Exercice 4

Gabriel Aubin-Moreau

2022-09-14

Tâche 1 : Réflexion

Pour les comptes:

Il faut porter une attention particulière à l'organisation de nos graphiques à barres selon le contexte. Par exemple, lorsque j'ai plusieurs catégories qui n'ont pas d'ordre naturel. Il faut mettre les barre en ordre décroissant ou croissant pour permettre une meilleure lecture. Si jamais on a beaucoup de catégories, on peut utiliser un tracé de point ou dot plot mais, on peut aussi utiliser une carte de chaleur ou heatmap.

Pour les proportions: J'ai compris en lisant que les avis sur les diagrammes circulaire ou pie chart était partagé. Cependant, si on les utilise dans le bon contexte, ils peuvent nous être très utile. Par exemple, si on veut visualiser des données faisant partie d'un tout. Le diagramme circulaire peut se servire utile. Sa meilleure utilisation est lorsqu'on veut visualiser des fractions simples comme 1/2, 1/3 ou 1/4. Une excellente alternative aux diagramme circulaire est le graphique gaufré ou waffle chart. Qui permet grâce aux carrés qui représente une unité d'évaluer facilement la taille d'une catégorie.

Tâche 2 : Construction pandémique essentielle

Charger et nettoyer les données

Nous chargeons et nettoyons d'abord les données (téléchargées en mai 2020, vous pouvez aller charger des données plus récentes!):

```
# Vous n'aurez besoin que de la bibliothèque tidyverse pour cet exercice
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
# Charger les données d'origine
essential_raw <- read_csv("data/EssentialConstruction.csv")
```

```
## Rows: 7209 Columns: 7

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): ADDRESS, BOROUGH, CATEGORY, SUBCATEGORIES, JOB NUMBERS
## dbl (2): BIN, CD

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

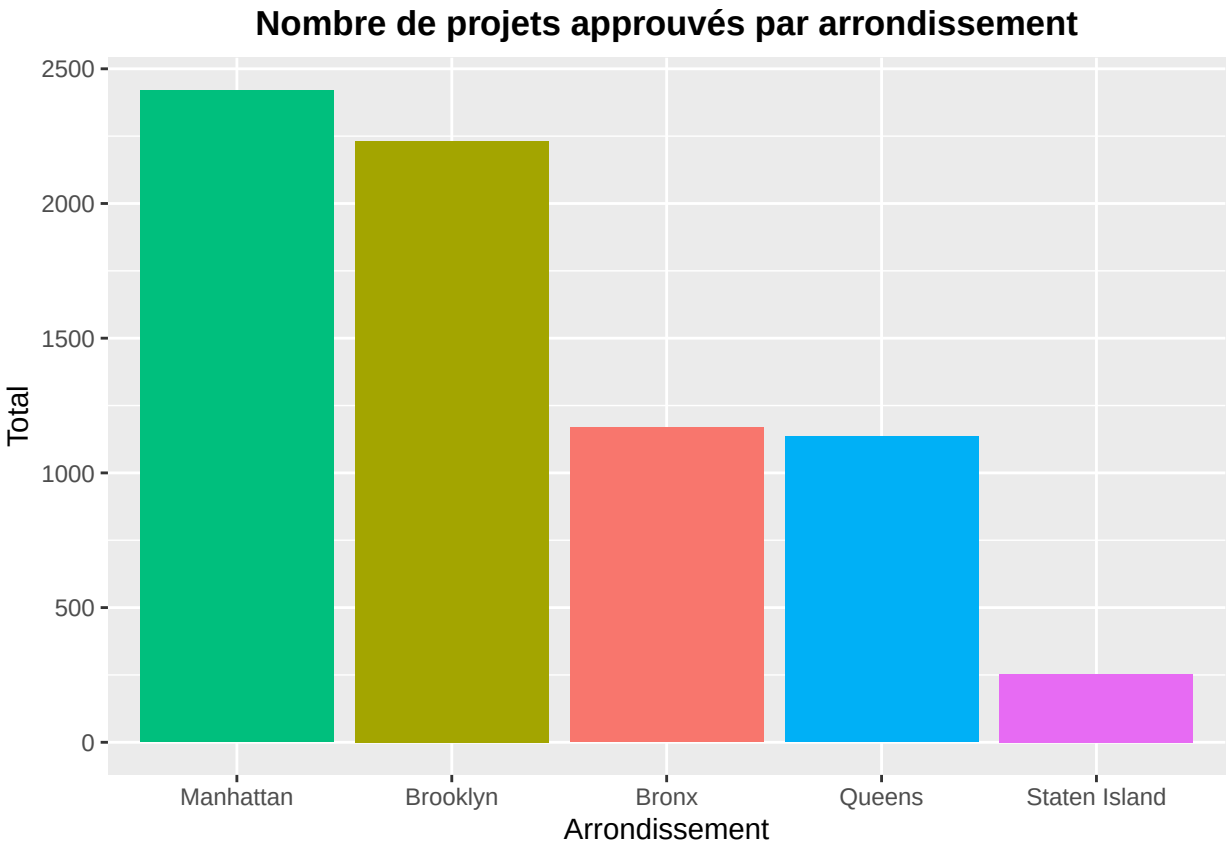
```r
# Nettoyer un peu les données
# Certains noms d'arrondissements sont en MAJUSCULES, nous utilisons donc str_to_title() pour convertir
# tout dans la colonne
# Nous faisons également des facteurs ARRONDISSEMENT et CATÉGORIE (ou variables catégorielles)
essentiel <- essential_raw %>%
  mutate(BOROUGH = str_to_title(BOROUGH),
         BOROUGH = factor(BOROUGH),
         CATEGORY = factor(CATEGORY))
```

Projets approuvés par arrondissement

À l'heure actuelle, il y a une ligne pour chaque chantier de construction approuvé. Nous devons condenser cela pour obtenir le nombre de chantiers de construction selon différentes variables. Nous pouvons le faire en utilisant `group_by()` et `summarize()`

