Package ‘iotables’

January 18, 2019

Type Package

Title Importing and Manipulating Symmetric Input-Output Tables

Version 0.4.2

Maintainer Daniel Antal <daniel.antal@ceemid.eu>

Description Pre-processing and basic analytical tasks related to working with Eurostat's symmetric input-output tables and provide basic input-output economics calculations. The package is a part of rOpenGov <http://ropengov.github.io/> to open source open government initiatives.

URL http://iotables.ceemid.eu/

BugReports https://github.com/rOpenGov/iotables/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports dplyr, eurostat, magrittr, tidyr, stats, forcats, utils, plyr, lubridate, knitr, kableExtra, tibble, readxl, purrr

Suggests testthat, rmarkdown, spelling

Depends R(>= 2.10.0)

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Daniel Antal [aut, cre] (<https://orcid.org/0000-0001-7513-6760>)

Repository CRAN

Date/Publication 2019-01-18 15:00:10 UTC
R topics documented:

backward_linkages .................................................. 3
check_digits ............................................................ 3
coefficient_matrix_create ........................................... 4
conforming_vector_create ........................................... 5
create_knit_table ...................................................... 5
croatia_2010_1700 ..................................................... 6
croatia_2010_1800 ..................................................... 7
croatia_2010_1900 ..................................................... 8
croatia_employment_2013 ............................................ 9
croatia_employment_aggregation ................................... 10
direct_effects_create ................................................. 10
employment_get ....................................................... 11
employment_metadata ............................................... 12
empty_remove .......................................................... 13
equation_solve ........................................................ 13
forward_linkages ...................................................... 14
germany_1990 .......................................................... 15
household_column_find ............................................ 16
household_column_get ............................................... 16
indirect_effects_create ............................................. 17
input_coefficient_matrix_create .................................. 17
input_flow_get ........................................................ 18
input_indicator_create ............................................. 19
input_multipliers_create ......................................... 20
iotables_download ................................................... 21
iota_table_get ........................................................ 22
iota_table_year_get .................................................. 23
is_html_output ........................................................ 24
is_latex_output ........................................................ 25
leontieff_inverse_create .......................................... 25
leontieff_matrix_create ........................................... 26
matrix_round .......................................................... 26
metadata ............................................................... 27
metadata_uk_2010 .................................................... 27
multiplier_create ..................................................... 28
netherlands_2006 ..................................................... 29
non_zero_columns_find ............................................. 30
output_coefficient_matrix_create ................................ 30
output_get ............................................................. 31
output_multiplier_create ......................................... 31
primary_inputs ........................................................ 32
primary_input_get .................................................... 32
quadrant_separator_find .......................................... 33
round_table ........................................................... 33
supplementary_add ................................................... 34
total_tax_add .......................................................... 34
**Description**

Backward linkages as defined by the Eurostat Manual of Supply, Use and Input-Output Tables (see p506-507.)

**Usage**

```
backward_linkages(Im)
```

**Arguments**

- `Im` A Leontieff inverse matrix created by the `leontieff_inverse_create` function.

**Examples**

```
de_coeff <- input_coefficient_matrix_create( iotable_get(), digits = 4)
I <- leontieff_inverse_create(de_coeff)
backward_linkages(I)
```

---

**Description**

This is an internal function to determine if the rounding can go ahead.

**Usage**

```
check_digits(digits)
```

**Arguments**

- `digits` Digit input to check for validity.
coefficient_matrix_create

Create a coefficient matrix

Description
Create a coefficient matrix from a Symmetric Input-Output Table. The coefficient matrix is related by default to output, but you can change this to total supply or other total aggregate if it exists in your table.

Usage

coefficient_matrix_create(data_table, total = "output", digits = NULL,
remove_empty = TRUE, households = FALSE, return_part = NULL)

Arguments

data_table A symmetric input-output table, a use table, a margins or tax table retrieved by the iotable_get function.
total Usually an output vector with a key column, defaults to "output" which equals "P1" or "output_bp". You can use other rows for comparison, for example "TS_BP" if it exists in the matrix.
digits An integer showing the precision of the technology matrix in digits. Default is NULL when no rounding is applied.
remove_empty Defaults to TRUE. If you want to keep empty primary input rows, choose FALSE. Empty product/industry rows are always removed to avoid division by zero error in the analytical functions.
households Defaults to NULL. Household column can be added with TRUE.
return_part Defaults to NULL. You can choose "product" or "industry" to return an input coefficient matrix or "primary_inputs" to get only the total intermediate use and proportional primary inputs.

Value
A data.frame that contains the matrix of data_table divided by total with a key column. Optionally the results are rounded to given digits.

References
See United Kingdom Input-Output Analytical Tables 2010 for explanation on the use of the Coefficient matrix.
**conforming_vector_create**

**Examples**
```
coefficient_matrix_create(data_table = iotable_get(source = "germany_1990"),
total = "output",
digits = 4)
```

**Description**
This helper function creates you a named vector that conforms your analytical objects, such as the use table, the Leontieff-matrix, etc. With 600x60 matrices it is easy to make mistakes with manual definition. The empty effects vector can be used in .csv format as a sample to import scenarios from a spreadsheet application.

**Usage**
```
conforming_vector_create(data_table)
```

**Arguments**
- `data_table`: A use table, Leontieff-matrix, Leontieff-inverse, a coefficient matrix or other named matrix / vector.

