dev2\_ComptesProportion

Mamadou Billo Diallo

2023-09-15

# Tâche 1 : Réflexion

Mettez votre réflexion ici des lectures sur les comptes et proportions il est important d’utiliser comptes et les proportions dans nos visualisations car elles nous permettent de visualisé les données en entier sans ambiguité.les comptes peuvent aider à résumer des données brutes en comptant le nombre d’occurrences de chaque catégorie ou élément. pour une analyse descriptive Ils permettent de répondre à des questions simples telles que “Combien de fois cela s’est-il produit ?” Les proportions sont aussi comme des comptes exprimés en pourcentage par rapport à un total. Ils sont particulièrement utiles pour comparer des sous-groupes par rapport à l’ensemble des données. Par exemple, vous pouvez comparer la proportion de personnes préférant un certain type de nourriture dans différentes régions.

# 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)

## Warning: le package 'tidyverse' a été compilé avec la version R 4.2.2

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

# Charger les données d'origine  
essential\_raw <- read\_csv("data/EssentialConstruction.csv",  
 show\_col\_types = FALSE)  
  
# 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))

essential\_raw

## # A tibble: 7,209 × 7  
## ADDRESS BOROUGH BIN CD CATEGORY SUBCA…¹ JOB N…²  
## <chr> <chr> <dbl> <dbl> <chr> <chr> <chr>   
## 1 1074 WASHINGTON AVENUE Bronx 2129119 203 Affordable Hous… <NA> All jo…  
## 2 1115 EAST 165TH STREET Bronx 2129161 202 Affordable Hous… <NA> All jo…  
## 3 112 EAST CLARKE PLACE Bronx 2129183 204 Affordable Hous… <NA> All jo…  
## 4 1164 RIVER AVENUE Bronx 2130367 204 Affordable Hous… <NA> All jo…  
## 5 1193 FULTON AVENUE BRONX 2004216 203 Affordable Hous… <NA> All jo…  
## 6 1193 FULTON AVENUE Bronx 2129503 203 Affordable Hous… <NA> All jo…  
## 7 1201 SPOFFORD AVENUE BRONX 2005938 202 Affordable Hous… <NA> All jo…  
## 8 1230 METCALF AVENUE Bronx 2025407 209 Affordable Hous… <NA> All jo…  
## 9 1240 GERARD AVENUE Bronx 2103799 204 Affordable Hous… <NA> All jo…  
## 10 1325 JEROME AVENUE Bronx 2129224 204 Affordable Hous… <NA> All jo…  
## # … with 7,199 more rows, and abbreviated variable names ¹​SUBCATEGORIES,  
## # ²​`JOB NUMBERS`

## Projets approuvés par arrondissement

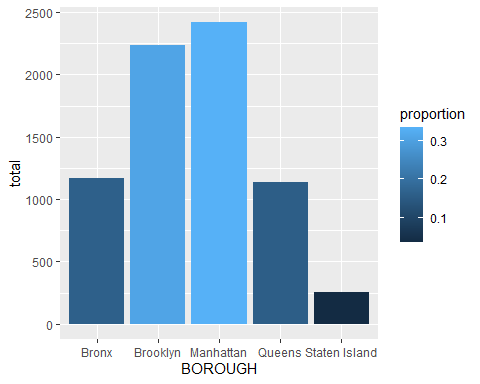
À l’heure actuelle, il y a une ligne pour chaque chantier de construction approuvé. Nous devons condenser cela pour obtenir le nombre de chantiers de construction selon différentes variables. Nous pouvons le faire en utilisant group\_by() et summarize()

essentiel\_par\_arrondissement <- essentiel %>%  
 group\_by(BOROUGH) %>%  
 summarise(total = n()) %>%  
 mutate(proportion = total / sum(total))

essentiel\_par\_arrondissement

## # A tibble: 5 × 3  
## BOROUGH total proportion  
## <fct> <int> <dbl>  
## 1 Bronx 1169 0.162   
## 2 Brooklyn 2231 0.309   
## 3 Manhattan 2420 0.336   
## 4 Queens 1135 0.157   
## 5 Staten Island 254 0.0352

# Ajouter un graphique avec geom\_col() ici  
ggplot(data = essentiel\_par\_arrondissement,mapping = aes(x=BOROUGH,y=total,fill=proportion))+  
 geom\_col(position=position\_dodge())



