


Cours de biostatistique ∞ Illustrations dans 

# Les Iris de Fisher

ou

## Comment se familiariser avec le logiciel

A.B. Dufour, J.R. Lobry, D.Chessel

3 mars 2008

Installation d'un raccourci dans son espace de travail Le fichier de données iris. Représentation des espèces d'Iris.

### Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation d'un raccourci dans son espace de travail</b>	<b>3</b>
<b>3</b>	<b>Consignes</b>	<b>4</b>
<b>4</b>	<b>Fichier de données : iris</b>	<b>4</b>
<b>5</b>	<b>Représentation des espèces d'Iris</b>	<b>5</b>
<b>6</b>	<b>Représentation de la longueur du pétale</b>	<b>9</b>
<b>7</b>	<b>Représentation de la longueur et de la largeur du pétale</b>	<b>11</b>
<b>8</b>	<b>Représentation de la longueur du pétale selon les différentes espèces</b>	<b>13</b>
<b>9</b>	<b>Pour aller plus loin ...</b>	<b>14</b>
	<b>Références</b>	<b>17</b>

## 1 Introduction

$\mathbb{R}$  est un logiciel de calcul statistique distribué selon la licence *GNU General Public License*. La version actuellement disponible sur le campus est la version 2.1.0. La dernière version 2.2.0 est disponible dans les archives du réseau CRAN (Comprehensive R Archive Network) dont un des miroirs est disponible sur le site de l'Université (<http://cran.univ-lyon1.fr/>)<sup>1</sup>. La version utilisée pour exécuter les exemples de ce document est donnée en pied de page.



FIG. 1 – Sir R.A. Fisher (1890-1962)

Les données utilisées ici sont célèbres. Elles ont été collectées par Edgar Anderson [1]. Ce sont les mesures en centimètres des variables suivantes : longueur du sépale (`Sepal.Length`), largeur du sépale (`Sepal.Width`), longueur du pétale (`Petal.Length`) et largeur du pétale (`Petal.Width`) pour trois espèces d'iris : *Iris setosa*, *I. versicolor* et *I. virginica*.

Sir R.A. Fisher a utilisé ces données pour construire des combinaisons linéaires des variables permettant de séparer au mieux les trois espèces d'iris [2].



FIG. 2 – *I.setosa*, *I.versicolor*, *I.Virginica*

<sup>1</sup>La liste de tous les miroirs est ici : <http://cran.r-project.org/mirrors.html>

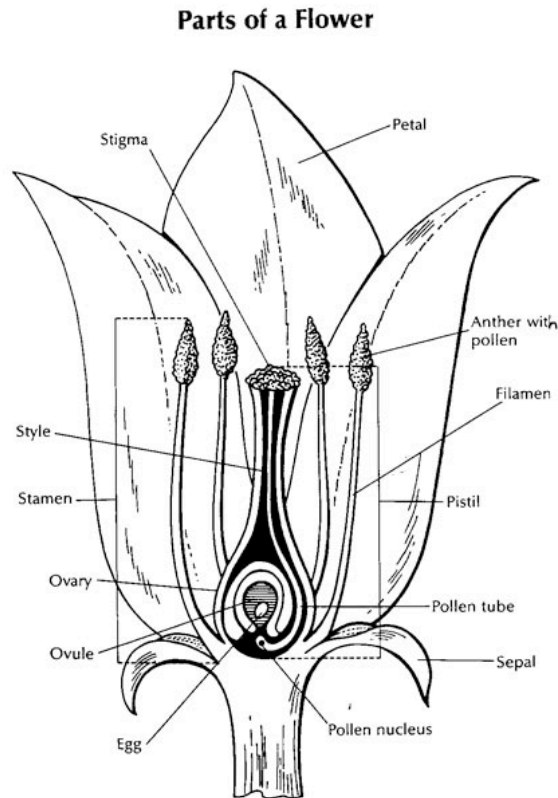


FIG. 3 – Description d’une fleur

## 2 Installation d’un raccourci dans son espace de travail

Si vous êtes dans une salle de TP de l’Université, normalement vous trouvez sur le bureau un raccourci qui lance le programme à partir d’un serveur. Copiez ce raccourci dans votre espace de travail. Si vous ne voyez pas de raccourci de sur le bureau, déroulez le menu Démarrer, cherchez et avec un clic droit sélectionnez ”Créer un raccourci”. Recopiez ce raccourci dans votre espace de travail. Puis avec le clic droit de la souris, aller dans **Propriétés** et remplacer dans la partie **Démarrer** dans l’information existante par le chemin d’accès à votre dossier de travail.

