

# Package ‘ITNr’

April 26, 2018

**Type** Package

**Title** Analysis of the International Trade Network

**Version** 0.2.0

**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 of 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 (>= 2.10)

**License** GPL-3

**Encoding** UTF-8

**Imports** stats,circlize,GoodmanKruskal,graphics,RColorBrewer, reshape2,maps,blockmodeling,igraph,utils,plyr,dplyr, statnet,animation,Matrix,cowplot,ggplot2,GGally, fastmatch,intergraph,sna,tnet,WDI,network,xergm, xergm.common,networkDynamic,ndtv,comtradr

**LazyData** true

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-04-26 17:24:22 UTC

## R topics documented:

cap_lat_lon . . . . .	2
Comtradrclean . . . . .	2
core_periphery_weighted . . . . .	4
ELEnet16 . . . . .	5
ELEnetList . . . . .	5
get.backbone . . . . .	6

isEmpty . . . . .	7
ITNadjust . . . . .	7
ITNblock_plot . . . . .	8
ITNblock_se . . . . .	9
ITNcentrality . . . . .	10
ITNcentrality_binary . . . . .	10
ITNcluster . . . . .	11
ITNcorr . . . . .	12
ITNdegdist . . . . .	12
ITNdynamic . . . . .	13
ITNhistdegdist . . . . .	14
ITNimvex . . . . .	15
ITNplotset . . . . .	15
ITNproperties . . . . .	16
ITN_make_plot . . . . .	17
ITN_map_plot . . . . .	18
plot_degree_distribution . . . . .	18
RegionalBrokerage . . . . .	19
region_circle_plot . . . . .	20
round_df . . . . .	20
WITSclean . . . . .	21

**Index** **23**

---

cap_lat_lon	<i>cap_lat_lon</i>
-------------	--------------------

---

**Description**

Dataframe of capital city latitude and longitude coordinates

**Usage**

cap\_lat\_lon

---

Comtradrclean	<i>Comtrader data clean</i>
---------------	-----------------------------

---

## Description

This function takes (import) trade data downloaded 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, 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 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 (amongst 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

### Usage

```
core_periphery_weighted(gs, type)
```

### Arguments

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

### Value

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

### References

Ma A, Mondragón RJ (2015) Rich-Cores in Networks. PLoS ONE 10(3): e0119678. <https://doi.org/10.1371/journal.pone.0119678>

### Examples

```
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")
```

---

ELEnet16

*Electical Automtoive Goods 2016 Network*

---

**Description**

Electical Automotive Goods 2016 Network. Electrical automotive goods category as defined by Amighini & Gogoni (2014)

**Usage**

ELEnet16

**References**

Amighini, A. and Gorgoni, S. (2014) The International Reorganisation of Auto Production, The World Economy, 37(7), pp. 923–952.

---

ELEnetList

*List of Electical Automtoive Goods Networks (2006-2016)*

---

**Description**

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

**Usage**

ELEnetList

**References**

Amighini, A. and Gorgoni, S. (2014) The International Reorganisation of Auto Production, The World Economy, 37(7), pp. 923–952. (list of igrph objects)

---

<code>get.backbone</code>	<i>get.backbone</i>
---------------------------	---------------------

---

### 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

Serrano, M. Á., Boguñá, M. and Vespignani, A. (2009) Extracting the multiscale backbone of complex weighted networks, *Proceedings of the National Academy of Sciences*, 106(16), pp. 6483–6488.

### Examples

```
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)
```

---

<code>isEmpty</code>	<i>isEmpty</i>
----------------------	----------------

---

**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**

```
isEmpty(x)
```

**Arguments**

<code>x</code>	Data
----------------	------

**Value**

NA or the data

---

<code>ITNadjust</code>	<i>Adjust ITN</i>
------------------------	-------------------

---

**Description**

This function adjusts ITN matrices so they are the same size

**Usage**

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

**Arguments**

<code>MATlist</code>	A list of ITN matrices
<code>j</code>	Element of matrix list to compare with others

