

# Atelier R pour le PMSI

EMOIS 2024 – Lille

Guillaume Pressiat - CHU de Brest

05/04/2024

## ● Reproduire des tableaux **e-PMSI** avec R

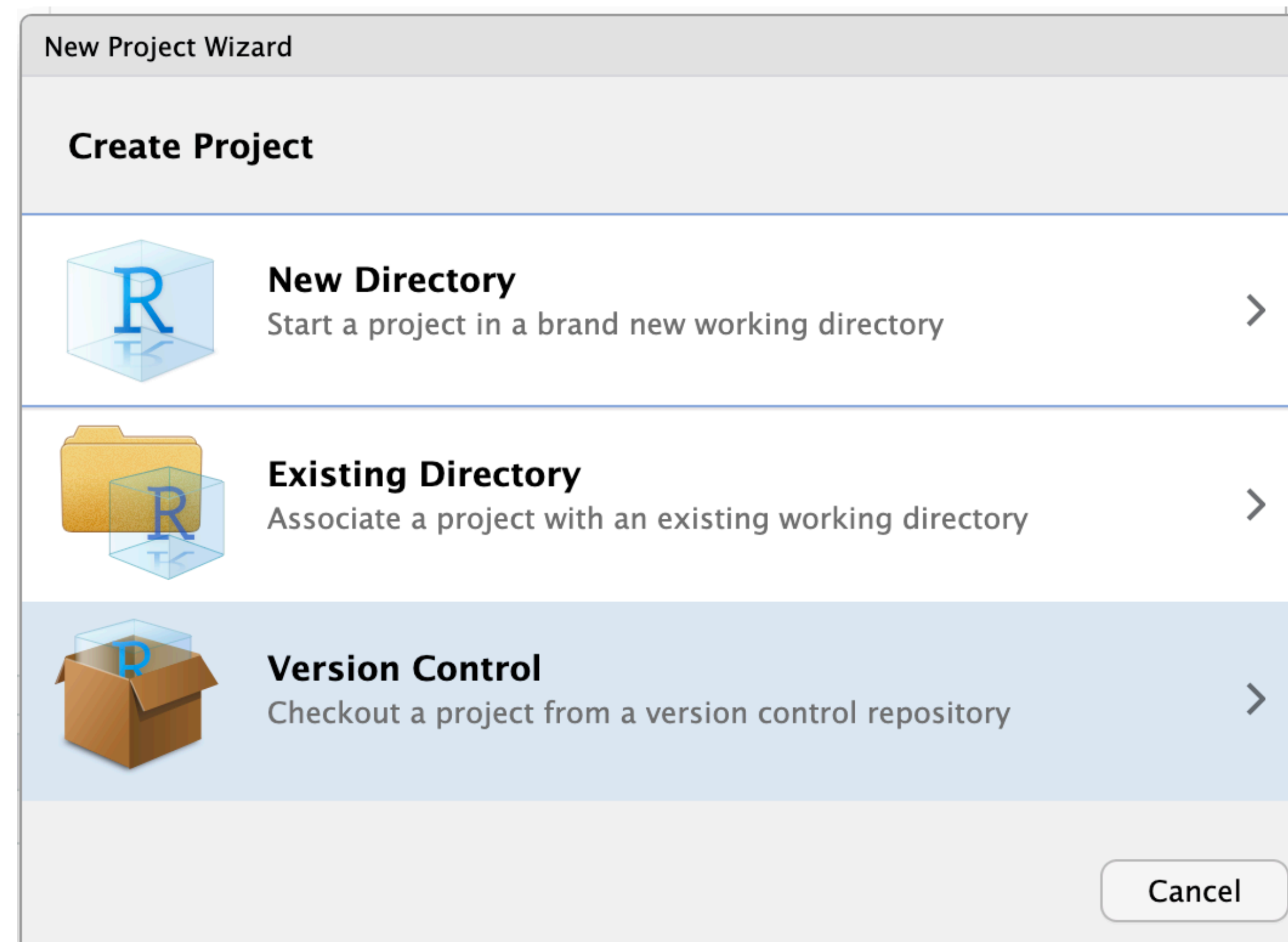
- Contexte, objectifs
- Introduction à pmeasyr et dplyr / tidyr
- **Tableau [1.D.2.RTP] - RSA transmis sur la période**
- Tableau [1.D.2.ME/S] - Répartition des modes d'entrée / sortie
- **Tableau [1.D.2.CMGS] - Case-mix GHM**
- Tableau [1.D.2.SUM] - Nombre et type de suppléments par unité médicale
- Tableau [1.Q.1.EG] - Synthèse des erreurs de groupage
- **Tableau [1.V.1.SV] - Séjours valorisés**
- **Tableau [1.V.1.RAV] - Récapitulation activité - valorisation**
- Tableau [1.D.2.IPA] - Suivi des indicateurs de pilotage des activités de soins autorisées (IPA)
- Tableau [1.V.2.VMED] - Valorisation des molécules onéreuses

