

# Package ‘aniSNA’

March 25, 2024

**Type** Package

**Title** Statistical Network Analysis of Animal Social Networks

**Version** 1.1.1

**Date** 2024-03-22

**Description** Obtain network structures from animal GPS telemetry observations and statistically analyse them to assess their adequacy for social network analysis. Methods include pre-network data permutations, bootstrapping techniques to obtain confidence intervals for global and node-level network metrics, and correlation and regression analysis of the local network metrics.

**License** MIT + file LICENSE

**Imports** dplyr, ggplot2, graphics, igraph, lubridate, magrittr, parallel, plotrix, Rcpp, reshape, rlang, stats, stringr, utils

**LinkingTo** Rcpp

**RoxygenNote** 7.2.1

**Encoding** UTF-8

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Depends** R (>= 2.10)

**LazyData** true

**LazyDataCompression** xz

**NeedsCompilation** yes

**Author** Prabhleen Kaur [aut, cre],  
Michael Salter-Townshend [ctb]

**Maintainer** Prabhleen Kaur <prabhleen.kaur.ucd@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-03-25 11:20:02 UTC

## R topics documented:

aniSNA	2
bootstrapped_difference_pvalues	3
correlation_analyze	4
distance_radian_coordinates	5
elk_2010_permutations	6
elk_all_interactions_2010	6
elk_data_2010	7
elk_interactions_2010	7
elk_network_2010	8
get_coordinates_in_radian	8
get_interactions	9
get_network_summary	9
get_spatial_threshold	10
global_CI	11
global_width_CI	12
interacting_pairs	13
network_from_interactions	14
node_level_CI	14
obtain_bootstrapped_samples	16
obtain_network_subsamples	17
obtain_permuted_network_versions	17
plot.bootstrapped_pvalue_matrix	18
plot.list_correlation_matrices	19
plot.list_node_level_CI	20
plot.list_permuted_networks	20
plot.list_regression_matrices	21
plot.Subsampled_Network_Metrics	22
plot.Subsampled_Permuted_Network_Metrics	23
plot.Width_CI_matrix	24
plot_network	25
regression_slope_analyze	26
subsampled_network_metrics	27
subsampled_permuted_network_metrics	28
<b>Index</b>	<b>30</b>

---

 aniSNA

*aniSNA*


---

### Description

Obtain network structures from animal GPS telemetry observations and statistically analyse them to assess their adequacy for social network analysis. Methods include pre-network data permutations, bootstrapping techniques to obtain confidence intervals for global network metrics, and correlation and regression analysis of the local network metrics.

**Author(s)**

Prabhleen Kaur

---

 bootstrapped\_difference\_pvalues

*To obtain two non-overlapping bootstrapped versions and obtain p-values for the significance of difference between them*

---

**Description**

To obtain two non-overlapping bootstrapped versions and obtain p-values for the significance of difference between them

**Usage**

```
bootstrapped_difference_pvalues(
  network,
  n_versions = 1000,
  seed = 12345,
  n.iter = 10,
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
  diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
  igraph::transitivity(x))
)
```

**Arguments**

network	An igraph object
n_versions	Number of bootstrapped versions to be used (default = 1000)
seed	seed number
n.iter	Number of iterations at each level
network_metrics_functions_list	A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = c("edge_density" = function(x) igraph::edge_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))

**Value**

A matrix of p-values whose rows correspond to the sub-sample size and columns correspond to the chosen network metric. The sub-sample size values (corresponding to rows) occur in multiples of 5 and range from 5 to a maximum of half the number of nodes in the network

**Examples**

```
data(elk_network_2010)
mean_pvalue_matrix <- bootstrapped_difference_pvalues(elk_network_2010, n_versions = 100)
plot(mean_pvalue_matrix)
```

---

correlation\_analyze    *To perform correlation analysis for node-level network metrics*

---

**Description**

To perform correlation analysis for node-level network metrics

**Usage**

```
correlation_analyze(
  network,
  n_simulations = 10,
  subsampling_proportion = c(0.1, 0.3, 0.5, 0.7, 0.9),
  network_metrics_functions_list = c(degree = function(net, sub_net) igraph::degree(net,
  v = igraph::V(sub_net)$name), strength = function(net, sub_net) igraph::strength(net,
  v = igraph::V(sub_net)$name), betweenness = function(net, sub_net)
  igraph::betweenness(net, v = igraph::V(sub_net)$name), clustering_coefficient =
  function(net, sub_net) igraph::transitivity(net, type = "local", vids =
  igraph::V(sub_net)$name), eigenvector_centrality = function(net, sub_net)
  igraph::eigen_centrality(net)$vector[igraph::V(sub_net)$name])
)
```

