

Package: pmeasyr (via r-universe)

September 10, 2024

Type Package

Title Donnees PMSI avec R

Version 0.4.0

Date 2023-04-24

Imports readr, tidyr, stringr, dplyr, tibble, lubridate, magrittr,
forcats, purrr, sjlabelled, knitr, DBI, jsonlite, sqldf, glue,
scales

Author Guillaume Pressiat

Depends R (>= 3.5.0)

Maintainer Guillaume Pressiat <guillaume.pressiat@chu-brest.fr>

Description Import de donnees PMSI. Gestion des archives. Formats
depuis 2011. Connexion et interface avec une db. requetr.
Valorisation des rsa, des rapss.

License GPL-2 | file LICENSE

LazyData TRUE

RoxygenNote 7.1.0

VignetteBuilder knitr

URL <https://guillaumepressiatgithub.io/pmeasyr/>

BugReports <https://github.com/GuillaumePressiat/pmeasyr/issues>

Encoding UTF-8

Suggests testthat

Repository <https://guillaumepressiat.r-universe.dev>

RemoteUrl <https://github.com/GuillaumePressiat/pmeasyr>

RemoteRef HEAD

RemoteSha f18482d28c009623a2bb0778187130ac56260248

Contents

adelete	4
adezip	5
adezip2	8
adezip3	9
ana_r_ca_gestes_marqueurs	10
ana_r_ghm_ambu_dms	11
astat	12
collect_rsa_from_db	14
creer_archives_vides	15
creer_archive_vider	15
creer_json	16
creer_nom_archive	17
db_generique	17
db_had_out	18
db_liste_tables	19
db_mco_in	20
db_mco_out	21
db_psy_out	22
db_rsf_out	23
db_ssr_out	24
dico	25
enrobeur	25
epmsi_mco_rae	26
epmsi_mco_rav	27
epmsi_mco_rav2	28
epmsi_mco_rav_rum	28
epmsi_mco_sv	29
epmsi_mco_vmed_f	30
extraire_types_fichiers	31
formats	31
iano_had	32
iano_mco	33
iano_psy	34
iano_rafael	35
iano_ssr	36
idiap	37
idmi_mco	38
iium	39
iium_corresp	40
iium_ssr	41
iivg_mco	42
ileg_had	43
ileg_mco	44
ileg_ssr	45
imed_had	46
imed_mco	47

imed_ssr	48
inner_tra	49
ipie	50
ipo	51
ir3a	52
iraa	53
irafael	54
irapss	55
irha	56
irhs	57
irps	58
irpsa	59
irsa	60
irsf	62
irum	63
issrha	64
itpt_mco	65
itra	67
labeleasier	68
lancer_requete	69
lancer_requete_db	70
noyau_pmeasyr	71
noyau_skeleton	72
parse_noms_fichiers	73
parse_nom_fichier	74
prepare_rsa	75
renomme_fichier	76
requete	77
requete_db	78
selectionne_archive	79
selectionne_fichiers	80
tbl_had	80
tbl_mco	81
tbl_psy	82
tbl_rsf	83
tbl_ssr	84
tdiag	84
vvr_ano_mco	85
vvr_ghs_supp	87
vvr_had_gh	88
vvr_libelles_valo	90
vvr_mco	91
vvr_mco_sv	92
vvr_rsa	93
vvr_rsa_hors_periode	94
vvr_rum	95
vvr_rum_check_rubriques_rav	96

adelete ~ *.zip - *Suppression des fichiers en fin de traitement*

Description

Supprime les fichiers de l'archive PMSI dezipés en debut de traitement

Usage

```
adelete(finess, annee, mois, path, liste, type, ...)
```

Arguments

finess	Finess du fichier a supprimer
annee	Annee du fichier
mois	Mois du fichier
path	Chemin d'accès aux fichiers
liste	Liste des fichiers a effacer : par default a "", efface tous les fichiers finess.annee.mois.
type	Type de fichier In / Out : par default a "", efface tous les fichiers finess.annee.mois.

Author(s)

G. Pressiat

See Also

[adezip](#), [adezip2](#), [astat](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
adelete('750712184',2016,2, path = '~/Exemple', liste = c("rss","ano"), type = "in")

adelete('750712184',2016,2, path = '~/Exemple')

## End(Not run)
```

```
adezip ~ *.zip - Identifie et dezippe des fichiers de l'archive PMSI
```

Description

Recherche et dezippe (décompresse) les fichiers contenus dans une archive **.in* ou **.out* du PMSI en fonction de parametres.

Usage

```
adezip(...)

## S3 method for class 'pm_param'
adezip(.params, ...)

## Default S3 method:
adezip(
  finess,
  annee,
  mois,
  path,
  liste = NULL,
  pathto = path,
  type = "in",
  recent = TRUE,
  nom_archive = NULL,
  quiet = FALSE,
  ...
)
```

Arguments

...	Paramètres supplémentaires. Permet par exemple de changer un des paramètres après avoir passé un noyau de paramètres sans changer le noyau de paramètres.
.params	Un noyau de paramètres définis par la fonction fonction noyau_pmeasyr
finess	Finess de l'archive.
annee	Année de l'archive.
mois	Mois de l'archive.
path	Chemin d'accès au répertoire contenant l'archive à décompresser
liste	Vecteur de caractère avec le type de fichiers à dézipper (ex: ano, rss, rsa, dmi, ...). Par défaut, liste = "" dezippe la totalite de l'archive.
pathto	Par defaut la même valeur que path, dézipper dans le même répertoire que l'archive, sinon préciser le chemin ou dezipper les fichiers dans le répertoire indiqué par pathto.
type	Type de l'archive : <i>in</i> ou <i>out</i> .

recent	par défaut TRUE, l'archive la plus recente sera utilisee, sinon propose à l'utilisateur de choisir quelle archive dezipper
nom_archive	Nom de l'archive à décompresser dans le dossier path. Par défaut, NULL, n'utilise pas ce paramètre. Si le chemin est spécifié, alors les paramètres finess, annee, mois et recent ne sont pas utilisés.
quiet	Affichage d'un message au dézipping TRUE / FALSE

Details

Il est possible de passer directement les paramètres permettant d'identifier l'archive à dézipper (méthode par défaut) ou à l'aide de paramètres enregistrés dans un noyau de paramètres (voir fonction [noyau_pmeasyr](#)).

Value

Les chemins d'accès des fichiers décompressés, de manière invisible.

Author(s)

G. Pressiat

See Also

[adezip2](#), [astat](#), [adelete](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)
[utils::unzip\(\)](#)

Examples

```
## Not run:
# Chemin vers un dossier temporaire
tmp_dir <- tempdir()

# Chemin vers un dossier contenant des archives simulées
dossier_archives <- system.file("extdata", "test_data", "test_adezip", package = "pmeasyr")

# Décompresser en fonction du finess, année et mois du
# fichier med d'une archive out
adezip('123456789', 2016, 2,
      path = dossier_archives,
      liste = 'med',
      pathto = tmp_dir,
      type = "out")

dir(tmp_dir)

# Décompresser plusieurs types de fichiers d'une archive in
adezip('123456789', 2016, 2,
      path = dossier_archives,
      liste = c('med', 'rapss', 'ano'),
      pathto = tmp_dir,
      type = "in")
```

```
dir(tmp_dir)

# Utilisation avec un noyau de paramètres
p <- noyau_pmeasyr(
  finess = '123456789',
  annee = 2016,
  mois = 12,
  path = dossier_archives,
  progress = FALSE
)

adezip(p, type = "in", pathto = tmp_dir)
dir(tmp_dir)
# Modification d'un paramètre du noyau
adezip(p, mois = 11, type = "in", pathto = tmp_dir)
dir(tmp_dir)

# Pour une même période (année/mois), il peut y avoir plusieurs archives si
# il y a eu plusieurs envois. Par exemple il y a deux version de l'archive
# out pour la période 2017.10 dans notre exemple simulé
dir(dossier_archives, pattern = "2017\\.10.*out\\.zip")

# Lorsque l'argument `recent` est `TRUE` alors la fonction adezip
# sélectionne automatiquement l'archive la plus récente
adezip(123456789, 2017, 10,
  path = dossier_archives,
  liste = 'med',
  pathto = tmp_dir,
  type = "out")
dir(tmp_dir, pattern = "med")

## End(Not run)

# Si l'argument `recent` est `FALSE` alors l'utilisateur est invité
# à choisir
## Not run:
adezip(123456789, 2017, 10,
  path = dossier_archives,
  liste = 'rsa',
  pathto = tmp_dir,
  recent = FALSE,
  type = "out")
dir(tmp_dir, pattern = "rsa")

## End(Not run)

# Dézipper les logs
## Not run:
adezip(p, type = "out", liste = c('chainage.log', 'comp.log', 'log'))

## End(Not run)
```

adezip2 ~ *.zip - Dezippe des fichiers de l'archive PMSI, avec en parametre le nom de l'archive

Description

Alternative à la fonction [adezip\(\)](#), si on connaît précisément l'archive que l'on veut utiliser.

Usage

```
adezip2(path, file, liste = "", pathto = "")
```

Arguments

path	Chemin d'accès au répertoire contenant l'archive à décompresser
file	Nom de l'archive zip (ex: 750712184.2016.2.05042016093044.in.zip)
liste	Vecteur de caractère avec le type de fichiers à dézipper (ex: ano, rss, rsa, dmi, ...). Par défaut, liste = "" dezippe la totalite de l'archive.
pathto	Par defaut la même valeur que path, dézipper dans le même répertoire que l'archive, sinon préciser le chemin ou dezipper les fichiers dans le répertoire indiqué par pathto.

Details

adezip2 est un simple wrapper autour de la fonction `adzip.default`. Cette fonction est dépréciée. En utilisant le paramètre `nom_archive` avec la fonction [adezip\(\)](#) on obtient le même résultat.

Author(s)

G. Pressiat

See Also

[adezip](#), [astat](#), [adelete](#)

Examples

```
# Chemin vers un dossier temporaire
tmp_dir <- tempdir()

# Chemin vers un dossier contenant des archives simulées
dossier_archives <- system.file("extdata", "test_data", "test_adezip", package = "pmeasyr")

# Décompresser en fonction du finess, année et mois du
# fichier med d'une archive out
adezip2(path = dossier_archives,
        file = "123456789.2017.7.21082017091715.out.zip",
        liste = 'med',
```

```

        pathto = tmp_dir)

dir(tmp_dir, pattern = "2017\\.7.*med")

```

```

adezip3      ~ *.zip - Dezippe des fichiers de l'archive PMSI en provenance de
              l'Intranet AP-HP, avec en parametre le nom de l'archive

```

Description

Version de la fonction [adezip2](#) pour des archives au format Intranet du DIM Siege de l'AP-HP, <http://dime.aphp.fr/>.

Usage

```
adezip3(finess, path, file, liste = "", pathto = "")
```

Arguments

finess	Finess du fichier a dezipper
path	Chemin d'accès au fichier
file	Nom de l'archive zip (ex: 'MCO_IN_00000_201603.zip')
liste	des fichiers a dezipper ex: ano, rss, rsa, dmi, ... ; si liste = "", dezippe la totalite de l'archive
pathto	Chemin ou déposer les fichiers dezippes, par défaut à "", les fichiers sont mis la ou se trouve l'archive

Author(s)

G. Pressiat

See Also

[adezip2](#), [adezip](#), [astat](#), [adelete](#)

Examples

```

## Not run:
# Fichier ano
adezip3(path = '~/Downloads',
        file = 'MCO_IN_00000_201603.zip',
        liste = 'ano')

# Totalité de l'archive
adezip3(path = '~/Downloads',
        file = 'MCO_IN_00000_201603.zip',
        liste = '')

## End(Not run)

```



```

list(an = 2013, moi = 12),
list(an = 2014, moi = 12),
list(an = 2015, moi = 12),
list(an = 2016, moi = 12),
list(an = 2017, moi = 12),
list(an = 2018, moi = 12),
list(an = 2019, moi = 11))

result <- periodes %>% purrr::map_dfr(ana_r_ca_gestes_marqueurs, p = p, gestes_marqueurs = lgm)
result <- result %>% arrange(`Geste marqueur`, `Période`)

pivot_result <- result %>%
  select(`Geste marqueur`, nofiness, taux_ambu, `Nb total`, `Période`) %>%
  mutate(stat = paste0(scales::percent(taux_ambu), '(', `Nb total`, ')')) %>%
  select(-taux_ambu, - `Nb total`) %>%
  tidyr::spread(`Période`, stat, '')