Les objets créés lors de la session sont enregistrés dans un fichier intitulé `.RData` situé dans le dossier de travail. L’historique de la session est conservée dans un fichier `.Rhistory`. Les fichiers `.RData` s’ouvrent directement dans par un double-clic ou par glisser-déposer sur le raccourci. Les fichiers `.Rhistory`

s'ouvrent directement avec l'éditeur de texte de votre choix.

Lancez le programme et vérifiez le dossier de travail :

```
getwd()
```

```
[1] "D:\\Nicolas\\Exo1"
```

L'exemple montre que le dossier de travail s'appelle `Exo1`. Il se trouve dans le dossier `Nicolas` qui lui même se trouve dans le répertoire des étudiants. Les commandes sont écrites en rouge, les réponses apparaissent en bleue.

### 3 Consignes

La manière la plus simple pour se familiariser avec `R` est de l'utiliser afin de comprendre un jeu de données particulier. Considérons donc les données provenant des iris de Fisher, données sur lesquelles vous pouvez avoir envie de faire une analyse ...de données. Suivez pas à pas les étapes de la session ci-dessous et voyez ce qui se passe. Faites les exercices proposés et n'hésitez pas à utiliser l'aide en ligne de `R`. Si vous voulez par exemple connaître le contenu de la fonction `hist`, il vous suffit de taper la commande `?hist`. Vous ne comprendrez peut-être pas tous les détails mais la meilleure chose à faire est de taper le code et de voir le résultat produit. **Soyez curieux.**

Le symbole `>` se trouve à chaque début de ligne de commandes (appelé prompt, ou invite de commande). Le symbole `<-` est le symbole d'affectation. Le symbole `#` signifie le début d'un commentaire.

Lorsque vous travaillez sous `R`, il peut être intéressant de conserver les résultats et les graphiques de vos analyses. Le plus simple, dans un premier temps, est de les enregistrer dans un document word à l'aide du copier / coller. Pour ce faire, allez dans le menu "File", sélectionnez "Copy to the clipboard" "as a Bitmap". Notez que les graphes peuvent être réduits ou agrandis sans déformation.

### 4 Fichier de données : iris

`R` est un ensemble de bibliothèques de fonctions appelées "packages". Chaque bibliothèque contient des jeux de données. Pour connaître par exemple les jeux de données de la distribution de base, entrez l'instruction suivante :

```
data()
```

En voici un extrait :

airmiles	Passenger Miles on Commercial US Airlines (1937-1960)
airquality	New York Air Quality Measurements
anscombe	Anscombe's Quartet of "Identical" Simple Linear Regressions
attenu	The Joyner-Boore Attenuation Data
...	...
iris	Edgar Anderson's Iris Data
...	...
USArrests	Violent Crime Rates by US State
USJudgeRatings	Lawyers' Ratings of State Judges in the US Superior Court
USPersonalExpenditure	Personal Expenditure Data
uspop	Populations Recorded by the US Census
VADeaths	Death Rates in Virginia (1940)
volcano	Topographic Information on Auckland's Maunga Whau Volcano
warpbreaks	The Number of Breaks in Yarn during Weaving
women	Average Heights and Weights for American Women

Notez la présence de `iris`. Pour analyser ces données, il faut les charger en mémoire à l'aide de l'instruction :

```
data(iris)
```

Il existe d'autres procédés pour charger un jeu de données dans le logiciel `R` mais ce n'est pas l'objet de ce travail.

**Exercice.** Tapez une à une chacune des instructions ci-dessous et notez le résultat obtenu. Attention, le logiciel `R` n'est pas indifférent aux majuscules et aux minuscules.

```
iris
dim(iris)
names(iris)
iris$Species
iris$Petal.Length
```

## 5 Représentation des espèces d'Iris

La dernière colonne des données `iris` contient le nom des espèces réparties en trois catégories : `setosa`, `versicolor` et `virginica`. Pour accéder à celle-ci, il faut utiliser l'instruction `iris$Species`. On dit que la dernière colonne contient une variable *qualitative* à trois *modalités* appelées *levels* dans `R`. La fonction `levels()` appliquée à la colonne `iris$Species` donne les modalités de la variable :

```
levels(iris$Species)
```

```
[1] "setosa"      "versicolor" "virginica"
```

