Package 'icio'

September 7, 2026

Version 1.0.0

Title Global Value Chain Decomposition of Inter-Country Input-Output Tables

Description Four global value chain (GVC) decompositions of gross exports from inter-country input-output tables are implemented. The Leontief decomposition derives the value added origin of exports by country and industry, as in Hummels, Ishii and Yi (2001) <doi:10.1016/S0022-1996(00)00093-3>. The Koopman, Wang and Wei (2014) <doi:10.1257/aer.104.2.459> decomposition splits country-level exports into 9 value added components, and the Wang, Wei and Zhu (2013) <doi:10.3386/w19677> decomposition splits bilateral exports into 16 value added components. The Borin and Mancini (2019) <doi:10.1596/1813-9450-8804> decomposition splits country-, sector- or bilateral-level exports into up to 13 value added and GVC components, and also provides a corrected version of the (biased) Koopman-Wang-Wei decomposition. It is the recommended method and reproduces the 'icio' command for 'Stata' described in Belotti, Borin and Mancini (2021) <doi:10.1177/1536867X211045573>.

Maintainer Sebastian Krantz <sebastian.krantz@graduateinstitute.ch>

Depends R (>= 3.5.0)

Imports data.table, matrixStats

License GPL-3

URL https://sebkrantz.github.io/icio/,
https://github.com/SebKrantz/icio

BugReports https://github.com/SebKrantz/icio/issues

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation yes

Config/roxygen2/version 8.0.0

Author Sebastian Krantz [aut, cre] (ORCID:
<https://orcid.org/0000-0001-6212-5229>),
Bastiaan Quast [aut] (ORCID: <https://orcid.org/0000-0002-2951-3577>),
Fei Wang [aut],
Victor Stolzenburg [aut],
Oliver Reiter [ctb]

Repository CRAN

Date/Publication 2026-09-07 10:20:02 UTC

Contents

`icio-package` *Global Value Chain Decomposition of Inter-Country Input-Output Tables*

Description

Four global value chain (GVC) decompositions are implemented. The Leontief decomposition derives the value added origin of exports by country and industry as in Hummels, Ishii and Yi (2001). The Koopman, Wang and Wei (2014) decomposition splits country-level exports into 9 value added components, and the Wang, Wei and Zhu (2013) decomposition splits bilateral exports into 16 value added components. The Borin and Mancini (2019) decomposition splits country-, sector- or bilateral-level exports into up to 13 value added and GVC components, and also provides a corrected version of the (biased) KWW decomposition. It is the recommended method and reproduces the Stata `icio` command.

Contents

Functions to load an ICIO table and create an 'icio' object

`load_icio()`
`load_icio_csv()`

Functions to perform GVC decompositions on an 'icio' object

`bm()`
`leontief()`
`kww()`
`wwz()`

Interface function dispatching on the method, also for lists of 'icio' objects (e.g. several years)

`decomp()`

Function to obtain KWW decomposition from WWZ decomposition

`wwz2kww()`

Example ICIO data

`data("leather")`

Note

`icio` is derived from the CRAN package `decompr` (Quast and Kummritz 2015), of which the author was a co-author and which is no longer maintained. `bm()` is the R counterpart of `decompose()` in the Julia package GlobalValueChains.jl.

Author(s)

Sebastian Krantz <sebastian.krantz@graduateinstitute.ch>
Bastiaan Quast
Fei Wang
Victor Stolzenburg

References

Hummels, D., Ishii, J., & Yi, K. M. (2001). The nature and growth of vertical specialization in world trade. *Journal of international Economics, 54*(1), 75-96.

Koopman, R., Wang, Z., & Wei, S. J. (2014). Tracing value-added and double counting in gross exports. *American Economic Review, 104*(2), 459-94.

Wang, Zhi, Shang-Jin Wei, and Kunfu Zhu (2013). Quantifying international production sharing at the bilateral and sector levels (No. w19677). *National Bureau of Economic Research*.

