — le caractère qui sépare les colonnes,
   — le nombre de lignes avant la première ligne du tableau.
3. Dans RStudio, créer un nouveau script pour le TD 2.
4. Préciser l'adresse de votre répertoire de travail avec la fonction `setwd`. Attention, il faut utiliser des slashs `/` et non des anti-slashs , comme séparateurs de chemins.
5. Importer la table de données `titanic.dat2.csv` avec la fonction `read.table`. Penser à affecter la table créée dans une variable. On la pourra la nommer `titanic`.
6. Vérifier que votre table est correctement importée avec les fonctions `head(titanic)` et `tail(titanic)`.
7. Recommencer l'opération avec la table `titanic.dat3.csv`.
8. Recommencer l'opération avec la table `titanic.dat4.txt`.

Exercice 2

1. Nous allons utiliser le jeux de données `iris` disponible dans la librairie `datasets`. Charger la librairie avec l'instruction `library(datasets)`.
2. Nommer la table `iris` avec l'instruction `iris=iris`. Cette table contient pour trois espèces d'iris différentes la largeur et la longueur des sépales et pétales mesurées sur 150 fleurs.
3. Produire le graphe de la longueur des sépales en fonction de la largeur des sépales. Vous utiliserez une couleur différente pour chaque espèce. N'oubliez pas de nommer les axes et de donner un titre à votre graphique.
4. Avec la fonction `legend()` légender les couleurs utilisées sur le graphe.
5. Produire un graphique avec des boites à moustache représentant la dispersion de la longeur des sépales par espèce. Utiliser des couleurs différentes pour chaque espèce.
6. En modifiant l'argument `mfrow` de la fonction `par`, produire sur la même fenêtre graphique 4 graphes représentant
   (a) la relation entre la longueur des sépales et la largeur des sépales
   (b) la dispersion de la longeur des sépales par espèce
   (c) la relation entre la longueur des sépales et la longeur des pétales
   (d) la relation entre la longuer des sépales et la largeur des péttales

   Vous utiliserez des couleurs différentes pour chaque espèces.

Exercice 3

Charger le fichier `Valeurs.csv` et représenter sur un même graphique l'évolution de l'indice des prix de plusieurs matières premières importées.