## End(Not run)

```

ana_r_ghm_ambu_dms *~ ANA - Analyse des taux ambulatoires et DMS sur un périmètre GHM à définir*

Description

~ ANA - Analyse des taux ambulatoires et DMS sur un périmètre GHM à définir

Usage

```
ana_r_ghm_ambu_dms(p, periode = list(an = 2018, moi = 12), requete)
```

Arguments

p	Noyau de paramètres
periode	paramètres année et mois de l'envoi
requete	liste de requêtes à définir (nomensland)

Value

Taux ambulatoire et DMS > 0 nuit par finess géographique, au globa, par racine et par DA

Examples

```

## Not run:

library(dplyr, warn.conflicts = FALSE)
library(pmeasyr)

p <- noyau_pmeasyr(finess = '290000017',
                  annee = 2018,
                  mois = 12,
                  path = '~/Documents/data/mco',
                  tolower_names = TRUE,
                  n_max = Inf)

library(nomensland)

ghmc_7r <- get_liste('chir_ambu_ghm_C_7_racines')

periodes <- list(
  list(an = 2013, moi = 12),
  list(an = 2014, moi = 12),
  list(an = 2015, moi = 12),
  list(an = 2016, moi = 12),
  list(an = 2017, moi = 12),
  list(an = 2018, moi = 12),
  list(an = 2019, moi = 11))

result <- periodes %>% purrr::map_dfr(ana_r_ghm_ambu_dms, p = p, requete = ghmc_7r)
result <- result %>% arrange(niveau, Requete, `Période`)

pivot_result <- result %>%
  select(niveau, Requete, nofiness, taux_ambu, `Nb total`, `Période`) %>%
  mutate(stat = paste0(scales::percent(taux_ambu), ' (', `Nb total`, ')')) %>%
  select(-taux_ambu, - `Nb total`) %>%
  tidyr::spread(`Période`, stat, '')

ghmk <- list(nom = 'GHM K', rsatype = 'K', Thematique = 'GHM K', abregé = 'ghmk')

lancer_requete(rsa, ghmk)

## End(Not run)

```

Description

Pour lister sans dezipper les fichiers d'une archive.

Usage

```
astat(path, file, view = TRUE)
```

Arguments

path	Chemin d'accès au dossier contenant les archives
file	Nom du fichier d'archive
view	par défaut TRUE : affiche la liste avec <code>utils::View()</code> Sinon retourne la table dans la console.

Value

Une `data.frame` contenant les colonnes :

- *Name* Nom du fichier
- *Date* Date de modification
- *Taille (Mo)*

Author(s)

G. Pressiat

See Also

`adezip()` et `adezip2()` permettent de décompresser les fichiers. Cette fonction est basée sur `utils::unzip()` en utilisant l'argument `list = TRUE`

Examples

```
## Not run:  
test_files_dir <- system.file("extdata", "test_data", "test_adezip", package = "pmeasyr")  
astat(path = test_files_dir, file = "123456789.2016.2.15032016152413.in.zip", view = FALSE)  
  
## End(Not run)
```

collect_rsa_from_db ~ req : collecter les rsa présents dans une db

Description

On selectionne certaines variables et on en cree d'autres utiles lors de l'execution de la requete

On selectionne certaines variables et on en cree d'autres utiles lors de l'execution de la requete

Usage

```
collect_rsa_from_db(con, an, n = Inf)
```

```
collect_rsa_from_db(con, an, n = Inf)
```

Arguments

con	Connexion à la base de données
an	année des rsa (ex: 17)
n	Nombre de lignes à importer (équivalent 'head')

Value

un liste rsa similaire à un import de pmeasyr

un liste rsa similaire à un import de pmeasyr

Author(s)

G. Pressiat

G. Pressiat

Examples

```
## Not run:  
rsa <- collect_rsa_from_db(con, 16, n = 1e5)
```

```
## End(Not run)
```

```
## Not run:  
rsa <- collect_rsa_from_db(con, 16, n = 1e5)
```

```
## End(Not run)
```

creer_archives_vides *Créer des archives simulées à partir de plusieurs archives réelles*

Description

Créer des archives simulées à partir de plusieurs archives réelles

Usage

```
creer_archives_vides(chemins_archives)
```

Arguments

chemins_archives

Chemin vers les archives qui servira de modèle pour l'archive simulée

Value

NULL

See Also

[creer_archive_vider\(\)](#)

creer_archive_vider *Créer une archive vide à partir d'une archive réelle*

Description

Créer une archive vide à partir d'une archive réelle

Usage

```
creer_archive_vider(  
  chemin_archive,  
  dossier_cible = "archive_simul/",  
  finess_simul = "123456789"  
)
```

Arguments

chemin_archive Chemin vers une archive réelle qui servira de modèle pour l'archive simulée

dossier_cible Chemin vers le dossier où l'archive simulée sera stockée

finess_simul Un numéro finess pour le fichier simulé

Value

NULL

creer_json *~ req : creer un json pour partager une liste*

Description

On selectionne certaines variables et on en cree d'autres utiles lors de l'execution de la requete

On selectionne certaines variables et on en cree d'autres utiles lors de l'execution de la requete

Usage

```
creer_json(requete, chemin)
```

```
creer_json(requete, chemin)
```

Arguments

requete	'list' qui contient la requête
chemin	Nom du fichier json qui sera créé

Value

un objet de classe rsa de pmeasyr

un objet de classe rsa de pmeasyr

Author(s)

G. Pressiat

G. Pressiat

Examples

```
## Not run:
liste = list(nom_abrege = "pac",
            thematique = "test",
            actes = c('EBLA003', 'EQLF002'),
            dureemax = 0,
            auteur = "John Doe",
            date_saisie = "1960-01-01")
creer_json(liste, paste0('~/Documents/listes/', liste$nom_abrege, '.json'))
```

```
## End(Not run)
```

```
## Not run:
liste = list(nom_abrege = "pac",
            thematique = "test",
            actes = c('EBLA003', 'EQLF002'),
            dureemax = 0,
            auteur = "John Doe",
```

```

        date_saisie = "1960-01-01")
    creer_json(liste, paste0('~/Documents/listes/', liste$nom_abrege, '.json'))

## End(Not run)

```

creer_nom_archive *Creer le nom d'une archive Permet de simuler ou de créer des noms d'archives PMSI*

Description

Creer le nom d'une archive Permet de simuler ou de créer des noms d'archives PMSI

Usage

```
creer_nom_archive(finess, annee, mois, type = "in", horodatage = NULL)
```

Arguments

finess	Finess de l'archive.
annee	Année de l'archive.
mois	Mois de l'archive.
type	Type de l'archive : <i>in</i> ou <i>out</i> .
horodatage	Une date au format <i>dmYHMS</i>

Value

Un nom d'archive compatible

db_generique *~ db - Copier un tibble dans une db*

Description

Copier une table R dans une db
 Copier une table R dans une db

Usage

```

db_generique(con, an, table, prefix, suffix, indexes = list(), remove = T)
db_generique(con, an, table, prefix, suffix, indexes = list(), remove = T)

```

Arguments

con	la connexion a la base de donnees (src_...)
an	l'annee pmsi
table	La table R (tibble) a copier dans la db
prefix	prefixe de la table dans la db (ex : mco, rsf, ssr, ...)
suffix	suffixe de la table dans la db (ex : rum_rum, rha_actes, rapss_rapss, ...)
indexes	index a ajouter a la table dans la base (voir copy_to)

Details

La tables déjà importée dans R est copiée dans la db

La tables déjà importée dans R est copiée dans la db

Value

nothing

nothing

Examples

```
## Not run:
purrr::quietly(db_generique)(con, 16, ma_table, 'had', 'rapss_ano') -> statuts ; gc(); #
# Result in db : had_16_rapss_ano

## End(Not run)
## Not run:
purrr::quietly(db_generique)(con, 16, ma_table, 'had', 'rapss_ano') -> statuts ; gc(); #
# Result in db : had_16_rapss_ano

## End(Not run)
```

db_had_out

~ db - Copier les rapss dans une db

Description

Copier les rapss, les actes, les diagnostics et la table ano des rapss dans une db

Usage

```
db_had_out(con, p, remove = T, zip = T, indexes = list(), ...)
```

Arguments

con	la connexion a la base de donnees (src_..)
p	le noyau pmeasyr
remove	a TRUE, les tables precedentes rapss de l'annee sont effacees avant
zip	a TRUE les fichiers des archives sont dezippees et effaces apres integration dans la db
indexes	index a ajouter a la table dans la base (voir copy_to)

Details

Les tables sont importées dans R puis copiées dans la db. Le tra est ajouté aux tables.

Value

nothing

Examples

```
## Not run:
purrr::quietly(db_had_out)(con, p) -> statuts ; gc(); #ok
purrr::quietly(db_had_out)(con, p, annee = 2017, mois = 7) -> statuts ; gc(); #..

## End(Not run)
```

db_liste_tables ~ db - Lister les tables d'une db en tableau

Description

~ db - Lister les tables d'une db en tableau

~ db - Lister les tables d'une db en tableau

Usage

```
db_liste_tables(con, nb = 15)
```

```
db_liste_tables(con, nb = 15)
```

Arguments

con	la connexion a la base de donnees (src_..)
nb	le nombre de lignes du tableau

Value

nothing

nothing

Examples

```
## Not run:
db_liste_tables(con)

## End(Not run)
## Not run:
db_liste_tables(con)

## End(Not run)
```

db_mco_in ~ db - Copier les rum dans une db

Description

Copier les rum, les actes et les diagnostics des rums dans une db

Usage

```
db_mco_in(con, p, remove = T, zip = T, indexes = list(), ...)
```

Arguments

con	la connexion a la base de donnees (src_..)
p	le noyau pmeasyr
remove	a TRUE, les tables precedentes rum de l'annee sont effacees avant
zip	a TRUE les fichiers des archives sont dezipes et effaces apres integration dans la db
indexes	index a ajouter a la table dans la base (voir copy_to)

Details

Les tables sont importées dans R puis copiées dans la db La table diag est créée et la durée des rum est calculée (DUREESEJPART)

Value

nothing

Examples

```
## Not run:
purrr::quietly(db_mco_in)(con, p) -> statuts ; gc(); #ok
purrr::quietly(db_mco_in)(con, p, annee = 2015) -> statuts ; gc(); #..

## End(Not run)
```

db_mco_out	~ db - Copier les rsa dans une db
------------	-----------------------------------

Description

Copier les rsa, les passages um, les actes et les diagnostics des rsa, et ano dans une db

Usage

```
db_mco_out(con, p, remove = T, zip = T, indexes = list(), ...)
```

Arguments

con	la connexion a la base de donnees (src_...)
p	le noyau pmeasyr
remove	a TRUE, les tables precedentes rsa de l'annee sont effacees avant
zip	a TRUE les fichiers des archives sont dezippees et effaces apres integration dans la db
indexes	index a ajouter a la table dans la base (voir copy_to)

Details

Les tables sont importées dans R puis copiées dans la db. La table diag est créée, les variables ghm, année séquentielle des tarifs et un champ caractère diagnostics sont ajoutés à la table rsa. Le tra est ajouté aux tables.

Value

nothing

Examples

```
## Not run:  
purrr::quietly(db_mco_out)(con, p) -> statuts ; gc(); #ok  
purrr::quietly(db_mco_out)(con, p, annee = 2017, mois = 7) -> statuts ; gc(); #..  
  
