

Package ‘inesss’

March 11, 2021

Title Institut National Excellence Sante Services Sociaux

Version 0.6.0

Description Cette librairie fournit des fonctionnalités pour une variété de tâches propices au domaine de la santé et des outils pour visualiser les résultats.

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports askpass (>= 1.1),
data.table (>= 1.13.4),
DBI (>= 1.1.0),
fs (>= 1.5.0),
knitr (>= 1.30),
lubridate (>= 1.7.9.2),
miniUI (>= 0.1.1.1),
parallel (>= 4.0.3),
odbc (>= 1.3.0),
readxl (>= 1.3.1),
rmarkdown (>= 2.6),
rstudioapi (>= 0.13),
shiny (>= 1.5.0),
shinydashboard (>= 0.7.1),
shinyFiles (>= 0.9.0),
stringr (>= 1.4.0),
testthat (>= 3.0.1),
writexl (>= 1.3.1)

VignetteBuilder knitr

Depends R (>= 2.10)

R topics documented:

comorbidity	2
Comorbidity_diagn_codes	3
Comorbidity_weights	4
date_ymd	5

formulaire	6
I_APME_DEM_AUTOR_CRITR_ETEN_CM	6
Pop_QC	7
query_naif_switch1	8
query_stat_gen1	10
RLS_convert	12
RLS_list	13
SQL_comorbidity	13
SQL_comorbidity_diagn	16
SQL_connexion	17
SQL_naif_switch1	18
SQL_obstetric	20
SQL_stat_gen1	21
V_DEM_PAINT_MED_CM	23
V_DENOM_COMNE_MED	24
V_DES_COD	25
V_PRODU_MED	26

Index	27
--------------	-----------

comorbidity	<i>Comorbidity</i>
-------------	--------------------

Description

Calcul des indicateurs de *Charlson*, *Elixhauser* et la combinaison des deux.

Usage

```
comorbidity(
  dt,
  ID,
  DIAGN,
  DATE_DX,
  SOURCE,
  n1 = 30,
  n2 = 730,
  method = c("Charlson", "Elixhauser"),
  scores = "CIM10",
  confirm_sourc = list(`MED-ECHO` = 1, BDCU = 2, SMOD = 2),
  keep_confirm_data = FALSE
)
```

Arguments

dt	Dataset ayant au moins les quatre (4) colonnes ID, DIAGN, DATE_DX et SOURCE.
ID	Nom de la colonne indiquant le numéro de l'utilisateur, de l'individu.
DIAGN	Nom de la colonne indiquant le code d'un diagnostic. Voir <code>names(inesss::Comorbidity_diagn_codes)</code> .
DATE_DX	Nom de la colonne indiquant la date du diagnostic.
SOURCE	Nom de la colonne indiquant la provenance du diagnostic.

n1, n2	Nombre de jours dans le but de construire l'intervalle [n1,n2]. Pour qu'un code de diagnostique soit confirmé, il faut que <i>DIAGNi</i> soit suivi de <i>DIAGNj</i> (où $i < j$) et que le nombre de jours entre les deux soit dans l'intervalle [n1,n2].
method	Méthode de calcul des indicateurs. 'Charlson' ou 'Elixhauser'. Inscire les deux crée la colonne Combined.
scores	'CIM9' ou 'CIM10'. Nom de la colonne du dataset Comorbidity_weights à utiliser pour le calcul des indicateurs.
confirm_sourc	list indiquant la <i>confiance</i> des SOURCE. Si une SOURCE doit être confirmée par une autre dans l'intervalle [n1,n2], inscrire 2, sinon 1. Inscire les sources sous le format : confirm_sourc = list(source1 = 1, source2 = 2, source3 = 2, ...). confirm_sourc doit contenir toutes les valeurs uniques de la colonne SOURCE.
keep_confirm_data	TRUE ou FALSE. Place en attribut le data confirm_data qui indique la date de repérage et la date de confirmation d'un diagnostic.

Details

confirm_sourc : Dans l'exemple confirm_sourc = list(source1=1, source2=2, source3=2, ...), la source3 pourrait confirmer la source2 et vice-versa.

Value

data.table avec les colonnes ID, Charlson (selon method), Elixhauser (selon method), Combined (selon method), et tous les codes de diagnosti indiquant leur poids.

Comorbidity_diagn_codes

Data - Codes diagnostics

Description

Codes SQL regex (se terminent par un '%') à utiliser lors de l'extraction des codes de diagnostics pour l'étude de la comorbidité.

Usage

```
data('Comorbidity_diagn_codes')
```

```
Charlson_diagn_codes
```

```
Elixhauser_diagn_codes
```

Format

```
list(nom_diagn = list(CIM9,CIM10)). 32 diagnostics.
```

```
list(nom_diagn = list(CIM9,CIM10)). 16 diagnostics.
```

```
list(nom_diagn = list(CIM9,CIM10)). 29 diagnostics.
```

Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

aids : AIDS/HIV
alcohol : Alcohol abuse
canc : Any tumor without metastasis
blane : Blood loss anemia
carit : Cardiac arrhythmias
cevd : Cerebrovascular disease
copd : Chronic pulmonary disease
coag : Coagulopathy
chf : Congestive heart failure
dane : Deficiency anemia
dementia : Dementia
depre : Depression
diab : Diabetes, complicated
diabwc : Diabetes, uncomplicated
drug : Drug abuse
fed : Fluid and electrolyte disorders
hyp : Hypertension
hypothy : Hypothyroidism
ld : Liver disease
metacanc : Metastatic cancer
mi : Myocardial infarction
nd : Neurological disorders
obes : Obesity
para : Paralysis
pvd : Peripheral vascular disorders
psycho : Psychoses
pcd : Pulmonary circulation disorders
rend : Renal disease
rheumd : Rheumatoid arth./collagen vascular disease
ud : Ulcer disease
valv : Valvular disease
wloss : Weight loss

Source

Validation of the Combined Comorbidity Index of Charlson and Elixhauser to Predict 30-Day Mortality Across ICD-9 and ICD-10. Voir PDF.

