Package 'oec'

November 1, 2017

Type Package

Title Observatory of Economic Complexity API Wrapper and Utility Program

Version 2.5

Date 2017-11-01

Author Cesar A. Hidalgo [aut],
Alexander Simoes [aut],
Mauricio Vargas S. [aut, cre],
Manuel Aristaran [ctb],
Mike Bostock [ctb] (D3),
Dave Landy [ctb] (D3Plus),
Pablo H. Paladino [ctb] (Version control),
Gabriela Perez P. [ctb] (Documentation),
UN Comtrade [dtc],
MIT Media Lab [dtc],
Datawheel [fnd, cph]

Maintainer Mauricio Vargas S. <oec@media.mit.edu>

URL http://atlas.media.mit.edu/en/

Description Access The Observatory of Economic Complexity API from R to download international trade data and create and D3Plus visualizations.

License MIT + file LICENSE

LazyData TRUE

Depends curl, dplyr, jsonlite, readr, servr, R (>= 2.10)

RoxygenNote 6.0.1

R topics documented:

`oec-package` *The Observatory of Economic Complexity*

Description

Package's details.

Details

This package was created to simplify user interaction with the OEC's API. It will download trade data from MIT Media Lab servers and it will save that both in CSV and JSON formats.

You can use this package just to download information but it also creates D3Plus visualizations that are suitable for presentations or a context where you need to show data. These visualizations do not need internet connection after you obtain the data.

All of the datasets provided within this package provide data that cannot be obtained from the API and do help creating better visualizations.

The functions provided within this package are:

install_d3plus Installs D3 and D3Plus.

demos Copies the demo file.

getdata Downloads and processes the data from the API for a certain year.

getdata_interval Downloads and processes the data from the API for an interval of years.

network Creates a network for a given year.

network_interval Creates a network for an interval of years.

treemap Creates a treemap for a given year.

treemap_interval Creates a treemap for an interval of years.

The datasets provided within this package are:

countries_list A list of all the countries in the world and its respective country code.

hs92 HS92 products and groups (4 and 6 characters codes).

sitc SITC rev.2 products and groups (4 characters codes).

The additional files provided within this package are:

treemap_template.html Template to display a treemap of the imports, exports or trade balance of a country for a certain year using HS92 or SITC (rev.2) product classification.

network_template.html Template to display a network of exports of a country for a certain year using HS92 or SITC (rev.2) product classification.

nodes_hs92_4.json Part of a pre-drawn network to create network visualizations using HS92 product classification.

edges_hs92_4.json Part of a pre-drawn network to create network visualizations using HS92 product classification.

nodes_sitc_4.json Part of a pre-drawn network to create network visualizations using SITC (rev.2) product classification.

edges_sitc_4.json Part of a pre-drawn network to create network visualizations using SITC (rev.2) product classification.

d3plus-1.9.8.zip Contains D3Plus and D3 to display the visualization.

`countries_list` *Countries list*

Description

A reference to know the country codes to be able to download and import data (e.g. "chl" stands for "Chile" in the OEC's API)

Usage

```
countries_list
```

Format

A data frame with 263 observations on the following 2 variables.

`country` Official countries' names spelled according to the United Nations.

`country_code` Three characters codes to refer to the countries in the OEC's API.

Examples

```
countries_list
```

d3plus-1.9.8.zip — *D3Plus visualization library*

Description

D3Plus is an extension to the D3 library that allows fast and easy creation of data visualizations.

This zip will be decompressed in your working directory when you use `treemap`, `treemap_interval`, `network` or `network_interval` for the first time in your working directory. It contains both the javascript and the icons required to display the visualizations in the browser.

D3Plus was created by Alexander Simoes and Dave Landry and D3 was created by Mike Bostock.

demos — *Copies the demo file*

Description

Copies the demo file

Usage

```
demos()
```

Value

Copies a file named `demo_examples.R` to the working directory.

