

# Atelier Quarto - Exercice 2

Guillaume Chrétien

2024-11-20

## Table des matières

Chargement des librairies .....	2
Des packages bien utiles... .....	2
Le détail du code .....	2
Chargement d'un json et sélection de colonnes .....	2
Affichage d'un tableau .....	3
Un premier graphe .....	6
Cartes .....	7
Une première carte pour vérifier les géométries .....	7
Une carte en aplat de couleurs .....	8
Figures sur 2 colonnes .....	9
Mélanger texte et résultats avec du inline .....	10
Quelques explications sur typst .....	10
Conclusion .....	11

## Chargement des librairies

### Des packages bien utiles...

Un premier bloc de code (chunk en anglais) est utilisé pour charger les librairies R qui seront utilisées dans les blocs de code suivant.

### Le détail du code

```
library(dplyr) # package pour manipuler les données
library(ggplot2) # package pour les graphes
library(sf) # package géographique de R
library(knitr)
```

## Chargement d'un json et sélection de colonnes

R peut lire tous les formats, le geojson est un classique. R permet ensuite de manipuler ce fichier geojson, par exemple, sélectionner quelques colonnes et supprimer la colonne géométrie.

```
#### Fonction st_read du package sf pour lire un geojson
communes_mrn <- st_read("data/communes_mrn.geojson")

#### Utilisation du vocabulaire de dplyr pour sélectionner quelques colonnes
communes_mrn_qqs_colonnes <- communes_mrn %>%
  st_drop_geometry() %>%
  select(nom, population, surface_ha)
```

## Affichage d'un tableau

Et voici un affichage sous forme de tableau adapté à des rendus pdf ou docx avec la fonction `kable`

```
#### Affichage du tableau de donnée, oui c'est juste une ligne...  
kable( communes_mrn_qqs_colonnes )
```

nom	popula- tion	surface_ha
Elbeuf	16087	1632
Yainville	1037	331
Hénouville	1366	1069
Canteleu	13807	1761
Saint-Étienne-du-Rouvray	28331	1825
Petit-Couronne	8732	1280
Isneauville	3601	820
Jumièges	1693	1875
Saint-Jacques-sur-Darnétal	3108	1671
Fontaine-sous-Préaux	558	352
Maromme	10845	401
Freneuse	960	318
Bihorel	8197	251
Le Houlme	4149	297
Darnétal	9780	493
Orival	881	955
Tourville-la-Rivière	2562	800
Grand-Couronne	9731	1691
Les Authieux-sur-le-Port-Saint-Ouen	1254	453
Le Petit-Quevilly	21997	435
Bonsecours	6466	376
Caudebec-lès-Elbeuf	9996	368
Moulineaux	922	349
Notre-Dame-de-Bondeville	6966	628

nom	popula- tion	surface_ha
Saint-Martin-du-Vivier	1675	500
Yville-sur-Seine	437	825
Cléon	4919	647
Saint-Léger-du-Bourg-Denis	3617	281
Épinay-sur-Duclair	510	661
Le Mesnil-sous-Jumièges	617	684
Berville-sur-Seine	544	701
Saint-Paër	1307	1836
Quévreville-la-Poterie	1029	468
Le Mesnil-Esnard	7925	507
Hautot-sur-Seine	396	216
Rouen	114187	2138
Malaunay	6160	925
La Londe	2351	3098
Ymare	1182	403
Saint-Pierre-de-Manneville	885	1021
Anneville-Ambourville	1176	2033
Sotteville-sous-le-Val	749	527
Déville-lès-Rouen	10644	316
Val-de-la-Haye	718	1016
Mont-Saint-Aignan	19686	794
Sotteville-lès-Rouen	29071	744
Belbeuf	2242	651
Saint-Aubin-lès-Elbeuf	8428	579
Roncherolles-sur-le-Vivier	1212	535
Bardouville	622	861
Saint-Pierre-lès-Elbeuf	8251	636
Franqueville-Saint-Pierre	6099	856
Gouy	898	497

