# Package 'aspace'

November 24, 2009

**Type** Package

**Title** A collection of functions for estimating centrographic statistcs and computational geometries for spatial point patterns

**Version** 2.2

**Date** 21-11-2009

**Author** Randy Bui, Ron N. Buliung, Tarmo K. Remmel

**Maintainer** Ron N. Buliung <ron.buliung@utoronto.ca>

**Description** A collection of functions for computing centrographic statistics (e.g., standard distance, standard deviation ellipse, standard deviation box), and minimum convex polygons (MCP) for observations taken at point locations. A tool is also provided for converting geometric objects associated with the centrographic statistics, and MCPs into ESRI Shapefiles. Separate plotting functions have been developed for each measure.

**License** GPL (>= 2)

**Depends** R (>= 2.0.1), adehabitat, ade4, gpclib, sp, maptools, splancs, Hmisc

**LazyData** yes

**Repository** CRAN

**Date/Publication** 2009-11-24 10:11:38

## R topics documented:

---

`aspace-package` *A collection of functions for estimating centrographic statistics and computational geometries from spatial point patterns*

---

## Description

A collection of functions for computing centrographic statistics (e.g., standard distance, standard deviation ellipse), and minimum convex polygons (MCP) from point data. A tool is also provided for converting geometric objects associated with the centrographic statistics, and MCPs into ESRI Shapefiles. This library was initially conceived to aid in the analysis of spatial patterns of travel behaviour (see Buliung and Remmel, 2008). Major changes in the current version include the separation of plotting and estimation capabilities, and the addition of other centrographic measures (e.g., standard deviation box, centre of minimum distance, central feature).

## Details

| | |
|---|---|
| Package: | aspace |
| Type: | Package |
| Version: | 2.2 |
| Date: | 21-11-2009 |
| License: | GPL (>= 2.0) |

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## References

Bachi, R. 1963. Standard distance measures and related methods for spatial analysis. Papers of the Regional Science Association 10: 83-132.

Buliung, R.N. and Remmel, T. (2008) Open source, spatial analysis, and activity travel behaviour research: capabilities of the aspace package. Journal of Geographical Systems, 10: 191-216.

Buliung, R.N. and Kanaroglou, P.S. (2006) Urban form and household activity-travel behaviour. Growth and Change, 37: 174-201.

Ebdon, D. 1988. Statistics in Geography 2nd Edition. Oxford UK: Blackwell.

Levine, N. 2002. CrimeStat II: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 2.0) Houston TX/National Institute of Justice, Washington DC: Ned Levine & Associates.

Mohr, C.O. 1947. Table of equivalent populations of North American mammals. American Midland Naturalist, 37: 223-249.

---

`acos_d` *Compute inverse cosine with angle given in degrees*

---

## Description

Provides the functionality of acos, but for input angles measured in degrees (not radians).

## Usage

```
acos_d(theta = 0)
```

## Arguments

| | |
|---|---|
| `theta` | A numeric angular measurement in degrees from north. |

## Details

Since the R default is to compute trigonometric functions on angular measurements stored in radians, this simple function performs the conversion from degrees, reducing the need to do so a priori, outside the function.

## Value

Returns a numeric value for the inverse cosine of the specified angular measurement

### Note

To reduce the need for unit conversions prior to calling trigonometric functions, this function accepts input in angular degrees rather than radians. Depending on the data source, this function may be preferred to the existing version requiring input in angular radians.

### Author(s)

Tarmo K. Remmel

### See Also

sin_d, cos_d, tan_d, asin_d, atan_d

### Examples

```
acos_d(theta = 90)
```

---

## activities *Demo Data: x and y coordinates of 10 specified point locations*

---

### Description

This is a simple two-column data frame (or matrix) containing x,y coordinates for a series of point locations. These data mimic UTM coordinates such that the first column contains Easting (x), and the second Northing (y) coordinates for the set of unique points.

### Usage

```
data(activities)
```

### Format

A data frame with 10 observations on the following 2 variables.

`col1` A numeric vector of x-coordinates

`col2` A numeric vector of y-coordinates

### Details

The coordinates of the points must have the same units and projection as the specified center.

### Source

This demonstration data has been manufactured for illustrative purposes only.

### Examples

```
data(activities)
str(activities)
plot(activities)
```

asin_d *Compute inverse sine with angle given in degrees*

## Description

Provides the functionality of asin, but for input angles measured in degrees (not radians).

## Usage

```
asin_d(theta = 0)
```

## Arguments

| | |
|---|---|
| theta | A numeric angular measurement in degrees from north. |

## Details

Since the R default is to compute trigonometric functions on angular measurements stored in radians, this simple function performs the conversion from degrees, reducing the need to do so a priori, outside the function.

## Value

Returns a numeric value for the inverse sine of the specified angular measurement.

## Note

To reduce the need for unit conversions prior to calling trigonometric functions, this function accepts input in angular degrees rather than radians. Depending on the data source, this function may be preferred to the existing version requiring input in angular radians.

## Author(s)

Tarmo K. Remmel

## See Also

sin_d, cos_d, tan_d, acos_d, atan_d

## Examples

```
asin_d(theta = 90)
```

`as_radians` *Converts degrees to radians*

## Description

This function converts an angular measure stored in degrees to radians. This is an alternative to the rad function available in the package circular.