Examples

```
# demos()
```

edges_hs92_4.json — *A part of a pre-drawn network to create network visualizations using HS92 product classification.*

Description

The network visualization is a bit different from the rest of D3Plus in that it requires extra files besides the data and attribute lookups. It is important to note that D3Plus network visualizations code does not attempt to create a dynamic layout based on a nodes and edges list.

This package provides a curated network provided in two JSON files of (X,Y) coordinates divided between elements positioning (nodes) and elements connections (edges). The Product Space described in the OEC website is a network of around 800 nodes and roughly 2000 edges, and this file is a part of a precomputed layout of the network that allows the user to display this large network without pushing all of the resources on the client's machine.

edges_sitc_4.json *A part of a pre-drawn network to create network visualizations using SITC (rev.2) product classification.*

Description

The network visualization is a bit different from the rest of D3Plus in that it requires extra files besides the data and attribute lookups. It is important to note that D3Plus network visualizations code does not attempt to create a dynamic layout based on a nodes and edges list.

This package provides a curated network provided in two JSON files of (X,Y) coordinates divided between elements positioning (nodes) and elements connections (edges). The Product Space described in the OEC website is a network of around 800 nodes and roughly 2000 edges, and this file is a part of a precomputed layout of the network that allows the user to display this large network without pushing all of the resources on the client's machine.

getdata *Downloads and processes the data from the API*

Description

Downloads and processes the data from the API

Usage

```
getdata(origin, dest, year, classification)
```

Arguments

origin	Country code of origin (e.g. "chl" for Chile)
dest	Country code of destination (e.g. "chn" for China)
year	The OEC's API ranges from 1962 to 2016
classification	Trade classification that can be "1" (HS92 4 characters since year 1995), "2" (SITC rev.2 4 characters since year 1962) or "3" (HS92 6 characters since year 1995)

Examples

```
# Run countries_list() to display the full list of countries
# For the example Chile is "chl" and China is "chn"

# Download trade between Chile and China
# Year 2016 (HS92 4 characters)
# getdata("chl", "chn", 2016)
# getdata("chl", "chn", 2016, 1) # equivalent to last command
```

```
# Download trade between Chile and China
# Year 2016 (SITC rev2 4 characters)
# getdata("chl", "chn", 2016, 2)

# Download trade between Chile and China
# Year 2016 (HS92 6 characters)
# getdata("chl", "chn", 2016, 3)
```

`getdata_interval` *Downloads and processes the data from the API*

Description

Downloads and processes the data from the API

Usage

```
getdata_interval(origin, dest, initial_year, final_year, classification,
  interval)
```

Arguments

`origin`	Country code of origin (e.g. "chl" for Chile)
`dest`	Country code of destination (e.g. "chn" for China)
`initial_year`	The OEC's API ranges from 1942 to 2016. This needs to be lower than 'final_year'
`final_year`	The OEC's API ranges from 1942 to 2016. This needs to be greater than 'initial_year'
`classification`	Trade classification that can be "1" (HS92 4 characters since year 1995), "2" (SITC rev.2 4 characters since year 1962) or "3" (HS92 6 characters since year 1995)
`interval`	is an optional parameter to define the distance between years (by default set to 1)

Examples

```
# Run countries_list() to display the full list of countries
# For the example Chile is "chl" and China is "chn"
# Download trade between Chile and China
# Years 2010-2016 (HS92 4 characters)
# getdata_interval("chl", "chn", 2010, 2016)
# getdata_interval("chl", "chn", 2010, 2016, 1, 1) # equivalent to last command

# Download trade between Chile and China
# Years 2010, 2012 and 2014 from OEC's API (HS92 4 characters)
# getdata_interval("chl", "chn", 2010, 2014, 1, 2)
```

```
# Download trade between Chile and China
# Years 2010, 2012 and 2014 from OEC's API (SITC rev2 4 characters)
# getdata_interval("chl", "chn", 2010, 2014, 2, 2)

# Download trade between Chile and China
# Years 2010, 2012 and 2014 from OEC's API (HS92 6 characters)
# getdata_interval("chl", "chn", 2010, 2014, 3, 2)
```

`hs92` *HS92 products*

Description

This file is used to create the visualizations and match product codes to product names and groups.

