Package: sejmRP (via r-universe)

October 13, 2024

Title An Information About Deputies and Votings in Polish Diet from Seventh to Eighth Term of Office

Version 1.3.4

Date 2017-03-28

Maintainer Piotr Smuda <piotrsmuda@gmail.com>

Description Set of functions that access information about deputies and votings in Polish diet from webpage <http://www.sejm.gov.pl>. The package was developed as a result of an internship in MI2 Group - <http://mi2.mini.pw.edu.pl>, Faculty of Mathematics and Information Science, Warsaw University of Technology.

BugReports <http://github.com/mi2-warsaw/sejmRP/issues>

Depends R (>= 3.1.0)

License GPL-2

LazyLoad true

LazyData true

Imports DBI, dbplyr, dplyr, tidyr, cluster, factoextra, MASS, ggplot2, RPostgreSQL, rvest, stringi, XML, xml2

RoxygenNote 6.0.1

Repository https://mi2-warsaw.r-universe.dev

RemoteUrl https://github.com/mi2-warsaw/sejmrp

RemoteRef HEAD

RemoteSha 4c7a629d375be9c22add22d3b65a1f425746a3a9

Contents

`create_database` *Creating database*

Description

Function `create_database` creates a database with four empty tables: deputies, votings, votes, statements.

Usage

```
create_database(dbname, user, password, host)
```

Arguments

dbname	name of database
user	name of user
password	password of database
host	name of host

Details

```
Created tables:
1. deputies with columns:
    1) id_deputy - deputy's id,
    2) nr_term_of_office - Polish Diet's number of term of office,
    3) surname_name - deputy's names and surnames,
2. votings with columns:
    1) id_voting - voting's id,
    2) nr_term_of_office - Polish Diet's number of term of office,
    3) nr_meeting - meeting's number,
    4) date_meeting - meeting's date,
    5) nr_voting - voting's number,
    6) topic_voting - voting's topic,
    7) link_results - link with voting's results,
3. votes with columns:
    1) id_vote - vote's id,
    2) nr_term_of_office - Polish Diet's number of term of office,
    3) id_deputy - deputy's id,
    4) id_voting - voting's id,
    5) vote - deputy's vote, one of: 'Za','Przeciw',
              'Wstrzymal sie','Nieobecny',
    6) club - deputy's club,
4. statements with columns:
    1) id_statement - statement's id, like:
    (meeting's number).(voting's number).(statement's number),
    2) nr_term_of_office - Polish Diet's number of term of office,
    3) surname_name - author of statement,
    4) date_statement - statement's date,
    5) titles_order_points - title of order points,
    6) statement - content of statement.
```

Value

invisible NULL

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
create_database(dbname, user, password, host)
## End(Not run)
```

`deputies_add_new` *Adding new deputies to table*

Description

Function `deputies_add_new` adds new deputies to a table with deputies.

Usage

```
deputies_add_new(dbname, user, password, host, type, id,
  nr_term_of_office = 8)
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host
`type`	type of deputies which be add to table with deputies: active, inactive
`id`	id of deputies from which we start add new deputies
`nr_term_of_office`	number of term of office of Polish Diet; default: 8

Details

Function `deputies_add_new` adds new deputies to a table with deputies. Also there is a choice between types of deputies, because on the page of Polish diet deputies are splitted into *active* and *inactive*. In addition id of the last added deputy in *deputies* table is needed.

Value

invisible NULL

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
deputies_add_new(dbname, user, password, host, 'active', id)
deputies_add_new(dbname, user, password, host, 'inactive', id)
## End(Not run)
```

---

deputies_create_table *Creating table with deputies*

Description

Function `deputies_create_table` creates a table with deputies.

Usage

```
deputies_create_table(dbname, user, password, host,
   nr_term_of_office = 8)
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host
`nr_term_of_office`	number of term of office of Polish Diet; default: 8

Value

invisible NULL

Note

Use only this function for first time, when the *deputies* table is empty. Then use `deputies_update_table`.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
deputies_create_table(dbname, user, password, host)
## End(Not run)
```

`deputies_get_data` *Getting data about deputies*

Description

Function `deputies_get_data` gets data about deputies.

Usage

```
deputies_get_data(type, nr_term_of_office = 8)
```

Arguments

`type`	type of deputies which be add to table with deputies: active, inactive
`nr_term_of_office`	number of term of office of Polish Diet; default: 8

Details

Function `deputies_get_data` gets deputies' ids and personal data like name and surname. Also there is a choice between types of deputies, because on the page of Polish diet deputies are splitted into *active* and *inactive*.

Value

data frame with two columns: id_deputy, surname_name

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
deputies_get_data('active')
deputies_get_data('inactive')
## End(Not run)
```

deputies_get_ids *Getting deputies' ids*

Description

Function deputies_get_ids gets deputies' ids from *deputies* table.