Pour résumer l'information contenue dans cette variable, on utilise l'instruction `summary()` :

```
summary(iris$Species)
```

```
setosa versicolor virginica
 50         50         50
```

Cette information peut être obtenue en construisant un tableau (`table()`) comptabilisant le nombre d'individus par modalité. Pour ce faire, tapez :

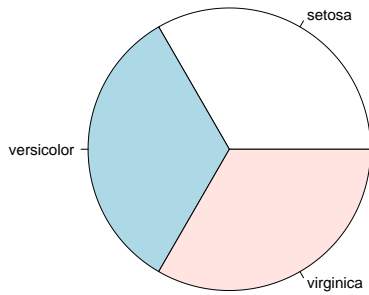
```
table(iris$Species)
```

```
setosa versicolor virginica
 50         50         50
```

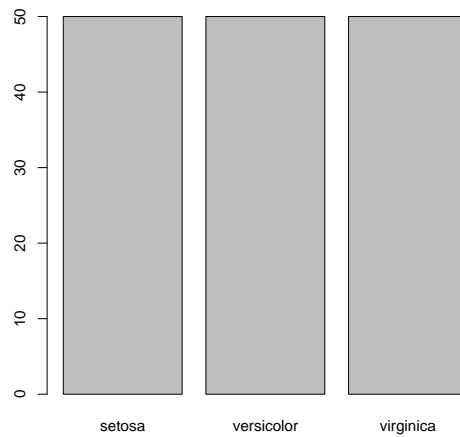
et comparez avec le résultat précédent.

Le logiciel `R` permet de réaliser d'excellents graphiques. Lorsqu'une instruction graphique est lancée, une nouvelle fenêtre "device" est ouverte. Les représentations graphiques classique liées aux variables qualitatives sont la représentation en secteurs ou camembert (`pie()`), la représentation en bâtons (`barplot()`), et la représentation de Cleveland (`dotchart()`). Entrez les instructions suivantes :

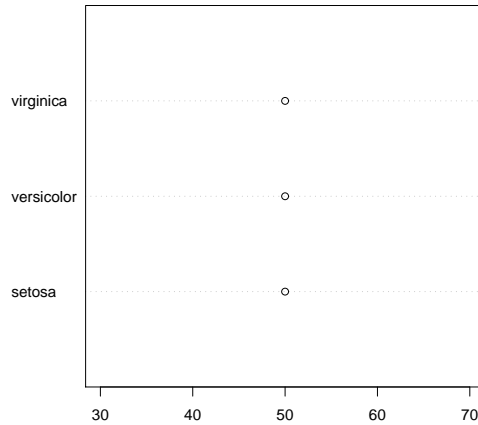
```
pie(table(iris$Species))
```



```
barplot(table(iris$Species))
```



```
dotchart(table(iris$Species))
```



Il existe un paramètre permettant de découper la fenêtre graphique : `par(mfrow = c(nl, nc))` ou `par(mfcol = c(nl, nc))`. `nl` définit le nombre de graphiques en lignes et `nc` définit le nombre de graphiques en colonnes. `mfrow` signifie que l'ordre d'entrée des graphiques s'effectue selon les lignes et `mfcol` signifie que l'ordre d'entrée des graphiques s'effectue selon les colonnes. Supposons que nous voulions représenter six graphiques dans une fenêtre en deux lignes et trois colonnes.

La première instruction conduit à entrer les graphiques selon l'ordre :

1	2	3
4	5	6

La seconde instruction conduit à entrer les graphiques selon l'ordre :