## ● Requêter les données sur la base du module requêteur de pmeasyr

- Contexte, objectifs
- CCAM, CIM-10, GHM : référentiels et libellés
- Recours exceptionnel
- Chirurgie bariatrique
- Liste définie par l'utilisateur

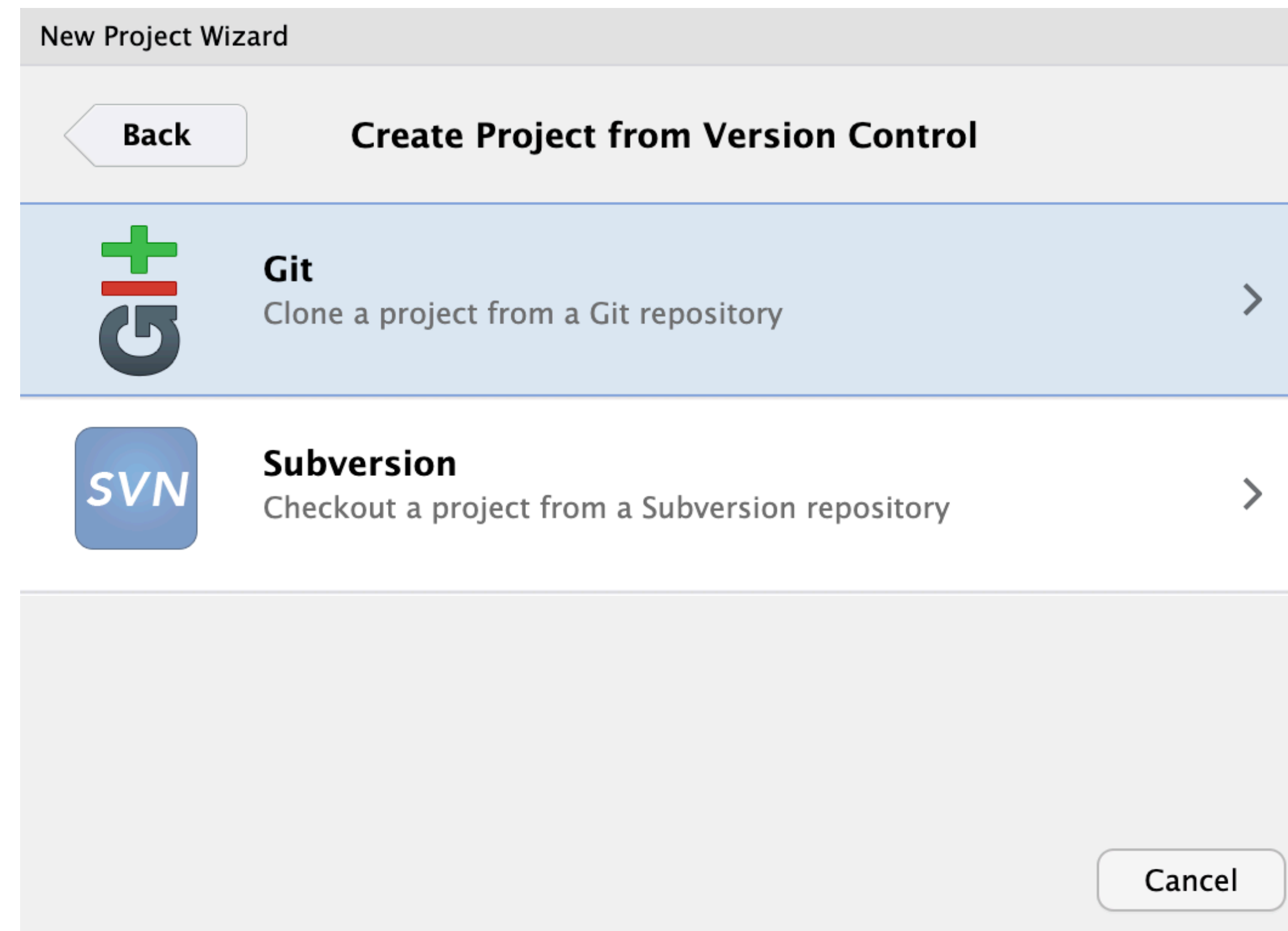
# Initialiser le R projet de l'atelier

- Dans RStudio > File > New project... (1)