**Arguments**

**network**            An igraph graph object consisting of observed network

**n\_simulations**    Number of sub-samples to be obtained at each level

**subsampling\_proportion**  
                     A vector depicting proportions of sub-sampled nodes

**network\_metrics\_functions\_list**  
                     A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Each function definition should include two parameters, one for the main network and another one for the subnetwork. See default example. Default = c("degree" = function(net, sub\_net) igraph::degree(net, v = igraph::V(sub\_net)\$name), "strength" = function(net, sub\_net) igraph::strength(net, v = igraph::V(sub\_net)\$name), "betweenness" = function(net, sub\_net) igraph::betweenness(net, v = igraph::V(sub\_net)\$name), "clustering\_coefficient" = function(net, sub\_net) igraph::transitivity(net, type = "local", vids = igraph::V(sub\_net)\$name), "eigenvector\_centrality" = function(net, sub\_net) igraph::eigen\_centrality(net)\$vector[igraph::V(sub\_net)\$name])

**Value**

A list of network metrics of class `list_correlation_matrices`. Each element of list is a matrix whose columns correspond to `subsampling_proportion` and rows correspond to `n_simulations`. The entries of the matrix provide value of correlation between the nodes in full network and the sub-sampled network for the corresponding metric.

**Examples**

```
data(elk_network_2010)
elk_correlation_analysis <- correlation_analyze(elk_network_2010)
plot(elk_correlation_analysis)
```

---

distance\_radian\_coordinates

*Calculate distance between two pairs of radian coordinates*

---

**Description**

Calculate distance between two pairs of radian coordinates

**Usage**

```
distance_radian_coordinates(latf, lonf, latt, lont)
```

**Arguments**

latf	latitude from
lonf	longitude from
latt	latitude to
lont	longitude to

**Value**

distance value in meters

---

`elk_2010_permutations` *A list of 100 igraph objects obtained by permuting the raw elk\_data\_2010 and obtaining network from those*

---

**Description**

A list of 100 igraph objects obtained by permuting the raw elk\_data\_2010 and obtaining network from those

**Usage**

```
elk_2010_permutations
```

**Format**

A list of 100 igraph objects

**Examples**

```
data(elk_2010_permutations)
```

---

`elk_all_interactions_2010`  
*Dataset of all possible interactions from elk\_data\_2010*

---

**Description**

Dataset of all possible interactions from elk\_data\_2010

**Usage**

```
elk_all_interactions_2010
```

**Format**

A dataframe with 7615 rows and 5 variables

**Animal\_A** First animal ID

**Animal\_B** Second animal ID

**Timestamp\_A** Observation timestamp of first animal

**Timestamp\_B** Observation timestamp of second animal

**distance** Distance in metres between the two animals

**Examples**

```
data(elk_all_interactions_2010)
```

---

elk\_data\_2010      *Data to showcase functions in our package*

---

**Description**

Contains GPS telemetry observations of the species elk in year 2010

**Usage**

```
elk_data_2010
```

**Format**

A dataframe with 123568 rows and 4 variables:

**animal\_id** Unique ID of individuals in the observed sample

**datetime** Date and timestamp of the observation

**latitude\_rad** Latitude of individual observation in radians

**longitude\_rad** Longitude of individual observation in radians

**Examples**

```
data(elk_data_2010)
```

---

elk\_interactions\_2010      *Dataset of interactions from elk\_data\_2010 using first mode as the spatial threshold*

---

**Description**

Dataset of interactions from elk\_data\_2010 using first mode as the spatial threshold

**Usage**

```
elk_interactions_2010
```

**Format**

A dataframe with 2393 rows and 5 variables

**Animal\_A** First animal ID

**Animal\_B** Second animal ID

**Timestamp\_A** Observation timestamp of first animal

**Timestamp\_B** Observation timestamp of second animal

**distance** Distance in metres between the two animals

**Examples**

```
data(elk_interactions_2010)
```

---

```
elk_network_2010    An igraph object depicting the network obtained from
                    elk_interactions_2010
```

---

**Description**

An igraph object depicting the network obtained from elk\_interactions\_2010

**Usage**

```
elk_network_2010
```

**Format**

An igraph object with 57 nodes and 114 edges

**Examples**

```
igraph::E(elk_network_2010)
```

---

```
get_coordinates_in_radian
    To convert latitude and longitude values from degrees to radians
```

---

**Description**

To convert latitude and longitude values from degrees to radians

**Usage**

```
get_coordinates_in_radian(species_raw)
```

**Arguments**

`species_raw` A DataFrame consisting of GPS observations. The DataFrame must have a "latitude" column and a "longitude" column whose values are specified in degrees.