## Usage

```
as_radians(theta = 0)
```

## Arguments

| | |
|---|---|
| `theta` | A numeric angular measurement in degrees from north. |

## Details

Achieves a very simple conversion with a convenient function call.

## Value

Returns a numeric value for an angle in radians that is equivalent to the input theta in degrees.

## Note

The purpose of this function is to reduce computer code clutter when using angular measuremnts in R. The simple function call ensures that degree to radian conversions are completed consistently and accurately. Since trigonometric functions in R require angular measures in radians rather than degrees, this simple function can be used for simple angular unit conversion.

## Author(s)

Tarmo K. Remmel

## See Also

`sin_d`, `cos_d`, `tan_d`, `asin_d`, `acos_d`, `atan_d`

## Examples

```
as_radians(theta = 90)
```

atan_d *Compute inverse tangent with angle given in degrees*

## Description

Provides the functionality of atan, but for input angles measured in degrees (not radians).

## Usage

```
atan_d(theta = 0)
```

## Arguments

| | |
|---|---|
| theta | A numeric angular measurement in degrees from north. |

## Details

Since the R default is to compute trigonometric functions on angular measurements stored in radians, this simple function performs the conversion from degrees, reducing the need to do so a priori, outside the function.

## Value

Returns a numeric value for the inverse tangent of the specified angular measurement.

## Note

To reduce the need for unit conversions prior to calling trigonometric functions, this function accepts input in angular degrees rather than radians. Depending on data, this function may be preferred to the existing version requiring input in angular radians.

## Author(s)

Tarmo K. Remmel

## See Also

sin_d, cos_d, tan_d, asin_d, acos_d

## Examples

```
atan_d(theta = 90)
```

calc_box *Calculate the Standard Deviation Box*

## Description

The orthogonal dispersion of a set of points can be described using the standard deviation of the x- and y-coordinates of a set of point observations. The orthogonal dispersion can then be visualized with a Standard Deviation Box. This function computes the properties of the Standard Deviation Box (SD Box) from a set of point observations.

## Usage

```
calc_box(id=1, filename="BOX_Output.txt", centre.xy=NULL, calccentre=TRUE, weighted
```

## Arguments

| | |
|---|---|
| id | A unique integer to identify a SD Box |
| filename | A string indicating the ASCII textfile where the box coordinates will be written |
| centre.xy | A vector of length 2, containing the x- and y-coordinates of the geographic centre of the SD Box |
| calccentre | Boolean: Set to TRUE if the mean center is to be calculated |
| weighted | Boolean: Set to TRUE if the weighted mean center is to be computed with weighted coordinates |
| weights | Weights applied to point observations, number of weights should equal the number of observations |
| CMD.npts | The approximate number of points to generate for the centre of minimum distance (CMD) calculation |
| points | A 2-column matrix or data frame containing the set of point observations input to the calc_box function |
| verbose | Boolean: Set to TRUE if extensive feedback is desired on the standard output |

## Details

This function is most powerful when used repetitively within a loop to compute the SD Box for subsets of points stored in a large table.

## Value

The returned result is a list:

| | |
|---|---|
| id | Identifier for the SD Box shape - it should be unique |
| calccentre | Boolean: TRUE if the mean centre was estimated |
| weighted | Boolean: TRUE if the weighted mean centre was estimated |
| CENTRE.x | X-coordinate of the centre |

| | |
|---|---|
| CENTRE.y | Y-coordinate of the centre |
| central.x | X-coordinate of the identified central feature |
| central.y | Y-coordinate of the identified central feature |
| median.x | X-coordinate of median centre, median value of the X-coordinate vector |
| median.y | Y-coordinate of median centre, median value of the Y-coordinate vector |
| CMD.x | X-coordinate of estimated centre of minimum distance |
| CMD.y | Y-coordinate of estimated centre of minimum distance |
| SD.x | Orthogonal standard deviation in the x-axis |
| SD.y | Orthogonal standard deviation in the y-axis |
| Box.area | Area of the standard deviation box |
| NW.coord | North-west coordinates of SD Box |
| NE.coord | North-east coordinates of SD Box |
| SW.coord | South-west coordinates of SD Box |
| SE.coord | South-east coordinates of SD Box |

## Note

Results are stored in the r.BOX object (required for graphical visualization using the plot_box function). This function can be used on its own (once) or repetitively in a loop to process grouped point data stored in a larger table. When used repetitively, be sure to increment the id parameter to ensure that each SD Box has a unique identifier. The output ASCII coordinate file can be further processed using the makeshapes function to generate an ESRI Shapefile for SD Box polygons.

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## See Also

plot_box, calc_mcp, calc_sde, calc_sdd, ellipse3, makeshapes, mcp, gridpts, wtd.var

## Examples

```
calc_box(id=1, filename="BOX_Output.txt", centre.xy=NULL, calccentre=TRUE, weighted=FALSE, w
```

`calc_mcp` *Computing the Minimum Convex Polygon (MCP)*

## Description

This function computes the Minimum Convex Polygon (MCP) from a set of points. The MCP is the minimum area polygon containing a set of point locations.

