Package ‘L1centrality’

April 25, 2024

Title Graph/Network Analysis Based on L1 Centrality

Version 0.1.1

Description Analyze graph/network data using L1 centrality and prestige. Functions for deriving global and local L1 centrality/prestige and L1 centrality/prestige-based neighborhoods of vertices are provided. Routines for visual inspection of a graph/network are also provided. Details are in Kang and Oh (2024) <doi:10.48550/arXiv.2404.13233>.

URL https://github.com/seungwoo-stat/L1centrality

BugReports https://github.com/seungwoo-stat/L1centrality/issues

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 2.10)

LazyData true

Imports graphics, igraph, Rcpp, stats, utils, withr

LinkingTo Rcpp

NeedsCompilation yes

Author Seungwoo Kang [aut, cre] (<https://orcid.org/0000-0001-8082-0794>), Hee-Seok Oh [aut]

Maintainer Seungwoo Kang <kangsw0401@snu.ac.kr>

Repository CRAN

Date/Publication 2024-04-25 14:50:05 UTC

R topics documented:

L1cent 2
L1centEDGE 4
L1centLOC 5
L1centMDS 7
L1centNB 9
Lorenz_plot 11
MCUmovie 12
rokassembly21 13

Index 15

L1cent *L1 Centrality/Prestige*

Description

Computes L_1 centrality or L_1 prestige for each vertex. The L_1 centrality/prestige is a graph centrality/prestige measure defined for the vertices of a graph. It is (roughly) defined by (1 − minimum multiplicity required for a selected vertex to become the median of the graph). For directed graphs, L_1 centrality quantifies the prominence of a vertex in *making* a choice and L_1 prestige quantifies the prominence of a vertex in *receiving* a choice. For undirected graphs, the two measures are identical.

Usage

```
L1cent(g, eta, mode)

## S3 method for class 'igraph'
L1cent(g, eta = NULL, mode = c("centrality", "prestige"))

## S3 method for class 'matrix'
L1cent(g, eta = NULL, mode = c("centrality", "prestige"))
```

Arguments

`g`	An `igraph` graph object or a distance matrix. The graph must be connected. For a directed graph, it must be strongly connected. Equivalently, all entries of the distance matrix must be finite. Here, the (i, j) component of the distance matrix is the geodesic distance from the ith vertex to the jth vertex.
`eta`	An optional nonnegative multiplicity (weight) vector for (vertex) weighted networks. The sum of its components must be positive. If set to `NULL` (the default), all vertices will have the same positive weight (multiplicity), i.e., `g` is treated as a vertex unweighted graph. The length of the `eta` must be equivalent to the number of vertices.
`mode`	A character string. For an undirected graph, either choice gives the same result. • `centrality` (the default): L_1 centrality (prominence of each vertex in terms of *making* a choice) is used for analysis. • `prestige`: L_1 prestige (prominence of each vertex in terms of *receiving* a choice) is used for analysis

Details

Suppose that `g` is a (strongly) connected graph consisting of n vertices $v_1, \ldots, v_n$ whose multiplicities (weights) are $\eta_1, \ldots, \eta_n \geq 0$, respectively, and $\eta_\cdot = \sum_{k=1}^{n} \eta_k > 0$.

The centrality median vertex of this graph is the node minimizing the weighted sum of distances. That is, v_i is the centrality median vertex if

$$\sum_{k=1}^{n} \eta_k d(v_i, v_k)$$

is minimized, where $d(v_i, v_k)$ denotes the geodesic (shortest path) distance from v_i to v_k. See `igraph::distances()` for algorithms for computing geodesic distances between vertices. When the indices are swapped to $d(v_k, v_i)$ in the display above, we call the node minimizing the weighted sum as the prestige median vertex. When the graph is undirected, the prestige median vertex and the centrality median vertex coincide, and we call it the graph median, following Hakimi (1964).