nom	popula- tion	surface_ha
Quevillon	593	1123
Houpeville	2918	2080
La Bouille	707	127
Saint-Aubin-Épinay	1020	983
Sahurs	1212	1123
Le Trait	4831	1752
Amfreville-la-Mi-Voie	3299	394
Montmain	1397	604
Saint-Aubin-Celloville	1180	672
Le Grand-Quevilly	26034	1111
Bois-Guillaume	14378	885
Saint-Pierre-de-Varengeville	2290	1318
Boos	3990	1403
Sainte-Marguerite-sur-Duclair	2027	726
La Neuville-Chant-d'Oisel	2378	2183
Oissel	12266	2219
Duclair	4010	1002
Saint-Martin-de-Boscherville	1536	1291

## Un premier graphe

Ensuite, il est assez simple de faire un graphe avec la grammaire de base de ggplot (si la syntaxe vous paraît complexe, cf extension « esquisse »).

```
#### Difficile en pdf ou docx de représenter des graphes aussi complet qu'en HTML
#### Limitons nous aux 10 communes les plus peuplées
communes_mrn_top_10_pop <- communes_mrn_qqs_colonnes %>%
  arrange(desc(population)) %>%
  slice_head(n=10)

ggplot(communes_mrn_top_10_pop, aes(x = reorder(nom, population), y =
population)) +
  geom_bar(stat = "identity", fill = "grey", color = "grey") +
  coord_flip() +
  labs(
    x = "Région",
    y = "Population"
  ) +
  theme_minimal()
```

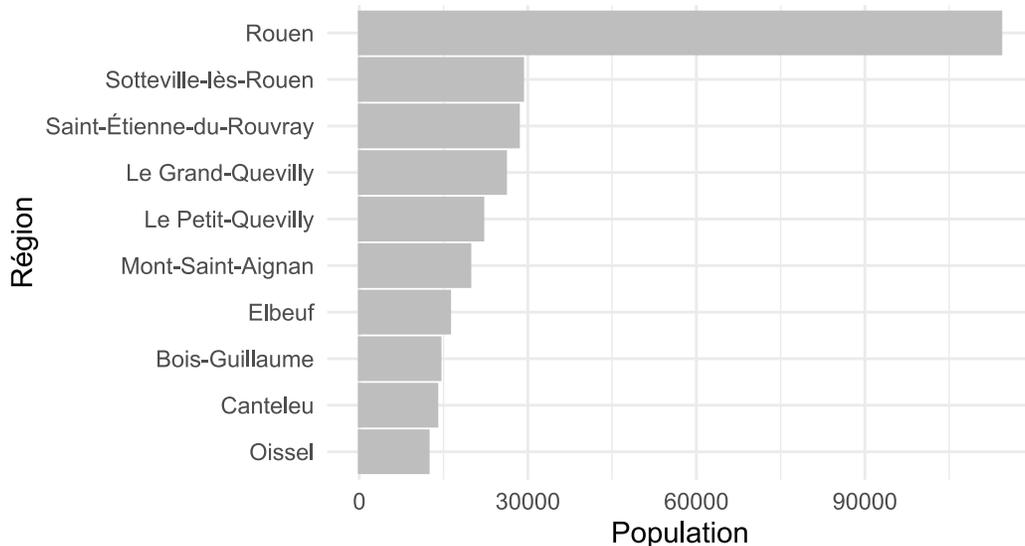


Figure 1. – Nombre d'habitants des 10 communes de la Métropole Rouen Normandie

## Cartes

### Une première carte pour vérifier les géométries

```
#### Au préalable, ici je reprojette mais ça n'est pas nécessaire  
communes_mrn <- st_transform(communes_mrn, crs = 4326)  
  
#### Ajout d'une carte leaflet  
ggplot(data = communes_mrn) +  
  geom_sf() +  
  theme_void()
```

