Package: Avionic24 (via r-universe)

August 22, 2024

Type Package

Title What the Package Does (Title Case)

Version 0.1.0

Author Who wrote it

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line) Use four spaces when indenting paragraphs within the Description.

License What license is it under?

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.11), RcppArmadillo, data.table, tictoc, xlsx, dplyr, stringr, readxl, ggplot2

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository https://aqlt.r-universe.dev

RemoteUrl https://github.com/InseeFrLab/Avionic24

RemoteRef HEAD

RemoteSha b56ec07aa0869c6b61cfc9e31a344c9451086811

Contents

`AddRownamesToFirstCol` *AddRownamesToFirstCol Add Rownames To First Column*

Description

AddRownamesToFirstCol Add Rownames To First Column

Usage

```
AddRownamesToFirstCol(df)
```

Arguments

`df`	dataframe

Value

df data frame

`Agreg_Manuf` *Agreg_Manuf greggation function TypAgreg= Manuf_CT ; Manuf_IP19 ; Manuf_CT_lrwiod ; Manuf_IP19_lrwiod MRIO= LRWIOD ; WIOD; FIGARO Warning : the variable to agreggate must be named "value" Use of StructDocs to manage classifications*

Description

Agreg_Manuf greggation function TypAgreg= Manuf_CT ; Manuf_IP19 ; Manuf_CT_lrwiod ; Manuf_IP19_lrwiod MRIO= LRWIOD ; WIOD; FIGARO Warning : the variable to agreggate must be named "value" Use of StructDocs to manage classifications

Usage

```
Agreg_Manuf(DT, MRIO, Var_To_Agreg, TypAgreg)
```

Arguments

`DT`	datatable
`MRIO`	MRIO object
`Var_To_Agreg`	text variable
`TypAgreg`	text

Value

dl data long

`AjoutPRBR` *AjoutPRBR Add PR and BR to help transposing tables to wide format (eg. matrix like A or L)*

Description

AjoutPRBR Add PR and BR to help transposing tables to wide format (eg. matrix like A or L)

Usage

```
AjoutPRBR(dl)
```

Arguments

dl	long data

Value

dl data long

Examples

```
## Not run:  f_dt <- AjoutPRBR(interm)
y_dt <- AjoutPRBR(y_dt)
## End(Not run)
```

`Attr_TxSimu_HRM_100MoE` *Attr_TxSimu_HRM_100MoE Function to affect the right Why 100Mo ? / because as we exchange imports with production we need to have enough imports. –> Linear Model => results*10 with no problem. TypeTx = "VA" or "PROD"*

Description

Attr_TxSimu_HRM_100MoE Function to affect the right Why 100Mo ? / because as we exchange imports with production we need to have enough imports. –> Linear Model => results*10 with no problem. TypeTx = "VA" or "PROD"

Usage

```
Attr_TxSimu_HRM_100MoE(
  Produit,
  pays,
  annee,
  TypeTx = "VA",
  OptSourceRDS = "XXXXX",
  OptMRIOlong = NULL
)
```

Arguments

Produit	text industry
pays	text country
annee	year
TypeTx	text
OptSourceRDS	binary
OptMRIOlong	binary

Value

numeric value

Autarky — *Autarky Function for calculating an autarky situation from a MRIO which is the starting world economy situation The MRIO has already undergone a CompoMRIO, and the Save option does a saveRDS in "MRIO_Autarky.rds".*

Description

Autarky Function for calculating an autarky situation from a MRIO which is the starting world economy situation The MRIO has already undergone a CompoMRIO, and the Save option does a saveRDS in "MRIO_Autarky.rds".

Usage

```
Autarky(dtdl, Optdl = FALSE, OptSaveRDS = FALSE, OptBaseDT = FALSE)
```

Arguments

dtdl	datatable
Optdl	binary
OptSaveRDS	binary
OptBaseDT	binary

Value

dl data long

`av_create_FIGAROixi_2022`

av_create_FIGAROixi_2022

Description

Figaro 2022 release, CSV flat format : link : https://ec.europa.eu/eurostat/web/esa-supply-use-input-tables/database#CSV

Usage

```
av_create_FIGAROixi_2022(Path_FIG22ixi, Path_out, OptAnnual = FALSE)
```

Arguments

`Path_FIG22ixi`	path of FIGARO ixi CSV flat format files
`Path_out`	path to save normalized data
`OptAnnual`	option to keep one files by year instead of one file with all years. Can provide a degraded solution if a single database requires too much memory.

Value

Nothing. Only save data into normalized format.

Examples

```
## Not run:
Path_FIG22ixi <- paste0(PathTemp, "ixi/")
Path_out <- PathTest
av_create_FIGAROixi_2022(Path_FIG22ixi, Path_out, OptAnnual = TRUE)
av_create_FIGAROixi_2022(Path_FIG22ixi, Path_out, OptAnnual = FALSE)
testAll <- readRDS(paste0(Path_out, "/BDn_FIG.rds"))
test2015 <- readRDS(paste0(Path_out, "/BDn_FIG_", 2015, ".rds"))
## End(Not run)
```

av_create_ICIO_2021 *av_create_ICIO_2021 ICIO 2021 release, CSV flat format : link : https://www.oecd.org/industry/ind/inter-country-input-output-tables.htm OECD (2021), OECD Inter-Country Input-Output Database, http://oe.cd/icio*

Description

av_create_ICIO_2021 ICIO 2021 release, CSV flat format : link : https://www.oecd.org/industry/ind/inter-country-input-output-tables.htm OECD (2021), OECD Inter-Country Input-Output Database, http://oe.cd/icio

Usage

```
av_create_ICIO_2021(
  Path_ICIO,
  Path_out,
  OptAnnual = FALSE,
  ConvertToEuro = TRUE
)
```

Arguments

Path_ICIO	path of ICIO flat CSV format files
Path_out	path to save normalized data
OptAnnual	option to keep one files by year instead of one file with all years. Can provide a degraded solution if a single database requires too much memory.
ConvertToEuro	Option to convert Dollars to Euros

Value

Nothing. Only save data into normalized format.