**Value**

The same DataFrame that has been passed as the argument with two additional columns namely "latitude\_rad" and "longitude\_rad"



---

get\_interactions      *To obtain interactions from raw GPS observations*

---

### Description

To obtain interactions from raw GPS observations

### Usage

```
get_interactions(species_raw, temporal_thresh = 7, spatial_thresh, n_cores = 1)
```

### Arguments

species_raw	A DataFrame consisting of GPS observations. It should have at least four columns namely "animal_id", "datetime", "latitude_rad", and "longitude_rad". "latitude_rad", and "longitude_rad" are latitude and longitude values in radians respectively. See function "get_coordinates_in_radian" to get these values.
temporal_thresh	Temporal threshold in minutes with default 7 minutes
spatial_thresh	The maximum distance in meters within which two animals are considered interacting
n_cores	Number of cores for parallel processing with default 1

### Value

A dataframe consisting of five columns. The first two columns contain animal ids, third and fourth column contain timestamp of their observations and the final column contains the distance between the two individuals.

### Examples

```
data(elk_data_2010)
get_interactions(elk_data_2010, temporal_thresh = 7, spatial_thresh = 15)
```

---

get\_network\_summary      *Calculates and prints network summary statistics*

---

### Description

Calculates and prints network summary statistics

### Usage

```
get_network_summary(network)
```

**Arguments**

`network` An undirected network with nodes representing animal IDs and edges representing associations between them.

**Value**

No return value, called for side effects. The function prints values of network metrics to the console.

**Examples**

```
data(elk_network_2010)
get_network_summary(elk_network_2010)
```

---

`get_spatial_threshold` *To obtain spatial threshold for calculating interactions from raw GPS observations. The threshold is obtained as the distance interval that captures maximum number of inter-individual interactions.*

---

**Description**

To obtain spatial threshold for calculating interactions from raw GPS observations. The threshold is obtained as the distance interval that captures maximum number of inter-individual interactions.

**Usage**

```
get_spatial_threshold(species_interactions, interval_size)
```

**Arguments**

`species_interactions`  
A dataframe consisting of individual interactions within maximum possible distance

`interval_size` Minimum interval size within which the number of interactions should be calculated

**Value**

Spatial threshold in meters

**Examples**

```
data(elk_all_interactions_2010)
get_spatial_threshold(elk_all_interactions_2010, interval_size = 2)
```

---

global_CI	<i>To obtain confidence intervals around the observed global network statistics</i>
-----------	---

---

### Description

To obtain confidence intervals around the observed global network statistics

### Usage

```
global_CI(
  network,
  n_versions = 100,
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
    diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
    igraph::transitivity(x)),
  CI_size = 0.95
)
```

### Arguments

network	An igraph object consisting of observed network.
n_versions	Number of bootstrapped versions to be used. (default = 100)
network_metrics_functions_list	A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = c("edge_density" = function(x) igraph::edge_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))
CI_size	Size of confidence interval. Default is 0.95 that generates a 95% confidence interval.

### Value

A DataFrame consisting of three columns. The first column contains the value of observed network metric, the second and third column represent the lower and upper limit of 95

### Examples

```
data(elk_network_2010)
global_CI(elk_network_2010, n_versions = 100,
network_metrics_functions_list = c("edge_density" = function(x) igraph::edge_density(x),
"diameter" = function(x) igraph::diameter(x, weights = NA),
"transitivity" = function(x) igraph::transitivity(x)))
```

---

global_width_CI	<i>To obtain width of confidence intervals for global network metrics using bootstrapped versions at each level of sub-sampling</i>
-----------------	---

---

### Description

To obtain width of confidence intervals for global network metrics using bootstrapped versions at each level of sub-sampling

### Usage

```
global_width_CI(
  network,
  n_versions = 100,
  seed = 12345,
  n.iter = 10,
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
  diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
  igraph::transitivity(x)),
  scaled_metrics = NULL,
  CI_size = 0.95
)
```

### Arguments

network	An igraph object consisting of observed network.
n_versions	Number of bootstrapped versions to be used. (default = 100)
seed	seed number
n.iter	Number of iterations at each level. (default = 10)
network_metrics_functions_list	A list consisting of function definitions of the global network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = c("edge_density" = function(x) igraph::edge_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))
scaled_metrics	Optional. A vector subset of the names of functions in network_metrics_functions_list with the metrics that should be scaled. For example scaled_metrics = c("diameter")
CI_size	Size of confidence interval. Default is 0.95 that generates a 95% confidence interval.

