Sujet TP3

Contexte

Le “Forever Pollution Project” est une enquête journalistique qui s’intéresse aux substances per-et-polyfluoroalkylées (PFAS), souvent surnommées “polluants éternels” en raison de leur extrême persistance dans l’environnement. Utilisés depuis les années 1940 pour leurs propriétés résistantes à l’eau et à la chaleur, ces composés chimiques sont aujourd’hui une source majeure de préoccupation environnementale et sanitaire. L’enquête se distingue par son approche rigoureuse, basée sur des méthodologies scientifiques reconnues et des prélèvements environnementaux étendus. Un article du journal Le Monde en date du 23 février 2023 résume cette enquête ainsi que la méthode d’analyse retenue : https://www.lemonde.fr/les-decodeurs/article/2023/02/23/polluants-eternels-explorez-la-carte-d-europe-de-la-contamination-par-les-pfas_6162942_4355770.html.

Sur le site https://pdh.cnrs.fr/fr/, le Centre National de la Recherche Scientifique (CNRS) met à disposition, en accès libre, les données et livre plusieurs analyses. L’objectif des TP3 et 4 est d’étudier un jeu de donnée, disponible sur ce site, relatif à l’eau du robinet en France

---

Exercice 1 : Analyse des données

Master 1 – BDIA Analyse et visualisation des données Sujet TP3 Contexte Le “Forever Pollution Project” est une enquête journalistique qui s’intéresse aux sub- stances per- et polyfluoroalkylées (PFAS), souvent surnommées “polluants éternels” en raison de leur extrême persistance dans l’environnement. Utilisés depuis les années 1940 pour leurs propriétés résistantes à l’eau et à la chaleur, ces composés chimiques sont au- jourd’hui une source majeure de préoccupation environnementale et sanitaire. L’enquête se distingue par son approche rigoureuse, basée sur des méthodologies scientifiques re- connues et des prélèvements environnementaux étendus. Un article du journal Le Monde en date du 23 février 2023 résume cette enquête ainsi que la méthode d’analyse retenue : https://www.lemonde.fr/les-decodeurs/article/2023/02/23/polluants-eternels -explorez-la-carte-d-europe-de-la-contamination-par-les-pfas_6162942_4355770.html. Sur le site https://pdh.cnrs.fr/fr/, le Centre National de la Recherche Scientifique (CNRS) met à disposition, en accès libre, les données et livre plusieurs analyses. L’objectif des TP3 et 4 est d’étudier un jeu de donnée, disponible sur ce site, relatif à l’eau du robinet en France. Exercice 1 : Analyse des données Sur le site https://pdh.cnrs.fr/fr/, télécharger le jeu de données numéro 124, désigné france_eaurob relatif à l’analyse de l’eau du robinet en France. L’objectif de cet exer- cice est de comprendre la structure ainsi que les données de ce dataset.

```
library(dplyr)
library(tidyr)
library(purrr)
library(jsonlite)
library(tm)
library(wordcloud)

dataset <- read.csv("data/pdh_export.csv")
```

1. Combien de lignes et combien de colonnes dans ce dataset ?

```
msg <- sprintf('Nbr lignes : %d | Nbr colonnes : %d', nrow(dataset), ncol(dataset))
print(msg)
```

```
## [1] "Nbr lignes : 39448 | Nbr colonnes : 21"
```

2. Afficher le nom des colonnes.

```
column_name <- names(dataset)
print(column_name)
```

```
##  [1] "category"              "lat"                   "lon"
##  [4] "name"                  "city"                  "country"
##  [7] "type"                  "sector"                "source_type"
## [10] "data_collection_method" "source_text"           "source_url"
## [13] "dataset_id"            "dataset_name"          "pfas_values"
## [16] "unit"                  "pfas_sum"              "details"
## [19] "matrix"                "date"                  "year"
```