Examples

```
## Not run:  Path_ICIO <- PathTemp
Path_out <- PathTest
av_create_ICIO_2021(Path_ICIO, Path_out, OptAnnual = TRUE, ConvertToEuro = TRUE)
test1998Euro <- readRDS(paste0(Path_out, "/BDn_ICIO_", 1998, ".rds"))
av_create_ICIO_2021(Path_ICIO, Path_out, OptAnnual = TRUE, ConvertToEuro = FALSE)
test1998Dolls <- readRDS(paste0(Path_out, "/BDn_ICIO_", 1998, ".rds"))
av_create_ICIO_2021(Path_ICIO, Path_out, OptAnnual = FALSE, ConvertToEuro = TRUE)
testEuro <- readRDS(paste0(Path_out, "/BDn_ICIO.rds"))
av_create_ICIO_2021(Path_ICIO, Path_out, OptAnnual = FALSE, ConvertToEuro = FALSE)
testDolls <- readRDS(paste0(Path_out, "/BDn_ICIO.rds"))
head(testEuro)
head(testDolls)
## End(Not run)
```

av_create_LRWIOD_2022	*av_create_LRWIOD_2022 LR-WIOD 2022 release : link : https://www.rug.nl/ggdc/valuechain/long-run-wiod?lang=en License and funding : Long-run WIOD is licensed under a Creative Commons Attribution 4.0 International License . The construction of the Long-run WIOD was supported by the Dutch Science Foundation (NWO) [grant number 453-14-012].*

Description

av_create_LRWIOD_2022 LR-WIOD 2022 release : link : https://www.rug.nl/ggdc/valuechain/long-run-wiod?lang=en License and funding : Long-run WIOD is licensed under a Creative Commons Attribution 4.0 International License . The construction of the Long-run WIOD was supported by the Dutch Science Foundation (NWO) [grant number 453-14-012].

Usage

```
av_create_LRWIOD_2022(Path_WIODLR, Path_out, ConvertToEuro = TRUE)
```

Arguments

Path_WIODLR	path of Long-run WIOD flat CSV format files
Path_out	path to save normalized data
ConvertToEuro	Option to convert Dollars to Euros

Value

Nothing. Only save data into normalized format.

Examples

```
## Not run:  Path_WIODLR <- PathTemp
Path_out <- PathTest
av_create_LRWIOD_2022(Path_WIODLR, Path_out, ConvertToEuro = TRUE)
testEuroDolls <- readRDS(paste0(Path_out, "/BDn_LR_WIOD.rds"))
Euros <- testEuroDolls
av_create_LRWIOD_2022(Path_WIODLR, Path_out, ConvertToEuro = FALSE)
testEuroDolls <- readRDS(paste0(Path_out, "/BDn_LR_WIOD.rds"))
dolls <- testEuroDolls
## End(Not run)
```

av_create_WIOD_2016	*av_create_WIOD_2016 WIOD 2016 release, RData format : link : https://www.rug.nl/ggdc/valuechain/wiod/wiod-2016-release?lang=en Timmer, M. P., Dietzenbacher, E., Los, B., Stehrer, R. and de Vries, G. J. (2015), "An Illustrated User Guide to the World Input–Output Database: the Case of Global Automotive Production" , Review of International Economics., 23: 575–605*

Description

av_create_WIOD_2016 WIOD 2016 release, RData format : link : https://www.rug.nl/ggdc/valuechain/wiod/wiod-2016-release?lang=en Timmer, M. P., Dietzenbacher, E., Los, B., Stehrer, R. and de Vries, G. J. (2015), "An Illustrated User Guide to the World Input–Output Database: the Case of Global Automotive Production" , Review of International Economics., 23: 575–605

Usage

```
av_create_WIOD_2016(
  Path_WIOD,
  Path_out,
  OptAnnual = FALSE,
  ConvertToEuro = TRUE
)
```

Arguments

Path_WIOD	path of WIOD RData format files
Path_out	path to save normalized data
OptAnnual	option to keep one files by year instead of one file with all years. Can provide a degraded solution if a single database requires too much memory.
ConvertToEuro	Option to convert Dollars to Euros

Value

Nothing. Only save data into normalized format.

Examples

```
## Not run:  Path_WIOD <- PathTemp
Path_out <- PathTest
av_create_WIOD_2016(Path_WIOD, Path_out, OptAnnual = TRUE, ConvertToEuro = TRUE)
test2015Euro <- readRDS(paste0(Path_out, "/BDn_WIOD_", 2013, ".rds"))
av_create_WIOD_2016(Path_WIOD, Path_out, OptAnnual = TRUE, ConvertToEuro = FALSE)
test2015Dolls <- readRDS(paste0(Path_out, "/BDn_WIOD_", 2013, ".rds"))
av_create_WIOD_2016(Path_WIOD, Path_out, OptAnnual = FALSE, ConvertToEuro = TRUE)
testEuro <- readRDS(paste0(Path_out, "/BDn_WIOD.rds"))
av_create_WIOD_2016(Path_WIOD, Path_out, OptAnnual = FALSE, ConvertToEuro = FALSE)
```

```
testDolls <- readRDS(paste0(Path_out, "/BDn_WIOD.rds"))
## End(Not run)
```

