

Package: auscensus (via r-universe)

October 14, 2024

Title Access Australian Census Data (2006-2021)

Version 0.0.1.0011

Description R package to interact with Australian Census Data Packs, providing an interface to extract data across multiple censuses.

URL <https://carlosyanez.github.io/auscensus/>

License MIT + file LICENSE

BugReports <https://github.com/carlosyanez/auscensus/issues>

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports arrow, dplyr, fs, lifecycle, methods, naniar, piggyback, progressr, rlang, stats, stringr, tibble, tidyr, utils, zip

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Collate 'auscensus-package.R' 'cache_management.R' 'internal.R' 'get_content.R' 'helpers.R' 'lists.R' 'onLoad.R'

Config/testthat/edition 3

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

RemoteUrl <https://github.com/carlosyanez/auscensus>

RemoteRef HEAD

RemoteSha 0d26e2b79b23b82713b6f84844e8caaaf753137f

Contents

attribute_tibble_to_list	2
calculate_percentage	3
census_datapacks	4

data_census_delete	4
data_census_download	5
data_census_import	5
data_census_info	6
find_census_cache	6
get_census_data	6
get_census_summary	7
list_census_attributes	9
list_census_geo	10
list_census_geo_tables	10
list_census_geo_types	11
list_census_tables	12
list_census_years	12
remove_census_cache_csv	13

Index	14
--------------	-----------

attribute_tibble_to_list

Helper function to convert attributes to list

Description

Little helper function that converts tibble into a list with vectors, which is the expected attributes input for `get_census_summary()`

Usage

```
attribute_tibble_to_list(df, original = colnames(df)[1], new = colnames(df)[2])
```

Arguments

<code>df</code>	tibble/data.frame. First column is the original value, the second the new label
<code>original</code>	name of original attribute
<code>new</code>	new naming

Value

list object

Examples

```
## Not run:
attributes <- tribble(~Census_stat, ~ Group,
  "Age_years_60_males", "60 year old male",
  "Age (Years): 60_males", "60 year old male",
  "Age_years_60_males", "60 year old female",
  "Age (Years): 60_females", "60 year old female")
```

```
attribute_tibble_to_list(attributes)

## End(Not run)
```

calculate_percentage *Helper function to convert attributes to list*

Description

Little helper function that converts tibble into a list with vectors, which is the expected attributes input for `get_census_summary()`

Usage

```
calculate_percentage(  
  df,  
  key_col,  
  value_col,  
  key_value = "Total",  
  percentage_scale = 1  
)
```

Arguments

df	data frame
key_col	name of the column containing the "Total Label"
value_col	name of the column containing values
key_value	total label
percentage_scale	1 if percentage to be presented in scale 0-1, or 100 to be shown as 0%-100%

Value

list object

census_datapacks *Helper function to download data*

Description

Helper function to download data

Usage

```
census_datapacks()
```

Value

nothing

data_census_delete *Helper function to update/download data*

Description

Helper function to update/download data

Usage

```
data_census_delete(file = NULL)
```

Arguments

file to delete - defaults to all of them (provide full path, can obtain from data_census_info)

Value

nothing

data_census_download *Helper function to download data*

Description

Helper function to download data

Usage

```
data_census_download(  
  download_dir,  
  census_year = NULL,  
  download_method = "wget"  
)
```

Arguments

download_dir Full path where to download census files (required)
census_year census year to download (default to all)
download_method method to pass to download.file() ("wget" as default)

Value

nothing

data_census_import *Helper function to update/download data*

Description

Helper function to update/download data

Usage

```
data_census_import(file)
```

Arguments

file file to import to the cache

Value

nothing

data_census_info	<i>Helper function to update/download data</i>
------------------	--

Description

Helper function to update/download data

Usage

```
data_census_info()
```

Value

nothing

find_census_cache	<i>Helper function to find cache folder</i>
-------------------	---

Description

Helper function to find cache folder

Usage

```
find_census_cache()
```

Value

nothing

get_census_data	<i>Get census data.</i>
-----------------	-------------------------

Description

This function extracts table files from each data pack (given tables and geo structure), and will collate them together into a list(), which it will return. By default it will save the processed tables in the cache folder (in parquet files), which it will use on subsequent calls.