## Usage

```
calc_mcp(id=1, points = activities, filename="MCP_Output.txt", verbose = FALSE, pct
```

## Arguments

| | |
|---|---|
| `id` | Provide a unique integer to identify an MCP from others that you may construct with other data points |
| `points` | Two-column matrix or data frame of point coordinates |
| `filename` | A character name for an ASCII output file |
| `verbose` | Boolean: set to TRUE if extended processing feedback is wanted |
| `pct` | Integer 0 <= pct <=100, the percentage of the MCP for which area is provided |

## Details

This function is most powerful when used repetitively within a loop to compute the MCP for subsets of points stored in a large data table.

## Value

The returned result is a list:

| | |
|---|---|
| `MCP.area` | The area of the MCP in square kilometers |
| `MCP.pct` | The desired percentage of the MCP for which the area is computed |
| `MCP.coords` | A matrix containing MCP vertices. Each row represents a unique point, the first column contains x-coordinates, and the second, y-coordinates |

## Note

Results are stored in the r.MCP object (required for graphical visualization using plot_mcp). This function can be used on its own (once) or repetitively in a loop to process grouped point data stored in a larger table. When used repetitively, be sure to increment the id parameter to ensure that each MCP has a unique identifier. The output ASCII coordinate file can be further processed using the makeshapes function to generate an ESRI Shapefile for MCP polygons.

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## References

Builds upon MCP functions available in the adehabitat package

## See Also

plot_mcp, mcp, calc_sdd, calc_sde, calc_box, makeshapes

## Examples

```
calc_mcp(id=1, points = activities, filename="MCP_Output.txt", verbose = FALSE, pct = 100)
```

---

# calc_sdd *Calculate the Standard Distance Deviation (Standard Distance)*

---

## Description

This function computes the Standard Distance Deviation (SDD) or Standard Distance from a set of points.

## Usage

```
calc_sdd(id=1, filename="SDD_Output.txt", centre.xy=NULL, calccentre=TRUE, weighted
```

## Arguments

| | |
|---|---|
| id | A unique integer to identify a SDD estimate |
| filename | A string indicating the ASCII textfile where shape coordinates will be written |
| centre.xy | A vector of length 2, containing the x- and y-coordinates of the SDD centre |
| calccentre | Boolean: Set to TRUE if the mean center is to be calculated |
| weighted | Boolean: Set to TRUE if the weighted mean center is to be computed with weighted coordinates |
| weights | Weights applied to point observations, number of weights should equal the number of observations |
| CMD.npts | The approximate number of points to generate for the centre of minimum distance (CMD) calculation |
| points | A 2-column matrix or data frame containing the set of point observations input to the calc_sdd function |
| verbose | Boolean: Set to TRUE if extensive feedback is desired on the standard output |

## Details

This function is most powerful when used repetitively within a loop to compute the SDD for subsets of points stored in a large table.

## Value

The result is a list of terms:

| | |
|---|---|
| `id` | Identifier for the SDD shape - it should be unique |
| `calccentre` | Boolean: TRUE if mean centre is computed |
| `weighted` | Boolean: TRUE if the weighted mean centre is to be used instead |
| `CENTRE.x` | X-coordinate of the centre |
| `CENTRE.y` | Y-coordinate of the centre |
| `central.x` | X-coordinate of the identified central feature |
| `central.y` | Y-coordinate of the identified central feature |
| `median.x` | X-coordinate of median centre, median value of the X-coordinate vector |
| `median.y` | Y-coordinate of median centre, median value of the Y-coordinate vector |
| `CMD.x` | X-coordinate of estimated centre of minimum distance |
| `CMD.y` | Y-coordinate of estimated centre of minimum distance |
| `SDD.radius` | SDD value, radius of the SDD |
| `SDD.area` | Area of the SDD circle |

## Note

Results are stored in the r.SDD object (required for graphical visualization using plot_sdd). As the value of 'CMD.npts' increases, the more precise the centre of minimum distance is estimated. This function can be used on its own (once) or repetitively in a loop to process grouped point data stored in a larger table. When used repetitively, be sure to increment the id parameter to ensure that each SDD has a unique identifier. The output ASCII coordinate file can be further processed using the makeshapes function to generate an ESRI Shapefile for SDD polygons.

## Author(s)

Randy Bui, Ron Buliung, Tarmo K. Remmel

## See Also

`plot_sdd`, `calc_mcp`, `calc_sde`, `calc_box`, `ellipse3`, `makeshapes`, `mcp`, `gridpts`

## Examples

```
calc_sdd(id=1, filename="SDD_Output.txt", centre.xy=NULL, calccentre=TRUE, weighted=TRUE, we
```

calc_sde *Calculating the Standard Deviation Ellipse*

## Description

This function computes the Standard Deviation Ellipse (SDE) from a set of points. The SDE is a centrographic measure used to characterize the dispersion of point observations along two orthogonal axes. The SDE also captures directional bias in a spatial point pattern, the ellipse will be oriented in the direction of maximum dispersion.

## Usage

```
calc_sde(id=1, filename="SDE_Output.txt", centre.xy=NULL, calccentre=TRUE, weighted
```

## Arguments

| | |
|---|---|
| id | A unique integer to identify the shape |
| filename | A string indicating the ASCII textfile where shape coordinates will be written |
| centre.xy | A vector of length 2, containing the x- and y-coordinates of the SDE centre (Planar Coordinates Only!) |
| calccentre | Boolean: Set to TRUE if the mean center is to be calculated |
| weighted | Boolean: Set to TRUE if the weighted mean center is to be computed with weighted coordinates |
| weights | Weights applied to point observations, number of weights should equal the number of observations |
| CMD.npts | The approximate number of points to generate for the centre of minimum distance (CMD) calculation |
| points | A 2-column matrix or data frame containing point coordinates |
| verbose | Boolean: Set to TRUE if extensive feedback is desired on the standard output |

## Details

This function is most powerful when used repetitively within a loop to compute the SDE for subsets of points stored in a large data table.