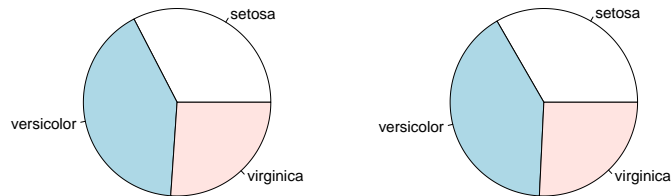
1	3	5
2	4	6

**Exercice.** Deux botanistes se sont également intéressés aux iris et ont collecté les espèces suivantes.

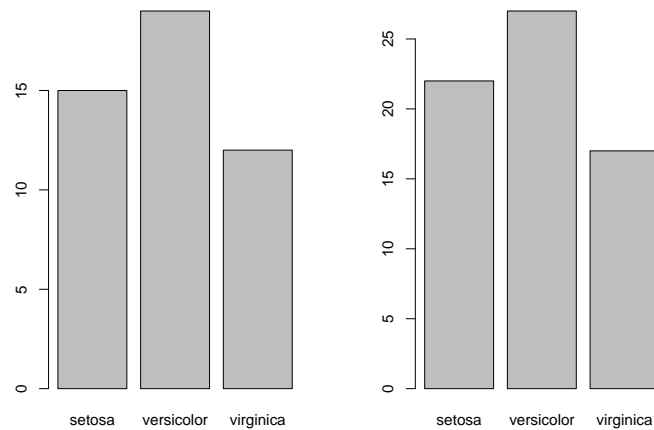
```
collection1 <- rep(c("setosa", "versicolor", "virginica"), c(15,
19, 12))
collection2 <- rep(c("setosa", "versicolor", "virginica"), c(22,
27, 17))
```

En utilisant la commande `par(mfrow = c(1, 2))`,

1. construire les camemberts liés à ces deux nouvelles distributions et commenter ;

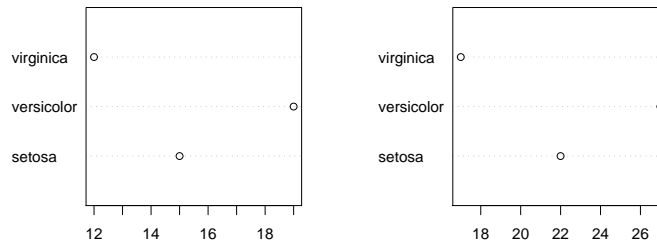


2. construire les représentations en bâtons de ces deux nouvelles distributions et commenter ;



3. construire les représentations de Cleveland de ces deux nouvelles distributions et commenter ;





4. discuter des avantages et des inconvénients de ces trois types de représentations.

## 6 Représentation de la longueur du pétale

La troisième colonne (`Petal.Length`) du jeu de données `iris` contient la longueur du pétale. Il s'agit d'une variable mesurée qualifiée alors de variable *quantitative*. Pour résumer l'information contenue dans cette variable, on utilise la fonction `summary()` et on obtient le résultat :

```
summary(iris$Petal.Length)

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.000  1.600   4.350   3.758  5.100   6.900
```

La plus petite (`Min`) longueur de pétale est 1 cm tandis que la plus grande (`Max`) est 6.9 cm. La moyenne (`Mean`) représente la somme des valeurs de la *distribution* divisée par le nombre total d'iris. Elle vaut 3.758 cm.

Si l'ensemble des 150 longueurs de pétale sont classées par ordre croissant, `1st Qu.`, `Median` et `3rd Qu.` sont les trois valeurs qui permettent de couper la distribution en quatre parties égales. On les appelle respectivement premier *quartile*, médiane (ou deuxième quartile) et troisième quartile.