Comorbidity_weights	<i>Data - Poids des codes de diagnostics</i>
---------------------	--

Description

Data - Poids des codes de diagnostics

Usage

```
data('Comorbidity_weights')
```

Format

Tableau de 4 variables et 32 observations.

DIAGN Identification du diagnostic. character.

DIAGN_CODE Code de diagnostic utilisé dans la programmation. character.

CIM9 Poids utilisé pour la 9^e révision du CIM. integer

CIM10 Poids utilisé pour la 10^e révision du CIM. integer.

Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

Source

Validation of the Combined Comorbidity Index of Charlson and Elixhauser to Predict 30-Day Mortality Across ICD-9 and ICD-10. Voir PDF.

date_ymd	<i>Date</i>
----------	-------------

Description

Retourne une date au format AAAA-MM-JJ. Utile dans des for loop, car dd peut prendre la valeur 'last' (au lieu d'un nombre), donc pas besoin de savoir si le dernier jour du mois est le 28 ou le 29 en février, ou un 30 ou un 31 pour les autres mois.

Usage

```
date_ymd(yyyy, mm, dd)
```

Arguments

yyyy	Nombre entier indiquant l'année.
mm	Nombre entier compris entre 1 et 12, où 1 indique janvier et 12 décembre.
dd	Nombre compris entre 1 et 31 selon les mois. Pour remplacer le dernier jour du mois (28, 29, 30, 31), il est possible d'inscrire dd = 'last'.

Value

lubridate::as_date

Examples

```
date_ymd(2020, 1, 15)
date_ymd(2020, 10, 31)
date_ymd()
for (yr in 1996:2004) {
  print(date_ymd(yyyy = yr, mm = 2, dd = 'last'))
}
```

formulaire

*Formulaire Interactif***Description**

Permet d'exécuter des requêtes Excel à partir d'un formulaire interactif. La documentation complète du formulaire, *AIDE_FORMULAIRE_DATE.pdf*, est disponible [ici](#).

Usage

```
formulaire()
```

Details**Requêtes via Excel :**

Il est conseillé d'utiliser les gabarits Excel pour éviter des erreurs de structures dans les tableaux d'arguments. Les fichiers Excel sont disponibles [ici](#).

I_APME_DEM_AUTOR_CRITR_ETEN_CM

*Data - Demandes d'autorisation de Patient-Médicament d'exceptions.***Description**

Data - Demandes d'autorisation de Patient-Médicament d'exceptions.

Usage

```
data('I_APME_DEM_AUTOR_CRITR_ETEN_CM')
```

Format

```
list
```

NO_SEQ_INDCN_RECNU_PME Indique la première et la dernière année d'utilisation.

- **NO_SEQ_INDCN_RECNU** : Numéro de séquence d'indication reconnue - PME. integer.
- **DD_TRAIT_DEM** : Date de début de traitement demandée.
- **DF_TRAIT_DEM** : Date de fin de traitement demandée.
- **DD_AUTOR** : Date de début de l'autorisation PME.
- **DF_AUTOR** : Date de fin de l'autorisation PME.
- **DD_APLIC_AUTOR** : Date de début de l'applicabilité de l'autorisation de PME.
- **DF_APLIC_AUTOR** : Date de fin de l'applicabilité de l'autorisation de PME.
- **DAT_STA_DEM** : Date de création ou de mise à jour du statut d'une demande d'autorisation correspondant à l'attribution du dernier statut de la demande.

Details

Certaines dates de fin sont plus grande que la date du jour (souvent 2078-12-31). Si c'est le cas, ces dates sont modifiées par la date du jour.

Source

Dictionnaire EI

Pop_QC	<i>Data - Estimations et projections de population comparables (1996-2041)</i>
--------	--

Description

Tableau de la population québécoise par niveau géographique.

Ce fichier présente une série continue de données populationnelles comparables composée de la série des estimations (1996-2019) et de la série des projections (2020-2041) de population. Ces données tiennent compte de l'évolution de la population selon les plus récentes données observées de naissances, décès et mouvements migratoires.

Il est à noter que ces données de population sont présentées sur la base du découpage territorial du réseau de la santé et des services sociaux, soit pour les territoires suivants : le Québec, les réseaux universitaires intégrés de santé et de services sociaux (RUISSS), les régions sociosanitaires (RSS), les réseaux territoriaux de services (RTS), les réseaux locaux de services (RLS) et les centres locaux de services communautaires (CLSC).

Usage

```
data('Pop_QC')
```

Format

Tableau de 8 variables et 2 595 320 observations :

GEO Niveau géographique : Québec, RUISSS, RSS, RTS, RLS, CLSC.

CODE Code du territoire.

AN Année.

TYPE Type de données : Estimations ou Projections.

STATUT Donnée révisée ou provisoire. NA indique que la donnée n'a pas été changée depuis la dernière publication.

SEXE

AGE

POP Population

Details

Attention AGE = 90 équivaut à 90 ans et plus.

La classe des colonnes est character lorsque c'est du texte ou integer lorsque c'est un nombre.

Mise en ligne : 25 février 2016.

Dernière modification : 24 avril 2020.

Publication no : EstimProjComp-ISQ.

La fiche d'information et technique de cette base de données est disponible avec le fichier Excel (voir *Source*).

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

Source

MSSS Données de population.

Fichier Excel utilisé.

query_naif_switch1	Code SQL
--------------------	----------

Description

Générateur de code SQL pour la méthode naif_switch1.