---

av_Diff_SPA *av_Diff_SPA Function for calculating the difference of Stuctural Path Analysis (SPA) for variant analysis The MRIO can only be in long format here TypeSPA="VA" or "Emi" ListThres = GenThres=,Thres_L2_1=,Thres_L3_1=,Thres_L3_2=,Thres_L3_3,Thres_L3_4=,Thres_L4_1=,Thres Fitted FIGARO VA : ListThres = GenThres=0.001,Thres_L2_1=0.001,Thres_L3_1=0.001,Thres_L3_2=0.001,Thres_L3_3=0.00001,T Fitted FIGARO Emi : ListThres = GenThres=0.001,Thres_L2_1=0.0001,Thres_L3_1=0.0001,Thres_L3_2=0.00001,Thres_L3_3=0.000*

Description

av_Diff_SPA Function for calculating the difference of Stuctural Path Analysis (SPA) for variant analysis The MRIO can only be in long format here TypeSPA="VA" or "Emi" ListThres = GenThres=,Thres_L2_1=,Thres_L3_1=,Thres_L3_2=,Thres_L3_3,Thres_L3_4=,Thres_L4_1=,Thres_L4_2=,Thres_L4_3,Thres
Fitted FIGARO VA : ListThres = GenThres=0.001,Thres_L2_1=0.001,Thres_L3_1=0.001,Thres_L3_2=0.001,Thres_L3_3=
Fitted FIGARO Emi : ListThres = GenThres=0.001,Thres_L2_1=0.0001,Thres_L3_1=0.0001,Thres_L3_2=0.00001,Thres_L

Usage

```
av_Diff_SPA(
  MRIOdt1,
  MRIOdt2,
  ListThres,
  TypContenu = "VA",
  TypeMRIO = "FIGARO",
  PathEmi = "",
  TargetCountry = "FRA"
)
```

Arguments

MRIOdt1	datatable
MRIOdt2	datatable
ListThres	list thresholds
TypContenu	text options
TypeMRIO	text
PathEmi	text path
TargetCountry	text country option

Value

dt data table same format as path analysis (SPA)

Examples

```
## Not run:  ListThres = {GenThres=0.001,Thres_L2_1=0.001,Thres_L3_1=0.001,Thres_L3_2=0.001,Thres_L3_3=0.001,Thr
res<-av_Diff_SPA(MRIOdt_REF, MRIOdt_bis, ListThres, TypContenu = "Emi", PathEmi = PathTemp,)
## End(Not run)
```

`av_dl_MicroMRIO` *av_dl_MicroMRIO*

Description

av_dl_MicroMRIO

Usage

```
av_dl_MicroMRIO(dl, OptXLSout = FALSE)
```

Arguments

`dl`	long data
`OptXLSout`	to export the mini-MRIO in Excel format

Value

dl micro

Examples

```
## Not run:  DT_micro <- av_dl_MicroMRIO(DT, OptXLSout = FALSE)
DT_microXls <- av_dl_MicroMRIO(DT, OptXLSout = TRUE)
## End(Not run)
```

`av_dl_UE27` *av_dl_UE27 Build a new MRIO with all EU countries aggragated into "UE27" item. Useful to calculate EU made-in for instance. WARNING : dl must be normalized (not raw data) for countries homogeneity reasons.*

Description

av_dl_UE27 Build a new MRIO with all EU countries aggragated into "UE27" item. Useful to calculate EU made-in for instance. WARNING : dl must be normalized (not raw data) for countries homogeneity reasons.

Usage

```
av_dl_UE27(dl, OptSaveRDS = "NO")
```

Arguments

dl	long data
OptSaveRDS	text

Value

dl with EU27 agreggate instead of each country

Examples

```
## Not run:  DT_UE27 <- av_dl_UE27(DT)
```

---

av_extend_MRIO_dw *av_extend_MRIO_dw Extend MRIO components by adding stressors like Value added or CO2 Emissions You can manage this function to add new cases : other MRIO options or other stressors etc.*

Description

av_extend_MRIO_dw Extend MRIO components by adding stressors like Value added or CO2 Emissions You can manage this function to add new cases : other MRIO options or other stressors etc.

Usage

```
av_extend_MRIO_dw(MRIO_dw, NameMRIO, TypExtension, Path1 = NULL)
```

Arguments

MRIO_dw	MRIO must be in wide format (not Long)
NameMRIO	text
TypExtension	text
Path1	text

Value

dw data wide (list)

Examples

```
## Not run:  Bonus <- CompoMRIO(DT, typeCompo = "OptFullOptionsBonus", OptTab = FALSE)
MRIO2 <- av_extend_MRIO_dw(MRIO_dw = Bonus, "FIGARO", TypExtension = "StressVA")
MRIO3 <- av_extend_MRIO_dw(MRIO_dw = Bonus, "FIGARO", TypExtension = "StressEmi", Path1 = PathTemp)
MRIO4 <- av_extend_MRIO_dw(MRIO3, "FIGARO", TypExtension = "StressVA")
## End(Not run)
```

av_fun_VarPostTransition

av_fun_VarPostTransition Function for robustness checks by applying structural changes at the wolrd economy to suit climate change Parameters description : - paramEmploi : Employment intensity (value = 0.9 => 10 - paramA : Intermediate consumption substitution intensity: value = 0.9 => the technical coefficient of each country's automotive industry (C29) in products C29 and C28 (domestic and imported) is reduced by 10 - paramDF : The intensity of final consumption substitution: value = 0.9 => each country's final demand for oil products (C19) is reduced by 10 The adjustments made are therefore zero-sum on intermediate consumption and final demand. The simulations are called by their X-Y-Z parameters: for example, 09-08-07 will correspond to a parameter of 0.9 for employment, 0.8 for intermediate consumption and 0.7 for final demand.