```r
essentiel_par_arrondissement <- essentiel %>%
  group_by(BOROUGH) %>%
  summarise(total = n()) %>%
  mutate(proportion = total / sum(total))
```

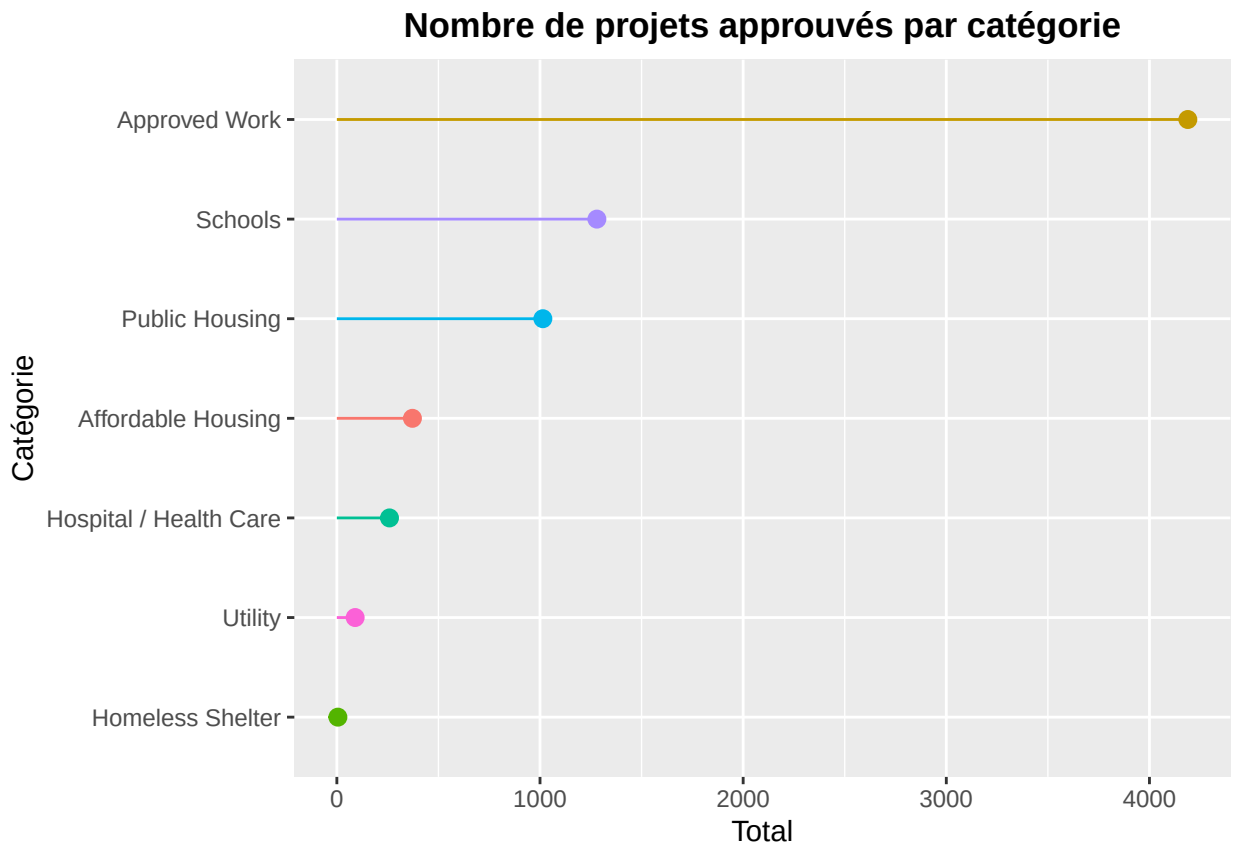
```r
ggplot(essentiel_par_arrondissement,
       mapping=aes(reorder(BOROUGH, -total), total, fill=BOROUGH)) +
  geom_col() +
  guides(fill = "none") +
  labs(title="Nombre de projets approuvés par arrondissement", y="Total", x="Arrondissement") +
  theme(plot.title=element_text(face="bold", hjust=0.5))
```



Projets approuvés par catégorie

