Package ‘oec’

February 10, 2017

Type Package

Title Use the Observatory of Economic Complexity's API in R

Version 3.0

Date 2017-02-10

Author Cesar A. Hidalgo <hidalgo@media.mit.edu> [aut],
Alexander Simoes <alex@datawheel.us> [aut, cph],
Mauricio Vargas S. <mvargas@dcc.uchile.cl> [aut, cre, cph],
Manuel Aristaran [ctb],
Mike Bostock [ctb] (D3),
Dave Landy [ctb] (D3Plus)

Maintainer Mauricio Vargas S. <mvargas@dcc.uchile.cl>

URL http://atlas.media.mit.edu/en/, https://github.com/pachamaltese/oec/

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

License MIT + file LICENSE

LazyData TRUE

Depends curl,
data.table,
jsonlite,
plyr,
servr

RoxygenNote 5.0.1

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_2char` HS92 groups.

`hs92_4char` HS92 products (4 characters codes).

`hs92_6char` HS92 products (6 characters codes).

`hs92_colors` HS92 colors.

`sitc_rev2_2char` SITC (rev. 2) groups.

`sitc_rev2_4char` SITC (rev. 2) products (4 characters codes).

`sitc_rev2_colors` SITC (rev. 2) colors.

The additional files provided within this package are:

`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.

`treemap_interval_template.html` A template to display a treemap of imports, exports or trade balance of a country for an interval of years using HS92 or SITC (rev.2) product classification.

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

`network_interval_template.html` A template to display a network of exports of a country for an interval of years using HS92 or SITC (rev.2) product classification.

`nodes_hs92_4char.json` A part of a pre-drawn network to create network visualizations using HS92 product classification.

`edges_hs92_4char.json` A with a part of a pre-drawn network to create network visualizations using HS92 product classification.

`nodes_sitc_rev2_4char.json` A file with a part of a pre-drawn network to create network visualizations using SITC (rev.2) product classification.

`edges_sitc_rev2_4char.json` A file with a 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` *A list of all the countries*

Description

A list of all the countries in the world and its country code. You need the country code (e.g. chl) to obtain data for a country (e.g Chile)

Usage

```
countries_list
```

Format

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

`country` the full names of the countries

`country_code` the ids of the countries

Examples

```
# see the list of countries
# countries_list
```

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_4char.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_rev2_4char.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, destination, year, classification)
```

Arguments

`origin`	Country code of origin (e.g. "chl" for Chile)
`destination`	Country code of destination (e.g. "chn" for China)
`year`	The OEC's API ranges from 1962 to 2014
`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
# Chile is "chl" and China is "chn"

# Download trade data from OEC's API (HS92 4 characters product list)
# for Chile and China in the year 2014
# getdata("chl", "chn", 2014)
# is the same as
# getdata("chl", "chn", 2014, 1)

# Download trade data from OEC's API (SITC rev.2 4 characters product list)
# for Chile and China in the year 2014
# getdata("chl", "chn", 2014, 2)

# Download trade data from OEC's API (HS92 6 characters product list)
# for Chile and China in the year 2014
# getdata("chl", "chn", 2014, 3)
```

getdata_interval *Downloads and processes the data from the API*

Description

Downloads and processes the data from the API

Usage

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

Arguments

`origin` Country code of origin (e.g. "chl" for Chile)

`destination` Country code of destination (e.g. "chn" for China)

`initial_year` The OEC’s API ranges from 1942 to 2014. This needs to be lower than ‘final_year‘

`final_year` The OEC’s API ranges from 1942 to 2014. 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
# Chile is "chl" and China is "chn"

# Download trade data from OEC's API (HS92 4 characters product list)
# for Chile and China in the years 2010 to 2014
# getdata_interval("chl", "chn", 2011, 2014)
# is the same as
# getdata_interval("chl", "chn", 2011, 2014, 1, 1)

# Download trade data from OEC's API (HS92 4 characters product list)
# for Chile and China in the years 2010, 2012 and 2014
# getdata_interval("chl", "chn", 2011, 2014, 1, 2)

# Download trade data from OEC's API (SITC rev.2 4 characters product list)
# for Chile and China in the years 2010, 2012 and 2014
# getdata_interval("chl", "chn", 2011, 2014, 2, 2)

# Download trade data from OEC's API (HS92 6 characters product list)
# for Chile and China in the years 2010, 2012 and 2014
# getdata_interval("chl", "chn", 2011, 2014, 3, 2)
```

`hs92_2char` *HS92 groups*

Description

HS92 groups. This file is used to create the visualizations.

Usage

```
hs92_2char
```

Format

A data frame with 22 observations on the following 2 variables.

`hs92_group_name` Contains the H292 groups (e.g. animal products, vegetable products, etc)

`hs92_group_id` Contains the associated codes of every group (e.g. animal products is 01)

Examples

```
# see the group codes for HS92 (6 and 8 characters)
# hs92_2char
```

`hs92_4char` *HS92 products (4 characters)*

Description

HS92 products (4 characters). This file is used to create the visualizations.

Usage

```
hs92_4char
```

Format

A data frame with 1242 observations on the following 4 variables.

`hs92_product_name` Contains the H292 products' names (e.g. horses, bovine, pigs, etc)

`hs92_group_name` Contains the H292 groups (e.g. animal products, vegetable products, etc)

`hs92_product_id` Contains the associated codes of every product (e.g. horses is 0101)

`hs92_group_id` Contains the associated codes of every group (e.g. animal products is 01)

Examples