**Exercice.** Essayons de retrouver ces six valeurs de *paramètres*.

```
min(iris$Petal.Length)
```

```
[1] 1
```

```
max(iris$Petal.Length)
```

```
[1] 6.9
```

```
sum(iris$Petal.Length)
```

```
[1] 563.7
```

```
length(iris$Petal.Length)
```

```
[1] 150
```

```
sum(iris$Petal.Length)/length(iris$Petal.Length)
```

```
[1] 3.758
```

```
sort(iris$Petal.Length)
```

```
[1] 1.0 1.1 1.2 1.2 1.3 1.3 1.3 1.3 1.3 1.3 1.3 1.3 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4
[21] 1.4 1.4 1.4 1.4 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.6 1.6 1.6
[41] 1.6 1.6 1.6 1.6 1.7 1.7 1.7 1.7 1.9 1.9 3.0 3.3 3.3 3.5 3.5 3.6 3.7 3.8 3.9 3.9
[61] 3.9 4.0 4.0 4.0 4.0 4.0 4.1 4.1 4.1 4.2 4.2 4.2 4.2 4.3 4.3 4.4 4.4 4.4 4.4 4.4 4.5
[81] 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.6 4.6 4.6 4.7 4.7 4.7 4.7 4.7 4.7 4.8 4.8 4.8 4.8 4.9
[101] 4.9 4.9 4.9 4.9 5.0 5.0 5.0 5.0 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.2 5.2 5.3 5.3
[121] 5.4 5.4 5.5 5.5 5.5 5.6 5.6 5.6 5.6 5.6 5.6 5.6 5.7 5.7 5.7 5.8 5.8 5.8 5.9 5.9 6.0
[141] 6.0 6.1 6.1 6.1 6.3 6.4 6.6 6.7 6.7 6.9
```

```
ordLpetal <- sort(iris$Petal.Length)
ordLpetal
```

```
[1] 1.0 1.1 1.2 1.2 1.3 1.3 1.3 1.3 1.3 1.3 1.3 1.3 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4 1.4
[21] 1.4 1.4 1.4 1.4 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.6 1.6 1.6
[41] 1.6 1.6 1.6 1.6 1.7 1.7 1.7 1.7 1.9 1.9 3.0 3.3 3.3 3.5 3.5 3.6 3.7 3.8 3.9 3.9
[61] 3.9 4.0 4.0 4.0 4.0 4.0 4.1 4.1 4.1 4.2 4.2 4.2 4.2 4.3 4.3 4.4 4.4 4.4 4.4 4.4 4.5
[81] 4.5 4.5 4.5 4.5 4.5 4.5 4.5 4.6 4.6 4.6 4.7 4.7 4.7 4.7 4.7 4.7 4.8 4.8 4.8 4.8 4.9
[101] 4.9 4.9 4.9 4.9 5.0 5.0 5.0 5.0 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.1 5.2 5.2 5.3 5.3
[121] 5.4 5.4 5.5 5.5 5.5 5.6 5.6 5.6 5.6 5.6 5.6 5.6 5.7 5.7 5.7 5.8 5.8 5.8 5.9 5.9 6.0
[141] 6.0 6.1 6.1 6.1 6.3 6.4 6.6 6.7 6.7 6.9
```

```
sum(ordLpetal)/length(ordLpetal)
```

```
[1] 3.758
```

```
ordLpetal[38]
```

```
[1] 1.6
```

```
(ordLpetal[75] + ordLpetal[76])/2
```

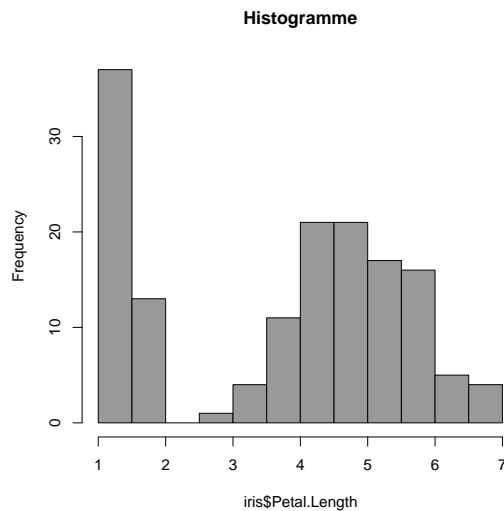
```
[1] 4.35
```

```
ordLpetal[113]
```

```
[1] 5.1
```

Une des représentations adéquates est l'histogramme (`hist()`) :

```
hist(iris$Petal.Length, col = grey(0.6), main = "Histogramme")
```

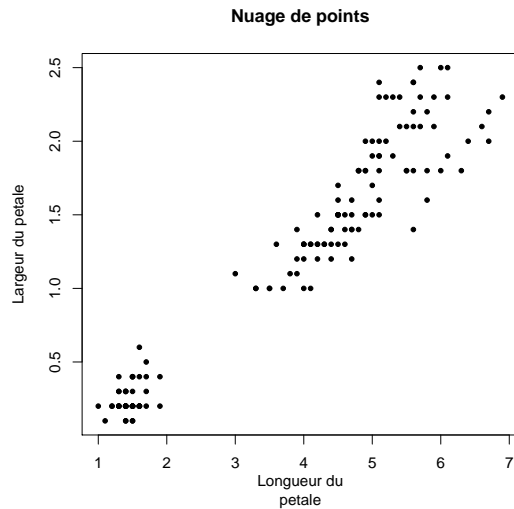


**Exercice.** Réaliser le même type d'analyse sur chacune des autres variables quantitatives : largeur du pétale, longueur du sépale et largeur du sépale. Notez que vous n'avez pas toutes les instructions à réécrire en utilisant le système de flèches du clavier. ↑ et ↓ vous permettent de retrouver les fonctions que vous avez utilisées. ← et → vous permettent de vous déplacer dans la fonction et donc, d'en changer certains paramètres.