Usage

```
get_census_data(
  census_table,
  geo_structure,
  selected_years = list_census_years(),
  ignore_cache = FALSE,
  collect_data = FALSE,
  attr = NULL
)
```

Arguments

census_table	list of tables, in the format of the output of list_census_tables()
geo_structure	vector with strings of geo structures (e.g. SA1,LGA,CED)
selected_years	years to filter
ignore_cache	If TRUE, it will ignore cached files
collect_data	if TRUE will return data. if FALSE (default) , it will return arrow bindings to cached files
attr	attributes to filter on, presented as a character vector (e.g c("Age_years_60_males","Age_years_60_females"))

Value

data frame with data from file, filtered by division and election year

Examples

```
## Not run:
data <- get_census_data(census_table = list_census_tables("04"),
                       geo_structure = "LGA")

names(data)

## End(Not run)
```

get_census_summary *Get a summary of one or a collection of statistics across Censuses*

Description

This function allows to produce a summary of one or many statistics across censuses. Results are presented in a simple summary table. The function allows to present individual statistics or an aggregation of several statistics (e.g. aggregate number of births by country to present a continental total). If the name statistic containing totals is provided, the function has an option to calculate percentages (presented either in 0-1 or 0-100 scale).

Usage

```

get_census_summary(
  table_number = NULL,
  geo_structure = NULL,
  attribute,
  geo_unit_names = NULL,
  geo_unit_codes = NULL,
  selected_years = list_census_years(),
  reference_total = NULL,
  percentage_scale = 1,
  ignore_cache = FALSE,
  data_source = NULL,
  data_collected = FALSE,
  census_table = NULL
)

```

Arguments

<code>table_number</code>	number of selected table
<code>geo_structure</code>	character presenting the geographical structure to present stats (e.g. SA1,LGA,CED)
<code>attribute</code>	list with vectors of statistics to be summarise. Each vector element will be aggregated and presented under the item's name, e.g. <code>list("60 year old male"=c("Age_years_60_males", "Age (Years): 60_males"))</code> ,
<code>geo_unit_names</code>	vector with names of the geographic structures to present. They need to correspond with <code>geo_structure</code> , e.g. if <code>geo_structure="LGA"</code> , acceptable values could be <code>c("Melbourne", "Stonnington", "Yarra")</code> . If both this and <code>geo_unit_codes</code> are null, it will present all available elements.
<code>geo_unit_codes</code>	vector with ABS codes of the geo structures to present. Similar to <code>geo_units_names</code> .
<code>selected_years</code>	vector with selected years to display.
<code>reference_total</code>	Optional. List containing the names of all statistics representing totals, e.g. <code>list("Total"=c("Total_persons"))</code>
<code>percentage_scale</code>	1 if percentage to be presented in scale 0-1, or 100 to be shown as 0%-100%
<code>ignore_cache</code>	If TRUE, it will ignore cached files
<code>data_source</code>	result of <code>get_census_data</code> (will ignore other parameters if this is provided)
<code>data_collected</code>	TRUE if <code>data_source</code> is a dataset, FALSE if is a DB, arrow binding
<code>census_table</code>	Instead of using a table number, this allows for a more complex filter table, e.g. containing different table numbers. Expected format matches the output of <code>list_census_table()</code> .

Value

data frame with data from file, filtered by division and election year

Examples

```
## Not run:
get_census_summary(table_number = "04",
  attribute = list("60 year old male"=c("Age_years_60_males", "Age (Years): 60_males"),
                "60 year old female"=c("Age_years_60_males", "Age (Years): 60_females")),
  geo_unit_names = c("Melbourne", "Stonnington", "Yarra"),
  reference_total = list("Total"=c("Total_persons")),

## End(Not run)
```

list_census_attributes

Get names of attributes for a given census tables, across all time

Description

Get list of available geographies, filterable by type and name.

