Package ‘ITNr’

March 11, 2020

Type Package
Title Analysis of the International Trade Network
Version 0.6.0
Author Matthew Smith
Maintainer Matthew Smith <matt_smith.90@hotmail.co.uk>
Description Functions to clean and process international trade data into an international trade network (ITN) are provided. It then provides a set a functions to undertake analysis and plots of the ITN (extract the backbone, centrality, blockmodels, clustering). Examining the key players in the ITN and regional trade patterns.
Depends R (>= 2.15.1), network
License GPL-3
Encoding UTF-8
Imports stats,circlize,graphics,RColorBrewer,
    reshape2,maps,blockmodeling,igraph,utils,plyr,plyr,
    cowplot,ggplot2,GGally,fastmatch,intergraph,
    sna,tnet,WDI,networkDynamic
LazyData true
RoxygenNote 7.0.2
NeedsCompilation no
Repository CRAN
Date/Publication 2020-03-11 17:30:13 UTC

R topics documented:

  adjust_mat ........................................ 2
cap_lat_lon ........................................ 3
Contradrclean ...................................... 3
core_periphery_weighted ............................ 4
ei_group ........................................... 5
ei_ind ............................................. 6
ei_network .......................................... 7
adjust_mat

Adjust Matrix

Description

adjust the dimensions of a source object to the dimensions of a target object

Usage

adjust_mat(
  source,
  target,
  remove = TRUE,
  add = TRUE,
  value = NA,
  returnlabels = FALSE
)
**Arguments**

- **source**: A matrix which should be adjusted (one-mode & directed).
- **target**: A matrix (one-mode & directed) to which the source object is compared with regard to its labels.
- **remove**: Should rows and columns that are not present in the target object be removed?
- **add**: Should rows and columns that are present in the target object but not in the source object be added to the source object?
- **value**: The value to be inserted if a new row or column is added. By default, new cells are filled with NA values, but other sensible values may include -Inf or 0.
- **returnlabels**: Return a list of added and removed row and column labels rather than the actual matrix, vector, or network object?

**Value**

Matrix

---

**Description**

Dataframe of capital city latitude and longitude coordinates

**Usage**

cap_lat_lon

---

**Description**

This function takes (import) trade data downloaded from comtrade - potentially using the comtradr package, cleans it and transforms it into a network. Adding a number of country level attributes to nodes in the network, including: regional partition, GDP, GDP per capita, GDP growth and FDI. However, it is important to note the limits of using comtradr to construct a network. Firstly when downloading the data with comtradr, you must specify reporters and partners – yet you cannot put “all” for both – only for either reporters or partners. Then for the other you are limited to a character vector of country names, length five or fewer. Therefore, this will not give you a full network. However, this function can be applied to trade data downloaded from UN Comtrade (download csv and read into R as a dataframe), or any other trade data which is in the same format as the comtradr dataframe.
Usage

Comtradrclean(DF, YEAR, threshold, cutoff)

Arguments

DF          Dataframe of trade data downloaded (potentially using the comtradr package)
YEAR        Year
threshold    Apply a threshold - TRUE, Extract the backbone - FALSE
cutoff      Threshold - cutoff level, Backbone - significance level

Value

International Trade Network - igraph object

Examples

##download data using comtradr
#require(comtradr)

##Download the trade data for tomatoes - code 0702
##All countries, Year - 2016
#ex_2 <- ct_search(reporters = "All",
#                  partners = c("USA","China",
#                  "Germany","Canada","Mexico"),
#                  trade_direction = "imports",
#                  start_date = "2016-01-01",
#                  end_date = "2016-12-31",
#                  commod_codes = "0702")

##this then gives a data frame which
##we can clean using the following function:
tomatoesITN<-Comtradrclean(ex_2,2016,TRUE,0.01)

##We apply a threshold - only retaining ties that are at least 0.01%
##of total tomatoes trade (amngst these countries)

core_periphery_weighted

Core-Periphery for Weighted Networks

Description

This function implements rich club core-periphery algorithm (Ma & Mondragón, 2015) to identify members of the core and periphery in weighted networks.
ei_group

Usage

core_periphery_weighted(gs, type)

Arguments

- **gs**: International Trade Network - igraph object. Note for networks not produced using ITNr there needs to be a vertex attribute "name" and edge attribute "weight"
- **type**: directed/undirected

Value

- List - 1.)igraph object with core-periphery results added as a node attribute. 2.) Dataframe of core-periphery results.

