

Package ‘spacejamr’

September 12, 2021

Title Simulate Spatial Bernoulli Networks

Version 0.1.1

Maintainer Darren Colby <dscolby17@gmail.com>

Description Social network analysis is becoming commonplace in many social science disciplines, but access to useful network data, especially among marginalized populations, still remains a formidable challenge. This package mitigates that problem by providing tools to simulate spatial Bernoulli networks as proposed in Carter T. Butts (2002, ISBN:978-0-493-72676-2), "Spatial models of large-scale interpersonal works. Using this package, network analysts can simulate a spatial point process or sequence with a given number of nodes inside a geographical boundary and estimate the probability of a tie formation between all node pairs. When simulating a network, an analyst can choose between five spatial interaction functions. The package also enables quick comparison of summary statistics for simulated networks and provides simple to use plotting methods for its classes that return plots which can be further refined with the 'ggplot2' package.

License MIT + file LICENSE

URL <https://github.com/dscolby/spacejamr>

BugReports <https://github.com/dscolby/spacejamr/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports crsuggest, dplyr, ggraph, ggplot2, igraph, magrittr, sf, spatstat.core, spatstat.geom, ggthemes, methods

Depends R (>= 3.5)

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Darren Colby [aut, cre] (<<https://orcid.org/0000-0001-8468-2755>>)

Repository CRAN

Date/Publication 2021-09-12 21:50:02 UTC

R topics documented:

as.spacejamr	2
compare_networks	3
NetSim	4
plot.NetSim	5
plot.PointSim	6
plot.spacejamr	7
PointSim	8
print.NetSim	9
print.PointSim	10
print.spacejamr	10
RI	11
spacejamr	12
summary.NetSim	12
summary.PointSim	13
summary.spacejamr	14
Index	15

as.spacejamr	<i>Initialize a new spacejamr object</i>
--------------	--

Description

Creates a new spacejamr object that for further analysis

Usage

```
as.spacejamr(path)
```

Arguments

path the path to a shapefile as a string.

Details

The returned spacejamr object will contain a window object containing a geographical boundary and its coordinate reference system. Since simulating a spatial point process or sequece requires a projected coordinate reference system, this method will automatically look for the most appropriate projected coordinate reference system and project the geographical boundary to that system. Therefore, this function may take some time to run but is necessary for later steps to simulate a spatial Bernoulli network.

Value

a spacejamr object containing two items. window: object of class 'owin' that stores geographical boundaries. crs: integer value referring to the coordinate reference system of the geographical boundaries.

Author(s)

Darren Colby
Email: dscolby@gmail.com

Examples

```
ri <- as.spacejamr(system.file("shape/ri.shp", package = "spacejamr"))
```

compare_networks	<i>Compare statistics from two simulated networks</i>
------------------	---

Description

Generates summary statistics for two simulated networks of class NetSim.

Usage

```
compare_networks(net1, net2)
```

Arguments

net1	a NetSim object
net2	a NetSim object

Details

This function enables a quick comparison of two simulated networks of class NetSim. This allows the user to see how different a network simulated from a Halton sequence is from network simulated from a Poisson point process with the same number of points in the same geographical window. Similarly, it enables comparison of networks generated from different spatial interaction functions. The density, mean degree, mean closeness, mean betweenness, and the size of the largest connected component are computed for each network.

Value

A dataframe with two rows and five columns. The first row displays statistics from the first network and the second row displays statistics from the second network. The values in each column are as follows. Density: the density of the network. Mean Degree: the average degree of all nodes in the network. Mean Closeness: the average closeness centrality of all nodes in each network. Mean Betweenness: the average betweenness centrality for the nodes in a network. Largest Component Size: The number of nodes in the largest connected component of a network.

Author(s)

Darren Colby
 Email: dscolby17@gmail.com

Examples

```
# load data
data("RI")

# Simulate point processes
points1 <- PointSim(10, RI)
points2 <- PointSim(20, RI)

# Create two networks
net1 <- NetSim(points1)
net2 <- NetSim(points2)

compare_networks(net1, net2)
```

 NetSim

Simulate a network from a point process or sequence

Description

Simulates a spatial Bernoulli network from a NetSim object using one of six probability law distributions.