**Examples**
```
de_input_flow <- input_flow_get(data_table = iotable_get())
conforming_vector_create(data_table = de_input_flow)
```

---

**create_knitr_table**

Create an output-independent, well formatted kable table

**Description**
Create an output-independent, well formatted kable table

**Usage**
```
create_knitr_table(data_table, digits = NULL, caption = NA,
col.names = NULL, col.width = NULL, width.unit = "cm",
col.align = NULL, border.right.cols = NULL, bold.cols = NULL,
bootstrap.options = c("striped", "hover", "condensed"),
latex.options = NULL, output.format = NULL, keep.pdf = FALSE,
latex.header.includes = c("\\usepackage{magyar}{babel}",
"\\usepackage[utf8]{inputenc}"")
```
Arguments

- `data_table`: data.frame, tibble, named matrix or a knitr_kable object.
- `digits`: Number of digits to display in the case of numeric variables.
- `caption`: A table caption, defaults to empty NA.
- `col_names`: The col_names parameter of the kable table, if NULL, filled with the names of the data_table.
- `col_width`: Defaults to NULL. In this case all col_align parameters will be "c" for centered.
- `width_unit`: Defaults to "cm".
- `col_align`: Defaults to NULL. In this case 'l' for the first column and 'r' for the rest of the columns, i.e."l", "c", ...., "c"
- `border_right_cols`: Defaults to NULL. In this case TRUE for the first column and FALSE for the rest of the columns, i.e.T, F, ...., F
- `bold_cols`: Defaults to NULL. In this case none of the the columns are bold, i.e. identical to F, F, ...., F.
- `bootstrap_options`: Defaults to c("striped", "hover", "condensed") and only used for output_format = "html".
- `latex_options`: Defaults to NULL.
- `output_format`: Defaults to 'html'. Alternatives are 'latex', 'image', recommended for Word files, is removed because it depends on magick which is not available on all R platforms.
- `keep_pdf`: Defaults to FALSE and only used if output_format = 'image'.
- `latex_header_includes`: Currently defaults to c("\usepackage{magyar}{babel}","\usepackage[utf8]{inputenc}") It can be any valid latex option setting, but if packages are used, the packages must be installed on your Latex engine.

Examples

```r
foo = data.frame (observation = c("indicator1", "indicator2", "indicator3"),
indicator_1 = c(100,105,95),
indicator_2 = c(102,104,76)
)
```

---


---

Description

1700 - Symmetric input-output table at basic prices (product x product) In thousand kunas (T_NAC)
Usage

data(croatia_2010_1800)

Format

A data frame with 13 variables.

t_rows2 Technology codes in row names, following the Eurostat convention.
t_rows2_lab Longer labels for t_rows2
t_cols2 Technology codes in column names, following the Eurostat convention.
t_cols2_lab Longer labels for t_cols2
iotables_col The standardized iotables column labelling for easier reading.
col_order The column ordering to keep the matrix legible.
row_order The row ordering to keep the matrix legible.
iotables_row The standardized iotables row labelling for easier reading.
unit Different from Eurostat tables, in thousand national currency units.
geo ISO / Eurostat country code for Croatia
geo_lab ISO / Eurostat country name, Croatia.
time Date of the SIOT
values The actual values of the table in thousand kunas

Source


---


Description

1800 - Symmetric input-output table for domestic production (product x product) In thousand kunas (T_NAC)

Usage

data(croatia_2010_1800)
Format

A data frame with 13 variables.

- **t_rows2**: Technology codes in row names, following the Eurostat convention.
- **t_rows2_lab**: Longer labels for t_rows2
- **values**: The actual values of the table in thousand kunas
- **t_cols2**: Column labels, following the Eurostat convention with differences. CPA_ suffix added to original DZS column names.
- **t_cols2_lab**: Longer labels for t_cols2
- **iotables_col**: The standardized iotables column labelling for easier reading.
- **col_order**: The column ordering to keep the matrix legible.
- **iotables_row**: The standardized iotables row labelling for easier reading.
- **row_order**: The row ordering to keep the matrix legible.
- **unit**: Different from Eurostat tables, in thousand national currency units.
- **geo**: ISO / Eurostat country code for Croatia
- **geo_lab**: ISO / Eurostat country name, Croatia.
- **time**: Date of the SIOT

Source


Description

1900 - Symmetric input-output table for imports (product x product) In thousand kunas (T_NAC)

Usage

data(croatia_2010_1900)

Format

A data frame with 13 variables.

- **t_rows2**: Technology codes in row names, following the Eurostat convention.
- **t_rows2_lab**: Longer labels for t_rows2
- **values**: The actual values of the table in thousand kunas
- **t_cols2**: Column labels, following the Eurostat convention with differences. CPA_ suffix added to original DZS column names.
- **t_cols2_lab**: Longer labels for t_cols2
**Description**

Aggregate Croatian detailed employment statistics into the Croatian (EU standard) Symmetric input-output table format.

**Usage**

data(croatia_employment_2013)

**Format**

A data frame with 107 observations in 2 variables:

- **code**  Short labels
- **iotables_row**  iotables style labels
- **employment**  Employment in the sector in Croatia, not in thousands!
croatia_employment_aggregation

*Aggregation table for Croatian employment statistics*

**Description**

Aggregate Croatian detailed employment statistics into the Croatian (EU standard) Symmetric input-output table format.

**Usage**

data(croatia_employment_aggregation)

**Format**

A data frame with 105 rows (including empty ones) and 2 variables.

- **employment_label** Labelling in DZS English language export
- **t_cols2** Labelling of EU/DZS SIOTs.

---

direct_effects_create  
*Create direct effects*

**Description**

The function creates the effects.

**Usage**

direct_effects_create(input_requirements, inverse, digits = NULL)

**Arguments**

- **input_requirements**
  A matrix or vector created by `input_indicator_create`
- **inverse**
  A Leontieff-inverse created by `leontieff_inverse_create`
- **digits**
  Rounding digits, defaults to `NULL`, in which case no rounding takes place.
Examples

```r
nl <- netherlands_2006

input_coeff_nl <- input_coefficient_matrix_create(
  data_table  = netherlands_2006,
  households  = FALSE)

compensation_indicator <- input_indicator_create(netherlands_2006, 'compensation_employees')

I_nl <- leontieff_inverse_create( input_coeff_nl )

direct_effects_create(input_requirements = compensation_indicator,
                      inverse = I_nl)
```

---

**employment_get**

*Get employment data*

Description

Download the employment data for a country and arrange it to the 64x64 SIOTS. Currently works only with product x product tables.

