
Graphiques avec R

Le logiciel R offre de puissantes facilités graphiques permettant de produire des graphiques statistiques de grande qualité. Son principal atout réside dans les larges possibilités de personnalisation.

Pour avoir un aperçu de la palette des possibilités vous pouvez taper `demo(graphics)`. Pour une perspective plus complète, vous pouvez consulter par exemple : <https://www.r-graph-gallery.com/>.

1 Représentations graphiques simples

1.1 Tracé d'une fonction

1. À l'aide de la fonction `plot`, tracer le graphe de la fonction sinus pour 50 points entre $-\pi$ et π .
2. Faire la même représentation avec une ligne, puis avec une ligne reliant les points (argument `type`)
3. Ajouter le titre suivant : **Graphe de la fonction sinus** (argument `main`)
4. Utiliser la fonction `curve` et obtenir la même représentation

1.2 Tracé de deux fonctions

1. Sur le même graphique, tracer les graphes des fonctions sinus et cosinus entre $-\pi$ et π . (fonction `lines` ou `points`).
2. Pour différencier les deux fonctions, tracer la fonction cosinus en rouge (argument `col`) et celle de sinus en pointillés (argument `lty`)
3. Ajouter une légende en haut à gauche pour différencier les deux courbes (fonction `legend`)

1.3 Variable quantitative continue

Importer le jeu de données `etat2`.

1. Représenter la variable `Income` sous forme d'histogramme.
2. Représenter le même histogramme mais avec les classes `[3000,4000]`, `[4000,5000]`, `[5000,6000]`, `[6000,7000]`.
3. Faire la même représentation mais en donnant les probabilités empiriques en ordonnée (autrement dit en affichant le pourcentage de l'effectif par classe).
4. Superposer à ce graphe la densité estimée sur la variable `Income` avec la fonction `lines`
5. Ajouter le titre **Histogramme des revenus** à l'aide de la fonction `title`
6. Visualiser la répartition des valeurs de revenus grâce à un boxplot (fonction `boxplot`)
7. L'instruction `par(mfrow=c(n,p))` partitionne la fenêtre graphique en np graphes répartis sur n lignes et p colonnes. Représenter côte à côte l'histogramme et le boxplot précédents.
Q.S : Quelle est la différence entre `par(mfrow=c(2,2))` et `par(mfcol=c(2,2))` ?

1.4 Variable qualitative ou quantitative discrète

Créer le facteur `Richesse` à 3 niveaux comme décrit dans le TP n° 3, partie 2.2.2.

1. Représenter la variable `Richesse`. Commenter le résultat.

2 Représentation de plusieurs variables

2.1 Variables quantitatives/qualitatives

Charger le jeu de données `UScereal` (disponible dans le package `MASS`)

1. Décrive brièvement le jeu de données
2. Représenter les boxplots du nombre de calories en fonction du niveau de l'étagère de présentation. Même chose pour la quantité de matières grasses. Commentez ces résultats.

2.2 Variables quantitatives

1. Toujours sur le jeu de données `UScereal`, représenter le nuage de points de la quantité de potassium en fonction de la quantité de fibres. Commentez le graphique obtenu.
2. Ajouter la droite d'équation $y = 49.5 + 28.3x$ sur le graphique en rouge (fonction `abline`). Doubler la largeur du trait (argument `lwd`).
3. Calculer le centre de gravité du nuage de points, le représenter par une croix bleue sur le graphique (fonction `points`, argument `pch`) et écrire "G", toujours en bleu, juste en dessous via la fonction `text`.
4. Représenter les nuages des points pour tous les croisements deux à deux de variables quantitatives (fonction `pairs`). Commentez les graphiques obtenus.
5. Charger à présent le jeu de données `EuStockMarkets`. Il est constitué de séries temporelles ou chronologiques, c'est-à-dire de quantités évoluant au cours du temps. Représenter sur un même graphique l'ensemble des séries à l'aide de la fonction `matplot`. Spécifier que les abscisses représentent le temps (`xlab`), les ordonnées, le prix de l'action (`ylab`) et ajouter une légende pour différencier les courbes. On peut utiliser la fonction `time` pour afficher les années en abscisse.