Usage

```
query_naif_switch1(
  debut,
  fin,
  type_rx = "DENOM",
  codes,
  rx_retrospect_a_exclure = NULL,
  njours_sans_conso = 365,
  code_serv = c("1", "AD"),
  code_serv_filtre = "Exclusion",
  code_list = NULL,
  code_list_filtre = "Inclusion",
  ...
)
```

Arguments

debut	Date de début de la période d'étude au format AAAA-MM-JJ (une seule valeur).
fin	Date de fin de la période d'étude au format AAAA-MM-JJ (une seule valeur).
codes	Traitement(s) d'intérêt(s). Vecteur comprenant le ou les codes d'analyse au format numérique, sans zéros.
rx_retrospect_a_exclure	Traitement(s) à inclure dans la période rétrospective. Voir <i>Details</i> . Un individu qui a au moins un traitement durant la période rétrospective ne sera pas considéré comme <i>naïf</i> ou <i>switch</i> .
njours_sans_conso	Nombre de jours qu'un individu ne doit pas avoir reçu de traitements avant sa date de référence (index) pour être considéré <i>naïf</i> ou <i>switch</i> .
code_serv	Vecteur de type character comprenant le ou les codes de service (SMED_COD_SERV_1) à exclure ou à inclure, sinon inscrire NULL.

code_serv_filtre	'Inclusion' ou 'Exclusion' des codes de service code_serv. Inscrive code_serv = NULL s'il n'y a pas de filtre à appliquer.
code_list	Vecteur de type character comprenant le ou les codes de catégories de liste de médicaments (SMED_COD_CATG_LISTE_MED) à exclure ou à inclure, sinon inscrire NULL.
code_list_filtre	'Inclusion' ou 'Exclusion' des codes de catégories de liste de médicaments code_list. Inscrive code_list = NULL s'il n'y a pas de filtre à appliquer.
type_Rx	Indique le type de code analysé : <ul style="list-style-type: none"> • 'DENOM' : Code de dénomination commune (SMED_COD_DENOM_COMNE). • 'DIN' : Code d'identification du médicament (SMED_COD_DIN).

Details

rx_retrospect_a_exclure :
La période rétrospective est construite à partir des dates de références (index) et de l'argument njours_sans_conso : [INDEX -njours_sans_conso; INDEX -1].

code_serv_filtre, code_list_filtre :
'Exclusion' inclus les NULL
'Inclusion' exclus les NULL.

Value

Chaîne de caractères à utiliser dans une requête SQL.

Examples

```
### DENOM
cat(query_naif_switch1(
  debut = '2019-01-01', fin = '2019-12-31',
  type_rx = 'DENOM', codes = 47946,
  rx_retrospect_a_exclure = NULL, njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
))

### DIN
cat(query_naif_switch1(
  debut = '2019-01-01', fin = '2019-12-31',
  type_rx = 'DIN', codes = 2257238,
  rx_retrospect_a_exclure = NULL, njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
))

### Plusieurs codes
cat(query_naif_switch1(
  debut = '2019-01-01', fin = '2019-12-31',
  type_rx = 'DENOM', codes = c(39, 47135, 48222),
  rx_retrospect_a_exclure = NULL, njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
))
```

```

))

### rx_retrospect_a_exclure
# Plusieurs, dont celui inclus dans 'codes'
cat(query_naif_switch1(
  debut = '2019-01-01', fin = '2019-12-31',
  type_rx = 'DENOM', codes = 47946,
  rx_retrospect_a_exclure = c(47946, 48222, 48259), njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
))
# Plusieurs, sans celui inclus dans 'codes'
cat(query_naif_switch1(
  debut = '2019-01-01', fin = '2019-12-31',
  type_rx = 'DENOM', codes = 47946,
  rx_retrospect_a_exclure = c(48222, 48259, 43012), njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
))

### Exclusion vs Inclusion
cat(query_naif_switch1(
  debut = '2019-01-01', fin = '2019-12-31',
  type_rx = 'DENOM', codes = 47946,
  rx_retrospect_a_exclure = NULL, njours_sans_conso = 365,
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion',
  code_list = c('40', '41'), code_list_filtre = 'Inclusion'
))

```

query_stat_gen1

Code SQL

Description

Générateur de code SQL pour la méthode stat_gen1.

Usage

```

query_stat_gen1(
  debut,
  fin,
  type_Rx = "DENOM",
  codes,
  group_by = "Codes",
  code_serv = c("1", "AD"),
  code_serv_filtre = "Exclusion",
  code_list = NULL,
  code_list_filtre = "Inclusion",
  ...
)

```

Arguments

debut	Date de début de la période d'étude au format AAAA-MM-JJ (une seule valeur).
fin	Date de fin de la période d'étude au format AAAA-MM-JJ (une seule valeur).
type_Rx	Indique le type de code analysé : <ul style="list-style-type: none"> 'DENOM' : Code de dénomination commune (SMED_COD_DENOM_COMNE). 'DIN' : Code d'identification du médicament (SMED_COD_DIN).
codes	Vecteur comprenant le ou les codes d'analyse au format numérique, sans zéros.
group_by	Regrouper (aggréger) les résultats par : <ul style="list-style-type: none"> 'Codes' : Résultats par code analysé. 'Teneur' : Résultats par teneur de médicament (SMED_COD_TENR_MED) incluant les valeurs absentes. 'Format' : Résultats par format d'acquisition du médicament (SMED_COD_FORMA_ACQ_MED) incluant les valeurs absentes.
code_serv	Vecteur de type character comprenant le ou les codes de service (SMED_COD_SERV_1) à exclure ou à inclure, sinon inscrire NULL.
code_serv_filtre	'Inclusion' ou 'Exclusion' des codes de service code_serv. Inscrire code_serv = NULL s'il n'y a pas de filtre à appliquer.
code_list	Vecteur de type character comprenant le ou les codes de catégories de liste de médicaments (SMED_COD_CATG_LISTE_MED) à exclure ou à inclure, sinon inscrire NULL.
code_list_filtre	'Inclusion' ou 'Exclusion' des codes de catégories de liste de médicaments code_list. Inscrire code_list = NULL s'il n'y a pas de filtre à appliquer.

Details**Méthode stat_gen1 :**

Statistiques descriptives tirées de la vue V_DEM_PAINT_MED_CM.

group_by : Revient à utiliser la commande group by dans le code SQL.

code_serv_filtre, code_list_filtre :

'Exclusion' inclus les NULL

'Inclusion' exclus les NULL.

Value

Chaîne de caractères à utiliser dans une requête SQL.