Borin, A., & Mancini, M. (2019). Measuring What Matters in Global Value Chains and Value-Added Trade. *World Bank Policy Research Working Paper 8804*.

Belotti, F., Borin, A., & Mancini, M. (2021). icio: Economic analysis with inter-country input-output tables. *The Stata Journal, 21*(3), 708-755.

See Also

https://sebkrantz.github.io/icio/

bm *Borin-Mancini Decomposition of Gross Exports and Imports*

Description

Decomposes gross exports (or imports) into value-added and Global Value Chain (GVC) components following the Borin and Mancini (2019) framework, as implemented in the Stata `icio` command (Belotti, Borin and Mancini 2021). It is the R counterpart of the `decompose()` function in the Julia package `GlobalValueChains.jl`, and operates on an `icio` object created by `load_icio`.

Usage

```
bm(
  x,
  aggregation = c("country", "sector", "bilateral"),
  perspective = c("exporter", "world", "self", "importer"),
  approach = c("source", "sink"),
  flow = c("exports", "imports")
)
```

Arguments

`x`	an object of class `icio` obtained from `load_icio`.
`aggregation`	character. The level of the decomposition: `"country"` (one row per exporting/importing country), `"sector"` (one row per exporting country-industry), or `"bilateral"` (one row per exporting country-industry and importing country for exports, or per importing country and value-added origin for imports). Default `"country"`.
`perspective`	character. The accounting perspective defining the perimeter for double counting: `"exporter"` (exporting-country perimeter, additive across sectors and destinations), `"world"` (world perimeter, "corrected KWW", country level only), `"self"` (the export flow's own perimeter, giving the broader Johnson (2018) / Los et al. (2016) value added $DVA^{\star} \supseteq DVA$; sector and bilateral levels only), or `"importer"` (for `flow = "imports"`). Default `"exporter"`.
`approach`	character. How double-counted items are allocated across shipments: `"source"` (value added recorded the first time it leaves the country of origin) or `"sink"` (the last time). The two coincide at the whole-country exporter perimeter (country level). `"world"` accepts both; `"self"` and imports ignore it. Default `"source"`.
`flow`	character. `"exports"` (default) decomposes gross exports; `"imports"` decomposes a country's gross imports from the importer perspective (Borin and Mancini 2019, eq. 51) into value added (`VA`) and double counting (`DC`).

Details

The supported combinations mirror the Stata `icio` command and `GlobalValueChains.jl`:

flow	aggregation	perspective	approach	terms
exports	country	exporter	source(=sink)	13
exports	country	world	source	9
exports	country	world	sink	9
exports	sector	exporter	source	13
exports	sector	exporter	sink	9
exports	sector	self	-	9
exports	bilateral	exporter	source	13
exports	bilateral	exporter	sink	10 (adds VAXIM)
exports	bilateral	self	-	9
imports	country	importer	-	3 (GIMP VA DC)
imports	bilateral	importer	-	2 (VA DC, by origin)

All terms are in the same units as the input-output table (e.g. millions of USD). The following accounting identities hold for exports: `GEXP` = `DC` + `FC`, `DC` = `DVA` + `DDC`, `FC` = `FVA` + `FDC`, `DVA` = `VAX` + `REF`, and (exporter/source only) `GVC` = `GVCB` + `GVCF` = `GEXP` − `DAVAX` and `GVCB` = `FC` + `DDC`; for imports `GIMP` = `VA` + `DC`.

`GEXP`	Gross exports.
`DC` / `FC`	Domestic / foreign content.
`DVA` / `FVA`	Domestic / foreign value added.
`DDC` / `FDC`	Domestic / foreign double counting.
`VAX`	Domestic value added absorbed abroad (Johnson and Noguera 2012).
`REF`	Reflection: domestic value added returning home.
`DAVAX`	Domestic value added directly absorbed by the importer (source approach).
`VAXIM`	Domestic value added absorbed by the direct importer, incl. re-processing (sink approach; `DAVAX` $\leq$ `VAXIM` $\leq$ `VA`
`GVC`	GVC-related trade (value added crossing more than one border).
`GVCB` / `GVCF`	Backward / forward GVC participation.
`GIMP`	Gross imports (= `VA` + `DC`).
`VA` / `DC`	Value added / double counting in imports (by origin at the bilateral level).