### Value

A matrix of class Width\_CI\_matrix containing width of Confidence Intervals where each row corresponds to the sub-sample size and columns correspond to the chosen network metric. Sub-sample size values occur in multiples of 10 and range from 10 to maximum multiple of 10 less than or equal to the number of nodes in the network.

## Examples

```
data(elk_network_2010)
width_CI_elk <- global_width_CI(elk_network_2010, n_versions = 100)
plot(width_CI_elk)
```

---

interacting\_pairs      *Function to obtain pairs of interacting animals*

---

## Description

Function to obtain pairs of interacting animals

## Usage

```
interacting_pairs(
  i,
  datetime,
  latitude,
  longitude,
  temporal_thresh,
  spatial_thresh
)
```

## Arguments

i	Index of the animal
datetime	DateTime vector
latitude	latitude vector
longitude	longitude vector
temporal_thresh	time threshold in minutes
spatial_thresh	spatial threshold in meters

## Value

A matrix consisting of two row. The first row corresponds to the interacting indices and the second row to the respective distances.

---

network\_from\_interactions

*Function to obtain a network structure from interactions dataframe*

---

### Description

Function to obtain a network structure from interactions dataframe

### Usage

```
network_from_interactions(species_raw, interactions, n_cores = 1)
```

### Arguments

species_raw	A dataframe consisting of raw GPS observations. It should have at least four columns namely "animal_id", "datetime", "latitude_rad", and "longitude_rad". "latitude_rad", and "longitude_rad" are latitude and longitude values in radians respectively. See function "get_coordinates_in_radian" to get these values.
interactions	A dataframe of interactions obtained from raw GPS observations using the function "get_interactions"
n_cores	Number of cores for parallel processing, default is 1

### Value

An object of class igraph

### Examples

```
data(elk_data_2010, elk_interactions_2010)
network_from_interactions(elk_data_2010, elk_interactions_2010)
```

---

node\_level\_CI

*To obtain confidence intervals for node-level network metrics*

---

### Description

To obtain confidence intervals for node-level network metrics

**Usage**

```
node_level_CI(
  network,
  n_versions = 100,
  network_metrics_functions_list = c(degree = igraph::degree, strength =
  igraph::strength, betweenness = igraph::betweenness, clustering_coefficient =
  function(x) {
    trans <- igraph::transitivity(x, type = "local", vids =
    igraph::V(x), isolates = "zero")
    names(trans) <- igraph::V(x)$name

    return(trans)
  }, eigenvector_centrality = function(x)
  igraph::eigen_centrality(x)$vector),
  n_cores = 1,
  CI_size = 0.95
)
```

**Arguments**

network	An igraph graph object consisting of observed network.
n_versions	Number of bootstrapped versions to be used. (default = 100)
network_metrics_functions_list	A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Each function definition should include two parameters, one for the main network and another one for the subnetwork. See default example. <code>network_metrics_functions_list = c("degree" = igraph::degree, "strength" = igraph::strength, "betweenness" = igraph::betweenness, "clustering_coefficient" = function(x) { trans &lt;- igraph::transitivity(x, type = "local", isolates = "zero") names(trans) &lt;- igraph::V(x)\$name; return(trans) }, "eigenvector_centrality" = function(x) igraph::eigen_centrality(x)\$vector )</code>
n_cores	Number of cores for parallel processing with default 1.
CI_size	Size of confidence interval. Default is 0.95 that generates a 95% confidence interval.

**Value**

A list of dataframes of class `list_node_level_CI`. Each element of list is a dataframe having five columns and having number of rows equal to number of nodes in the network. The five columns correspond to `node_number`, `node_name`, `metric_value`, `lower_CI`, `upper_CI`. `correlation` correspond to sub-sampling\_proportion and rows correspond to `n_simulations`. The entries of the matrix provide value of correlation between the nodes in full network and the sub-sampled network for the corresponding metric.