3 Graphique en 3 dimensions

Les fonctions classiques de représentation 3D sur des grilles de points sont les fonctions `persp` pour la représentation des surfaces avec effets de perspective, `contour` pour la représentation des lignes de niveau et `image` pour la représentation des lignes de niveau avec effets de couleur.

On s'intéresse à la représentation de la fonction suivante ("chapeau mexicain") :

$$f : (x, y) \mapsto z = \frac{10 \sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$$

1. On commence par programmer la fonction `f`.

```
> f = function(x,y) {a = sqrt(x^2+y^2) ; 10*sin(a)/a }
```
2. Créer deux vecteurs `x` et `y` de 200 valeurs chacun, comprises entre -10 et 10
3. Calculer les valeurs de la fonctions en chaque couple de points de la grille (fonction `outer`)
4. Tracer cette fonction à l'aide de la fonction `persp`
5. Faire pivoter cette figure à l'aide des arguments `theta` et `phi`

6. Représenter la même fonction à l'aide des fonctions `contour` et `image`

À noter qu'il existe aussi un package appelé `rgl` permettant de faire des représentations 3D dans un mode interactif (c'est-à-dire permettant de faire des rotations graphiques à l'aide de la souris ou d'utiliser des effets d'éclairage).

Q.S. En utilisant la fonction `cloud` du package `lattice`, représenter les calories en fonction de la quantité de sucre et matières grasses (du jeu de données `UScereal`).

4 Graphes multiples

Reprenons à présent le jeu de données `UScereal`.

1. À l'aide de la fonction `par` et de l'argument `mfrow`, représenter en lignes les 3 graphiques suivants :
 - Un premier représentant la quantité de sodium en fonction du producteur avec une couleur par niveau. Redéfinir les limites du graphique avec `ylim` pour avoir un graphique entre 0 et 400.
 - Un deuxième représentant le nuage de points de la quantité de sodium en fonction de la quantité de sucre
 - Un dernier représentant l'histogramme des quantités de sodium observées
2. Ajouter titre, nom des abscisses et ordonnées à chacun des graphes
3. Faire les 3 mêmes graphiques (sans titre et sans légende des axes) mais sur deux lignes avec uniquement le premier graphe sur la première ligne et les deux autres côte à côte sur la deuxième ligne (fonction `layout`)
4. Rétrécir les marges de manière à maximiser les zones graphiques (fonction `par`, argument `mar`).
5. Ajouter le titre **Le sodium dans les céréales**, commun aux trois graphiques à l'aide de la fonction `par`, argument `oma` (pour réserver une place pour le titre) et de la fonction `mtext`.

5 Sauvegarde des graphiques

La sauvegarde des graphiques peut s'effectuer sous une grande variété de formats comme `postscript` (.eps), `pdf`, `jpeg`, `png` pour ne citer que les plus courants. On pourra distinguer principalement les formats dits vectoriels (`postscript`, `pdf`) qui sont composés d'objets géométriques auxquels on peut appliquer des transformations (agrandissement ou rétrécissement par exemple) des formats dits matriciels qui sont composés de pixels et qui ne peuvent donc pas subir de transformations sans dégrader la qualité de l'image.

La sauvegarde de graphiques sous R se fait (en lignes de commande) en plusieurs étapes :

- Ouvrir un dispositif graphique au format désiré (fonctions `postscript`, `pdf`, `jpeg` ... etc)
- Faire la représentation graphique souhaitée
- Fermer le dispositif graphique avec la commande `dev.off()`

Sauvegarder le dernier graphique dans les formats `postscript`, `pdf` et `jpeg`.