3. Afficher la première ligne du dataset.

```
first_row <- head(dataset)
print(first_row)
```

```
##   category lat lon              name                 city country
## 1 Sampling  NA  NA Sampling location               SAFFRE  France
## 2 Sampling  NA  NA Sampling location               SAFFRE  France
## 3 Sampling  NA  NA Sampling location SAINT-MARS-DU-DESERT  France
## 4 Sampling  NA  NA Sampling location             MASSERAC  France
## 5 Sampling  NA  NA Sampling location             MASSERAC  France
## 6 Sampling  NA  NA Sampling location       NORT-SUR-ERDRE  France
##                type sector source_type data_collection_method
## 1 Sampling location     NA Authorities                     NA
## 2 Sampling location     NA Authorities                     NA
## 3 Sampling location     NA Authorities                     NA
## 4 Sampling location     NA Authorities                     NA
## 5 Sampling location     NA Authorities                     NA
## 6 Sampling location     NA Authorities                     NA
##                 source_text
## 1 French ministry of health
## 2 French ministry of health
## 3 French ministry of health
## 4 French ministry of health
## 5 French ministry of health
## 6 French ministry of health
##                                                                                source_url
## 1 https://www.data.gouv.fr/fr/datasets/resultats-du-controle-sanitaire-de-leau-du-robinet/
## 2 https://www.data.gouv.fr/fr/datasets/resultats-du-controle-sanitaire-de-leau-du-robinet/
## 3 https://www.data.gouv.fr/fr/datasets/resultats-du-controle-sanitaire-de-leau-du-robinet/
## 4 https://www.data.gouv.fr/fr/datasets/resultats-du-controle-sanitaire-de-leau-du-robinet/
## 5 https://www.data.gouv.fr/fr/datasets/resultats-du-controle-sanitaire-de-leau-du-robinet/
## 6 https://www.data.gouv.fr/fr/datasets/resultats-du-controle-sanitaire-de-leau-du-robinet/
##   dataset_id  dataset_name
```

```
## 1        124 france_eaurob
## 2        124 france_eaurob
## 3        124 france_eaurob
## 4        124 france_eaurob
## 5        124 france_eaurob
## 6        124 france_eaurob
##
## 1
## 2
## 3 [{"cas_id":"103055-07-8","unit":"ng/l","substance":"Lufenuron","isomer":null,"less_than":"100.0","
## 4
## 5
## 6
##   unit pfas_sum
## 1 ng/l        0
## 2 ng/l        0
## 3 ng/l        0
## 4 ng/l        0
## 5 ng/l        0
## 6 ng/l        0
##
## 1                {"inae":"ESO","referenceprel":"04400255074","distrlib":"VEOLIA","libtypeeau":"EAU
## 2                {"inae":"ESO","referenceprel":"04400255075","distrlib":"VEOLIA","libtypeeau":"EAU
## 3 {"inae":"ESO","referenceprel":"04400255095","distrlib":"VEOLIA","libtypeeau":"ESU+ESO TURB >2 APPL
## 4              {"inae":"ESO","referenceprel":"04400255145","distrlib":"SAUR","libtypeeau":"ESU+ESO T
## 5                    {"inae":"ESO","referenceprel":"04400255147","distrlib":"SAUR","libtypeeau":"EA
## 6      {"inae":"ESO","referenceprel":"04400255177","distrlib":"VEOLIA","libtypeeau":"ESO A TURB. < 2
##        matrix       date year
## 1 Groundwater 2024-04-19 2024
## 2 Groundwater 2024-04-19 2024
## 3 Groundwater 2024-04-19 2024
## 4 Groundwater 2024-04-15 2024
## 5 Groundwater 2024-04-15 2024
## 6 Groundwater 2024-04-26 2024
```

4. Vérifier que les données concernent uniquement la France.

```
onlyFrenchData <- unique(dataset$country == "France") && length(unique(dataset$country) == 1)
print(onlyFrenchData)
```

```
## [1] TRUE
```

5. Les prélèvements concernent combien de villes différentes ?

```
nbrUniqueCity <- length(unique(dataset$city))
print(nbrUniqueCity)
```

```
## [1] 11092
```

6. Est-ce que des prélèvements sont réalisés à Dijon, si oui combien ?

```
nbrDijonData <- sum(dataset$city == "DIJON")
print(nbrDijonData)
```

```
## [1] 40
```

7. En étudiant *matrix*, préciser les différentes sources de prélèvement et indiquer le nombre de prélèvements par sources.

```
repartition <- dataset %>% group_by(matrix) %>% summarise(n())
print(repartition)
```

```
## # A tibble: 5 x 2
##   matrix         `n()`
##   <chr>          <int>
## 1 Drinking water 11251
## 2 Groundwater    21606
## 3 Sea water         10
## 4 Surface water   5654
## 5 Unknown          927
```

8. Quelle est la période couverte par les prélèvements ?

```
periode <- dataset %>%
  arrange(date) %>%
  summarise(Debut = first(date), Fin = last(date))
print(periode)
```

```
##        Debut        Fin
## 1 2015-06-23 2024-12-31
```

9. Indiquer le nombre de prélèvement par an.

```
prelevementParAn <- dataset %>%
  group_by(year) %>%
  summarise('Nombre de prélèvement' = n()) %>%
  rename('Année' = year)
print(prelevementParAn)
```

```
## # A tibble: 10 x 2
##    Année `Nombre de prélèvement`
##    <int>                   <int>
##  1  2015                       5
##  2  2016                       4
##  3  2017                       5
##  4  2018                       2
##  5  2019                       9
##  6  2020                      71
##  7  2021                      52
##  8  2022                      76
##  9  2023                    3442
## 10  2024                   35782
```