References


Examples

```r
require(igraph)
##Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(50,0.05,directed = TRUE)

##Add edge weights
E(ITN)$weight<-runif(ecount(ITN), 0, 1)

##Add vertex names
V(ITN)$name<-1:vcount(ITN)

##Implement core-periphery algorithm
ITNcp<-core_periphery_weighted(ITN,"directed")
```

ei_group

*Group level E-I Index*

Description

This function calculates the E-I Index (External-internal) at the group/attribute level

Usage

`ei_group(gs, attrname)`

Arguments

- **gs**: igraph object
- **attrname**: Attribute name
ei_ind

Value

Group level results dataframe

Examples

```r
require(igraph)
##Create random network (igraph object)
gs<-erdos.renyi.game(75,0.05,directed = TRUE)

##Add vertex names
V(gs)$name<-1:vcount(gs)

## Add an attribute
V(gs)$letters<- rep(LETTERS[1:5],15)

##Calculate the Group E-I Results
EI_GROUP_DATAFRAME<-ei_group(gs,"letters")
```

ei_ind

**Individual/Node level E-I Index**

Description

This function calculates the E-I Index (External-internal) at the individual/node level

Usage

```r
ei_ind(gs, attrname)
```

Arguments

- `gs`: igraph object
- `attrname`: Attribute name

Value

Group level results dataframe

Examples

```r
require(igraph)
##Create random network (igraph object)
gs<-erdos.renyi.game(75,0.05,directed = TRUE)

##Add vertex names
V(gs)$name<-1:vcount(gs)

## Add an attribute
V(gs)$letters<- rep(LETTERS[1:5],15)

##Calculate the Group E-I Results
EI_GROUP_DATAFRAME<-ei_group(gs,"letters")
```
ei_network

V(gs)$letters<- rep(LETTERS[1:5],6)

##Calculate the Individual E-I Results
EI_IND_DATAFRAME<-ei_ind(gs,"letters")

---

ei_network

*Network level E-I Index*

**Description**

This function calculates the E-I Index (External-internal) at the network level

**Usage**

```r
ei_network(gs, attrname)
```

**Arguments**

- `gs`: igraph object
- `attrname`: Attribute name

**Value**

Group level results dataframe

**Examples**

```r
require(igraph)
##Create random network (igraph object)
gs<-erdos.renyi.game(75,0.05,directed = TRUE)

##Add vertex names
V(gs)$name<-1:vcount(gs)

## Add an attribute
V(gs)$letters<- rep(LETTERS[1:5],15)

##Calculate the Group E-I Results
EI_NETWORK<-ei_network(gs,"letters")
```
ELEnet16

*Electrical Automotive Goods 2016 Network*

**Description**


**Usage**

ELEnet16

**References**


ELEnetList

*List of Electrical Automotive Goods Networks (2006-2016)*

**Description**

List of Electrical Automotive Goods Networks for 2006 - 2016. Electrical automotive goods category as defined by Amighini & Gorgoni (2014)

**Usage**

ELEnetList

**References**

get.backbone

Description

This function extracts the backbone of a network

Usage

get.backbone(G, alpha, directed = TRUE)

Arguments

G  igraph network
alpha  Significance level
directed  Default is TRUE

Value

Backbone of the network

References


Examples

```r
require(igraph)

##Create a random (directed) network
gs<-erdos.renyi.game(50, 0.2, directed = TRUE)

##Add edge weights to the network
E(gs)$weight<-runif(ecount(gs), 0, 1)

##Extract backbone at 0.05 significance level
backbone<-get.backbone(gs, 0.1)
```
**isEmpty**

**Description**

This function check whether data is numeric(0) and give returns an NA if this is true and the value of the data otherwise.

**Usage**

```r
isEmpty(x)
```

**Arguments**

- **x**  Data

**Value**

NA or the data

---

**ITNadjust**

**Adjust ITN**

**Description**

This function adjusts ITN matrices so they are the same size

**Usage**

```r
ITNadjust(MATlist, j)
```

**Arguments**

- **MATlist**  A list of ITN matrices
- **j**  Element of matrix list to compare with others

**Value**

Matrix
Examples

```r
##Create a list of random matrices (of different sizes)
##Labels - letters of alphabet (can represent actor names)
mat1<- matrix(round(runif(10*10)), 10, 10)
rownames(mat1)<-LETTERS[1:10]
colnames(mat1)<-LETTERS[1:10]

mat2<- matrix(round(runif(10*10)), 10, 10)
rownames(mat2)<-LETTERS[10:19]
colnames(mat2)<-LETTERS[10:19]

mat3<- matrix(round(runif(12*12)), 12, 12)
rownames(mat3)<-LETTERS[15:26]
colnames(mat3)<-LETTERS[15:26]

##Create matrix list
MATlist<-list(mat1,mat2,mat3)

##Adjust matrix 1 so that it has additional rows/actors not
##in the original matrix
mat1adjust<-ITNadjust(MATlist,1)
```

**ITNblock_plot**

**ITN Blockmodel Plot**

**Description**

This function calculates block membership for the ITN and then plots the network, with node colour according to block membership.

