

Package ‘hammond’

August 2, 2019

Type Package

Title Useful analysis utilities

Version 0.1.0

Author David Hammond

Maintainer David Hammond <anotherdavidhammond@gmail.com>

Description Useful funtions

License GPL-3

Encoding UTF-8

Depends R (>= 3.5.0)

Imports tidyverse (>= 1.2.1),
countrycode (>= 1.1.0),
RPostgreSQL (>= 0.6.2),
whereami (>= 0.1.8.1),
digest (>= 0.6.19),
processx (>= 3.3.1),
fs (>= 1.3.1),
devtools (>= 2.0.2),
pbapply (>= 1.4-0),
Hmisc (>= 4.2-0),
padr (>= 0.4.2),
scales (>= 1.0.0)

LazyData true

RoxygenNote 6.1.1

R topics documented:

hammond-package	2
haddcountryinfo	2
hcorr	3
hcountrycode	3
hcountryexampledata	4
hcountryinfo	4
hdbkill	4
hdb_backup	5
hdb_connect	5
hdb_create_db	6

hdb_get	6
hdb_login	7
hdb_search	7
hdb_toc	8
hinterpolate	8
hpack_manual	9
hpc_change	9
hpopulation	9

Index	11
--------------	-----------

hammond-package	<i>hammond: some stuff</i>
-----------------	----------------------------

Description

Useful funtions

Installation

```
devtools::install_github("david-hammond/hammond")
```

haddcountryinfo	<i>haddcountryinfo</i>
-----------------	------------------------

Description

This function adds country specific information to a dataframe by matching countries to country codes.

Usage

```
haddcountryinfo(df)
```

Arguments

countries	list of countries
countries	list of countries

Examples

```
#need 4 column data frame, geocode, variablename, year, value
hcountryinfo(hcountryexempladata)
hcountry_info
```

This function calculates correlations between variables

```
#need 4 column data frame, geocode, variablename, year, value
```

hcorr	<i>hcorr</i>
-------	--------------

Description

This function calculates correlations between variables

Usage

```
hcorr(df, min.pairs = 20, verbose = TRUE, filter.by.p = FALSE)
```

Arguments

df	name of dataframe to use for correlation, needs to be long format 4 column data frame: geocode, variablename, year, value
min.pairs	minimum number of pairs to correlate
verbose	enable n and p values reporting, TRUE or FALSE
filter.by.p	Do you want to filter for significant p values?

Examples

```
#need 4 column data frame, geocode, variablename, year, value
library(hammond)
corr = hcorr(hcountryexempladata)
```

hcountrycode	<i>hcountrycode</i>
--------------	---------------------

Description

This function replaces country name or code with iso3c country codes. Can also be used in reverse.

Usage

```
hcountrycode(x, source_file = whereami::thisfile())
```

Arguments

countries	list of countries
-----------	-------------------

Examples

```
hcountrycode(hcountryexempladata$geocode)
```

hcountryexampledata	<i>countryinfo</i>
---------------------	--------------------

Description

countryinfo

Usage

hcountryexampledata

Format

An object of class `data.frame` with 11934 rows and 5 columns.

hcountryinfo	<i>countryinfo</i>
--------------	--------------------

Description

countryinfo

Usage

hcountryinfo

Format

An object of class `data.frame` with 233 rows and 7 columns.

hdbkill	<i>hdb_kill</i>
---------	-----------------

Description

This function kills all connections to the database, use as a last resort if you get a db connection error

Usage

hdbkill()

Arguments

countries list of countries

Examples

```
#need 4 column data frame, geocode, variablename, year, value
```

hdb_backup	<i>hdb_backup</i>
------------	-------------------

Description

This function backup a database, use only if you know what you are doing

Usage

```
hdb_backup(host = "192.168.0.98", user = "postgres",  
           password = "peace123", port = 5432)
```

Arguments

countries	list of countries
-----------	-------------------

Examples

```
#need 4 column data frame, geocode, variablename, year, value
```

hdb_connect	<i>hdb_connect</i>
-------------	--------------------

Description

This function calculates correlations between variables

Usage

```
hdb_connect(port = 5432)
```

Arguments

countries	list of countries
-----------	-------------------

Examples

```
#need 4 column data frame, geocode, variablename, year, value
```

hdb_create_db	<i>hdb_create_db</i>
---------------	----------------------

Description

This function creates a database, use only if you know what you are doing

Usage

```
hdb_create_db(host, db, user, password)
```

Arguments

countries	list of countries
-----------	-------------------

Examples

```
#need 4 column data frame, geocode, variablename, year, value
```

hdb_get	<i>hdb_get</i>
---------	----------------

Description

This function retrieves and caches data from any source in the database.