The exporter / source decomposition is additive: the `"sector"` result is the sum of the `"bilateral"` result over importers, and the `"country"` result is the sum of the `"sector"` result over industries. The `"sink"` approach shares the domestic content `DC` and foreign content `FC` with `"source"` at every cell; only the value-added vs double-counted split differs. The `"self"` perimeter draws the boundary at the export flow itself, so `DVA` (there $DVA^{\star}$) is weakly larger than under either exporter approach.

Value

A `data.table` with one row per unit and one column per value-added term, preceded by factor identifier columns: `Exporting_Country` (country exports); `Exporting_Country`, `Exporting_Industry` (sector); `Exporting_Country`, `Exporting_Industry`, `Importing_Country` (bilateral exports); `Importing_Country` (country imports); or `Importing_Country`, `Origin_Country` (bilateral imports). The attribute `"decomposition"` is set to `"bm"`.

Author(s)

Sebastian Krantz

References

Borin, A. and Mancini, M. (2019). Measuring What Matters in Global Value Chains and Value-Added Trade. *World Bank Policy Research Working Paper 8804*.

Belotti, F., Borin, A. and Mancini, M. (2021). icio: Economic analysis with intercountry input-output tables. *The Stata Journal, 21*(3), 708-755.

See Also

kww, wwz, leontief, icio-package

Examples

```
# Load example data and create an 'icio' object
data(leather)
dec <- load_icio(leather)

# Country-level decomposition (exporter perspective, source approach; 13 terms)
bm(dec)

# Country-level "corrected KWW" (world perspective, sink approach; 9 terms)
bm(dec, perspective = "world", approach = "sink")

# Sector- and bilateral-sector-level decompositions
bm(dec, aggregation = "sector")
bm(dec, aggregation = "bilateral", approach = "sink")   # adds VAXIM

# Self (own-flow) perimeter, and the importer-perspective import decomposition
bm(dec, aggregation = "bilateral", perspective = "self")
bm(dec, flow = "imports")
```

---

decomp *Run a GVC Decomposition*

Description

A compact interface to the four decompositions: it dispatches on `method` and, given a list of 'icio' objects (e.g. one per year), runs the decomposition on each and stacks the results.

Usage

```
decomp(x, method = c("bm", "leontief", "kww", "wwz"), ..., idcol = "Label")
```

Arguments

x	an 'icio' class object from load_icio or load_icio_csv, or a (preferably named) list of such objects, e.g. one ICIO table per year.
method	character. The decomposition method: "bm" (the default and recommended method, see bm), "leontief", "kww" or "wwz".
...	further arguments passed to bm, leontief, kww or wwz.
idcol	character. Only used if x is a list: the name of the identifier column prepended to the stacked result. It holds the names of x, or the list indices if x is unnamed. Set to NULL to omit it.

Details

Building an 'icio' object is by far the most expensive step (it involves inverting a GN x GN matrix), so it is done once by load_icio and reused across decompositions. Pass the object to bm, leontief, kww or wwz directly if you prefer.

Value

A data.table - see bm, leontief, kww or wwz for the columns of each decomposition. If x is a list, the results are stacked with rbindlist and prefixed with idcol.

Author(s)

Sebastian Krantz, Bastiaan Quast

References

Hummels, D., Ishii, J., & Yi, K. M. (2001). The nature and growth of vertical specialization in world trade. *Journal of international Economics, 54*(1), 75-96.

Koopman, R., Wang, Z., & Wei, S. J. (2014). Tracing value-added and double counting in gross exports. *American Economic Review, 104*(2), 459-94.

