

Package: iopspackage (via r-universe)

August 19, 2024

Type Package

Title IO-PS Framework Package

Version 2.1.0

Date 2023-12-17

Description A developmental R tool related to the input-output product space (IO-PS). The package requires two compulsory user inputs (raw CEPPi BACI trade data, and any acceptable ISO country code) and has 4 optional user inputs (a value chain map, chosen complexity method, number of iterations to be performed, and a trade digit level). Various metrics are calculated, such as Economic- and Product complexity, distance, opportunity gain, and inequality metrics, to facilitate better decision making regarding industrial policy making.

Imports dplyr, economiccomplexity, readxl, tidyr, openxlsx, usethis

Suggests testthat

LazyData true

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 3.5.0)

NeedsCompilation no

Author Stellenbosch University Industrial Policy Research Group [aut, cph], Christiaan Pieterse [aut, cre, trl] (<<https://orcid.org/0009-0006-4067-3797>>), Wouter Bam [aut, ths] (<<https://orcid.org/0000-0002-0912-0225>>)

Maintainer Christiaan Pieterse <pietie.cjp.1908@gmail.com>

Date/Publication 2024-01-09 09:10:02 UTC

Additional_repositories <https://cranhaven.r-universe.dev>

Repository <https://cranhaven.r-universe.dev>

RemoteUrl <https://github.com/cranhaven/cranhaven.r-universe.dev>

RemoteRef package/iopspackage

RemoteSha ffb45a4688021bc4d3ceb780e565f4d3cc509fc0

Contents

ExampleTradeData	2
IOPS	3
Index	5

ExampleTradeData	<i>BACI_HS17_Y2020_V202301 data</i>
------------------	-------------------------------------

Description

A dataset containing the trade data for country codes 4 and 710.

Usage

ExampleTradeData

Format

‘ExampleTradeData‘ A data frame with 72097 rows and 6 variables:

- t** Year
- k** Product category, HS 6-digit code
- i** Exporter, ISO 3-digit country code
- j** Importer, ISO 3-digit country code
- v** Value of the trade flow, in thousands current USD
- q** Quantity, in metric tons

Source

http://www.cepii.fr/CEPII/en/bdd_modele/bdd_modele_item.asp?id=37

IOPS

*IOPS***Description**

Takes user inputted trade data, any acceptable ISO country code and industrial value chain mapping to calculate various metrics (Economic- and Product complexity metrics, distance metrics, opportunity gain, and inequality metrics) of a given country in order to facilitate better decision making regarding industrial policymaking.

Usage

```
IOPS(
  CountryCode,
  tradeData,
  ComplexMethod = "eigenvalues",
  iterCompl = 20,
  GVCMapping = NULL,
  tradedigit = 6
)
```

Arguments

CountryCode	(Type: character/integer) Any accepted ISO country code could be used, e.g. "United Kingdom", "GBR", "GB", 828 would all be accepted if the United Kingdom is the desired country. This version of the package uses country_codes_V202201 from the BACI Hs17 2020 dataset, but can be changed in inst/extdata.
tradeData	(Type: csv) Accepts any CEPII BACI trade data. NOTE: tradeData and GVCMapping must be from the same "H" Family, e.g. both are from H3, etc., in order for the program to work correctly.
ComplexMethod	(Type: character) Methods used to calculate complexity measures. Can be any one of these methods: "fitness", "reflections" or "eigenvalues". Defaults to "eigenvalues".
iterCompl	(Type: integer) The number of iterations that the chosen complexity measure must use. Defaults to iterCompl = 20.
GVCMapping	(Type: csv) The desired value chain to be analysed. With Columns "Tiers", "Activity", and "HSCCode". NOTE: tradeData and GVCMapping must be from the same "H" Family, e.g. both are from H3, etc., in order for the program to work correctly.
tradedigit	(Type: integer) Indicate if the raw trade digit summation should be done on a 6- or 4-digit level. Defaults to tradedigit = 6.

Value

A list that constrains ECI, PCI, Opportunity_Gain, distance, density, M_absolute, M_binary, Tier_Results, Product_Category_Results, Product_Results, respectively.

Examples

```
# Create a temporary directory
temp_dir <- tempfile()
dir.create(temp_dir)

# Set the working directory to the temporary directory
old_dir <- setwd(temp_dir)

#Generate example trade data
GeneratedTradeData <- data.frame(
  t = c(2020, 2020, 2020),
  i = c(4, 710, 710),
  j = c(842, 124, 251),
  k = c(842410, 110220, 845210),
  v = c(4.776, 0.088, 0.057),
  q = c(0.025, 0.007, 0.005)
)

# Use your function with generated trade data
IOPS(
  CountryCode = 710,
  tradeData = GeneratedTradeData,
  ComplexMethod = "reflections",
  iterCompl = 2,
  GVCMapping = NULL,
  tradedigit = 6
)

# Use your function with real trade data
IOPS(
  CountryCode = 710,
  tradeData = ExampleTradeData,
  ComplexMethod = "reflections",
  iterCompl = 2,
  GVCMapping = NULL,
  tradedigit = 6
)

# Clean up the temporary directory
setwd(old_dir) # Restore the original working directory
unlink(temp_dir, recursive = TRUE)
```

Index

* **datasets**

ExampleTradeData, [2](#)

ExampleTradeData, [2](#)

IOPS, [3](#)