## End(Not run)
```

db_psy_out *~ db - Copier les rpsa dans une db*

Description

Copier les rpsa, les actes, les diagnostics et la table ano des rpsa dans une db

Usage

```
db_psy_out(con, p, remove = T, zip = T, indexes = list(), ...)
```

Arguments

con	la connexion a la base de donnees (src_..)
p	le noyau pmeasyr
remove	a TRUE, les tables precedentes rpsa de l'annee sont effacees avant
zip	a TRUE les fichiers des archives sont dezippees et effaces apres integration dans la db
indexes	index a ajouter a la table dans la base (voir copy_to)

Details

Les tables sont importées dans R puis copiées dans la db. Le tra est ajouté aux tables.

Value

nothing

Examples

```
## Not run:  
purrr::quietly(db_psy_out)(con, p) -> statuts ; gc(); #ok  
purrr::quietly(db_psy_out)(con, p, annee = 2017, mois = 6) -> statuts ; gc(); #..  
  
## End(Not run)
```

db_rsf_out ~ db - Copier les rsf dans une db

Description

Copier tous les rsf (lettre par lettre, A, B, C, ...), les ano-ace dans une db

Copier tous les rsf (lettre par lettre, A, B, C, ...), les ano-ace dans une db

Usage

```
db_rsf_out(con, p, remove = T, zip = T, indexes = list(), ...)
```

```
db_rsf_out(con, p, remove = T, zip = T, indexes = list(), ...)
```

Arguments

con	la connexion a la base de donnees (src_...)
p	le noyau pmeasyr
remove	a TRUE, les tables precedentes rafael de l'annee sont effacees avant
zip	a TRUE les fichiers des archives sont dezippees et effaces apres integration dans la db
indexes	index a ajouter a la table dans la base (voir copy_to)

Details

Les tables sont importées dans R puis copiées dans la db

Les tables sont importées dans R puis copiées dans la db

Value

nothing

nothing

Examples

```
## Not run:
purrr::quietly(db_rsf_out)(con, p) -> statuts ; gc(); #ok
purrr::quietly(db_rsf_out)(con, p, annee = 2014) -> statuts ; gc(); #ok

## End(Not run)
## Not run:
purrr::quietly(db_rsf_out)(con, p) -> statuts ; gc(); #ok
purrr::quietly(db_rsf_out)(con, p, annee = 2014) -> statuts ; gc(); #ok

## End(Not run)
```

db_ssr_out *~ db - Copier les rha dans une db*

Description

Copier les rha, les actes, les diagnostics des rha, ssrha et ano dans une db

Usage

```
db_ssr_out(con, p, remove = T, zip = T, indexes = list(), ...)
```

Arguments

con	la connexion a la base de donnees (src_..)
p	le noyau pmeasyr
remove	a TRUE, les tables precedentes rha de l'annee sont effacees avant
zip	a TRUE les fichiers des archives sont dezipes et effaces apres integration dans la db
indexes	index a ajouter a la table dans la base (voir copy_to)

Details

Les tables sont importées dans R puis copiées dans la db. Le tra est ajouté aux tables.

Value

nothing

Examples

```
## Not run:  
purrr::quietly(db_ssr_out)(con, p) -> statuts ; gc(); #ok  
purrr::quietly(db_ssr_out)(con, p, annee = 2017, mois = 7) -> statuts ; gc(); #..  
  
## End(Not run)
```

dico *~ Dico - Dictionnaire des tables*

Description

Obtenir le dictionnaire d'une table

Usage

dico(table)

Arguments

table Table dont on veut le dictionnaire de variables

Author(s)

G. Pressiat

See Also

[irsa](#), [irum](#)

Examples

```
## Not run:
# N'importer qu'une ligne du fichier :
  irsa('750712184', 2016, 8, '~/path/path', typi= 1, n_max = 1) -> import
  dico(import$rsa)

## End(Not run)
```

enrobeur *~ req : mise en forme d'une liste de codes*

Description

~ req : mise en forme d'une liste de codes

~ req : mise en forme d'une liste de codes

Usage

enrobeur(a, robe = "", colonne = F, interstice = ", ", symetrique = F)

enrobeur(a, robe = "", colonne = F, interstice = ", ", symetrique = F)

Author(s)

G. Pressiat voir <https://guillaumepressiat.shinyapps.io/transcodeur/> pour son utilisation interactive hors AP-HP

G. Pressiat voir <https://guillaumepressiat.shinyapps.io/transcodeur/> pour son utilisation interactive hors AP-HP

Examples

```
## Not run:
li <- c('QEFA003', 'QEFA005', 'QEFA010', 'QEFA013', 'QEFA015', 'QEFA019', 'QEFA020')

enrobeur(li, robe="", interstice="|") %>% cat()

enrobeur(li, robe="\'", interstice=",") %>% cat()

enrobeur(li, robe="\'%", interstice="\n", symetrique = T) %>% cat()

## End(Not run)

## Not run:
li <- c('QEFA003', 'QEFA005', 'QEFA010', 'QEFA013', 'QEFA015', 'QEFA019', 'QEFA020')

enrobeur(li, robe="", interstice="|") %>% cat()

enrobeur(li, robe="\'", interstice=",") %>% cat()

enrobeur(li, robe="\'%", interstice="\n", symetrique = T) %>% cat()

## End(Not run)
```

epmsi_mco_rae

~ VVR - Reproduire le tableau RAE

Description

~ VVR - Reproduire le tableau RAE

Usage

```
epmsi_mco_rae(valo, knit = FALSE)
```

Arguments

valo	Un tibble résultant de vvr_mco
knit	à TRUE, appliquer une sortie 'knitr::kable' au résultat

Value

Un tibble similaire au tableau RAE epmsi

Author(s)

G. Pressiat

Examples

```
## Not run:  
epmsi_mco_rae(valo)  
  
## End(Not run)
```

epmsi_mco_rav ~ VVR - Reproduire le tableau RAV

Description

~ VVR - Reproduire le tableau RAV

Usage

```
epmsi_mco_rav(valo, knit = FALSE)
```

Arguments

valo	Un tibble résultant de vvr_mco
knit	à TRUE, appliquer une sortie 'knitr::kable' au résultat

Value

Un tibble similaire au tableau RAV epmsi (montant BR* ou BR si coeff prud = 1)

Author(s)

G. Pressiat, fbrcdnj

Examples

```
## Not run:  
epmsi_mco_rav(valo)  
  
## End(Not run)
```

epmsi_mco_rav2 ~ VVR - Reproduire le tableau RAV (works)

Description

~ VVR - Reproduire le tableau RAV (works)

Usage

```
epmsi_mco_rav2(valo, theorique = TRUE)
```

Arguments

valo	Un tibble résultant de vvr_mco
knit	à TRUE, appliquer une sortie 'knitr::kable' au résultat

Value

Un tibble similaire au tableau RAV epmsi (montant BR* ou BR si coeff prud = 1)

Author(s)

G. Pressiat

Examples

```
## Not run:  
epmsi_mco_rav2(valo)  
  
## End(Not run)
```

epmsi_mco_rav_rum ~ VVR - Synthese des rubriques de valo apres valo des RUM

Description

~ VVR - Synthese des rubriques de valo apres valo des RUM

Usage

```
epmsi_mco_rav_rum(valo_rum, theorique = TRUE)
```

Arguments

valo_rum	Un tibble résultant de vvr_rum
----------	--

Value

Un tibble similaire au tableau RAV epmsi

Author(s)

G. Pressiat

See Also

epmsi_mco_rav, epmsi_mco_rav_rum, epmsi_mco_rav2,

Examples

```
## Not run:  
epmsi_mco_rav_rum(valo)  
  
## End(Not run)
```

epmsi_mco_sv

~ VVR - Reproduire le tableau SV

Description

Il s'agit d'un tableau similaire au tableau "Séjours Valorisés"

Usage

```
epmsi_mco_sv(valo, knit = FALSE)
```

Arguments

valo	Un tibble résultant de vvr_mco
knit	à TRUE, appliquer une sortie 'knitr::kable' au résultat

Value

Un tibble similaire au tableau RAV epmsi

Il s'agit d'un tableau similaire au tableau "Récapitulation Activité - Valorisation"

Author(s)

G. Pressiat

Examples

```
## Not run:
epmsi_mco_sv(valo)

## End(Not run)
```

epmsi_mco_vmed_f ~ MCO - Comprendre le tableau epmsi 1.V.2.VMED - F

Description

~ MCO - Comprendre le tableau epmsi 1.V.2.VMED - F

Usage

```
epmsi_mco_vmed_f(p, ref_indic)
```

Arguments

p	Noyau de paramètres
ref_indic	table des indications ATIH

Value

Liste de deux tables : tableau ePMSI et détail avec NAS, UCD, Indcations, etc.

Examples

```
## Not run:
library(pmeasyr)
library(dplyr, warn.conflicts = FALSE)

p <- noyau_pmeasyr(
  finess = '123456789',
  annee = 2019,
  mois = 3,
  path = '~/Documents/data/mco',
  tolower_names = TRUE,
  lib = FALSE,
  progress = FALSE
)

# devtools::install_github('GuillaumePressiat/nomensland')
library(nomensland)
ref_indic <- get_table('mco_medref_atih_indications') %>%
  mutate(mois = substr(anseqta,1,2),
         annee = substr(anseqta, 3,6))
```

```
vmed_f <- epmsi_mco_vmed_f(p,  
                           ref_indic)  
  
## End(Not run)
```

extraire_types_fichiers

*Extraire les types de fichiers Permet d'afficher les types de fichiers
d'une archive (ano, rss, rsa, ...)*

Description

Extraire les types de fichiers Permet d'afficher les types de fichiers d'une archive (ano, rss, rsa, ...)

Usage

```
extraire_types_fichiers(selection_fichiers_a_extraire, info_archive)
```

Arguments

selection_fichiers_a_extraire Fichiers sélectionnés dans le zip (résultat de selectionne_fichiers)
info_archive informatin de l'archive (résultat de selectionne_archive)

Value

un vecteur

formats

Table des formats

Description

Table des formats

Author(s)

G. Pressiat

iano_had ~ HAD - Import des Anohosp

Description

Imports du fichier Ano

Usage

```
iano_had(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
typano	Type de donnees in / out
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu (sortie de Paprica) : *finess.annee.moisc.ano*

750712184.2016.2.ano

Value

Une table (data.frame, tibble) contenant les données Anohosp HAD du Out.

Author(s)

G. Pressiat

See Also

[irapss](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  anoh <- iano_had('750712184',2015,12,"~/Documents/data/had")

## End(Not run)
```

iano_mco ~ MCO - Import des Anohosp

Description

Import du fichier ANO In ou Out.

Usage

```
iano_mco(finess, annee, mois, path, lib = T, tolower_names = F, typano = "out")
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des données sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typano	Type de donnees In / Out
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu : *finess.annee.moisc.ano*
finess.annee.moisc.ano.txt

750712184.2016.2.ano 750712184.2016.2.ano.txt

Value

Une table (data.frame ou tibble) qui contient les données Anohosp in / out

Author(s)

G. Pressiat

See Also

[irum](#), [irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
iano_mco('750712184',2015,12,'~/Documents/data/mco') -> ano_out15
iano_mco('750712184',2015,12,'~/Documents/data/mco', typano = "in") -> ano_in15

## End(Not run)
```

iano_psy

~ PSY - Import des Anohosp

Description

Import du fichier Ano

Usage

```
iano_psy(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typano	Type de donnees in / out
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2012 pris en charge Structure du nom du fichier attendu (sortie de Genrha) : *finess.annee.moisc.ano*

750712184.2016.2.ano

Value

Une table (data.frame, tibble) contenant les données Anohosp SSR du Out.

