

Feuille de route pour construire ma boîte outils de dataviz sous R

Exercice 1 – les graphiques Base

Dans cet exercice très simple nous allons manipuler quelques fonctions du package Base – et mesurer les customisations des graphiques de sorties.

1. Réaliser un graphique permettant d'analyser très rapidement des nuages de points 2 à 2 de la table mtcars (du package ggplot2)
2. Modifier la couleur des points (choisir d'afficher les points en bleu)
3. Réaliser un histogramme de la variable wt de la table mtcars
4. Modifier la couleur des barres (choisir de les afficher en gris)

Pour personnaliser davantage un graphique, il faut connaître les options par défaut.

Pour les retrouver : `?plot.default`.

- `main`: pour contrôler le titre.
- `xlab, ylab`: pour les labels des axes x et y.
- `col`: modifier la couleur.
- `cex`: modifier la taille des points.
- `pch`: modifier le type de points (circle, dot, triangle, etc...)
- `lwd`: épaisseur de la ligne.
- `lty`: type de ligne (solid, dashed, dotted, etc...).

Parfois nous avons besoin de transformer les données pour réaliser le graphique adéquat :

Nous allons utiliser 4 fonctions :

- `table()` : pour réaliser le tableau de fréquence
- `density()` : pour calculer une densité
- `loess()` : calcule la courbe de lissage
- `predict()` : prédit les nouvelles valeurs basées sur un modèle

1. Créer une variable object contenant la densité de la variable wt de la table mtcars
2. Créer le graphique de densité
3. Créer l'histogramme de densité

Pour une variable discrète

Créer un diagramme en barres pour la variable carb de la table mtcars

Pour aller plus loin : s'entraîner avec les exemples de la feuille « base-r-cheatsheet »

Exercice 2 – package lattice

De la même manière, nous allons faire des exercices très simples pour vous permettre de prendre en main le package. Les liens vous permettront d'aller vers des exemples / exercices afin d'approfondir si vous le souhaitez.

1. Nous allons utiliser la table USCancerRates du package latticeExtra (contient le taux pour 100 000 mort par cancer, ventilé par sexe de 1999 à 2003)

```
library(lattice)
data(USCancerRates, package = "latticeExtra")
str(USCancerRates)
```

2. Créer l'histogramme du taux de cancer chez les hommes
3. Créer l'histogramme du taux de cancer chez les femmes
4. Créer un nuage de point entre rate.female et rate.male

Les arguments communs à toutes les fonctions

- `main`: pour contrôler le titre.
 - `xlab`, `ylab`: pour les labels des axes x et y.
5. Créer l'histogramme du taux de cancer masculin, en y ajoutant un titre + forcer le nombre d'intervalles à 30 (argument : `nint`)
 6. Sur la table `airquality` du package `lattice`, créer l'histogramme de la variable Ozone. Ajouter l'option `type = « count »`
 - *"percent", the default, gives percentage or relative frequency.*
 - *"count" gives bin count, which is the default in hist().*
 - *"density" gives a density histogram.*
 7. Pour afficher les observations en dessous de la fonction de densité, on peut utiliser l'option `plot.points` spécifique à la fonction `densityplot()`. Tracer ce type de graphique pour la variable Ozone de la table `airquality`

Box and whisker plot : pour analyser une variable continue mais peut également être utilisé à des fins de comparaison.

Analyser le taux de cancer masculin par état avec la table USCancerRates

Variables conditionnelles

Créer un nuage de points `rate.female*rate.male` par état (variable state) (ajouter une droite $y=x$ sur chaque graphique (options `grid` et `abline`))

Créer deux histogrammes pour rate.female et rate.male

(NB :

- Explicit: $y \sim x \mid g$ or $\sim x \mid g$
- More than one conditioning variable also allowed: $\sim x \mid g1 + g2$
- Implicit: $\sim x1 + x2$

)

Des options de la fonction histogram()

outer = TRUE | FALSE permet de contrôler comment les variables séparées par un + doivent être interprétées.

layout= c(ncols,nrows,npages) contrôle l'arrangement des graphiques

Pour s'entraîner et aller plus loin avec le package lattice – quelques exemples de prise en main

<https://www.statmethods.net/advgraphs/trellis.html>

<https://data-flair.training/blogs/r-lattice-package/>

Exercice 3 - package ggplot2

Voir feuille exercice ggplot2

Pour s'entraîner et pour aller plus loin

Le package patchwork : utile pour gérer plusieurs visualisations :

<https://www.data-imaginist.com/2019/patch-it-up-and-send-it-out/>

Exercice 4 – package esquisse

Installer et charger le package, lire <https://github.com/dreamRs/esquisse> pour prise en main. Tester sur la table mtcars du package ggplot2

Exercice 5 – package plotly

Utiliser le package plotly le plus simplement possible (en transformant un graphique réalisé avec le package ggplot2 en graphique interactif)

Exercice 6 – package corrplot

A partir de la fiche corplot.html, reproduire les exemples sur la table de votre choix

Liens intéressants pour étudier des exemples pouvant servir

Tutoriel pour réaliser des graphiques rainclouds plots : <https://wellcomeopenresearch.org/articles/4-63>

Pour customiser davantage sa palette de couleur, le package wesanderson : <https://epijim.uk/code-snippets/wes-anderson/>

Cours Yann Holtz pour s'entraîner avec des exercices supplémentaires : <https://www.yannholtz.com/teaching>

Exemple OECD Data

<https://data.oecd.org/unemp/unemployment-rate.htm>

Découvrez les outils pour faire de la datavisualisation

<https://graphism.fr/decouvrez-tous-les-outils-pour-faire-de-la-datavisualisation/>