10. Citer les 5 villes dans lesquelles on a réalisé le plus de prélèvement.

```
topFiveCity <- dataset %>%
  count(city, sort = TRUE) %>%
```

```
  slice(1:5)
print(topFiveCity)
```

```
##            city   n
## 1 MERY-SUR-OISE 117
## 2      NARBONNE  88
## 3  SAINT-JOSEPH  79
## 4     MARSEILLE  64
## 5 CHOISY-LE-ROI  62
```

Exercice 2 : Statistiques selon les 5 sources de prélèvement

1. Pour chacune de ces 5 sources de prélèvement, calculer la moyenne, la médiane et l'écart type des PFAS *pfas_sum*

```
prelevementStat <- dataset %>%
  group_by(matrix) %>%
  summarise(
    Moyenne = mean(pfas_sum, na.rm = TRUE),
    Mediane = median(pfas_sum, na.rm = TRUE),
    EcartType = sd(pfas_sum, na.rm = TRUE)
  )
print(prelevementStat)
```

```
## # A tibble: 5 x 4
##   matrix         Moyenne Mediane EcartType
##   <chr>            <dbl>   <dbl>     <dbl>
## 1 Drinking water    3.79       0      25.9
## 2 Groundwater       2.37       0      61.8
## 3 Sea water         0          0       0
## 4 Surface water     4.05       0      59.0
## 5 Unknown           2.00       0      13.8
```

2. Quelles sont les sources de prélèvement pour Dijon ?

```
sourceInDijon <- dataset %>%
  filter(city == "DIJON") %>%
  pull(matrix) %>%
  unique()
print(sourceInDijon)
```

```
## [1] "Unknown"        "Drinking water" "Groundwater"
```

3. Les prélèvements "Sea water" concernent quelles villes ?

```
seawaterCity <- dataset %>%
        filter(matrix == "Sea water") %>%
        pull(city) %>%
        unique()
print(seawaterCity)
```

```
## [1] "BREVILLE-SUR-MER" "SAINT-MARTIN"     "SAINT-BARTHELEMY"
```

Exercice 3 : Qualité des données

L'objectif de cet exercice est double. D'une part, il s'agit de nettoyer les données. D'autre part, il s'agit de réduire le dataset aux informations pertinentes et de supprimer toutes les autres, notamment celles liées au contexte de l'étude. Par exemple, toutes les lignes du dataset comporte la valeur 124 dans le champs dataset_id. Cette information liée au contexte de l'étude n'apporte pas d'information pertinente dans l'analyse des données. Les fonctions suivantes vous permettront de répondre aux questions de cet exercice

1. Les noms de ville *city* doivent être en majuscules

```
dataset$city <- toupper(dataset$city)
```

2. Y a-t-il des valeurs manquantes dans certains colonnes ?

```
nbrMiss <- sapply(dataset, function(x) sum(is.na(x)))
print(nbrMiss)
```

```
##               category                    lat                    lon
##                      0                  37370                  37370
##                   name                   city                country
##                      0                      0                      0
##                   type                 sector            source_type
##                      0                  39448                      0
## data_collection_method            source_text             source_url
##                  39448                      0                      0
##             dataset_id           dataset_name            pfas_values
##                      0                      0                      0
##                   unit              pfas_sum                details
##                      0                      0                      0
##                 matrix                   date                   year
##                      0                      0                      0
```

3. Quels sont les types de données présents dans chaque colonne ?

```
typeDataColumn <- sapply(dataset, typeof)
print(typeDataColumn)
```

```
##               category                    lat                    lon
##            "character"               "double"               "double"
##                   name                   city                country
##            "character"            "character"            "character"
##                   type                 sector            source_type
##            "character"              "logical"            "character"
## data_collection_method            source_text             source_url
##              "logical"            "character"            "character"
##             dataset_id           dataset_name            pfas_values
##              "integer"            "character"            "character"
##                   unit              pfas_sum                details
##            "character"               "double"            "character"
```

```
##                matrix                  date                  year
##           "character"           "character"           "integer"
```

4. D'où proviennent les relevés *source_text* ?

```
typeSource <- unique(dataset$source_text)
print(typeSource)
```

```
## [1] "French ministry of health"
```

5. Quelles colonnes comportent des valeurs uniques comme par ex. *dataset_id* ?

```
uniqueValueColumn <- names(dataset)[sapply(dataset, function(x) length(unique(x)) == 1)]
print(uniqueValueColumn)
```

```
##  [1] "category"              "name"                  "country"
##  [4] "type"                  "sector"                "source_type"
##  [7] "data_collection_method" "source_text"           "source_url"
## [10] "dataset_id"            "dataset_name"          "unit"
```

6. En fonction des réponses obtenues aux questions précédentes, supprimer toutes les colonnes qui ne vous paraissent pas indispensables et sauvegarder votre nouveau dataset dans data2. Poursuivre les exercices avec data2.