Wang, Zhi, Shang-Jin Wei, and Kunfu Zhu (2013). Quantifying international production sharing at the bilateral and sector levels (No. w19677). *National Bureau of Economic Research*.

Borin, A., & Mancini, M. (2019). Measuring What Matters in Global Value Chains and Value-Added Trade. *World Bank Policy Research Working Paper 8804*.

See Also

load_icio, bm, icio-package

Examples

```
# Load leather example data
data(leather)

# Explore the data
str(leather)
```

```
# Create the 'icio' object
m <- load_icio(leather)

## Decomposing gross exports:

# Borin-Mancini (2019), the recommended method
decomp(m)
decomp(m, aggregation = "bilateral")

# Leontief, Koopman-Wang-Wei and Wang-Wei-Zhu
decomp(m, method = "leontief")
decomp(m, method = "kww")
decomp(m, method = "wwz")

# Multiple tables at once, e.g. one per year, stacked with a 'Year' column
decomp(list(`2015` = m, `2016` = m), idcol = "Year")
```

---

kww *Koopman-Wang-Wei Decomposition of Gross Exports*

Description

This function performs the Koopman-Wang-Wei (2014) decomposition of a countries gross exports into 9 separate value added components.

Usage

```
kww(x)
```

Arguments

x	an object of the class 'icio' obtained from load_icio.

Value

A data.table where a country's gross exports is decomposed into 9 components (columns), as detailed in Figure 1 of the AER paper:

Term	*Description*
DVA_FIN	Domestic VA in final goods exports.
DVA_INT	Domestic VA in intermediate exports absorbed by direct importers (used to produce a locally consumed final g
DVA_INTrex	Domestic VA in intermediate exports reexported to third countries and absorbed there.
RDV_FIN	Domestic VA in intermediate exports that returns home via final imports.
RDV_INT	Domestic VA in intermediate exports that returns home via intermediate imports (used to produce a domestical
DDC	Double counted DVA in intermediate exports (arising from 2-way trade in intermediate goods).
FVA_FIN	Foreign VA in final goods exports.
FVA_INT	Foreign VA in intermediate exports.
FDC	Double counted FVA in intermediate exports (arising from 2-way trade in intermediate goods).

Note

The KWW decomposition is known to be biased. As shown by Borin and Mancini (2019), it systematically underestimates the foreign value added in exports – and correspondingly overstates foreign double counting – because the entire foreign content that the direct importer re-exports to third countries is classified as 'foreign double counted', including the part (value added generated in the importing country and re-exported onwards) that is never recorded as foreign value added in any other flow. KWW also overlooks the bilateral dimension of trade, so it cannot correctly split domestic value added between absorption by the direct importer and by third markets (hence indicators such as DAVAX cannot be derived from it). Borin and Mancini (2019) correct these issues using a sink-based, world-level perspective for the foreign content of exports; this corrected KWW decomposition is available as `bm(x, perspective = "world", approach = "sink")`.

Author(s)

Sebastian Krantz

References

Koopman, R., Wang, Z., & Wei, S. J. (2014). Tracing value-added and double counting in gross exports. *American Economic Review, 104*(2), 459-94.

Borin, A., & Mancini, M. (2019). Measuring What Matters in Global Value Chains and Value-Added Trade. *World Bank Policy Research Working Paper 8804*.

See Also

`bm`, `wwz`, `wwz2kww`, `icio-package`

Examples

```
# Load example data
data(leather)

# Create intermediate object (class 'icio')
m <- load_icio(leather)

# Perform the KWW decomposition
kww(m)
```

leather *Leather Example ICIO Data*

Description

An example 3 x 3 ICIO table describing a GVC for leather products with industries 'Agriculture', 'Textile and Leather' and 'Transport Equipment' for the countries 'Argentina', 'Turkey' and 'Germany'.

Usage

```
data("leather")
```

Format

A list of class 'iot' with the following elements:

`inter` 9 x 9 input output matrix where each column gives the value of inputs supplied to the corresponding country-industry by each row country-industry.