Description

av_fun_VarPostTransition Function for robustness checks by applying structural changes at the wolrd economy to suit climate change Parameters description : - paramEmploi : Employment intensity (value = 0.9 => 10 - paramA : Intermediate consumption substitution intensity: value = 0.9 => the technical coefficient of each country's automotive industry (C29) in products C29 and C28 (domestic and imported) is reduced by 10 - paramDF : The intensity of final consumption substitution: value = 0.9 => each country's final demand for oil products (C19) is reduced by 10 The adjustments made are therefore zero-sum on intermediate consumption and final demand. The simulations are called by their X-Y-Z parameters: for example, 09-08-07 will correspond to a parameter of 0.9 for employment, 0.8 for intermediate consumption and 0.7 for final demand.

Usage

```
av_fun_VarPostTransition(paramEmploi, paramA, paramDF, OptRDS = FALSE)
```

Arguments

paramEmploi	value percentage
paramA	value percentage
paramDF	value percentage
OptRDS	binary

Value

comparison table

Examples

```
## Not run:  av_fun_VarPostTransition(paramEmploi = 1, paramA = 1, paramDF = 1)
av_fun_VarPostTransition(paramEmploi = 0.9, paramA = 0.8, paramDF = 0.7)
## End(Not run)
```

---

av_HRM *av_HRM Master function in the ecosystem : Function HRM (hypothetical repatriation method) repat_pct can come from an other function to calibrate VARIANTs options : OptVarianteDemande : ALL=Normal ; CIdom=CI domestics ; CIall=all CI (dom+exp) ; DFdom=DF domestics ; DFall=all final demand (dom+exp) ; OptVariantePaysImp : ALL=Normal ; <country>= normal but 1 country only ; horsUE=substitution outside EU only We implement a repat_pct repatriation of repat_pct to repatriate 1 Md??? of production, we set the previous repatriation The calculations are transversal to TEI and FD: the entire product is uniformly concerned.*

Description

av_HRM Master function in the ecosystem : Function HRM (hypothetical repatriation method) repat_pct can come from an other function to calibrate VARIANTs options : OptVarianteDemande : ALL=Normal ; CIdom=CI domestics ; CIall=all CI (dom+exp) ; DFdom=DF domestics ; DFall=all final demand (dom+exp) ; OptVariantePaysImp : ALL=Normal ; <country>= normal but 1 country only ; horsUE=substitution outside EU only We implement a repat_pct repatriation of repat_pct to repatriate 1 Md??? of production, we set the previous repatriation The calculations are transversal to TEI and FD: the entire product is uniformly concerned.

Usage

```
av_HRM(
  dl,
  repat_country,
  repat_indus,
  repat_pct,
  verboseCheck = FALSE,
  OptSommeDFenP3_S14 = TRUE,
  OptVarianteDemande = "ALL",
  OptVariantePaysImp = "ALL",
  OptBaseIntermAvantRecalcProd = TRUE
)
```

Arguments

dl	datatable
repat_country	text country
repat_indus	text industry

repat_pct	value percentage
verboseCheck	binary
OptSommeDFenP3_S14	binary
OptVarianteDemande	binary
OptVariantePaysImp	binary
OptBaseIntermAvantRecalcProd	binary

Value

list of dl and dw

`av_MadeIn` *av_MadeIn Function calculation of MADE-IN*

Description

av_MadeIn Function calculation of MADE-IN

Usage

```
av_MadeIn(
  dtdl,
  Optdl = TRUE,
  annee,
  OptDonneesBrutes = FALSE,
  MadeInPays = "FRA",
  OptUE27 = FALSE
)
```

Arguments

dtdl	data
Optdl	binary
annee	year
OptDonneesBrutes	binary
MadeInPays	text country
OptUE27	binary

Value

dl data long

---

av_MRIO_comparison *av_MRIO_comparison Function to compare 2 MRIOs MRIOs can be initially in long format (Optdl) or in wide format (split comparison of all components of the MRIO)*

Description

av_MRIO_comparison Function to compare 2 MRIOs MRIOs can be initially in long format (Optdl) or in wide format (split comparison of all components of the MRIO)

Usage

```
av_MRIO_comparison(MRIO1, MRIO2, Optdl = TRUE, OptVerbose = FALSE)
```

Arguments

MRIO1	MRIO object
MRIO2	MRIO object
Optdl	long data option
OptVerbose	verbose option

Value

An ordered comparison (if dl) or a list of ordered comparisons (if wide format)

Examples

```
## Not run:  M_test <- av_MRIO_comparison(Bonus, BonusBis, Optdl = F, OptVerbose = T)
```

---

av_SPA *av_SPA Master function in the environment : Structural Path Analysis (SPA) Function for calculating the Stuctural Path Analysis (SPA) of a MRIO The MRIO can be in long format or it has already undergone a CompoMRIO (dt) TypeSPA="VA" or "Emi" ListThres = GenThres=,Thres_L2_1=,Thres_L3_1=,Thres_L3_2=,Thres_L3_3,Thres_L3_4=,Thres_L4_1=,Thres Fitted FIGARO VA : ListThres = GenThres=0.001,Thres_L2_1=0.001,Thres_L3_1=0.001,Thres_L3_2=0.001,Thres_L3_3=0.00001,T Fitted FIGARO Emi : ListThres = GenThres=0.001,Thres_L2_1=0.0001,Thres_L3_1=0.0001,Thres_L3_2=0.00001,Thres_L3_3=0.000*

Description

av_SPA Master function in the environment : Structural Path Analysis (SPA) Function for calculating the Stuctural Path Analysis (SPA) of a MRIO The MRIO can be in long format or it has already undergone a CompoMRIO (dt) TypeSPA="VA" or "Emi" ListThres = GenThres=,Thres_L2_1=,Thres_L3_1=,Thres_L3_2=,T
Fitted FIGARO VA : ListThres = GenThres=0.001,Thres_L2_1=0.001,Thres_L3_1=0.001,Thres_L3_2=0.001,Thres_L3_3=
Fitted FIGARO Emi : ListThres = GenThres=0.001,Thres_L2_1=0.0001,Thres_L3_1=0.0001,Thres_L3_2=0.00001,Thres_L