Examples

```
### Avantages d'utiliser cat() si c'est pour afficher du code dans la console.
# Avec cat()
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222), group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))
# Sans cat()
```

```

query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222), group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
)

### GROUP_BY
# group_by = NULL : resultats par periode d'etude.
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = NULL,
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))
# group_by = 'Codes' : resultats par code d'analyse.
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))
# group_by = c('Teneur', 'Format') : resultats par teneur et format.
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = c('Teneur', 'Format'),
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))

### Exclusion VS Inclusion
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = NULL,
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion',
  code_list = c('40', '41'), code_list_filtre = 'Inclusion'
))

```

RLS_convert

Data - Correspondance RLS Loi 10

Description

Établir la correspondance des RLS avant et après l'adoption de la loi 10.

Usage

```
data('RLS_convert')
```

Format

Tableau de 2 variables et 84 observations :

RLS14 Code de RLS **avant** l'adoption de la loi 10.

RLS15 Code de RLS **après** l'adoption de la loi 10.

Details

Certains RLS ne peuvent être convertis, car leur valeur se retrouve avant et après l'adoption de la loi 10.

`attr(RLS_convert, "RLS_exclus")` indique les quatre (4) RLS exclus : 611, 612, 1611, 1612.

`attr(RLS_convert, "RLS_exclus_value")` renvoie un tableau indiquant les valeurs avant et après l'adoption de la loi 10 pour ces quatre (4) RLS.

L'attribut `MaJ` indique la dernière mise à jour ou la date de création du tableau.

Source

Correspondance Etablissement Public Loi 10.

Fichier Excel utilisé.

RLS_list	<i>Data - Liste des RLS</i>
----------	-----------------------------

Description

Vecteur contenant la liste des 93 RLS plus 3 valeurs utiles lors d'analyse : 1001, 1701, 1801.

Usage

```
data('RLS_list')
```

Format

Vecteur integer de 96 nombres.

Details

L'attribut `MaJ` indique la dernière mise à jour ou la date de création du tableau.

SQL_comorbidity	<i>Comorbidity</i>
-----------------	--------------------

Description

Extraction des codes de diagnostics CIM pour ensuite calculer les indicateurs de Charlson et Elixhauser.

Usage

```
SQL_comorbidity(
  conn,
  uid,
  pwd,
  dt,
  ID,
  DATE_INDEX,
  method = c("Charlson", "Elixhauser"),
  CIM = c("CIM9", "CIM10"),
  scores = "CIM10",
  lookup = 2,
  n1 = 30,
  n2 = 730,
  dt_source = c("V_DIAGN_SEJ_HOSP_CM", "V_SEJ_SERV_HOSP_CM", "V_EPISO_SOIN_DURG_CM",
    "I_SMOD_SERV_MD_CM"),
  dt_desc = list(V_DIAGN_SEJ_HOSP_CM = "MED-ECHO", V_SEJ_SERV_HOSP_CM = "MED-ECHO",
    V_EPISO_SOIN_DURG_CM = "BDCU", I_SMOD_SERV_MD_CM = "SMOD"),
  confirm_sourc = list(`MED-ECHO` = 1, BDCU = 2, SMOD = 2),
  obstetric_exclu = TRUE,
  exclu_diagn = NULL,
  verbose = TRUE,
  keep_confirm_data = FALSE
)
```

Arguments

conn	Variable contenant la connexion entre R et Teradata. Voir SQL_connexion . Évite d'utiliser les arguments uid et pwd.
uid	Nom de l'identifiant pour la connexion SQL Teradata.
pwd	Mot de passe associé à l'identifiant. Si NULL, le programme demande le mot passe. Cela permet de ne pas afficher le mot de passe dans un script.
dt	Tableau ayant au moins deux colonnes : ID et DATE_INDEX.
ID	Nom de la colonne contenant l'identifiant unique de l'utilisateur.
DATE_INDEX	Nom de la colonne contenant la date index de chaque usager.
method	Extraire les diagnostics associés aux méthodes de calcul <i>Charlson</i> , <i>Elixhauser</i> ou les deux.
CIM	'CIM9', 'CIM10' ou les deux. Permet de filtrer les codes de diagnostics selon le numéro de révision de la <i>Classification statistique internationale des maladies et des problèmes de santé connexes</i> (CIM).
scores	'CIM9' ou 'CIM10'. Nom de la colonne du dataset Comorbidity_weights à utiliser pour le calcul des indicateurs.
lookup	Nombre entier. Années à analyser avant la date index de chaque individu.
n1	Nombre de jours dans le but de construire l'intervalle [n1,n2]. Pour qu'un code de diagnostic soit confirmé, il faut que <i>DIAGNi</i> soit suivi de <i>DIAGNj</i> (où $i < j$) et que le nombre de jours entre les deux soit dans l'intervalle [n1,n2].
n2	Nombre de jours dans le but de construire l'intervalle [n1,n2]. Pour qu'un code de diagnostic soit confirmé, il faut que <i>DIAGNi</i> soit suivi de <i>DIAGNj</i> (où $i < j$) et que le nombre de jours entre les deux soit dans l'intervalle [n1,n2].

dt_source	Vecteur comprenant la ou les bases de données où aller chercher l'information. Voir <i>Details</i> .
dt_desc	list décrivant les bases de données demandées dans dt_source au format list(BD = 'MaDescription'). Voir <i>Details</i> .
confirm_sourc	list indiquant la <i>confiance</i> des SOURCE. Si une SOURCE doit être confirmée par une autre dans l'intervalle [n1,n2], inscrire 2, sinon 1. Inscrire les sources sous le format : confirm_sourc = list(source1 = 1, source2 = 2, source3 = 2, ...). confirm_sourc doit contenir toutes les valeurs uniques de la colonne SOURCE.
obstetric_exclu	TRUE ou FALSE. Si l'on doit exclure (TRUE) les diabètes et les hypertensions de type gestationnel. Voir <i>Détails</i> .
exclu_diagn	Vecteur contenant le nom du ou des diagnostics à exclure de l'analyse. Voir names(inesss::Comorbidity_diagn_codes) pour connaître les codes de diagnostics pouvant être exclus.
verbose	TRUE ou FALSE. Affiche le temps qui a été nécessaire pour extraire les diagnostics d'une source (dt_source). Utile pour suivre le déroulement de l'extraction.
keep_confirm_data	TRUE ou FALSE. Place en attribut le data confirm_data qui indique la date de repérage et la date de confirmation d'un diagnostic.

