Package: skynet (via r-universe)

August 30, 2026

Type Package

Version 1.4.4

Title Generates Networks from BTS Data

Maintainer Filipe Teixeira <Filipe.MarquesTeixeira@Ugent.be>

URL https://github.com/ropensci/skynet

Description A flexible tool that allows generating bespoke air transport statistics for urban studies based on publicly available data from the Bureau of Transport Statistics (BTS) in the United States <https://www.transtats.bts.gov/databases.asp?Z1qr_VQ=E&Z1qr_Qr5p=N8vn6v10&f7owrp6_VQF=D>.

License MIT + file LICENSE

BugReports https://github.com/ropensci/skynet/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Collate 'PowerLaw.R' 'createNodes.R' 'data_ODsample.R' 'data_airports.R' 'data_carriers.R' 'data_metro.R' 'disparityfilter.R' 'findAirport.R' 'fromto.R' 'netImport.R' 'netDir.R' 'netUnd.R' 'netInt.R' 'netPath.R' 'make_net_trip.R' 'skynet.R' 'nodeStats.R' 'import_t100.R' 'import_db1b.R' 'import_ontime.R' 'findCarrier.R' 'plotMap.R' 'skynet_example.R' 'bootnet.R' 'summary.R' 'download_db1b.R' 'download_t100.R' 'download_t100_int.R' 'download_ontime.R'

Depends R (>= 3.1.2)

Imports data.table, igraph, dplyr, ggplot2, ggrepel, stringr, httr, geosphere, leaflet, RCurl, maps

Suggests knitr, rmarkdown, testthat, kableExtra, covr

VignetteBuilder knitr

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2024-09-12 13:29:02 UTC

RemoteUrl https://github.com/ropensci/skynet

RemoteRef master

RemoteSha fb902f7b3677c84919e8776bda3890f861703568

Contents

aircraft_type . . . 3
airportCode . . . 3
airportCodeFull . . . 3
airportMaster . . . 3
boot_network . . . 4
carriers . . . 4
createNodes . . . 5
disparity_filter . . . 5
download_db1b . . . 6
download_ontime . . . 6
download_t100 . . . 7
download_t100_int . . . 8
find_airport . . . 8
find_carrier . . . 9
fit_power . . . 9
from_to_stats . . . 10
import_db1b . . . 10
import_ontime . . . 11
import_t100 . . . 12
make.netInt . . . 13
make_net_dir . . . 13
make_net_path . . . 14
make_net_trip . . . 15
make_net_und . . . 15
MetroFull . . . 16
MetroLookup . . . 17
net_map . . . 17
netImport . . . 18
node_stats . . . 18
nodeStatsMetro . . . 19
OD_Sample . . . 19
skynet_example . . . 20
summary.skynet . . . 20

Index **21**

aircraft_type *Aircraft type data*

Description

This data comes from the RITA/Transtats database

Format

A dataframe with 422 observations and 2 variables

airportCode *Airport Data - clean*

Description

USA airport data from the RITA/Transtats database

Format

A dataframe with 6435 observations and 5 variables

airportCodeFull *Airport Data - full*

Description

USA airport data from the RITA/Transtats database

Format

A dataframe with 6435 observations and 9 variables

airportMaster *Airport Data - master*

Description

World airport data from the RITA/Transtats database

Format

A dataframe with 13555 observations and 28 variables

boot_network *Network bootstrapping*

Description

Bootstraps a network and returns output containing three network statistics: Average Path Length, Transitivity, Mean Betweenness.

Usage

```
boot_network(g, n = 500, left_ci = 0.005, right_ci = 0.995)
```

Arguments

g	iGraph graph or skynet object.
n	Number of bootstraps to run. (500 default)
left_ci	Confidence interval left limit. (0.005 default)
right_ci	Confidence interval left limit (0.995 default)

Examples

```
## Not run:
boot_net(g, n = 500)


## End(Not run)
```

carriers *Carrier data*

Description

This data comes from the RITA/Transtats database

Format

A dataframe with 1738 observations and 2 variables

createNodes *Create Nodes*

Description

Creates nodes for SKYNET's functions. Despite being possible to use it individually, it's mainly meant to be used as a complimentary function.