Usage

```
av_SPA(
  dtdl,
  Optdl = FALSE,
  TypeSPA = "VA",
  TypeMRIO = "FIGARO",
  ListThres,
  TargetCountry = "FRA",
  OptRDS = "",
  OptRDSDetail = "",
  OptUE27 = FALSE,
  verbose = FALSE,
  PathEmi = ""
)
```

Arguments

dtdl	datatable
Optdl	binary
TypeSPA	text options
TypeMRIO	text
ListThres	list thresholds
TargetCountry	text country
OptRDS	binary
OptRDSDetail	binary
OptUE27	binary
verbose	binary
PathEmi	text data link

Value

dt data table with path analysis results

Examples

```
## Not run: ListThres = {GenThres=0.001,Thres_L2_1=0.001,Thres_L3_1=0.001,Thres_L3_2=0.001,Thres_L3_3=0.001,Thr
```

`BoucleAnneesMADEINs` *BoucleAnneesMADEINs Loop to build insight database on MADE-IN 2 options : (1) On the full MRIO database (if memory allows) (2) by year but need to load all the dl and put them into a list (ListdlAnnual)*

Description

BoucleAnneesMADEINs Loop to build insight database on MADE-IN 2 options : (1) On the full MRIO database (if memory allows) (2) by year but need to load all the dl and put them into a list (ListdlAnnual)

Usage

```
BoucleAnneesMADEINs(
  dtdl,
  Optdl = TRUE,
  period,
  OptDonneesBrutes = FALSE,
  OptAnnual = FALSE,
  ListdlAnnual = list(),
  OptUE27 = FALSE
)
```

Arguments

`dtdl`	text
`Optdl`	binary
`period`	time period
`OptDonneesBrutes`	binary
`OptAnnual`	binary
`ListdlAnnual`	list
`OptUE27`	binary

Value

dl data long

BoucleLinkageBwdFwd *BoucleLinkageBwdFwd Function loop to calculate Backward and Forward Linkage indicators databases*

Description

BoucleLinkageBwdFwd Function loop to calculate Backward and Forward Linkage indicators databases

Usage

```
BoucleLinkageBwdFwd(base_dt, period)
```

Arguments

base_dt	datatable
period	time period

Value

dl data long

BouclePaysContVAdesExports *BouclePaysContVAdesExports Calculation loop of content of value addend embedded in exports for a list of countries and a given year Foster-McGregor, N., et R. Stehrer (2013) : "Value added content of trade : A comprehensive approach," Economics Letters, 120(2), 354–357.*

Description

BouclePaysContVAdesExports Calculation loop of content of value addend embedded in exports for a list of countries and a given year Foster-McGregor, N., et R. Stehrer (2013) : "Value added content of trade : A comprehensive approach," Economics Letters, 120(2), 354–357.

Usage

```
BouclePaysContVAdesExports(
  dtdl,
  Optdl = TRUE,
  annee,
  ListCountries = list(),
  OptUE27 = FALSE
)
```

Arguments

dtdl	data
Optdl	binary
annee	year
ListCountries	list
OptUE27	binary

Value

dl data long

BouclePaysEtAnneesContVAdesExports

BouclePaysEtAnneesContVAdesExports Calculation loop of content of value addend embedded in exports for a list of countries and a time period (several years) if OptAnnual=TRUE then dtdl must be a list of dtdl Foster-McGregor, N., et R. Stehrer (2013) : "Value added content of trade : A comprehensive approach," Economics Letters, 120(2), 354–357.

Description

BouclePaysEtAnneesContVAdesExports Calculation loop of content of value addend embedded in exports for a list of countries and a time period (several years) if OptAnnual=TRUE then dtdl must be a list of dtdl Foster-McGregor, N., et R. Stehrer (2013) : "Value added content of trade : A comprehensive approach," Economics Letters, 120(2), 354–357.

Usage

```
BouclePaysEtAnneesContVAdesExports(
  dtdl,
  Optdl = TRUE,
  period,
  ListCountries = list(),
  OptUE27 = FALSE,
  OptAnnual = FALSE
)
```

Arguments

dtdl	data
Optdl	binary
period	time period
ListCountries	list
OptUE27	binary
OptAnnual	binary

Value

dl data long

`Build_MadeIn_byOrigin` *Build_MadeIn_byOrigin Build contents of VA by country of origin*

Description

Build_MadeIn_byOrigin Build contents of VA by country of origin

Usage

```
Build_MadeIn_byOrigin(
  BDn_LR_WIOD,
  BDn_WIOD,
  BDn_FIG,
  BDn_ICIO,
  SelectCountry = "FRA",
  OptSaveRDS = FALSE
)
```

Arguments

`BDn_LR_WIOD`	datatable
`BDn_WIOD`	datatable
`BDn_FIG`	datatable
`BDn_ICIO`	datatable
`SelectCountry`	text country
`OptSaveRDS`	binary

Value

dl data long

`CFPcalculationRCPP` *Matrix Multiplication : 3 components*

Description

This function returns the matrix multiplication of 3 matrix as for footprint calculation

Usage

```
CFPcalculationRCPP(ef, L, FD)
```

Arguments

`ef`	matrix
`L`	matrix
`FD`	matrix

`CompoMRIO` *CompoMRIO Master function in the ecosystem Allow to build Wide MRIO (dw) from Long MRIO (dl), by splitting components You can add other components if needed Afterward you can use av_extend_MRIO_dw() function to extend with environmental data for example*