**Usage**

```r
ITNblock_plot(gs, LABEL)
```

**Arguments**

- `gs` : International Trade Network - igraph object
- `LABEL` : Should labels be present - TRUE/FALSE

**Value**

Network Plot - nodes coloured based on block membership
ITNblock_se

Description
This function calculates block membership for ITN and structural equivalence between countries

Usage
ITNblock_se(gs)

Arguments

gs International Trade Network - igraph object

Value
List object containing block membership and structural equivalence matrix results

Examples

```
require(igraph)
require(sna)
require(intergraph)

##Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(75,0.05,directed = TRUE)

##Add edge weights
E(ITN)$weight<-runif(ecount(ITN), 0, 1)

##Blockmodel plot
block_plot<-ITNblock_plot(ITN,FALSE)
```

```
#Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(50,0.05,directed = TRUE)

##Add edge weights
E(ITN)$weight<-runif(ecount(ITN), 0, 1)

##Blockmodel & structural equivalence analysis
blockse<-ITNblock_se(ITN)
```
### ITNcentrality

**Description**

This function calculates a number of centrality metrics for the weighted International Trade Network (ITN).

**Usage**

```r
ITNcentrality(gs)
```

**Arguments**

- `gs`: International Trade Network - igraph object

**Value**

Table of centrality results (dataframe)

**Examples**

```r
require(igraph)
# Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(75,0.05,directed = TRUE)

# Add edge weights
E(ITN)$weight<-runif(ecount(ITN), 0, 1)

# Add vertex names
V(ITN)$name<-1:vcount(ITN)

# Calculate the centrality measures
ITNCENT<-ITNcentrality(ITN)
```

---

### ITNcentrality_binary

**Description**

This function calculates a number of centrality metrics for the binary International Trade Network (ITN).

**Usage**

```r
ITNcentrality_binary(gs)
```
Arguments

gs International Trade Network - binary igraph object

Value

Table of centrality results (dataframe)

Examples

```r
require(igraph)
## Create random International Trade Network (igraph object)
ITN<-.erdos.renyi.game(75,0.05,directed = TRUE)

## Add vertex names
V(ITN)$name<-'1:vcount(ITN)

## Calculate the centrality measures
ITNCENT<-.ITNcentrality_binary(ITN)
```

---

**ITNcluster**  
*ITN Cluster*

Description

This function calculates cluster membership for ITN

Usage

```
ITNcluster(gs)
```

Arguments

gs International Trade Network - igraph object (with region attribute)

Value

Cluster object containing various cluster membership results

Examples

```r
## Load ITN
data(ELEnet16)

## Cluster Analysis
CLU<-.ITNcluster(ELEnet16)
```
ITNcorr

**ITN Correlation Plot**

**Description**

This function plots the correlation between degree and strength scores.

**Usage**

ITNcorr(gs)

**Arguments**

- **gs** International Trade Network - igraph object

**Value**

Correlation plot

**Examples**

```
require(igraph)

##Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(75,0.05,directed = TRUE)

##Add edge weights
E(ITN)$weight<-runif(ecount(ITN), 0, 1)

##Plot correlation matrix between degree and strength scores.
corr_plot<-ITNcorr(ITN)
```

ITNdegdist

**ITN Degree Distribution**

**Description**

This function plots the ITN (probability) degree distribution.

**Usage**

ITNdegdist(gs)

**Arguments**

- **gs** International Trade Network - igraph object
ITNdynamic

Value
Panel of ITN degree distribution plots

Examples
require(igraph)

##Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(75,0.05,directed = TRUE)

##Plot degree distribution
deg_dist_plot<-ITNdegdist(ITN)

ITNdynamic  Dynamic ITN

Description
This function produces a dynamic network object for ITNs. It cleans and adjusts the individual networks, so they are the same size. This dynamic network object can then be used to create animations, mapping changes over time and to calculate temporal network statistics

Usage
ITNdynamic(NETlist)

Arguments
NETlist A list of International Trade Networks (igraph objects)

