

Reproduisez ces graphiques

Reproduisez les graphiques sur cette page

- Chargez la librairie tidyverse
- Pour tout autre package autre que tidyverse, assurez vous de les installer avant d'utiliser la librairie. Ex. `install.packages(gapminder)`

```
```{r}
library(tidyverse)
library(nycflights13)
library(gapminder)
library(datasets)

ou

pacman::p_load(tidyvers, nycflights13, gapminder, datasets)
```
```

Indication de style

Vous pouvez commencer votre graphique avec le nom de votre jeu de données et la “pipe function” `%>%` ça permet d’être plus flexible lorsqu’on veut faire des modifications avec les autres packages du tidyverse.

```
ggplot(iris, aes(Sepal.Length)) +
  geom_density()

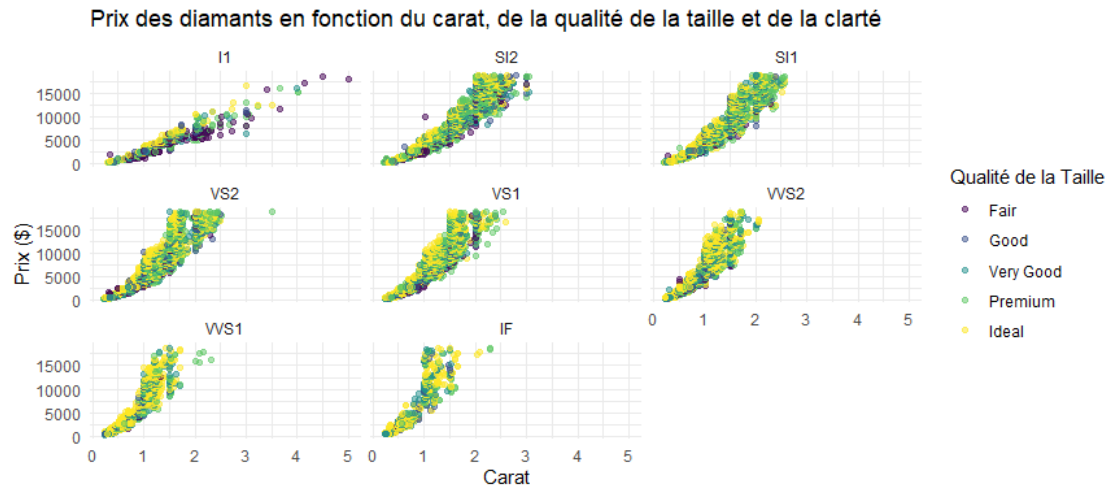
iris %>%
  ggplot(aes(Sepal.Length)) +
  geom_density()
```

Graphique 1: Diamonds (“diamonds” dataset)

Données: `data(diamonds)`

Variables utilisées:

- **carat:** Carat
- **price:** Prix en dollar
- **cut:** Qualité de la taille
- **clarity:** I1, SI2, SI1, VS1, VS2, etc.



Graphique 2: Miles par Gallon ("mpg" dataset)

Données: `data(mpg)`

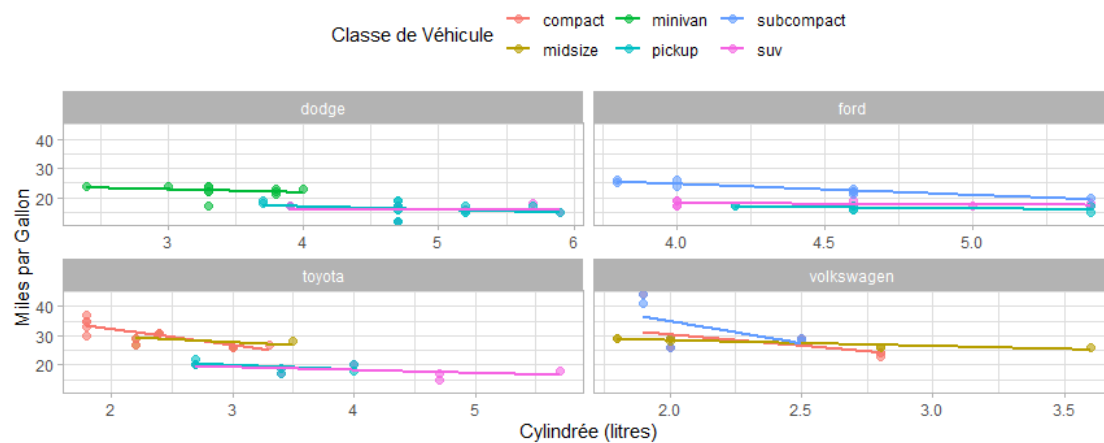
Variables:

- **displ**: Cylindrée en litre
- **hwy**: Miles par Gallon
- **class**: Classe de véhicule
- **manufacturer**: dodge, ford, toyota, volkswagen, etc.

Nettoyage:

- `filter(manufacturer %in% c("dodge", "ford", "toyota", "volkswagen"))`

Relation entre la cylindrée et la consommation sur autoroute



Graphique 3: Vols (flights dataset)

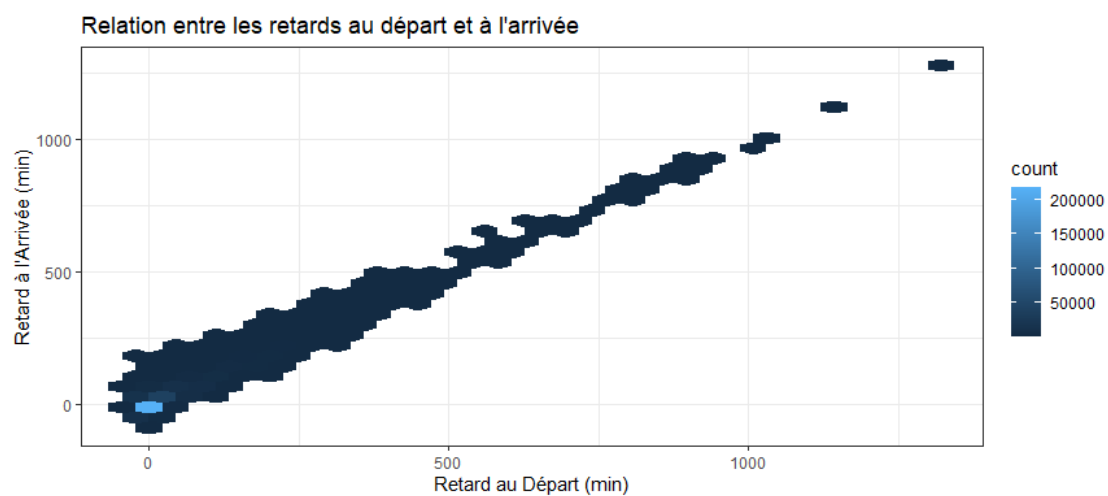
Données: `library(nycflights13)` et `data(flights)`

Variables:

- **dep_delay:** Retard au départ en minutes
- **arr_delay:** Retard à l'arrivée en minutes

Nettoyage:

- `tidyr: drop_na(dep_delay, arr_delay)`
ou
- `dplyr: filter(!is.na(dep_delay), !is.na(arr_delay))`