Author(s)

G. Pressiat

See Also

[irpsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  anoh <- iano_psy('750712184',2015,12,"~/Documents/data/psy")

## End(Not run)
```

iano_rafael ~ RSF - Import des Anohosp RSFA

Description

Import du fichier ANO-ACE RSF Out ou le ano-ace-maj (reprise)

Usage

```
iano_rafael(finess, annee, mois, path, lib = T, lamda = F, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lamda	a TRUE, importe le fichier ano-ace-maj
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2012 pris en charge pour les ano-ace Formats depuis 2014 pris en charge pour les ano-ace-maj (reprise 2013)

Structure du nom du fichier attendu : *finess.annee.moisc.ano*

750712184.2016.2.ano

Value

Une table (data.frame ou tibble) qui contient les données Anohosp in / out

Author(s)

G. Pressiat

See Also[irafael](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)**Examples**

```
## Not run:
  iano_rafael('750712184', 2015, 12, '~/Documents/data/rsf') -> ano_out15
  iano_rafael('750712184', 2015, 12, '~/Documents/data/rsf', lamda = T) -> lamda_maj_ano_out14

## End(Not run)
```

iano_ssr

~ SSR - Import des Anohosp

Description

Import du fichier Ano

Usage

```
iano_ssr(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typano	Type de donnees in / out
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	paramètres supplémentaires à passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premières lignes, progress = F, skip = ...

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu (sortie de Genrha) : *finess.annee.moisc.ano*

750712184.2016.2.ano

Value

Une table (data.frame, tibble) contenant les données Anohosp SSR du Out.

Author(s)

G. Pressiat

See Also

[irha](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  anoh <- iano_ssr('750712184',2015,12,"~/Documents/data/ssr")

## End(Not run)
```

 idiap

 ~ MCO - Import des DIAP

Description

Imports des fichiers DIAP In / Out

Usage

```
idiap(finess, annee, mois, path,
      typdiap = c("out", "in"),
      lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typdiap	Type de donnees In / Out
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

Value

Une table (data.frame, tibble) contenant les dialyses péritonéales In ou Out.

Author(s)

G. Pressiat

See Also

[irum](#), [irsa](#)

Examples

```
## Not run:
  idiap <- idiap('750712184',2015,12,"~/Documents/data/mco")

## End(Not run)
```

idmi_mco

~ MCO - Import des DMI

Description

Import des fichiers DMI In ou Out.

Usage

```
idmi_mco(...)
```

Arguments

<code>finess</code>	Finess du Out a importer : dans le nom du fichier
<code>annee</code>	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
<code>mois</code>	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
<code>path</code>	Localisation du fichier de donnees
<code>typdmi</code>	Type de donnees In / Out
<code>lib</code>	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package <code>sjlabelled</code>
<code>tolower_names</code>	a TRUE les noms de colonnes sont tous en minuscules
<code>~...</code>	parametres supplementaires a passer dans la fonction <code>read_fwf</code> , par exemple <code>n_max = 1e3</code> pour lire les 1000 premieres lignes, <code>progress = F</code> , <code>skip = 1e3</code>

Details

Formats depuis 2011 pris en charge

Value

Une table (data.frame, tibble) contenant les dispositifs médicaux implantables In ou Out (T2A, ATU et thrombo selon l'existence des fichiers : si le fichier n'existe pas, pas de donnée importée). Pour discriminer le type de prestation, la colonne TYPEPREST donne l'information : T2A 06 - ATU 09 - THROMBO 10

Author(s)

G. Pressiat

See Also

[irum](#), [irsa](#)

Examples

```
## Not run:
idmi_mco('750712184',2015,12,'~/Documents/data/mco') -> dmi_out15
idmi_mco('750712184',2015,12,'~/Documents/data/mco', typdmi = "in") -> dmi_in15

## End(Not run)
```

iium

~ MCO - Import des donnees UM du Out

Description

Imports du fichier IUM MCO

Usage

```
iium(finness, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finness	Finness du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires à passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

Value

Une table (data.frame, tibble) contenant les informations structures du Out.

Author(s)

G. Pressiat

See Also

[irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
um <- iium('750712184',2015,12,"~/Documents/data/mco")

## End(Not run)
```

iium_corresp *~ MCO - Import des donnees correspondances d'UM du Out*

Description

Imports du fichier IUM corresp MCO

Usage

```
iium_corresp(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2020)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires à passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2020 pris en charge

Value

Une table (data.frame, tibble) contenant les informations Um out/UM in.

Author(s)

G. Pressiat

See Also

[irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  um_corresp <- iium_corresp('750712184',2020,12,"~/Documents/data/mco")

## End(Not run)
```

iium_ssr ~ SSR - Import des donnees UM du Out

Description

Imports du fichier IUM SSR

Usage

```
iium_ssr(finess, annee, mois, path, lib = T, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires à passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2013 pris en charge

Value

Une table (data.frame, tibble) contenant les informations structures du Out.

Author(s)

G. Pressiat

See Also

[irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
um <- iium_ssr('750712184',2015,12,"~/Documents/data/ssr")

## End(Not run)
```

iivg_mco

~ MCO - Import des IVG

Description

Import des fichiers IVG In ou Out.

Usage

```
iivg_mco(finess, annee, mois, path, lib = T, tolower_names = F, typivg = c('out', 'in'))
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typivg	Type de donnees In / Out
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2017 pris en charge

Value

Une table (data.frame, tibble) contenant les IVG In ou Out (si le fichier n'existe pas, pas de donnée importée). Le TYPEPREST est par défaut 53

Author(s)

G. Pressiat, N. Taright

See Also

[irum](#), [irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  iivg_mco('750712184', 2017, 12, '~/Documents/data/mco') -> ivg_out17
  iivg_mco('750712184', 2017, 12, '~/Documents/data/mco', typivg = "in") -> ivg_in17

## End(Not run)
```

 ileg_had

 ~ HAD - Import des erreurs Leg

Description

Import de la liste d'erreurs de génération Paprica

Usage

```
ileg_had(finiss, annee, mois, path, reshape = F, tolower_names = F, ...)
```

Arguments

finiss	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
reshape	booleen TRUE/FALSE : la donnee doit-elle etre restructuree ? une ligne = une erreur, sinon, une ligne = un sejour. par default a F
tolower_names	a TRUE les noms de colonnes sont tous en minuscules

Value

Une table (data.frame, tibble) contenant les erreurs Out.

Author(s)

G. Pressiat

See Also[irapss](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)**Examples**

```
## Not run:
  ileg_had('750712184',2015,12,'~/Documents/data/had') -> leg15

## End(Not run)
```

 ileg_mco

 ~ MCO - Import des erreurs Leg

Description

Import de la liste d'erreurs de génération Genrsa

Usage

```
ileg_mco(finess, annee, mois, path, reshape = F, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
reshape	booleen TRUE/FALSE : la donnee doit-elle etre restructuree ? une ligne = une erreur, sinon, une ligne = un sejour. par default a F
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

Value

Une table (data.frame, tibble) contenant les erreurs Out.

Author(s)

G. Pressiat

See Also

[irum](#), [irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:  
  ileg_mco('750712184',2015,12,'~/Documents/data/mco') -> leg15  
  
## End(Not run)
```

ileg_ssr

~ SSR - Import des erreurs Leg

Description

Import de la liste d'erreurs de génération Genrha

Usage

```
ileg_ssr(finess, annee, mois, path, reshape = F, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
reshape	booleen TRUE/FALSE : la donnee doit-elle etre restructuree ? une ligne = une erreur, sinon, une ligne = un sejour. par default a F
tolower_names	a TRUE les noms de colonnes sont tous en minuscules

Value

Une table (data.frame, tibble) contenant les erreurs Out.

Author(s)

G. Pressiat

See Also

[irha](#), [issrha](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  ileg_had('750712184',2015,12,'~/Documents/data/ssr') -> leg15

## End(Not run)
```

imed_had ~ HAD - Import des Med

Description

Imports du fichier Med Out

Usage

```
imed_had(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

import des med, medatu et mchl si le fichier existe

Structure du nom du fichier attendu (sortie de Paprica) : *finess.annee.moisc.med*

750712184.2016.2.med

Value

Une table (data.frame, tibble) contenant les données médicaments HAD du Out.

Author(s)

G. Pressiat

See Also

[irapss](#) utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
medh <- imed_had('750712184',2015,12,"~/Documents/data/had")

## End(Not run)
```

imed_mco ~ MCO - Import des Med

Description

Import des fichiers MED In ou Out.

Usage

```
imed_mco(finess, annee, mois, path, lib = T, tolower_names = F, typmed = c('out', 'in'))
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typmed	Type de donnees In / Out
lib	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

Value

Une table (data.frame, tibble) contenant les médicaments In ou Out (T2A, ATU et thrombo selon l'existence des fichiers : si le fichier n'existe pas, pas de donnée importée). Pour discriminer le type de prestation, la colonne TYPEPREST donne l'information : T2A 06 - ATU 09 - THROMBO 10

Author(s)

G. Pressiat

See Also[irum](#), [irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)**Examples**

```
## Not run:
imed_mco('750712184',2015,12,'~/Documents/data/mco') -> med_out15
imed_mco('750712184',2015,12,'~/Documents/data/mco', typmed = "in") -> med_in15

## End(Not run)
```

imed_ssr

~ SSR - Import des Med

Description

Imports du fichier Med Out

Usage

```
imed_ssr(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu (sortie de Genrha) : *finess.annee.moisc.med*

750712184.2017.2.med

Value

Une table (data.frame, tibble) contenant les données médicaments SSR du Out.

Author(s)

G. Pressiat

See Also

[irapss](#) utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  meds <- imed_ssr('750712184',2015,12,"~/Documents/data/ssr")

## End(Not run)
```

inner_tra ~ TRA - Ajout du TRA aux donnees Out

Description

Ajout du TRA par dplyr::inner_join

Usage

```
inner_tra(table, tra, sel = 1, champ = "mco")
```

Arguments

table	Table a laquelle rajouter le tra
tra	tra a rajouter
sel	Variable a garder du tra ; sel = 1 : numero de sejour, sel = 2 : toutes les variables
champ	Champ PMSI : mco, had, ssr, psy ; deux tra en psy : psy_rpsa, psy_r3a

Value

Une table contenant le inner_join entre table et tra

Author(s)

G. Pressiat

See Also

[irum](#), [irsa](#), [imed_mco](#), [irpsa](#), [irha](#), [irapss](#)

Examples

```
## Not run:
med <- imed_mco('750712184',2015,12,"~/Documents/data/mco","out")
tra <- itra('750712184',2015,12,"~/Documents/data/mco")
med <- inner_tra(med,tra)

## End(Not run)
```

ipie

~ MCO - Import des PIE

Description

Imports des fichiers PIE Out

Usage

```
ipie(finess, annee, mois, path, typpie = c("out", "in"), lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typpie	Type de donnees In / Out (seulement out pour le moment)
lib	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

Value

Une table (data.frame, tibble) contenant les prestation inter-établissement Out.

Author(s)

G. Pressiat

See Also[irsa](#), [irum](#)**Examples**

```
## Not run:
  pie <- ipie('750712184',2018,4,"~/Documents/data/mco")

## End(Not run)
```

ipo ~ MCO - Import des PO

Description

Imports des fichiers PO In / Out

Usage

```
ipo(finess, annee, mois, path, typpo = c("out", "in"), lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
typpo	Type de donnees In / Out
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

Value

Une table (data.frame, tibble) contenant les prélèvements d'organes In ou Out.

Author(s)

G. Pressiat

See Also

[irum](#), [irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  po <- ipo('750712184',2015,12,"~/Documents/data/mco")

## End(Not run)
```

ir3a ~ PSY - Import des R3A

Description

Import du fichier R3A

Usage

```
ir3a(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des données (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2012 pris en charge Structure du nom du fichier attendu (sortie de Pivoine) : *finess.annee.moisc.r3a*

750712184.2016.3.r3a

Value

Une table (data.frame, tibble) contenant les données R3A.

