TP cartographie avec R

OBJECTIFS DU TP:

- Le but de ce TP est de se sensibiliser à la discrétisation de variables, et découvrir des packages de cartographie avec R. Nous verrons notamment l'utilisation du package `mapsf`pour les cartes "statiques" et `leaflet` pour les cartes dynamiques.

Afin d'utiliser une version de R plus récente (et une version du package sf plus récente aussi), vous travaillerez sur le datalab (plateforme du sspcloud, service de l'Insee) : https://datalab.sspcloud.fr.

Nous travaillerons avec les fonds disponibles sous "U:/Eleves/Cartographie/Fonds_carte".

PS : Ce répertoire ne doit pas être votre répertoire de travail ! Il s'agit d'un répertoire où l'on met à disposition des fonds de carte au service de tout le monde. Leur modification pénaliserait donc tous ses utilisateurs.

Commencez par créer un projet pour cette séance de TP, n'oubliez pas de cocher "Create a git repository". Pour ce TP, vous aurez besoin des packages suivants pour le TP :

```
# Chargement des packages
library(dplyr)
library(sf)
library(mapsf)
library(classInt)
library(leaflet)
```

Exercice 1

Le but de cet exercice est de discrétiser une variable continue et d'en observer les différents résultats selon la méthode choisie.

Vous utiliserez le fond des communes de France metropolitaine sur lequel vous calculerez une variable de densité. Pour la variable de population, vous disposez notamment du fichier "Pop_legales_2019.xlsx" présent dans le dossier "U:/Eleves/Cartographie/Donnees".

1. Commencez par vous créer votre jeu de données (jointure et création de variable). Attention avant de joindre vos données, il vous faudra d'abord homogénéiser la commune de Paris. Dans un fichier (fond communal), Paris est renseigné sous son code communal (75056). Dans l'autre, Paris est renseigné par arrondissement (75101 à 75120). Vous devrez donc regrouper les arrondissements pour avoir une seule ligne pour Paris. Cette ligne sera renseignée avec le CODGEO 75056.
2. Regarder rapidement la distribution de la variable de densite

3. On souhaite représenter la variable de densite sous forme d'une carte choroplèthe. Faire cela en utilisant la fonction `plot()`. La variable de densité sera selectionnée par les crochets sur le modèle suivant : `ma_table["ma_variable_continue"]`. Ajouter également l'argument `border=FALSE` pour supprimer les bordures des polygônes.

4. On se rend compte que la carte est peu informative et qu'il vaut mieux discrétiser notre variable. Représenter le résultat de la discrétisation de la densité de population selon les méthodes des quantiles, jenks et des écarts-types, ainsi que la méthode `pretty`. Vous utiliserez la fonction `plot` en ajoutant l'argument `breaks=` + le nom de la méthode. Vous analyserez les différences entre les différentes cartes. Laquelle retiendriez-vous? Pourquoi?

5. Pour obtenir une classification satisfaisante, il faut pouvoir comparer la distribution de la variable continue avec celle de la variable discrétisée. Le package `classInt` est très utile pour construire et analyser les classes.

a. Discrétiser la variable de densité avec la fonction `classInt::classIntervals` avec la méthode des quantiles (argument `style`) et 5 classes (argument `n`). Vous pourrez vous appuyer sur le modèle suivant :

```
objet_decoupe <- classIntervals(
  ma_table$ma_var_continue,
  style = "quantile",
  n = 5
)
```

Analyser ensuite l'objet obtenu. Quelles informations contient-il ? Quelles sont les bornes des intervalles qui ont été construites?

b. Construire une palette de couleurs avec le code suivant:

```
pal1 <- RColorBrewer::brewer.pal(n = 5, name = "YlOrRd")
```

Représenter ensuite l'objet précédent (découpage quantile) avec la fonction `plot`et cette palette de couleur (argument `pal=`). Vous ajouterez l'argument `main` à votre fonction `plot` pour préciser un titre. Analyser le graphique.

c. Relancer l'analyse pour les méthodes sd, jenks et pretty.

d. Finalement, on décide de discrétiser notre variable avec les bornes suivantes : [0;40[,[40;162[,[162;1000[,[1000;8000[et [8000;27200[. Ajouter la variable discrétisée dans le fond communal. Vous utiliserez la fonction `cut`. Vous ferez attention à l'inclusion des bornes inférieures et à l'exclusion des bornes supérieures.

e. Analyser la distribution de cette variable. Représenter la variable discrétisée sur une carte, en créant préalablement une nouvelle palette de couleurs ayant le bon nombre de classes.

Exercice 2

Représenter sous forme de carte le taux de pauvreté par département. Vous utiliserez le package `mapsf`. Vous trouverez de la documentation sur ce package ici. Vous utiliserez le fond "dep_francemetro_2021" ainsi que le fichier "Taux_pauvrete_dept_2021.xlsx" présent dans le dossier "U:/Eleves/Cartographie/Donnees". Pour l'import de ce fichier, vous pouvez utiliser la fonction `openxlsx::read.xlsx()`.