`final` 9 x 3 final demand matrix showing the final demand in each country (column) for each country-industry's (rows) produce.

`countries` character vector of country names (matching columns of `final`).

`industries` character vector of industries, such that `as.vector(t(outer(countries, industries, FUN = paste, sep = ".")))` generates the row- and column-names of `inter` and the row-names of `final`.

`out` A vector of gross country-industry output. In a complete productive system it should be equal to `rowSums(inter) + rowSums(final)`.

See Also

icio-package

leontief *Leontief Decomposition*

Description

The Leontief decomposition of gross flows (exports, final demand, output) into their value added origins.

Usage

```
leontief(x, post = c("exports", "output", "final_demand", "none"), long = TRUE)
```

Arguments

x	an object of class 'icio'.
post	post-multiply the value added multiplier matrix [$VB = V(I - A)^{-1}$] with something to deduce the value added origins thereof. The default is "exports" $VAE = V(I-A)^{-1}E$, where E is a diagonal matrix with exports along the diagonal yielding the country-industry level sources of value added (rows) for each using (column) country-industry; similarly for "output". Option "final_demand" computes value added origins of final demand by source country-industry and importing country, by computing $VAY = V(I - A)^{-1}Y$ where Y is the corresponding GN x G matrix contained in x. Option "none" just returns VB which gives the value added shares.
long	logical. Transform the output data into a long (tidy) data set or not, default is TRUE.

Details

The Leontief decomposition is obtained by pre-multiplying the flow measure (e.g. exports) with the value added multiplier matrix [$VB = V(I - A)^{-1}$], obtained by pre-multiplying the Leontief Inverse matrix [$B = (I - A)^{-1}$] with a diagonal matrix [V] containing the direct value added share in each industries output.

V is obtained as `diag(v / o)` where `o` is total industry output. `v` is either supplied to `load_icio` or computed as `o - colSums(x)` with `x` the raw IO matrix. If `o` is not supplied to `load_icio`, it is computed as `rowSums(x) + rowSums(y)` where `y` is the matrix of final demands. If both `o` and `v` are not supplied to `load_icio`, this is equivalent to computing V as `diag(1 - colSums(A))`, with A is the row-normalized IO matrix also used to compute the Leontief Inverse [B].

Value

If `long = TRUE` a molten `data.table` containing the elements of the decomposed flows matrix in the final column, preceded by several identifier columns. If `long = FALSE` the decomposed flows matrix is simply returned.

Author(s)

Bastiaan Quast

References

Leontief, W. (Ed.). (1986). Input-output economics. *Oxford University Press*.

Hummels, D., Ishii, J., & Yi, K. M. (2001). The nature and growth of vertical specialization in world trade. *Journal of international Economics, 54*(1), 75-96.

Wang, Zhi, Shang-Jin Wei, and Kunfu Zhu (2013). Quantifying international production sharing at the bilateral and sector levels (No. w19677). *National Bureau of Economic Research*.

See Also

`bm`, `kww`, `wwz`, `icio-package`

Examples

```
# Load example data
data(leather)

# Create intermediate object (class 'icio')
m <- load_icio(leather)

# Perform the Leontief decomposition of each country-industries
# exports into their value added origins by country-industry
leontief(m)
```

`load_icio` *Load an Inter-Country Input-Output Table*

Description

Reads the raw ICIO matrices and precomputes everything the decompositions need, returning an 'icio' class object. This is the entry point for all decompositions.