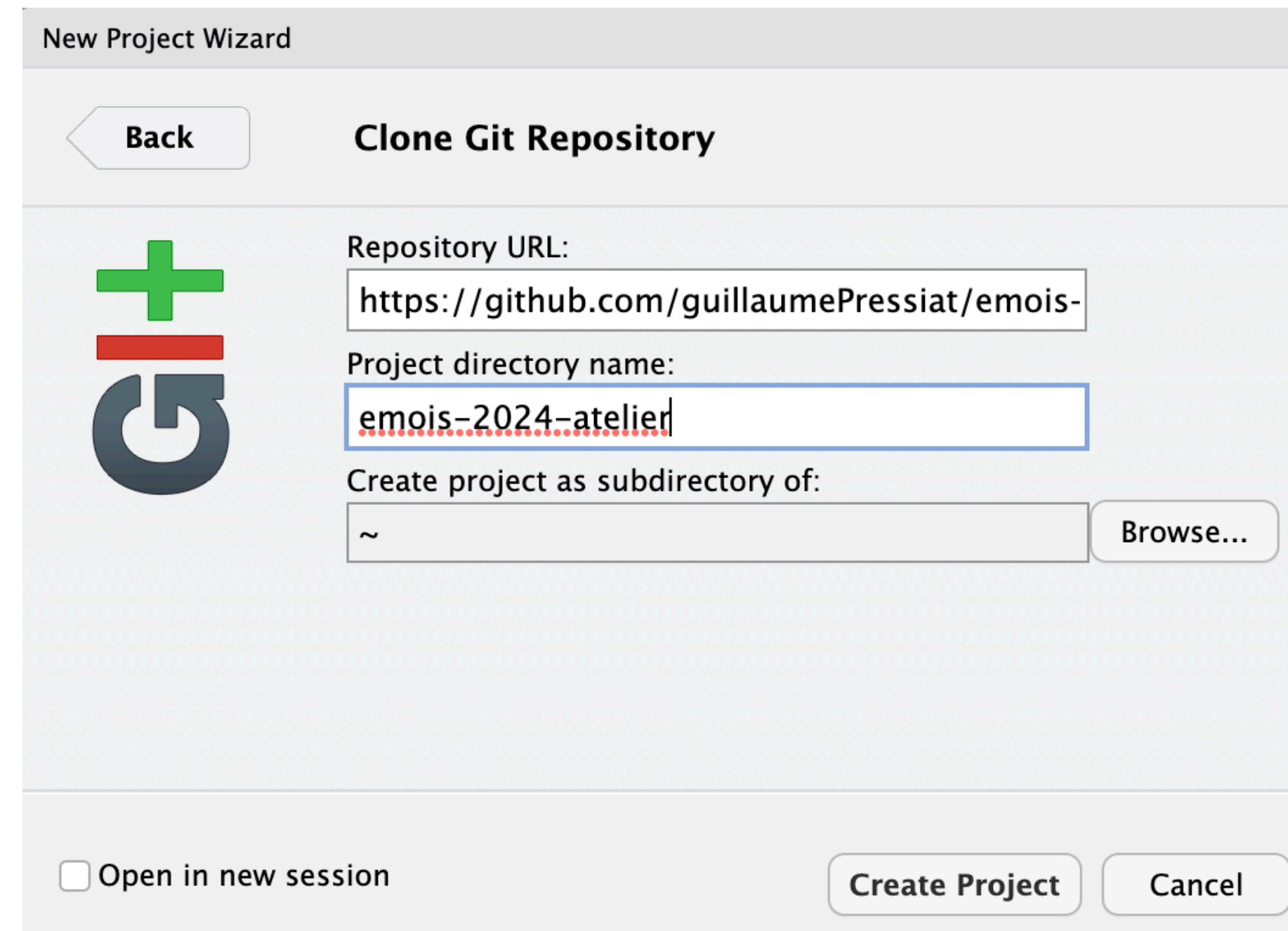
# Initialiser le R projet de l'atelier

- Dans RStudio > File > New project... (2)



# Initialiser le R projet de l'atelier

- Dans RStudio > File > New project... (3)