The L_1 centrality for an arbitrary node v_i is defined as 'one minus the minimum weight that is required to make it a centrality median.' This concept of centrality is closely related to the data depth for ranking multivariate data, as defined in Vardi and Zhang (2000). It turns out that the following formula computes the L_1 centrality for the vertex v_i:

$$1 - \mathcal{S}(\mathrm{g}) \max_{j \neq i} \left\{ \frac{\sum_{k=1}^{n} \eta_k (d(v_i, v_k) - d(v_j, v_k))}{\eta_\cdot d(v_j, v_i)} \right\}^+,$$

where $\{\cdot\}^+ = \max(\cdot, 0)$ and $\mathcal{S}(\mathrm{g}) = \min_{i \neq j} d(v_i, v_j)/d(v_j, v_i)$. The L_1 centrality of a vertex is in $[0, 1]$ by the triangle inequality, and the centrality median vertex has centrality 1. The L_1 prestige is defined analogously, with the indices inside the distance function swapped.

For an undirected graph, $\mathcal{S}(\mathrm{g}) = 1$ since the distance function is symmetric. Moreover, L_1 centrality and L_1 prestige measures concide.

For details, refer to Kang and Oh (2024a) for undirected graphs, and Kang and Oh (2024b) for directed graphs.

Value

A numeric vector whose length is equivalent to the number of vertices in the graph g. Each component of the vector is the L_1 centrality (if `mode = "centrality"`) or the L_1 prestige (if `mode = "prestige"`) of each vertex in the given graph.

Note

The function is valid only for connected graphs. If the graph is directed, it must be strongly connected.

References

S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024a.

S. Kang and H.-S. Oh. L_1 prominence measures for directed graphs. Manuscript. 2024b.

Y. Vardi and C.-H. Zhang. The multivariate L_1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4):1423–1426, 2000.

See Also

L1centLOC(), L1centNB(), L1centMDS(), L1centEDGE(), Lorenz_plot() for L_1 centrality- or prestige-based analysis. See L1centrality-package for each function's support range.

igraph::betweenness(), igraph::closeness(), igraph::degree(), igraph::eigen_centrality() for centrality measures.

Examples

```
# igraph object and distance matrix as an input lead to the same result
vertex_weight <- igraph::V(MCUmovie)$worldwidegross
cent_igraph <- L1cent(MCUmovie, eta=vertex_weight)
cent_matrix <- L1cent(igraph::distances(MCUmovie), eta=vertex_weight)
all(cent_igraph == cent_matrix)

# Top 6 vertices with the highest L1 centrality
utils::head(sort(cent_igraph, decreasing = TRUE))
```

L1centEDGE *Multiscale Edge Representation*

Description

Derives a multiscale edge representation. Each vertex is connected to its local median, which is found in its L_1 centrality-based neighborhood.

Usage

```
L1centEDGE(g, eta, alpha)

## S3 method for class 'igraph'
L1centEDGE(g, eta = NULL, alpha)

## S3 method for class 'matrix'
L1centEDGE(g, eta = NULL, alpha)
```

Arguments

g	An igraph graph object or a distance matrix. The graph must be undirected and connected. Equivalently, the distance matrix must be symmetric, and all entries must be finite.
eta	An optional nonnegative multiplicity (weight) vector for (vertex) weighted networks. The sum of its components must be positive. If set to NULL (the default), all vertices will have the same positive weight (multiplicity), i.e., g is treated as a vertex unweighted graph. The length of the eta must be equivalent to the number of vertices.
alpha	A number or a numeric vector of locality levels. Values must be between 0 and 1.

Details

In a global perspective, any given graph can be represented as a star-shaped (directed) graph, with each vertex making a connection to the median vertex. Based on this idea, a graph can be represented as a directed graph, with each vertex making a connection to the *local* median vertex. The local median vertex of, say, v_i, is defined as a median vertex among the L_1 centrality-based neighborhood of v_i. By varying the level of locality, the given graph can be visually inspected at multiple scales.

Value

A list of 'edge lists'. The length of the list is equivalent to the length of `alpha`, and the names of the list are the values of `alpha`. The `i`th component of the list is a 2-column matrix, and each row defines one directed edge, i.e., it is an edge list. The second column is the local (level `alpha[i]`) median of the vertex at the first column. There may be more than one edge from each vertex, since there may be more than one local median.

Note

The function is valid only for undirected and connected graphs.

References

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024.

