

# CARTOGRAPHIE

## AIDE MÉMOIRE

La bibliothèque *GeoPandas* est une bibliothèque qui permet de travailler avec des données cartographiques.

Pour charger la bibliothèque *GeoPandas* :

```
import geopandas as gpd
```

La bibliothèque *GeoVizir* propose différentes fonctions pour faciliter la réalisation de cartes statistiques.

Pour charger la bibliothèque *GeoVizir* :

```
import geovizir as gvz
```

### Charger les données géographiques

```
# naturalearth_lowres est un fichier avec des données géographique pour les pays du monde, ce
fichier fait parti des *datasets* (sets de données) inclus dans GeoPandas
df_geo = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
```

### Charger des données statistiques

La fonction `get_data()` de la bibliothèque *GeoVizir* permet de récupérer des données statistiques depuis la Banque Mondiale <https://donnes.banquemonde.org>. L'argument `indicator` permet de spécifier l'identifiant de l'indicateur ; l'argument `year` permet de spécifier l'année.

```
df_stat = gvz.get_data(indicator='SP.POP.TOTL', year=2020) # Récupère les données de la population
par pays pour 2020
```

### Combiner les données

Les données géographiques peuvent être combinées avec d'autres tableaux en utilisant la méthode `merge()`. On appelle cette combinaison une *jointure*. Pour combiner deux tableaux, une des colonnes de chaque tableau doit contenir une information identique ; on appelle ça la *clé de jointure*. L'argument `how` permet de spécifier le sens de la jointure ; l'argument `left_on` permet de spécifier le nom de la colonne qui servira de clé de jointure dans le tableau de gauche ; idem pour le tableau de droite avec `right_on`.

```
df_map = df_geo.merge(df_stat, how = "left", left_on='iso_a3', right_on='iso3c')
```

### Faire une carte

Pour faire une carte à partir d'un tableau *GeoPandas* on peut utiliser la méthode `plot()`. L'argument `column` permet d'indiquer quelle colonne contient l'indicateur à utiliser. L'argument `legend` permet de contrôler l'affichage de la légende.

```
df_map.plot(column='value', legend=True)
```