Usage

```
hdb_get(vars)
```

Arguments

countries	list of countries
-----------	-------------------

Examples

```
#need 4 column data frame, geocode, variablename, year, value
hdb_login("192.168.0.98", password = "peace123")
db_get("Perceptions of Criminality Raw")
```

hdb_login	<i>hdb_login</i>
-----------	------------------

Description

This function allows access to a database by entering an IP adress and passcode.

Usage

```
hdb_login(host = NULL, db = NULL, user = NULL, password = NULL)
```

Arguments

countries	list of countries
-----------	-------------------

Examples

```
hdb_login("192.168.0.64", db = "countrydata", user = "postgres", password = "peace123")
```

hdb_search	<i>hdb_search</i>
------------	-------------------

Description

This function searches the database and retrieves specified data.

Usage

```
hdb_search(vars)
```

Arguments

countries	list of countries
-----------	-------------------

```
hdb_login("192.168.0.98", password = "peace123") hdb_search("Criminal")
```

`hdb_toc``hdb_get_toc`

Description

This function retrieves the Table of Contents from a specified database.

Usage

```
hdb_toc()
```

Arguments

`countries` list of countries

Examples

```
#need 4 column data frame, geocode, variablename, year, value
hdb_login("192.168.0.98", password = "peace123")
hdb_get_toc()
```

`hinterpolate``hinterpolate`

Description

This is a wrapper function takes a data frame and fills in interpolated and extrapolated data for the whole time series

Usage

```
hinterpolate(df)
```

Arguments

`df` dataframe in iep format

Value

Returns list with filled in time series, column yhat is the interpolated value. Please check original value with yhat column to make sure you are happy with the results

Author(s)

Dave

Examples

```
hinterpolate(hcountryexampledata)
```

hpack_manual	<i>create package manual</i>
--------------	------------------------------

Description

This function calculates combinations for efficient correlation calculations

Usage

```
hpack_manual(pack = "hammond")
```

Arguments

pack	name of package
------	-----------------

hpc_change	<i>Calculate proportional change</i>
------------	--------------------------------------

Description

This function calculates proportional change in GPI for a country from one year to another.

Usage

```
hpc_change(all)
```

Arguments

all	the dataframe to be processed
-----	-------------------------------

Value

Returns a dataframe containing the raw and annual growths in GPI for each country `hpc_change(hcountryexampledata)`

hpopulation	<i>hpopulation</i>
-------------	--------------------

Description

This function appends a column of populations to a dataframe

Usage

```
hpopulation(df)
```

Arguments

df

Examples

```
x = hpopulation(hammond::hcountryexampledata)
```

Index

*Topic **analysis-utils**

hpc_change, [9](#)

*Topic **datasets**

hcountryexampledata, [4](#)

hcountryinfo, [4](#)

*Topic **imputation**

hinterpolate, [8](#)

*Topic **utilities**

hpc_change, [9](#)

haddcountryinfo, [2](#)

hammond (hammond-package), [2](#)

hammond-package, [2](#)

hcorr, [3](#)

hcountrycode, [3](#)

hcountryexampledata, [4](#)

hcountryinfo, [4](#)

hdb_backup, [5](#)

hdb_connect, [5](#)

hdb_create_db, [6](#)

hdb_get, [6](#)

hdb_login, [7](#)

hdb_search, [7](#)

hdb_toc, [8](#)

hdbkill, [4](#)

hinterpolate, [8](#)

hpack_manual, [9](#)

hpc_change, [9](#)

hpopulation, [9](#)