Usage

```r
employment_get(geo = "CZ", year = "2010", sex = "Total",
                age = "Y_GE15", labelling = "iotables", data_directory = NULL,
                force_download = TRUE)
```

Arguments

- **geo**
  - The country code.

- **year**
  - The year. The average employment will be created for the given year, starting with 2008, when the NACE Rev 2 was introduced in employment statistics.

- **sex**
  - Defaults to "Total". Enter "Females" or "F" for female employment, "Males" or "M" for male employment.

- **age**
  - Defaults to "Y_GE15", which is the Eurostat code for employment in all age groups starting from 15-years-old. Any Eurostat code can be used as a parameter.

- **labelling**
  - Either "iotables" or the applicable short code, for product x product SIOTs "prod_na" and in the case of industry x industry SIOTs "induse".

- **data_directory**
  - Defaults to NULL, if a valid directory, it will try to save the pre-processed data file here with labelling.

- **force_download**
  - Defaults to TRUE. If FALSE it will use the existing downloaded file in the data_directory or the temporary directory, if it exists.
Source

Eurostat statistic Employment by sex, age and detailed economic activity (from 2008 onwards, NACE Rev. 2 two digit level) - 1 000

Examples

```r
## Not run:
io_tables <- get_employment (  
  geo = "CZ",  
  year = "2010",  
  sex = "Total",  
  age = "Y_GE15",  
  data_directory = NULL,  
  force_download = TRUE
)

## End(Not run)
```

employment_metadata        Employment metadata

Description

An arrangement of the Eurostat national accounts vocabulary to match with employment statistics data.

Usage

data(metadata)

Format

A data frame with 6 variables.

- **emp_code** code used in the employment statistics
- **code** Eurostat labels for SIOTs corresponding to emp_code
- **label** Eurostat label descriptions for SIOTs corresponding to emp_code
- **variable** Eurostat vocabulary source, i.e. t_rows, t_cols, prod_na, induse
- **group** Different from Eurostat tables, in thousand national currency units.
- **iottables_label** Custom, machine readable snake format variable names
empty_remove

Symmetrically remove empty rows and columns

Description

This is an internal function to determine where to separate quadrants if necessary.

Usage

empty_remove(data_table)

Arguments

data_table A symmetric input-output table, or a symmetric part of a use table or a supply table.

Value

A tibble/data.frame with a key row and a symmetric matrix, after removing all empty columns and rows at the same time.

equation_solve

Solve a basic equation

Description

The function matches to parts of the matrix equation, using the named formats with row names and solves the matrix equation. This function is used in wrapper functions, such as multiplier_create, to solve particular problems, but it can be used directly, too. The function only performs the lhs pairing industries and checking for exceptions.

Usage

equation_solve(LHS = NULL, Im = NULL)

Arguments

LHS A left-hand side vector with a key column containing the industry or product names for matching, for example the employment coefficients.

Im A Leontieff-inverse with a key column containing the industry or product names for matching.
**forward_linkages**

**Examples**

```r
Im = data.frame (
a = c("row1", "row2"),
b = c(1,1),
c = c(2,0))
LHS = data.frame (
a = "1hs",
b = 1,
c = 0.5)
equation_solve (Im = Im, LHS = LHS)
```

**Description**

Forward linkages as defined by the Eurostat Manual of Supply, Use and Input-Output Tables (see p506-507.)

**Usage**

```r
forward_linkages(output_coefficient_matrix, digits = NULL)
```

**Arguments**

- `output_coefficient_matrix` An output coefficient matrix created with the `output_coefficient_matrix_create` function.
- `digits` Number of decimals for rounding, defaults to NULL.

**Examples**

```r
data_table = iotable_get()

dde_out <- output_coefficient_matrix_create (
da_table, "tfu", digits = 4
)

forward_linkages ( output_coefficient_matrix = de_out,
digits = 4 )
```
Simple input-output table for Germany, 1990.

Description

For testing purposes a well documented example data set is used from the Eurostat manual. The table in the Eurostat manual is brought to the format used by the Eurostat database. It is a small dataset for examples, but it is also instructive to understand how Eurostat stores the highly structured SIOTs in long-form tidy datasets. The labels were slightly altered to reflect the transition from the vocabulary of ESA95 to ESA2010 since the publication of the Manual.

Usage

data(germany_1990)

Format

A data frame with 228 observations and 10 variables.

- **prod_na**: Technology codes in row names, following the Eurostat convention.
- **prod_na_lab**: Longer labels for t_rows2.
- **induse**: Column labels, following the Eurostat convention with differences.
- **iotables_row**: Row labels, i.e. to be used in key column, for iotables package abbreviations.
- **iotables_col**: Column labels for iotables package abbreviations.
- **values**: The actual values of the table in million euros.
- **unit**: MIO_EUR, the same as Eurostat.
- **unit_lab**: Million euros. Eurostat usually has euro and national currency unit values, too.
- **geo**: ISO / Eurostat country code for Germany, i.e. DE.
- **geo_lab**: ISO / Eurostat country name, Germany.
- **time**: Date of the SIOT.