## 7 Représentation de la longueur et de la largeur du pétale

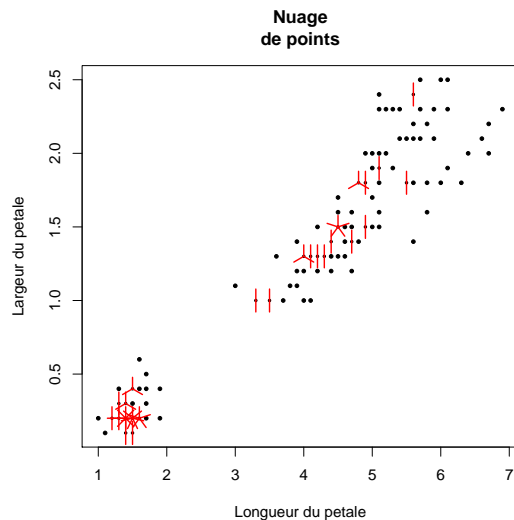
Une fois réalisés les graphiques pour chaque variable prise séparément, l'étude peut porter sur la relation entre deux variables. On parle de croisement de deux variables ou d'étude *bivariée*. La représentation graphique liant deux variables quantitatives est le nuage de points. Représentons par exemple la longueur et la largeur du pétale pour les 150 iris contenus dans le fichier de données. Commentaire.

```
plot(iris$Petal.Length, iris$Petal.Width, xlab = "Longueur du\npetale",  
     ylab = "Largeur du petale", main = "Nuage de points", pch = 20)
```



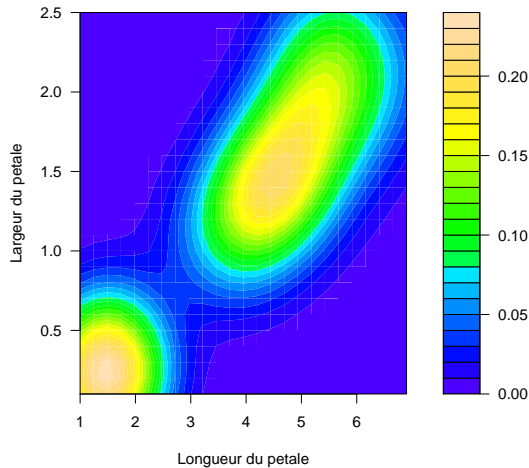
Dans cette représentation graphique, plusieurs individus peuvent être situés sur un même point. La fonction `sunflowerplot()` permet de visualiser ces superpositions.

```
sunflowerplot(iris$Petal.Length, iris$Petal.Width, xlab = "Longueur du petale",
              ylab = "Largeur du petale", main = "Nuage\nde points", pch = 20)
```



Quand le nombre de points devient trop important, on peut alors représenter la densité des points au lieu des points eux-mêmes, par exemple :

```
library(MASS)
densite <- kde2d(iris$Petal.Length, iris$Petal.Width)
filled.contour(densite, color = topo.colors, xlab = "Longueur du petale",
              ylab = "Largeur du petale")
```

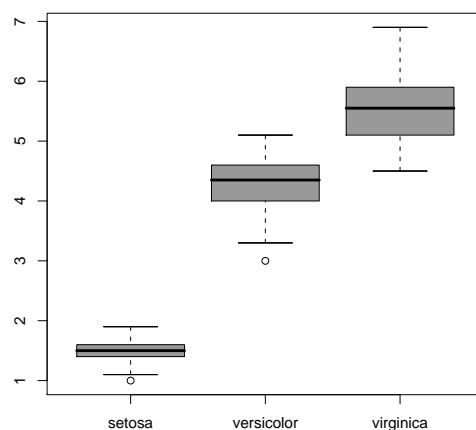


**Exercice.** Réaliser l'étude du croisement de deux variables quantitatives de votre choix. Il est clair que le sens biologique de l'étude ne doit pas être négligé.

## 8 Représentation de la longueur du pétale selon les différentes espèces

La représentation graphique permettant de lier une variable qualitative et une variable quantitative est la boîte à moustaches (`boxplot()`). Représentons par exemple la longueur des pétales en fonction de l'espèce. Commentaire.

```
boxplot(iris$Petal.Length ~ iris$Species, col = grey(0.6))
```