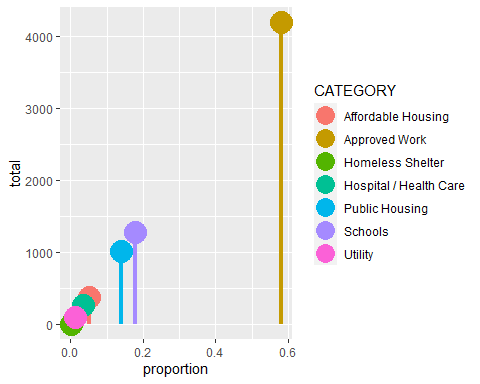
## Projets approuvés par catégorie

# Créer un jeu de données résumé des projets par catégorie  
#  
# Je ne vous donnerai pas le code pour cela (gros indice cependant : copiez le code pour le  
# résumé d'arrondissement et ne changer qu'une chose)  
essentiel\_par\_CATEGORY <- essentiel %>%  
 group\_by(CATEGORY) %>%  
 summarise(total = n()) %>%  
 mutate(proportion = total / sum(total))

essentiel\_par\_CATEGORY

## # A tibble: 7 × 3  
## CATEGORY total proportion  
## <fct> <int> <dbl>  
## 1 Affordable Housing 372 0.0516   
## 2 Approved Work 4189 0.581   
## 3 Homeless Shelter 5 0.000694  
## 4 Hospital / Health Care 259 0.0359   
## 5 Public Housing 1014 0.141   
## 6 Schools 1280 0.178   
## 7 Utility 90 0.0125

# Ajoutez un graphique en forme de sucette (lollipop) ici  
  
ggplot(data = essentiel\_par\_CATEGORY,  
 mapping = aes(x = proportion, y = total, color = CATEGORY)) +  
 geom\_pointrange(aes(ymin = 0, ymax = total),  
 # Rendre les lignes un peu plus épaisses et les points un peu plus gros  
 fatten = 5, size = 1.5)



## Projets approuvés dans l’arrondissement et la catégorie

# Créer un ensemble de données résumé des projets par arrondissement et par catégorie  
#  
# Je ne vous donnerai pas non plus le code pour faire le résumé de la heatmap. Vous avez  
# besoin de grouper par deux variables pour faire le résumé. IMPORTANT, vous aurez également  
# besoin d'ajouter un autre group\_by() entre summarise() et mutate(), sinon,  
# R calculera des pourcentages dans des groupes inattendus.  
#  
# Si vous voulez que les pourcentages des catégories s'additionnent à 100% dans chaque arrondissement,  
# vous voudrez grouper par arrondissement avant de calculer la proportion ; si tu veux  
# les pourcentages d'arrondissements à additionner jusqu'à 100% dans chaque catégorie, vous voudrez  
# groupe par catégorie  
  
  
essentiel\_par\_arrondissement\_et\_category<- essentiel %>%  
 group\_by(BOROUGH,CATEGORY) %>%  
 summarise(total = n()) %>%  
 group\_by(CATEGORY)%>%  
 mutate(proportion = total / sum(total))

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

essentiel\_par\_arrondissement\_et\_category

## # A tibble: 33 × 4  
## # Groups: CATEGORY [7]  
## BOROUGH CATEGORY total proportion  
## <fct> <fct> <int> <dbl>  
## 1 Bronx Affordable Housing 80 0.215  
## 2 Bronx Approved Work 518 0.124  
## 3 Bronx Homeless Shelter 1 0.2   
## 4 Bronx Hospital / Health Care 55 0.212  
## 5 Bronx Public Housing 276 0.272  
## 6 Bronx Schools 229 0.179  
## 7 Bronx Utility 10 0.111  
## 8 Brooklyn Affordable Housing 168 0.452  
## 9 Brooklyn Approved Work 1223 0.292  
## 10 Brooklyn Hospital / Health Care 66 0.255  
## # … with 23 more rows

# Ajoutez une heatmap ici avec geom\_tile()  
  
  
ggplot(data = essentiel\_par\_arrondissement\_et\_category,  
 mapping = aes(x = BOROUGH, y = CATEGORY, fill = total)) +  
 geom\_tile() +  
 labs(  
 x = "Arrondissement",  
 y = "Catégorie",  
 fill = "Total de projets"  
 ) +  
 scale\_fill\_viridis\_c()