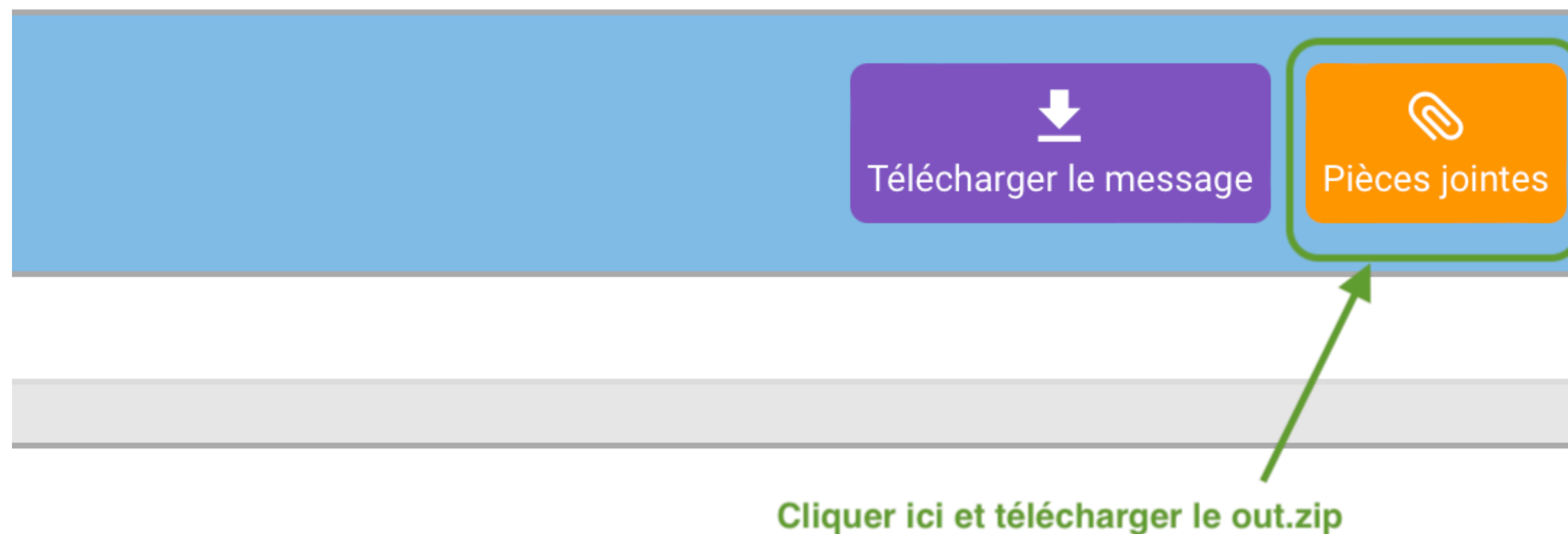


The screenshot shows the 'New Project Wizard' dialog box in RStudio, specifically the 'Clone Git Repository' step. The dialog has a title bar 'New Project Wizard' and a 'Back' button. On the left is a large Git logo. The main area contains three text input fields: 'Repository URL:' with the value 'https://github.com/guillaumePressiat/emois-', 'Project directory name:' with the value 'emois-2024-atelier', and 'Create project as subdirectory of:' with the value '~'. A 'Browse...' button is next to the last field. At the bottom, there is a checkbox 'Open in new session' which is unchecked, and two buttons: 'Create Project' and 'Cancel'.

- <https://github.com/guillaumePressiat/emois-2024-atelier>

## Copier les données

- Télécharger les données ici <https://tinyurl.com/emois-r-2024>



- Placer le fichier 290000017.2023.12.\*\*\*\*.out.zip dans le répertoire emois-2024-atelier/data du projet
- ou, si vous les avez, les données de votre établissement

## Ouvrir un nouveau script

- File > New File > R Script
- Enregistrer le script

# Installation des bibliothèques

- Si vous ne l'avez pas déjà fait en amont :

```
##### Dépendances
install.packages('tidyverse')
install.packages('sjlabelled')
install.packages('sqldf')

##### pmeasyr / nomencland
install.packages('pmeasyr', repos = 'https://guillaumepressiat.r-universe.dev')
install.packages('nomencland', repos = 'https://guillaumepressiat.r-universe.dev')

##### ou
# install.packages('remotes')
# remotes::install_github('guillaumepressiat/pmeasyr')
# remotes::install_github('guillaumepressiat/nomencland')
```