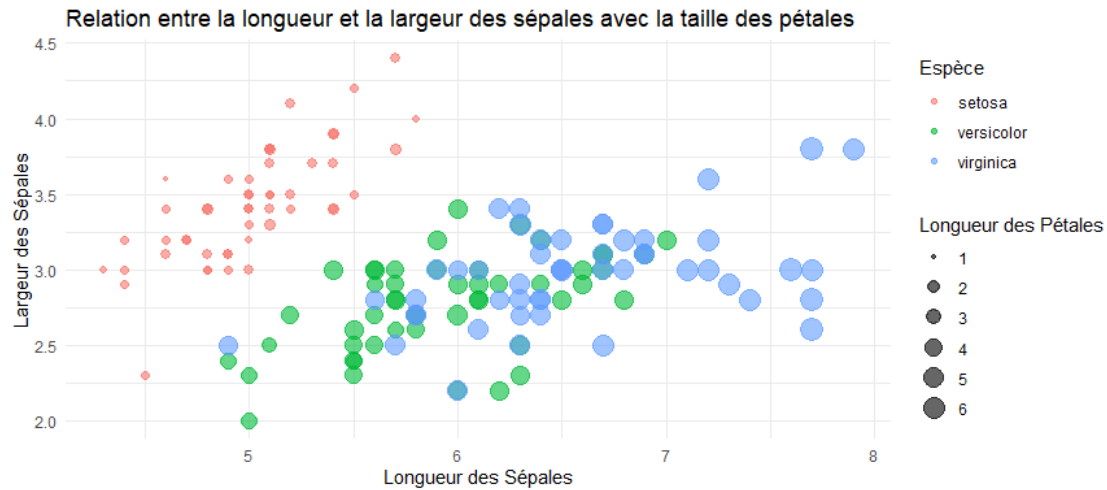


Graphique 4: Iris

Données: `iris data(iris)`

Variables:

- **Sepal.Length:** Longueur du sépale
- **Sepal.Width:** Largeur du sépale
- **Species:** Espèce de la fleur
- **Petal.Length:** Longueur de la pétale

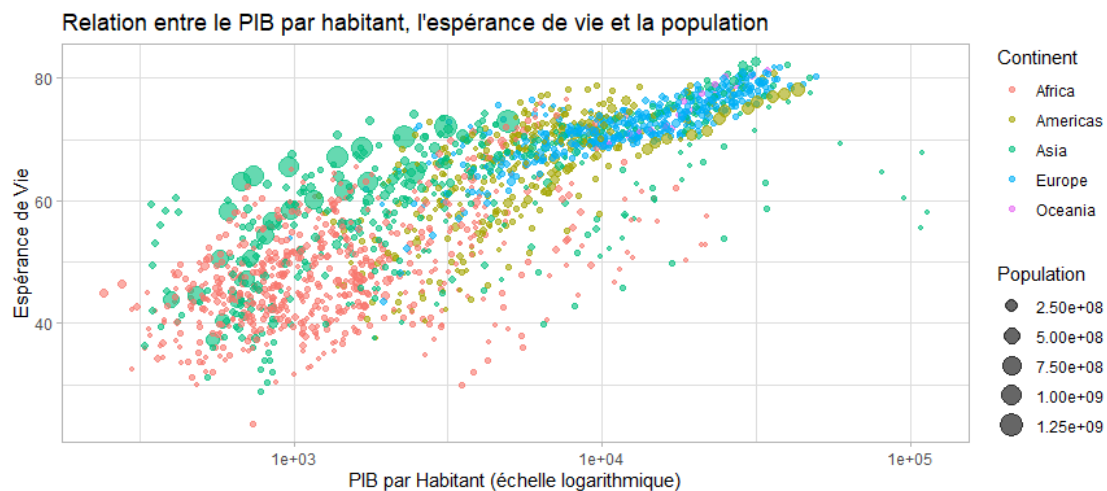


Graphique 5: Gapminder (nécessite le package gapminder)

Données: `library(gapminder)` et `data(gapminder)`

Variables:

- **gdpPercap**: PIB par habitant
- **lifeExp**: Espérance de vie
- **pop**: Taille de la population
- **continent**: Continent