Usage

```
load_icio(
  inter,
  final,
  countries,
  industries,
  output = NULL,
  va = NULL,
  null_inventory = FALSE
)
```

Arguments

`inter`	intermediate demand table supplied as a numeric matrix of dimensions GN x GN (G = no. of countries, N = no. of industries). Both rows and columns should be arranged first by country, then by industry (e.g. C1I1, C1I2, ..., C2I1, C2I2, ...) and should match (symmetry), such that rows and columns refer to the same country-industries. Alternatively, an Input-Output Table object of class 'iot' can be passed here - a list with elements 'inter', 'final', 'countries', 'industries' and (optionally) 'output' - in which case the remaining table arguments are taken from it. See `leather`.
`final`	final demand table supplied as a numeric matrix of dimensions GN x GM (M = no. of final demand categories recorded for each country). The rows of `final` need to match the rows of `inter`, and the columns should also be arranged first by country, then by final demand category (e.g. C1FD1, C1FD2, ..., C2FD1, C2FD2, ...) with the order of the countries the same as in `inter`.

countries	character. A vector of country or region names of length G, arranged in the same order as they occur in the rows and columns of inter and final.
industries	character. A vector of industry names of length N, arranged in the same order as they occur in the rows and columns of inter and the rows of final.
output	numeric. A vector of gross outputs for each country-industry matching the rows of inter and final. If not provided it will be computed as rowSums(inter) + rowSums(final).
va	numeric. A vector of value added for each country-industry matching the columns of inter. If not provided it will be computed as output - colSums(inter), which is what the Stata icio command does.
null_inventory	logical. TRUE sets the inventory (last final demand category for each country) to zero.

Details

Only A and B are dense GN x GN matrices. The masked and block-diagonal variants of them that the decompositions require (A with the domestic blocks zeroed, the domestic and foreign parts of B) are derived on the fly by the functions that need them, being either recoverable from A and B in a few operations or, in the case of the domestic Leontief inverse, block-diagonal and thus $1 - 1/G$ structural zeros.

Value added defaults to the column residual of the table (output - colSums(inter)), which reproduces the Stata icio command exactly. Supplying a va that differs from it makes the column sums of VB deviate from 1, so identities such as GEXP = DC + FC may no longer hold exactly - faithful to the supplied data.

Value

An 'icio' class object - a list with the following elements:

A	Input coefficients matrix (inter column-normalized by output), including the domestic blocks.
B	Leontief Inverse matrix $(I - A)^{-1}$.
Lb	List of G domestic (local) Leontief Inverse blocks $(I - A_{gg})^{-1}$, one N x N matrix per country.
E	Total Exports (output of each country-industry servicing foreign production or foreign final demand).
ESR	Total Exports by destination country.
Vc	Value added content of output (va / output).
G	Number of countries.
N	Number of industries.
GN	Number of country-industries.
k	Vector of country names.
i	Vector of industry names.
X	Total Output (= output).
Y	Total Final Demand by destination country.
Yd	Domestic Final Demand.
Ym	Foreign Final Demand.

The country-industry names identifying the rows and columns are available as names(x$Vc) or dimnames(x$B)[[1L]].

Author(s)

Sebastian Krantz, Bastiaan Quast. Adapted from code by Fei Wang.

See Also

load_icio_csv, decomp, bm, icio-package

Examples

```
# Load example data
data(leather)

# Create intermediate object (class 'icio') from an 'iot' object
m <- load_icio(leather)

# Equivalent: passing the matrices directly
m <- load_icio(leather$inter, leather$final, leather$countries, leather$industries)

# Examine the object
str(m)
```

---

load_icio_csv *Load an ICIO Table from the 'icio' CSV Format*

Description

Reads an Inter-Country Input-Output table from the CSV format used by the Stata `icio` command and returns an 'icio' class object, ready for `decomp` and the decomposition functions.

Usage

```
load_icio_csv(
  table,
  countries,
  industries = NULL,
  output = NULL,
  va = NULL,
  ...
)
```

Arguments

`table`	character. Path to a headerless `GN x (GN + G)` CSV file holding the matrix `[inter | final]`: the first GN columns are the intermediate transactions, the last G columns the final demand (one aggregated column per country). Read with `fread`.

countries	character. Either a vector of G country codes, or the path to a headerless one-column CSV file containing them (the country list file of the Stata `icio` command).
industries	character. Either a vector of N industry codes, the path to a headerless one-column CSV file containing them, or `NULL` (the default) to generate `"sector1"`, `...`, `"sectorN"`. The number of industries is inferred as `N = GN / G`.
output	numeric. Optional vector of gross outputs, see `load_icio`.
va	numeric. Optional vector of value added, see `load_icio`. The default is the column residual of the table, which reproduces the Stata `icio` command.
...	further arguments passed to `fread`.

