

# Package ‘oec’

July 7, 2017

**Type** Package

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

**Version** 2.5

**Date** 2017-06-28

**Author** Cesar A. Hidalgo [aut],  
Alexander Simoes [aut],  
Mauricio Vargas S. [aut, cre],  
Manuel Aristaran [ctb],  
Mike Bostock [ctb] (D3),  
Dave Landy [ctb] (D3Plus),  
UN Comtrade [dct],  
MIT Media Lab [dct],  
Datawheel [fnd,cph]

**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** dplyr,  
jsonlite,  
readr,  
servr

**RoxygenNote** 6.0.1

## R topics documented:

oec-package . . . . .	2
demos . . . . .	3
getdata . . . . .	3
getdata_interval . . . . .	4
install_d3plus . . . . .	5

network	5
network_interval	6
treemap	7
treemap_interval	8
<b>Index</b>	<b>9</b>

---

oec-package	<i>The Observatory of Economic Complexity</i>
-------------	---

---

**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.

---

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()
```

---

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 2015
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 in the year 2015 from OEC's API (HS92 4 characters)
getdata("chl", "chn", 2014)
getdata("chl", "chn", 2014, 1) # equivalent to last command

# Download trade between Chile and China in the year 2015 from OEC's API (SITC rev2 4 characters)
getdata("chl", "chn", 2015, 2)

# Download trade between Chile and China in the year 2015 from OEC's API (HS92 6 characters)
getdata("chl", "chn", 2015, 3)
```

---

getdata_interval	<i>Downloads and processes the data from the API</i>
------------------	--

---

**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 2015. This needs to be lower than 'final_year'
final_year	The OEC's API ranges from 1942 to 2015. 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 in the years 2010-2015 from OEC's API (HS92 4 characters)
getdata_interval("chl", "chn", 2010, 2015)
getdata_interval("chl", "chn", 2010, 2015, 1, 1) # equivalent to last command

# Download trade between Chile and China in the years 2010, 2012 and 2014 from OEC's API (HS92 4 characters)
getdata_interval("chl", "chn", 2010, 2015, 1, 2)

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

---

install_d3plus	<i>Installs D3 and D3Plus</i>
----------------	-------------------------------

---

**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	<i>Creates a network of exports for a given year</i>
---------	--

---

**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? (2015, trade with China) (HS92 4 characters)
network("chl", "chn", 2015)
network("chl", "chn", 2015, 1)
```

---

network_interval	<i>Creates a network of exports for a given period of years</i>
------------------	---

---

**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 #' @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? (2010-2015, trade with China) (HS92 4 characters) network_interval("chl", "chn", 2010, 2015) network_interval("chl", "chn", 2010, 2015, 1, 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 given period of years.

---

treemap	<i>Creates a treemap for a given year</i>
---------	---

---

**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? (2015) (HS92 4 characters)
treemap("chl", "chn", "exports", 2015)
treemap("chl", "chn", "exports", 2015, 1) # equivalent to last command
```

---

treemap_interval	<i>Creates a treemap for a given period of years</i>
------------------	--

---

## 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? (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
```



# Index

## \*Topic **functions**

- demos, [3](#)
- getdata, [3](#)
- getdata\_interval, [4](#)
- install\_d3plus, [5](#)
- network, [5](#)
- network\_interval, [6](#)
- treemap, [7](#)
- treemap\_interval, [8](#)

countries\_list, [2](#)

demos, [2](#), [3](#)

edges\_hs92\_4.json, [3](#)

edges\_sitc\_4.json, [3](#)

getdata, [2](#), [3](#)

getdata\_interval, [2](#), [4](#)

hs92, [2](#)

install\_d3plus, [2](#), [5](#)

network, [2](#), [5](#)

network\_interval, [2](#), [6](#)

network\_template.html, [2](#)

nodes\_hs92\_4.json, [3](#)

nodes\_sitc\_4.json, [3](#)

oec (oec-package), [2](#)

oec-package, [2](#)

sitc, [2](#)

treemap, [2](#), [7](#)

treemap\_interval, [2](#), [8](#)

treemap\_template.html, [2](#)