Usage

```
list_census_attributes(number = NULL, attribute_regex = NULL)
```

Arguments

number vector containing one or more table numbers
attribute_regex string with a regular expression to filter attribute names

Value

tibble, showing the geo type, available for each year

Examples

```
## Not run:
# Get list of all divisions
list_census_attributes()

## End(Not run)
```

list_census_geo	<i>Get census geographies, filterable</i>
-----------------	---

Description

Get list of available geographies, filterable by type and name.

Usage

```
list_census_geo(geo_types = NULL, geo_names = NULL, geo_name_regex = NULL)
```

Arguments

geo_types	vector containing one or more geography types (i.e. "STE","CED","SA1"). NULL by default.
geo_names	vector containing one or more geography names (i.e. "Melbourne", "Yarra","Stonnington" for LGAs). NULL by default.
geo_name_regex	string with a regular expression to filter geograhpy names (i.e. for all elements starting with M : "\$M")

Value

tibble, showing the geo type, available for each year

Examples

```
## Not run:
# Get list of all Commonwealth electoral divisions and Local Government Areas that start with "Mel"
list_census_geo(geo_types=c("CED", "LGA"), geo_name_regex="^Mel")

## End(Not run)
```

list_census_geo_tables	<i>List if a geo structure is available for a particular table , in a particular year</i>
------------------------	---

Description

List if a geo structure is available for a particular table , in a particular year

Usage

```
list_census_geo_tables(year = NULL, geo = NULL, table_number = NULL)
```

Arguments

year vector with years
geo vector with geo structure
table_number table number

Value

tibble

Examples

```
## Not run:  
# Get list of all divisions  
list_census_geo()  
  
## End(Not run)
```

list_census_geo_types *Get Geography types.*

Description

Very simple function listing geography types (e.g. SAx, CED, etc.), for which data pack has been imported.

Usage

```
list_census_geo_types()
```

Value

tibble, showing the geotype, available for each year

Examples

```
## Not run:  
# Get list of all divisions  
list_census_geo_types()  
  
## End(Not run)
```

list_census_tables *Get census geographies, filterable*

Description

Get list of available geographies, filterable by type and name.

Usage

```
list_census_tables(number = NULL, table_name_regex = NULL)
```

Arguments

number vector containing one or more table numbers
table_name_regex string with a regular expression to filter table names (i.e. for all elements containing with Country : "Country")

Value

tibble, showing the geo type, available for each year

Examples

```
## Not run:  
# Get list of all divisions  
list_census_geo()  
  
## End(Not run)
```

list_census_years *Get Census years.*

Description

Very simple function listing the Census years included in this package, for which data pack has been imported.

Usage

```
list_census_years(mode = "available")
```

Arguments

mode Either "listed" or "available"

Value

vector with years

Examples

```
## Not run:  
# Get list of all divisions  
list_census_years()  
  
## End(Not run)
```

remove_census_cache_csv

Helper function to delete all csv in cache

Description

Helper function to delete all csv in cache

Usage

```
remove_census_cache_csv()
```

Value

nothing

Index

* **getdata**

get_census_data, [6](#)
get_census_summary, [7](#)

* **helpers**

attribute_tibble_to_list, [2](#)
calculate_percentage, [3](#)
census_datapacks, [4](#)
data_census_delete, [4](#)
data_census_download, [5](#)
data_census_import, [5](#)
data_census_info, [6](#)
find_census_cache, [6](#)
remove_census_cache_csv, [13](#)

* **lists**

list_census_attributes, [9](#)
list_census_geo, [10](#)
list_census_geo_tables, [10](#)
list_census_geo_types, [11](#)
list_census_tables, [12](#)
list_census_years, [12](#)

[attribute_tibble_to_list, 2](#)

[calculate_percentage, 3](#)

[census_datapacks, 4](#)

[data_census_delete, 4](#)

[data_census_download, 5](#)

[data_census_import, 5](#)

[data_census_info, 6](#)

[find_census_cache, 6](#)

[get_census_data, 6](#)

[get_census_summary, 7](#)

[list_census_attributes, 9](#)

[list_census_geo, 10](#)

[list_census_geo_tables, 10](#)

[list_census_geo_types, 11](#)

[list_census_tables, 12](#)

[list_census_years, 12](#)

[remove_census_cache_csv, 13](#)