Details

conn, uid, pwd : Pour se connecter à Teradata, utiliser conn ou la combinaison uid et pwd.

dt : Si un ID a plus d'une date index, seule la première, la plus ancienne, sera conservée.

obstetric_exclu : Lorsqu'un cas de diabète ou d'hypertension a lieu 120 jours avant ou 180 jours après un événement obstétrique, on les considère de type gestationnel. Ces cas sont alors exclus de l'analyse.

dt_source :

- **V_DIAGN_SEJ_HOSP_CM** : Cette structure contient tous les diagnostics associés à un séjour hospitalier.
- **V_SEJ_SERV_HOSP_CM** : Cette structure contient les séjours dans un service effectués par l'individu hospitalisé.
- **V_EPISO_SOIN_DURG_CM** : Cette structure contient les épisodes de soins des départements d'urgence de la province.
- **I_SMOD_SERV_MD_CM** : Cette vue retourne différentes informations se rapportant aux Services rendus à l'acte par des médecins.

Value

data.table :

- ID : Colonne contenant l'identifiant unique de l'utilisateur.
- Charlson : Indicateur, seulement si method contient 'Charlson'.
- Elixhauser : Indicateur, seulement si method contient 'Elixhauser'.
- Combined : Indicateur, seulement si method contient 'Charlson' et 'Elixhauser'.
- Tous les diagnostics ainsi que leur poids (score).

SQL_comorbidity_diagn *Extraction - Codes diagn comorbidity*

Description

Extraction SQL des diagnostics pour l'étude de la comorbidité.

Usage

```
SQL_comorbidity_diagn(
  conn,
  uid,
  pwd,
  cohort,
  debut,
  fin,
  method = c("Charlson", "Elixhauser"),
  CIM = c("CIM9", "CIM10"),
  dt_source = c("V_DIAGN_SEJ_HOSP_CM", "V_SEJ_SERV_HOSP_CM", "V_EPISO_SOIN_DURG_CM",
    "I_SMOD_SERV_MD_CM"),
  dt_desc = list(V_DIAGN_SEJ_HOSP_CM = "MED-ECHO", V_SEJ_SERV_HOSP_CM = "MED-ECHO",
    V_EPISO_SOIN_DURG_CM = "BDCU", I_SMOD_SERV_MD_CM = "SMOD"),
  exclu_diagn = NULL,
  verbose = TRUE
)
```

Arguments

conn	Variable contenant la connexion entre R et Teradata. Voir SQL_connexion . Évite d'utiliser les arguments uid et pwd.
uid	Nom de l'identifiant pour la connexion SQL Teradata.
pwd	Mot de passe associé à l'identifiant. Si NULL, le programme demande le mot passe. Cela permet de ne pas afficher le mot de passe dans un script.
cohort	Cohorte d'étude. Vecteur comprenant les numéros d'identification des individus à conserver.
debut	Date de début de la période d'étude au format AAAA-MM-JJ.
fin	Date de fin de la période d'étude au format AAAA-MM-JJ.
method	Extraire les diagnostics associés aux méthodes de calcul <i>Charlson</i> , <i>Elixhauser</i> ou les deux.
CIM	'CIM9', 'CIM10' ou les deux. Permet de filtrer les codes de diagnostics selon le numéro de révision de la <i>Classification statistique internationale des maladies et des problèmes de santé connexes</i> (CIM).
dt_source	Vecteur comprenant la ou les bases de données où aller chercher l'information. Voir <i>Details</i> .
dt_desc	list décrivant les bases de données demandées dans dt_source au format list(BD = 'MaDescription'). Voir <i>Details</i> .
exclu_diagn	Vecteur contenant le nom du ou des diagnostics à exclusion de l'analyse. Voir <code>names(inesss::Comorbidity_diagn_codes)</code> pour connaître les codes de diagnostics pouvant être exclus.

verbose TRUE ou FALSE. Affiche le temps qui a été nécessaire pour extraire les diagnostics d'une source (dt_source). Utile pour suivre le déroulement de l'extraction.

Details

conn, uid, pwd : Pour se connecter à Teradata, utiliser conn ou la combinaison uid et pwd.

dt_source :

- **V_DIAGN_SEJ_HOSP_CM** : Cette structure contient tous les diagnostics associés à un séjour hospitalier.
- **V_SEJ_SERV_HOSP_CM** : Cette structure contient les séjours dans un service effectués par l'individu hospitalisé.
- **V_EPISO_SOIN_DURG_CM** : Cette structure contient les épisodes de soins des départements d'urgence de la province.
- **I_SMOD_SERV_MD_CM** : Cette vue retourne différentes informations se rapportant aux Services rendus à l'acte par des médecins.

method : Voir les listes [Comorbidity_diagn_codes](#) pour connaître les codes de diagnostics extraits.

Value

data.table de 4 variables :

- ID : Numéro d'identification de l'utilisateur.
- DATE_DX : Date de diagnostic.
- DIAGN : Code descriptif des diagnostics provenant de diagn_codes.
- SOURCE : Indique d'où provient l'information. Une valeur parmi dt_source.

SQL_connexion	<i>Connexion Teradata</i>
---------------	---------------------------

Description

Connexion entre R et SQL Teradata.

Usage

```
SQL_connexion(uid, pwd = NULL, dsn = "PEI_PRD", encoding = "latin1")
```

Arguments

uid	Identifiant.
pwd	Mot de passe. Si NULL, le mot de passe est demandé lors de l'exécution.
dsn	Data Source Name. Par défaut 'PEI_PRD'.
encoding	'latin1' ou 'UTF-8'. Encodage de la base de données. Par défaut 'latin1'.

Value

Connexion Teradata, sinon NULL.