## Value

The returned result is a list:

| | |
|---|---|
| id | Identifier for the SDE shape - it should be unique |
| calccentre | Boolean: TRUE if mean centre is computed |
| weighted | Boolean: TRUE if the weighted mean centre is to be used instead |
| CENTRE.x | X-coordinate of the centre |
| CENTRE.y | Y-coordinate of the centre |

| | |
|---|---|
| Sigma.x | Half-length of axis along x-axis |
| Sigma.y | Half-length of axis along y-axis |
| Major | String indicating which axis is the major elliptical axis |
| Minor | String indicating which axis is the minor elliptical axis |
| Theta | Rotation angle in degrees |
| Eccentricity | A measure of eccentricity (i.e., the flatness of the ellipse) |
| Area.sde | Area of the SDE |
| TanTheta | Trigonometric result |
| SinTheta | Trigonometric result |
| CosTheta | Trigonometric result |
| SinThetaCosTheta | Trigonometric result |
| Sin2Theta | Trigonometric result |
| Cos2Theta | Trigonometric result |
| ThetaCorr | Corrected theta angle for rotation of major axis from north |
| central.x | X-coordinate of the identified central feature |
| central.y | Y-coordinate of the identified central feature |
| median.x | X-coordinate of median centre, median value of the X-coordinate vector |
| median.y | Y-coordinate of median centre, median value of the Y-coordinate vector |
| CMD.x | X-coordinate of estimated centre of minimum distance |
| CMD.y | Y-coordinate of estimated centre of minimum distance |

## Note

Results are stored in the r.SDE object (required for graphical visualization using plot_sde). This function can be used on its own (once) or repetitively in a loop to process grouped point data stored in a larger table. When used repetitively, be sure to increment the id parameter to ensure that each SDE has a unique identifier. The output ASCII coordinate file can be further processed using the makeshapes function to generate an ESRI Shapefile for SDE polygons.

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## References

See chapter 4 of the documentation manual for CrimeStat at http://www.icpsr.umich.edu/CRIMESTAT/ and Ebdon, D. 1987. Statistics in geography. 2nd edition. New York, NY Basil Blackwell Ltd. 232 p.

## See Also

plot_sde, calc_sdd, calc_mcp, calc_box makeshapes, mcp, gridpts, ellipse3

## Examples

```
calc_sde(id=1, filename="SDE_Output.txt", centre.xy=NULL, calccentre=TRUE, weighted=FALSE, w
```

---

`centre` *Demo Data: Coordinates of a single source, centre, location*

---

## Description

This is a simple two-element vector containing x,y coordinates for a source or central location associated with a spatial point pattern. In this example, the center location represents a point of importance in an individuals daily activity pattern. Surrounding point locations are places physically contacted by an individual during a particular time interval. Demonstration data mimics UTM coordinates such that the first element represents Easting (x), and the second, Northing (y).

## Usage

```
data(centre)
```

## Format

The format is a two-element vector of numeric entries.

## Details

The coordinates of the center must have the same units and projection as the remaining point observations.

## Source

This demonstration data has been manufactured for illustrative purposes only.

## Examples

```
data(centre)
str(centre)
plot(centre)
```

cos_d *Compute cosine with angle given in degrees*

## Description

Provides the functionality of cos, but for input angles measured in degrees (not radians).

## Usage

```
cos_d(theta = 0)
```

## Arguments

| | |
|---|---|
| theta | A numeric angular measurement in degrees from north. |

## Details

Since the R default is to compute trigonometric functions on angular measurements stored in radians, this simple function performs the conversion from degrees, reducing the need to do so a priori, outside the function.

## Value

Returns a numeric value for the cosine of the specified angular measurement

## Note

To reduce the need for unit conversions prior to calling trigonometric functions, this function accepts input in angular degrees rather than radians. Depending on data, this function may be preferred to the existing version requiring input in angular radians.

## Author(s)

Tarmo K. Remmel

## See Also

sin_d, tan_d, asin_d, acos_d, atan_d

## Examples

```
cos_d(theta = 90)
```

`distances` *Multiple Euclidean distance calculator*

## Description

Compute distances from a source location (point) to a series of destination locations (points).

## Usage

```
distances(centre.xy = centre, destmat = activities, verbose = FALSE)
```

## Arguments

| | |
|---|---|
| `centre.xy` | Two-element vector containing x,y coordinates of the source location |
| `destmat` | Two-column matrix or data frame containing x,y coordinates of the activity locations |
| `verbose` | Boolean: Set to T if verbose output is desired |

## Details

Distance computations are strictly Euclidean between the source point and each destination point.

## Value

A vector of distances, where each element corresponds to one of the distance between the source point and a destination (one row) from the destinations matrix.

## Note

The order of distances in the output vector corresponds to the order of destination points in the destinations object starting at row = 1 through row = n.