Usage

```
hs92
```

Format

A data frame with 6282 observations on the following 5 variables.

`product_name` Contains the H292 products' names (e.g. horses, bovine, pigs, etc)

`group_name` Contains the H292 groups (e.g. animal products, vegetable products, etc)

`group_id` Contains the associated codes of every group (e.g. animal products is 01)

`hs92` Contains the associated codes of every product (e.g. horses is 010101)

`color` One colour per group used to create visualizations

Examples

```
hs92
```

`install_d3plus` *Installs D3 and D3Plus*

Description

Installs D3 and D3Plus

Usage

```
install_d3plus()
```

Value

Copies a folder named d3plus to the working directory and it contains the js files and icons to make the visualizations

Examples

```
# install_d3plus()
```

---

network *Creates a network of exports for a given year*

Description

Creates a network of exports for a given year

Usage

```
network(origin, dest, year, classification)
```

Arguments

origin	is the country code of origin (e.g. "chl" for Chile)
dest	is the country code of destination (e.g. "chn" for China)
year	is the year and the OEC's API ranges from 1962 to 2014
classification	Trade classification that can be "1" (HS92 4 characters since year 1995) or "2" (SITC rev.2 4 characters since year 1962)

Value

Creates an HTML file with a network visualization for a given year.

Examples

```
# Run countries_list() to display the full list of countries
# For the example Chile is "chl" and China is "chn"

# What are the export opportunities of Chile?
# Year 2015, trade with China (HS92 4 characters)
# network("chl", "chn", 2015)
# network("chl", "chn", 2015, 1) # equivalent to last command
```

`network_interval` *Creates a network of exports for a given period of years*

Description

Creates a network of exports for a given period of years

Usage

```
network_interval(origin, dest, initial_year, final_year, classification,
  interval)
```

Arguments

`origin`	is the country code of origin (e.g. "chl" for Chile)
`dest`	is the country code of destination (e.g. "chn" for China)
`initial_year`	is the initial year and the OEC's API ranges from 1942 to 2014
`final_year`	is the final year and the OEC's API ranges from 1942 to 2014
`classification`	Trade classification that can be "1" (HS92 4 characters since year 1995) or "2" (SITC rev.2 4 characters since year 1962)
`interval`	is an optional parameter to define the distance between years (by default set to 1) #' @examples # Run countries_list() to display the full list of countries # For the example Chile is "chl" and China is "chn" # What are the export opportunities of Chile? # Years 2010-2015, trade with China (HS92 4 characters) # network_interval("chl", "chn", 2010, 2015) # network_interval("chl", "chn", 2010, 2015, 1, 1) # equivalent to last command

Value

Creates an `HTML` file with a network visualization for a given given period of years.

`network_template.html` *A template to display a network of the exports of a country for a certain year using HS92 or SITC (rev.2) product classification.*

Description

Contains a formatted template with fields that `network` or `network_interval` functions will find and replace accordingly to the data you want to display and the final visualization will be saved in an HTML file to your working directory.

nodes_hs92_4.json *A part of a pre-drawn network to create network visualizations using HS92 product classification.*

Description

The network visualization is a bit different from the rest of D3Plus in that it requires extra files besides the data and attribute lookups. It is important to note that D3Plus network visualizations code does not attempt to create a dynamic layout based on a nodes and edges list.

This package provides a curated network provided in two JSON files of (X,Y) coordinates divided between elements positioning (nodes) and elements connections (edges). The Product Space described in the OEC website is a network of around 800 nodes and roughly 2000 edges, and this file is a part of a precomputed layout of the network that allows the user to display this large network without pushing all of the resources on the client's machine.

nodes_sitc_4.json *A part of a pre-drawn network to create network visualizations using SITC (rev.2) product classification.*

Description

The network visualization is a bit different from the rest of D3Plus in that it requires extra files besides the data and attribute lookups. It is important to note that D3Plus network visualizations code does not attempt to create a dynamic layout based on a nodes and edges list.