Examples

```
## Not run:
conn <- SQL_connexion('abc007')
conn <- SQL_connexion(uid = 'abc007', pwd = 'MonMotDePasse', dsn = 'PEI-PRD')

## End(Not run)
```

SQL_naif_switch1

Naïfs et Switchs

Description

Statistiques générales pour un ou des médicaments à partir d'une cohorte consommant ce(s) médicament(s) pour la première fois.

Un individu est considéré *naïf* lorsqu'il a un traitement pour la première fois et qu'il n'a jamais eu d'autres traitements *de la même famille*.

Un individu est considéré *switch* lorsqu'il a un traitement pour la première fois, mais qu'il a eu un autre traitement dans le passé appartenant à *la même famille*.

Vue utilisée : **V_DEM_PAINT_MED_CM**.

Usage

```
SQL_naif_switch1(
  conn,
  debut,
  fin,
  type_rx = "DENOM",
  codes,
  grouper_par = NULL,
  rx_retrospect_a_exclure = NULL,
  njours_sans_conso = 365,
  code_serv = c("1", "AD"),
  code_serv_filtre = "Exclusion",
  code_list = NULL,
  code_list_filtre = "Inclusion",
  ...
)
```

Arguments

conn	Variable contenant la connexion entre R et Teradata. Voir SQL_connexion .
debut	Date de début de la période d'étude au format AAAA-MM-JJ (une seule valeur).
fin	Date de fin de la période d'étude au format AAAA-MM-JJ (une seule valeur).
codes	Traitement(s) d'intérêt(s). Vecteur comprenant le ou les codes d'analyse au format numérique, sans zéros.
grouper_par	Regrouper (aggréger) les résultats par : <ul style="list-style-type: none"> 'Codes' : Résultats par code analysé.

rx_retrospect_a_exclure	Traitement(s) à inclure dans la période rétrospective. Voir <i>Details</i> . Un individu qui a au moins un traitement durant la période rétrospective ne sera pas considéré comme <i>naïf</i> ou <i>switch</i> .
njours_sans_conso	Nombre de jours qu'un individu ne doit pas avoir reçu de traitements avant sa date de référence (index) pour être considéré <i>naïf</i> ou <i>switch</i> .
code_serv	Vecteur de type character comprenant le ou les codes de service (SMED_COD_SERV_1) à exclure ou à inclure, sinon inscrire NULL.
code_serv_filtre	'Inclusion' ou 'Exclusion' des codes de service code_serv. Inscrire code_serv = NULL s'il n'y a pas de filtre à appliquer.
code_list	Vecteur de type character comprenant le ou les codes de catégories de liste de médicaments (SMED_COD_CATG_LISTE_MED) à exclure ou à inclure, sinon inscrire NULL.
code_list_filtre	'Inclusion' ou 'Exclusion' des codes de catégories de liste de médicaments code_list. Inscrire code_list = NULL s'il n'y a pas de filtre à appliquer.

Details

rx_retrospect_a_exclure :
 La période rétrospective est construite à partir des dates de références (index) et de l'argument njours_sans_conso : [INDEX -njours_sans_conso; INDEX -1].

code_serv_filtre, code_list_filtre :
 'Exclusion' inclus les NULL
 'Inclusion' exclus les NULL.

Value

data.table contenant certaines de ces colonnes selon les cas :

- DATE_DEBUT : Date(s) de début de la période d'étude.
- DATE_FIN : Date(s) de fin de la période d'étude.
- MNT_MED : Montant autorisé par la RAMQ pour le médicament ou le produit. Il comprend la part du grossiste (s'il y a lieu) et la part du manufacturier.
- MNT_SERV : Montant de frais de service autorisé par la RAMQ à la date du service.
- MNT_TOT : Somme des variables MNT_MED et MNT_SERV.
- COHORTE : Nombre d'individus unique.
- NBRE_RX : Nombre de prescriptions/paiements.
- QTE_MED : Quantité totale des médicaments ou des fournitures dispensés.
- DUREE_TX : Durée de traitement totale des prescriptions en jours.
- DENOM ou DIN : Code(s) analysé(s) à l'intérieur de la période d'étude.
- RX_RETROSPECT_A_EXCLURE : Code(s) de médicament(s) qui n'ont *jamais* été consommé(s) durant la période rétrospective.
- NJOURS_SANS_CONSO : Nombre de jours qu'un individu ne doit pas avoir reçu de traitements avant sa date de référence (index) pour être considéré comme *naïf* ou *switch*.

Examples

```
## Not run:
### DENOM
dt1 <- SQL_naif_switch1(
  conn, debut = c('2018-01-01', '2019-01-01'), fin = c('2018-12-31', '2019-12-31'),
  type_rx = 'DENOM', codes = c(47873, 47958, 48213), grouper_par = 'Codes',
  rx_retrospect_a_exclure = NULL, njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
)
### GROUPER_PAR = NULL - Plusieurs codes pour un meme type de traitement
dt2 <- SQL_naif_switch1(
  conn, debut = c('2018-01-01', '2019-01-01'), fin = c('2018-12-31', '2019-12-31'),
  type_rx = 'DENOM', codes = c(47873, 47958, 48213), grouper_par = NULL,
  rx_retrospect_a_exclure = NULL, njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
)
### DIN
dt3 <- SQL_naif_switch1(
  conn, debut = c('2018-01-01', '2019-01-01'), fin = c('2018-12-31', '2019-12-31'),
  type_rx = 'DIN', codes = 2241927, grouper_par = NULL,
  rx_retrospect_a_exclure = NULL, njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
)
### RX_RETROSPECT_A_EXCLUDE conteant plus de valeurs que CODES (avec ou sans les memes valeurs)
dt4 <- SQL_naif_switch1(
  conn, debut = c('2018-01-01', '2019-01-01'), fin = c('2018-12-31', '2019-12-31'),
  type_rx = 'DIN', codes = c(2257238, 2272903), grouper_par = NULL,
  rx_retrospect_a_exclure = c(2042479, 2042487, 2257238, 2272903, 2317192, 2317206),
  njours_sans_conso = 365,
  code_serv = '1', code_serv_filtre = 'Exclusion',
  code_list = NULL, code_list_filtre = 'Inclusion'
)

## End(Not run)
```

SQL_obstetric

Extraction - Codes diagn obstetriques

Description

Extraction des événements obstétriques.