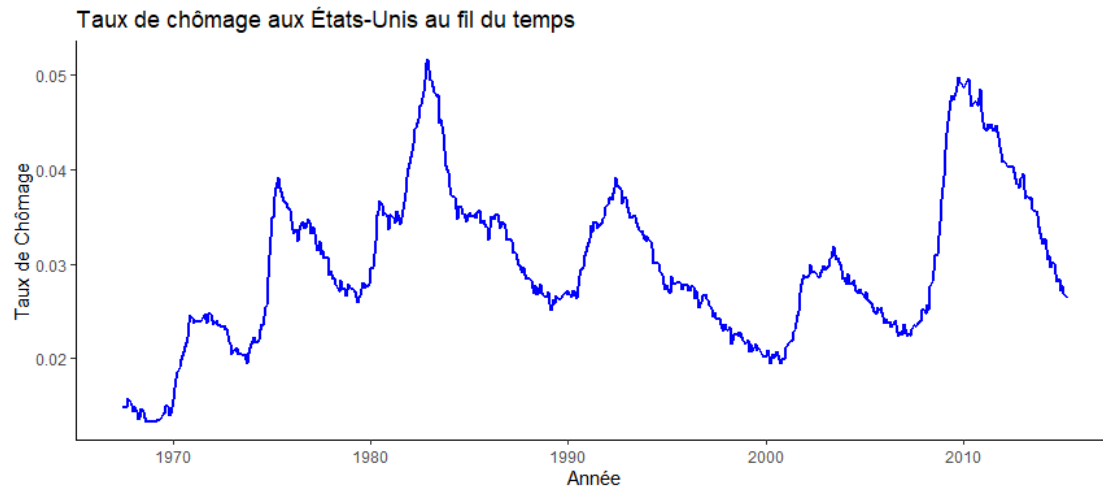
Graphique 6: Économie

Données: `data(economics)`

Variables:

- **date**: Date

- **unemploy:** Taux de chômage
- **pop:** Taille de la population



Graphique 7: txhousing

Données: `data(txhousing)`

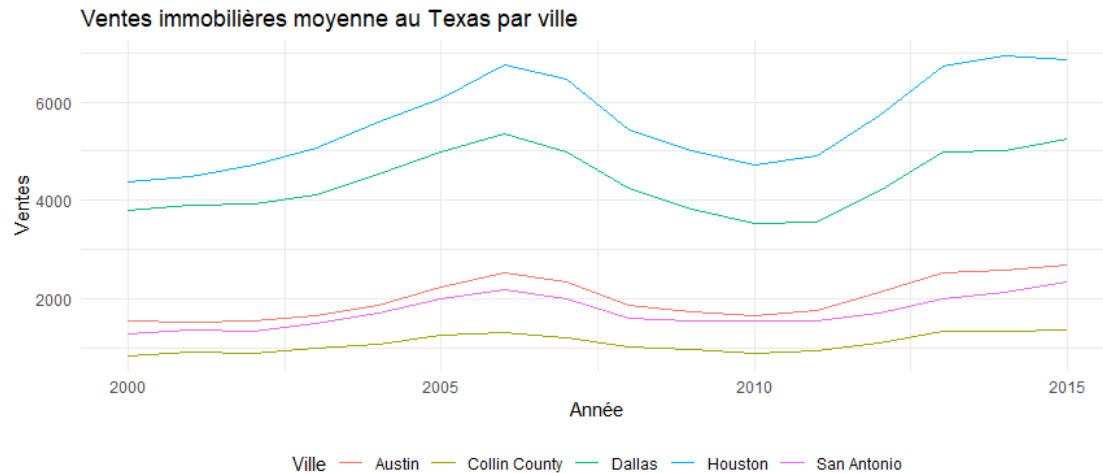
variables:

- **year:** Années
- **sales:** Ventes immobilières au Texas
- **city:** Villes

Nettoyage:

- Partie 1: `filter(city %in% c("Austin", "Collin County", "Dallas", "Houston", "San Antonio"))`
- Partie 2: `summarise(sales = mean(sales), .by = c(city, year))`

Note: Il est nécessaire de faire la moyenne des vente par an.



