Package ‘VecStatGraphs2D’

May 25, 2010

Type Package

Title Vector analysis using graphical and analytical methods in 2D.

Version 1.0

Date 2010-05-25

Author Juan Carlos Ruiz Cuetos, Maria Eugenia Polo Garcia, Pablo Garcia Rodriguez

Maintainer Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Pablo Garcia Rodriguez <pablogr@unex.es>

Depends R (>= 2.10.1), MASS

Description A vector is defined by comparing the coordinates between the origin position and the ending position in a 2D or 3D reference system. A vector has two properties, module and azimuth. To deal properly with a set of vectors, we can analyze modules and azimuths. This package performs statistical analysis and graphics of modules and azimuths distributions. The azimuths are treated as circular data. The joint analysis of modules and azimuths by means of a density maps, is due to the limitation of circular statistics in analyzing vectors with non-unitary modules.

R topics documented:

`VecStatGraphs2D-package` *Vector analysis using graphical and analytical methods in 2D.*

Description

A vector is defined by comparing the coordinates between the origin position and the ending position in a 2D or 3D reference system. A vector has two properties, module and azimuth. To deal properly with a set of vectors, we can analyze modules and azimuths. This package performs statistical analysis and graphics of modules and azimuths distributions. The azimuths are treated as circular data. The joint analysis of modules and azimuths by means of a density maps, is due to the limitation of circular statistics in analyzing vectors with non-unitary modules.

Details

Package:	VecStatGraphs2D
Type:	Package
Version:	1.0
Date:	2010-05-25
License:	GPL-3
LazyLoad:	yes

~~ An overview of how to use the package, including the most important ~~ ~~ functions ~~

Author(s)

Juan Carlos Ruiz Cuetos, Maria Eugenia Polo Garcia, Pablo Garcia Rodriguez. Maintainer: Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Pablo Garcia Rodriguez <pablogr@unex.es>

See Also

http://fegtepinta.byethost11.com/index.html

AllAzimuthStatistics *Calculation of All Statistics of the Azimuths*

Description

This function calculates several circular data statistics from a set of input azimuths.

Usage

```
AllAzimuthStatistics(azimuths)
```

Arguments

azimuths	Vector containing the values of the azimuths

Details

One way to get a set of azimuths from X and Y coordinates of the origin position and end position (coordinates X, Y of the vector), it is using the LoadData function.

The statistics calculated are: Mean Azimuth, Mean Module, Circular Variance, Circular Standard Deviation, Circular Dispersal, Von Mises Parameter, Kurtosis Coefficient, Skewness Coefficient.

Value

The value All Azimuth Statistics of all input azimuths.

Author(s)

Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Maria Eugenia Polo Garcia <mepolo@unex.es>, Pablo Garcia Rodriguez <pablogr@unex.es>

References

Bachelet E. (1981) Circular statistics in biology. London. London Academic Press.

Dixon K. R., Chapman J. A. (1980) Harmonic mean measure of animal activity areas. Ecology 61, 1040-1044.

Fisher N.I. (1995) Statistical analysis of circular data. Cambridge. Cambridge University Press.

Jammalamadaka S.R., Sengupta A. (2001) Topic in circular statistics. Singapore. World Scientific Publihing.

Mardia K.V. , Jupp, P.E. (2000) Directional statistics. Chichester. Wiley.

Russell G.S. , Levitin D.J. (1995) An expanded table of probability values for Rao´s spacing test. Communications in Statistics: Simulation and Computation, 24(4), 879-888.

Website http://fegtepinta.byethost11.com/

See Also

MeanModule, MeanAzimuth, CircularVariance, CircularStandardDeviation, CircularDispersal, KurtosisAzimuthCoefficient, VonMisesParameter, SkewnessAzimuthCoe

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
AllAzimuthStatistics(azimuths)
```

AllModuleStatistics *Calculation of All Statistics of the Modules.*

Description

This function calculates several statistics from a set of input modules.

Usage

```
AllModuleStatistics(modules)
```

Arguments

modules	Vector containing the values of the modules

Details

One way to get a set of azimuths from X and Y coordinates of the origin position and end position (coordinates X, Y of the vector), it is using the LoadData function.

The statistics calculated are: Number Of Elements, Max Value, Min Value, Range, Module Sum, Mean Arithmetic, Standard Error, Standard Deviation Module, Variance Module, Variance Module Population, Standard Deviation Module Population, Skewness Module Coefficient, Kurtosis Module Coefficient.

Value

The value All Module Statistics of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Maria Eugenia Polo Garcia <mepolo@unex.es>, Pablo Garcia Rodriguez <pablogr@unex.es>

References

Website http://fegtepinta.byethost11.com/

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `Range`, `ModuleSum`, `MeanArithmetic`, `StandardError`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviationModulePopulation`, `SkewnessModuleCoefficient`, `KurtosisModuleCoeffic`

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
AllModuleStatistics(modules)
```

CircularDispersalOfSample *Calculation of the Circular Dispersal of the Azimuths*

Description

This function calculates the Circular Dispersal from a set of input azimuths.