Value
It returns the Dynamic Network Object

Examples
require(igraph)

##Create a set of random International Trade Networks (igraph objects)
##and add vertex names
ITN1<-erdos.renyi.game(75,0.05,directed = TRUE)
V(ITN1)$name<-1:vcount(ITN1)
ITN2<-erdos.renyi.game(100,0.01,directed = TRUE)
V(ITN2)$name<-1:vcount(ITN2)
ITN3<-erdos.renyi.game(55,0.1,directed = TRUE)
V(ITN3)$name<-1:vcount(ITN3)
##Create network list
NETlist<-list(ITN1,ITN2,ITN3)

##Create Dynamic Network Object
ITNdyn<-ITNdynamic(NETlist)

---

###ITN Histogram Degree Distribution

**Description**

This function plots the histogram degree distribution for the ITN

**Usage**

ITNhistdegdist(gs)

**Arguments**

- **gs**: International Trade Network - igraph object

**Value**

Panel of ITN histogram degree distribution plots

**Examples**

```
require(igraph)

##Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(75,0.05,directed = TRUE)

##Add edge weights
E(ITN)$weight<-runif(ecount(ITN), 0, 1)

##Plot degree distribution histogram
hist_deg_dist<-ITNhistdegdist(ITN)
```
**ITNimvex**

ITN - Exports vs Imports Plot

**Description**

The following function produces a plot showing imports (in degree) vs exports (out degree). This allows us to identify whether in the ITN, countries that export high levels also import high levels. The plot can be produced for either weighted or binary import and export ties.

**Usage**

`ITNimvex(gs, weighted)`

**Arguments**

- `gs` International Trade Network - igraph object
- `weighted` TRUE - plot import strength vs export strength. FALSE - Import count Vs export count

**Value**

Imports Vs Exports Plot

**Examples**

```r
require(igraph)

##Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(75,0.05,directed = TRUE)

##Add edge weights
E(ITN)$weight<-runif(ecount(ITN), 0, 1)

##Plot binary import vs exports
imvex_plot<-ITNimvex(ITN,FALSE)
```

---

**ITNplotset**

ITN Plots

**Description**

This function creates a panel of four plots of the ITN for a quick inspection. These include plots:
(i) highlighting clusters using the fast greedy algorithm.(ii)node colours for communities detected using the spinglass algorithm. (iii)nodes coloured by regional partition and (iv)with nodes coloured by regional partition and node size based on outdegree centrality.
ITNproperties

Usage

ITNplotset(gs)

Arguments

gs

International Trade Network - igraph object

Value

Panel of ITN plots

Examples

# Load the network
data(ELEnet16)

# Plot set of network visualisations
ITNplotset(ELEnet16)

---

ITNproperties

ITN Properties

Description

This function calculates network level properties for the ITN. These include:
- Size (number of nodes)
- Density
- Reciprocity
- Diameter
- Average path length
- Average node strength
- Average Degree
- Betweenness Centralisation
- Closeness Centralisation
- Eigenvector Centralisation
- Out Degree Centralisation
- In Degree Centralisation
- All Degree Centralisation
- Clustering coefficient (transitivity)
- Clustering Weighted
- Region Homophily
- Degree Assortativity

Usage

ITNproperties(gs)

Arguments

gs

International Trade Network - igraph object

Value

Table of centrality results (dataframe)
Examples

```r
# Load the network
data(ELEnet16)

# Calculate the network properties
ITNPROP <- ITNproperties(ELEnet16)
```

### Description

This function plots a single/clean ITN.

### Usage

```r
ITN_make_plot(gs, LABEL, REGION)
```

### Arguments

- **gs**: International Trade Network - igraph object
- **LABEL**: Should labels be present - TRUE/FALSE
- **REGION**: Should nodes be coloured on the basis of region TRUE/FALSE

### Value

Panel of ITN plots

### Examples

```r
# Load graph
data("ELEnet16")

# Otherwise download data from WITS and create an
# International Trade Network using WITSclean()

# Plot the network - No Label, colour by region
ITN_plot_example <- ITN_make_plot(ELEnet16, FALSE, TRUE)
```
**ITN_map_plot**  
*ITN plot on world map*

**Description**

This function plots the international trade network on a world map.

**Usage**

```r
ITN_map_plot(gs)
```

**Arguments**

- `gs`: International Trade Network - igraph object.

**Value**

Plot of the ITN on world map.