```
# see the group codes and product codes for HS92 (4 characters)
# hs92_4char
```

`hs92_6char` *HS92 products (6 characters)*

Description

HS92 products (6 characters). This file is used to create the visualizations.

Usage

```
hs92_6char
```

Format

A data frame with 5040 observations on the following 4 variables.

`hs92_product_name` Contains the H292 products' names (e.g. horses, bovine, pigs, etc)

`hs92_group_name` Contains the H292 groups (e.g. animal products, vegetable products, etc)

`hs92_product_id` Contains the associated codes of every product (e.g. horses is 010101)

`hs92_group_id` Contains the associated codes of every group (e.g. animal products is 01)

Examples

```
# see the group codes and product codes for HS92 (6 characters)
# hs92_6char
```

`hs92_colors` *HS92 colors*

Description

HS92 colors. This file is used to create the visualizations.

Usage

```
hs_colors
```

Format

A data frame with 21 observations on the following 2 variables.

`hs92_group_name` Contains the H292 groups (e.g. animal products, vegetable products, etc)

`hs92_color` Contains the associated colors of every group (e.g. mineral products is #330000)

Examples

```
# see the group codes and group colors for HS92 (6 and 8 characters)
# hs92_colors
```

`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, destination, year, classification)
```

Arguments

`origin`	is the country code of origin (e.g. "chl" for Chile)
`destination`	is the country code of origin (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
# Chile is "chl" and China is "chn"
# Visualize trade data from OEC's API (HS92 4 characters product list)
# for exports from Chile to China in the year 2014
# network("chl", "chn", 2014, 1)
# is the same as
# network("chl", "chn", 2014)
```

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

Description

Creates a network of exports for a given year

Usage

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

Arguments

origin is the country code of origin (e.g. "chl" for Chile)

destination is the country code of origin (e.g. "chn" for China)

initial_year The OEC’s API ranges from 1962 to 2014. This needs to be lower than ‘final_year‘

final_year The OEC’s API ranges from 1962 to 2014. This needs to be greater than ‘initial_year‘

interval is an optional parameter to define the distance between years (by default set to 1)

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
# Chile is "chl" and China is "chn"
# Visualize trade data from OEC's API (HS92 4 characters product list)
# for exports from Chile to China in the year 2014
# network("chl", "chn", 2014, 1)
# is the same as
# network("chl", "chn", 2014)
```

---

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_4char.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_rev2_4char.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_rev2_2char *SITC (rev. 2) groups*

Description

SITC (rev. 2) groups. This file is used to create the visualizations.

Usage

```
hs92_2char
```

Format

A data frame with 36 observations on the following 2 variables.

`sitc_rev2_group_name` Contains the SITC (rev.2) groups (e.g. machinery, electronics, etc)

`sitc_rev2_group_id` Contains the associated codes of every group (e.g. machinery is 10)

Examples

```
# see the group codes for SITC rev.2 (4 characters)
# sitc_rev2_2char
```

sitc_rev2_4char *SITC (rev.2) products (4 characters)*

Description

SITC (rev. 2) products (4 characters). This file is used to create the visualizations.

Usage

```
sitc_rev2_4char
```

Format

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

- `sitc_rev2_product_name` Contains the SITC rev.2 products' names (e.g. initiating devices, polymerization ion exchangers, etc)
- `sitc_rev2_group_name` Contains the SITC rev.2 groups (e.g. machinery, electronics products, etc)
- `sitc_rev2_product_id` Contains the associated codes of every product (e.g. initiating devices is 5722)
- `sitc_rev2_group_id` Contains the associated codes of every group (e.g. machinery is 10)
- `sitc_rev2_product_code` Contains the associated extended codes of every group (e.g. machinery is 105722 that is sitc group + sitc id)

Examples

```
# see the group codes and product codes for SITC rev.2 (4 characters)
# sitc_rev2_4char
```

sitc_rev2_colors *SITC (rev. 2) colors*

Description

SITC (rev. 2) colors. This file is used to create the visualizations.

Usage

```
hs_colors
```

Format

A data frame with 36 observations on the following 2 variables.

- `sitc_rev2_group_name` Contains the SITC (rev.2) groups (e.g. machinery, electronics, etc)
- `sitc_rev2_color` Contains the associated colors of every group (e.g. machinery is #17bcef)

Examples

```
# see the group codes and group colors for SITC rev.2 (4 characters)
# sitc_rev2_colors
```

treemap *Creates a treemap for a given year*

Description

Creates a treemap for a given year

Usage

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

Arguments

`origin`	is the country code of origin (e.g. "chl" for Chile)
`destination`	is the country code of origin (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 characters since year 1995), "2" (SITC rev.3 4 characters since year 1962) or "3" (HS92 6 characters 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
# Chile is "chl" and China is "chn"
# Visualize trade data from OEC's API (HS92 4 characters product list)
# for Chile and China in the year 2014
# treemap("chl", "chn", "exports", 2014, 1)
# is the same as
# treemap("chl", "chn", "exports", 2014)
```

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

Description

Creates a treemap for a given given period of years

Usage

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

Arguments

`origin`	is the country code of origin (e.g. "chl" for Chile)
`destination`	is the country code of origin (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
`interval`	is an optional parameter to define the distance between years (by default set to 1)
`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)
`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
# Chile is "chl" and China is "chn"
# Visualize trade data from OEC's API (HS92 4 characters product list)
# for Chile and China in the years 2011 to 2014
# treemap_interval("chl", "chn", "exports", 2011, 2014, 1, 1 ,1)
# is the same as
# treemap_interval("chl", "chn", "exports", 2011, 2014)
```

`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