Description

CompoMRIO Master function in the ecosystem Allow to build Wide MRIO (dw) from Long MRIO (dl), by splitting components You can add other components if needed Afterward you can use av_extend_MRIO_dw() function to extend with environmental data for example

Usage

```
CompoMRIO(MRIO_dl, typeCompo, date = 9999, OptTab = FALSE, OptUE27 = FALSE)
```

Arguments

`MRIO_dl`	long format MRIO
`typeCompo`	option type of composition
`date`	year
`OptTab`	tabular output option
`OptUE27`	option with UE27 modality

Value

dw data wide (list of components)

Examples

```
## Not run:  DT <- readRDS(paste0(PathTest, "BDn_FIG_2010.rds"))
Bonus <- CompoMRIO(DT, typeCompo = "OptFullOptionsBonus", date = 2010, OptTab = FALSE)
## End(Not run)
```

ContentVAExports_Retropolation

ContentVAExports_Retropolation Retropolation of content in value added of exports WIOD is the reference and LRWIOD is used to retropolate backward, and FIGARO is used to retropolate forward. ICIO is not used here because of systematic retropolation method Very specific task : can be adapted depending on data you're using

---

Description

ContentVAExports_Retropolation Retropolation of content in value added of exports WIOD is the reference and LRWIOD is used to retropolate backward, and FIGARO is used to retropolate forward. ICIO is not used here because of systematic retropolation method Very specific task : can be adapted depending on data you're using

Usage

```
ContentVAExports_Retropolation(
  ContVAdesExports_LRWIOD,
  ContVAdesExports_WIOD,
  ContVAdesExports_FIGARO,
  OptSaveRDS = FALSE
)
```

Arguments

ContVAdesExports_LRWIOD	datatable
ContVAdesExports_WIOD	datatable
ContVAdesExports_FIGARO	datatable
OptSaveRDS	binary

Value

dl data long

ContentVAExports_Retropolation_UE27

ContentVAExports_Retropolation_UE27 Retropolation of content in value added of exports for UE27 and big areas Need to aggregate data before to have only one area EU Very specific task : can be adapted depending on data you're using

Description

ContentVAExports_Retropolation_UE27 Retropolation of content in value added of exports for UE27 and big areas Need to aggregate data before to have only one area EU Very specific task : can be adapted depending on data you're using

Usage

```
ContentVAExports_Retropolation_UE27(
  resMadeInUE27_LRWIOD,
  resMadeInUE27_WIOD,
  resMadeInUE27_FIGARO,
  resMadeInUE27_ICIO,
  OptGraph = FALSE,
  OptSaveRDS = FALSE
)
```

Arguments

resMadeInUE27_LRWIOD	datatable
resMadeInUE27_WIOD	datatable
resMadeInUE27_FIGARO	datatable
resMadeInUE27_ICIO	datatable
OptGraph	binary
OptSaveRDS	binary

Value

dl data long

Contenus — *Contenus Master function in the ecosystem Function calculation of contents (in VA, émissions CO2, emploi) : allow footprint calculation along different formats WARNING Needs CompoMRIO with bonus and extensions*

Description

Contenus Master function in the ecosystem Function calculation of contents (in VA, émissions CO2, emploi) : allow footprint calculation along different formats WARNING Needs CompoMRIO with bonus and extensions

Usage

```
Contenus(dw, typeContenu, MethContenu = "MatDF", EmprPays = "FRA")
```

Arguments

dw	wide data MRIO object
typeContenu	text
MethContenu	text
EmprPays	text country

Value

dl data long

Examples

```
## Not run: List_Contenus <- Contenus(List_Interm, typeContenu = "VA", MethContenu = "MatDF", EmprPays = SelectCou
```

ContVAdesExports — *ContVAdesExports Calculation of content of value addend embedded in exports for a given country and a given year Foster-McGregor, N., et R. Stehrer (2013) : "Value added content of trade : A comprehensive approach," Economics Letters, 120(2), 354–357.*

Description

ContVAdesExports Calculation of content of value addend embedded in exports for a given country and a given year Foster-McGregor, N., et R. Stehrer (2013) : "Value added content of trade : A comprehensive approach," Economics Letters, 120(2), 354–357.

Usage

```
ContVAdesExports(dtdl, Optdl = TRUE, annee, pays, OptUE27 = FALSE)
```

Arguments

dtdl	data
Optdl	binary
annee	year
pays	text country
OptUE27	binary

Value

dl data long

---

dw_to_dl *dw_to_dl Convert dw to dl*

Description

dw_to_dl Convert dw to dl

Usage

```
dw_to_dl(dw)
```

Arguments

dw	wide data

Value

dl data long

EmissionsProd *EmissionsProd Function to calculate Inventory Emissions (coming from production) Be careful : [["EmiOverOuput"]] must have been introduced in our MRIO previously (with av_extend_MRIO_dw() function)*

Description

EmissionsProd Function to calculate Inventory Emissions (coming from production) Be careful : [["EmiOverOuput"]] must have been introduced in our MRIO previously (with av_extend_MRIO_dw() function)

Usage

```
EmissionsProd(MRIOinterm)
```

Arguments

MRIOinterm	MRIO object

Value

df data frame

Examples

```
## Not run:  EmiProd_MRIO <- EmissionsProd(MRIO)
```

GereInfNA — *GereInfNA Function (very useful) to convert infinity data into NA and NA data into 0 or 1 for example*

Description

GereInfNA Function (very useful) to convert infinity data into NA and NA data into 0 or 1 for example

Usage

```
GereInfNA(df, impute = 0)
```

Arguments

df	dataframe
impute	value to inpute

Value

df data frame

Examples

```
## Not run:  if (TypeSPA == "VA") {
  f_dt <- vectDF(diag(MRIO[["VAOverOuput"]]))
  f_dt <- GereInfNA(f_dt)
}
## End(Not run)
```

GetRownamesFromFirstCol *GetRownamesFromFirstCol Get Rownames From First Column*

Description

GetRownamesFromFirstCol Get Rownames From First Column

Usage

```
GetRownamesFromFirstCol(df)
```

Arguments

df	dataframe

Value

df data frame

Herfindahl *Herfindahl Function Herfindahl (concentration indicator)*

Description

Herfindahl Function Herfindahl (concentration indicator)

Usage

```
Herfindahl(MRIOinterm, verbose = F)
```

Arguments

MRIOinterm	MRIO object
verbose	binary

Value

dl data long

`ImportedContentInVA` *ImportedContentInVA Function to calculate the imported content in value added it works with 1 country but you can loop all over the contries available in the MRIO.*

Description

ImportedContentInVA Function to calculate the imported content in value added it works with 1 country but you can loop all over the contries available in the MRIO.