Usage

```
deputies_get_ids(dbname, user, password, host,
   nr_term_of_office = 8, windows = .Platform$OS.type == 'windows')
```

Arguments

dbname	name of database
user	name of user
password	password of database
host	name of host
nr_term_of_office	number of term of office of Polish Diet; default: 8
windows	information of used operation system; default: .Platform$OS.type == 'windows'

Details

Function deputies_get_ids gets deputies' ids from *deputies* table. As result of this function you get named character vector with ids, where their names are names and surnames of deputies. Because of encoding issue on Windows operation system, you need to select if you use Windows.

Value

named character vector

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
deputies_get_ids(dbname, user, password, host, TRUE)
deputies_get_ids(dbname, user, password, host, FALSE)
## End(Not run)
```

`deputies_update_table` *Updating table with deputies*

Description

Function `deputies_update_table` updates a table with deputies.

Usage

```
deputies_update_table(dbname, user, password, host,
   nr_term_of_office = 8)
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host
`nr_term_of_office`	number of term of office of Polish Diet; default: 8

Value

invisible NULL

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
deputies_update_table(dbname, user, password, host)
## End(Not run)
```

get_deputies_dendrogram *Converts Deputies Distance Matrix into a Deputies Dendrogram*

Description

Function get_deputies_dendrogram converts the distance matrix between deputies into a dendrogram or ggplot of the dendrogram.

Usage

```
get_deputies_dendrogram(distances, plot = TRUE, method = "ward", k = NULL)
```

Arguments

distances	a distance matrix, preferably created with get_distance_matrix
plot	if TRUE then the ggplot object will be returned instead of the dendrogram itself
method	clustering method, see the agnes for more details
k	number of groups, will be passed to fviz_dend

Value

dendrogram or a ggplot

Author(s)

Przemyslaw Biecek

Examples

```
# votes <- get_filtered_votes(terms_of_office = c(7,7))
data(votes)
v <- c(`Za` = 5, `Przeciw` = -5, `Wstrzyma<c5><82> si<c4><99>` = 2, `Nieobecny` = 0)/10
mat2 <- get_distance_matrix(votes[,c("surname_name", "id_voting", "vote")], weights = v)
get_deputies_dendrogram(mat2, k=5)
```

get_deputies_mds *Converts Deputies Distance Matrix into a Multidimensional Scaling Plot*

Description

Function get_deputies_mds converts the distance matrix between deputies into a 2D representation (Multidimensional Scaling)

Usage

```
get_deputies_mds(distances, clubs = NULL, plot = TRUE,
  remove_missing_clubs = TRUE)
```

Arguments

distances	a distance matrix, preferably created with get_distance_matrix
clubs	a data.frame that maps deputy_id (names from distances) on clubs. It should contain at least one column - club and rows should have names corresponding to names in the distances.
plot	if TRUE then the ggplot object will be returned instead of the dendrogram itself
remove_missing_clubs	if TRUE then rows of distances that are not mapped in clubs will be removed

Value

MDS coordinates or a ggplot

Author(s)

Przemyslaw Biecek

Examples

```
# votes <- get_filtered_votes(terms_of_office = c(7,7))
library(dplyr)
library(ggplot2)
data(votes)
v <- c(`Za` = 5, `Przeciw` = -5, `Wstrzyma<c5><82> si<c4><99>` = 2, `Nieobecny` = 0)/10
mat2 <- get_distance_matrix(votes[,c("surname_name", "id_voting", "vote")], weights = v)
df <- votes[,c("surname_name","club")]
df %>%
  group_by(surname_name, club) %>%
  summarise(n = n()) %>%
  arrange(-n) %>%
  group_by(surname_name) %>%
  top_n(1) %>%
  as.data.frame() -> clubs
```

```
row.names(clubs) <- clubs[,1]
clubs$club[clubs$club == "niez."] = "cross-bencher"
get_deputies_mds(mat2, clubs)

# without cross bencher deputies
clubs2 <- clubs[clubs$club != "cross-bencher",]
get_deputies_mds(mat2, clubs2)
```

`get_deputies_silhouette` *Converts Deputies Distance Matrix into a Silhouette Plot*

Description

Function `get_deputies_silhouette` converts the distance matrix between deputies and their clubs into a silhouette plot

Usage

```
get_deputies_silhouette(distances, clubs = NULL, plot = TRUE,
  remove_missing_clubs = TRUE)
```

Arguments

`distances`	a distance matrix, preferably created with `get_distance_matrix`
`clubs`	a data.frame that maps `deputy_id` (names from `distances`) on clubs. It should contain at least one column - `club` and rows should have names corresponding to names in the `distances`.
`plot`	if `TRUE` then the `ggplot` object will be returned instead of the dendrogram itself
`remove_missing_clubs`	if TRUE then rows of `distances` that are not mapped in `clubs` will be removed

Value

silhouette object or a ggplot

Author(s)

Przemyslaw Biecek

Examples