## Author(s)

Tarmo K. Remmel

## Examples

```
data(centre)
data(activities)
distances(centre.xy=centre, destmat=activities, verbose=FALSE)
```

`ellipse3` *Ellipse drawing tool*

## Description

A convenient tool for plotting circles or ellipses with the functionality of rotation about an angle theta (in radians). The function can also be used to capture coordinates of the perimeter of the shape for further usage outside the function.

## Usage

```
ellipse3(cx, cy, rx, ry, theta = 0, yaxis = TRUE, pointsonly = FALSE, fill = FALSE,
```

## Arguments

| | |
|---|---|
| `cx` | x-coordinate of ellipse center |
| `cy` | y-coordinate of ellipse center |
| `rx` | radius along x-axis |
| `ry` | radius along y-axis |
| `theta` | rotation angle in radians from north |
| `yaxis` | Depreciated. This parameter adjusts the size correct in the y- or x-axis, as plotting is generally not square. Therefore, use par(pty="s") to eliminate the need for this parameter. |
| `pointsonly` | Boolean: If TRUE, ellipse will not be plotted, but rather the coordinates of the perimeter are returned in a list object for further use. The calc.sde function utilizes these coordinates to build a textfile of coordinates which the makeshapes function uses to build ESRI Shapefiles. |
| `fill` | Boolean: If TRUE, the plotted ellipse will be shaded in |
| `...` | Any additional parameters suitable for plotting |

## Details

This function plots an ellipse with center (cx, cy). The rotation angle in radians (form north) is given as theta. Note that a circle can be obtained by rx=ry, in which case theta is not very useful. Additional parameters (e.g., colour and fill density) can be provided as indicated by the '...' in the function call.

## Value

This function returns a plot of an ellipse when pointsonly = FALSE. When pointsonly = TRUE, the result is a list of x,y coordinates

| | |
|---|---|
| `x` | A numeric vector of x-coordinates |
| `y` | A numeric vector of y-coordinates |

### Note

This function is an an adjustment by Brad Biggerstaff (CDC) and Tarmo K. Remmel of the function circle() written by John Wallace (University of Washington) and obtained from the S-News listserv.

### Author(s)

Tarmo K. Remmel, Randy Bui

### See Also

calc_sde, as_radians

### Examples

```
plot.new()
plot(10,10, type="n")
ellipse3(cx = 10, cy = 8, rx = 2, ry = 1, theta = as_radians(45), yaxis = TRUE, pointsonly
```

---

## makeshapes *Builder of ESRI Shapefiles*

---

### Description

This function is a basic ESRI Shapefile builder. The SDD, SDE, and MCP functions included in this library produce ASCII output files that represent the input required for this function. However, any similarly formatted file will also work. This function provides capabilities for converting the centrographic and geometric summary measures of geographical extent and dispersion into GIS databases for further cartographic rendering and analysis.

### Usage

```
makeshapes(asciiname = "SDD_Output.txt", headerskip = 0, outname = "Test", verbose
```

### Arguments

| | |
|---|---|
| asciiname | The name of the ASCII file containing coordinate info. input to the shape building procedure |
| headerskip | An integer to indicate how many lines of the ASCII file to skip at the top if a header has been added |
| outname | The name of the output Shapefile. Do not use spaces or illegal filename characters. |
| verbose | Boolean: Set to TRUE if extended feedback to the standard output is required |

### Details

The level of detail recorded in the Shapefile will be determined by the weed tolerance of points defining the shapes in the input ASCII file.

### Value

The result is an ESRI format Shapefile containing 3 files (.shp, .dbf, .shx). The base filename will be that specified by the parameter `outname`.

### Note

Currently, the unique identifier is the only attribute that separates polygon objects within the Shapefile. Once the Shapefile is built, a collection of other attributes can be joined to the database via the unique identifier.

### Author(s)

Tarmo K. Remmel with significant help from Rick Reeves - NCEAS

### See Also

`calc_sdd`, `calc_sde`, `calc_mcp`

### Examples

```
calc_sdd(weights=NULL)
makeshapes(asciiname="SDD_Output.txt", headerskip=0, outname="Test", verbose=TRUE)
```

---

## plot_box *Plot the Standard Distance Box*

---

### Description

This function plots the standard deviation of x- and y-coordinates as a box, with the edges set, respectively, to the standard deviation of the x- and y-coordinates.

### Usage

```
plot_box(plotnew=TRUE, plothv=FALSE, plotweightedpts=FALSE, weightedpts.col='black'
```

### Arguments

| | |
|---|---|
| `plotnew` | Boolean: Set to TRUE to create a new plot. Set to FALSE to overlay current plot. |
| `plothv` | Boolean: Set to TRUE if the orthogonal N-S, E-W axes are to be plotted through the centre |
| `plotweightedpts` | Boolean: Set to TRUE if the weighted point observations are to be plotted |
| `weightedpts.col` | Specify a colour for the weighted point observations |
| `weightedpts.pch` | Specify a plotting symbol for the weighted point observations |