Usage

```
ImportedContentInVA(pays)
```

Arguments

`pays`	text country

Value

dl data long

`IndicVariant_IndusCountry` *IndicVariant_IndusCountry Function to calculate differences of indicators resulting from a HRM variant for instance*

Description

IndicVariant_IndusCountry Function to calculate differences of indicators resulting from a HRM variant for instance

Usage

```
IndicVariant_IndusCountry(
  IndusREF,
  CountryREF = "FRA",
  ListCountryREF = c("FRA", "DEU", "GBR", "ESP", "ITA"),
  MRIO,
  MRIObis,
  RatioEmploiVA
)
```

Arguments

IndusREF	text industry
CountryREF	text country
ListCountryREF	list
MRIO	MRIO object
MRIObis	MRIO object
RatioEmploiVA	text data link

Value

dl data long

Info_MRIO

*Info_MRIO Info_MRIO function: provides information on a product * country (equivalent to branch * country) MRIO must be derived from compoMRIO nb_top : number of crosses to display (ex: nb of countries of origin) Output in print format and save possible.*

Description

Info_MRIO Info_MRIO function: provides information on a product * country (equivalent to branch * country) MRIO must be derived from compoMRIO nb_top : number of crosses to display (ex: nb of countries of origin) Output in print format and save possible.

Usage

```
Info_MRIO(MRIO, Indus_select, Country_select, nb_top = 5, verbose = T)
```

Arguments

MRIO	MRIO object
Indus_select	Industry selection
Country_select	Country selection
nb_top	Number of outputs
verbose	verbose option

Value

text

Examples

```
## Not run:  Info_MRIO(MRIO, "A01", "FRA", nb_top = 5)
test <- Info_MRIO(MRIO, "C19", "FRA", nb_top = 10) / print(test)
## End(Not run)
```

LinkageBwdFwd *LinkageBwdFwd Function Backward and Forward Linkage*

Description

LinkageBwdFwd Function Backward and Forward Linkage

Usage

```
LinkageBwdFwd(base_dt, annee)
```

Arguments

base_dt	datatable
annee	year

Value

dl data long

ListsReferential *ListsReferential Function to return lists of countries or operations or both You can add lists depending of new MRIO for exemple, or depending of releases*

Description

ListsReferential Function to return lists of countries or operations or both You can add lists depending of new MRIO for exemple, or depending of releases

Usage

```
ListsReferential(TypeList)
```

Arguments

TypeList	Choose the list type

Value

list

Examples

```
## Not run:  if (TypeList == "BRFD_FIG_WIOD_LRWIOD_ICIO") {
 List_out <- append(ListsReferential("BR_FIG_WIOD_LRWIOD_ICIO"), ListsReferential("FD_FIG_WIOD_LRWIOD_ICIO"))
}
return(List_out)
## End(Not run)
```

`MadeIn_byOrigin` *MadeIn_byOrigin Calculate MAdeIn by country of origin, for analytics*

Description

MadeIn_byOrigin Calculate MAdeIn by country of origin, for analytics

Usage

```
MadeIn_byOrigin(
  RESULT_MADEINs_byCountry,
  SelectCountries = c("FRA", "DEU", "GBR", "ESP", "ITA", "USA", "CHN"),
  MadeInOf = "FRA",
  OptSaveRDS = FALSE
)
```

Arguments

`RESULT_MADEINs_byCountry`	datatable
`SelectCountries`	list
`MadeInOf`	text country
`OptSaveRDS`	binary

Value

dl data long

`MadeIn_Manuf` *MadeIn_Manuf Calculation of manuf's made-in by agreggation (2 classifications)*

Description

MadeIn_Manuf Calculation of manuf's made-in by agreggation (2 classifications)

Usage

```
MadeIn_Manuf(
  resMadeIn_LRWIOD,
  resMadeIn_WIOD,
  resMadeIn_FIGARO,
  resMadeIn_ICIO,
  OptSaveRDS = FALSE
)
```

Arguments

`resMadeIn_LRWIOD`	datatable
`resMadeIn_WIOD`	datatable
`resMadeIn_FIGARO`	datatable
`resMadeIn_ICIO`	datatable
`OptSaveRDS`	binary

Value

dl data long

`MadeIn_Retropolation` *MadeIn_Retropolation Retropolation of made-in to fit long series inter-MRIO You can manage the MRIO list*

Description

MadeIn_Retropolation Retropolation of made-in to fit long series inter-MRIO You can manage the MRIO list

Usage

```
MadeIn_Retropolation(
  resMadeIn_manuf,
  MadeIn_levels,
  resMadeIn_LRWIOD,
  resMadeIn_WIOD,
  resMadeIn_FIGARO,
  resMadeIn_ICIO,
  OptSaveRDS = FALSE
)
```

Arguments

resMadeIn_manuf	datatable
MadeIn_levels	datatable
resMadeIn_LRWIOD	datatable
resMadeIn_WIOD	datatable
resMadeIn_FIGARO	datatable
resMadeIn_ICIO	datatable
OptSaveRDS	binary