**Exercice.** Choisissez une autre variable quantitative, croisez-la avec la variable espèce d'iris et commentez.

## 9 Pour aller plus loin ...

Le nuage de points comme les boîtes à moustaches montrent que les données morphologiques des iris semblent liées à l'espèce. Il pourrait donc être intéressant de réaliser des graphiques différents pour chacune des modalités *I. setosa*, *I. versicolor* et *I. virginica* ou de superposer l'information espèce dans le graphique des nuages de points. Nous vous proposons ici quelques développements. Libre à vous, de les refaire ou d'en trouver d'autres ...

```
summary(iris)
```

```

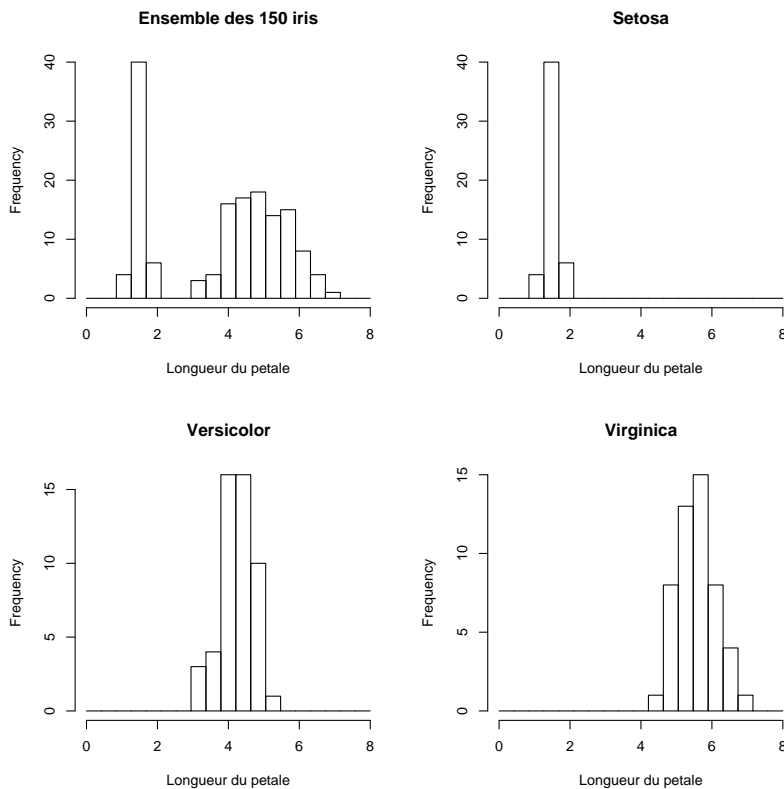
Sepal.Length   Sepal.Width   Petal.Length   Petal.Width   Species
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa   :50
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