See Also

`L1cent()`, `L1centNB()`, `L1centLOC()`. Using the output, one can use `igraph::graph_from_edgelist()` for creating an `igraph` object. See the example code below.

Examples

```
library(igraph)
MCU_edge <- L1centEDGE(MCUmovie, eta = V(MCUmovie)$worldwidegross, alpha = 5/32)
graph <- graph_from_edgelist(MCU_edge[[1]], directed = TRUE)
plot(graph)
```

`L1centLOC` *Local L1 Centrality/Prestige*

Description

Computes local L_1 centrality or prestige at each `alpha` level for every vertex. For undirected graphs, the two measures are identical.

Usage

```
L1centLOC(g, eta, alpha, mode)

## S3 method for class 'igraph'
L1centLOC(g, eta = NULL, alpha, mode = c("centrality", "prestige"))

## S3 method for class 'matrix'
L1centLOC(g, eta = NULL, alpha, mode = c("centrality", "prestige"))
```

Arguments

g	An igraph graph object or a distance matrix. The graph must be connected. For a directed graph, it must be strongly connected. Equivalently, all entries of the distance matrix must be finite. Here, the (i, j) component of the distance matrix is the geodesic distance from the ith vertex to the jth vertex.
eta	An optional nonnegative multiplicity (weight) vector for (vertex) weighted networks. The sum of its components must be positive. If set to NULL (the default), all vertices will have the same positive weight (multiplicity), i.e., g is treated as a vertex unweighted graph. The length of the eta must be equivalent to the number of vertices.
alpha	A number or a numeric vector of locality levels. Values must be between 0 and 1.
mode	A character string. For an undirected graph, either choice gives the same result. • centrality (the default): L_1 centrality (prominence of each vertex in terms of *making* a choice) is used for analysis. • prestige: L_1 prestige (prominence of each vertex in terms of *receiving* a choice) is used for analysis

Details

Suppose that the given graph has n vertices. We choose about $n\alpha$ vertices (L_1 centrality- or prestige-based neighborhood) for each vertex (see L1centNB()), and compute the L_1 centrality or prestige of the vertex conditioned on these vertices, i.e., derive the L_1 centrality or prestige locally. For details, refer to Kang and Oh (2024a) for undirected graphs, and Kang and Oh (2024b) for directed graphs.

Value

A list of numeric vectors. The length of the list is equivalent to the length of alpha, and the names of the list are the values of alpha. Each component of the list is a numeric vector whose length is equivalent to the number of vertices in the graph g. Specifically, the ith component of the list is a vector of local L_1 centrality at level alpha[i] for each vertex (if mode = "centrality") or local L_1 prestige at level alpha[i] for each vertex (if mode = "prestige").

Note

The function is valid only for connected graphs. If the graph is directed, it must be strongly connected.

References

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024a.

S. Kang and H.-S. Oh. L_1 prominence measures for directed graphs. Manuscript. 2024b.

See Also

`L1cent()` for L_1 centrality/prestige, `L1centNB()` for L_1 centrality/prestige-based neighborhood.

Examples

```
weight <- igraph::V(MCUmovie)$worldwidegross
MCUmovie_cent <- L1cent(MCUmovie, eta = weight)
MCUmovie_loc_cent <- L1centLOC(MCUmovie, eta = weight, alpha = 5/32)
plot(MCUmovie_cent, MCUmovie_loc_cent[[1]],
     xlab="Global L1 centrality", ylab="Local L1 centrality (alpha = 5/32)",
     main="MCU movie network: global vs. local centrality")
graphics::text(MCUmovie_cent, MCUmovie_loc_cent[[1]], igraph::V(MCUmovie)$name)
```

`L1centMDS` *Fitting a Target Plot*

Description

`L1centMDS()` and `plot.L1centMDS()` are used together to draw a target plot, which is a target-shaped 2D plot that aids in the visual inspection of a network using the L_1 centrality. See Kang and Oh (2024) for a formal definition of a target plot.