Author(s)

G. Pressiat

See Also[irpsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)**Examples**

```
## Not run:
r3a <- ir3a('750712184',2015,12,"~/Documents/data/psy")

## End(Not run)
```

```
iraa ~ PSY - Import des raa
```

Description

Import du fichier raa

Usage

```
iraa(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

<code>finess</code>	Finness du Out a importer : dans le nom du fichier
<code>annee</code>	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
<code>mois</code>	Mois PMSI (nb) des données (janvier : 1, decembre : 12)
<code>path</code>	Localisation du fichier de donnees
<code>lib</code>	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
<code>tolower_names</code>	a TRUE les noms de colonnes sont tous en minuscules
<code>~...</code>	parametres supplementaires a passer dans la fonction read_fwf , par exemple <code>n_max = 1e3</code> pour lire les 1000 premieres lignes, <code>progress = F</code> , <code>skip = 1e3</code>

Details

Formats depuis 2012 pris en charge Structure du nom du fichier attendu (sortie de Pivoine) : *finess.annee.moisc.rpa.txt*

750712184.2016.3.rpa.txt

Value

Une table (data.frame, tibble) contenant les données raa.

Author(s)

G. Pressiat

See Also

[irpsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  raa <- iraa('750712184', 2015, 12, "~/Documents/data/psy")

## End(Not run)
```

 irafael

 ~ *RSF - Import des Rafael*

Description

Import des Rafael et des Rafael reprises

Usage

```
irafael(finess, annee, mois, path, lib = T, tolower_names = F,
  stat = T, lister = c("A", "B",
  "C", "H", "L", "M", "P"), lamda = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par défaut a TRUE ; necessite le package sjlabelled
stat	avec stat = T, un tableau synthetise le nombre de lignes par type de rafael
lister	Liste des types d'enregistrements a importer
lamda	a TRUE, importe les fichiers rsfa-maj de reprise de l'annee passee
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	Autres parametres a specifier n_max = 1e3, ...

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu (sortie de Paprica) : *finess.annee.moisc.rapss*

750712184.2016.2.rapss

Value

Une classe S3 contenant les tables (data.frame, tibble ou tbl) importées (rapss, acdi, ght).

Author(s)

G. Pressiat

See Also

[iano_had](#), [ileg_had](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
um <- iium('750712184',2015,12,"~/Documents/data/had")

## End(Not run)
```

irha

~ SSR - Import des RHA

Description

Import des RHA

Usage

```
irha(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du fichier Out de GENRHA a integrer
annee	Annee de la periode (du fichier Out)
mois	Mois de la periode (du fichier Out)
path	Chemin d'accès au fichier .rha
lib	Attribution de libelles aux colonnes
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max=10e3 pour lire les 10000 premieres lignes

See Also

[iano_ssr](#), [ileg_ssr](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  irhs('750712184',2015,12,'pathpath/') -> rhs15

## End(Not run)
```

 irps

 ~ PSY - Import des RPS

Description

Import du fichier rps

Usage

```
irps(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2012 pris en charge Structure du nom du fichier attendu (sortie de Pivoine) : *finess.annee.moisc.rps.txt*

750712184.2016.2.rps.txt

Value

Une table (data.frame, tibble) contenant les données rps.

Author(s)

G. Pressiat

See Also

[ir3a](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  rps <- irps('750712184',2015,12,"~/Documents/data/psy")

## End(Not run)
```

```
irpsa          ~ PSY - Import des RPSA
```

Description

Import du fichier RPSA

Usage

```
irpsa(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2012 pris en charge Structure du nom du fichier attendu (sortie de Pivoine) : *finess.annee.moisc.rpsa*

750712184.2016.2.rpsa

Value

Une table (data.frame, tibble) contenant les données RPSA.

Author(s)

G. Pressiat

See Also[ir3a](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)**Examples**

```
## Not run:
  rpsa <- irpsa('750712184',2015,12,"~/Documents/data/psy")

## End(Not run)
```

 irsa

 ~ MCO - Import des RSA

Description

Import des RSA. 6 types d'imports possibles.

Usage

```
irsa(finess, annee, mois, path, lib = T, typi = 4, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
typi	Type d'import, par defaut a 4, a 0 : propose a l'utilisateur de choisir au lancement
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu (sortie de Genrsa) : *fitness.annee.moisc.rsa*

750712184.2016.2.rsa

Types d'imports :

- 1 Light : partie fixe (très rapide)
- 2 Light+ : Partie fixe + stream en ligne (+) actes et das
- 3 Light++ : Partie fixe + stream en ligne (++) actes, das, typaut um et dpdr des um
- 4 Standard : Partie fixe + création des tables acdi et rsa_um
- 5 Standard+ : Partie fixe + création des tables acdi et rsa_um + stream (+)
- 6 Standard++ : Partie fixe + création des tables acdi et rsa_um + stream (++)

Principe du streaming : Mise en chaîne de caractères de la succession d'actes CCAM au cours du RUM, par exemple, pour un RUM : "ACQK001, LFQK002, MCQK001, NAQK015, PAQK002, PAQK900, YYYY600, ZZQP004"

La recherche d'un (ou d'une liste d') acte(s) sur un RUM est largement accélérée, comparée à une requête sur la large table acdi par une requête du type :

```
grep1("ZZQP004",rsa$actes) # toutes les lignes de RSA avec au moins un ZZQP004
e66 <- grep1('E66',das)|grep1('E66',dpdrum) # toutes les lignes de RSA avec un diagnostic E66
```

Cela permet de n'utiliser que la seule table rsa avec stream et d'avoir les infos sur les séjours directement : nb séjours, journées, entrée / sortie (...) plutôt que d'avoir à utiliser et croiser les tables acdi, rsa_um avec rsa.

Value

Une classe S3 contenant les tables (data.frame, tibble ou tbl) importées (rsa, rsa_um, actes et das si import > 3)

Author(s)

G. Pressiat

See Also

[irum](#), [ileg_mco](#), [iano_mco](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  irsa('750712184',2015,12,'~/Documents/data/mco') -> rsa15

## End(Not run)
```

 irsf

 ~ RSF - Import des RSF

Description

Import des RSF en année courante

Usage

```
irafael(finess, annee, mois, path, lib = T, tolower_names = F,
stat = T, lister = c("A", "B",
"C", "H", "L", "M", "P"), lamda = F, ...)
```

Arguments

finess	Finess du In a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par défaut a TRUE ; necessite le package sjlabelled
stat	avec stat = T, un tableau synthetise le nombre de lignes par type de rafaël
lister	Liste des types d'enregistrements a importer
lamda	a TRUE, importe les fichiers rsfa-maj de reprise de l'annee passee
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	Autres parametres a specifier n_max = 1e3, ...

Details

Formats depuis 2019 pour les rsf

Value

Une classe S3 contenant les tables (data.frame, tbl_df ou tbl) importées (rafaels)

Author(s)

G. Pressiat

See Also

[iano_rafael](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  irsf('750712184',2019,12,'~/Documents/data/rsf') -> rsf19

## End(Not run)
```

irum ~ MCO - Import des RUM

Description

Import des RUM. 4 types d'imports possibles.

Usage

```
irum(finess, annee, mois, path, lib = T, typi = 3, tolower_names = F, ...)
```

Arguments

finess	Finess du In a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package sjlabelled
typi	Type d'import, par default a 3, a 0 : propose a l'utilisateur de choisir au lancement
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~...	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge

Structure du nom du fichier attendu (entrée pour Genrsa) : *finess.annee.moisc.rum*

750712184.2016.2.rum

Types d'imports :

- 1 XLight : partie fixe
- 2 Light : partie fixe + streaming des actes, dad et das
- 3 Standard : partie fixe + table acdi
- 4 Standard+ : Import standard (3) + stream

Principe du streaming : Mise en chaîne de caractères de la succession d'actes CCAM au cours du RUM, par exemple, pour un RUM : "ACQK001, LFQK002, MCQK001, NAQK015, PAQK002, PAQK900, YYYY600, ZZQP004"

La recherche d'un (ou d'une liste d') acte(s) sur un RUM est largement accélérée, comparée à une requête sur la large table acdi par une requête du type :

```
grep1("ZZQP004", rum$actes) # toutes les lignes de RUM avec au moins un ZZQP004
grep1("ZZQP004|EBLA003", rum$actes) # toutes les lignes de RUM avec au moins un ZZQP004 ou un EBLA003
```

Value

Une classe S3 contenant les tables (data.frame, tibble) importées (rum, actes, das et dad si import 3 et 4)

Author(s)

G. Pressiat

See Also

[irsa](#), [ileg_mco](#), [iano_mco](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  irum('750712184', 2015, 12, '~/Documents/data/mco', typi = 1) -> rum15

## End(Not run)
```

issrha

~ SSR - Import des SSRHA

Description

Import du fichier SHA

Usage

```
issrha(finess, annee, mois, path, lib = T, tolower_names = F, ...)
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles a la table : T

tolower_names a TRUE les noms de colonnes sont tous en minuscules
~... parametres supplementaires a passer dans la fonction [read_fwf](#), par exemple
n_max = 1e3 pour lire les 1000 premieres lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu (sortie de Genrha) : *fitness.annee.moisc.sha*

750712184.2016.2.sha

Value

Une table (data.frame, tibble) contenant les données SHA, et a partir de 2017 une liste de deux tables (sha et gme)

Author(s)

G. Pressiat

See Also

[irha](#), [ileg_ssr](#), [iano_ssr](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:  
sha <- issrha('750712184',2015,12,"~/Documents/data/ssr")  
  
## End(Not run)
```

itpt_mco

~ MCO - Import des transports (TPT)

Description

Import des fichiers TPT In ou Out.

Usage

```
itpt_mco(finess, annee, mois, path, lib = T, tolower_names = F, tytpt = c('out', 'in'))
```

Arguments

<code>finess</code>	Finess du Out a importer : dans le nom du fichier
<code>annee</code>	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
<code>mois</code>	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
<code>path</code>	Localisation du fichier de donnees
<code>tytpt</code>	Type de donnees In / Out
<code>lib</code>	Ajout des libelles de colonnes aux tables, par default a TRUE ; necessite le package <code>sjlabelled</code>
<code>tolower_names</code>	a TRUE les noms de colonnes sont tous en minuscules
<code>~...</code>	parametres supplementaires a passer dans la fonction <code>read_fwf</code> , par exemple <code>n_max = 1e3</code> pour lire les 1000 premieres lignes, <code>progress = F</code> , <code>skip = 1e3</code>

Details

Formats depuis 2021 pris en charge

Value

Une table (`data.frame`, `tibble`) contenant les transports In ou Out (si le fichier n'existe pas, pas de donnée importée). Le `TYPEPREST` est par default 15

Author(s)

G. Pressiat, N. Taright

See Also

[irum](#), [irsa](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:
  ipt_mco('750712184', 2021, 12, '~/Documents/data/mco') -> tpt_out15
  ipt_mco('750712184', 2021, 12, '~/Documents/data/mco', tytpt = "in") -> tpt_in15

## End(Not run)
```

```
itra ~ TRA - Import du TRA
```

Description

Import du fichier TRA, 4 champs PMSI couverts.

Usage

```
itra(finess, annee, mois, path, lib = T, tolower_names = F, champ = "mco")
```

Arguments

finess	Finess du Out a importer : dans le nom du fichier
annee	Annee PMSI (nb) des donnees sur 4 caracteres (2016)
mois	Mois PMSI (nb) des donnees (janvier : 1, decembre : 12)
path	Localisation du fichier de donnees
lib	Ajout des libelles de colonnes aux tables, par defaut a TRUE ; necessite le package sjlabelled
champ	Champ PMSI du TRA a integrer ("mco", "ssr", "had", "psy_rpsa", "psy_r3a"), par defaut "mco"
tolower_names	a TRUE les noms de colonnes sont tous en minuscules
~ . . .	parametres supplementaires a passer dans la fonction read_fwf , par exemple n_max = 1e3 pour lire les 1000 premières lignes, progress = F, skip = 1e3

Details

Formats depuis 2011 pris en charge Structure du nom du fichier attendu (sortie de Genrsa) : *finess.annee.moisc.tra*

750712184.2016.2.tra

Value

Une table (data.frame ou tibble) qui contient : - Clé RSA - NORSS - Numéro de ligne du fichier RSS d'origine (rss.ini) - NAS - Date d'entrée du séjour - GHM groupage du RSS (origine) - Date de sortie du séjour

Author(s)

G. Pressiat

See Also

[irum](#), [irsa](#), [ileg_mco](#), [iano_mco](#), [irha](#), [irapss](#), [irpsa](#), [ir3a](#), utiliser un noyau de parametres avec [noyau_pmeasyr](#)

Examples

```
## Not run:  
  itra('750712184',2015,12,'~/Documents/data/champ_pmsi') -> tra15  
  
## End(Not run)
```

labeleasier

~ Labels pour le PMSI

Description

Attribuer des libelles aux colonnes PMSI

Usage

```
labeleasier(  
  col,  
  Mode_entree = F,  
  Mode_sortie = F,  
  Provenance = F,  
  Destination = F,  
  Sexe = F,  
  facteur = F  
)
```

Arguments

col	Colonne à laquelle attribuer le libellé
Mode_entree	'6' : 'Mutation'
Mode_sortie	'9' : 'Décès'
Provenance	'1' : 'MCO'
Destination	'6' : 'HAD'
Sexe	'2' : 'Femme'

Value

Un vecteur caractère ou facteur

Author(s)

G. Pressiat

Examples

```
## Not run:
  labeleasier(rsa$rsa$SEXE, Sexe = T, F)
  labeleasier(rsa$rsa$DEST, Destination = T, F)

## End(Not run)
```

lancer_requete *~ req : lancer une ou plusieurs requetes avec une ou des listes*

Description

~ req : lancer une ou plusieurs requetes avec une ou des listes
~ req : lancer une ou plusieurs requetes avec une ou des listes

Usage

```
lancer_requete(tables, elements, vars = NULL)

lancer_requete(tables, elements, vars = NULL)
```

Value

un tibble concatenant les resultats de toutes les requetes : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.
un tibble concatenant les resultats de toutes les requetes : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.