Source

Eurostat Manual of Supply, Use and Input-Output Tables p 492
household_column_find  
*Return the position of final household expenditure*

**Description**
Return the position of final household expenditure

**Usage**
```python
household_column_find(data_table)
```

**Arguments**
- `data_table` A symmetric input output table, a use table or a supply table.

**Value**
An integer value with the final household expenditure. Returns NULL if not found.

**Examples**
```
household_column_find( iotable_get ( source = 'germany_1990') )
```

---

household_column_get  
*Return final household expenditure*

**Description**
Return final household expenditure

**Usage**
```python
household_column_get(data_table)
```

**Arguments**
- `data_table` A symmetric input output table, a use table or a supply table.

**Value**
The column containing final household expenditure. If not found NULL is returned.

**Examples**
```
household_column_get( iotable_get ( source = 'germany_1990') )
```
**indirect_effects_create**

*Create indirect effects*

**Description**

The function creates the effects.

**Usage**

```r
indirect_effects_create(input_requirements, inverse, digits = NULL)
```

**Arguments**

- **input_requirements**
  A matrix or vector created by `input_indicator_create`

- **inverse**
  A Leontief-inverse created by `leontieff_inverse_create`.  

- **digits**
  Rounding digits, defaults to NULL, in which case no rounding takes place.

**Examples**

```r
nl <- netherlands_2006

input_coeff_nl <- input_coefficient_matrix_create(
  data_table = netherlands_2006, 
  households = FALSE)

compensation_indicator <- input_indicator_create(netherlands_2006, 'compensation_employees')

I_nl <- leontieff_inverse_create(input_coeff_nl)

direct_effects_create(input_requirements = compensation_indicator, 
  inverse = I_nl)
```

**input_coefficient_matrix_create**

*Create an input coefficient matrix*

**Description**

Create an input coefficient matrix from the input flow matrix and the output vector. The two input vectors must have consistent labelling, i.e the same column names must be found in the use table (input flow) and the output vector.
Usage

```r
input_coefficient_matrix_create(data_table, households = FALSE, digits = NULL)
```

Arguments

- `data_table` A symmetric input-output table, a use table, a margins or tax table retrieved by the `iotable_get` function.
- `households` Defaults to `NULL`. Household column can be added with `TRUE`.
- `digits` An integer showing the precision of the technology matrix in digits. Default is `NULL` when no rounding is applied.

Details

The terminology follows the Eurostat Manual of Supply, Use and Input-Output Tables. Input-Output Multipliers Specification Sheet and Supporting Material, Spicosa Project Report, which cannot be linked due to a malformatted url, but can be found with a search engine. This matrix is called 'technological coefficients'. The results of the function are tested on both sources. This is a wrapper function around `coefficient_matrix_create`.

Value

A data frame that contains the matrix of first quadrant of the use table as `input_flow` divided by `output` supported by a key column of product or industries, with a key column. Optionally the results are rounded to given `digits`.

Examples

```r
input_coefficient_matrix_create (  
  iotable_get(),  
  digits = 4 )

#This is a wrapper function and equivalent to

coefficient_matrix_create( iotable_get(),  
  total = "total",  
  return = "products")
```

Description

Select the use table from a symmetric input-output table.

Usage

```r
input_flow_get(data_table, empty_remove = FALSE, households = TRUE)
```
**input_indicator_create**

Create input indicator(s)

**Description**

The function creates the input indicators from the inputs and the outputs.

**Usage**

```r
input_indicator_create(data_table, input_vector = c("gva_bp", "net_tax_production"), digits = NULL, households = FALSE, indicator_names = NULL)
```

**Arguments**

- `data_table`  
  A symmetric input-output table, a use table, a margins or tax table retrieved by the `iotable_get` function.

- `input_vector`  
  The name of inputs for which you want to create the indicators. They must be found in the `data_table`.

- `digits`  
  Rounding digits, if omitted, no rounding takes place.

- `households`  
  If the households column should be added, defaults to `FALSE`.

- `indicator_names`  
  The names of new indicators. Defaults to `NULL` when the names in the key column of `input_matrix` will be used to create the indicator names.

**Value**

A tibble (data frame) containing `input_matrix` divided by the `output_vector` with a key column for products or industries.
input_multipliers_create

Create input indicators

Description

The function creates the multipliers (direct + indirect effects).

Usage

input_multipliers_create(input_requirements, inverse, digits = NULL)

Arguments

- **input_requirements**
  A matrix or vector created by `input_indicator_create`

- **inverse**
  A Leontief-inverse created by `leontieff_inverse_create`.

- **digits**
  Rounding digits, defaults to NULL, in which case no rounding takes place.

Examples

```r
nl <- netherlands_2006

input_coeff_nl <- input_coefficient_matrix_create(
  data_table = netherlands_2006,
  households = FALSE)

compensation_indicator <- input_indicator_create(netherlands_2006, 'compensation_employees')

I_nl <- leontieff_inverse_create( input_coeff_nl )

input_multipliers_create(input_requirements = compensation_indicator, inverse = I_nl)
```
Description

This function downloads standard input-output table files. Currently only Eurostat files are supported. You are not likely to use this function, because `iotable_get` will call this function if necessary and properly filter out an input-output table. The only parameter is the Eurostat code of the table. The data is downloaded in the `tempdir()` under the name the statistical product as an rds file. (For example: `naio_10_cp1750.rds`) The temporary directory is emptied at every normal R session exit. To save the file for further use (which is necessary in analytical work because download times are long) set the `download_directory` [see parameters]. The function will make a copy of the rds file in this directory.

