

Data Management & visualisation

Plusieurs logiciels et langages

Direction de la Santé Publique (DSP)
Ville de Paris

Marie EXBRAYAT
Alternance Master II

DBeaver

Utilisation du logiciel DBeaver pour la création de requêtes SQL à partir des données internes de la DSP sur un driver jdbc

The screenshot displays the DBeaver 24.2.5 interface. The main window shows a SQL query editor with the following text:

```
-- Prevalence de la drépanocytose & dépistage de la drépanocytose
-- Indicateur : prévalence de la drépanocytose parmi les personnes étant dépistée

*SELECT
  arrondissement,
  annee,
  CAST(CONCAT(annee, '-01-01') AS DATE) AS date,
  COUNT(DISTINCT id_patient) AS nb_depistes,
  SUM(CASE WHEN id_classe_formulaire = 15065 AND valeur = 'Oui' THEN 1 ELSE 0 END) AS nb_atteints
FROM (
  SELECT
    DISTINCT id_patient,
    libelle_classe_formulaire,
    id_classe_formulaire,
    CASE
      WHEN code_postal_residence = '75000' THEN NULL
      WHEN code_postal_residence IN ('75001', '75002', '75003', '75004') THEN 'Centre'
      WHEN SUBSTRING(code_postal_residence, 1, 2) = '75' THEN SUBSTRING(code_postal_residence, 4, 2)
      ELSE NULL
    END AS arrondissement,
    CAST(EXTRACT(YEAR FROM date_event) AS VARCHAR) AS annee,
    valeur_contenu_formulaire AS valeur
  FROM iceberg.ref_dsp_si_pmi.evenements
  WHERE id_classe_formulaire IN (15040, 15065) -- Questions sur la drépanocytose
  AND valeur_contenu_formulaire IS NOT NULL
  AND date_event BETWEEN CAST('2016-01-01' AS DATE) AND CAST('2024-12-31' AS DATE)
  AND id_patient IN (
    SELECT DISTINCT id_patient
    FROM iceberg.ref_dsp_si_pmi.evenements
    WHERE id_classe_formulaire = 15040 AND valeur_contenu_formulaire = 'Oui'
  )
) AS sub
WHERE arrondissement IS NOT NULL AND annee IS NOT NULL
GROUP BY arrondissement, annee
ORDER BY annee, arrondissement;
```

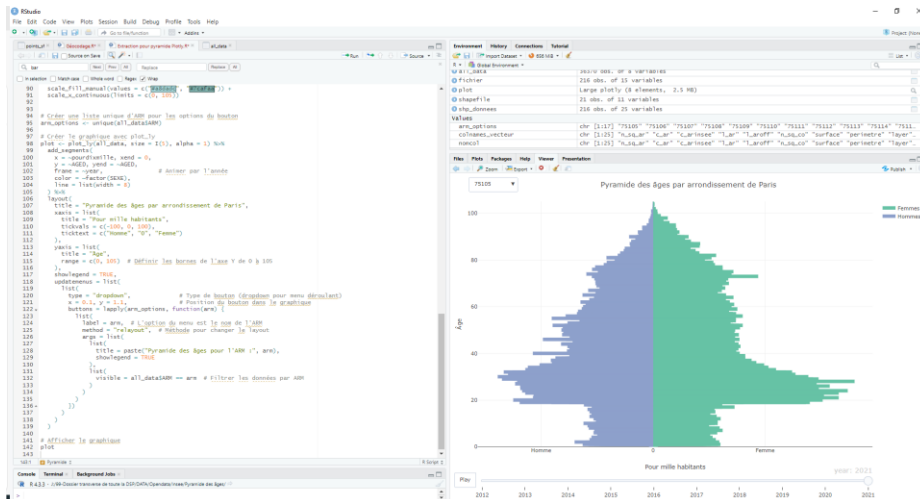
The bottom panel shows the results of the query in a table format:

id_patient	libelle_classe_formulaire	id_classe_formulaire	arrondissement	annee	valeur	age_years
410-000	Dépistage de la drépanocytose	15065	17	2022	Non	0
420-407	Dépistage de la drépanocytose	15065	17	2022	Non	0
500-018	Dépistage de la drépanocytose	15065	17	2022	Non	0
500-080	Dépistage de la drépanocytose	15065	13	2022	Non	2
410-022	Dépistage de la drépanocytose	15065	20	2022	Oui	0

RSTUDIO

Utilisation du logiciel RSTUDIO pour divers projets :