Usage

```
createNodes(y)
```

Arguments

y	Data Frame

disparity_filter *Disparity Filter*

Description

Uses the Serrano's disparity filter (https://en.wikipedia.org/wiki/Disparity_filter_algorithm_of_weighted_network) to extract the backbone of the network in "Extracting the multiscale backbone of complex weighted networks"

Usage

```
disparity_filter(g, alpha = 0.003)
```

Arguments

g	igraph object
alpha	Alpha value. Default 0.003

Examples

```
## Not run:
netDir <- make.netDir(OD_Sample)
disparity_filter(netDir$gDir, alpha = 0.003)

## End(Not run)
```

download_db1b *Download Data from DB1B files*

Description

Downloads data from BTS/RITA/Transtats and imports it into R

Usage

```
download_db1b(y = NULL, q = NULL)
```

Arguments

y	year to be imported
q	quarter to be imported

Details

Coupon files can be found at https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FLM. Ticket files can be found at https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FKF.

Note: The BTS often changes the way we can access these files. So please be warned that this is still an experimental feature.

Examples

```
## Not run:

download_db1b(2010, 1)


## End(Not run)
```

download_ontime *Download On-Time*

Description

Download On-Time Performance Data directly from BTS/RITA/Transtats website raw data (prezipped file), for SKYNET's import function.

Usage

```
download_ontime(y, m, auto = TRUE)
```

Arguments

y	year to be imported
m	month to be imported
auto	Automatically assigns object

Examples

```
## Not run:

import_ontime(skynet_example("Ontime.csv"))


## End(Not run)
```

`download_t100` *Download Data from T100 files*

Description

Downloads data from BTS/RITA/Transtats and imports it into R

Usage

```
download_t100(y = NULL, type = NULL)
```

Arguments

y	year to be imported
type	"mkt" for Market, "seg" for Segment databases respectively

Details

Note: The BTS often changes the way we can access these files. So please be warned that this is still an experimental feature.

Examples

```
## Not run:

download_t100(2010, "mkt")


## End(Not run)
```

download_t100_int *Download Data from T100 international files*

Description

Downloads data from BTS/RITA/Transtats and imports it into R

Usage

```
download_t100_int(y = NULL, type = NULL)
```

Arguments

y	year to be imported
type	"mkt" for Market, "seg" for Segment databases respectively

Details

Note: The BTS often changes the way we can access these files. So please be warned that this is still an experimental feature.

Examples

```
## Not run:

download_t100_int(2010, "mkt")


## End(Not run)
```

find_airport *Find Airport function*

Description

Searches for airport information based on its IATA code or city name It will display multiple airports as it works with partial names.

Usage

```
find_airport(x)
```

Arguments

x	airport IATA code or city name

Examples

```
## Not run:
find_airport("Atlanta")

find_airport("ATL")

## End(Not run)
```

`find_carrier` *Find Carrier function*

Description

Searches for airport information based on its IATA code or city name

Usage

```
find_carrier(x)
```

Arguments

x	Carrier

Examples

```
## Not run:
find_carrier("United")

find_carrier("UA")

## End(Not run)
```

`fit_power` *Power Law*

Description

Plots power law fit

Usage

```
fit_power(graph)
```

Arguments

graph	iGraph object

Examples

```
## Not run:
netDir <- make.netDir(OD_Sample)
fit_power(netDir$gDir)

## End(Not run)
```

`from_to_stats` *From To function*

Description

Calculate edges weight from IATA Code

Usage

```
from_to_stats(x, y, orig)
```

Arguments

`x`	igraph object to query
`y`	origin airport IATA code
`orig`	"from" or "to" options

Examples

```
## Not run:
netDir <- make.netDir(OD_Sample)
from_to_stats(netDir$gDir, "JFK", orig = "from")

from_to_stats(netDir$gDir, "JFK", orig = "to")

## End(Not run)
```

`import_db1b` *Import Data from DB1B files*

Description

Imports data from BTS/RITA/Transtats files

Usage

```
import_db1b(c, t, zip = FALSE, auto = TRUE)
```

Arguments

c	Coupon csv file to be imported, in case of DB1B database
t	Ticket csv file to be imported, in case of DB1B database
zip	Should equal TRUE if original file comes from the BTS prezipped option.
auto	Automatically assigns object

Details

Coupon files can be found at https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FLM. Ticket files can be found at https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FKF. Both files should belong to the same year and same quarter. **Note**: We do recommend sparklyr to be used for larger sets of data. More information on variables to select and type of files to use can be found here

Examples

```
## Not run:

import_db1b(skynet_example("Coupon_2001Q1.csv"), skynet_example("Ticket_2001Q1.csv"))


## End(Not run)
```

---

import_ontime *Import on-time Data*

Description

Imports on-time Data directly from BTS/RITA/Transtats website raw data (prezipped file), for SKYNET's import function.

