

# Package ‘oec’

July 29, 2016

**Type** Package

**Title** The Observatory of Economic Complexity

**Version** 1.0.3

**Date** 2016-07-29

**Author** Mauricio Vargas S. <mauriciovargas@ug.uchile.cl>

**Maintainer** Mauricio Vargas S. <mauriciovargas@ug.uchile.cl>

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

**Description** Use The Observatory of Economic Complexity's API from R console to obtain international trade data to create spreadsheets (csv format) 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	2
countries_list	2
d3plus	3
demos	3
getdata	4
hs92_2char	4
hs92_6char	5
hs92_8char	5
hs_colors	6
network	6
network.compare	7
sitc_colors	8
sitc_rev2_2char	8
sitc_rev2_4char	9
treemap	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

 oec-package

*oec: The Observatory of Economic Complexity*


---

## Description

Use The Observatory of Economic Complexity's API from R console to obtain international trade data to create spreadsheets (csv format) and D3Plus visualizations.

## Details

The functions provided within this package are:

[d3plus](#) Install D3 and D3plus in your working directory.

[demos](#) Copies demo file with examples.

[getdata](#) Download trade data from OEC's API.

[network](#) Create an animated network with nodes and edges.

[network.compare](#) Creates a network that compares the exporting opportunities of a country in two different years.

[treemap](#) Creates an animated treemap.

The datasets provided within this package are:

[countries\\_list](#) Contains a list of all the countries in the world and its respective country code.

[hs92\\_2char](#) This file contains the HS92 groups. This file is used to create spreadsheets and visualizations with trade data.

[hs92\\_6char](#) This file contains the HS92 products (6 characters codes). This file is used to create spreadsheets and visualizations with trade data.

[hs92\\_8char](#) This file contains the HS92 products (8 characters codes). This file is used to create spreadsheets and visualizations with trade data.

[hs\\_colors](#) This file contains the HS92 colors. This file is used to create spreadsheets and visualizations based on trade data.

[sitc\\_rev2\\_2char](#) This file contains the SITC (rev. 2) groups. This file is used to create spreadsheets and visualizations with trade data.

[sitc\\_rev2\\_4char](#) This file contains the SITC (rev. 2) products (4 characters codes). This file is used to create spreadsheets and visualizations with trade data.

[sitc\\_colors](#) This file contains the SITC (rev. 2) colors. This file is used to create spreadsheets and visualizations with trade data.

---

 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 of a country (e.g Chile)

**Usage**

```
countries_list
```

**Format**

A data frame with 262 observations on the following 2 variables.

country the full names of the countries

country\_code the ids of the countries

**Examples**

```
countries_list
```

---

d3plus	<i>Installs D3Plus</i>
--------	------------------------

---

**Description**

Installs D3Plus

**Usage**

```
d3plus()
```

**Value**

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

**Examples**

```
d3plus()
```

---

demos	<i>Copies the demo file</i>
-------	-----------------------------

---

**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, DESTINATION, CLASSIFICATION, YEAR)
```

### 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)
CLASSIFICATION	refers to the trade classification that can be "6" (HS92 6 characters) or "8" (HS92 8 characters) for the year 1995 and going or "4" (SITC rev.2 4 characters) for the year 1962 and ongoing
YEAR	is the year and the OEC's API ranges from 1962 to 2014

### Value

Downloads the data from the OEC's API and creates the data files in json and csv formats that are needed to create the visualizations.

### Examples

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

# Download Chile (chl) and China (chn) trade data (imports, export and trade balance)
getdata("chl", "chn", 6, 2010)

# Download trade data from OEC's API (HS92 6 characters product lists)
getdata("chl", "chn", 6, 2010)

# Download trade data from OEC's API (SITC rev.2 4 characters product lists)
getdata("chl", "chn", 4, 2010)
```

---

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.

group 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)

**Examples**

hs92\_2char

---

hs92_6char	<i>HS92 products (6 characters)</i>
------------	-------------------------------------

---

**Description**

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

**Usage**

hs92\_6char

**Format**

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

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

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

product\_id Contains the associated codes of every product (e.g. horses is 010101)

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

**Examples**

hs92\_6char

---

hs92_8char	<i>HS92 products (8 characters)</i>
------------	-------------------------------------

---

**Description**

HS92 products (8 characters). This file is used to create the visualizations.