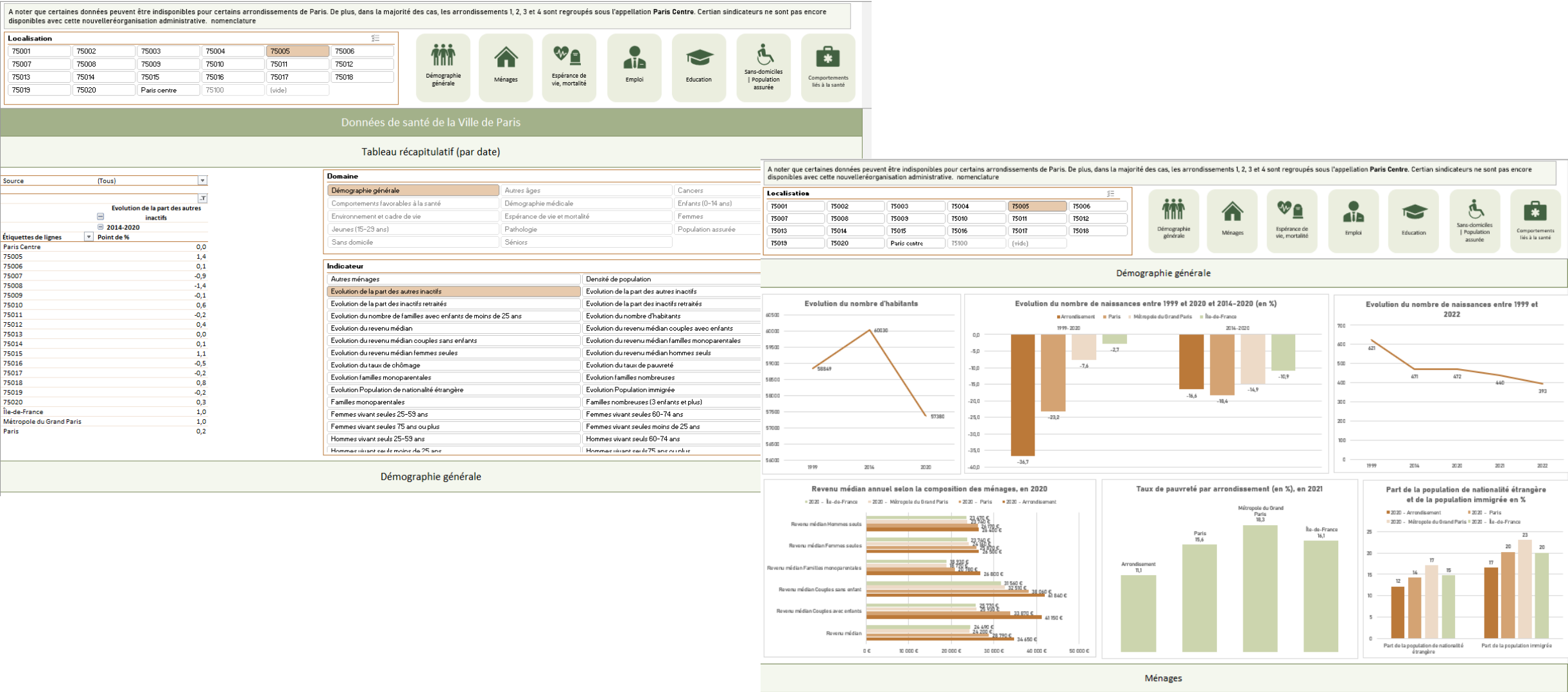
- Data management et nettoyage de bases de données.
- Géocodage de données et jointures spatiales avec le package sf.
- Application Rshiny : cartographie des indicateurs de santé opendata.
- Application Plotly : pyramide des âges interactive



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
points_sf Géocodage.R Extraction pour pyramide Plotly.R
Source on Save
Run
8 # Charger les données
9 file_path <- file.choose()
10 base <- read.csv2(file_path)
11
12 # Géocodage des QPV -----
13
14 qpv <- st_read("Géographie prioritaire parisienne (QPV - QPOP - Secteur de veille).geojson")
15 qpv <- qpv %>%
16   select(id_pgp, l_pgp, l_type_pgp, geometry) %>%
17   st_crs()
18
19 # Convertir la base des adresses en objet spatial sf
20 points_sf <- st_as_sf(base, coords = c("longitude", "latitude"), crs = 4326)
21 points_sf$longitude <- base$longitude
22 points_sf$latitude <- base$latitude
23
24 # Trouver les intersections entre points et quartiers prioritaires
25 intersections <- st_intersects(points_sf, qpv)
26 max_qpv <- max(sapply(intersections, length))
27
28 # Ajouter dynamiquement les colonnes nécessaires à points_sf
29 for (i in seq_len(max_qpv)) {
30   points_sf[[paste0("id_pgp_", i)]] <- NA
31   points_sf[[paste0("l_pgp_", i)]] <- NA
32   points_sf[[paste0("l_type_pgp_", i)]] <- NA
33 }
34
35 # Remplir les colonnes avec les ID, noms et types des quartiers associés
36 for (i in seq_along(intersections)) {
37   if (length(intersections[[i]]) > 0) {
38     ids <- qpv$id_pgp[intersections[[i]]]
39     noms <- qpv$l_pgp[intersections[[i]]]
40     types <- qpv$l_type_pgp[intersections[[i]]]
41
42     for (j in seq_along(ids)) {
43       points_sf[[paste0("id_pgp_", j)]][[i]] <- ids[j]
44       points_sf[[paste0("l_pgp_", j)]][[i]] <- noms[j]
45       points_sf[[paste0("l_type_pgp_", j)]][[i]] <- types[j]
46     }
47   }
48 }
49
50 rm(j, j, max_qpv, ids, noms, types)
51
52 points_sf <- points_sf %>%
53   mutate(
54     "quartier" = case_when(
55       is.na(id_pgp_1) ~ "Oui",
56       T ~ "Non"
57     ),
58     "quartier_val" = case_when(
59       is.na(id_pgp_1) ~ 1,
60       T ~ 0
61     ),
62     "nb_quartier" = case_when(
63       is.na(id_pgp_2) ~ 2,
64       is.na(id_pgp_1) ~ 1,
65       T ~ 0
66     )
67   )
68
69 51:1 Géocodage des QPV
R Script
```

EXCEL & Power Query

Utilisation de Excel pour certains projets de tableaux de bord. Traitement de la base INSEE via Query.



KNIME

Utilisation du logiciel KNIME, qui regroupe diverses fonctionnalités de Excel, Python, SQL, R et la possibilité d'installer des packages pour la gestion de données géographiques.