| | |
|---|---|
| plotpoints | Boolean: Set to TRUE if the point observations are to be plotted |
| points.col | Specify a colour for the point observations |
| points.pch | Specify a plotting symbol for the point observations |
| plotcentre | Boolean: Set to TRUE if the mean/weighted/user-defined centre is to be plotted |
| centre.col | Specify a colour for the centre |
| centre.pch | Specify a plotting symbol for the centre |
| plotcentral | Boolean: Set to TRUE if the central feature is to be highlighted |
| central.col | Specify a colour for the central feature |
| central.pch | Specify a plotting symbol for the central feature |
| plotmedian | Boolean: Set to TRUE if the median centre is to be plotted |
| median.col | Specify a colour for the median centre |
| median.pch | Specify a plotting symbol for the median centre |
| plotCMD | Boolean: Set to TRUE if the centre of minimum distance is to be plotted |
| CMD.col | Specify a colour for the centre of minimum distance |
| CMD.pch | Specify a plotting symbol for the centre of minimum distance |
| titletxt | A string to indicate the title for the plot |
| xaxis | A string to label the x-axis of the plot |
| yaxis | A string to label the y-axis of the plot |
| box.col | Specify a line colour for the SD Box |
| box.lwd | Specify a line width for the SD Box |
| jpeg | Boolean: Set to TRUE if the plot should be saved in JPEG format |
| ... | Arguments to be passed to graphical parameters |

## Details

The r.BOX object (generated using the calc_box function) is required to plot an SD Box.

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## See Also

plot_sdd, plot_mcp, plot_sde, makeshapes, ellipse3,

## Examples

```
plot_box(plotnew=TRUE, plothv=FALSE, plotweightedpts=FALSE, plotpoints=TRUE, plotcentre=TRUE
```

plot_mcp *Plot the Minimum Convex Polygon*

## Description

This function plots the MCP as a polygon, which covers the geographical extent of a set of points on a Cartesian plane.

## Usage

```
plot_mcp(plotnew=TRUE, plotpoints=TRUE, points.col='black', points.pch=1, titletxt=
```

## Arguments

| | |
|---|---|
| plotnew | Boolean: Set to TRUE to create a new plot. Set to FALSE to overlay current plot. |
| plotpoints | Boolean: Set to TRUE if the point observations are to be plotted |
| points.col | Specify a colour for the point observations |
| points.pch | Specify a plotting symbol for the point observations |
| titletxt | A string to use as the title on the plot |
| xaxis | A string to label the x-axis of the plot |
| yaxis | A string to label the y-axis of the plot |
| mcp.col | Specify the line colour for the MCP |
| mcp.lwd | Specify the line width for the MCP |
| fill.col | Specify a fill colour for the MCP |
| jpeg | Boolean: Set to TRUE if the plot should be saved in JPEG format |
| ... | Arguments to be passed to graphical parameters |

## Details

The r.MCP object (generated in calc_mcp function) is required to plot the MCP.

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## See Also

plot_sdd, plot_box, plot_sde, makeshapes, ellipse3,

## Examples

```
plot_mcp(plotnew=TRUE, plotpoints=TRUE, titletxt="Title", xaxis= "Easting (m)", yaxis="North
```

plot_sdd *Plot the Standard Distance Deviation (Standard Distance)*

## Description

This function plots the SDD as a circle with radius (standard distance), centred on a mean/weighted-mean/user-defined centre of a set of point observations.

## Usage

```
plot_sdd(plotnew=TRUE, plothv=FALSE, plotweightedpts=FALSE, weightedpts.col='black'
```

## Arguments

| | |
|---|---|
| plotnew | Boolean: Set to TRUE to create a new plot. Set to FALSE to overlay current plot. |
| plothv | Boolean: Set to TRUE if the orthogonal N-S, E-W axes are to be plotted through the centre |
| plotweightedpts | Boolean: Set to TRUE if the weighted point observations are to be plotted |
| weightedpts.col | Specify a colour for the weighted point observations |
| weightedpts.pch | Specify a plotting symbol for the weighted point observations |
| plotpoints | Boolean: Set to TRUE if the point observations are to be plotted |
| points.col | Specify a colour for the point observations |
| points.pch | Specify a plotting symbol for the point observations |
| plotcentre | Boolean: Set to TRUE if the mean/weighted/user-defined centre is to be plotted |
| centre.col | Specify a colour for the centre |
| centre.pch | Specify a plotting symbol for the centre |
| plotcentral | Boolean: Set to TRUE if the central feature is to be highlighted |
| central.col | Specify a colour for the central feature |
| central.pch | Specify a plotting symbol for the central feature |
| plotmedian | Boolean: Set to TRUE if the median centre is to be plotted |
| median.col | Specify a colour for the median centre |
| median.pch | Specify a plotting symbol for the median centre |
| plotCMD | Boolean: Set to TRUE if the centre of minimum distance is to be plotted |
| CMD.col | Specify a colour for the centre of minimum distance |
| CMD.pch | Specify a plotting symbol for the centre of minimum distance |
| titletxt | A string to indicate the title on the plot |
| xaxis | A string to label the x-axis of the plot |

| | |
|---|---|
| yaxis | A string to label the y-axis of the plot |
| sdd.col | Specify a line colour for the SDD circle |
| sdd.lwd | Specify a line width for the SDD circle |
| jpeg | Boolean: Set to TRUE if the plot should be saved in JPEG format |
| ... | Arguments to be passed to graphical parameters |

## Details

The r.SDD object (generated in calc_sdd function) is required to plot the SDD circle.