**Examples**

```
data(elk_network_2010)
```

```
elk_node_level_CI <- node_level_CI(elk_network_2010)
plot(elk_node_level_CI)
```

---

obtain\_bootstrapped\_samples

*To obtain bootstrapped versions of a network's adjacency matrix*

---

### Description

To obtain bootstrapped versions of a network's adjacency matrix

### Usage

```
obtain_bootstrapped_samples(  
  network,  
  n_nodes = igraph::gorder(network),  
  n_versions = 1000,  
  seed = 12345  
)
```

### Arguments

network	An igraph object
n_nodes	Number of nodes to be selected in bootstrapped versions (default : All nodes)
n_versions	Number of bootstrapped versions required
seed	seed number

### Value

A list of class `bootstrapped_pvalue_matrix` consisting of two elements. The first element contains the adjacency matrix of the original network and the second element contains bootstrapped versions of the adjacency matrices.

### Examples

```
data(elk_network_2010)
obtain_bootstrapped_samples(elk_network_2010, n_versions = 100)
```



---

`obtain_network_subsamples`*To obtain sub-networks of the observed network*

---

**Description**

To obtain sub-networks of the observed network

**Usage**

```
obtain_network_subsamples(  
  network,  
  n_subsamples = 1,  
  subsampling_proportion = 0.5  
)
```

**Arguments**

<code>network</code>	An igraph object
<code>n_subsamples</code>	Number of sub-networks to be obtained. (default = 1)
<code>subsampling_proportion</code>	A value depicting the level (in proportion) at which sub-samples to be taken. (default = 0.5). This value should lie between 0 and 1 depicting the proportion of observed nodes to be included in the sub-network.

**Value**

A list of size `n_subsamples`, where each element of the list is an igraph object representing a sub-network of the observed network.

**Examples**

```
data(elk_network_2010)  
obtain_network_subsamples(elk_network_2010, 1, 0.5)
```

---

`obtain_permuted_network_versions`*Function to obtain permuted networks from raw datastream*

---

**Description**

Function to obtain permuted networks from raw datastream

**Usage**

```

obtain_permuted_network_versions(
  species_raw,
  temporal_thresh,
  spatial_thresh,
  n_permutations,
  n_cores = 1
)

```

**Arguments**

species\_raw     A dataframe consisting of raw GPS observations

temporal\_thresh     Temporal threshold in minutes

spatial\_thresh    Spatial threshold

n\_permutations    Number of permuted versions to obtain

n\_cores            Number of cores for parallel processing with default 1

**Value**

An object of class "list\_permuted\_networks" of size n\_permutations where each element is a network of class igraph obtained by permuting raw datastream

**Examples**

```

data(elk_data_2010)
permuted_versions <- obtain_permuted_network_versions(elk_data_2010,
temporal_thresh = 7, spatial_thresh = 15, n_permutations = 10, n_cores = 2)

```

---

```
plot.bootstrapped_pvalue_matrix
```

*To plot the results obtained from bootstrapped\_difference\_pvalues function*

---

**Description**

To plot the results obtained from bootstrapped\_difference\_pvalues function

**Usage**

```

## S3 method for class 'bootstrapped_pvalue_matrix'
plot(x, ...)

```

**Arguments**

x                    A matrix of p-values obtained from bootstrapped\_difference\_pvalues function  
...                   Further arguments are ignored.

**Value**

No return value, called for side effects. The plot shows p-values between 0 and 1 corresponding to each sample size.

**Examples**

```
data(elk_network_2010)
mean_pvalue_matrix <- bootstrapped_difference_pvalues(elk_network_2010, n_versions = 100)
plot(mean_pvalue_matrix)
```

---

plot.list\_correlation\_matrices

*To plot correlation analysis results*

---

**Description**

To plot correlation analysis results

**Usage**

```
## S3 method for class 'list_correlation_matrices'
plot(x, ...)
```

**Arguments**

x                    A list of matrices obtained from correlation\_analyze function.  
...                   Further arguments are ignored.

**Value**

No return value, called for side effects. The plots show mean and standard deviation of correlation coefficients obtained over multiple iterations.

**Examples**

```
data(elk_network_2010)
elk_correlation_analysis <- correlation_analyze(elk_network_2010)
plot(elk_correlation_analysis)
```

---

`plot.list_node_level_CI`*To plot the results for node-level confidence intervals*

---

**Description**

To plot the results for node-level confidence intervals

**Usage**

```
## S3 method for class 'list_node_level_CI'  
plot(x, ...)
```

**Arguments**

`x` A list of dataframes obtained from `node_level_CI` function.  
`...` Further arguments are ignored.