# Reproduire des tableaux *e*-PMSI avec R

## ● Quelques mots

### > **pouvoir reproduire les chiffres e-PMSI**

- donne de la confiance dans les résultats
- permet de mieux comprendre les données et les problèmes
- fait vivre la notion de République numérique, pour ouvrir les savoirs numériques (de haut en bas, de bas en haut)
- permet de rassembler des acteurs de l'hôpital autour de la donnée PMSI (soignants, facturation, pharmacie, etc.)

### > **des chiffres homogènes au niveau EJ permettent**

- de ventiler les indicateurs (drill-down)
- d'harmoniser les données au sein de l'hôpital (cohérence entre systèmes, DIM, décisionnel, recherche)
- ces données sont figées et décorréliées de la production
- partager des méthodes entre établissements et avec les agences

### > **partager des erreurs pour les corriger**

- Charger les packages, définir un noyau de paramètres, dézipper et lire les RSA

```
library(pmeasyr)
library(tidyverse)

# noyau_skeleton()
noyau_pmeasyr(
  finess = '290000017',
  annee  = 2023L,
  mois   = 12L,
  path   = 'data',
  tolower_names = TRUE
) -> p

adezip(p, type = "out")

rsa <- irsa(p, typi = 1)
rsa
View(rsa$rsa)
```

- Accéder à l'aide

```
?irsa
```

- Changer le type d'import (**typi**)

```
rsa <- irsa(p, typi = 1)  
rsa <- irsa(p, typi = 4)  
rsa <- irsa(p, typi = 6)
```

## Tableau [1.D.2.RTP]

- RSA transmis sur la période
  - Compter le nombre de lignes

```
rsa$rsa %>% count()
```

```
# Par mois de sortie des RSA  
rsa$rsa %>% count(moissor)  
# équivalent à  
rsa$rsa %>%  
  group_by(moissor) %>%  
  summarize(n = n())
```

