Package 'geosocial'

June 24, 2023

Type Package

Title IRISS WP3 GeoSocial solution - Toolbox

Version 1.0

Author Australian Urban Research Infrastructure Network (AURIN)

Maintainer Pascal Perez <pascal.perez@unimelb.edu.au>,German Gonzalez <german.gonzalez@unimelb.edu.au>, Masoud Rahimi <masoud.rahimi@unimelb.edu.au>

Description

License GLP-3

Encoding UTF-8

LazyData true

Imports haven, dplyr, sjlabelled, openxlsx, rjson, dataverse, tidyverse

Depends haven,
dplyr,
sjlabelled,
openxlsx,rjson,dataverse,tidyverse,
R (>= 2.10)

RoxygenNote 7.2.3

R topics documented:

`checkLSAY` *checkLSAY*

Description

Check that the dataset is an LSAY dataset and the dataset does not have modifications affecting the linkage process.

Usage

```
checkLSAY(dataset, cohort)
```

Arguments

`dataset`	The dataset that would be analysed.
`cohort`	year of the cohort (For example, LSAY 2009, cohort = 2009).

Value

True if the dataset does not have modifications that affect the linkage process.

`checkNamesDuplicates` *checkNamesDuplicates*

Description

It's important to check for duplicate names to avoid errors during data linkage and ensure accurate results. Stata does not allow duplicate variable names, so this process ensures the joined datasets don't have any variables with the same name.

Usage

```
checkNamesDuplicates(dataset)
```

Arguments

`dataset`	The dataset that would be analysed.

Value

True if none of the variable names are duplicates, otherwise false if overlap exists.

`checkPostcodeStructure` *checkPostcodeStructure*

Description

Checks that the postcodes are valid values.

Usage

```
checkPostcodeStructure(dataset)
```

Arguments

`dataset`	The dataset that would be analysed.

Value

True if all the postcodes are valid.

`checkVariableNames` *checkVariableNames*

Description

Check that the variable name is accepted by Stata: Stata variable names must adhere to the following rules: • Contain 1-32 characters. • Only contain the characters A-Z, 0-9, and underscore (_). • Begin with a letter or an underscore.

Usage

```
checkVariableNames(dataset)
```

Arguments

`dataset`	The dataset that would be analysed

Value

if the variable names are valid. Prints a message describing problem and specific variable that is the problem if invalid.

`concordances` *Concordances ABS data The ABS has developed a suite of geographical correspondences, primarily to assist users make comparisons and maintain time series between different editions of the Australian Statistical Geography Standard (ASGS). Correspondences are a mathematical method of reassigning data from one geographic region to another geographic region.*

Description

This file combines the concordances

Usage

```
concordances
```

Format

'concordances' A data frame with 8149 rows and 10 columns:

origin_unit Geographical unit - Origin

destination_unit Geographical unit - Destination

year_in Year of Geographical unit - Origin

year_out Year of Geographical unit - Destination

origin Geographical code - Origin

destination Geographical code - Destination

ratio Year

origin_areasqkm Area in square kilometres - Origin

destination_areasqkm Area in square kilometres - Destination

Source

<https://www.abs.gov.au/statistics/standards/australian-statistical-geography-standard-asgs-edition-3/jul2021-jun2026/access-and-downloads/correspondences>

`CreateFolders` *CreateFolders*

Description

This function generates the folders the script needs to run.

Usage

```
CreateFolders(path = getwd())
```

Arguments

name	the path

Value

True if the folders are created correctly.

---

downloadDataverseData *downloadDataverseData*

Description

Function takes inputs doi (get from dataverse listing of desired data) and the year of the data

Usage

```
downloadDataverseData(id, file)
```

Arguments

id	unique doi (get from dataverse listing of desired data)
file	where the file would be storage

Value

True if the download is correct

FilterConcordance *FilterConcordance*

Description

Narrow down the concordances and select the highest ratio that meets the threshold set by the user.

Usage

```
FilterConcordance(concordances, id, ratio_threshold = NULL)
```

Arguments

concordances	Concorcondaces file.
id	Geospatial identificator in que concordances table.
ratio_threshold	ratio.

Value

True if the file is storage correctly.

GenerateLog *GenerateLog*

Description

This function generates the log file that would store all the events generated by the code.

Usage

```
GenerateLog(name)
```

Arguments

name	the name of the log file

Value

True if the log file is generated

GetTerm *GetTerm*

Description

Based on a term, get the metadata associated to it. For example: GetTerm(keywords='http://example.com/census/IFAGEI

Usage

```
GetTerm(term)
```

Arguments

term	A String with the direction of the term in the ARDC server.

Value

Return label, dcterms_created, dcterms_modified, creator, dc_publisher, dc_source, dcterms_title, has_top_concept, history_note, scope_note, type and identifier.

LoadLSAY *LoadLSAY*

Description

Load or download the LSAY data, given a cohort and wave. (The demonstrator only supports LSAY cohort 2009)

Usage

```
LoadLSAY(cohort, wave, LSAY_topics)
```

Arguments

cohort	year of the cohort (For example, LSAY 2009, cohort = 2009)
wave	vector of years. (For example, wave = [2012,2013,2014])
LSAY_topics	vector with the sub-topics that would be included in the analysis, (For example["School transition" "Current",..]).

Value

a list which contains the survey data and the geospatial data.

LoadParameters *LoadParameters*

Description

Loads the parameters and sets the global environment.

Usage

```
LoadParameters(file)
```

Arguments

file	is the path where the JSON file is located.