Value

An 'icio' class object, see `load_icio`.

Author(s)

Sebastian Krantz

See Also

`load_icio`, `decomp`, `icio-package`

Examples

```
# Write the example table out in the 'icio' CSV format: a headerless
# GN x (GN + G) matrix [inter | final], plus a one-column country list
data(leather)
tbl <- tempfile(fileext = ".csv")
cls <- tempfile(fileext = ".csv")
write.table(cbind(leather$inter, leather$final), tbl,
            sep = ",", row.names = FALSE, col.names = FALSE)
write.table(leather$countries, cls,
            sep = ",", row.names = FALSE, col.names = FALSE, quote = FALSE)

# Read it back: industries default to "sector1", ..., "sectorN"
m <- load_icio_csv(tbl, cls)
str(m$i)

# Supplying the real industry codes so they appear in the results
m <- load_icio_csv(tbl, cls, industries = leather$industries)
decomp(m, aggregation = "sector")

# The country list may also be given directly as a character vector
m <- load_icio_csv(tbl, leather$countries, industries = leather$industries)

unlink(c(tbl, cls))
```

wwz *Wang-Wei-Zhu Decomposition of Gross Exports*

Description

This function performs the Wang-Wei-Zhu decomposition of country-sector level gross exports into 16 value added components by importing country.

Usage

```
wwz(x, verbose = FALSE)
```

Arguments

x	an object of the class 'icio' obtained from load_icio.
verbose	logical, should timings of the calculation be displayed? Default is FALSE

Details

Adapted from code by Fei Wang.

Value

A long-format data.table with one row per (exporting country-industry, importing country) pair and columns Exporting_Country, Exporting_Industry, Importing_Country followed by the 16 decomposition terms (as detailed in Table E1 in the appendix of Wang, Wei & Zhu 2013) and diagnostic items:

Term	*Description*
DVA_FIN	Domestic VA in final goods exports.
DVA_INT	Domestic VA in intermediate exports used by direct importer to produce domestic final goods consur
DVA_INTrexI1	Domestic VA in intermediate exports used by the direct importer to produce intermediate exports for
DVA_INTrexF	Domestic VA in intermediate exports used by the direct importer to produce final goods exports to th
DVA_INTrexI2	Domestic VA in intermediate exports used by the direct importer to produce intermediate exports to t
RDV_INT	Domestic VA in intermediate exports that returns via intermediate imports (i.e. is used to produce a l
RDV_FIN	Domestic VA in intermediate exports that returns home via final goods imports from the direct impor
RDV_FIN2	Domestic VA in intermediate exports that returns home via final goods imports from third countries.
OVA_FIN	Third countries' VA in final goods exports.
MVA_FIN	Direct importer's VA in final goods exports.
OVA_INT	Third countries' VA in intermediate exports.
MVA_INT	Direct importer's VA in intermediate exports.
DDC_FIN	Double counted domestic VA used to produce final goods exports.
DDC_INT	Double counted domestic VA used to produce intermediate exports.
ODC	Double counted third countries' VA in home country's exports production.
MDC	Double counted direct importer's VA in home country's exports production.
Diagnostic Item	*Description*
texp	Total exports (matrix ESR from load_icio).

texpint	Exports for intermediate production (matrix Eint from load_icio).
texpfd	Exports for final demand (matrix Efd from load_icio).
texpdiff	Difference between total exports and the sum of the 16 terms.
texpdiffpercent	... in percent of total exports.
texpfddiff	Difference between final exports and the sum of DVA_FIN, OVA_FIN and MVA_FIN.
texpfddiffpercent	... in percent of final exports.
texpintdiff	Difference between intermediate exports and the sum of all remaining terms.
texpintdiffpercent	... in percent of intermediate exports.
DViX_Fsr	DVA embodied in gross exports based on forward linkage.