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## See Also

plot_sde, plot_mcp, plot_box, makeshapes, ellipse3,

## Examples

```
plot_sdd(plotnew=TRUE, plothv=FALSE, plotweightedpts=FALSE, plotpoints=TRUE, plotcentre=TRUE
```

---

# plot_sde *Plot the Standard Deviation Ellipse*

---

## Description

This function plots the SDE as an ellipse centred on the mean/weighted/user-defined centre of a set of point observations. The plot characterizes the dispersion of point observations along two orthogonal axes.

## Usage

```
plot_sde(plotnew=TRUE, plotSDEaxes=FALSE, plotweightedpts=FALSE, weightedpts.col='k
```

## Arguments

| | |
|---|---|
| plotnew | Boolean: Set to TRUE to create a new plot. Set to FALSE to overlay current plot. |
| plotSDEaxes | Boolean: Set to TRUE if the orthogonal axes through the centroid are to be plotted |
| plotweightedpts | Boolean: Set to TRUE if the weighted point observations are to be plotted |
| weightedpts.col | Specify a colour for the weighted point observations |

| | |
|---|---|
| weightedpts.pch | Specify a plotting symbol for the weighted point observations |
| plotpoints | Boolean: Set to TRUE if the point observations are to be plotted |
| points.col | Specify a colour for the point observations |
| points.pch | Specify a plotting symbol for the point observations |
| plotcentre | Boolean: Set to TRUE if the mean/weighted/user-defined centre is to be plotted |
| centre.col | Specify a colour for the centre |
| centre.pch | Specify a plotting symbol for the centre |
| plotcentral | Boolean: Set to TRUE if the central feature is to be highlighted |
| central.col | Specify a colour for the central feature |
| central.pch | Specify a plotting symbol for the central feature |
| plotmedian | Boolean: Set to TRUE if the median centre is to be plotted |
| median.col | Specify a colour for the median centre |
| median.pch | Specify a plotting symbol for the median centre |
| plotCMD | Boolean: Set to TRUE if the centre of minimum distance is to be plotted |
| CMD.col | Specify a colour for the centre of minimum distance |
| CMD.pch | Specify a plotting symbol for the centre of minimum distance |
| titletxt | A string to indicate the title on the plot |
| xaxis | A string to label the x-axis of the plot |
| yaxis | A string to label the y-axis of the plot |
| sde.col | Specify a line colour for the SDE circle |
| sde.lwd | Specify a line width for the SDE circle |
| jpeg | Boolean: Set to TRUE if the plot should be saved in JPEG format |
| ... | Arguments to be passed to graphical parameters |

## Details

The r.SDE object (generated in calc_sde function) is required to plot the SDE circle.

## Author(s)

Randy Bui, Ron N. Buliung, Tarmo K. Remmel

## See Also

plot_sdd, plot_mcp, plot_box, makeshapes, ellipse3,

## Examples

```
plot_sde(plotnew=TRUE, plotSDEaxes=FALSE, plotweightedpts=FALSE, plotpoints=TRUE, plotcentre
```

r.BOX *Demo Data: Standard Deviation Box Output Object*

## Description

Results from the Standard Deviation Box Calculator (calc_box) are stored in a list object. This object is required for the plot function (plot_box).

## Usage

```
data(r.BOX)
```

## Format

The list object contains the following results:

**id** Identifier for the SD box

**points** a simple two-column data frame (or matrix) containing x,y coordinates for a series of point locations.

**calcentre** Boolean: Indicates whether the mean centre was computed

**weighted** Boolean: TRUE if the weighted mean centre is to be used instead

**weights** Weights applied to point observations

**CENTRE.x** Actual, used x-coordinate of centre

**CENTRE.y** Actual, used y-coordinate of centre

**central.x** X-coordinate of central feature

**central.y** Y-coordinate of central feature

**median.x** X-coordinate of median centre

**median.y** Y-coordinate of median centre

**CMD.x** X-coordinate of centre of minimum distance

**CMD.y** Y-coordinate of centre of minimum distance

**SDD** Standard deviation distance value

**SDx** Orthogonal standard deviation in x-direction

**SDy** Orthogonal standard deviation in y-direction

**Box.area** Area of orthogonal standard deviation box

**NW.coord** Coordinates of the north-west extent of the SD Box

**NE.coord** Coordinates of the north-east extent of the SD Box

**SW.coord** Coordinates of the south-west extent of the SD Box

**SE.coord** Coordinates of the south-east extent of the SD Box

## Details

The coordinates of the points must have the same units and projection as the specified center.

### Source

This demonstration data has been manufactured for illustrative purposes only.

### Examples

```
data(r.BOX)
str(r.BOX)
```

---

## r.MCP *Demo Data: Minimum Convex Polygon Output Object*

---

### Description

Results from the Minimum Convex Polygon Calculator (calc_mcp) are stored in a list object. This object is required for the plot function (plot_mcp).

### Usage

```
data(r.MCP)
```

### Format

The list object contains the following results:

**MCP** A matrix containing MCP vertices. Each row represents a unique point, the first column contains x-coordinates, and the second, y-coordinates

**points** a simple two-column data frame (or matrix) containing x,y coordinates for a series of point locations.

**id** Identifier for the MCP

**MCP.area** The area of the MCP in square kilometers

**MCP.pct** The desired percentage of the MCP for which area is computed

### Details

The coordinates of the points must have the same units and projection as the specified center.

### Source

This demonstration data has been manufactured for illustrative purposes only.