Value

True if the parameters are valid

LoadTSP2021 *LoadTSP2021*

Description

Load and filter the TSP 2021 by the demonstrator variables and the closest year published census. For example, if the year is [2017,2018,2019], the TSP2021 only will have the closet census to this years that is 2016.

Usage

```
LoadTSP2021(year = NULL, variables = NULL)
```

Arguments

year	Vector of years.

Value

a list which contains the data and metadata

LSAY_metadata *Metadata - Longitudinal Surveys of Australian Youth*

Description

A complete listing of the variables and their associated formats and value labels contained in the LSAY data files for all six LSAY cohorts.

Usage

```
LSAY_metadata
```

Format

'LSAY_metadata' A data frame with 37,895 rows and 15 columns:

Cohort Y95,Y98,Y03,Y06,Y09,Y15

Wave

Wave/year

Section

Data element ID

Major topic area

Sub-major topic area

Minor topic area

Data element

Variable

Type

Label

Question

Base

Source

<https://www.lsay.edu.au/publications/search-for-lsay-publications/2621>

`PotentialCensus` *PotentialCensus*

Description

Given a year, find the closest census year.

Usage

```
PotentialCensus(year)
```

Arguments

`year`	year of analysis.

Value

String which contains the ABS_year and the original year.

`QualityIndicator` *QualityIndicator*

Description

Implementation of the ABS quality indicator which provides a considered measure of the quality of the correspondence. in relation to the weighting unit. (ABS, 2021).

Usage

```
QualityIndicator(ratio)
```

Arguments

`ratio`	This field describes the Ratio of the FROM region that is being donated to the TO region. The Ratio is a figure between 0 and 1. (ABS, 2021).

Value

Returns the equivalent SA3 for each row.

SearchConcept *SearchConcept*

Description

Based on a keyword, query the ARDC vocabs server and return related terms. For example: SearchConcept(keywords='AGED').

Usage

```
SearchConcept(keywords)
```

Arguments

keywords	A vector of strings that contains keywords.

Value

Terms related to the keywords.

SummaryConcordances *SummaryConcordances*

Description

Create a summary of the concordances involved in the data linkage.

Usage

```
SummaryConcordances(concordances_POA, concordances_SA3)
```

Arguments

concordances_POA	concordances using in the POA linkage
concordances_SA3	concordances using in the SA3 linkage

Value

a data frame with the concordances involved in the data linkage.

SummaryLog *SummaryLog*

Description

Consolidate a log with all the information on the data linkage.

Usage

```
SummaryLog(summaryResults)
```

Arguments

summaryResults	List which contains the output after the data linkage.

SummaryMetrics *SummaryMetrics*

Description

Create a summary with important information about the data linkage, such as year, cohort, linkage, missing values, not linked areas, not linked individuals and the quality of the data linkage.

Usage

```
SummaryMetrics(metrics_POA, metrics_SA3)
```

Arguments

metrics_POA	Metrics created in the POA linkage.
metrics_SA3	Metrics created in the SA3 linkage.

Value

a data frame with all the information mentioned before.

SummaryReport *SummaryReport*

Description

Create the summary report

Usage

```
SummaryReport(concordances_POA, concordances_SA3, metrics_POA, metrics_SA3)
```

Arguments

concordances_POA	Concordances used during the first stage: POA to SA3.
concordances_SA3	Concordances used during the second stage: SA3 to SA3.
metrics_POA	Metrics used during the first stage: POA to SA3.
metrics_SA3	Metrics used during the second stage: SA3 to SA3.

Value

Returns a list with all the information about the data linkage.

TestDataverseConnection *TestDataverseConnection*

Description

Tests connection to ADA dataverse. Requires dataverse token to be loaded in the system environment.

Usage

```
TestDataverseConnection()
```

Value

True if the connection is correct

`TransformPOA` *TransformPOA*

Description

Recieve a vector of POAS, This function transformate POA to SA3.

Usage

```
TransformPOA(data, concordances, year, ratio_threshold = NULL)
```

Arguments

`year`	Year.

Value

Returns the equivalent SA3 for each row.

the metric of missing values, miss matching and POAS that doenst match.

`TransformSA3` *TransformSA3*

Description

Recieve a vector of SA3 ABS year, This function transformate POA to SA3.

Usage

```
TransformSA3(data, concordances, year_in, year_out, ratio_threshold)
```

Arguments

`data`	Year.
`concordances`	Year.

Value

Returns the equivalent SA3 for each row.

the metric of missing values, miss matching and POAS that doenst match.

TSP2021 *Time series profile SA3*

Description

The Time series profile contains the Census characteristics of persons, families and dwellings over time. The data is based on place of usual residence.

Usage

```
TSP2021
```

Format

'TSP2021' A data frame with 358 Statistical Area Level 3 (SA3) and 35 Sociodemographic characteristics.

Details

The 2021 Time series profile contains data from the 2011, 2016, and 2021 Censuses. Where classifications have been revised, output are based on the classification used for the 2021 Census.

When interpreting the results from different time periods, take care as censuses are based on a point in time. Changes to the Census form design, collection procedures and processing may impact the comparability of data.

Source

<https://www.abs.gov.au/census/guide-census-data/about-census-tools/datapacks#:~:text=The

WriteStata *WriteStata*

Description

Write the outcome of the data linkage.

Usage

```
WriteStata(DataJoined, SurveyResponses, waves, path)
```

Arguments

DataJoined	List which contains the output after the data linkage.
waves	vector of years. (For example, wave = [2012,2013,2014])
path	Location where the files are going to be written.
SurveyResponses:	Dataframe which contains the survey responses.

Value

True if the folders are created correctly.

Index