**Value**

No return value, called for side effects. The plots show confidence intervals along with the observed metric value for each of the nodes in the network.

**Examples**

```
data(elk_network_2010)  
elk_node_level_CI <- node_level_CI(elk_network_2010)  
plot(elk_node_level_CI)
```

---

`plot.list_permuted_networks`*Function to plot the network metrics distribution of permuted networks*

---

**Description**

Function to plot the network metrics distribution of permuted networks

**Usage**

```
## S3 method for class 'list_permuted_networks'
plot(
  x,
  species_original_network,
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
  diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
  igraph::transitivity(x)),
  ...
)
```

**Arguments**

`x` A list of igraph objects obtained obtained using the function `obtain_permuted_network_versions`

`species_original_network` An igraph object which is the original network

`network_metrics_functions_list` A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = `c("edge_density" = function(x) igraph::edge_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))`

... Further arguments are ignored.

**Value**

No return value, called for side effects.

**Examples**

```
data(elk_data_2010, elk_network_2010)
permuted_versions <- obtain_permuted_network_versions(elk_data_2010,
temporal_thresh = 7, spatial_thresh = 15, n_permutations = 10, n_cores = 2)
plot(permuted_versions, elk_network_2010)
```

---

plot.list\_regression\_matrices

*To plot regression analysis results*

---

**Description**

To plot regression analysis results

**Usage**

```
## S3 method for class 'list_regression_matrices'
plot(x, ...)
```

**Arguments**

x                    A list of matrices obtained from regression\_slope\_analyze function

...                  Further arguments are ignored

**Value**

No return value, called for side effects. The plots show regression slope values corresponding to proportion of individuals in the sample.

**Examples**

```
data(elk_network_2010)
elk_regression_analysis <- regression_slope_analyze(elk_network_2010)
plot(elk_regression_analysis)
```

---

```
plot.Subsampled_Network_Metrics
      To plot sub-sampling results
```

---

**Description**

To plot sub-sampling results

**Usage**

```
## S3 method for class 'Subsampled_Network_Metrics'
plot(
  x,
  network,
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
  diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
  igraph::transitivity(x)),
  ...
)
```

**Arguments**

x	A list of matrices belonging to class "Subsampled_Network_Metrics" and is obtained from subsampled_network_metrics function
network	An igraph graph object consisting of the observed network
network_metrics_functions_list	This is the same argument that is passed for obtaining the results from the function subsampled_network_metrics. A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = c("edge_density" = function(x) igraph::edge_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))
...	Further arguments are ignored

**Value**

No return value, called for side effects. The boxplots depict range of values, network metrics take when multiple subsamples are chosen from the observed sample.

**Examples**

```
data(elk_network_2010)
elk_subsamples <- subsampled_network_metrics(elk_network_2010)
plot(elk_subsamples, elk_network_2010,
network_metrics_functions_list = c("edge_density" = function(x) igraph::edge_density(x),
"diameter" = function(x) igraph::diameter(x, weights = NA),
"transitivity" = function(x) igraph::transitivity(x)))
```

---

```
plot.Subsampled_Permutated_Network_Metrics
```

*To plot sub-sampling results of the original network and permuted networks*

---

**Description**

To plot sub-sampling results of the original network and permuted networks

**Usage**

```
## S3 method for class 'Subsampled_Permutated_Network_Metrics'
plot(
  x,
  network,
  n_simulations = 100,
  subsampling_proportion = c(0.1, 0.3, 0.5, 0.7, 0.9),
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
```

```
diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
  igraph::transitivity(x)),
  ...
)
```

### Arguments

**x** A list of matrices obtained from `subsampled_permuted_network_metrics` function of class "Subsampled\_Permuted\_Network\_Metrics"

**network** An igraph graph object consisting of observed network

**n\_simulations** For subsampling results of original network, this determines the number of subsamples to be obtained at each level

**subsampling\_proportion** A vector depicting the levels (in proportion) at which subsamples to be taken. This parameter should be the same as the `subsampling_proportion` parameter passed for the function `subsampled_permuted_network_metrics` to obtain `x`.