```

```

par(mfrow = c(2, 2))
brk = seq(from = 0, to = 8, length = 20)
hist(iris$Petal.Length, main = "Ensemble des 150 iris", xlab = "Longueur du petale",
     breaks = brk)
hist(iris$Petal.Length[iris$Species == "setosa"], main = "Setosa",
     xlab = "Longueur du petale", breaks = brk)
hist(iris$Petal.Length[iris$Species == "versicolor"], main = "Versicolor",
     xlab = "Longueur du petale", breaks = brk)
hist(iris$Petal.Length[iris$Species == "virginica"], main = "Virginica",
     xlab = "Longueur du petale", breaks = brk)

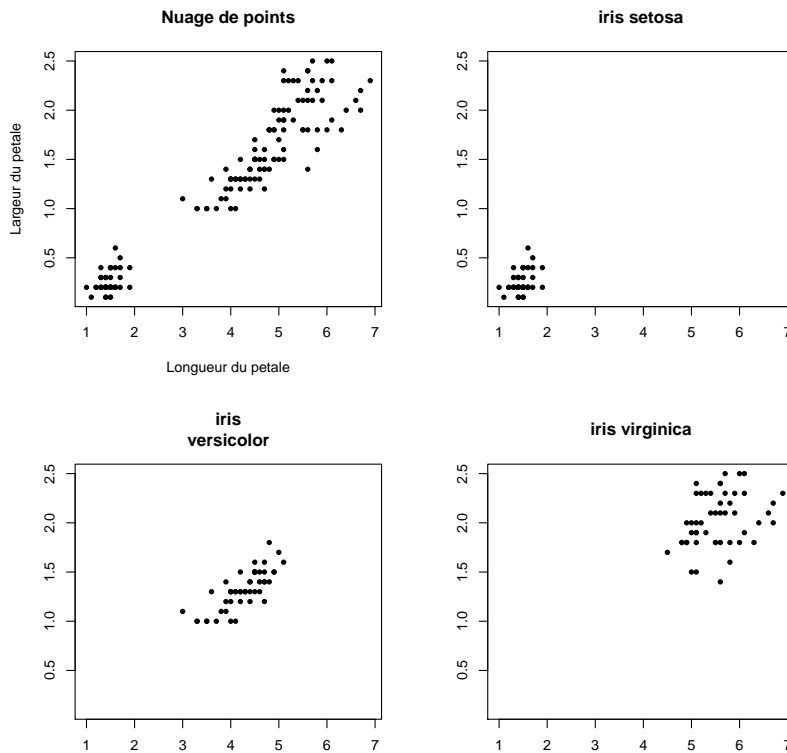
```



```

par(mfrow = c(2, 2))
plot(iris$Petal.Length, iris$Petal.Width, xlab = "Longueur du petale",
     ylab = "Largeur du petale", main = "Nuage de points", pch = 20)
plot(iris$Petal.Length[iris$Species == "setosa"], iris$Petal.Width[iris$Species ==
"setosa"], xlim = c(1, 6.9), ylim = c(0.1, 2.5), xlab = "",
     ylab = "", main = "iris setosa", pch = 20)
plot(iris$Petal.Length[iris$Species == "versicolor"], iris$Petal.Width[iris$Species ==
"versicolor"], xlim = c(1, 6.9), ylim = c(0.1, 2.5), xlab = "",
     ylab = "", main = "iris\versicolor", pch = 20)
plot(iris$Petal.Length[iris$Species == "virginica"], iris$Petal.Width[iris$Species ==
"virginica"], xlim = c(1, 6.9), ylim = c(0.1, 2.5), xlab = "",
     ylab = "", main = "iris virginica", pch = 20)

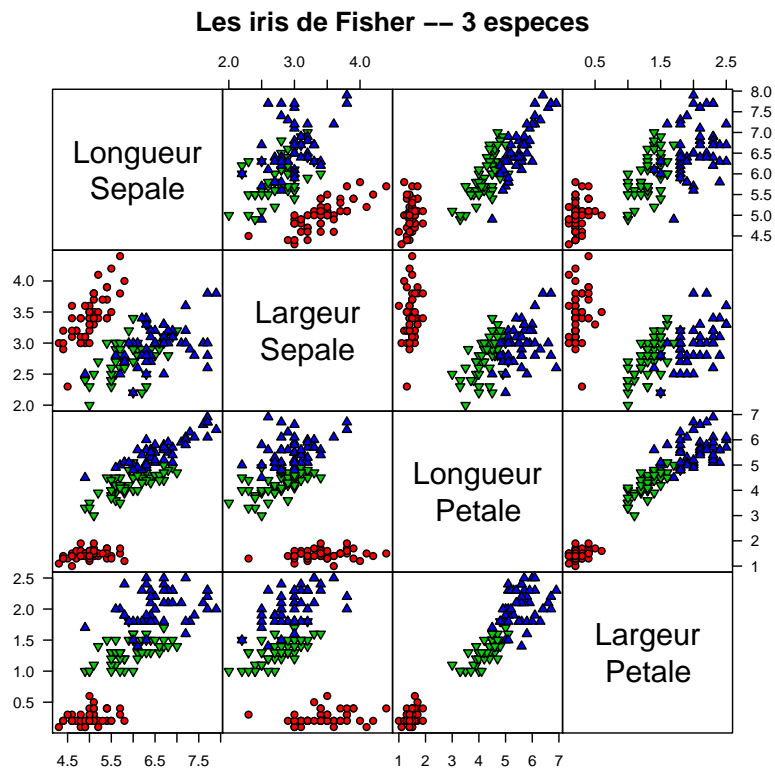
```



```

spiris <- unclass(iris$Species)
pairs(iris[1:4], main = "Les iris de Fisher -- 3 especes", pch = c(21,
25, 24)[spiris], bg = c("red", "green3", "blue")[spiris], las = 1,
     gap = 0, labels = c("Longueur\nSepale", "Largeur\nSepale", "Longueur\nPetale",
"Largeur\nPetale"))

```



En guise de conclusion, soulignons le fait que les représentations graphiques sont une étape fondamentale dans la connaissance des données et que le logiciel est un *excellent* outil. Les représentations graphiques sont là pour éclairer la nature des données et non pour souligner votre côté artistique. Chaque information ajoutée à un graphe, comme par exemple une couleur, doit contribuer à cet éclairage.



## Références

- [1] Anderson E. The irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, 59 :2-5, 1935.
- [2] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2) :179-188, 1936.