Author(s)

Bastiaan Quast

References

Wang, Zhi, Shang-Jin Wei, and Kunfu Zhu (2013). Quantifying international production sharing at the bilateral and sector levels (No. w19677). *National Bureau of Economic Research*.

See Also

bm, kww, wwz2kww, icio-package

Examples

```
# Load example data
data(leather)

# Create intermediate object (class 'icio')
m <- load_icio(leather)

# Perform the WWZ decomposition
wwz(m)
```

---

wwz2kww *Koopman-Wang-Wei from Wang-Wei-Zhu Decomposition*

Description

This function by default returns a disaggregated version of the the Koopman-Wang-Wei (KWW) decomposition breaking up sector-level gross exports into 9 value added terms, from an already computed and more detailed (16 term) Wang-Wei-Zhu decomposition of sector-level gross exports. An aggregation option also allows obtaining the aggregate KWW decomposition.

Usage

```
wwz2kww(x, aggregate = FALSE)
```

Arguments

x	a data.table with the WWZ decomposition obtained from wwz. Alternatively an 'icio' class object from load_icio can be supplied, which will toggle calling wwz() first.
aggregate	logical. TRUE aggregates the KWW decomposition to the country level, giving exactly the same output as kww. FALSE maintains the sector level decomposition in KWW format.

Details

The mapping of the 16 terms in the WWZ decomposition to the 9 terms in the KWW decomposition is provided in table E2 in the appendix of the WWZ (2013) paper. The table is reproduced here using the term naming conventions followed in this package.

WWZ Terms	*KWW Term*	*Description*
DVA_FIN	DVA_FIN	Domestic VA in final goods exports.
DVA_INT, DVA_INTrexI1	DVA_INT	Domestic VA in intermediate exports absorbed by direct importers. WWZ sep
DVA_INTrexF, DVA_INTrexI2	DVA_INTrex	Domestic VA in intermediate exports reexported to third countries and absorb
RDV_FIN, RDV_FIN2	RDV_FIN	Domestic VA in intermediate exports that returns home via final imports. WW
RDV_INT	RDV_INT	Domestic VA in intermediate exports that returns via intermediate imports (us
DDC_FIN, DDC_INT	DDC	Double counted domestic VA in gross exports. WWZ separates double counti
MVA_FIN, OVA_FIN	FVA_FIN	Foreign VA in final goods exports. WWZ separates FVA from direct importer
MVA_INT, OVA_INT	FVA_INT	Foreign VA in intermediate exports. WWZ separates FVA from direct import
MDC, ODC	FDC	Double counted foreign VA in gross exports. WWZ separates FDC from direc

Value

A data.table with exports decomposed into 9 components (columns), see the table above and kww for a shorter description of the 9 terms.

Note

If both WWZ and KWW decompositions are required, it is computationally more efficient to call wwz2kww(x, aggregate = TRUE) on an already computed WWZ decomposition, than to call kww on an 'icio' object.

Author(s)

Sebastian Krantz

References

Koopman, R., Wang, Z., & Wei, S. J. (2014). Tracing value-added and double counting in gross exports. *American Economic Review, 104*(2), 459-94.

Wang, Zhi, Shang-Jin Wei, and Kunfu Zhu (2013). Quantifying international production sharing at the bilateral and sector levels (No. w19677). *National Bureau of Economic Research*.

See Also

wwz, kww, icio-package

Examples

```
# Load example data
data(leather)

# Create intermediate object (class 'icio')
m <- load_icio(leather)

# Perform the WWZ decomposition
WWZ <- wwz(m)

# Obtain a disaggregated KWW decomposition
KWW <- wwz2kww(WWZ)

# Aggregate KWW
wwz2kww(WWZ, aggregate = TRUE)

# Same as running KWW directly, but the former is more efficient
# if we already have the WWZ
kww(m)
```

Index