```
library("dplyr")
# votes <- get_filtered_votes(terms_of_office = c(7,7))
data(votes)
v <- c(`Za` = 5, `Przeciw` = -5, `Wstrzyma<c5><82> si<c4><99>` = 2, `Nieobecny` = 0)/10
mat2 <- get_distance_matrix(votes[,c("surname_name", "id_voting", "vote")], weights = v)
df <- votes[,c("surname_name","club")]
df %>%
  group_by(surname_name, club) %>%
  summarise(n = n()) %>%
  arrange(-n) %>%
  group_by(surname_name) %>%
  top_n(1) %>%
  as.data.frame() -> clubs
row.names(clubs) <- clubs[,1]
clubs$club[clubs$club == "niez."] = "cross-bencher"

get_deputies_silhouette(mat2, clubs)
```

---

get_deputies_table *Importing deputies table from a database*

Description

Function get_deputies_table imports deputies table from a database.

Usage

```
get_deputies_table(dbname = 'sejmrp', user = 'reader',
  password = 'qux94874', host = 'services.mini.pw.edu.pl',
  sorted_by_id = TRUE, windows = .Platform$OS.type == 'windows')
```

Arguments

dbname	name of database; default: 'sejmrp'
user	name of user; default: 'reader'
password	password of database; default: 'qux94874'
host	name of host; default: 'services.mini.pw.edu.pl'
sorted_by_id	information if table should be sorted by id; default: TRUE
windows	information of used operation system; default: .Platform$OS.type == 'windows'

Details

Function get_deputies_table imports deputies table from a database. The result of this function is a data frame with deputies' data. Because of encoding issue on Windows operation system, you need to select if you use Windows.

Value

data frame

Note

Default parameters use privilages of 'reader'. It can only SELECT data from database.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
deputies <- get_deputies_table()
dim(deputies)
# [1] 983 3
names(deputies)
# [1] 'id_deputy' 'nr_term_of_office' 'surname_name'
## End(Not run)
```

---

get_distance_matrix *Converts Voting Table into the Deputies Distance Matrix*

Description

Function `get_distance_matrix` converts a data frame with tree columns (deputies' ids, votings' ids and votes) into a deputies distance matrix.

Usage

```
get_distance_matrix(votes, weights = NULL, allowMissings = 0)
```

Arguments

`votes`	a data frame with three columns, respectively: deputy id, voting id, vote
`weights`	if supplied it should be a named vector that converts votes into a numeric values that correspond to their similarity
`allowMissings`	maximum number of missing votigs allowd for deputy.

Details

Function `get_distance_matrix` calculated distances among deputies based on their votes The more similar are the voting the smaller distance between deputies.

Value

distance matrix

Author(s)

Przemyslaw Biecek

Examples

```
# votes <- get_filtered_votes(terms_of_office = c(7,7))
data(votes)
v <- c(`Za` = 5, `Przeciw` = -5, `Wstrzyma<c5><82> si<c4><99>` = 2, `Nieobecny` = 0)/10
mat2 <- get_distance_matrix(votes[,c("surname_name", "id_voting", "vote")], weights = v)
## Not run:
mat1 <- get_distance_matrix(votes[,c("surname_name", "id_voting", "vote")])

## End(Not run)
```

---

get_filtered_statements

Retrieve filtered statements from a database

Description

Function `get_filtered_statements` reads filtered statements from a database.

Usage

```
get_filtered_statements(dbname = 'sejmrp', user = 'reader',
  password = 'qux94874', host = 'services.mini.pw.edu.pl',
  windows = .Platform$OS.type == 'windows', terms_of_office = integer(0),
  deputies = character(0), dates = character(0), topics = character(0),
  content = character(0), max_rows = Inf)
```

Arguments

`dbname`	name of database; default: 'sejmrp'
`user`	name of user; default: 'reader'
`password`	password of database; default: 'qux94874'
`host`	name of host; default: 'services.mini.pw.edu.pl'
`windows`	information of used operation system; default: .Platform$OS.type == 'windows'
`terms_of_office`	range of terms of office's numbers that will be taken to filter data from database; default: integer(0)

deputies	full names of deputies that will be taken to filter data from database; default: character(0)
dates	period of time that will be taken to filter data from database; default: character(0)
topics	text patterns that will be taken to filter data from database; default: character(0)
content	text patterns that will be taken to filter data from database; default: character(0)
max_rows	maximum number of rows to download; default: Inf

Details

Function get_filtered_statements reads filtered statements from a database. The result of this function is an invisible data frame with statements' data.