This package provides a curated network provided in two JSON files of (X,Y) coordinates divided between elements positioning (nodes) and elements connections (edges). The Product Space described in the OEC website is a network of around 800 nodes and roughly 2000 edges, and this file is a part of a precomputed layout of the network that allows the user to display this large network without pushing all of the resources on the client's machine.

sitc *SITC products*

Description

This file is used to create the visualizations and match product codes to product names and groups.

Usage

```
sitc
```

Format

A data frame with 988 observations on the following 5 variables.

`product_name` Contains the H292 products' names (e.g. initiating devices, polymerization ion exchangers, etc)

`group_name` Contains the H292 groups (e.g. machinery, electronics products, etc)

`group_id` Contains the associated codes of every group (e.g. animal products is 10)

`sitc` Contains the associated codes of every product (e.g. initiating devices is 5722)

`color` One colour per group used to create visualizations

Examples

```
sitc
```

`treemap` *Creates a treemap for a given year*

Description

Creates a treemap for a given year

Usage

```
treemap(origin, dest, variable, year, classification, depth)
```

Arguments

`origin`	is the country code of origin (e.g. "chl" for Chile)
`dest`	is the country code of destination (e.g. "chn" for China)
`variable`	is the variable to visualize and it can be "imports", "exports" or "exchange" (trade exchange)
`year`	is the year and the OEC's API ranges from 1962 to 2014
`classification`	Trade classification that can be "1" (HS92 4 charactersacters since year 1995), "2" (SITC rev.3 4 charactersacters since year 1962) or "3" (HS92 6 charactersacters since year 1995)
`depth`	is an optional parameter that can take values "0" (group's detail) or "1" (product's detail)

Value

Creates an `HTML` file with a treemap visualization for a given year.

Examples

```
# Run countries_list() to display the full list of countries
# For the example Chile is "chl" and China is "chn"

# What does Chile export to China?
# Year 2015 (HS92 4 characters)
# treemap("chl", "chn", "exports", 2015)
# treemap("chl", "chn", "exports", 2015, 1) # equivalent to last command
```

`treemap_interval` *Creates a treemap for a given period of years*

Description

Creates a treemap for a given period of years

Usage

```
treemap_interval(origin, dest, variable, initial_year, final_year,
  classification, interval, depth)
```

Arguments

`origin`	is the country code of origin (e.g. "chl" for Chile)
`dest`	is the country code of destination (e.g. "chn" for China)
`variable`	is the variable to visualize and it can be "imports", "exports" or "exchange" (trade exchange)
`initial_year`	is the initial year and the OEC's API ranges from 1942 to 2014
`final_year`	is the final year and the OEC's API ranges from 1942 to 2014
`classification`	Trade classification that can be "1" (HS92 4 characters since year 1995), "2" (SITC rev.3 4 characters since year 1962) or "3" (HS92 6 characters since year 1995)
`interval`	is an optional parameter to define the distance between years (by default set to 1)
`depth`	is an optional parameter that can take values "0" (group's detail) or "1" (product's detail), by defaults its set to 1

Value

Creates an `HTML` file with a treemap visualization for a given period of years.

Examples

```
# Run countries_list() to display the full list of countries
# For the example Chile is "chl" and China is "chn"

# What does Chile export to China?
# Years 2010-2015 (HS92 4 characters)
# treemap_interval("chl", "chn", "exports", 2010, 2015)
# treemap_interval("chl", "chn", "exports", 2010, 2015, 1, 1, 1) # equivalent to last command
```

`treemap_template.html` *A template to display a treemap of the imports, exports or trade balance of a country for a certain year using HS92 or SITC (rev.2) product classification.*

Description

Contains a formatted template with fields that `treemap` or `treemap_interval` functions will find and replace accordingly to the data you want to display and the final visualization will be saved in an HTML file to your working directory.

Index