**Value**

Matrix

## Examples

```
##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

```
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



**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)
```

---

ITNblock\_se

*ITN Blockmodel & Structural Equivalence*

---

**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(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	<i>ITN Centrality</i>
---------------	-----------------------

---

**Description**

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

**Usage**

```
ITNcentrality(gs)
```

**Arguments**

gs                    International Trade Network - igraph object

**Value**

Table of centrality results (dataframe)

**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)

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

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

---

ITNcentrality_binary	<i>ITN Centrality for binary networks</i>
----------------------	---

---

**Description**

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

**Usage**

```
ITNcentrality_binary(gs)
```

**Arguments**

`gs` International Trade Network - binary igraph object

**Value**

Table of centrality results (dataframe)

**Examples**

```
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 and compares with regional groupings

**Usage**

```
ITNcluster(gs)
```

**Arguments**

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

**Value**

Cluster object containing various cluster membership and correlation results

**Examples**

```
##Load ITN
data(ELEnet16)

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

---

ITNcorr	<i>ITN Correlation Plot</i>
---------	-----------------------------

---

**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	<i>ITN Degree Distribution</i>
------------	--------------------------------

---

**Description**

This function plots the ITN (probability) degree distribuion

**Usage**

```
ITNdegdist(gs)
```

**Arguments**

gs                    International Trade Network - igraph object

**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)
```

---

ITNhistdegdist      *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**

```
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.

**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**

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

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

---

ITN_make_plot	<i>Single Clean ITN Plot</i>
---------------	------------------------------

---

**Description**

This function plots a single/clean ITN

**Usage**

```
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**

```
##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	<i>ITN plot on world map</i>
--------------	------------------------------

---

**Description**

This function plots the international trade network on a world map

**Usage**

```
ITN_map_plot(gs)
```

**Arguments**

gs                    International Trade Network - igraph object

**Value**

Plot of the ITN on world map

**Examples**

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

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

---

plot_degree_distribution	<i>Plot Degree Distribution</i>
--------------------------	---------------------------------

---

**Description**

This function plots degree distribution for any graph

**Usage**

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

**Arguments**

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

**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 out degree distribution
plot_degree_distribution(ITN,"in")
```

---

RegionalBrokerage      *ITN Gould-Fernandez Brokerage Analysis*

---

**Description**

This function conducts Gould-Fernandez Brokerage Analysis for the regional partition

**Usage**

```
RegionalBrokerage(gs, filename)
```

**Arguments**

gs	International Trade Network - igraph object
filename	Filename for the Brokerage Analysis

**Value**

Results Dataframe

**References**

Gould, R. V. and Fernandez, R. M. (1989) Structures of mediation: A formal approach to brokerage in transaction networks, *Sociological methodology*, 19(1989), pp. 89–126.

**Examples**

```
## Not run:
##Load graph
data("ELEnet16")

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

##Calculate Regional Brokerage
REGkey<-RegionalBrokerage(ELEnet16,"ELE_REG_BROKERAGE")
```

```
## End(Not run)
```

---

```
region_circle_plot    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)
```

---

```
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

*WITS data clean*


---

**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**

```
##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, [2](#)
- ELEnet16, [5](#)
- ELEnetList, [5](#)

cap\_lat\_lon, [2](#)  
Comtradrclinean, [2](#)  
core\_periphery\_weighted, [4](#)

ELEnet16, [5](#)  
ELEnetList, [5](#)

get.backbone, [6](#)

isEmpty, [7](#)  
ITN\_make\_plot, [17](#)  
ITN\_map\_plot, [18](#)  
ITNadjust, [7](#)  
ITNblock\_plot, [8](#)  
ITNblock\_se, [9](#)  
ITNcentrality, [10](#)  
ITNcentrality\_binary, [10](#)  
ITNcluster, [11](#)  
ITNcorr, [12](#)  
ITNdegdist, [12](#)  
ITNdynamic, [13](#)  
ITNhistdegdist, [14](#)  
ITNimvex, [15](#)  
ITNplotset, [15](#)  
ITNproperties, [16](#)

plot\_degree\_distribution, [18](#)

region\_circle\_plot, [20](#)  
RegionalBrokerage, [19](#)  
round\_df, [20](#)

WITSclean, [21](#)