Usage

```
NetSim(point_sim, sif, base_prob, scale, threshold, power)
```

Arguments

point_sim	a PointSim object
sif	the spatial interaction function to use. Use attenuated to use an attenuated power law; arctan to use an arctangent probability law; decay to use an exponential decay law; or logistic to use a logistic probability law. Default is a standard power law function.
base_prob	the theoretical probability that two nodes (points) with distance 0 share a tie. Default is 0.9.
scale	a coefficient to multiply the distance by. Default is 1.
threshold	if two node exceed this probability, they will be coded as having a tie. Default is 0.5.
power	the exponent at which tie probability decays. Default is -2.8.

Details

The algorithm proceeds in three steps. First, it calculates the distance between simulated points from a PointSim object. Then it calculates the distance between all pairs of points. Finally, it uses a spatial interaction function to calculate that any two point share a tie. If the threshold is exceeded, a tie is created.

Value

A network object of class 'NetSim' and 'igraph' that can be manipulated using the igraph' package.

Author(s)

Darren Colby
Email: dscolby17@gmail.com

References

Butts, Carter T. Spatial models of large-scale interpersonal networks. Dissertation. (2002).

Examples

```
# Load spacejamr object
data("RI")

ri_points <- PointSim(points = 10, window = RI, seed = 42)
power_law <- NetSim(ri_points, base_prob = 0.92, scale = 1, threshold = 0.5,
                    power = -2.4)
```

plot.NetSim

Plot a simulated network from a NetSim object

Description

This can take either a PowerLawNetwork or APLNetwork object as input, both of which are children of the NetSim class.

Usage

```
## S3 method for class 'NetSim'
plot(
  x,
  y,
  ...,
  layout = "stress",
  title = "Network Simulation",
  node_color = "red",
  edge_color = "blue"
)
```

Arguments

x	a NetSim graph
y	ignored.
...	ignored.
layout	a layout to display the graph. Layout must be a valid string. from the ggraph package. Default is "stress".
title	an optional title.
node_color	a color for the nodes. Default is blue.
edge_color	a color for the edges. Default is red.

Details

This method returns a ggraph object, which can be further refined using standard ggraph and ggplot facilities.

Value

A plot of classes 'ggraph' 'gg' and 'ggplot'

Author(s)

Darren Colby
Email:dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

ri_points <- PointSim(points = 10, window = RI, seed = 42)
spl_points <- NetSim(ri_points, base_prob = 0.92, scale = 1, threshold = 0.5,
                    power = -2.4)

plot(spl_points)
```

plot.PointSim *Plot simulated points from a PointSim object*

Description

Plots the results of points simulated in a PointProcess or HaltonSeq class, whcih obht inherit methods from the PointSim class.

Usage

```
## S3 method for class 'PointSim'
plot(x, y, ..., title = "Simulated Points", color = "red")
```

Arguments

x	an object of class PointSim or one of its child classes
y	ignored.
...	ignored
title	an optional title. Default is "Simulated Points".
color	an optional color for the simulated points. Default is red.

Details

The returned plot can be refined with standard ggplot2 functions

Value

A plot of classes 'gg' and 'ggplot'

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

ri_points <- PointSim(points = 10, window = RI, seed = 42)
plot(ri_points)
```

plot.spacejamr	<i>Plot the spatial extent of a spacejamr object</i>
----------------	--

Description

Plot method for the spacejamr class

Usage

```
## S3 method for class 'spacejamr'
plot(x, y, ..., title = "Spatial Window", fill = "blue")
```

Arguments

x	an object of class spacejamr.
y	ignored.
...	ignored.
title	an optional title for the plot. Default is "Spatial Window".
fill	an optional fill for the plot. Default is blue.

Details

The returned plot can be refined with standard ggplot2 functions

Value

A plot of classes 'gg' and 'ggplot'

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

plot(RI)
```

PointSim

Simulate a spatial point process or sequence

Description

Creates a new Poisson Point Process in a spacejamr object

Usage

```
PointSim(points, window, type, seed)
```

Arguments

points	the number of points to simulate
window	a spacejamr object to use as the spatial extent
type	the type of simulation. poisson_process simulates a spatial Poisson process. halton simulates a Halton sequence where the maximal number of simulated points is the first argument.
seed	an optional seed

Value

An object of class PointSim that contains a geographical window of class 'owin'. Within this window are four objects. n: the number of simulated points. x: the x coordinates of the simulated points. y: the y coordinates of the simulated points. markformat: an empty place holder.

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

# Poisson process
ri_points <- PointSim(points = 10, window = RI, seed = 42)
```

`print.NetSim` *Print information from a NetSim object*

Description

Plots a NetSim object and returns a ggraph object

Usage

```
## S3 method for class 'NetSim'
print(x, ...)
```

Arguments

x a NetSim object
... ignored.