Possible filters:

1. terms_of_office - range of terms of office's numbers. This filter is a integer vector with two elements, where the first describes a left boundary of range and the second a right boundary. It is possible to choose only one term of office, just try the same number as first and second element of vector.
2. deputies - full names of deputies. This filter is a character vector with full names of deputies in format: 'surname first_name second_name'. If you are not sure if the deputy you were thinking about has second name, try 'surname first_name' or just 'surname'. There is high probability that proper deputy will be chosen. It is possible to choose more than one deputy.
3. dates - period of time. This filter is a character vector with two elements in date format 'YYYY-MM-DD', where the first describes left boundary of period and the second right boundary. It is possible to choose only one day, just try the same date as first and second element of vector.
4. topics - text patterns. This filter is a character vector with text patterns of topics in order points. Note that the order points are written like sentences, so remember about case inflection of nouns and adjectives and use stems of words as patterns. For example if you want to find order points about education (in Polish: szkolnictwo) try 'szkolnictw'. It is possible to choose more than one pattern.
5. content - text patterns. This filter is a character vector with text patterns in statements. Note that strings with statements are sentences, so remember about case inflection of nouns and adjectives and use stems of words as patterns. For example if you want to find order points about education (in Polish: szkolnictwo) try 'szkolnictw'. It is possible to choose more than one pattern.

If you did not choose any filter, the whole database will be downloaded. Note that, due to data size (<= ~150 MB) it may take few seconds / minutes to download all statements.

Because of encoding issue on Windows operation system, you also need to select if you use Windows.

Value

data frame with NULL

Note

Default parameters use privilages of 'reader'. It can only SELECT data from database.

All information is stored in PostgreSQL database.

Author(s)

Tomasz Mikolajczyk, Piotr Smuda

Examples

```
## Not run:
filtered_statements <- get_filtered_statements()
dim(filtered_statements)
# [1] 2568       6
names(filtered_statements)
[1] 'id_statement' 'nr_term_of_office' 'surname_name' 'date_statement'
[5] 'titles_order_points' 'statement'
object.size(filtered_statements)
# 6488552 bytes
## End(Not run)
```

`get_filtered_votes` *Retrieve filtered votes from a database*

Description

Function `get_filtered_votes` reads filtered votes from a database.

Usage

```
get_filtered_votes(dbname = 'sejmrp', user = 'reader',
  password = 'qux94874', host = 'services.mini.pw.edu.pl',
  windows = .Platform$OS.type == 'windows', clubs = character(0),
  dates = character(0), terms_of_office = integer(0),
  meetings = integer(0), votings = integer(0),
  deputies = character(0), topics = character(0), max_rows = Inf)
```

Arguments

`dbname`	name of database; default: 'sejmrp'
`user`	name of user; default: 'reader'
`password`	password of database; default: 'qux94874'
`host`	name of host; default: 'services.mini.pw.edu.pl'
`windows`	information of used operation system; default: .Platform$OS.type == 'windows'

clubs	names of clubs that will be taken to filter data from database; default: character(0)
dates	period of time that will be taken to filter data from database; default: character(0)
terms_of_office	range of terms of office's numbers that will be taken to filter data from database; default: integer(0)
meetings	range of meetings' numbers that will be taken to filter data from database; default: integer(0)
votings	range of votings' numbers that will be taken to filter data from database; default: integer(0)
deputies	full names of deputies that will be taken to filter data from database; default: character(0)
topics	text patterns that will be taken to filter data from database; default: character(0)
max_rows	maximum number of rows to download; default: Inf

Details

Function `get_filtered_votes` reads filtered votes from a database. The result of this function is an invisible data frame with statements' data.

Possible filters:

1. clubs - names of clubs. This filter is a character vector with elements like for example: 'PO', 'PiS', 'SLD'. It is possible to choose more than one club.
2. dates - period of time. This filter is a character vector with two elements in date format 'YYYY-MM-DD', where the first describes left boundary of period and the second right boundary. It is possible to choose only one day, just try the same date as first and second element of vector.
3. terms_of_office - range of terms of office's numbers. This filter is a integer vector with two elements, where the first describes a left boundary of range and the second a right boundary. It is possible to choose only one term of office, just try the same number as first and second element of vector.
4. meetings - range of meetings' numbers. This filter is a integer vector with two elements, where the first describes a left boundary of range and the second a right boundary. It is possible to choose only one meeting, just try the same number as first and second element of vector.
5. votings - range of votings' numbers. This filter is a integer vector with two elements, where the first describes a left boundary of range and the second a right boundary. It is possible to choose only one voting, just try the same number as first and second element of vector.
6. deputies - full names of deputies. This filter is a character vector with full names of deputies in format: 'surname first_name second_name'. If you are not sure if the deputy you were thinking about has second name, try 'surname first_name' or just 'surname'. There is high probability that proper deputy will be chosen. It is possible to choose more than one deputy.
7. topics - text patterns. This filter is a character vector with text patterns of topics that you are interested about. Note that the votings' topics are written like sentences, so remember about case inflection of nouns and adjectives and use stems of words as patterns. For example if you want to find votings about education (in Polish: szkolnictwo) try 'szkolnictw'. It is possible to choose more than one pattern.

If you did not choose any filter, the whole database will be downloaded. Note that, due to data size (<= ~150 MB) it may take few seconds / minutes to download all votes.

Because of encoding issue on Windows operation system, you also need to select if you use Windows.