- `naio_10_cp1700` Symmetric input-output table at basic prices (product by product)
- `naio_10_pyp1700` Symmetric input-output table at basic prices (product by product) (previous years prices)
- `naio_10_cp1750` Symmetric input-output table at basic prices (industry by industry)
- `naio_10_pyp1750` Symmetric input-output table at basic prices (industry by industry) (previous years prices)
- `naio_10_cp15` Supply table at basic prices incl. transformation into purchasers’ prices
- `naio_10_cp16` Use table at purchasers’ prices
- `naio_10_cp1610` Use table at basic prices
- `naio_10_pyp1610` Use table at basic prices (previous years prices) (naio_10_pyp1610)
- `naio_10_cp1620` Table of trade and transport margins at basic prices
- `naio_10_pyp1620` Table of trade and transport margins at previous years’ prices
- `naio_10_cp1630` Table of taxes less subsidies on products at basic prices
- `naio_10_pyp1630` Table of taxes less subsidies on products at previous years’ prices
- `uk_2010_siot` United Kingdom Input-Output Analytical Tables data

Usage

```r
iotables_download(source = "naio_10_cp1700", data_directory = NULL, force_download = TRUE)
```

Arguments

- `source` See the available list of sources above in the Description.
- `data_directory` Defaults to NULL, if a valid directory, it will try to save the pre-processed data file here with labelling.
- `force_download` Defaults to TRUE. If FALSE it will use the existing downloaded file in the data_directory or the temporary directory, if it exists.
Examples

```r
## Not run:
io_tables <- iotables_download ( source = "naio_10_cp1700"
)

## End(Not run)
```

### iotable_get

*Get an input-output table from bulk file*

#### Description

This function is used to filter out a single input-output table from a database, for example a raw file downloaded from the Eurostat website. It provides some functionality to avoid some pitfalls. Unless you want to work with bulk data files, you should not invoke `iotables_download` directly, rather via this function, if and when it is necessary.

#### Usage

```r
iotable_get(labelled_io_data = NULL, source = "germany_1990",
geo = "DE", year = 1990, unit = "MIO_EUR", stk_flow = "DOM",
labelling = "iotables", data_directory = NULL,
force_download = TRUE)
```

#### Arguments

- **labelled_io_data**
  
  If you have downloaded a bulk data file with `iotables_download`, it is faster to work with the data in the memory. Defaults to `NULL` when the data will be retrieved from the hard disk or from the Eurostat website invoking the same function.

- **source**
  
  A data source, for example `naio_10_cp1700`.

  - `naio_10_cp1700` Symmetric input-output table at basic prices (product by product)
  - `naio_10_pyp1700` Symmetric input-output table at basic prices (product by product) (previous years prices)
  - `naio_10_cp1750` Symmetric input-output table at basic prices (industry by industry)
  - `naio_10_pyp1750` Symmetric input-output table at basic prices (industry by industry) (previous years prices)
  - `naio_10_cp15` Supply table at basic prices incl. transformation into purchasers' prices
  - `naio_10_cp16` Use table at purchasers' prices
  - `naio_10_cp1610` Use table at basic prices
  - `naio_10_pyp1610` Use table at basic prices (previous years prices) (`naio_10_pyp1610`)
  - `naio_10_cp1620` Table of trade and transport margins at basic prices
• `naio_10_pyp1620` Table of trade and transport margins at previous years’ prices
• `naio_10_cp1630` Table of taxes less subsidies on products at basic prices
• `naio_10_pyp1630` Table of taxes less subsidies on products at previous years’ prices

For further information consult the Eurostat Symmetric Input-Output Tables page.

**geo**
A country code or a country name. For example, SK or as Slovakia.

**year**
A numeric variable containing the year. Defaults to 2010, because this year has the most data.

**unit**
A character string containing the currency unit, defaults to `MIO_NAC` (million national currency unit). The alternative is `MIO_EUR`.

**stk_flow**
Defaults to DOM as domestic output, alternative IMP for imports and TOTAL for total output. For source = ‘naio_10_cp1620’ and trade and transport margins and source = ‘naio_10_cp1630’ taxes less subsidies only TOTAL is not used.

**labelling**
Defaults to `iotables` which gives standard row and column names regardless of the source of the table, or if it is a product x product, industry x industry or product x industry table. The alternative is `short` which is the original short row or column code of Eurostat or OECD.

**data_directory**
Defaults to NULL, if a valid directory, it will try to save the pre-processed data file here with labelling.

**force_download**
Defaults to TRUE. If FALSE it will use the existing downloaded file in the data_directory or the temporary directory, if it exists. Will force download only in a new session.

**Examples**
```r
germany_table <- iotable_year_get( source = "germany_1990", geo = 'DE',
year = 1990, unit = "MIO_EUR",
labelling = "iotables")
```

**Description**
The function selects the available tables by year or time as a date for a specific country and currency unit in the Eurostat bulk file. Unless you want to work with bulk data files, you should not invoke `iotables_download` directly, rather via this function, if and when it is necessary.

**Usage**
```r
iotable_year_get(labelled_io_data = NULL, source = "germany_1990", geo = "DE", unit = "MIO_EUR", time_unit = "year",
stk_flow = "TOTAL", data_directory = NULL, force_download = TRUE)
```
Arguments

labelled_io_data

If you have downloaded a bulk data file with `iotables_download`, it is faster to work with the data in the memory. Defaults to NULL when the data will be retrieved from the hard disk or from the Eurostat website invoking the same function.

source

A data source, for example `naio_10_cp1700`. Symmetric input-output table at basic prices (product by product) (naio_10_cp1700) Symmetric input-output table at basic prices (industry by industry) (naio_10_cp1750) Symmetric input-output table at basic prices (product by product) (previous years prices) (naio_10_pyp1700) Symmetric input-output table at basic prices (industry by industry) (previous years prices) (naio_10_pyp1750) Table of trade and transport margins at basic prices (naio_10_cp1620) and at previous’ years prices (naio_10_pyp1620) Table of taxes less subsidies on products at basic prices (naio_10_cp1630) and at previous’ years prices (naio_10_pyp1630) For further information consult the Eurostat Symmetric Input-Output Tables page.

go

A country code or a country name. For example, SK or as Slovakia.

unit

A character string containing the currency unit, defaults to MIO_NAC (million national currency unit). The alternative is MIO_EUR.

time_unit

Defaults to ‘year’ and years are returned as numbers. Alternative is to return ‘time’ as vector of dates.

stk_flow

Defaults to DOM as domestic output, alternative IMP for imports and TOTAL for total output. For source = ‘naio_10_cp1620’ and trade and transport margins and source = ‘naio_10_cp1630’ taxes less subsidies only TOTAL is not used.

data_directory

Defaults to NULL. Use if you used a data_directory parameter with `iotable_get` or `iotables_download`.

force_download

Defaults to TRUE. If FALSE it will use the existing downloaded file in the data_directory or the temporary directory, if it exists. Will force download only in a new session.