Usage

```
import_ontime(x, auto = TRUE)
```

Arguments

x	On-time csv (from zipped file)
auto	Automatically assigns object

Details

Files can be found here https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FGJ. More information on variables to select and type of files to use can be found here

Examples

```
## Not run:

import_ontime(skynet_example("Ontime_2011_1.csv"))


## End(Not run)
```

`import_t100` *Import T-100 Data*

Description

Imports T-100 Data directly from BTS/RITA/Transtats website raw data (prezipped file), for SKYNET's import function.

Usage

```
import_t100(x, nonsch = FALSE, auto = TRUE)
```

Arguments

x	T-100 csv
nonsch	Should equal TRUE to include non-scheduled flights
auto	Automatically assigns object

Details

Files can be found here https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FIL. More information on variables to select and type of files to use can be found here

Examples

```
## Not run:

import_t100(skynet_example("T100_2011_mkt.csv"))


## End(Not run)
```

make.netInt *International Data*

Description

Imports International data to complement to the DB1B data set. NOTE: When using this function, certain variables will be skewed as the T100 dataset does not contain all the data the DB1B dataset contains.

Usage

```
make.netInt(x = NULL, m = NULL, Q = NULL)
```

Arguments

x	T-100 International Segment csv file
m	Data set to merge with
Q	Desired T-100 Quarter. Should be equal to 1, 2, 3 or 4.

Examples

```
## Not run:

make.netInt(skynet_example("T100_2011_int.csv"), OD_Sample, 1)


## End(Not run)
```

make_net_dir *Directed network*

Description

Generates Directed Network with an iGraph **gDir** object, a Data Frame **netDir** and a Data Frame with Airport/Nodes statistics **nodes**.

Usage

```
make_net_dir(
  x,
  disp = FALSE,
  alpha = 0.003,
  cap = FALSE,
  pct = 10,
  carrier = FALSE,
  metro = FALSE
)
```

Arguments

x	Data frame
disp	Uses the Serrano's disparity filter (https://en.wikipedia.org/wiki/Disparity_filter_algorithm_of_weighted_network) to extract the backbone of the network.
alpha	Argument for disparity filter.
cap	Filters original data based on the edge weight.
pct	Argument for cap filter. Value should be imput as percentage.
carrier	Groups data per carrier and OD
metro	Groups data by metropolitan area

Examples

```
## Not run:
make_net_dir(OD_Sample)

# Apply Disparity Filter
make_net_dir(OD_Sample, disp = TRUE, alpha = 0.05)

# Apply Percentage Cap
make_net_dir(OD_Sample, cap = TRUE, pct = 20)

## End(Not run)
```

---

make_net_path *Path and OD Network*

Description

Generates an OD network and a Leg Count data frame(on demand)

Usage

```
make_net_path(x, leg = FALSE, zero = FALSE, carrier = FALSE)
```

Arguments

x	Data frame
leg	Generates Leg Count Data frame, based on Path taken.
zero	Displays percentage of 0 usd tickets
carrier	Groups data per airline For example, all passengers doing the BOS-ATL-LAX path, are summed by Air Carrier.

Examples

```
## Not run:
make_net_path(OD_Sample)

# Generate Leg Count
make_net_path(OD_Sample, leg = TRUE)

## End(Not run)
```

`make_net_trip` *Trip directed network*

Description

Generates Trip/Route based Directed Network with an iGraph **gDir** object, a Data Frame **netDir** and a Data Frame with Airport/Nodes statistics **nodes**. Returns type of trip: OD = Origin/Destination pair, OT = Origin/Transfer pair, TT = Transfer/Transfer pair, TD = Transfer/Destination pair

Usage

```
make_net_trip(x, carrier = FALSE)
```

Arguments

`x`	Data frame
`carrier`	Groups data per carrier and OD

Examples

```
## Not run:
make_net_trip(OD_Sample)

## End(Not run)
```

`make_net_und` *Undirected Network*

Description

Generates Undirected Network with an iGraph **gUnd** object, a Data Frame **netUnd** and a Data Frame with Airport/Nodes statistics **nodes**.