**Examples**

```r
require(maps)
##Load the ITN
data(ELEnet16)

## Plot ITN on map - node size based on outdegree
ITN_map_plot(ELEnet16)
```

---

**mixing_matrix_igraph**  
*Mixing Matrix*

**Description**

This function calculates the mixing matrix for an igraph object.

**Usage**

```r
mixing_matrix_igraph(gs, attrname)
```

**Arguments**

- `gs`: igraph object.
- `attrname`: Attribute name (vertex attribute).
Value

Mixing matrix

Examples

```r
require(igraph)
##Create random International Trade Network (igraph object)
gs<-erdos.renyi.game(50,0.05,directed = TRUE)

##Add vertex attributes
V(gs)$LETTER<-rep(LETTERS[1:5],10)

##Add vertex names
V(gs)$name<1:vcount(gs)

##Calculate mixing matrix
mixing_matrix<-mixing_matrix_igraph(gs,"LETTER")
```

```r
plot_degree_distribution
```

### Plot Degree Distribution

Description

This function plots degree distribution for any graph

Usage

```r
plot_degree_distribution(graph, a)
```

Arguments

- **graph**: igraph object
- **a**: mode - "in","out","all"

Value

Panel of ITN degree distribution plots

Examples

```r
require(igraph)
##Create random International Trade Network (igraph object)
ITN<-erdos.renyi.game(75,0.05,directed = TRUE)

##Plot out degree distribution
plot_degree_distribution(ITN,"in")
```
region_circle_plot

Description
This function creates a chord diagram/circle plot for levels of trade between regional partitions

Usage
region_circle_plot(gs)

Arguments
- gs: igraph ITN object (with attributes added)

Value
Circle Plot

Examples
```
##Load graph
data("ELEnet16")

##Create region circle plot
region_circle_plot(ELEnet16)
```

reorder_df

Description
Reorders the rows of one dataframe according to another vector (id vector)

Usage
reorder_df(df, col_sort, reorder_data)

Arguments
- df: dataframe to reorder
- col_sort: column on which the rows will be reordered
- reorder_data: vector with the new order
Value

Reordered dataframe

Examples

df <- data.frame(a = letters[1:3], b = LETTERS[4:6], c = 7:9)
reorder_data <- c("c", "a", "b")
df_new <- reorder_df(df, "a", reorder_data)
df_new

round_df

round_df

Description

This function rounds the numeric variables in a dataframe containing numeric and non-numeric data

Usage

round_df(x, digits)

Arguments

x dataframe
digits digits to round to

Value

Dataframe with rounded numbers

Examples

##Create dataframe
ID = c("a", "b", "c", "d", "e")
Value1 = c(3.445662, 6.44566, 8.75551, 1.114522, 1.5551)
Value2 = c(8.2, 1.7, 6.4, 19.45459, 10.34524)
df <- data.frame(ID, Value1, Value2)

##Round to 2 digits
rounddf <- round_df(df, 2)
WITSclean

Description
This function takes (import) trade data downloaded from WITS, cleans it and transforms it into a network. Adding a number of country level attributes to nodes in the network, including: regional partition, GDP, GDP per capita, GDP growth and FDI.

Usage
WITSclean(CSVfile, YEAR, threshold, cutoff)

Arguments
- CSVfile: WITS csv file
- YEAR: Year
- threshold: Apply a threshold - TRUE, Extract the backbone - FALSE
- cutoff: Threshold - cutoff level, Backbone - significance level

Value
International Trade Network - igraph object

Examples

```r
#Create an igraph object from international trade data downloaded from WITS

##Applies a threshold
##only retains ties that are at least 0.01% of total trade

ITN<-WITSclean("WITS_CSV_FILE_NAME.csv",2015,TRUE,0.01)
```
Index

*Topic datasets
  cap_lat_lon, 3
  ELENet16, 8
  ELENetList, 8

adjust_mat, 2

cap_lat_lon, 3
Comtradrclean, 3
core_periphery_weighted, 4

ei_group, 5
ei_ind, 6
ei_network, 7
ELENet16, 8
ELENetList, 8

get.backbone, 9

isEmpty, 10
ITN_make_plot, 20
ITN_map_plot, 21
ITN_adjust, 10
ITN_block_plot, 11
ITN_block_se, 12
ITN_centrality, 13
ITN_centrality_binary, 13
ITN_cluster, 14
ITN_corr, 15
ITN_degdist, 15
ITN_dynamic, 16
ITN_histdegdist, 17
ITN_imvex, 18
ITN_plotset, 18
ITN_properties, 19

mixing_matrix_igraph, 21
plot_degree_distribution, 22
region_circle_plot, 23