Exercice 4 : Exploitation des données au format json

Le champ *pfas_values* contient les données suivantes sous forme de listes au format JSON. • cas_id : identifiant unique de la substance (CAS). • unit : unité de mesure. • substance : nom de la substance mesurée. • isomer : information sur l'isomère. • less_than : limite de détection. • value : valeur mesurée

1. Convertir *pfas_values* pour l'exploiter en R

```
# Parser les données JSON de la colonne pfas_values
pfasValues <- lapply(dataset$pfas_values, fromJSON)
# Convertir la liste en data frame
pfasValues_df <- bind_rows(pfasValues)
```

2. Lister et compter les différentes substances présentes.

```
substances <- pfasValues_df %>%
         group_by(substance) %>%
        summarise(n())
print(substances)
```

```
## # A tibble: 63 x 2
##    substance                                      `n()`
##    <chr>                                          <int>
##  1 1-Heptanesulfonic acid, pentadecafluoro- (6CI)  2089
##  2 Acifluorfen                                    15582
##  3 Beflubutamid                                   11304
```

```
##  4 Benfluralin                                20196
##  5 Cyflufenamid                                8543
##  6 Diflufenican                               30374
##  7 Fipronil                                   24808
##  8 Flazasulfuron                              21843
##  9 Flonicamid                                 20818
## 10 Fluazifop-P butyl                           4532
## # i 53 more rows
```

3. Lister les 5 substances les plus fréquentes.

```
topFiveSubstances <- pfasValues_df %>%
        count(substance, sort = TRUE) %>%
        slice(1:5)
print(topFiveSubstances)
```

```
##       substance     n
## 1  Diflufenican 30374
## 2    Flufenacet 30105
## 3   Fludioxonil 29276
## 4   Norflurazon 26110
## 5 Tetraconazole 26095
```

4. Les substances PFOS, PFOA, PFDA et PFNA sont elles présentes, si oui, à quelle fréquence ?

```
fourSubstances <- pfasValues_df %>%
        filter(substance %in% c("PFOS","PFOA","PFDA","PFNA")) %>%
        group_by(substance) %>%
        summarise(frequence = n())
print(fourSubstances)
```

```
## # A tibble: 4 x 2
##   substance frequence
##   <chr>         <int>
## 1 PFDA           2092
## 2 PFNA           2092
## 3 PFOA           2093
## 4 PFOS             24
```

Exercice 5 : Analyse des prélèvements de Dijon

L'objectif de cet exercice est d'étudier les substances prélevées à Dijon et d'établir un classement des villes.

1. Quelles sont les substances (*substance*) détectées à Dijon, leurs valeurs ? qu'en déduisez-vous ?

```
# Parser les données JSON de la colonne pfas_values pour Dijon
pfasValuesDijon <- dataset %>%
        filter(city == "DIJON") %>%
        pull(pfas_values) %>%
```

```
        lapply(fromJSON)

# Convertir la liste en data frame
pfasValuesDijon_df <- bind_rows(pfasValuesDijon)

substancesDijon <- pfasValuesDijon_df %>%
        filter(!is.na(value)) %>%
        select(substance, value)

print(substancesDijon)
```

```
##    substance value
## 1 Flufenacet  23.0
```

2. Etablir un classement des villes par ordre décroissant sur pfas_sum. Le classement prend en compte que les villes dont la somme est supérieure à 1.

```
cityPfasSumRank <- dataset %>%
        filter(pfas_sum > 1) %>%
        arrange(desc(pfas_sum)) %>%
        select(city, pfas_sum)
head(cityPfasSumRank, 10)
```

```
##                  city pfas_sum
## 1               VICHY   8480.0
## 2               BREST   3900.0
## 3               VICHY   1546.0
## 4               VICHY    991.4
## 5        SAINT-PIERRE    906.0
## 6   JOINVILLE-LE-PONT    848.0
## 7  VILLERS-LE-CHATEAU    843.0
## 8               VICHY    839.0
## 9               VICHY    838.5
## 10              CUISY    791.0
```

Exercice 6 : Graphique “nuage de mots” (word cloud)

L’objectif de cet exercice est de réaliser un graphique “nuage de mots” afin de mettre en évidence le nom des substances (voir Exercice 4, question 2).

```
set.seed(123456)  # permet de "fixer un graine" pour l'alea, afin de pouvoir regenerer plusieurs fois l

substancesFiltered <- substances %>%
        filter("n()" > 3) %>%
        arrange(desc("n()")) %>%
        head(50)

png("./wordcloud.png")
wordcloud(
        words = substancesFiltered$substance,
        freq = substancesFiltered$`n()`,
        colors = brewer.pal(8, "Dark2"),
        random.order=FALSE,
```

```
)
dev.off()
```

```
## pdf
##   2
```

```
knitr::include_graphics("./wordcloud.png")
```