Examples

germany_years <- iotable_year_get ( source = "germany_1990", geo = 'DE',
                                    unit = "MIO_EUR" )

---

is_html_output

Check if HTML output is required

Description

Check if HTML output is required
Check if Latex output is required

Description

Check if Latex output is required

leontieff_inverse_create

Create the inverse of a Leontieff-matrix.

Description

The inversion takes place after the basic properties of the Leontieff matrix.

Usage

leontieff_inverse_create(technology_coefficients_matrix, digits = NULL)

Arguments

technology_coefficients_matrix

A technology coefficient matrix created by the input_coefficient_matrix_create or output_coefficient_matrix_create.

digits

An integer showing the precision of the technology matrix in digits. Default is NULL when no rounding is applied.

Examples

```r
tm <- input_flow_get(
  data_table = iotable_get(),
  households = FALSE)
I <- leontieff_inverse_create( technology_coefficients_matrix = tm )
```
Matrix Round

leontieff_matrix_create

Create a Leontieff matrix

Description

Create a Leontieff matrix from technology matrix after some basic error handling. Most likely you will need this function as a step to invoke the function to create its inverse: leontieff_inverse_create.

Usage

leontieff_matrix_create(technology_coefficients_matrix)

Arguments

technology_coefficients_matrix

A technology coefficient matrix created by the input_coefficient_matrix_create or output_coefficient_matrix_create.

Examples


tm <- input_flow_get(
    data_table = iotable_get(),
    households = FALSE)
l <- leontieff_matrix_create( technology_coefficients_matrix = tm )


matrix_round

Round all matrix values to required number of digits.

Description

Round all matrix values to required number of digits.

Usage

matrix_round(data_table, digits = 0)

Arguments

data_table

A symmetric input output table, a use table or a supply table.

digits

An integer number, defaults to 0.

Value

The matrix, with the intact key column and the numeric columns rounded.
Description

An arrangement of the Eurostat national accounts vocabulary, used to correctly order wide format rows and columns from bulk long-form tables.

Usage

data(metadat)
multiplier_create

Description
This function is in fact a wrapper around the equation_solve function, adding a key column with the name to the multiplier the maintain structural consistency.

Usage
multiplier_create(input_vector, Im, multiplier_name = "multiplier", digits = NULL)

Arguments
input_vector  An input matrix or vector created by the input_indicator_create function.
Im  The Leontieff inverse as a named object created by the leontieff_inverse_create function.
multiplier_name  A variable name to be given to the returned multipliers. Defaults to multiplier.
digits  Rounding digits, if omitted, no rounding takes place.

Details
As opposed to direct effects, multipliers are expressed per input of product/industry.
Examples

data_table <- iotable_get()

coeff_de <- input_coefficient_matrix_create( data_table )

degva_indicator <- input_indicator_create ( 
  data_table = data_table,
  input = 'gva')  #this is a correct input

I_de <- leontieff_inverse_create( coeff_de )

degva_multipliers <- multiplier_create ( 
  input_vector = degva_indicator,
  Im = I_de,
  multiplier_name = "employment_multiplier",
  digits = 4 )

---


Description

This simplified SIOT is taken from the Science Policy Integration for Coastal Systems Assessment project’s input-output multiplier specification sheet. It is used as a simple example SIOT for controlled analytical results. The column names were slightly altered to resemble more the current Eurostat conventions and the main example dataset germany_1990.

Usage

data(netherlands_2006)

Format

A data frame with 14 observations and 13 variables.

Source

Source: Input-Output Multipliers Specification Sheet and Supporting Material in the Spicosa Project Report
**non_zero_columns_find**  
*Find non-zero columns*

**Description**

This is an internal function to help finding empty columns and rows in symmetric tables.

**Usage**

\[
\text{non_zero_columns_find}(\text{data_table})
\]

**Arguments**

- **data_table**: A symmetric input output table, a use table or a supply table.

**Value**

A vector of TRUE and FALSE values for the table.

**output_coefficient_matrix_create**  
*Create an output coefficient matrix*

**Description**

Create an output coefficient matrix from the input flow matrix or a symmetric input-output table. If there are zero values in present, they will be changed to 0.000001 and you will get a warning. Some analytical equations cannot be solved with zero elements. You either have faulty input data, or you have to use some sort of data modification to carry on your analysis.

**Usage**

\[
\text{output_coefficient_matrix_create}(\text{io_table}, \text{total} = \text{"tfu"}, \text{digits} = \text{NULL})
\]

**Arguments**

- **io_table**: A symmetric input-output table or use table created with the \text{iotable_get} function which contains the 'total' column. In case you use \text{type="tfu"} you need to input a full iotable, create by the \text{iotable_get}, because you will need the final demand column.

- **total**: The output=’total’ (or CPA_TOTAL, depending on the names in your table, default) returns the output coefficients for products (intermediates) while the \text{final_demand} returns output coefficients for final demand. See Eurostat Manual, p495 and p507.

- **digits**: An integer showing the precision of the technology matrix in digits. Default is NULL when no rounding is applied.
**output_get**

**Examples**

\[
\text{io_table} \leftarrow \text{iotable_get}()
\]

\[
\text{output_coefficient_matrix_create} \left( \text{io_table = io_table,}
\text{total = 'tfu'},
\text{digits = 4} \right)
\]

**Description**

This is a wrapper function around the `primary_input_get` function.