Value

data frame with NULL

Note

Default parameters use privilages of 'reader'. It can only SELECT data from database.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
filtered_votes <- get_filtered_votes()
dim(filtered_votes)
# [1] 2826483       9
names(filtered_votes)
[1] 'surname_name' 'nr_term_of_office' 'club' 'vote' 'id_voting'
[6] 'nr_meeting' 'nr_voting' 'date_meeting' 'topic_voting'
object.size(filtered_votes)
# 148694336 bytes
## End(Not run)
```

---

get_most_frequent_club

Gets the Most Frequent Club for Each Deputy

Description

One deputy may belong to many different clubs and change clubs over time. The function `get_most_frequent_club` calculates the most frequent club for each deputy.

Usage

```
get_most_frequent_club(deputy_id, club)
```

Arguments

`deputy_id`	a vector with deputy unique ids (may be also a vactor with characters name/surname)
`club`	vector of length equal to `deputy_id` with club memberships for particular deputies

Value

a data frame with club memberships of deputies

Author(s)

Przemyslaw Biecek

Examples

```
# votes <- get_filtered_votes(terms_of_office = c(7,7))
data(votes)
clubs <- get_most_frequent_club(votes$surname_name, votes$club)
head(clubs)
```

---

get_statements_table *Importing statements table from a database*

Description

Function get_statements_table imports statements table from a database.

Usage

```
get_statements_table(dbname = 'sejmrp', user = 'reader',
  password = 'qux94874', host = 'services.mini.pw.edu.pl',
  sorted_by_id = TRUE, windows = .Platform$OS.type == 'windows')
```

Arguments

dbname	name of database; default: 'sejmrp'
user	name of user; default: 'reader'
password	password of database; default: 'qux94874'
host	name of host; default: 'services.mini.pw.edu.pl'
sorted_by_id	information if table should be sorted by id; default: TRUE
windows	information of used operation system; default: .Platform$OS.type == 'windows'

Details

Function get_statements_table imports statements table from a database. The result of this function is a data frame with statements' data. Because of encoding issue on Windows operation system, you need to select if you use Windows.

Value

data frame

Note

Default parameters use privilages of 'reader'. It can only SELECT data from database.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
statements <- get_statements_table()
dim(statements)
# [1] 43432 6
names(statements)
# [1] 'id_statement' 'nr_term_of_office' 'surname_name'
# [4] 'date_statement' 'titles_order_points' 'statement'
## End(Not run)
```

---

get_votes_table *Importing votes table from a database*

Description

Function `get_votes_table` imports votes table from a database.

Usage

```
get_votes_table(dbname = 'sejmrp', user = 'reader',
  password = 'qux94874', host = 'services.mini.pw.edu.pl',
  sorted_by_id = TRUE, windows = .Platform$OS.type == 'windows')
```

Arguments

`dbname`	name of database; default: 'sejmrp'
`user`	name of user; default: 'reader'
`password`	password of database; default: 'qux94874'
`host`	name of host; default: 'services.mini.pw.edu.pl'
`sorted_by_id`	information if table should be sorted by id; default: TRUE
`windows`	information of used operation system; default: .Platform$OS.type == 'windows'

Details

Function `get_votes_table` imports votes table from a database. The result of this function is a data frame with votes' data. Because of encoding issue on Windows operation system, you need to select if you use Windows.

Value

data frame

Note

Default parameters use privilages of 'reader'. It can only SELECT data from database.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votes <- get_votes_table()
dim(votes)
# [1] 2826483 6
names(votes)
# [1] 'id_vote' 'nr_term_of_office' 'id_deputy' 'id_voting' 'vote' 'club'
object.size(votes)
# 90474040 bytes
## End(Not run)
```

---

get_votings_table *Importing votings table from a database*

Description

Function `get_votings_table` imports votings table from a database.

Usage

```
get_votings_table(dbname = 'sejmrp', user = 'reader',
  password = 'qux94874', host = 'services.mini.pw.edu.pl',
  sorted_by_id = TRUE, windows = .Platform$OS.type == 'windows')
```

Arguments

`dbname`	name of database; default: 'sejmrp'
`user`	name of user; default: 'reader'
`password`	password of database; default: 'qux94874'
`host`	name of host; default: 'services.mini.pw.edu.pl'
`sorted_by_id`	information if table should be sorted by id; default: TRUE
`windows`	information of used operation system; default: .Platform$OS.type == 'windows'

Details

Function `get_votings_table` imports votings table from a database. The result of this function is a data frame with votings' data. Because of encoding issue on Windows operation system, you need to select if you use Windows.

Value

data frame

Note

Default parameters use privilages of 'reader'. It can only SELECT data from database.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votings <- get_votings_table()
dim(votings)
# [1] 6212 7
names(votings)
# [1] 'id_voting' 'nr_term_of_office' 'nr_meeting'
# [4] 'date_meeting' 'nr_voting' 'topic_voting'
# [7] 'link_results'
## End(Not run)
```

`remove_database` *Removing database*

Description

Function `remove_database` remove whole database.