Author(s)

G. Pressiat
G. Pressiat

Examples

```
## Not run:
prepare_rsa(rsa) -> rsa
get_all_listes('Recours Exceptionnel') -> listes_re
lancer_requete(rsa, listes_re)

get_liste('chir_bariatrique_total') -> liste_bari
lancer_requete(rsa, liste_bari)

## End(Not run)

## Not run:
prepare_rsa(rsa) -> rsa
```

```

get_all_listes('Recours Exceptionnel') -> listes_re
lancer_requete(rsa, listes_re)

get_liste('chir_bariatrique_total') -> liste_bari
lancer_requete(rsa, liste_bari)

## End(Not run)

```

lancer_requete_db *~ req : lancer une ou plusieurs requetes dans une db avec une ou des listes*

Description

~ req : lancer une ou plusieurs requetes dans une db avec une ou des listes
 ~ req : lancer une ou plusieurs requetes dans une db avec une ou des listes

Usage

```

lancer_requete_db(con, an, elements, vars = NULL)

lancer_requete_db(con, an, elements, vars = NULL)

```

Value

un tibble concatenant les resultats de toutes les requetes : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.
 un tibble concatenant les resultats de toutes les requetes : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.

Author(s)

G. Pressiat
 G. Pressiat

Examples

```

## Not run:
get_all_listes('Recours Exceptionnel') -> listes_re
lancer_requete_db(con, an, listes_re)

get_liste('chir_bariatrique_total') -> liste_bari
lancer_requete_db(con, an, liste_bari)

## End(Not run)

## Not run:

```

```
get_all_listes('Recours Exceptionnel') -> listes_re
lancer_requete_db(con, an, listes_re)

get_liste('chir_bariatrique_total') -> liste_bari
lancer_requete_db(con, an, liste_bari)

## End(Not run)
```

noyau_pmeasyr *~ par - Noyau de parametres*

Description

Définir un noyau de paramètres

Usage

```
noyau_pmeasyr(...)

## S3 method for class 'pm_param'
print(x, ...)
```

Details

Voir exemple

Author(s)

G. Pressiat

See Also

[noyau_skeleton](#)

Examples

```
## Not run:
library(magrittr)

p <- noyau_pmeasyr(
  finess = '750712184',
  annee = 2016,
  mois = 12,
  path = '~/Documents/data/mco',
  progress = F
)

p %>% azip(type = "out", liste = "")
```

```
p %>% irsa()      -> rsa
p %>% iano_mco() -> ano
p %>% ipo()       -> po

p %>% adezip(type = "in", liste = "")

p %>% irum()      -> rum

# Modifier le type d'import :
irsa(p, typi = 6) -> rsa

# Pour visualiser p :
p
print(p)

## End(Not run)
```

noyau_skeleton ~ par - Noyau de parametres

Description

Générer un squelette de noyau de paramètres

Usage

```
noyau_skeleton(nom = "p", zip = F)
```

Details

Voir exemple

Author(s)

G. Pressiat

See Also

[noyau_pmeasyr](#)

Examples

```
## Not run:

noyau_skeleton()
## résultat :
## noyau_pmeasyr(
##   fitness = '.....',
##   annee = ....,
```

```
## mois = ...,
## path = ''
## ) -> p

noyau_skeleton("alpha_bravo", T)

## noyau_pmeasyr(
##   finess = '.....',
##   annee = .....,
##   mois = ..,
##   path = ''
## ) -> alpha_bravo
##
## # adezip(alpha_bravo, type = 'out')
## # adezip(alpha_bravo, type = 'in')

## End(Not run)
```

parse_noms_fichiers *Extraire les informations d'un nom de fichier*

Description

Les noms de fichiers des archives PMSI ont toujours le même format, de la forme *finess.annee_envois.mois_envois*. suivi par *horodatage* pour les archives *in* et *ou* et enfin le *type* de fichier. Cette fonction permet d'extraire cette information.

Usage

```
parse_noms_fichiers(noms_fichiers, return_tibble = TRUE)
```

Arguments

noms_fichiers Des noms de fichiers d'archive sous la forme d'un vecteur de caractères.
return_tibble Retourne une table si TRUE, sinon retourne une liste.

Value

Une liste ou une table avec les informations extraites du nom de fichier :

- **nom_fichier** Le nom de fichier passé en argument
- **finess**
- **annee**
- **mois**
- **horodatage_production** L'horodatage de production pour les fichiers *in* et *out* au format POSIXlt Si il y a plusieurs fichiers, la liste peut être facilement transformée en table avec `dplyr::bind_rows()`.
- **type** Type de fichier : *in*, *out*, *rss*...

Exemples

```
## Not run:
noms_de_fichiers <- c("671234567.2016.1.12032016140012.in.zip",
                    "671234567.2016.1.rum.txt")
x <- parse_noms_fichiers(noms_de_fichiers)
dplyr::bind_rows(x)

## End(Not run)
```

parse_nom_fichier *Decoupe un seul nom de fichier*

Description

Les noms de fichiers des archives PMSI ont toujours le même format, de la forme *finess.annee_envois.mois_envois*, suivi par *horodatage* pour les archives *in* et *ou* et enfin le *type* de fichier. Cette fonction permet d'extraire cette information.

Usage

```
parse_nom_fichier(nom_fichier, format_date_archive = "%d%m%Y%H%M%S")
```

Arguments

`nom_fichier` Le nom du fichier à découper. Chaîne de caractère de longueur 1.
`format_date_archive` Format de date d'horodatage pour les fichiers archive avec la notation de [base::strptime\(\)](#).

Value

Une liste ou une table avec les informations extraites du nom de fichier :

- `nom_fichier` Le nom de fichier passé en argument
- `finess`
- `annee`
- `mois`
- `horodatage_production` L'horodatage de production pour les fichiers *in* et *out* au format POSIXlt Si il y a plusieurs fichiers, la liste peut être facilement transformée en table avec [dplyr::bind_rows\(\)](#).
- `type` Type de fichier : *in*, *out*, *rss*...

See Also

[parse_noms_fichiers\(\)](#) pour traiter plusieurs noms de fichiers.

Examples

```
## Not run:
noms_de_fichiers <- c("671234567.2016.1.12032016140012.in.zip",
                    "671234567.2016.1.rum.txt")
x <- parse_noms_fichiers(noms_de_fichiers)
dplyr::bind_rows(x)

## End(Not run)
```

prepare_rsa

~ req : preparer les rsa pour la requete

Description

On selectionne certaines variables et on en cree d'autres utiles lors de l'execution de la requete

On selectionne certaines variables et on en cree d'autres utiles lors de l'execution de la requete

Usage

```
prepare_rsa(rsa)
```

```
prepare_rsa(rsa)
```

Value

un objet de classe rsa de pmeasyr

un objet de classe rsa de pmeasyr

Author(s)

G. Pressiat

G. Pressiat

Examples

```
## Not run:
tab <- irsa(typi = 6)
prepare_rsa(tab)
```

```
## End(Not run)
```

```
## Not run:
tab <- irsa(typi = 6)
prepare_rsa(tab)
```

```
## End(Not run)
```

renomme_fichier	<i>Renommer un fichier au format standardisé</i>
-----------------	--

Description

Cette fonction permet de renommer un fichier, par exemple directement produit à partir du système d'information (hospitalier), sous une forme standardisée afin de pouvoir l'importer avec pmeasyr tout en gardant optionnellement le fichier original.

Usage

```
renomme_fichier(  
  finess = "00000000",  
  annee,  
  mois,  
  path,  
  nom_fichier,  
  type_fichier,  
  garder_original = TRUE,  
  copie_temporaire = TRUE,  
  dossier_cible = path  
)
```

Arguments

nom_fichier	Nom du fichier original à renommer
type_fichier	Extension du fichier. Par exemple 'rss.txt'.
garder_original	Si 'TRUE' alors le fichier original n'est pas modifié. Sinon, il est renommé.
copie_temporaire	Si 'TRUE' alors le fichier renommé est copié dans un répertoire temporaire. Si 'FALSE', alors le nouveau fichier est Cet argument est pris en compte que si garder_original == TRUE est valable.
dossier_cible	Dossier où la copie renommée du fichier doit être enregistré. Valable que si garder_original == TRUE.

Value

Chemin vers le fichier renommé.

Examples

```
## Not run:  
library(pmeasyr)  
# classique  
renomme_fichier(annee = 2019,  
                 mois = 12,
```

```
      path = '~/Documents/data/mco/',
      nom_fichier = 'monrss.txt',
      type_fichier = 'rss.txt',
      garder_original = TRUE)

# avec noyau de paramètres
p <- noyau_pmeasyr(
  fitness = '123456789',
  annee = 2019,
  mois = 12,
  path = '~/Documents/data/mco',
  tolower_names = TRUE,
  lib = FALSE,
  progress = FALSE
)

renomme_fichier(p,
  nom_fichier = 'monrss.txt',
  type_fichier = 'rss.txt',
  garder_original = TRUE)

## End(Not run)
```

requete

~ req : requeter les rsa avec une liste

Description

~ req : requeter les rsa avec une liste

~ req : requeter les rsa avec une liste

Usage

```
requete(tables, elements, vars = NULL)
```

```
requete(tables, elements, vars = NULL)
```

Value

un tibble contenant les rsa respectant la requete : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.

un tibble contenant les rsa respectant la requete : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.

Author(s)

G. Pressiat

G. Pressiat

Examples

```
## Not run:
liste = list(actes = c('EBLA003', 'EQLF002'))
requete(rsa, liste)

liste = list(actes = c('EBLA003', 'EQLF002'), dureemax = 0)
requete(rsa, liste)

## End(Not run)

## Not run:
liste = list(actes = c('EBLA003', 'EQLF002'))
requete(rsa, liste)

liste = list(actes = c('EBLA003', 'EQLF002'), dureemax = 0)
requete(rsa, liste)

## End(Not run)
```

requete_db

~ req : requeter les rsa dans une db avec une liste

Description

~ req : requeter les rsa dans une db avec une liste

~ req : requeter les rsa dans une db avec une liste

Usage

```
requete_db(con, an, elements, vars = NULL)
```

```
requete_db(con, an, elements, vars = NULL)
```

Value

un tibble contenant les rsa respectant la requete : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.

un tibble contenant les rsa respectant la requete : les rsa qui ont un acte de la liste, un diag, une duree correspondante, etc.