1. Dans un premier temps, vous pourrez essayer de faire 3 cartes basiques : une en découpant la variables d'intérêt selon la méthode de Fisher (breaks="fisher"), une autre avec des classes de même amplitude ("equal") et enfin selon la méthode des quantiles ("quantile").

2. Dans un 2e temps, vous ferez un découpages manuel avec les seuils suivants : 0, 13, 17, 25, max(Tx_pauvrete). La carte contiendra également un zoom sur Paris et sa petite couronne (departements 75, 92, 93, 94).

3. Sauvegarder le jeu de données ayant servi à faire la carte. Pour cela, vous utilisez la fonction `sf::st_write`. Votre sauvegarde se fera sous le format gpkg (appelé geopackage) sous le nom "dept_tx_pauvrete_2018.gpkg". Ce format regroupe à lui seul les 4 extensions usuelles (shp, shx, dbf, prj).

Exercice 3

Dans la poursuite de la découverte de `mapsf`, réaliser une carte choroplèthe (sur une variable de densite) avec ronds proportionnels (sur une variable de population). Pour cela, vous utiliserez le fond des regions de France metropolitaine sur lequel vous calculerez une variable de densité. Pour la variable de population, vous disposez notamment du fichier "pop_region_2019.xlsx" présent dans le dossier "U:/Eleves/Cartographie/Donnees". La carte se fera avec l'option `type="prop_choro"` de la fonction de mf_map.

Exercice 4

1.a. Importer le fond des communes de France métropolitaine `commune_francemetro_2021.shp` en utilisant l'argument `options = "ENCODING=WINDOWS-1252"`.

b. Importer le fichier `bpe20_sport_loisir_xy.csv` (séparateur de colonnes `;` et séparateur des décimales `.`) et prenez connaissance des données. Il s'agit de la liste géolocalisée des équipements de sports et loisirs en France. Une ligne est donc un équipement. Pour la géolocalisation des équipements, le système de projection utilisé est le Lambert-93 (epsg=2154) pour la Métropole. Le séparateur de colonnes est le `;`. Le fichier est issu du site de l'Insee: https://www.insee.fr/fr/statistiques/3568638?sommaire=3568656. Vous y trouverez la documentation nécessaire pour comprendre les données. Repérer notamment les variables renseignant les coordonnées géographiques des équipements, ainsi que la variable indiquant le type d'équipement.

2. Charger un fond OpenStreetMap centré sur la France métropolitaine. Pour cela, repérer les coordonnées (longitude/latitude) d'un point situé approximativement au centre de la France sur https://www.openstreetmap.org/ (un clic-droit + afficher l'adresse). Ensuite utiliser le code suivant où longitude et latitude correspondent aux coordonnées récupérées :

```
leaflet() %>%
  setView(lng = longitude, lat = latitude, zoom = 5)%>%
  addTiles()
```

3. Nous souhaitons positionner sur la carte l'emplacement des bowlings (`TYPEQU == "F119"`) sur la carte.

a. Récupérer de la base d'équipements les bowlings en France Métropolitaine (`REG > 10`) et les stocker dans une table à part. Retirer également les bowlings dont les coordonnées sont maquantes.

b. Les cartes réalisées avec `leaflet` nécessitent d'utiliser des coordonnées dans le système de projection `WGS84 - epsg = 4326`. Transformer la base de bowlings en un objet sf avec la fonction `st_as_sf` et en utilisant l'argument `coords` pour préciser le nom des variables de coordonnées ainsi que l'argument `crs = 2154` (système de projection : RGF93). Ensuite, reprojeter le fond obtenu dans le système de projection adéquat avec la fonction `st_transform(crs = 4326)`.

c. Ajouter un marqueur localisant chacun des bowlings sur la carte interactive précédemment initialisée. Utiliser la fonction `addMarkers` et l'argument `data`.

d. Il est posible d'ajouter un `popup` qui affiche une information complémentaire quand on clique sur un marqueur. Reprendre le code précédent et ajouter l'argument `popup=~DEPCOM` pour afficher le code de la commune dans laquelle le bowling est installé.