Usage

```
remove_database(dbname, user, password, host)
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host

Value

invisible NULL

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
remove_database(dbname, user, password, host)
## End(Not run)
```

---

safe_html *Safe html scrapping*

Description

Function `safe_html` tries to download the URL several times.

Usage

```
safe_html(page, time = 60, attempts = 10)
```

Arguments

`page`	requested URL
`time`	sleep interval after each failure
`attempts`	max number of tries (if there is a problem with connection)

Details

Function `safe_html` performes 10 (by default) attempts to download the URL and waits 60sec (by default) after each failure

Value

character vector

Author(s)

Przemyslaw Biecek

Examples

```
## Not run:
page <- paste0('http://www.sejm.gov.pl/Sejm7.nsf/',
               'wypowiedz.xsp?posiedzenie=15&dzien=1&wyp=008')
safe_html(page)
## End(Not run)
```

`safe_readHTMLTable` *Safe html table scrapping*

Description

Function `safe_readHTMLTable` tries to download the table from given URL several times.

Usage

```
safe_readHTMLTable(..., time = 60, attempts = 10)
```

Arguments

`...`	arguments that will be passed to readHTMLTable
`time`	sleep interval after each failure
`attempts`	max number of tries (if there is a problem with connection)

Details

Function `safe_readHTMLTable` performes 10 (by default) attempts to download the URL and waits 60sec (by default) after each failure

Value

character vector

Author(s)

Przemyslaw Biecek

Examples

```
## Not run:
page <- paste0('http://www.sejm.gov.pl/Sejm7.nsf/',
               'posiedzenie.xsp?posiedzenie=99&dzien=2')
safe_readHTMLTable(page)
## End(Not run)
```

statements_create_table *Creating table with deputies' statements*

Description

Function `statements_create_table` creates a table with deputies' statements.

Usage

```
statements_create_table(dbname, user, password, host,
  nr_term_of_office = 8)
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host
`nr_term_of_office`	number of term of office of Polish Diet; default: 8

Value

invisible NULL

Note

Use only this function for first time, when the *statements* table is empty. Then use `statements_update_table`.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda, Tomasz Mikolajczyk

Examples

```
## Not run:
statements_create_table(dbname, user, password, host)
## End(Not run)
```

`statements_get_statement` *Getting statements*

Description

Function `statements_get_statement` gets statement's content.

Usage

```
statements_get_statement(page, ...)
```

Arguments

`page`	deputy's statement's page
`...`	other arguments, that will be passed to safe_html()

Details

Function `statements_get_statement` gets statement's content. Example of page with deputy's statement: http://www.sejm.gov.pl/Sejm7.nsf/wypowiedz.xsp?posiedzenie=15&dzien=1&wyp=008

Value

character vector

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda, Tomasz Mikolajczyk

Examples

```
## Not run:
page <- paste0('http://www.sejm.gov.pl/Sejm7.nsf/',
               'wypowiedz.xsp?posiedzenie=15&dzien=1&wyp=008')
statements_get_statement(page)
## End(Not run)
```

statements_get_statements_data *Getting data about statements*

Description

Function statements_get_statements_data gets data about statements.

Usage

```
statements_get_statements_data(statements_links,
  home_page = 'http://www.sejm.gov.pl/')
```

Arguments

statements_links	list of elements of XMLNodeSet class with statements' ids, links and their's authors
home_page	main page of polish diet: http://www.sejm.gov.pl/

Details

Function statements_get_statements_data gets data about statements like author, page with content of statement and it's id.

Value

data frame with three columns: names, statements_links, ids

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda, Tomasz Mikolajczyk

Examples

```
## Not run:
page <- safe_html(paste0('http://www.sejm.gov.pl/Sejm7.nsf/',
               'wypowiedz.xsp?posiedzenie=15&dzien=1&wyp=0'))
page <- html_nodes(page, '.stenogram')
statements_links <- html_nodes(page, 'h2 a')
statements_get_statements_data(statements_links,
  home_page = 'http://www.sejm.gov.pl/Sejm7.nsf/')
## End(Not run)
```

`statements_get_statements_table` *Getting statements' table*

Description

Function `statements_get_statements_table` gets statements' table from meeting's page.

Usage

```
statements_get_statements_table(page)
```

Arguments

`page`	meeting's page

Details

Function `statements_get_statements_table` gets statements' table. from meeting's page. Example of a meeting's page: http://www.sejm.gov.pl/Sejm7.nsf/posiedzenie.xsp?posiedzenie=99&dzien=2 The result of this function is a data frame with three columns, where the first includes author of statement, the second the number of order point and the third is a title of order point.

Value

data frame with three unnamed columns

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
page <- 'http://www.sejm.gov.pl/Sejm7.nsf/posiedzenie.xsp?posiedzenie=99&dzien=2'
statements_get_statements_table(page)
## End(Not run)
```

statements_update_table *Updating table with deputies' statements*

Description

Function `statements_update_table` updates a table with deputies' statements.

