

TP3 : Représentation graphique des données avec ggplot2



Objectifs du TP

Dans ce TP, vous allez apprendre à **représenter graphiquement des données** à l'aide du package **ggplot2**.

Vous explorerez différents types de graphiques, la personnalisation et la mise en page.

Vous pouvez consulter les cheat sheets de posit et datacamp sur ggplot2 pour une référence rapide :

[Posit \(Rstudio\)](#)

[DataCamp](#)

1. Introduction à ggplot2

Le package **ggplot2** repose sur la *Grammar of Graphics* : chaque graphique se construit en ajoutant des couches.

Exemple simple

```
library(ggplot2)

data(mpg)
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point()
```

Exercice 1

1. Remplacez `geom_point()` par `geom_line()`.
 2. Quelle différence observez-vous ? Est ce que ce type de representation graphique est approprié pour ce type de données ?
-

2. Graphiques de base : distributions

Histogramme

```
ggplot(mpg, aes(x = hwy)) +
  geom_histogram(binwidth = 2, fill = "skyblue", color = "black")
```

Densité

```
ggplot(mpg, aes(x = hwy, fill = class)) +  
  geom_density(alpha = 0.5)
```

Exercice 2

1. Modifiez le paramètre `binwidth` pour affiner ou élargir les classes.
 2. Tracez la densité de `displ` selon `drv`.
 3. Comparez les deux approches (histogramme vs densité).
-

3. Comparaison entre groupes

Boxplot

```
ggplot(mpg, aes(x = class, y = hwy)) +  
  geom_boxplot(fill = "lightgreen") +  
  coord_flip()
```

Exercice 3

1. Créez un boxplot de `cty` selon `drv`.
2. Ajoutez la moyenne sur le boxplot à l'aide de :

```
stat_summary(fun = mean, geom = "point", shape = 18, color = "red", size = 3)
```

4. Relations entre variables numériques

Nuage de points avec ligne de tendance

```
ggplot(mpg, aes(x = displ, y = hwy, color = class)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE)
```

Courbe d'évolution (exemple : economics)

```
ggplot(economics, aes(x = date, y = unemploy)) +  
  geom_line(color = "steelblue")
```

Exercice 4

1. Créez un scatter plot entre `displ` et `cty`.
 2. Ajoutez une tendance non linéaire avec `geom_smooth(method = "loess")`.
 3. Comparez avec la tendance linéaire précédente.
-

5. Personnalisation et thèmes

Titres, axes et thèmes

```
ggplot(mpg, aes(x = displ, y = hwy, color = class)) +  
  geom_point() +  
  labs(  
    title = "Relation entre la cylindrée et la consommation",  
    subtitle = "Jeu de données mpg",  
    x = "Cylindrée (litres)",  
    y = "Consommation (miles/galon)",  
    color = "Catégorie"
```

```
) +  
theme_minimal()
```

Facettes

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  facet_wrap(~ class)
```

Exercice 5

1. Essayez différents thèmes (`theme_light()`, `theme_classic()`, `theme_bw()`).
 2. Créez un graphique avec facettes selon `drv` et `class` (`facet_grid(drv ~ class)`).
-

6. Autres visualisations

Diagramme en barres

```
ggplot(mpg, aes(x = class, fill = drv)) +  
  geom_bar(position = "dodge")
```

Diagramme circulaire (coordonnées polaires)

```
ggplot(mpg, aes(x = "", fill = class)) +  
  geom_bar(width = 1) +  
  coord_polar("y", start = 0)
```

7. Application sur iris

Jeu de données iris

```
data(iris)
head(iris)
```

Exercice 6

1. Créez un histogramme de `Sepal.Length`.
 2. Créez un boxplot de `Petal.Length` selon `Species`.
 3. Créez un nuage de points de `Sepal.Length` vs `Petal.Length`, coloré par `Species`.
 4. Ajoutez un titre, une légende claire et un thème personnalisé.
-