Une carte choroplèthe interactive

4.a. Initialiser un fond de carte interactif centré sur le département de votre choix. Adapter la démarche vue en 2.

b. Importer le fichier `base-cc-evol-struct-pop-2018_echantillon.csv`. Restreindre le tableau aux communes du département que vous aurez choisi et aux variables `CODGEO`, `P18_POP` (population communale) et `P18_POP0014` (population de moins de 14 ans). Calculer la part des moins de 14 ans dans la population communale.

c. Fusionner le fond communal et le tableau que vous venez de constituer. Assurez-vous que le fond communal soit restreint aux communes du département choisi. Changer le système de projection pour qu'il soit de type WGS84.

d. Ajouter à la carte interactive initialisée en a. les polygônes du fond communal du département choisi. Utiliser la fonction `addPolygons` en utilisant les arguments `data` (le fond communal), `weight` (pour préciser l'épaisseur des bordures), `color` (couleur des bordures des polygônes) et `fill=NA` (absence de couleur de remplissage).

e. On souhaite ajouter une analyse thématique en représentant la part de moins de 14 ans dans chaque commune. Pour cela, il faut préalablement discrétiser la variable étudiée. On utilisera une discrétisation par la méthode des quantiles avec la fonction `colorQuantile`. Ses arguments sont :

- `palette` : nom d'une palette, apr exemple "YlOrRd". Une liste de palettes disponibles peut s'obtenir avec `RColorBrewer::display.brewer.all()`.
- `domain` : variable numérique à discrétiser.
- `n` : nombre de classes

Créer une palette avec la fonction `colorQuantile` en utilisant le modèle suivant :

```
pal <- colorQuantile(
  "Blues",
  domain = ma_table$var_a_discretiser,
  n = 5
)
```

f. Ajouter l'analyse thématique à la carte interactive en ajoutant l'argument `fillColor` dans la fonction `addPolygons` utilisée en d. Retirer l'argument `fill` et ajouter l'argument `fillOpacity` pour gérer l'opacité de l'analyse thématique (1=opaque, 0=transparent).

g. Ajouter la légende de l'analyse avec la fonction `addLegend`.

h. On souhaite ajouter un popup à chaque commune affichant :

- le nom de la commune (variable `libelle`);
- la population de la commune (variable `P18_POP`);
- la part de la population de moins de 14 ans.

Pour cela, créer une variable `contenu_popup` dans le fond communal qui concatène les différentes informations ci-dessus. Pour ceux qui connaissent le html, on peut utiliser des balises html pour la mise en forme du popup sur le modèle suivant :

```
ma_table$contenu_popup <- paste0(
    "<b>",ma_table$libelle,"</b>",
    "<br>",
    "Population: ",
    format(ma_table$population,big.mark = " "),
    "<br>",
    "Part moins de 15 ans: ",
    round(ma_table$part_moins_15ans,1),
    "%"
    )
```

Ajouter ensuite à la fonction `addPolygons` l'argument `popup = ~contenu_pop`.

i. On souhaite enfin ajouter l'emplacement des bassins de natation.Construire une base des équipements de natation (`TYPEQU == "F101"`) situés dans le département choisi. Transformer ce dataframe en un objet sf comme en 3.b

Enfin, ajouter les emplacements des bassins de natation sur la carte interactive communale. Inspirez-vous de 4.c

```
map_choro_popup %>%
  addMarkers(
    data = natation_sf,
    label = ~as.character(DEPCOM)
    )
```

j. La dernière étape consiste à ajouter un contrôle des couches à faire apparaître. Reprendre l'ensemble du code de fabrication de la carte interactive pour constituer un seul bloc de code. Ajouter l'argument `group=` avec un nom identifiant la couche dans chaque fonction correspondant à une nouvelle couche. Ajouter une dernière instruction à l'enchaînement en utilisant la fonction `addLayersControl()` et l'argument `overlayGroups` en y listant les différents groupes de couches.

COMMENTAIRES/INDICATIONS

Exercice 1

Une représentation réalisée directement sur une variable continue peut être pertinente quand le nombre d'observations est très grand et quand la distribution est suffisamment équilibrée. Ce n'est souvent pas le cas.

La discrétisation Il est alors préférable de discrétiser la variable continue, c'est-à-dire de répartir les observations dans un certain nombre de classes.

Pour cela, il faut pouvoir déterminer une méthode et un nombre de classes. Les méthodes les plus utilisées sont les suivantes:

- **les quantiles** : chaque classe est composée du même nombre d'observations;
- **les seuils dits "naturels"** (k-means,jenks,fisher) : les classes sont construites pour optimiser le rapprochement des valeurs similaires et ainsi obtenir des classes les plus différentes entre elles. Quand l'algorithme des k-means est généralisable à un espace de dimension `n`, l'algorithme de Jenks est optimisé pour les variables à une dimension qui nous occupe ici.
- **les écarts-types** : la construction des classes est réalisée à partir de la moyenne de la distribution avec une longueur d'un écart-type. Cette méthode est adaptée aux variables suivant approximativement une distribution normale.
- **les seuils manuels** : Le seuils sont choisis manuellement. Cette méthode peut être utilisée pour affiner les classes obtenues avec une méthode automatique.

Exercices 2 et 3

Vous pouvez aussi vous reporter au lien suivant.

Exercice 4

Pas mal de documentation sur leaflet avec R:

- le blog thinkR (blog sur R avec souvent de très bons tutoriels)
- site de Laurent Rouvière (tres complet sur R)
- github Rstudio présentant leaflet