Usage

```
statements_update_table(dbname, user, password, host,
  nr_term_of_office = 8, verbose = FALSE)
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host
`nr_term_of_office`	number of term of office of Polish Diet; default: 8
`verbose`	if TRUE then additional info will be printed

Value

invisible NULL

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda, Tomasz Mikolajczyk

Examples

```
## Not run:
statements_update_table(dbname, user, password, host)
## End(Not run)
```

votes *Votes from 7th Office of Polish Sejm*

Description

Votes taken in the 7th office of Polish Sejm (2011-2015)

- surname_name Surname and name of a deputy
- nr_term_of_office Which office? In this sample there is only 7th office
- club club of the deputy at the moment of voting (may change in time)
- vote vote taken in the voting
- id_voting unique id of the voting
- nr_meeting no of the meeting
- nr_voting no of the voting
- date_meeting data of the meeting
- topic_voting full title of the voting

Usage

```
data(votes)
```

Format

2890479 rows and 9 columns

votes_create_table *Creating table with votes*

Description

Function votes_create_table creates a table with votes.

Usage

```
votes_create_table(dbname, user, password, host,
  nr_term_of_office = 8, windows = .Platform$OS.type == 'windows')
```

Arguments

dbname	name of database
user	name of user
password	password of database
host	name of host
nr_term_of_office	number of term of office of Polish Diet; default: 8
windows	information of used operation system; default: .Platform$OS.type == 'windows'

Value

invisible NULL

Note

Use only this function for first time, when the *votes* table is empty. Then use `votes_update_table`.

There is a possibility that someone's voice reader broke during voting and this situation is treated like this deputy was absent. Even if deputy made a decision, he's/she's vote is 'Nieobecny'.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votes_create_table(dbname, user, password, host, 7, TRUE)
votes_create_table(dbname, user, password, host, 7, FALSE)
## End(Not run)
```

`votes_get_clubs_links` *Getting links with voting's results for each club*

Description

Function `votes_get_clubs_links` gets links with voting's results for each club from voting's page.

Usage

```
votes_get_clubs_links(home_page = 'http://www.sejm.gov.pl/Sejm8.nsf/',
  page)
```

Arguments

`home_page`	main page of polish diet: http://www.sejm.gov.pl/Sejm8.nsf/
`page`	voting's page

Details

Function `votes_get_clubs_links` gets links with voting's results for each club from voting's page. Example of a voting's page: http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=glosowania&NrKadencji=7&NrPosiedzenia=1&NrGlosowania=1

Value

data frame with two columns: club, links

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
home_page <- 'http://www.sejm.gov.pl/Sejm7.nsf/'
page <- paste0('http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?',
  'symbol=glosowania&NrKadencji=7&NrPosiedzenia=1&NrGlosowania=1')
votes_get_clubs_links(home_page, page)
## End(Not run)
```

---

votes_get_results *Getting voting's results for each club*

Description

Function `votes_get_results` gets voting's results for each club.

Usage

```
votes_get_results(page)
```

Arguments

page	club's voting's results page

Details

Function `votes_get_results` gets voting's results for each club. Example of page with voting's results of PO club: http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=klubglos& IdGlosowania=37494&KodKlubu=PO

Value

data frame with two columns: deputy, vote

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
page <- paste0('http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?',
           'symbol=klubglos&IdGlosowania=37494&KodKlubu=PO')
votes_get_results(page)
## End(Not run)
```

`votes_match_deputies_ids` *Matching deputies to theirs' ids*

Description

Function `votes_match_deputies_ids` matches deputies from voting's results page to theirs' ids from *deputies* table.

Usage

```
votes_match_deputies_ids(dbname, user, password, host, page,
   nr_term_of_office = 8, windows = .Platform$OS.type == 'windows')
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host
`page`	club's voting's results page
`nr_term_of_office`	number of term of office of Polish Diet; default: 8
`windows`	information of used operation system; default: .Platform$OS.type == 'windows'

Details

Function `votes_match_deputies_ids` matches deputies from voting's results page to theirs' ids from *deputies* table. The result of this function is a data frame with deputies' data, ids and votes. Because of encoding issue on Windows operation system, you need to select if you use Windows. Example of page with voting's results of PO club: http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=klubglos&IdGlosowania=37494&KodKlubu=PO

Value

data frame with three columns: deputy, vote, id

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
page <- paste0('http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?',
            'symbol=klubglos&IdGlosowania=37494&KodKlubu=PO')
votes_match_deputies_ids(dbname, user, password, host, page, 7, TRUE)
votes_match_deputies_ids(dbname, user, password, host, page, 7, FALSE)
## End(Not run)
```

---

votes_update_table *Updating table with votes*

Description

Function votes_update_table updates a table with votes.