Usage

```
L1centMDS(g, tol, maxiter, verbose)

## S3 method for class 'igraph'
L1centMDS(g, tol = 1e-05, maxiter = 1000, verbose = TRUE)

## S3 method for class 'matrix'
L1centMDS(g, tol = 1e-05, maxiter = 1000, verbose = TRUE)

## S3 method for class 'L1centMDS'
plot(x, zoom = 1, main = NULL, ...)
```

Arguments

`g`	An `igraph` graph object or a distance matrix. The graph must be undirected and connected. Equivalently, the distance matrix must be symmetric, and all entries must be finite.

`tol`	A numerical tolerance. The gradient descent method terminates if the relative magnitude of the gradient falls below `tol` as in Kruskal (1964b). By default set to 10^{-5}.
`maxiter`	A number of maximum iteration allowances for the gradient descent algorithm. By default set to 1000.
`verbose`	A boolean. • `TRUE` (the default): for each iteration, prints (1) current number of iterations, (2) current stress, and (3) relative magnitude of the gradient to the console. At the end, the final message is printed to the console; total number of iterations and final stress. • `FALSE`: suppress message to the console.
`x`	An `L1centMDS` object, obtained as a result of the function `L1centMDS()`.
`zoom`	A numerical value on how much to zoom-in the plot. By default set to 1 (no zoom).
`main`	Title of the plot. If set to `NULL` (the default), the title prints "Target plot / Stress = X".
`...`	Further graphical parameters supplied to the internal `base::plot()` (for points) and `graphics::text()` (for labels) function. See `graphics::par()` document. To supply an argument to the former one, use the prefix '`plot.`' and for the latter, '`text.`'. For instance, `plot.cex = 1` sets the size of the point, whereas `text.cex = 1` sets the size of the label.

Details

Denoting the L_1 centrality of vertex i as $c_i \in (0, 1]$, a point representing that vertex is placed on a concentric circle with radius $r_i = -\log(c_i)$. Representing each vertex as (r_i, θ_i) (in circular coordinates), the values of θ_i are derived using nonmetric multidimensional scaling proposed in Kruskal (1964a,b). The initial configuration is derived using classical multidimensional scaling (`stats::cmdscale()`). A gradient descent algorithm is employed in deriving optimal θ_is.

Value

`L1centMDS()` returns an object of class `L1centMDS`. It is a list consisting of four vectors:

- 'label': If `g` is an `igraph` object, `name` attribute of the vertex. If `g` is a distance matrix, row names (`rownames`). Is set to `NULL` if missing.
- 'radius': Radius of a point representing each vertex in the target plot's circular coordinate system, i.e., $-\log(L_1 \text{ centrality})$ for each vertex.
- 'theta': Angle (in radians) of a point representing each vertex in the target plot's circular coordinate system.
- 'stress': Stress measure defined in Kruskal (1964a).

`plot.L1centMDS()` draws a target plot. Four concentric circles denote the 1st to 4th quartiles of the radius, and the values of the L_1 centrality quartiles are shown in red text. Note that red texts denote the L_1 centrality quartiles, *not* radius quartiles.

Note

The function `L1centMDS()` is valid only for undirected and connected graphs. Also, `L1centMDS()` only considers graphs with equal vertex multiplicities.

References

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964a.

J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2): 115–129, 1964b.

See Also

`L1cent()` for L_1 centrality/prestige, `MASS::isoMDS()` and `stats::cmdscale()` for multidimensional scaling methods.

Examples

```
parameters <- L1centMDS(MCUmovie, verbose = FALSE)
plot(parameters)
```

`L1centNB` *L1 Centrality/Prestige-Based Neighborhood*

Description

Derives L_1 centrality- or prestige-based neighborhood of each vertex. For undirected graphs, the two neighborhood are identical.

