

Repérage d'une condition médicale

SQL_reperage_cond_med

Guillaume Boucher

17 novembre 2021

Installation de la librairie R *inesss*

Fichier nécessaire

Télécharger le fichier `inesss_1.x.y.tar.gz` à partir de Microsoft Teams.
[INESSS - Documentation BDCA > Documents > Outils > Librairie R INESSS > inesss_x.y.z.tar.gz.](#)

Ouvrir RStudio.

Dans la console, écrire le code suivant :

```
remotes::install_local("C:\\msXXX\\Desktop\\inesss_x.y.z.tar.gz")
```

Prendre note que les «\» sont doubles.

Si la librairie `remotes` n'est pas installée, écrire dans la console le code suivant : `install.packages("remotes")`.

Avant d'installer une librairie, il est préférable de ne pas l'avoir *appelé* auparavant.

Il est déconseillé de mettre à jour les librairies utilisées par *inesss*.

L'installation se termine lorsqu'on peut lire dans la console : *DONE (inesss)*.

Activer la librairie

Après avoir installé la librairie *inesss*, il suffit de l'activer pour avoir accès aux tables et aux fonctions :

```
library(inesss)
```

SQL_reperage_cond_med

Usage de la fonction

```
NomDeLaTable <- SQL_reperage_cond_med(  
  conn = SQL_connexion(),  
  debut, fin,  
  Dx_table,  
  CIM = c("CIM9", "CIM10"),  
  by_Dx = FALSE,  
  date_dx_var = "admis",  
  n1 = 30, n2 = 730,  
  keep_all = FALSE,  
  verbose = TRUE  
)
```

conn

La fonction `SQL_connexion()` permet à R d'avoir accès aux tables de Teradata.

Inscrire l'identifiant à l'intérieur des parenthèses et le programme demandera le mot de passe lorsque la fonction `SQL_reperage_cond_med()` sera exécutée.

```
conn = SQL_connexion("msxxx")
```

debut & fin

Indiquer le début et la fin de la période d'étude au format AAAA-MM-JJ.

```
debut = "2020-01-01", fin = "2020-12-31"
```


Dx_table

list contenant le ou les codes à analyser.

```
Dx_table = list(  
  NomDiagnostic = list(  
    CIM9 = c("Code1%", "Code2%", "Code3%"),  
    CIM10 = c("Code4%", "Code5%")  
  ),  
  ...  
)  
  
Dx_table = list(  
  mal_coron = list(  
    CIM9 = paste0(410:415, "%"),  
    CIM10 = paste0("I", 20:25, "%")  
  ),  
  ...  
)
```

CIM

Permet de sélectionner un type de code en particulier. Par exemple, si

```
CIM = c("CIM9", "CIM10") # par défaut  
CIM = "CIM9"  
CIM = "CIM10"
```

Seuls les codes identifiés CIM9 seront utilisés, et ce, même si Dx_table contient des codes CIM10.

by_Dx

Distinction des codes de diagnostics (TRUE) ou pas (FALSE) inscrits dans la table Dx_Table. Par défaut FALSE.

Supposons que

```
Dx_table = list(diab = list(CIM9 = "code1%", CIM10 = "code11%"),  
               canc = list(CIM9 = "code2%", CIM10 = "code22%"))
```

by_Dx = TRUE : on aurait jusqu'à deux lignes par individu où la colonne DIAGN pourrait indiquer les valeurs diab et canc.

by_Dx = FALSE : on considère le diabète et le cancer comme un même code à analyser, on a donc au maximum une seule ligne par individu, la colonne DIAGN est absente.

date_dx_var

Permet de choisir entre la date d'admission ou la date de départ comme date de diagnostic pour l'étude dans les vues suivantes :

- V_DIAGN_SEJ_HOSP_CM
- V_SEJ_SERV_HOSP_CM
- V_EPISO_SOIN_DURG_CM

```
date_dx_var = "admis"  # par défaut  
date_dx_var = "depar"
```

n1 & n2

Nombre de jours nécessaire entre deux diagnostics pour que le deuxième confirme le premier. Permet de construire l'intervalle $[n1, n2]$ où le nombre de jours entre deux diagnostics doit être dans cet intervalle pour être confirmé.

```
n1 = 30    # par défaut  
n2 = 730   # par défaut
```

keep_all

Par défaut FALSE. TRUE est utile si on cherche la date la plus récente de chaque individu (voir D_Recent).

```
keep_all = FALSE # Par défaut  
keep_all = TRUE
```

verbose

TRUE ou FALSE. Message de progression qui est affiché dans la console.

```
Étape 1 :  
V_DIAGN_SEJ_HOSP_CM  
- mal_coron (1.35 secs)  
V_SEJ_SERV_HOSP_CM  
- mal_coron (0.68 secs)  
  
Étape 2 :  
V_DIAGN_SEJ_HOSP_CM  
- mal_coron (1.35 secs)  
V_SEJ_SERV_HOSP_CM  
- mal_coron (0.55 secs)  
V_EPISO_SOIN_DURG_CM  
- mal_coron (2.33 secs)  
I_SMOD_SERV_MD_CM  
- mal_coron (12.01 secs)  
  
Confirmation des diagnostics...  
Arrangement de la table finale...  
FIN.
```

Table Résultante

Variables

- **ID** : Numéro d'identification de l'individu.
- **DIAGN** : Seulement si `by_Dx = TRUE`, nom du diagnostic tel qu'inscrit dans `Dx_table`.
- **DI_Finale** : Date d'incidence retenue [`min(DI_Hospit, DI_Acte)`]
- **DI_Hospit** : Date d'incidence d'hospitalisation.
- **DI_Acte** : Date d'incidence de l'acte.
- **DC_Acte** : Date de confirmation de `DI_Acte`.