## Tableau [1.D.2.RTP]

- RSA transmis sur la période

```
rsa$rsa %>%  
  # group_by(moissor) %>%  
  summarise(`Nb de RSA transmis` = n(),  
            `Nb de CM90` = n_distinct(cle_rsa[rsacmd == "90"]),  
            `Nb de PIE` = n_distinct(cle_rsa[typesej == "B"]),  
            `Nb de GHS 9999` = n_distinct(cle_rsa[noghs == "9999" & rsacmd != "90"]),  
            `Nb de séances` = n_distinct(cle_rsa[rsacmd == "28"])  
  ) %>%  
  tidyr::pivot_longer(everything(), #!moissor,  
                      names_to = c("Rubrique"),  
                      values_to = "Année n")
```

## Tableau [1.D.2.RTP]

### ● RSA transmis sur la période

```
rsa$rsa %>%
  group_by(moissor) %>%
  summarise(`Nb de RSA transmis` = n(),
            `Nb de RSA en CMD 90 (*)` = n_distinct(cle_rsa[rsacmd == "90"]),
            `Nb de RSA prestation inter-établissement` = n_distinct(cle_rsa[typesej == "B"]),
            `Nb de RSA en GHS 9999` = n_distinct(cle_rsa[noghs == "9999" & rsacmd != "90"]),
            `Nb de RSA séances` = n_distinct(cle_rsa[rsacmd == "28"]),
            `Nb de séances` = sum(nbseance[rsacmd == "28"]),
            `Nb de RSA DS=0` = n_distinct(cle_rsa[duree == 0 & rsacmd != "28" & rsacmd != "90"]),
            `dont Nb de J ou T0` = n_distinct(cle_rsa[duree == 0 & rsacmd != "28" & rsacmd != "90" &
rsacomp %in% c('J', 'T')]), # wrong
            `Nb de RSA hors séjour sans nuitée` = n_distinct(cle_rsa[duree > 0 & rsacmd != "90"]),
            `Nb de journées hors séjour sans nuitée` = sum(duree[rsacmd != "90"]),
            `Nb de RSA en UHCD réaffecté` = sum(uhcd == 1)
  ) %>%
  tidyr::pivot_longer(!moissor,
                      names_to = c("Rubrique"),
                      values_to = "Année n")
```

- Répartition des modes d'entrée / sortie

```
rsa$rsa %>%  
  filter(rsacmd != "90") %>%  
  mutate(mep = paste0(echpmsi, prov)) %>%  
  count(mep) # Mode entrée provenance
```

```
rsa$rsa %>%  
  filter(rsacmd != "90") %>%  
  mutate(msd = paste0(schpmsi, dest)) %>%  
  count(msd) # Mode sortie destination
```

```
rsa$rsa %>%  
  filter(rsacmd != "90") %>%  
  mutate(mep = paste0(echpmsi, prov)) %>%  
  count(mep, cdpu) %>%  
  mutate(p = n / sum(n)) # MEP + passage par les urgences
```



- Case-mix GHM

```
rsa$rsa %>%  
  group_by(anseqta, ghm) %>%  
  summarize(n = n(),  
            dms = mean(duree[duree > 0]),  
            n0 = sum(duree == 0L))
```

## Tableau [1.D.2.SUM]

- Nombre et type de suppléments par unité médicale

```
rsa$rsa_um %>%  
  filter(substr(typaut1,1,2) %in% c('01', '02', '03',  
                                   '04', '06', '13', '14',  
                                   '16', '18')) %>%  
  count(typaut = substr(typaut1,1,2), nat supp1, wt = nb supp1)
```

- Synthèse des erreurs de groupage

```
eg <- ileg_mco(p, reshape = TRUE)

group_by(eg, eg) %>%
  summarise(`Nb erreurs` = n(),
            `Nb séjours` = n_distinct(cle_rsa))
```