Usage

```
L1centNB(g, eta, mode)

## S3 method for class 'igraph'
L1centNB(g, eta = NULL, mode = c("centrality", "prestige"))

## S3 method for class 'matrix'
L1centNB(g, eta = NULL, mode = c("centrality", "prestige"))
```

Arguments

- `g` An `igraph` graph object or a distance matrix. The graph must be connected. For a directed graph, it must be strongly connected. Equivalently, all entries of the distance matrix must be finite. Here, the (i, j) component of the distance matrix is the geodesic distance from the ith vertex to the jth vertex.
- `eta` An optional nonnegative multiplicity (weight) vector for (vertex) weighted networks. The sum of its components must be positive. If set to `NULL` (the default), all vertices will have the same positive weight (multiplicity), i.e., `g` is treated as a vertex unweighted graph. The length of the `eta` must be equivalent to the number of vertices.
- `mode` A character string. For an undirected graph, either choice gives the same result.
  - `centrality` (the default): L_1 centrality (prominence of each vertex in terms of *making* a choice) is used for analysis.
  - `prestige`: L_1 prestige (prominence of each vertex in terms of *receiving* a choice) is used for analysis

Details

For an undirected graph, if the graph is symmetrized (in a way defined in Kang and Oh (2024a)) w.r.t. a vertex v, vertex v becomes the graph median (Kang and Oh, 2024a), i.e., v has L_1 centrality 1. Based on this property, we define the L_1 centrality-based neighborhood of vertex v as vertices that have large L_1 centrality on the symmetrized graph w.r.t. vertex v.

For a directed graph, a vertex of interest, say v, is made to a centrality and prestige median vertex by the procedure described in Kang and Oh (2024b). We call the resulting graph as the modified graph w.r.t. v. L_1 centrality(prestige)-based neighborhood of vertex v is a set of vertices that have large L_1 centrality(prestige) on the modified graph w.r.t. vertex v.

Value

A list of numeric vectors. The length of the list is equivalent to the number of vertices in the graph `g`, and the names of the list are vertex names. Each component of the list is a numeric vector whose length is equivalent to the number of vertices in the graph `g`. Specifically, the `i`th component of the list is a vector of the L_1 centrality of each vertex, for the modified graph `g` w.r.t. the `i`th vertex (if `mode = "centrality"`) or the L_1 prestige of each vertex, for the modified graph `g` w.r.t. the `i`th vertex (if `mode = "prestige"`).

Note

The function is valid only for connected graphs. If the graph is directed, it must be strongly connected.

References

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024a.

S. Kang and H.-S. Oh. L_1 prominence measures for directed graphs. Manuscript. 2024b.

See Also

L1cent() for L_1 centrality/prestige, L1centLOC() and L1centEDGE() internally uses L1centNB().

Examples

```
NB <- L1centNB(MCUmovie, eta = igraph::V(MCUmovie)$worldwidegross)
# Top 6 L1 centrality-based neighbors of "Iron Man"
utils::head(sort(NB$"Iron Man", decreasing = TRUE))
```

Lorenz_plot *Lorenz Curve and the Gini Coefficient*

Description

Draws a Lorenz curve (the group heterogeneity plot) and computes the Gini coefficient (the group heterogeneity index).

Usage

```
Lorenz_plot(x, add = FALSE, ...)
```

Arguments

x	A numeric vector.
add	A logical value. • TRUE: add the Lorenz curve to an already existing plot. • FALSE (the default): draw the Lorenz curve to a new graphic device.
...	Further graphical parameters supplied to the internal base::plot() (when add = FALSE) or graphics::lines() (when add = TRUE) function. See graphics::par() document.

Value

Draws a Lorenz curve (the group heterogeneity plot) and returns an invisible copy of a Gini coefficient (the group heterogeneity index).

References

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024.

M. O. Lorenz. Methods of measuring the concentration of wealth. *Publications of the American Statistical Association*, 9(70):209–219, 1905.

See Also

Use the function with L1cent() or L1centLOC(), and compare distributions of the centrality measurements across several groups and graphs.

Examples

```
vertex_weight <- igraph::V(MCUmovie)$worldwidegross
cent <- L1cent(MCUmovie, eta=vertex_weight)
gini <- Lorenz_plot(cent, asp=1)
graphics::abline(0,1,lty=2)
# group heterogeneity index
gini
```

MCUmovie — *Marvel Cinematic Universe Movie Network*

Description

Network between 32 movies from the Marvel Cinematic Universe (MCU) that were released between 2008 and 2023. Each movie represents one vertex.

An edge between movies i and j is formed if there is at least one cast in common. Denoting the set of casts of movie i as A_i, the weight of this edge is given as $(|A_i \cap A_j|/|A_i \cup A_j|)^{-1}$, where $|\cdot|$ denotes the cardinality of a set.

