

Package ‘googlePublicData’

July 2, 2014

Type Package

Title An R library to build Google's Public Data Explorer DSPL Metadata files

Version 0.12.05

Date 2012-05-15

Author George Vega Yon

Maintainer George Vega Yon <george.vega@nodoschile.org>

Description package provides a collection of functions to set up Google Public Data Explorer data visualization tool with your own data, building automatically the corresponding DSPL (XML) metadata file jointly with the CSV files. All zipped up and ready to be published at Public Data Explorer.

Depends R (>= 2.11.0), XML, utils, XLConnect

License GPL (>= 3)

URL <http://code.google.com/p/rdspl/>

LazyLoad yes

Repository CRAN

Date/Publication 2012-05-17 12:55:21

NeedsCompilation no

R topics documented:

checkTimeFormat	2
countries	3
country_slice	3
dspl	4
dspl Methods	6
genders	7

gender_country_slice	8
genMoreInfo	8
googlePublicData	10
states	11
state_slice	11

Index	12
--------------	-----------

checkTimeFormat	<i>DSPL time format verification</i>
-----------------	--------------------------------------

Description

Checks if a string fulfills the joda-times class specifications supported by DSPL language.

Usage

```
checkTimeFormat(fmt)
```

Arguments

fmt	String representing a time format to be verified.
-----	---

Details

Public Data Explorer currently supports daily, monthly and yearly distributed data. Joda-time, the corresponding time format on which DSPL times is based, allows declaring time formats using small case "d" (for days), capitalized "M" (for months) and small case "y" for years. Some examples:

Format Specification	Data Example
"yyyy"	1988
"yyyy-MM"	1988-03
"yyyy-MMM"	1988-Mar
"dd-MM-yyyy"	02-03-1988

Value

Logical. TRUE if the string passes the test.

Author(s)

George Vega Yon <george.vega@nodoschile.org>

References

- Google Public Data Explorer DSPL time definition: <https://developers.google.com/public-data/docs/canonical/time?hl=es>
- Google Public Data Explorer Cookbook for time definitions: https://developers.google.com/public-data/docs/cookbook#time_recipes

- Joda Time 2.1 API: <http://joda-time.sourceforge.net/api-release/org/joda/time/format/DateTimeFormat.html>

See Also

See also [dspl](#)

Examples

```
checkTimeFormat("yyyy-MM") # TRUE
checkTimeFormat("MMMyyyy") # TRUE
checkTimeFormat("mmmyyyy") # FALSE
```

countries	<i>World countries example data set</i>
-----------	---

Description

This data set is one used in the DSPL Tutorial. Specifically, it contains the basic columns used to define geographical dimensions, in this case, countries.

Usage

```
data(countries)
```

Format

A data frame containing 5 observations.

Source

DSPL Google Code Page Downloads: <https://code.google.com/p/dspl/downloads/detail?name=tutorial1.0.zip&can=2&q=>

country_slice	<i>Some countries statistics</i>
---------------	----------------------------------

Description

This data set is one used in the DSPL Tutorial. Specifically, it contains the population magnitudes at country level since 1960 to 1963.

Usage

```
data(country_slice)
```

Format

A data frame containing 13 observations.

Source

DSPL Google Code Page Downloads: <https://code.google.com/p/dspl/downloads/detail?name=tutorial1.0.zip&can=2&q=>

dspl	<i>Builds DSPL metadata file</i>
------	----------------------------------

Description

Parsing *csv*, *tab* or *xls(x)* files at a specific directory path, dspl generates a complete DSPL file. If an output string is specified, the function generates the complete ZIP (DSPL file plus csv files) ready to be uploaded to Google Public Data Explorer.

Usage

```
dspl(path, output = NA, replace = F, targetNamespace = "", timeFormat = "yyyy",
      lang = c("es", "en"), name = NA, description = NA, url = NA, providerName = NA,
      providerURL = NA, sep = ";", dec=".", encoding = getOption("encoding"),
      moreinfo = NA)
```

Arguments

path	String. Path to the folder where the tables (csv/tab/xls) are at.
output	String, optional. Path to the output ZIP file.
replace	Logical. If output ZIP file is defined exists, dspl replaces it.
targetNamespace	String. As DSPL documentation states “Provides a URI that identifies your dataset. This URI is not required to point to an actual resource, but it’s a good idea to have the URI resolve to a document describing your content or dataset”.
timeFormat	String. The corresponding time format of the collection. Should be specified accordingly to joda-time format. See the Details section for more information.
lang	A list of strings of the languages supported by the dataset. Could be only one.
name	List of strings. The name of the dataset as defined accordingly to the lang list.
description	List of strings. Description of the dataset. It also supports multiple description as the name
url	The corresponding URL for the dataset.
providerName	List of strings. The data provider name.
providerURL	List of strings. The data provider website url.

sep	The separation character of the tables in the 'path' folder. Currently supports introducing the following arguments: “,” or “;” (for .csv files), “\t” (for .tab files) and “xls” or “xlsx” (for microsofts excel files); the last one using XLConnect library.
dec	String. Decimal point.
encoding	The char encoding of the input tables. Currently ignored for microsoft excel files.
moreinfo	A special tab file generated by the function genMoreInfo that contains a dataframe of the dataset concepts with more specifications such as description, topic, url, etc.