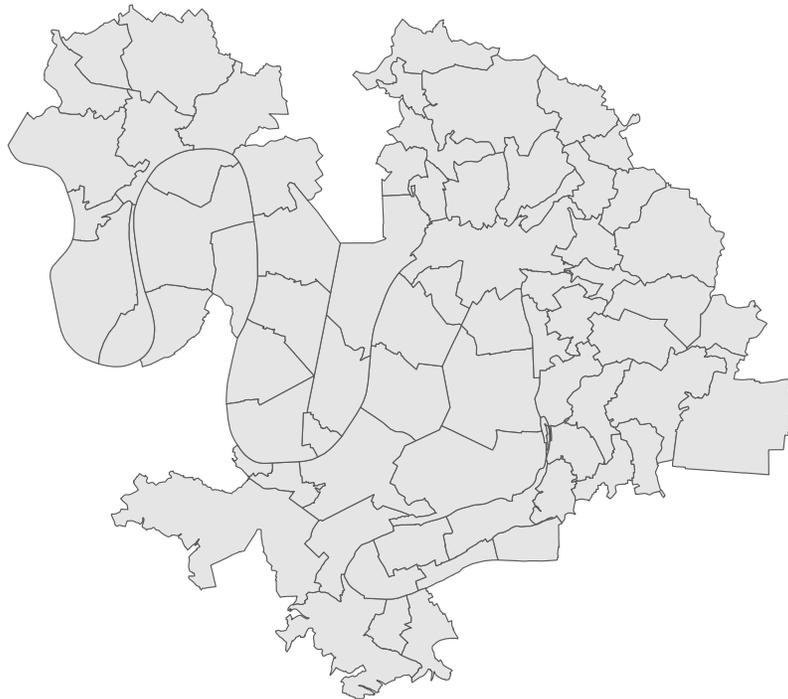


Figure 2. – Géométrie des communes de la Métropole Rouen Normandie

## Une carte en aplat de couleurs

```
ggplot(data = communes_mrn) +  
  geom_sf(aes(fill = population)) +  
  scale_fill_gradient(low = "white", high = "darkgreen") +  
  theme_void() +  
  theme(  
    legend.position = "right" # Place la légende à droite  
  )
```