Usage

```
SQL_obstetric(
  conn,
  uid,
  pwd,
  cohort,
  debut,
```

```

    fin,
    CIM = c("CIM9", "CIM10"),
    dt_source = c("V_DIAGN_SEJ_HOSP_CM", "V_SEJ_SERV_HOSP_CM", "V_EPISO_SOIN_DURG_CM",
                  "I_SMOD_SERV_MD_CM"),
    dt_desc = list(V_DIAGN_SEJ_HOSP_CM = "MED-ECHO", V_SEJ_SERV_HOSP_CM = "MED-ECHO",
                   V_EPISO_SOIN_DURG_CM = "BDCU", I_SMOD_SERV_MD_CM = "SMOD"),
    verbose = TRUE
  )

```

Details

dt_source :

- **V_DIAGN_SEJ_HOSP_CM** : Cette structure contient tous les diagnostics associés à un séjour hospitalier.
- **V_SEJ_SERV_HOSP_CM** : Cette structure contient les séjours dans un service effectués par l'individu hospitalisé.
- **V_EPISO_SOIN_DURG_CM** : Cette structure contient les épisodes de soins des départements d'urgence de la province.
- **I_SMOD_SERV_MD_CM** : Cette vue retourne différentes informations se rapportant aux Services rendus à l'acte par des médecins.

SQL_stat_gen1

Statistiques générales

Description

Statistiques d'un ou de plusieurs codes de médicaments selon certains critères.
 Vue utilisée : **V_DEM_PAINT_MED_CM**.

Usage

```

SQL_stat_gen1(
  conn,
  debut,
  fin,
  type_Rx = "DENOM",
  codes,
  group_by = "Codes",
  code_serv = c("1", "AD"),
  code_serv_filtre = "Exclusion",
  code_list = NULL,
  code_list_filtre = "Inclusion"
)

```

Arguments

conn	Variable contenant la connexion entre R et Teradata. Voir SQL_connexion .
debut	Date de début de la période d'étude au format AAAA-MM-JJ (une seule valeur).
fin	Date de fin de la période d'étude au format AAAA-MM-JJ (une seule valeur).

type_Rx	Indique le type de code analysé : <ul style="list-style-type: none"> 'DENOM' : Code de dénomination commune (SMED_COD_DENOM_COMNE). 'DIN' : Code d'identification du médicament (SMED_COD_DIN).
codes	Vecteur comprenant le ou les codes d'analyse au format numérique, sans zéros.
group_by	Regrouper (aggréger) les résultats par : <ul style="list-style-type: none"> 'Codes' : Résultats par code analysé. 'Teneur' : Résultats par teneur de médicament (SMED_COD_TENR_MED) incluant les valeurs absentes. 'Format' : Résultats par format d'acquisition du médicament (SMED_COD_FORMA_ACQ_MED) incluant les valeurs absentes.
code_serv	Vecteur de type character comprenant le ou les codes de service (SMED_COD_SERV_1) à exclure ou à inclure, sinon inscrire NULL.
code_serv_filtre	'Inclusion' ou 'Exclusion' des codes de service code_serv. Inscrire code_serv = NULL s'il n'y a pas de filtre à appliquer.
code_list	Vecteur de type character comprenant le ou les codes de catégories de liste de médicaments (SMED_COD_CATG_LISTE_MED) à exclure ou à inclure, sinon inscrire NULL.
code_list_filtre	'Inclusion' ou 'Exclusion' des codes de catégories de liste de médicaments code_list. Inscrire code_list = NULL s'il n'y a pas de filtre à appliquer.

Details

code_serv_filtre, code_list_filtre :
 'Exclusion' inclus les NULL
 'Inclusion' exclus les NULL.

Value

data.table contenant certaines de ces colonnes selon les cas :

- DATE_DEBUT : Date(s) de début de la période d'étude.
- DATE_FIN : Date(s) de fin de la période d'étude.
- DENOM ou DIN : Code(s) analysé(s) à l'intérieur de la période d'étude.
- NOM_DENOM ou NOM_MARQ_COMRC : Nom de la dénomination commune ou le nom de la marque de commerce.
- TENEUR : Teneur du médicament.
- FORMAT_ACQ : Format d'acquisition du médicament.
- MNT_MED : Montant autorisé par la RAMQ pour le médicament ou le produit. Il comprend la part du grossiste (s'il y a lieu) et la part du manufacturier.
- MNT_SERV : Montant de frais de service autorisé par la RAMQ à la date du service.
- MNT_TOT : Somme des variables MNT_MED et MNT_SERV.
- COHORTE : Nombre d'individus unique.
- NBRE_RX : Nombre de prescriptions/paiements.
- QTE_MED : Quantité totale des médicaments ou des fournitures dispensés.
- DUREE_TX : Durée de traitement totale des prescriptions en jours.

Examples

```
## Not run:
### GROUP_BY
# Aucun group by
dt1 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = NULL
)
# Codes - DENOM
dt2 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = 'Codes'
)
# Codes - DIN
dt3 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DIN', codes = c(30848, 585092), group_by = 'Codes'
)
# Codes et Format
dt4 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = c('Codes', 'Format')
)
# Teneur
dt5 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = 'Teneur'
)

### Exclusion & Inclusion
dt6 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion',
  code_list = c('03', '40', '41'), code_list_filtre = 'Inclusion'
)

## End(Not run)
```

V_DEM_PAIMT_MED_CM	<i>Data</i>
--------------------	-------------

Description

Base de données sur les demandes de paiement de médicaments.

Usage

```
data('V_DEM_PAIMT_MED_CM')
```

Format

list de trois (3) éléments :

COD_DIN Description des codes d'identification du médicament :

- DIN : Code d'identification du médicament. integer.
- DEBUT : Première année où le code a été inscrit dans la base de données. integer.
- FIN : Dernière année où le code a été inscrit dans la base de données. integer.