```
essentiel_par_categorie <- essentiel %>%
  group_by(CATEGORY) %>%
  summarise(Total = n()) %>%
  mutate(Proportion = Total / sum(Total))
```

```
ggplot(essentiel_par_categorie,
       mapping = aes(Total, reorder(CATEGORY, Total), color = CATEGORY)) +
  geom_pointrange(aes(xmin = 0, xmax = Total)) +
  guides(color = "none") +
  labs(title = "Nombre de projets approuvés par catégorie", y = "Catégorie") +
  theme(plot.title=element_text(face="bold", hjust=0.5))
```

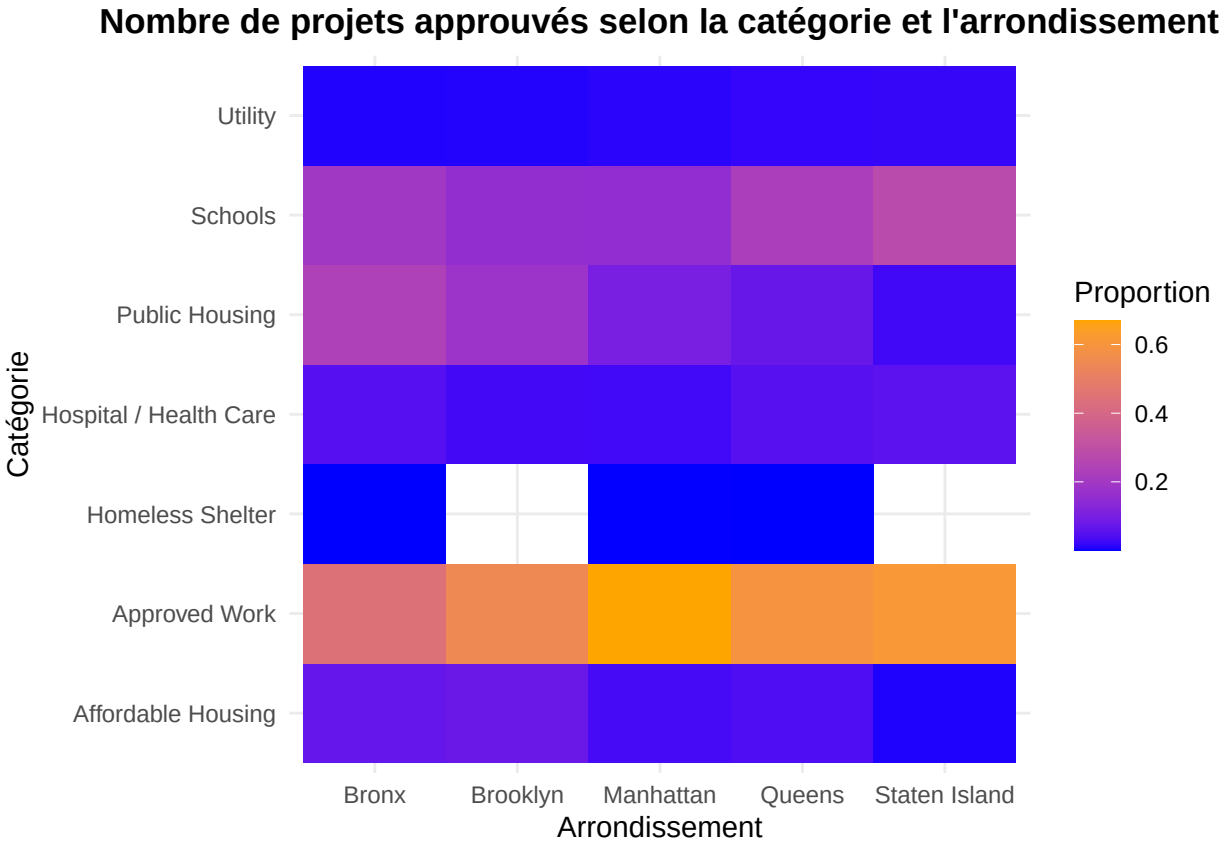


Projets approuvés dans l'arrondissement et la catégorie

```
essentiel_par_categorie_arrondissement <- essentiel %>%
  group_by(CATEGORY, BOROUGH) %>%
  summarise(Total = n()) %>%
  group_by(BOROUGH) %>%
  mutate(Proportion = Total / sum(Total))
```

```
## `summarise()` has grouped output by 'CATEGORY'. You can override using the `.groups` argument.
```

```
ggplot(essentiel_par_categorie_arrondissement,
       mapping = aes(BOROUGH, CATEGORY,  fill = Proportion)) +
  geom_tile() +
  scale_fill_gradient(low = "Blue", high = "Orange") +
  labs(title = "Nombre de projets approuvés selon la catégorie et l'arrondissement",
       x = "Arrondissement", y = "Catégorie") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5))
```


Nombre de projets approuvés selon la catégorie et l'arrondissement
Utility
Schools
Public Housing
Hospital / Health Care
Homeless Shelter
Approved Work
Affordable Housing
Catégorie
Bronx
Brooklyn
Manhattan
Queens
Staten Island
Arrondissement
Proportion
0.6
0.4
0.2