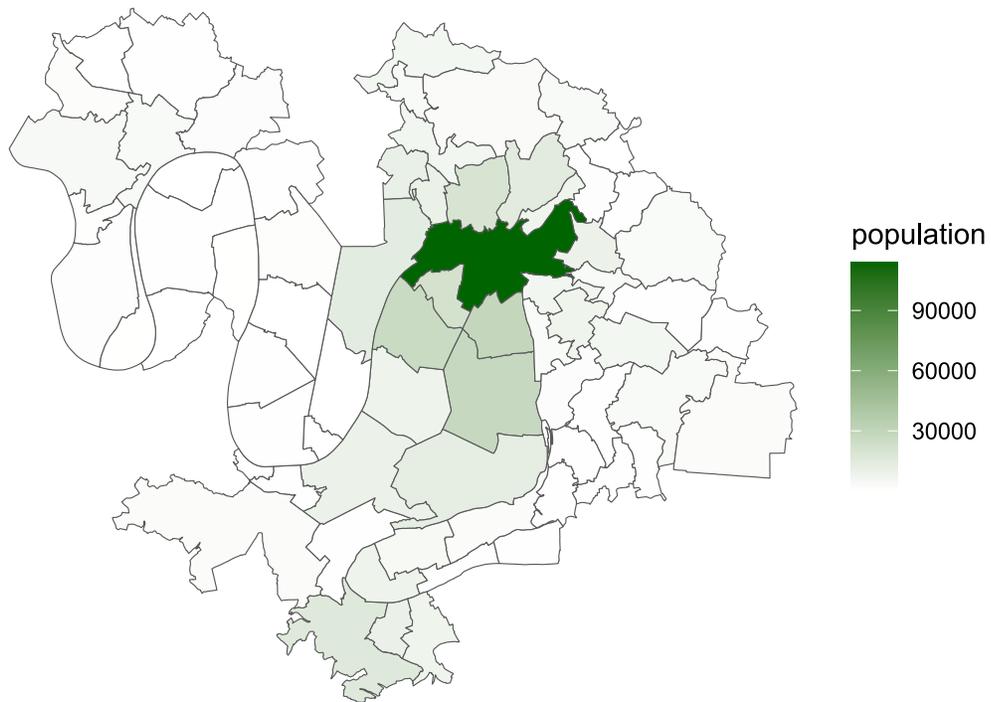


Figure 3. – Population des communes de la Métropole Rouen Normandie

## Figures sur 2 colonnes

En paramétrant les options d'exécution d'un bloc de code, il est assez simple de jouer sur la mise en page : <https://quarto.org/docs/authoring/cross-references.html#computations>. Ici dans les options du bloc de code il a été précisé :

```
#| echo: true
#| fig-cap: "Les communes de la Métropole Rouen Normandie"
#| fig-subcap:
#|   - "Géométrie"
#|   - "Population"
#| layout-ncol: 2
```

```
ggplot(data = communes_mrn) +
  geom_sf() +
  theme_void()

ggplot(data = communes_mrn) +
  geom_sf(aes(fill = population)) +
  scale_fill_gradient(low = "white", high = "darkgreen") +
  theme_void() +
  theme(
    legend.position = "right" # Place la légende à droite
  )
```



Fig. 1. – Géométrie

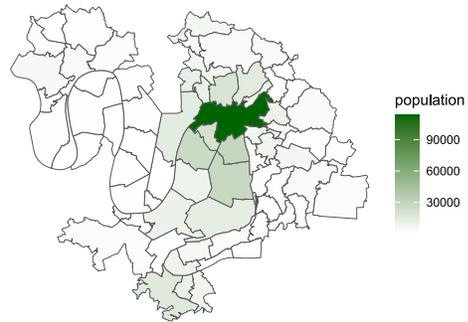


Fig. 2. – Population

Les communes de la Métropole Rouen  
Normandie

## Mélanger texte et résultats avec du inline

Il est aussi possible d'insérer des résultats dans le texte. Voici un bloc de code pour filtrer la population de la commune de rouen et formater le nombre avec séparateur de milliers.

```
pop_rouen <- communes_mrn %>%  
  st_drop_geometry() %>%  
  filter(insee_com=='76540') %>%  
  select(population) %>%  
  format(big.mark = " ")
```

Rouen compte 114 187 habitants.

Dans cette phrase, le nombre d'habitants est issu directement du code grâce à la syntaxe inline code.

## Quelques explications sur typst

La doc du site Quarto sur typst laisse entrevoir des possibilités assez sympa même si il y a encore quelques bugs non réglés (cf atelier avec pdf généré en latex si vous rencontrez des problèmes).

Exemple de mise en page générée avec un formatage propre à typst et avec une condition pour apparaître en pdf (attention bien mettre ="typst") mais pas en html :

This is a block with gray background and slightly rounded corners.

## Conclusion

L'en-tête YAML du document précisait :

```
format:
  html:
    theme: [flatly, custom.scss]
    embed-resources: true
  typst:
    toc: true
  docx:
    reference-doc: custom-reference-doc.docx
```

En tapant `quarto render` dans le terminal, les 3 formats sont générés ! Très pratique, mais il faut avoir en tête qu'il n'y a pas de miracle :

- il faut un peu tâtonner pour trouver la bonne mise en page pdf, il est complexe de personnaliser le look
- il faut revenir manuellement sur la mise en page .docx mais il est assez facile de personnaliser le look pour adopter une charte graphique... et il est ensuite possible de générer un pdf classiquement via exporter en pdf.