**Usage**

\[
\text{output_get(data_table)}
\]

**Arguments**

- `data_table`: A symmetric input-output table or use table retrieved by the `iotable_get` function.

**Examples**

\[
\text{output_get} \left( \text{data_table = iotable_get()} \right)
\]

---

**output_multiplier_create**

**Output multipliers**

**Description**

Output multipliers as defined by the Eurostat Manual of Supply, Use and Input-Output Tables on p500.

**Usage**

\[
\text{output_multiplier_create(input_coefficient_matrix)}
\]

**Arguments**

- `input_coefficient_matrix`: A Leontieff inverse matrix created by the `input_coefficient_matrix_create` function.
Examples

de_input_coef <- input_coefficient_matrix_create(
    iotable_get(), digits = 4)

output_multiplier_create ( de_input_coef )

<table>
<thead>
<tr>
<th>primary_inputs</th>
<th>Primary input abbreviations</th>
</tr>
</thead>
</table>

Description

Only currently used primary inputs. Abbreviations for filtering.

Usage

data("croatia_employment_aggregation")

Format

A data frame with 105 rows (including empty ones) and 2 variables.

- `t_rows2` Eurostat code of the input.
- `t_rows2_lab` Labelling of the input by Eurostat.
- `source` Eurostat / DZS
- `indicator` Human readable abbreviation

<table>
<thead>
<tr>
<th>primary_input_get</th>
<th>Get primary inputs</th>
</tr>
</thead>
</table>

Description

This function will retrieve any primary input from the input-output table.

Usage

primary_input_get(data_table, primary_input = "compensation_employees")

Arguments

- `data_table` A symmetric input-output table, a use table, or a supply table retrieved by the `iotable_get` function.
- `primary_input` The primary input to be returned from the table.
quadrant_separator_find

Examples

```r
comp_employees_de <- primary_input_get(
  data_table = iotable_get(),
  primary_input = "compensation_employees")
```

---

**Determine the end of Quadrant I and III.**

**Description**

This is an internal function to determine where to separate quadrants if necessary.

**Usage**

```r
quadrant_separator_find(data_table, include_total = FALSE)
```

**Arguments**

- `data_table`: A symmetric input output table, a use table or a supply table.
- `include_total`: Should the total (intermediary) output column be included TRUE or excluded FALSE (default)?

**Value**

An integer value with the last column of Quadrant I and III. If the last column is not found, 2 is returned with a warning to avoid stopping a pipeline.

---

**round_table**

**Systematically round values in a table.**

**Description**

This is an internal function to do the rounding on all numeric elements of the table.

**Usage**

```r
round_table(data_table, digits = NULL)
```

**Arguments**

- `data_table`: A symmetric input-output table, a use table, a supply table, a margin or tax table.
- `digits`: Number of digits for rounding.

**Value**

A tibble/data.frame with a key row and a symmetric matrix, after removing all empty columns and rows at the same time.
**supplementary_add**  
*Add supplementary data*

**Description**

Download the employment data for a country and arrange it to the 64x64 SIOTS. Currently works only with product x product tables.

**Usage**

```r
supplementary_add(data_table, supplementary_data,  
  supplementary_names = NULL)
```

**Arguments**

- `data_table`  
  A SIOT, a use table, a supply table, or a margins table.

- `supplementary_data`  
  Supplementary data to be added. It must be a data.frame or tibble with a key column containing the indicator’s name, and the column names must match with the `data_table`. Can be a vector or a data frame of several rows.

- `supplementary_names`  
  Optional names for the new supplementary rows. Defaults to `NULL`.

**Examples**

```r
de_io <- iotable_get()
CO2 <- c(0.2379, 0.5172, 0.0456, 0.1320, 0.0127, 0.0530)
names(CO2) <- c("agriculture_group", "industry_group", "construction",  
  "trade_group", "business_services_group", "other_services_group")
CO2 <- cbind(  
  data.frame(iotables_row = "CO2"), as.data.frame(t(CO2)))
de_coeff <- input_coefficient_matrix_create(iotable_get())
supplementary_add(de_io, CO2)
```

---

**total_tax_add**  
*Summarize and add tax data*

**Description**

Summarize and add tax data

**Usage**

```r
total_tax_add(data_table, tax_names = c("d21x31", "d29x39"),  
  total_tax_name = "TOTAL_TAX")
```
Arguments

data_table A SIOT, a use table, a supply table, or a margins table that has product and production tax rows in among the primary inputs.
tax_names Defaults to ("d21x31", "d29x39"), which are the Eurostat names for taxes. The parameter is not case sensitive.
total_tax_name Defaults to 'TOTAL_TAX'. The name of the summarized row. It is case sensitive.

Examples

dc_io <- iotable_get()
total_tax_add ( dc_io,
tax_names = c("net_tax_products", "net_tax_production"),
total_tax_name = "total_tax")

uk_2010_data United Kingdom Input-Output Analytical Tables, 2010

Description

The Excel-imported UK data.

Usage

data(uk_2010_data)

Format

A data frame with 10 variables.

uk_row The UK row identifier. Dots and '&' converted to '·'.
uk_row_lab The original UK row labels.
uk_col The UK row identifier. Dots and '&' converted to '·'.
uk_col_lab The original UK column labels.
geo Eurostat-style geocode, i.e. UK
geo_lab United Kingdom
indicator The name of the indicator, i.e. Excel sheet.
unit Eurostat label equivalents units, i.e. MIO_NAC.
unit_lab Eurostat label equivalents, i.e. millions of national currency unit.
values The numeric values of the variable
year Contant = 2010.

Source

United Kingdom Input-Output Analytical Tables 2010
uk_2010_get  Get United Kingdom Input-Output Analytical Tables, 2010

Description

This function will retrieve any primary input from the input-output table. United Kingdom Input-Output Analytical Tables, 2010 (consistent with UK National Accounts Blue Book 2013 & UK Balance of Payments Pink Book 2013) by Richard Wild.