Usage

```
make_net_und(
  x,
  disp = FALSE,
  alpha = 0.003,
  cap = FALSE,
  pct = 10,
  merge = TRUE,
  carrier = FALSE,
  metro = FALSE
)
```

Arguments

x	Data frame
disp	Uses the Serrano's disparity filter (https://en.wikipedia.org/wiki/Disparity_filter_algorithm_of_weighted_network) to extract the backbone of the network.
alpha	Argument for disparity filter.
cap	Filters original data based on the edge weight.
pct	Argument for cap filter. Value should be imput as percentage.
merge	When set to FALSE, it keeps parallel edges instead of collapsing them and summing their weights.
carrier	Groups data per carrier and OD
metro	Groups data by metropolitan area

Examples

```
## Not run:
make_net_und(OD_Sample)

# Apply Disparity Filter
make_net_und(OD_Sample, disp = TRUE, alpha = 0.05)

# Apply Percentage Cap
make_net_und(OD_Sample, cap = TRUE, pct = 20)

## End(Not run)
```

MetroFull *Metro (Full) Data*

Description

This data comes from the RITA/Transtats database

Format

A dataframe with 5802 observations and 5 variables

`MetroLookup` *Metro Data*

Description

This data comes from the RITA/Transtats database

Format

A dataframe with 5802 observations and 2 variables

`net_map` *Plot Skynet*

Description

Creates OD ggplot2 generated maps from make.net functions Shows sample of 60

Usage

```
net_map(x, pct = 60)
```

Arguments

`x`	Skynet Object (generated by make_net_dir,make_net_und or make_net_path)
`pct`	percentage of edges to include

Examples

```
## Not run:
network <- make.netDir(OD_Sample)
net_map(network, pct = 10)

## End(Not run)
```

netImport *Import Data*

Description

Imports data from BTS/RITA/Transtats website File order doesn't matter, but it is recommended to name the files using the following syntax: *"Coupon YearQuarter.csv", "Ticket YearQuarter.csv", "T100 Year"*. Note: We do recommend sparklyr to be used for larger sets of data.

Usage

```
netImport(x = NULL, y = NULL)
```

Arguments

x	First csv file to be imported, in case of DB1B database, or in case of using the T-100 database, the only file to be included.
y	Second csv file to be imported.

Examples

```
## Not run:

netImport(skynet_example("Coupon_2001Q1.csv"), skynet_example("Ticket_2001Q1.csv"))


## End(Not run)
```

node_stats *Get node info*

Description

Creates node statistics Generates Number of Passenger Arrivals, Departures and Transfers

Usage

```
node_stats(x)
```

Arguments

x	Data Frame to extract information from

Examples

```
## Not run:

node_stats(OD_Sample)


## End(Not run)
```

`nodeStatsMetro` *Create Metro Nodes*

Description

Create Metro Nodes

Usage

```
nodeStatsMetro(y)
```

Arguments

y	Data Frame

Examples

```
## Not run:

nodeStatsMetro(OD_Sample)


## End(Not run)
```

`OD_Sample` *Sample OD data*

Description

Sample data to use with SKYNET functions

Format

A dataframe with 500.000 observations and 19 variables

skynet_example *Get path to skynet examples*

Description

To access csv examples from SKYNET

Usage

```
skynet_example(path = NULL)
```

Arguments

path	File name.

Examples

```
## Not run:
skynet_example()
skynet_example("Coupon 2001Q1.csv")

## End(Not run)
```

summary.skynet *Displays a summary of a skynet object*

Description

Displays a summary of a skynet object

Usage

```
## S3 method for class 'skynet'
summary(object, ...)
```

Arguments

object	skynet object to summarise
...	other arguments ignored (for compatibility with generic)

Examples

```
net <- make_net_dir(OD_Sample)
summary(net)
```

Index

aircraft_type, 3
airportCode, 3
airportCodeFull, 3
airportMaster, 3

boot_network, 4

carriers, 4
createNodes, 5

disparity_filter, 5
download_db1b, 6
download_ontime, 6
download_t100, 7
download_t100_int, 8

find_airport, 8
find_carrier, 9
fit_power, 9
from_to_stats, 10

import_db1b, 10
import_ontime, 11
import_t100, 12

make.netInt, 13
make_net_dir, 13
make_net_path, 14
make_net_trip, 15
make_net_und, 15
MetroFull, 16
MetroLookup, 17

net_map, 17
netImport, 18
node_stats, 18
nodeStatsMetro, 19

OD_Sample, 19

skynet_example, 20
summary.skynet, 20