Usage

```
votes_update_table(dbname, user, password, host,
  nr_term_of_office = 8, windows = .Platform$OS.type == 'windows',
  verbose = FALSE)
```

Arguments

dbname	name of database
user	name of user
password	password of database
host	name of host
nr_term_of_office	number of term of office of Polish Diet; default: 8
windows	information of used operation system; default: .Platform$OS.type == 'windows'
verbose	if TRUE then additional info will be printed

Value

invisible NULL

Note

There is a possibility that someone's voice reader broke during voting and this situation is treated like this deputy was absent. Even if deputy made a decision, he's/she's vote is 'Nieobecny'.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votes_update_table(dbname, user, password, host, 7, TRUE)
votes_update_table(dbname, user, password, host, 7, FALSE)
## End(Not run)
```

---

votings_create_table *Creating table with votings*

Description

Function `votings_create_table` creates a table with votings.

Usage

```
votings_create_table(dbname, user, password, host,
  nr_term_of_office = 8)
```

Arguments

`dbname`	name of database
`user`	name of user
`password`	password of database
`host`	name of host
`nr_term_of_office`	number of term of office of Polish Diet; default: 8

Value

invisible NULL

Note

Use only this function for first time, when the *votings* table is empty. Then use `votings_update_table`.

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votings_create_table(dbname, user, password, host)
## End(Not run)
```

---

votings_get_date *Getting date of meeting*

Description

Function votings_get_date gets a date of meeting.

Usage

```
votings_get_date(page)
```

Arguments

page	meeting's page

Details

Example of a meeting's page: http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=listaglos&IdDnia=1179

Value

date in format YYYY-MM-DD as character

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
page <- 'http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=listaglos&IdDnia=1179'
votings_get_date(page)
## End(Not run)
```

votings_get_meetings_links *Getting meetings' links*

Description

Function votings_get_meetings_links gets meetings' links.

Usage

```
votings_get_meetings_links(
  home_page = 'http://www.sejm.gov.pl/Sejm8.nsf/', page =
  'http://www.sejm.gov.pl/Sejm8.nsf/agent.xsp?symbol=posglos&NrKadencji=8')
```

Arguments

home_page	main page of polish diet: http://www.sejm.gov.pl/Sejm8.nsf/
page	page with votings in polish diet: http://www.sejm.gov.pl/Sejm8.nsf/agent.xsp?symbol=posglos&NrKadencji=8

Value

character vector

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votings_get_meetings_links()
## End(Not run)
```

votings_get_meetings_table *Getting meetings' table*

Description

Function votings_get_meetings_table gets meetings' table.

Usage

```
votings_get_meetings_table(page =
  'http://www.sejm.gov.pl/Sejm8.nsf/agent.xsp?symbol=posglos&NrKadencji=8')
```

Arguments

page	page with votings in polish diet: http://www.sejm.gov.pl/Sejm8.nsf/agent.xsp?symbol=posglos&NrKadencji=8

Details

Function votings_get_meetings_table gets meetings' table. The result of this function is a data frame with three columns, where the first includes numbers of meetings, the second theirs' dates in Polish and the third is with numbers of votings on each meeting.

Value

data frame with three unnamed columns

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votings_get_meetings_table()
## End(Not run)
```

votings_get_votings_links *Getting votings' links*

Description

Function votings_get_votings_links gets votings' links from meeting's page.

Usage

```
votings_get_votings_links(home_page = 'http://www.sejm.gov.pl/Sejm8.nsf/',
  page)
```

Arguments

home_page	main page of polish diet: http://www.sejm.gov.pl/Sejm8.nsf/
page	meeting's page

Details

Example of a meeting's page: http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=listaglos&IdDnia=1179

Value

character vector

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
home_page <- 'http://www.sejm.gov.pl/Sejm7.nsf/'
page <- 'http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=listaglos&IdDnia=1179'
votings_get_votings_links(home_page, page)
## End(Not run)
```

votings_get_votings_table *Getting votings' table*

Description

Function votings_get_votings_table gets votings' table from meeting's page.

Usage

```
votings_get_votings_table(page)
```

Arguments

page	meeting's page

Details

Function votings_get_votings_table gets votings' table from meeting's page. Example of a meeting's page: http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=listaglos&IdDnia=1179 The result of this function is a data frame with three columns, where the first includes numbers of votings, the second voting's time and the third is with voting's topics.

Value

data frame with three columns: Nr, Godzina (Time), Temat (Topic)

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
page <- 'http://www.sejm.gov.pl/Sejm7.nsf/agent.xsp?symbol=listaglos&IdDnia=1179'
votings_get_votings_table(page)
## End(Not run)
```

votings_update_table *Updating table with votings*

Description

Function votings_update_table updates table with votings.

Usage

```
votings_update_table(dbname, user, password, host,
  nr_term_of_office = 8, verbose = FALSE)
```

Arguments

dbname	name of database
user	name of user
password	password of database
host	name of host
nr_term_of_office	number of term of office of Polish Diet; default: 8
verbose	if TRUE then additional info will be printed

Value

invisible NULL

Note

All information is stored in PostgreSQL database.

Author(s)

Piotr Smuda

Examples

```
## Not run:
votings_update_table(dbname, user, password, host)
## End(Not run)
```

Index