### Examples

```
data(r.MCP)
str(r.MCP)
```

r.SDD *Demo Data: Standard Deviation Distance Output Object*

## Description

Results from the Standard Deviation Distance Calculator (calc_sdd) are stored in a list object. This object is required for the plot function (plot_sdd).

## Usage

```
data(r.SDD)
```

## Format

The list object contains the following results:

**id** Identifier for the SDD estimation - it should be unique

**points** a simple two-column data frame (or matrix) containing x,y coordinates for a series of point locations.

**SDD** SDD value, radius of the SDD

**calccentre** Boolean: TRUE if mean centre is computed

**weighted** Boolean: TRUE if the weighted mean centre is to be used instead

**weights** Weights applied to point observations

**CENTRE.x** X-coordinate of the centre

**CENTRE.y** Y-coordinate of the centre

**central.x** X-coordinate of central feature

**central.y** Y-coordinate of central feature

**median.x** X-coordinate of median centre

**median.y** Y-coordinate of median centre

**CMD.x** X-coordinate of centre of minimum distance

**CMD.y** Y-coordinate of centre of minimum distance

**SDD.area** Area of the SDD circle

## Details

The coordinates of the points must have the same units and projection as the specified center.

## Source

This demonstration data has been manufactured for illustrative purposes only.

## Examples

```
data(r.SDD)
str(r.SDD)
```

r.SDE *Demo Data: Standard Deviation Ellipse Output Object*

## Description

Results from the Standard Deviation Ellipse Calculator (calc_sde) are stored in a list object. This object is required for the plot function (plot_sde).

## Usage

```
data(r.SDE)
```

## Format

The list object contains the following results:

**id** Identifier for the SDE estimate - it should be unique

**points** a simple two-column data frame (or matrix) containing x,y coordinates for a series of point locations.

**calccentre** Boolean: TRUE if mean centre is computed

**CENTRE.x** X-coordinate of the centre

**CENTRE.y** Y-coordinate of the centre

**Sigma.x** Half-length of axis along x-axis

**Sigma.y** Half-length of axis along y-axis

**Major** String indicating which axis is the major elliptical axis

**Minor** String indicating which axis is the minor elliptical axis

**Theta** Rotation angle in degrees

**Eccentricity** A measure of eccentricity (i.e., the flatness of the ellipse)

**Area.sde** Area of the SDE

**TanTheta** Trigonometric result

**SinTheta** Trigonometric result

**CosTheta** Trigonometric result

**SinThetaCosTheta** Trigonometric result

**Sin2Theta** Trigonometric result

**Cos2Theta** Trigonometric result

**ThetaCorr** Corrected theta angle for rotation of major axis from north

**weighted** Boolean: TRUE if the weighted mean centre is to be used instead

**weights** Weights applied to point observations

**central.x** X-coordinate of central feature

**central.y** Y-coordinate of central feature

**median.x** X-coordinate of median centre

**median.y** Y-coordinate of median centre

**CMD.x** X-coordinate of centre of minimum distance

**CMD.y** Y-coordinate of centre of minimum distance

### Details

The coordinates of the points must have the same units and projection as the specified center.

### Source

This demonstration data has been manufactured for illustrative purposes only.

### Examples

```
data(r.SDE)
str(r.SDE)
```

---

## sin_d *Compute sine with angle given in degrees*

---

### Description

Provides the functionality of sin, but for input angles measured in degrees (not radians).

### Usage

```
sin_d(theta = 0)
```

### Arguments

| | |
|---|---|
| theta | A numeric angular measurement in degrees from north. |

### Details

Since the R default is to compute trigonometric functions on angular measurements stored in radians, this simple function performs the conversion from degrees, reducing the need to do so a priori, outside the function.

### Value

Returns a numeric value for the sine of the specified angular measurement

### Note

To reduce the need for unit conversions prior to calling trigonometric functions, this function accepts input in angular degrees rather than radians. Depending on data, this function may be preferred to the existing version requiring input in angular radians.

### Author(s)

Tarmo K. Remmel

## See Also

cos_d, tan_d, asin_d, acos_d, atan_d

## Examples

```
sin_d(theta = 90)
```

---

`tan_d` *Compute tangent with angle given in degrees*

---

## Description

Provides the functionality of tan, but for input angles measured in degrees (not radians).

## Usage

```
tan_d(theta = 0)
```

## Arguments

| | |
|---|---|
| `theta` | A numeric angular measurement in degrees from north. |

## Details

Since the R default is to compute trigonometric functions on angular measurements stored in radians, this simple function performs the conversion from degrees, reducing the need to do so a priori, outside the function.

## Value

Returns a numeric value for the tangent of the specified angular measurement

## Note

To reduce the need for unit conversions prior to calling trigonometric functions, this function accepts input in angular degrees rather than radians. Depending on data, this function may be preferred to the existing version requiring input in angular radians.

## Author(s)

Tarmo K. Remmel

## See Also

sin_d, cos_d, asin_d, acos_d, atan_d

## Examples

```
tan_d(theta = 45)
```

wts *Weights vector*

## Description

This is a single column vector for weighting the importance of point locations.

## Usage

```
data(wts)
```

## Format

A single column vector of numeric values.

## Details

The weights can be specified according to any reasonable criteria specified by the user

## Source

This demonstration data has been manufactured for illustrative purposes only.

## Examples

```
data(wts)
str(wts)
plot(wts)
```

# Index