Value

No return value, called for side effects

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Create spacejamr object
data("RI")

ri_points <- PointSim(points = 10, window = RI, seed = 42)
spl_points <- NetSim(ri_points, base_prob = 0.92, scale = 1, threshold = 0.5,
                    power = -2.4)
print(spl_points)
```

print.PointSim *Print information from a PointSim class*

Description

Print method for both the PointProcess and HaltonSeq classes, which inherit methods from the PointSim class.

Usage

```
## S3 method for class 'PointSim'  
print(x, ...)
```

Arguments

x a PointSim object or a child object
... ignored.

Value

No return value, called for side effects

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object  
data("RI")  
  
ri_points <- PointSim(points = 10, window = RI, seed = 42)  
print(ri_points)
```

print.spacejamr *Print information from a spacejamr instance*

Description

Print method for the spacejamr class

Usage

```
## S3 method for class 'spacejamr'  
print(x, ...)
```

Arguments

x a spacejamr object
... ignored.

Details

Provides a wrapper for the print.owin method in the spatstats.geom package.

Value

No return value, called for side effects

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

print(RI)
```

RI *Geographical boundary of Rhode Island.*

Description

A dataset containing Mexico's border and the coordinate reference system in which Rhode Island's border is projected.

Usage

RI

Format

A spacejamr object with an owin window object and an integer denoting the projected coordinate reference system:

window geographical boundary of Rhode Island

crs the EPSG code for the coordinate reference system of the window ...

Source

<https://www.rigis.org/datasets/edc::state-boundary-1989/about>

`spacejamr`*spacejamr: A package for simulating spatial Bernoulli networks*

Description

The spacejamr package provides three main types of functions: spacejamr, PointSim, and NetSim. They create new objects used in subsequent steps of network simulation, simulate spatial point processes, and create network objects. There are also plot, print, and summary methods for all objects and a compare_networks function to quickly compare two simulated networks.

Functions

APLNetwork
compare_networks
HaltonSeq
RI
plot.NetSim
plot.PointSim
plot.spacejamr
PointProcess
PowerLawNetwork
print.NetSim
print.PointSim
print.spacejamr
spacejamr
summary.NetSim
summary.spacejamr
summary.PointSim

Author(s)

Darren Colby
Email: dscolby17@gmail.com
ORCID: 0000-0001-8468-2755

`summary.NetSim`*Summary of NetSim graphs*

Description

Prints a summary of a NetSim object

Usage

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

Arguments

object a NetSim object
... ignored.

Value

No return value, called for side effects

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

ri_points <- PointSim(points = 10, window = RI, seed = 42)
spl_points <- NetSim(ri_points, base_prob = 0.92, scale = 1, threshold = 0.5,
                    power = -2.4)
summary(spl_points)
```

summary.PointSim *Display summary information from a PointSim instance*

Description

Prints a summary of information from either a PointProcess or HaltonSeq object, which are both child classes of the PointSim class.

Usage

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

Arguments

object a PointSim object
... ignored.

Value

No return value, called for side effects

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

ri_points <- PointSim(points = 10, window = RI, seed = 42)
summary(ri_points)
```

summary.spacejamr *Print summary information of a spacejamr instance*

Description

Summary method for the spacejamr class

Usage

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

Arguments

object	a spacejamr object
...	ignored.

Details

Provides a wrapper for the summary.owin method in the spatstats.geom package.

Value

No return value, called for side effects

Author(s)

Darren Colby
Email: dscolby17@gmail.com

Examples

```
# Load spacejamr object
data("RI")

summary(RI)
```

Index

* datasets

RI, [11](#)

as.spacejamr, [2](#)

compare_networks, [3](#)

NetSim, [4](#)

plot.NetSim, [5](#)

plot.PointSim, [6](#)

plot.spacejamr, [7](#)

PointSim, [8](#)

print.NetSim, [9](#)

print.PointSim, [10](#)

print.spacejamr, [10](#)

RI, [11](#)

spacejamr, [12](#)

summary.NetSim, [12](#)

summary.PointSim, [13](#)

summary.spacejamr, [14](#)