Usage

```
data(MCUmovie)
```

Format

An undirected, connected, and (edge) weighted `igraph` graph object with 32 vertices and 278 edges.

Vertex attributes:

- 'name': name of the movie. e.g., *Guardians of the Galaxy Vol. 3*.
- 'worldwidegross': worldwide gross in USD. Archived from IMDb on Nov. 3rd, 2023.
- 'year': release year of the movie.

Edge attribute: 'weight'. Given as a dissimilarity between two vertices. See the description above.

Source

IMDb: https://www.imdb.com

References

G. Choi and H.-S. Oh. Heavy-snow transform: A new method for graph signals. Manuscript, 2021.

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024.

`rokassembly21` *Republic of Korea's 21st National Assembly Bill Cosponsorship Network*

Description

Network between 317 members of the Republic of Korea's 21st National Assembly (May 30th, 2020–May 29th, 2024). Each member of the assembly represents one vertex.

An edge between two members is formed if there is at least one cosponsored bill. The weight of this edge is given as 1/(number of cosponsored bills between two members) during the first 40 months of the 21st assembly (Jun. 2020–Sep. 2023).

Usage

```
data(rokassembly21)
```

Format

An undirected, connected, and (edge) weighted `igraph` graph object with 317 vertices and 47,657 edges.

Vertex attributes:

- 'name': Pseudonyms of each member. They are in the format of the party's initial character, followed by a random number (e.g., D4). Each party's initial character is:
  - 'D': Democratic Party of Korea.
  - 'P': People Power Party.
  - 'J': Justice Party.
  - 'O': Others (Basic Income Party, Hope of Korea, The Progressive Party, Transition Korea).
- 'party': Factor with 7 levels. Denotes the political party of each member as of Sep. 2023. Note that independent members are assigned to their original party.
- 'gender': Factor with 2 levels. 'M' (male) or 'F' (female).
- 'nelect': Number of legislative terms in the assembly for each member. Ranges from 1 to 6.
- 'district': Indicates if each member is a district representative (`TRUE`) or a proportional representative (`FALSE`).
- 'full': Indicates if each member was in the assembly for the first 40 months. `TRUE` for the members in the office for all 40 months. Members who started their term via by-election, resigned, or lost their seat for any reason during the 40 months are coded as `FALSE`.
- 'nbill': Number of bills cosponsored by each member.

Edge attribute: 'weight'. Given as a dissimilarity between two vertices. See the description above.

Source

The National Assembly of the Republic of Korea

- Bill information: https://likms.assembly.go.kr/bill/main.do
- Member information: https://open.assembly.go.kr/portal/assm/search/memberSchPage.do

References

S. Kang and H.-S. Oh. On a notion of graph centrality based on L_1 data depth. *arXiv preprint arXiv:2404.13233*, 2024.

Index

∗ **datasets**
MCUmovie, 12
rokassembly21, 13

base::plot(), *8*, *11*

graphics::lines(), *11*
graphics::par(), *8*, *11*
graphics::text(), *8*

igraph::betweenness(), *4*
igraph::closeness(), *4*
igraph::degree(), *4*
igraph::distances(), *3*
igraph::eigen_centrality(), *4*
igraph::graph_from_edgelist(), *5*

L1cent, 2
L1cent(), *5*, *7*, *9*, *11*
L1centEDGE, 4
L1centEDGE(), *4*, *11*
L1centLOC, 5
L1centLOC(), *4*, *5*, *11*
L1centMDS, 7
L1centMDS(), *4*
L1centNB, 9
L1centNB(), *4–7*
L1centrality-package, *4*
L1pres *(*L1cent*)*, 2
L1presLOC *(*L1centLOC*)*, 5
L1presNB *(*L1centNB*)*, 9
L1prestige *(*L1cent*)*, 2
Lorenz_plot, 11
Lorenz_plot(), *4*

MASS::isoMDS(), *9*
MCUmovie, 12

plot.L1centMDS *(*L1centMDS*)*, 7

rokassembly21, 13

stats::cmdscale(), *8*, *9*