Author(s)

G. Pressiat

G. Pressiat

Examples

```

## Not run:
liste = list(actes = c('EBLA003', 'EQLF002'))
requete_db(con, an, liste)

liste = list(actes = c('EBLA003', 'EQLF002'), dureemax = 0)
requete_db(con, 16, liste)

## End(Not run)

## Not run:
liste = list(actes = c('EBLA003', 'EQLF002'))
requete_db(con, an, liste)

liste = list(actes = c('EBLA003', 'EQLF002'), dureemax = 0)
requete_db(con, 16, liste)

## End(Not run)

```

selectionne_archive *Selectionne une archive in/out*

Description

Selectionne une archive in/out

Usage

```

selectionne_archive(
  finess,
  mois,
  annee,
  dossier_archives,
  type_archive = "in",
  recent = TRUE
)

```

Arguments

finess	Finess de l'archive.
mois	Mois de l'archive.
annee	Année de l'archive.
dossier_archives	Chemin vers le dossier contenant les archives. Vecteur de caractère de longueur 1.
type_archive	Type d'archive : <i>in</i> (par défaut) ou <i>out</i>
recent	par défaut TRUE, l'archive la plus recente sera utilisee, sinon propose à l'utilisateur de choisir quelle archive unzipper

Value

Une liste ou une table avec les informations extraites du nom de fichier :

- `nom_fichier` Le nom de fichier passé en argument
- `finess`
- `annee`
- `mois`
- `horodatage_production` L'horodatage de production pour les fichiers *in* et *out* au format POSIXlt Si il y a plusieurs fichiers, la liste peut être facilement transformée en table avec `dplyr::bind_rows()`.
- `type` Type de fichier : *in, out, rss...*

See Also

`adezip.default()` et `parse_nom_fichier()`

`selectionne_fichiers` *Selectionne les fichiers dans une archive*

Description

Selectionne les fichiers dans une archive

Usage

```
selectionne_fichiers(chemin_archive, types_fichier)
```

Value

Le noms dans fichiers dans l'archive

`tbl_had` *~ db - remote access aux tables had*

Description

~ db - remote access aux tables had

~ db - remote access aux tables had

Usage

```
tbl_had(con, an, table)
```

```
tbl_had(con, an, table)
```

Arguments

con	Connexion à la base de données
an	Année pmsi (ex: 16)
table	Table à requêter

Value

remote table
tibble
remote table
tibble

Examples

```
## Not run:  
tbl_had(con, 16, 'rapss_rapss')  
  
## End(Not run)  
## Not run:  
tbl_had(con, 16, 'rapss_rapss')  
  
## End(Not run)
```

tbl_mco ~ db - remote access aux tables mco

Description

~ db - remote access aux tables mco
~ db - remote access aux tables mco

Usage

```
tbl_mco(con, an, table)  
  
tbl_mco(con, an, table)
```

Arguments

con	Connexion à la base de données
an	Année pmsi (ex: 16)
table	Table à requêter

Value

remote table

remote table

Examples

```
## Not run:
tbl_mco(con, 16, 'rsa_rsa')
```

```
## End(Not run)
## Not run:
tbl_mco(con, 16, 'rsa_rsa')
```

```
## End(Not run)
```

tbl_psy	~ db - remote access aux tables psy
---------	-------------------------------------

Description

~ db - remote access aux tables psy

~ db - remote access aux tables psy

Usage

```
tbl_psy(con, an, table)
```

```
tbl_psy(con, an, table)
```

Arguments

con Connexion à la base de données

an Année pmsi (ex: 16)

table Table à requêter

Value

remote table

tibble

remote table

tibble

Examples

```
## Not run:
tbl_psy(con, 16, 'rpsa_rpsa')

## End(Not run)
## Not run:
tbl_psy(con, 16, 'rpsa_rpsa')

## End(Not run)
```

tbl_rsf ~ db - remote access aux tables rsf

Description

~ db - remote access aux tables rsf
~ db - remote access aux tables rsf

Usage

```
tbl_rsf(con, an, table)

tbl_rsf(con, an, table)
```

Arguments

con	Connexion à la base de données
an	Année pmsi (ex: 16)
table	Table à requêter

Value

remote table
remote table

Examples

```
## Not run:
tbl_rsf(con, 16, 'rsf_rafael_ano')

## End(Not run)
## Not run:
tbl_rsf(con, 16, 'rsf_rafael_ano')

## End(Not run)
```

tbl_ssr ~ db - remote access aux tables SSR

Description

~ db - remote access aux tables SSR

~ db - remote access aux tables SSR

Usage

```
tbl_ssr(con, an, table)
```

```
tbl_ssr(con, an, table)
```

Arguments

con	Connexion à la base de données
an	Année pmsi (ex: 16)
table	Table à requêter

Value

remote table

remote table

Examples

```
## Not run:  
tbl_ssr(con, 16, 'rha_rha')
```

```
## End(Not run)  
## Not run:  
tbl_ssr(con, 16, 'rha_rha')
```

```
## End(Not run)
```

tdiag ~ Tidy - Tidy Diagnostics

Description

Restructurer les diagnostics

Usage

```
tdiag(d, include = T)
```

Arguments

d Objet S3 resultat de l'import pmeasyr (irsa, irum, irha)
 include booleen : default a T; T : restructure l'objet S3 (agglomerer dp, dr, das et dad, par exemple)

Details

On obtient une table contenant tous les diagnostics par séjour, sur le principe suivant : Une variable numérique indique la position des diagnostics - pour les rsa : 1 : DP du rsa, 2 : DR du rsa, 3 : DPUM, 4 : DRUM, 5 : DAS - pour les rum : 1 : DP du rum, 2 : DR du rum, 3 : DAS, 4 : DAD - pour les rha : 1 : MMP du rha, 2 : FPPC du rha, 3 : AE, 4 : DA

Author(s)

G. Pressiat

See Also

[irsa](#), [irum](#), [irha](#)

Examples

```
## Not run:
# avec include = T
irum('750712184', 2016, 8, '~/path/path', typi = 3) -> d1
tdiag(d1) -> d1
d1$diags
d1$actes
d1$dad
irsa('750712184', 2016, 8, '~/path/path', typi = 4) -> d1
tdiag(d1, include = F) -> alldiag

## End(Not run)
```

vvr_ano_mco

~ VVR - preparer les ano pour valoriser les rsa

Description

Importer ou collecter les variables des anohosp out nécessaires pour attribuer le caractère facturable d'un séjour

Usage

```
vvr_ano_mco(...)
```

Arguments

p	Un noyau de paramètres noyau_pmeasyr
con	Une connexion vers une db contenant les données PMSI
annee	Dans le cas d'une con db, préciser l'année en integer sur deux caractères

Details

Deux méthodes sont disponibles : une utilisant l'import avec un noyau pmeasyr (p), l'autre utilisant les rsa stockés dans une base de données (con)

Value

Un tibble contenant les variables ano

Author(s)

G. Pressiat

See Also

[epmsi_mco_sv](#), [vvr_rsa](#), [vvr_mco](#)

Examples

```
## Not run:
# avec un noyau pmeasyr (importer les données)
p <- noyau_pmeasyr(
  finesse = '750712184',
  annee   = 2000 + 18,
  mois    = 4,
  path    = '~/Documents/data/mco',
  progress = FALSE,
  n_max   = Inf,
  lib     = FALSE,
  tolower_names = TRUE)

vano <- vvr_ano_mco(p)

# depuis une base de données (collecter les données)
dbdir <- "~/Documents/data/monetdb"
con <- src_monetdblite(dbdir)

vano <- vvr_ano_mco(con, 18)

## End(Not run)
```

vvr_ghs_supp

~ VVR - Attribuer les recettes GHS et suppléments sur des rsa

Description

Reproduire la valorisation BR et coefficients géo/prudentiels du tableau RAV d'epmsi, à partir des tables résultant des fonctions [vvr_rsa](#), [vvr_ano_mco](#), et de tables contenant les fichcomp PO et DIAP, PIE et MO

Usage

```
vvr_ghs_supp(
  rsa,
  tarifs,
  supplements = NULL,
  ano = NULL,
  porg = dplyr::tibble(),
  diap = dplyr::tibble(),
  pie = dplyr::tibble(),
  mo = dplyr::tibble(),
  full = FALSE,
  cgeo = 1.07,
  prudent = NULL,
  bee = TRUE
)
```

Arguments

rsa	Un tibble rsa partie fixe (créé avec vvr_rsa)
tarifs	Un tibble contenant une ligne par tarif GHS - année séquentielle des tarifs
supplements	Un tibble contenant une ligne par année et une colonne par tarif de supplément
ano	Un tibble, facultatif si 'r bee = TRUE', créé avec vvr_ano_mco
porg	Un tibble contenant les prélèvements d'organes du out (importés avec ipo)
diap	Un tibble contenant les dialyses péritonéales du out (importés avec idiap)
pie	Un tibble contenant les prestations inter-établissements du out (importés avec ipie)
mo	Un tibble contenant les molécules onéreuses du out (importés avec imed_mco)
full	Booléen, à 'r TRUE' toutes les variables intermédiaires de valo sont gardées
cgeo	Coefficient géographique, par défaut celui de l'Île-de-France (1.07)
prudent	coefficient prudentiel, par défaut à 'r NULL', le coefficient prudentiel est appliqué par année séquentielles des tarifs
bee	par défaut à 'r TRUE', seule la valorisation de GHS de base + extrême haut - extrême bas est calculée

Details

Cette fonction ne tient pas compte de la rubrique "Minoration forfaitaire liste en sus, car elle a été supprimée en 2018

Value

Un tibble contenant les différentes rubriques de valorisation, une ligne par clé rsa

Author(s)

G. Pressiat

See Also

[vvr_ano_mco](#), [vvr_rsa](#), [vvr_mco](#)

Examples

```
## Not run:
# Récupérer les tarifs GHS et des suppléments (ex-DGF) :
tarifs <- nomsland::get_table('tarifs_mco_ghs') %>% distinct(ghs, anseqta, .keep_all = TRUE)
supplements <- nomsland::get_table('tarifs_mco_supplements') %>% mutate_if(is.numeric, tidyr::replace_na, 0)

# Recette GHS de base et suppléments EXB, EXH
vvr_ghs_supp(rsa = vrsa, tarifs = tarifs)

# Recette GHS de base et suppléments EXB, EXH, et des suppléments
vvr_ghs_supp(vrsa, tarifs, supplements, vano, ipo(p), idiap(p), ipie(p), imed_mco(p), bee = FALSE)

## End(Not run)
```

vvr_had_gh

~ VVR - Attribuer les recettes GHT sur des rapss

Description

Reproduire la valorisation BR et coefficients géo/prudentiels du tableau VALR d'epmsi

Usage

```
vvr_had_gh(p, ghts, coeff_geo = 1.07, coeff_prudent = NULL)
```

Arguments

p	Un noyau de paramètres créé avec noyau_pmeasyr
gh	Un tibble contenant une ligne par tarif GHT - année séquentielle des tarifs - type de domicile
coeff_geo	Coefficient géographique, au choix (peut être mis à 1)
coeff_prudent	Coefficient prudentiel, par défaut la fonction créé ce coefficient automatiquement, sinon il peut-être mis à 1 ou autre

Details

Cette fonction ne tient pas compte du fichier de conventions ESMS (cas des finess non conventionnés non pec ici)

Value

Un tibble contenant les valorisations GHT des sous-séquences de la table rapss\$ght PAPRICA (et donc des séjours), les coefficient géo et prudentiels sont présents dans la table, la colonne tarif n'en tient pas compte, et il faut multiplier par le nb journées GHT pour obtenir la valorisation totale.

Author(s)

G. Pressiat

See Also

[irapss](#), [iano_had](#)

Examples

```
## Not run:
library(pmeasyr)
library(dplyr, warn.conflicts = F)

p <- noyau_pmeasyr(
  finess = '750712184',
  annee = 2018,
  mois = 12,
  path = '~/Documents/data/had',
  progress = FALSE,
  tolower_names = TRUE,
  lib = FALSE
)

adezip(p, type = "out", liste = c('rapss', 'ano'))

library(nomensland)
ghs <- get_table('tarifs_had_gh') %>%
  tidyr::gather(type_tarif, tarif, - paprica_numght, - lib_gh) %>%
  mutate(anseqta = stringr::str_extract(type_tarif, '[0-9]{4}'),
         typdom = substr(type_tarif, 6, nchar(type_tarif)))
```

```

# Utiliser cette fonction
base_gh<math>t <- vvr\_had\_gh<math>t(p, ghts)

ano <- iano_had(p)

library(stringfix)
# calculer le montant Base remboursement et le nb de journées valorisées
base_gh<math>t %>%
  inner_join(distinct(ano, noseqsej, .keep_all = TRUE), by = 'noseqsej') %>%
  # filtre sur séjours facturables
  filter(factam %in% c('1', '2')) %>%
  mutate(tarif = tarif * joursght * cgeo * cprudent) %>%
  summarise(
    euros = sum(tarif, na.rm = TRUE) %>% round(.,2) %>%
      formatC(., big.mark = " ", format = "f", digits = 2) %+% "€",
    nbj = sum(joursght))

##| euros      | nbj |
##|-----|-----|
##| xxx xxx,xx€| x xxx |

## End(Not run)

```

vvr_libelles_valo ~ VVR - Libelles des rubriques de valorisation et de types de rsa

Description

Fonction pour obtenir les tables de libellés du tableau SV, RAV, et le type de caractère bloquant des données VIDHOSP.