Graphique 8: Worldphones (du package datasets)

Données: `library(datasets)` et `data(WorldPhones)` et `as_tibble(WorldPhones)`

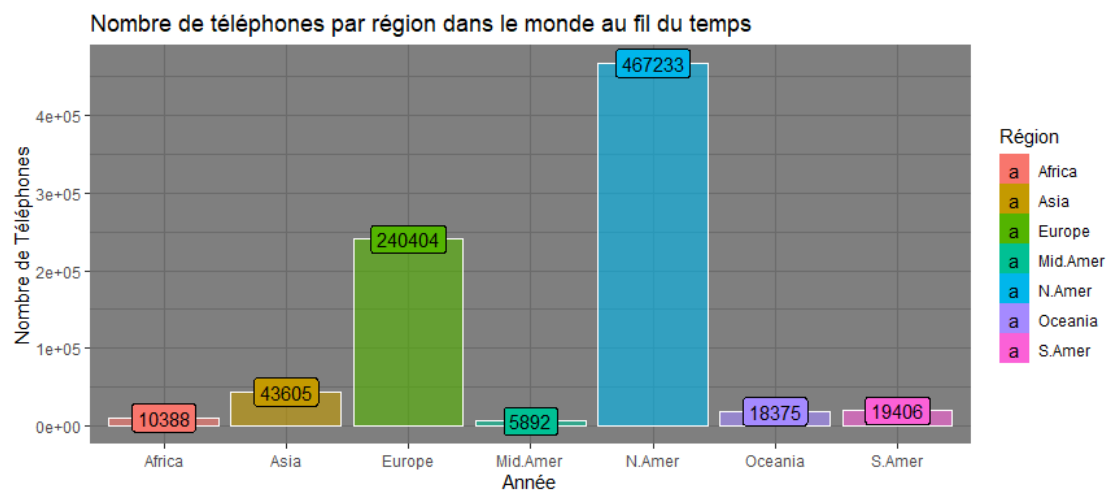
Variables:

- **region:** Régions
- **n_phones:** Nombre de téléphones

Nettoyage:

- Partie 1:

Note: Il est nécessaire de faire la somme du nombre de téléphone par région

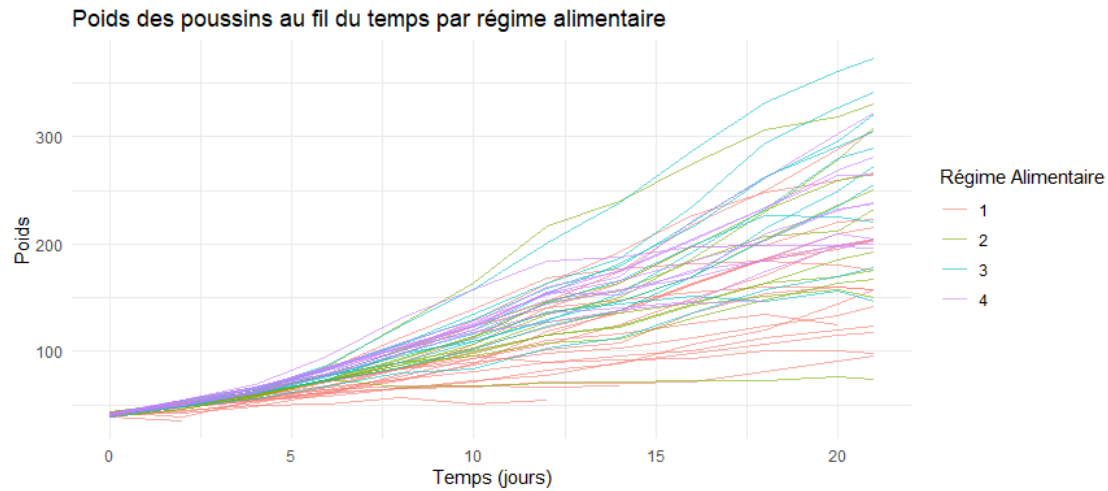


Graphique 9: ChickWeight (du package datasets)

Données: `library(datasets)` et `data(ChickWeight)`

Variables:

- **Time:** Temps en jour
- **weight:** Poids
- **Chick:** Type de poussin
- **Diet:** Régime alimentaire



Graphique 10: mtcars

Données: data(mtcars)

Variables:

- **wt:** Poids du véhicule
- **mpg:** Miles par Gallon
- **gear:** Engrenage
- **qsec:** Quart de Mile en seconde