**network\_metrics\_functions\_list** This is the same argument that is passed for obtaining the results from the function `subsampled_permuted_network_metrics`. A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = `c("edge_density" = function(x) igraph::edge_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))`

... Further arguments are ignored

### Value

No return value, called for side effects. The boxplots show side-by-side comparison of network metrics distribution from subsamples of observed network and subsamples from permuted networks.

### Examples

```
data(elk_2010_permutations, elk_network_2010)
elk_subsamples_permuted_networks <- subsampled_permuted_network_metrics(elk_2010_permutations)
plot(elk_subsamples_permuted_networks, elk_network_2010)
```

---

`plot.Width_CI_matrix` *To plot the results obtained from width\_CI function*

---

### Description

To plot the results obtained from `width_CI` function



**Usage**

```
## S3 method for class 'Width_CI_matrix'
plot(x, ...)
```

**Arguments**

x                    A matrix of width of Confidence Intervals obtained from global\_width\_CI function

...                   Further arguments are ignored.

**Value**

No return value, called for side effects. Plots show width of confidence intervals corresponding to number of individuals in the sub-sample.

**Examples**

```
data(elk_network_2010)
width_CI_elk <- global_width_CI(elk_network_2010, n_versions = 100)
plot(width_CI_elk)
```

---

plot_network	<i>Visualize Animal Network</i>
--------------	---------------------------------

---

**Description**

Visualize Animal Network

**Usage**

```
plot_network(species_network, seed = 1)
```

**Arguments**

species\_network                    An igraph graph object consisting of observed network.

seed                                Seed to be set for layout.

**Value**

No return value, called for side effects. The plots depict a visualisation of network structure.

**Examples**

```
data(elk_network_2010)
plot_network(elk_network_2010)
```

---

```
regression_slope_analyze
```

*To perform regression analysis for local network metrics*

---

## Description

To perform regression analysis for local network metrics

## Usage

```
regression_slope_analyze(
  network,
  n_simulations = 10,
  subsampling_proportion = c(0.1, 0.3, 0.5, 0.7, 0.9),
  network_metrics_functions_list = c(degree = function(net, sub_net) igraph::degree(net,
    v = igraph::V(sub_net)$name), strength = function(net, sub_net) igraph::strength(net,
    v = igraph::V(sub_net)$name), betweenness = function(net, sub_net)
    igraph::betweenness(net, v = igraph::V(sub_net)$name), clustering_coefficient =
    function(net, sub_net) igraph::transitivity(net, type = "local", vids =
    igraph::V(sub_net)$name), eigenvector_centrality = function(net, sub_net)
    igraph::eigen_centrality(net)$vector[igraph::V(sub_net)$name])
)
```

## Arguments

<code>network</code>	An igraph graph object consisting of observed network
<code>n_simulations</code>	Number of sub-samples to be obtained at each level
<code>subsampling_proportion</code>	A vector depicting proportions of sub-sampled nodes
<code>network_metrics_functions_list</code>	A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Each function definition should include two parameters, one for the main network and another one for the subnetwork. See default example. Default = <code>c("degree" = function(net, sub_net) igraph::degree(net, v = igraph::V(sub_net)\$name), "strength" = function(net, sub_net) igraph::strength(net, v = igraph::V(sub_net)\$name), "betweenness" = function(net, sub_net) igraph::betweenness(net, v = igraph::V(sub_net)\$name), "clustering_coefficient" = function(net, sub_net) igraph::transitivity(net, type = "local", vids = igraph::V(sub_net)\$name), "eigenvector_centrality" = function(net, sub_net) igraph::eigen_centrality(net)\$vector[igraph::V(sub_net)\$name])</code>

## Value

A list of network metrics of class `list_regression_matrices`. Each element of list is a matrix whose columns correspond to `subsampling_proportion` and rows correspond to `n_simulations`. The entries of the matrix provide value of the slope of regression when the nodal values in sub-sampled network are regressed upon the values of the same nodes in the full network for the corresponding metric.

**Examples**

```
data(elk_network_2010)
elk_regression_analysis <- regression_slope_analyze(elk_network_2010)
plot(elk_regression_analysis)
```

---

subsampling\_network\_metrics

*To generate subsamples and obtain network metrics of the subsamples*

---

**Description**

To generate subsamples and obtain network metrics of the subsamples

**Usage**

```
subsampling_network_metrics(
  network,
  n_simulations = 100,
  subsampling_proportion = c(0.1, 0.3, 0.5, 0.7, 0.9),
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
  diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
  igraph::transitivity(x))
)
```

**Arguments**

**network** An igraph graph object consisting of observed network

**n\_simulations** Number of sub-samples to be obtained at each level

**subsampling\_proportion** A vector depicting the levels (in proportion) at which subsamples to be taken

**network\_metrics\_functions\_list** A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = c("edge\_density" = function(x) igraph::edge\_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))

**Value**

A list of network metrics of class "Subsampling\_Network\_Metrics". Each element of list is a matrix whose columns correspond to subsampling\_proportion and rows correspond to n\_simulations. The entries of the matrix provide values of the corresponding metric.