Details

If there isn't any output defined the function returns a list of class `dspl` that among its contents has a xml object (DSPL file); otherwise, if an output is defined, the results consists on two things, an already ZIP file containing a all the necessary to be uploaded at publicdata.google.com (a collection of csv files and the XML DSPL written file) and a message (character object).

Internally, the parsing process consists on the following steps: (1) Loading the data, (2) generating each column corresponding id, (3) Identifying the data types, (4) building concepts, (5) Identifying dimensional concepts and distinguishing between categorical, geographical and time dimentions, and finally (6) executing internal checks.

In order to properly load the zip file (DSPL file plus CSV data files), the function executes a series of internal checks upon the data structure. The detailed list:

- Slices with the same dimentions: DSPL requires that each slice represents one dimensional cut, this is, there shouldn't be more than one data table with the same dimensions.
- Duplicated concetps: As a result of multiple data types, e.g a single concept (statistic) as integer in one table and float in other, `dspl` may get confused, so during the parsing process, if there is a chance, it collapses duplicated concepts into only one concept and assigns it the common data type (float).
- Correct time format definition: Using [checkTimeFormat](#) ensures that the time format specified is compatible with DSPL.

Value

If there isn't any output defined, `dspl` returns list of `class "dspl"`.

An object of class `"dspl"` is a list containing:

<code>dspl</code>	A character string containing the DSPL XML document as defined by the saveXML function.
<code>concepts.by.table</code>	A data frame object of concepts stored by table.
<code>dimtabs</code>	A data frame containing dimensional tables.
<code>slices</code>	A data frame of slices.
<code>concepts</code>	A data frame of concepts (all of them).
<code>dimentions</code>	A data frame of dimensional concepts.

statistics A matrix of statistics.

otherwise the function will build a ZIP file as specified in the output containing the CSV and DSPL (XML) files.

Author(s)

George Vega Yon <george.vega@nodoschile.org>

References

- Google Public Data Explorer Tutorial: <https://developers.google.com/public-data/docs/tutorial>

Examples

```
## Not run:  
demo(dspl)  
  
## End(Not run)
```

dspl Methods

Print and summarize dspl objects

Description

Methods to print and summarize dspl class objects

Usage

```
## S3 method for class 'dspl'  
print(x, path=NULL, replace=F, ...)  
  
## S3 method for class 'dspl'  
summary(object, ...)
```

Arguments

x	An object of class dspl to be printed.
object	An object of class dspl to be summarized.
path	String. Output path where to save the XML DSPL file.
replace	Logical. If path exists, TRUE would replace the file.
...	arguments passed on to <code>cat</code> (<code>print.dspl</code>)

Value

print.dspl None (invisible NULL).
summary.dspl Returns the class attributes and a list containing as defined by [dspl](#) function.
For more information see its value section.

Author(s)

George Vega Yon <george.vega@nodoschile.org>

See Also

See also [dspl](#)

Examples

```
## Not run:  
# Parsing some xlsx files at "my stats folder"  
myspl <- dspl(path="my stats folder/", ext="xls")  
  
# Checking the summary of the data bundle  
summary(myspl)  
  
# Writing the DSPL XML definition into a file  
print(myspl, path="my stats folder/myspl.xml")  
  
## End(Not run)
```

genders

Genders example data set

Description

This data set is one used in the DSPL Tutorial. Specifically, it contains the basic columns used to define a categorical dimention such as gender.

Usage

```
data(genders)
```

Format

A data frame containing 2 observations.

Source

DSPL Google Code Page Downloads: <https://code.google.com/p/dspl/downloads/detail?name=tutorial1.0.zip&can=2&q=>

gender_country_slice *Some Countries statistics at Gender level*

Description

This data set is one used in the DSPL Tutorial. Specifically, it contains the population magnitudes at country and gender level since 1960 to 1961.

Usage

```
data(gender_country_slice)
```

Format

A data frame containing 13 observations.

Source

DSPL Google Code Page Downloads: <https://code.google.com/p/dspl/downloads/detail?name=tutorial1.0.zip&can=2&q=>

genMoreInfo *Generates a dataframe used to complement a DSPL bundle*

Description

Parsing *csv*, *tab* or *xls(x)* files at a specific directory path, genMore info generates a dataframe used to complete a DSPL bundle with a more complete concepts definition including description, url, etc..