COD_SERV Description et années d'utilisation des codes de service. NA indique que le code n'a pas été utilisé.

- COD_SERV : Code de service. character.
- SERV_1 : Première et dernière année que le code de service a été inscrit dans la colonne **SMED_COD_SERV_1**. character.
- SERV_2 : Première et dernière année que le code de service a été inscrit dans la colonne **SMED_COD_SERV_2**. character.
- SERV_3 : Première et dernière année que le code de service a été inscrit dans la colonne **SMED_COD_SERV_3**. character.
- COD_SERV_DESC : Description du code de service. character.

COD_STA_DECIS Codes de statut de décision qui existent dans la base de données **V_DEM_PAINT_MED_CM**.

- COD_STA_DECIS: Code de statut de décision. character.
- DEBUT : Première année où le code a été inscrit dans la base de données. integer.
- FIN : Dernière année où le code a été inscrit dans la base de données. integer.

COD_DENOM_COMNE Codes de dénominations communes qui existent dans la base de données **V_DEM_PAINT_MED_CM**.

- DENOM : Code de dénomination commune. character.
- DEBUT : Première année où le code a été inscrit dans la base de données. integer.
- FIN : Dernière année où le code a été inscrit dans la base de données. integer.

Details

L'attribut MaJ indique la dernière mise à jour ou la date de création

Source

Dictionnaire EI

V_DENOM_COMNE_MED	<i>Data</i>
-------------------	-------------

Description

Description des codes de dénomination commune.

Usage

```
data('V_DENOM_COMNE_MED')
```


Format

Tableau de 7 variables :

DENOM Code de dénomination commune (NMED_COD_DENOM_COMNE). character.

DATE_DEBUT Date à laquelle cette dénomination commune est apparue pour la première fois (NMED_DD_DENOM_COMNE). Date.

DATE_FIN Date à laquelle la dénomination commune a cessé d'être utilisée (NMED_DF_DENOM_COMNE). Date.

NOM_DENOM Nom de la dénomination commune du médicament (NMED_NOM_DENOM_COMNE). character.

NOM_DENOM_SYNON Synonyme du nom de la dénomination commune du médicament (NMED_NOM_DENOM_COMNE_SYNON). character.

NOM_DENOM_ANGLAIS Nom anglais de la dénomination commune du médicament (NMED_NOM_ANGL_DENOM_COMNE). character.

NOM_DENOM_SYNON_ANGLAIS Synonyme du nom anglais de la dénomination commune du médicament (NMED_NOM_ANGL_DENOM_SYNON). character.

Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

Source

[Dictionnaire EI.](#)

V_DES_COD	<i>Data</i>
-----------	-------------

Description

Domaine de valeurs pour les différents codes de l'environnement informationnel.

Usage

```
data('V_DES_COD')
```

Format

Tableau de 5 variables :

CODE Valeurs codifiées que peut prendre un élément (CODE_VAL_COD). character.

TYPE_CODE Nom identifiant un élément de données (CODE_NOM_COD). character.

CODE_DESC Description du code (CODE_DES). character.

DATE_DEBUT Date de début de la période d'application (CODE_DD_DES_COD). Date.

DATE_FIN Date de fin de la période d'application (CODE_DF_DES_COD). Date.

Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

Source

Dictionnaire EI.

V_PRODU_MED	<i>Data</i>
-------------	-------------

Description

Produit qui peut faire l'objet d'une facturation. Règle générale, c'est un médicament conçu par un fabricant.

Usage

```
data('V_PRODU_MED')
```

Format

Tableau de 5 variables :

- NOM_MARQ_COMRC** Nom sous lequel est commercialisé un produit pharmaceutique.
- DENOM : Code de dénomination commune (NMED_COD_DENOM_COMNE). character.
 - DIN : Code d'identification du médicament (NMED_COD_DIN). integer.
 - NOM_MARQ_COMRC : Nom sous lequel est commercialisé un produit pharmaceutique (NMED_NOM_MARQ_COMRC). character.
 - DATE_DEBUT : Date d'entrée en vigueur de la mise à jour à laquelle est relié l'ajout ou la modification de cette occurrence (NMED_DD_PRODU_MED). Date.
 - DATE_FIN : Date d'entrée en vigueur de la mise à jour **moins un jour** de l'occurrence suivante (NMED_DF_PRODU_MED). Date.

Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

Source

Dictionnaire EI.

Index

* datasets

- Comorbidity_diagn_codes, [3](#)
- Comorbidity_weights, [4](#)
- I_APME_DEM_AUTOR_CRITR_ETEN_CM, [6](#)
- Pop_QC, [7](#)
- RLS_convert, [12](#)
- RLS_list, [13](#)
- V_DEM_PAIMT_MED_CM, [23](#)
- V_DENOM_COMNE_MED, [24](#)
- V_DES_COD, [25](#)
- V_PRODU_MED, [26](#)

- Charlson_diagn_codes
 - (Comorbidity_diagn_codes), [3](#)
- comorbidity, [2](#)
- Comorbidity_diagn_codes, [3](#), [17](#)
- Comorbidity_weights, [4](#)

- date_ymd, [5](#)

- Elixhauser_diagn_codes
 - (Comorbidity_diagn_codes), [3](#)

- formulaire, [6](#)

- I_APME_DEM_AUTOR_CRITR_ETEN_CM, [6](#)

- Pop_QC, [7](#)

- query_naif_switch1, [8](#)
- query_stat_gen1, [10](#)

- RLS_convert, [12](#)
- RLS_list, [13](#)

- SQL_comorbidity, [13](#)
- SQL_comorbidity_diagn, [16](#)
- SQL_connexion, [14](#), [16](#), [17](#), [18](#), [21](#)
- SQL_naif_switch1, [18](#)
- SQL_obstetric, [20](#)
- SQL_stat_gen1, [21](#)

- V_DEM_PAIMT_MED_CM, [23](#)
- V_DENOM_COMNE_MED, [24](#)
- V_DES_COD, [25](#)
- V_PRODU_MED, [26](#)