Value

dl data long

MadeIn_Retropolation_UE27

MadeIn_Retropolation_UE27 Retropolation of made-in to fit long series inter-MRIO

Description

MadeIn_Retropolation_UE27 Retropolation of made-in to fit long series inter-MRIO

Usage

```
MadeIn_Retropolation_UE27(
  resMadeInUE27_LRWIOD,
  resMadeInUE27_WIOD,
  resMadeInUE27_FIGARO,
  resMadeInUE27_ICIO,
  OptSaveRDS = FALSE
)
```

Arguments

resMadeInUE27_LRWIOD	datatable
resMadeInUE27_WIOD	datatable
resMadeInUE27_FIGARO	datatable
resMadeInUE27_ICIO	datatable
OptSaveRDS	binary

Value

dl data long

Mult2_rcpp3 *Matrix Multiplication : 2 components*

Description

This function returns the matrix multiplication of 2 matrix

Usage

```
Mult2_rcpp3(matA, matB)
```

Arguments

matA	matrix
matB	matrix

PasteN *PasteN Function to concatenate the amounts (in value) of several columns of a DT (in list form)*

Description

PasteN Function to concatenate the amounts (in value) of several columns of a DT (in list form)

Usage

```
PasteN(dt, column_ref)
```

Arguments

dt	datatable
column_ref	column number

Value

text

Examples

```
## Not run:  P2 <- PasteN(DF_TOT_EXP_topN_w, c("Col_Country", "Col_Indus"))
```

`PastePRBR` *PastePRBR Paste (Lig_Country and Lig_Indus) into PR and/or (Col_Country and Col_Indus) into BR. Please note: country names, branch names and product names must not contain underscores.*

Description

PastePRBR Paste (Lig_Country and Lig_Indus) into PR and/or (Col_Country and Col_Indus) into BR. Please note: country names, branch names and product names must not contain underscores.

Usage

```
PastePRBR(dl)
```

Arguments

dl	data long, like normalized database

Value

dl data long

Examples

```
## Not run:  test <- PastePRBR(DT)
```

RatioNoguera *RatioNoguera Calculation of Noguera indicator*

Description

RatioNoguera Calculation of Noguera indicator

Usage

```
RatioNoguera(BaseResult, nomMRIO)
```

Arguments

BaseResult	datatable
nomMRIO	text name MRIO

Value

dl data long

ReqSum *ReqSum Query function to calculate a sum from a list of dimensions to be summed ListDimASommer is the normalized list of dimensions on wich you wants to sum up. OptStruct keeps initial format of data bur compute structure on the given list of dimensions (ListDimASommer)*

Description

ReqSum Query function to calculate a sum from a list of dimensions to be summed ListDimASommer is the normalized list of dimensions on wich you wants to sum up. OptStruct keeps initial format of data bur compute structure on the given list of dimensions (ListDimASommer)

Usage

```
ReqSum(dl, ListDimASommer, OptStruct = FALSE)
```

Arguments

dl	data_long format database
ListDimASommer	list of dimensions
OptStruct	option for structure results

Value

dl data long

Examples

```
## Not run:  Cadre <- Base_depart[["DF"]]
Cadre_Tot <- ReqSum(Cadre, "Col_Indus") # Sum components of final demand
## End(Not run)
```

`rndN` *rndN Rounding function, default 0 decimal places*

Description

rndN Rounding function, default 0 decimal places

Usage

```
rndN(dataa, rnd = 0)
```

Arguments

`dataa`	data to round
`rnd`	round decimal option

Value

rounded value

`SoldExtPays` *SoldExtPays Function to calculate external balance for all countries of a MRIO*

Description

SoldExtPays Function to calculate external balance for all countries of a MRIO

Usage

```
SoldExtPays(MRIOinterm)
```

Arguments

`MRIOinterm`	MRIO object

Value

df data frame

Examples

```
## Not run:  SoldeExt <- SoldExtPays(MRIO)
```

SommeDFenP3_S14 *SommeDFenP3_S14 Function which sums up all final demand and stores it in P3_14 Caution: potentially biases detailed interpretation of MRIO components On the other hand, remains compatible with all functions and avoids the weird things of P5M (stock variatins that make HRM jump on C22 in particular) Only works with FIGARO at this time, needs more lists if you wants to expand*

Description

SommeDFenP3_S14 Function which sums up all final demand and stores it in P3_14 Caution: potentially biases detailed interpretation of MRIO components On the other hand, remains compatible with all functions and avoids the weird things of P5M (stock variatins that make HRM jump on C22 in particular) Only works with FIGARO at this time, needs more lists if you wants to expand

Usage

```
SommeDFenP3_S14(base_dl)
```

Arguments

base_dl	long database

Value

dl data long

Examples

```
## Not run:  if (OptSommeDFenP3_S14 == TRUE) {
  Base_init <- SommeDFenP3_S14(data.table::copy(dl))
}
## End(Not run)
```

SplitPRBR *SplitPRBR Split of PR and BR into four parts or only PR or only BR Please note: country names, branch names and product names must not contain underscores.*

Description

SplitPRBR Split of PR and BR into four parts or only PR or only BR Please note: country names, branch names and product names must not contain underscores.

Usage

```
SplitPRBR(dl)
```

Arguments

dl	data long, like normalized database

Value

dl data long

Examples

```
## Not run:  test <- SplitPRBR(DT)
```

---

vectDF *vectDF Function to convert a vector into dataframe*

Description

vectDF Function to convert a vector into dataframe

Usage

```
vectDF(vect)
```

Arguments

vect	vector

Value

df data frame

Examples

```
## Not run:  MyDF <- vectDF(DF[, 1])
```

Index