Usage

```
genMoreInfo(path, encoding=getOption("encoding"), sep=";", output=NA,
             action="merge", dec=".")
```

Arguments

path	String. Path to the folder where the tables are saved.
encoding	The encoding of the files to be parsed.
sep	The separation character of the tables in the 'path' folder. Currently supports introducing the following arguments: “;” or “;” (for .csv files), “\t” (for .tab files) and “xls” or “xlsx” (for microsofts excel files); the last one using XLConnect library.
output	If defined, the place where to save the dataframe as tab file. Otherwise it returns a data frame object.

action	Tells the function what to do if there's a copy of the file. Available actions are "merge" and "replace".
dec	String. Decimal point.

Details

If there isn't any output defined (NA) the function returns a dataframe containing concepts as observations. Using this, the user may add more describing info about concepts. In turn it writes a tab file with the dataframe described above. The user may recycle this file writing "append" in the action argument.

Value

If no output defined, genMoreInfo returns a dataframe with the following columns.

id	XML id of the concept (autogenerated)
label	The label of the concept (autogenerated)
description	A brief description of the concept
topic	The topic of the concept
url	A URL for the concept where, for example, to get more info
totalName	A total name as specified by DSPL language (works for dimensional concepts)
pluralName	A total name as specified by DSPL language (works for dimensional concepts)

Author(s)

George Vega Yon <george.vega@nodoschile.org>

References

Google Public Data Explorer: <http://publicdata.google.com>

Examples

```
## Not run:
# Parsing some xlsx files at "my stats folder" to gen a "moreinfo" dataframe
INFO <- genMoreInfo(path="my stats folder/", sep="xls")

# Rows 1 to 10 are about "Poverty" and rows 11 to 20 about "Education"
# So we fill the "topic" column with it.
INFO$topic[1:10] <- "Poverty"
INFO$topic[11:20] <- "Education"

# Finally, we build the DSPL ZIP including more info
dspl(path="my stats folder/", sep="xls", moreinfo=INFO)

## End(Not run)
```

`googlePublicData`*An R library to build Google's Public Data Explorer DSPL Metadata*

Description

`googlePublicData` package provides a collection of functions to set up Google Public Data Explorer data visualization tool with your own data, building automatically the corresponding DSPL (XML) metadata file jointly with the CSV files. All zipped up and ready to be published at Public Data Explorer.

Also includes several data structure verifiers in order to avoid surprises while loading your ZIP file to Public Data Explorer page.

Please visit the project home for more information and examples: <https://code.google.com/p/rdspl>.

Details

Package: `googlePublicData`
Type: Package
Version: 0.12.05
Date: 2012-05-15
License: GPL version 2 or later

Author(s)

George Vega Yon <george.vega@nodoschile.org>

References

- `googlePublicData` project site: <https://code.google.com/p/rdspl>
- Public Data Explorer site: <http://publicdata.google.com/>
- Public Data Explorer Developers site: <https://developers.google.com/public-data/>
- DSPL Code site: <http://code.google.com/p/dspl/>
- `googleVis` package: <http://code.google.com/p/google-motion-charts-with-r/>

Examples

```
## Not run:  
  demo(dspl)  
  
## End(Not run)
```

states	<i>US states example data set</i>
--------	-----------------------------------

Description

This data set is one used in the DSPL Tutorial. Specifically, it contains the basic columns used to define geographical dimensions, in this case, US States.

Usage

```
data(states)
```

Format

A data frame containing 8 observations.

Source

DSPL Google Code Page Downloads: <https://code.google.com/p/dspl/downloads/detail?name=tutorial1.0.zip&can=2&q=>

state_slice	<i>Some US States statistics</i>
-------------	----------------------------------

Description

This data set is one used in the DSPL Tutorial. Specifically, it contains the population magnitudes and unemployment rate at state level since 1960 to 1963.

Usage

```
data(state_slice)
```

Format

A data frame containing 9 observations.

Source

DSPL Google Code Page Downloads: <https://code.google.com/p/dspl/downloads/detail?name=tutorial1.0.zip&can=2&q=>

Index

*Topic **IO**

dspl, 4
genMoreInfo, 8

*Topic **datasets**

countries, 3
country_slice, 3
gender_country_slice, 8
genders, 7
state_slice, 11
states, 11

*Topic **methods**

dspl Methods, 6

*Topic **package**

googlePublicData, 10

*Topic **utilities**

checkTimeFormat, 2

cat, 6

checkTimeFormat, 2, 5

class, 5

countries, 3

country_slice, 3

dspl, 3, 4, 7

dspl Methods, 6

gender_country_slice, 8

genders, 7

genMoreInfo, 5, 8

googlePublicData, 10

googlePublicData-package
(googlePublicData), 10

joda-times (checkTimeFormat), 2

print.dspl (dspl Methods), 6

saveXML, 5

state_slice, 11

states, 11

summary.dspl (dspl Methods), 6

timeFormat (checkTimeFormat), 2

XLConnect, 5, 8