Usage

uk_2010_get(path = NULL)

Arguments

path  A path to the downloaded file, if already exists, given with file.path() function.

Source

ukioanalyticaltablesio1062010detailedpubversion.xls

Examples

## Not run:
uk2010 <- uk_2010_get()

## End(Not run)

uk_2010_results_get  Get United Kingdom Multipliers and Effects, 2010

Description

This function will retrieve the published effects and multipliers from the United Kingdom Input-Output Analytical Tables, 2010 (consistent with UK National Accounts Blue Book 2013 & UK Balance of Payments Pink Book 2013) by Richard Wild.

Usage

uk_2010_results_get(path = NULL)

Arguments

path  A path to the downloaded file, if already exists, given with file.path() function.
Source

`ukioanalyticaltablesio1062010detailedpubversion.xls`

Examples

```r
## Not run:
uk_results <- iotables::uk_2010_results_get ()

## End(Not run)
```

---

**uk_test_results**  
*Multipliers and effects (product) for testing from the United Kingdom Input-Output Analytical Tables, 2010*

---

Description

The Excel-imported UK data.

Usage

```r
data(uk_test_results)
```

Format

A data frame with 12 variables.

- **uk_row_label**: The UK row label
- **Output multiplier**: The imported Output multipliers
- **output_multiplier_rank**: The imported ranking of output multipliers
- **Employment cost multiplier**: The imported Employment cost multipliers.
- **employment_cost_multiplier**: The imported ranking of Employment cost multipliers.
- **Employment cost effects**: The imported Employment cost multipliers.
- **employment_cost_effects**: The imported ranking of employment cost multipliers.
- **GVA effects**: The imported GVA effects.
- **gva_effects_rank**: The imported ranking GVA effects.
- **gva_multiplier_rank**: The imported ranking GVA multipliers.
- **GVA multiplier**: The imported GVA multipliers.
- **indicator**: Indicator names.
Pipe operator

Description

Pipe operator

Arguments

lhs, rhs  A visualisation and a function to apply to it

Examples

mtcars %>% summary
Index

*Topic Croatia,
  croatia_2010_1900, 8
  croatia_employment_2013, 9
  croatia_employment_aggregation, 10
*Topic Croatia
  croatia_2010_1700, 6
  croatia_2010_1800, 7
*Topic Eurostat,
  primary_inputs, 32
*Topic Germany
  germany_1990, 15
*Topic Netherlands
  netherlands_2006, 29
*Topic UK
  metadata_uk_2010, 27
  uk_2010_data, 35
  uk_test_results, 37
*Topic data,
  croatia_2010_1700, 6
  croatia_2010_1800, 7
  croatia_2010_1900, 8
  croatia_employment_aggregation, 10
  employment_metadata, 12
  germany_1990, 15
  metadata, 27
  netherlands_2006, 29
  primary_inputs, 32
  uk_2010_data, 35
  uk_test_results, 37
*Topic datasets,
  croatia_2010_1700, 6
  croatia_2010_1800, 7
  croatia_2010_1900, 8
  croatia_employment_aggregation, 10
  employment_metadata, 12
  germany_1990, 15
  metadata, 27
  netherlands_2006, 29
  primary_inputs, 32
  uk_2010_data, 35
  uk_test_results, 37
*Topic effects,
  uk_test_results, 37
*Topic employment
  employment_metadata, 12
*Topic employment
  croatia_employment_2013, 9
  croatia_employment_aggregation, 10
*Topic import
  croatia_2010_1900, 8
*Topic input-output
  croatia_2010_1700, 6
  croatia_2010_1800, 7
  croatia_2010_1900, 8
  croatia_employment_aggregation, 10
  employment_metadata, 12
  germany_1990, 15
  metadata, 27
  metadata_uk_2010, 27
  netherlands_2006, 29
  primary_inputs, 32
  uk_2010_data, 35
  uk_test_results, 37
*Topic inputs
  primary_inputs, 32
*Topic metadata,
  croatia_employment_aggregation, 10
  employment_metadata, 12
  metadata, 27
  metadata_uk_2010, 27
  netherlands_2006, 29
  primary_inputs, 32
  uk_2010_data, 35
  uk_test_results, 37
*Topic multipliers,
  uk_test_results, 37
*Topic table,
INDEX

*Topic vocabulary, employment_metadata, 12
*Topic vocabulary, metadata, 27

backward_linkages, 3
check_digits, 3
coefficient_matrix_create, 4, 18
conforming_vector_create, 5
direct_effects_create, 10
employment_get, 11
employment_metadata, 12
empty_remove, 13
equation_solve, 13, 28
forward_linkages, 14
germany_1990, 15, 29
household_column_find, 16
household_column_get, 16
indirect_effects_create, 17
input_coefficient_matrix_create, 17, 25, 26, 31
input_flow_get, 18
input_indicator_create, 10, 17, 19, 20, 28
input_multipliers_create, 20
iottable_get, 4, 18, 19, 21, 22, 24, 30–32
iottable_year_get, 23
iottables_download, 21, 22–24
is_html_output, 24
is_latex_output, 25
leontieff_inverse_create, 3, 10, 17, 20, 25, 26, 28
leontieff_matrix_create, 26
matrix_round, 26
metadata, 27
metadata_uk_2010, 27
multiplier_create, 13, 28
netherlands_2006, 29
non_zero_columns_find, 30
output_coefficient_matrix_create, 14, 25, 26, 30
output_get, 31
output_multiplier_create, 31
primary_input_get, 31, 32
primary_inputs, 32
quadrant_separator_find, 33
round_table, 33
supplementary_add, 34
total_tax_add, 34
uk_2010_data, 35
uk_2010_get, 36
uk_2010_results_get, 36
uk_test_results, 37