Usage

```
vvr_libelles_valo(wich)
```

Arguments

wich Chaîne de caractères parmi : 'lib_type_sej', 'lib_vidhosp', 'lib_valo'

Value

Un tibble avec les codes et libellés

Author(s)

G. Pressiat

Examples

```
## Not run:  
# Libellés des types de séjours (tableau SV)  
vvr_libelles_valo('lib_type_sej')  
  
# Libellés des types de vidhosp  
vvr_libelles_valo('lib_vidhosp')  
  
# Libellés des types de valo (tableau RAV)  
vvr_libelles_valo('lib_valo')  
  
## End(Not run)
```

vvr_mco

~ VVR - Croiser parties GHS / supplements et ano

Description

On ajoute les données facturation (vvr_mco_sv) aux données de valorisation 100

Usage

```
vvr_mco(rsa_v, ano_sv)
```

Arguments

rsa_v	tibble résultant de la fonction vvr_rsa
ano_sv	tibble résultant de la fonction vvr_ano_mco

Details

C'est un left join

Value

Un tibble final contenant la catégorie du tableau SV epmsi, et les variables rec_ de recette par séjour

Author(s)

G. Pressiat

`vvr_mco_sv`*~ VVR - Attribuer le caractere facturable par cle_rsa*

Description

Reproduire les catégories du tableau SV d'epmsi, à partir des tables résultant des fonctions `vvr_rsa`, `vvr_ano_mco` et éventuellement d'une table contenant le fichcomp PO

Usage

```
vvr_mco_sv(rsa, ano, porg = dplyr::tibble(cle_rsa = ""))
```

Arguments

<code>rsa</code>	un tibble rsa contenant les variables nécessaires (créé avec <code>vvr_rsa</code>)
<code>ano</code>	un tibble ano contenant les variables nécessaires (créé avec <code>vvr_mco_ano</code>)
<code>porg</code>	un tibble porg contenant les prélèvements d'organes du out (créé avec <code>ipo</code>)

Value

Un tibble contenant la catégorie du tableau SV epmsi, une ligne par clé rsa

Author(s)

G. Pressiat

See Also

`vvr_ano_mco`, `vvr_rsa`, `vvr_mco`

Examples

```
## Not run:  
# Tenir compte des porg  
vvr_mco_sv(vrsa, vano, porg = ipo(p))  
  
# ne pas tenir compte des porg  
vvr_mco_sv(vrsa, vano)  
  
## End(Not run)
```

vvr_rsa ~ VVR - preparer les rsa pour la valorisation

Description

Importer ou collecter les variables des rsa nécessaires à leur valorisation GHS + suppléments

Usage

```
vvr_rsa(...)
```

Arguments

p	Un noyau de paramètres noyau_pmeasyr
con	Une connexion vers une db contenant les données PMSI
annee	Dans le cas d'une con db, préciser l'année en integer sur deux caractères

Details

Deux méthodes sont disponibles : une utilisant l'import avec un noyau pmeasyr (p), l'autre utilisant les rsa stockés dans une base de données (con)

Value

Un tibble contenant les variables des rsa nécessaires pour calculer les recettes ghs et suppléments

Author(s)

G. Pressiat

See Also

[epmsi_mco_sv](#), [vvr_ano_mco](#), [vvr_mco](#)

Examples

```
## Not run:  
# avec un noyau pmeasyr (importer les données)  
annee <- 18  
p <- noyau_pmeasyr(  
  finess = '750712184',  
  annee = 2000 + annee,  
  mois = 4,  
  path = '~/Documents/data/mco',  
  progress = FALSE,  
  n_max = Inf,  
  lib = FALSE,  
  tolower_names = TRUE)
```

```
vrsa <- vvr_rsa(p)

# depuis une base de données (collecter les données)
dbdir <- "~/Documents/data/monetdb"
con <- src_monetdblite(dbdir)

vrsa <- vvr_rsa(con, annee)

## End(Not run)
```

vvr_rsa_hors_période ~ VVR - Forcer le groupage des RSA hors période en erreur

Description

Pour les RSA transmis hors période, on modifie CMD, GHM, et GHS en 90, 90Z99Z, et 9999 pour ne pas les valoriser

Usage

```
vvr_rsa_hors_période(vrsa, an_v, mois_v)
```

Author(s)

G. Pressiat

See Also

[epmsi_mco_sv](#), [vvr_rsa](#), [vvr_mco](#)

Examples

```
## Not run:
noyau_pmeasyr(
  fitness = '290000017',
  annee = 2019,
  mois = 11,
  path = '~/Documents/data/mco',
  progress = FALSE,
  lib = FALSE,
  tolower_names = TRUE
) -> p

vrsa <- vvr_rsa(p)
vrsa <- vrsa %>%
  vvr_rsa_hors_période(as.character(p$annee), stringr::str_pad(p$mois, 2, "left", '0'))

## End(Not run)
```

vvr_rum	~ VVR - Distribuer la valorisation des rsa au niveau des rum ou des passages UM
---------	---

Description

~ VVR - Distribuer la valorisation des rsa au niveau des rum ou des passages UM

Usage

```
vvr_rum(
  p,
  valo,
  repartition_multi = "{prop_pmct_um}*0.5+{prop_pass}*0.5",
  pmct_mono = c(FALSE, TRUE),
  seuil_pmct = 10,
  type_passage = c("RUM", "Passage unique")
)
```

Arguments

p	Un noyau de paramètres
valo	Un tibble résultant de vvr_mco
repartition_multi	pour renseigner les paramètres de la clef de répartition (entre durée de passage et PMCT des UM fréquentées)
pmct_mono	Calcul du PMCT par UM sur les mono-RUM (TRUE), ou sur tous les séjours par l'UM fournissant le DP (FALSE)
seuil_pmct	En dessous de quel nombre on considère le PMCT non robuste, dans ce cas, on passe à une distribution uniquement sur les durées de passages
type_passage	La table résultat est soit au niveau RUM, soit au niveau passage unique (pas de UM A, UM B, UM A, juste UM A, UM B)

Value

Un tibble

Author(s)

G. Pressiat
fbrcdnj

See Also

[epmsi_mco_rav_rum](#), [vvr_mco](#)

Examples

```
## Not run:  
vvr_rum(p, valo, type_passage = "RUM", pmct_mono = FALSE)  
  
## End(Not run)
```

vvr_rum_check_rubriques_rav
~ VVR - Confronter la valo rum a la valo rsa par rubrique epmsi

Description

~ VVR - Confronter la valo rum a la valo rsa par rubrique epmsi

Usage

```
vvr_rum_check_rubriques_rav(valo, valo_rum)
```

Arguments

valo	Un tibble résultant de vvr_mco
valo_rum	Un tibble résultant de vvr_rum

Value

Un tibble similaire au tableau RAV epmsi avec deux colonnes (rsa et rum)

Author(s)

G. Pressiat

See Also

[epmsi_mco_rav](#), [epmsi_mco_rav_rum](#), [epmsi_mco_rav2](#),

Examples

```
## Not run:  
vvr_rum_check_rubriques_rav(valo, valo_rum)  
  
## End(Not run)
```

Index

- * **data**
 - formats, 31

- adelete, 4, 6, 8, 9
- adezip, 4, 5, 8, 9
- adezip(), 8, 13
- adezip.default(), 80
- adezip2, 4, 6, 8, 9
- adezip2(), 13
- adezip3, 9
- ana_r_ca_gestes_marqueurs, 10
- ana_r_ghm_ambu_dms, 11
- astat, 4, 6, 8, 9, 12

- base::strptime(), 74

- collect_rsa_from_db, 14
- copy_to, 18–24
- creer_archive_vide, 15
- creer_archive_vide(), 15
- creer_archives_vides, 15
- creer_json, 16
- creer_nom_archive, 17

- db_generique, 17
- db_had_out, 18
- db_liste_tables, 19
- db_mco_in, 20
- db_mco_out, 21
- db_psy_out, 22
- db_rsf_out, 23
- db_ssr_out, 24
- dico, 25
- dplyr::bind_rows(), 73, 74, 80

- enrobeur, 25
- epmsi_mco_rae, 26
- epmsi_mco_rav, 27
- epmsi_mco_rav2, 28
- epmsi_mco_rav_rum, 28
- epmsi_mco_sv, 29, 86, 93, 94

- epmsi_mco_vmed_f, 30
- extraire_types_fichiers, 31

- formats, 31

- iano_had, 32, 56, 89
- iano_mco, 33, 61, 64, 67
- iano_psy, 34
- iano_rafael, 35, 55, 62
- iano_ssr, 36, 57, 58, 65
- idiap, 37, 87
- idmi_mco, 38
- iium, 39
- iium_corresp, 40
- iium_ssr, 41
- iivg_mco, 42
- ileg_had, 43, 56
- ileg_mco, 44, 61, 64, 67
- ileg_ssr, 45, 57, 58, 65
- imed_had, 46
- imed_mco, 47, 49, 87
- imed_ssr, 48
- inner_tra, 49
- ipie, 50, 87
- ipo, 51, 87, 92
- ir3a, 52, 59, 60, 67
- iraa, 53
- irafael, 36, 54
- irapss, 32, 44, 47, 49, 55, 67, 89
- irha, 37, 45, 49, 56, 65, 67, 85
- irhs, 57
- irps, 58
- irpsa, 35, 49, 53, 54, 59, 67
- irsa, 25, 34, 38–43, 45, 48, 49, 51, 52, 60, 64, 66, 67, 85
- irsf, 62
- irum, 25, 34, 38, 39, 43, 45, 48, 49, 51, 52, 61, 63, 66, 67, 85
- issrha, 45, 64
- itpt_mco, 65

itra, [67](#)

labeleasier, [68](#)
lancer_requete, [69](#)
lancer_requete_db, [70](#)

noyau_pmeasyr, [4–6](#), [32](#), [34–37](#), [40–45](#),
[47–49](#), [52–62](#), [64–67](#), [71](#), [72](#), [86](#), [89](#),
[93](#)
noyau_skeleton, [71](#), [72](#)

parse_nom_fichier, [74](#)
parse_nom_fichier(), [80](#)
parse_noms_fichiers, [73](#)
parse_noms_fichiers(), [74](#)
prepare_rsa, [75](#)
print.pm_param (noyau_pmeasyr), [71](#)

read_fwf, [32–42](#), [44](#), [46–48](#), [50–53](#), [55–60](#),
[63](#), [65–67](#)
renomme_fichier, [76](#)
requete, [77](#)
requete_db, [78](#)

selectionne_archive, [79](#)
selectionne_fichiers, [80](#)

tbl_had, [80](#)
tbl_mco, [81](#)
tbl_psy, [82](#)
tbl_rsf, [83](#)
tbl_ssr, [84](#)
tdiag, [84](#)

utils::unzip(), [6](#), [13](#)
utils::View(), [13](#)

vvr_ano_mco, [85](#), [87](#), [88](#), [91–93](#)
vvr_ghs_supp, [87](#)
vvr_had_ghs, [88](#)
vvr_libelles_valo, [90](#)
vvr_mco, [26–29](#), [86](#), [88](#), [91](#), [92–96](#)
vvr_mco_ano, [92](#)
vvr_mco_sv, [92](#)
vvr_rsa, [86–88](#), [91](#), [92](#), [93](#), [94](#)
vvr_rsa_hors_periode, [94](#)
vvr_rum, [28](#), [95](#), [96](#)
vvr_rum_check_rubriques_rav, [96](#)