- Séjours valorisés

```
library(nomensland)

tarifs_ghs <- dplyr::distinct(get_table('tarifs_mco_ghs'), ghs, anseqta, .keep_all = TRUE)

tarifs_supp <- get_table('tarifs_mco_supplements') %>% mutate_if(is.numeric, tidyr::replace_na,
0) %>%
  select(-cgeo)

vrsta <- vvr_rsa(p)
vano <- vvr_ano_mco(p)
```

- Séjours valorisés

```
rsa_valo <- vvr_mco(  
  vvr_ghs_supp(rsa = vrsa,  
    tarifs = tarifs_ghs,  
    supplements = tarifs_supp,  
    ano = vano,  
    full = FALSE,  
    cgeo = 1L,  
    prudent = 1,  
    bee = FALSE),  
  vvr_mco_sv(vrsa, vano)  
)  
epmsi_mco_sv(rsa_valo)
```

## Tableau [1.V.1.RAV]

- Récapitulation activité - valorisation

```
epmsi_mco_rav(rsa_valo)
```

## Tableau [1.D.2.IPA]

- Suivi des indicateurs de pilotage des activités de soins autorisées (IPA)
- Voir programme annexe `exemple_ipa.R`
- et le lien ici : <https://toudim.netlify.app/>

## Tableau [1.V.2.VMED]

- Valorisation des molécules onéreuses et indications rejetées

```
imed_mco(p) %>% filter(indication == 'IGLOUBI')
```

```
imed_mco(p) %>% filter(indication == 'IBOULGA')
```



Requêter les données sur la base du  
module requêteur de pmeasyr

## ● Quelques mots

### > **pouvoir partager des référentiels et des méthodes pour requêter les données**

- favoriser les échanges entre établissements
- ne pas réinventer la roue
- gagner du temps
- profiter d'un écosystème riche

### > **décorrélér le programme qui requête les données de la requête elle-même**

- profiter d'un écosystème riche
- faire des requêtes des objets versionnables
- les codes CIM, CCAM, GHM ne sont plus émiétés dans les programmes

## ● CCAM

```
library(nomensland)
ccam_actes <- get_table('ccam_actes')
rsa$actes %>%
  count(cdccam) %>%
  left_join(ccam_actes, by = c('cdccam' = 'code'))
```

## ● Regroupements de GHM

```
rgp_ghm <- get_table('ghm_ghm_regroupement')
rsa$rsa %>%
  count(ghm, anseqta) %>%
  left_join(rgp_ghm, by = c('ghm', 'ansqta'))
```

## Requête : Recours exceptionnel

```
library(nomensland)

rsa <- irsa(p, typi = 6) %>% prepare_rsa()

requetes_re <- get_all_listes('Recours Exceptionnel')

lancer_requete(rsa, requetes_re)
```

# Requête : Chirurgie bariatrique

```
library(nomensland)

rsa <- irsa(p, typi = 6) %>% prepare_rsa()

requetes_bari <- get_all_listes("Chirurgie bariatrique")

lancer_requete(rsa, requetes_bari)
```

## Requête : 55 gestes marqueurs

```
library(nomensland)

rsa <- irsa(p, typi = 6) %>% prepare_rsa()

requetes_gm <- get_all_listes("Chir ambu : 55 GM")

lancer_requete(rsa, requetes_gm)
```

## Requête : ad hoc

```
library(nomensland)

rsa <- irsa(p, typi = 6) %>% prepare_rsa()

requete_complexe = list(diags = "E66",
                        positions_diags = 5,
                        dureemax = 0,
                        ghm = '28Z',
                        agemin = 59,
                        agemax = 89,
                        diags_exclus = "C")

requete(rsa, requete_complexe, vars = c('ghm', 'agean'))
```

merci de votre attention

- Pour plus d'informations, vous pouvez vous rendre sur cette page web
- <https://guillaumepressiat.github.io/>





M'en... %>% fin