**Examples**

```
data(elk_network_2010)
elk_subsamples <- subsampled_network_metrics(elk_network_2010)
plot(elk_subsamples, elk_network_2010,
network_metrics_functions_list = c("edge_density" = function(x) igraph::edge_density(x),
"diameter" = function(x) igraph::diameter(x, weights = NA),
"transitivity" = function(x) igraph::transitivity(x)))
```

---

```
subsampled_permuted_network_metrics
```

*To generate subsamples of the permuted networks and obtain network metrics of those subsamples*

---

**Description**

To generate subsamples of the permuted networks and obtain network metrics of those subsamples

**Usage**

```
subsampled_permuted_network_metrics(
  networks_list,
  subsampling_proportion = c(0.1, 0.3, 0.5, 0.7, 0.9),
  network_metrics_functions_list = c(edge_density = function(x) igraph::edge_density(x),
  diameter = function(x) igraph::diameter(x, weights = NA), transitivity = function(x)
  igraph::transitivity(x))
)
```

**Arguments**

**networks\_list** A list of igraph objects obtained by permuting the observed network

**subsampling\_proportion**  
A vector depicting the levels (in proportion) at which subsamples to be taken

**network\_metrics\_functions\_list**  
A list consisting of function definitions of the network metrics that the user wants to evaluate. Each element in the list should have an assigned name. Default = c("edge\_density" = function(x) igraph::edge\_density(x), "diameter" = function(x) igraph::diameter(x, weights = NA), "transitivity" = function(x) igraph::transitivity(x))

**Value**

A list of network metrics of class "Subsampled\_Permuted\_Network\_Metrics". Each element of list is a matrix whose columns correspond to subsampling\_proportion and rows correspond to the number of networks in networks\_list. The entries of the matrix provide values of the corresponding metric.

**Examples**

```
data(elk_2010_permutations)
elk_subsamples_permuted_networks <- subsampled_permuted_network_metrics(elk_2010_permutations)
plot(elk_subsamples_permuted_networks, elk_network_2010)
```

# Index

## \* datasets

- elk\_2010\_permutations, [6](#)
  - elk\_all\_interactions\_2010, [6](#)
  - elk\_data\_2010, [7](#)
  - elk\_interactions\_2010, [7](#)
  - elk\_network\_2010, [8](#)
- aniSNA, [2](#)
- bootstrapped\_difference\_pvalues, [3](#)
- correlation\_analyze, [4](#)
- distance\_radian\_coordinates, [5](#)
- elk\_2010\_permutations, [6](#)
- elk\_all\_interactions\_2010, [6](#)
- elk\_data\_2010, [7](#)
- elk\_interactions\_2010, [7](#)
- elk\_network\_2010, [8](#)
- get\_coordinates\_in\_radian, [8](#)
- get\_interactions, [9](#)
- get\_network\_summary, [9](#)
- get\_spatial\_threshold, [10](#)
- global\_CI, [11](#)
- global\_width\_CI, [12](#)
- interacting\_pairs, [13](#)
- network\_from\_interactions, [14](#)
- node\_level\_CI, [14](#)
- obtain\_bootstrapped\_samples, [16](#)
- obtain\_network\_subsamples, [17](#)
- obtain\_permuted\_network\_versions, [17](#)
- plot.bootstrapped\_pvalue\_matrix, [18](#)
- plot.list\_correlation\_matrices, [19](#)
- plot.list\_node\_level\_CI, [20](#)
- plot.list\_permuted\_networks, [20](#)
- plot.list\_regression\_matrices, [21](#)
- plot.Subsampled\_Network\_Metrics, [22](#)
- plot.Subsampled\_Permuted\_Network\_Metrics, [23](#)
- plot.Width\_CI\_matrix, [24](#)
- plot\_network, [25](#)
- regression\_slope\_analyze, [26](#)
- subsampled\_network\_metrics, [27](#)
- subsampled\_permuted\_network\_metrics, [28](#)