**Usage**

hs92\_8char

**Format**

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

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

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

product\_id Contains the associated codes of every product (e.g. horses is 010101)

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

**Examples**

```
hs92_8char
```

---

hs_colors	<i>HS92 colors</i>
-----------	--------------------

---

**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.

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

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

**Examples**

```
hs_colors
```

---

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

---

**Description**

Creates a network for a given year

**Usage**

```
network(ORIGIN, DESTINATION, CLASSIFICATION, YEAR)
```

**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)
CLASSIFICATION	refers to the trade classification that can be "6" (HS92 6 characters) or "8" (HS92 8 characters) for the year 1995 and going or "4" (SITC rev.2 4 characters) for the year 1962 and ongoing
YEAR	is the year and the OEC's API ranges from 1962 to 2014

**Value**

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

**Examples**

```
network("chl", "chn", 6, 2004)
```

---

network.compare	<i>Creates a network to compare two years</i>
-----------------	---

---

**Description**

Creates a network to compare two years

**Usage**

```
network.compare(ORIGIN, DESTINATION, CLASSIFICATION, YEAR1, YEAR2)
```

**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)
CLASSIFICATION	refers to the trade classification that can be "6" (HS92 6 characters) or "8" (HS92 8 characters) for the year 1995 and going or "4" (SITC rev.2 4 characters) for the year 1962 and ongoing
YEAR1	is the initial year and the OEC's API ranges from 1962 to 2014
YEAR2	is the final year and the OEC's API ranges from 1962 to 2014

**Value**

Creates an HTML file with a network visualization that compares two given years.

**Examples**

```
network.compare("chl", "chn", 6, 2000, 2010)
```

---

sitc_colors	<i>SITC (rev. 2) colors</i>
-------------	-----------------------------

---

### 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.

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

color Contains the associated colors of every group (e.g. machinery is #17bcef)

### Examples

sitc\_colors

---

sitc_rev2_2char	<i>SITC (rev. 2) groups</i>
-----------------	-----------------------------

---

### 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.

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

group\_id Contains the associated codes of every group (e.g. machinery is 10)

### Examples

sitc\_rev2\_2char



---

sitc_rev2_4char	<i>SITC (rev. 2) products (4 characters)</i>
-----------------	--

---

### Description

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

### Usage

sitc\_rev2\_4char

### Format

A data frame with 1242 observations on the following 5 variables.

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

**group** Contains the H292 groups (e.g. machinery, electronics products, etc)

**product\_id** Contains the associated codes of every product (e.g. initiating devices is 5722)

**group\_id** Contains the associated codes of every group (e.g. machinery is 10)

**id** Contains the associated extended codes of every group (e.g. machinery is 105722 that is group code + product code)

### Examples

sitc\_rev2\_4char

---

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

---

### Description

Creates a treemap for a given year

### Usage

```
treemap(ORIGIN, DESTINATION, VARIABLE, CLASSIFICATION, YEAR)
```

### 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)
CLASSIFICATION	refers to the trade classification that can be "6" (HS92 6 characters) or "8" (HS92 8 characters) for the year 1995 and going or "4" (SITC rev.2 4 characters) for the year 1962 and ongoing
YEAR	is the year and the OEC's API ranges from 1962 to 2014

**Value**

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

**Examples**

```
treemap("chl", "chn", "exports", 6, 2004)
```

# Index

## \*Topic **datasets**

- [countries\\_list](#), [2](#)
- [hs92\\_2char](#), [4](#)
- [hs92\\_6char](#), [5](#)
- [hs92\\_8char](#), [5](#)
- [hs\\_colors](#), [6](#)
- [sitc\\_colors](#), [8](#)
- [sitc\\_rev2\\_2char](#), [8](#)
- [sitc\\_rev2\\_4char](#), [9](#)

## \*Topic **functions**

- [d3plus](#), [3](#)
- [demos](#), [3](#)
- [getdata](#), [4](#)
- [network](#), [6](#)
- [network.compare](#), [7](#)
- [treemap](#), [9](#)

[countries\\_list](#), [2](#), [2](#)

[d3plus](#), [2](#), [3](#)

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

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

[hs92\\_2char](#), [2](#), [4](#)

[hs92\\_6char](#), [2](#), [5](#)

[hs92\\_8char](#), [2](#), [5](#)

[hs\\_colors](#), [2](#), [6](#)

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

[network.compare](#), [2](#), [7](#)

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

[sitc\\_colors](#), [2](#), [8](#)

[sitc\\_rev2\\_2char](#), [2](#), [8](#)

[sitc\\_rev2\\_4char](#), [2](#), [9](#)

[treemap](#), [2](#), [9](#)