Usage

```
CircularDispersalOfSample(azimuths)
```

Arguments

`azimuths`	Vector containing the values of the azimuths

Details

One way to get a set of azimuths from X and Y coordinates of the origin position and end position (coordinates X, Y of the vector), it is using the `LoadData` function.

In the case of uniform distribution this value is infinite, and with values close to 0, a highly concentrated data distribution is expected.

Value

The value ciruclar Dispersal of all input azimuths.

Author(s)

Juan Carlos Ruiz Cuetos `<bilba_t@hotmail.com>`, Maria Eugenia Polo Garcia `<mepolo@unex.es>`, Pablo Garcia Rodriguez `<pablogr@unex.es>`

References

Fisher N.I. (1995) Statistical analysis of circular data. Cambridge. Cambridge University Press.

Website `http://fegtepinta.byethost11.com/`

See Also

MeanModule, MeanAzimuth, CircularVariance, CircularStandardDeviation, KurtosisAzimuthCoefficient, VonMisesParameter, SkewnessAzimuthCoefficient, AllAzimuthStatistics.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
CircularDispersal(azimuths)
```

`CircularStandardDeviation`

Calculation of the Circular Standard Deviation of the azimuths

---

Description

This function calculates the Circular Standard Deviation from a set of input azimuths.

Usage

```
CircularStandardDeviation(azimuths)
```

Arguments

`azimuths`	Vector containing the values of the azimuths

Details

One way to get a set of azimuths from X and Y coordinates of the origin position and end position (coordinates X, Y of the vector), it is using the LoadData function.

Value

The value of the Circular Standard Deviation of all input azimuths.

Author(s)

Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Maria Eugenia Polo Garcia <mepolo@unex.es>, Pablo Garcia Rodriguez <pablogr@unex.es>

References

Fisher N.I. (1995) Statistical analysis of circular data. Cambridge. Cambridge University Press.

Website http://fegtepinta.byethost11.com/

See Also

MeanModule, MeanAzimuth, CircularVariance, CircularDispersal, KurtosisAzimuthCoeffici VonMisesParameter, SkewnessAzimuthCoefficient, AllAzimuthStatistics.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
CircularStandardDeviation(azimuths)
```

`CircularVariance` *Calculation of the Circular Variance of the azimuths*

Description

This function calculates the Circular Variance from a set of input azimuths.

Usage

```
CircularVariance(azimuths)
```

Arguments

`azimuths`	Vector containing the values of the azimuths

Details

One way to get a set of azimuths from X and Y coordinates of the origin position and end position (coordinates X, Y of the vector), it is using the `LoadData` function.

Value

The value of the Circular Variance of all input azimuths. This value ranges between 0 and 1.

Author(s)

Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Maria Eugenia Polo Garcia <mepolo@unex.es>, Pablo Garcia Rodriguez <pablogr@unex.es>

References

Fisher N.I. (1995) Statistical analysis of circular data. Cambridge. Cambridge University Press.

Website http://fegtepinta.byethost11.com/

See Also

`MeanModule`, `MeanAzimuth`, `CircularStandardDeviation`, `CircularDispersal`, `KurtosisAzimuthCoefficient`, `VonMisesParameter`, `SkewnessAzimuthCoefficient`, `AllAzimuthStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
CircularVariance(azimuths)
```

DrawDensityMap *Graphic representation of a point Density Map*

Description

The density map was built as follows: a) vectors are moved to a common origin without changing its azimuth and module, b) end position perform a point cloud of varying density that depends on the accumulation of vectors with similar properties, c) end positions are used for to calculate the "density map" as a surface with values depending on the end-position accumulation per area unit. The density map gives us information about the joint distribution of modules and azimuths and allows the detection of errors or outliers. Areas where there is a greater number of points (end position of the vectors) will have a deeper color (red), while the areas a lesser density will have a less intense color (white). The outliers can be calculated by the higher value of the Modules or the Harmonic Mean.

Usage

```
DrawDensityMap(data_x, data_y)
DrawDensityMap(data_x, data_y, Div = 100)
DrawDensityMap(data_x, data_y, PaintPoint = TRUE, HarmonicMean = TRUE, PaintAxis
DrawDensityMap(data_x, data_y, PercentajeOutliers = 5, PaintPoint = TRUE, Div =
...
```

Arguments

data_x	Vector containing the values of the X coordinate
data_y	Vector containing the values of the Y coordinate
PercentajeOutliers	Integer value that indicates the percentage of outliers. The default value is 5
PaintPoint	Logical value, if PaintPoint=TRUE samples represented by points, if PaintPoint=FALSE samples not represented by points. The defaul is FALSE
Div	IntegerValue that indicates the number of divisions that will have the density map. The default value is 250
HarmonicMean	Logical value, if HarmonicMean=FALSE the module is used for the calculation of outliers, if HarmonicMean=TRUE the Harmonic Mean is used for the calculation of outliers. The default value is FALSE
PaintAxis	Logical value, if PaintAxis=TRUE draw axis, if PaintAxis=FALSE draw not axis. The default value is FALSE

Details

To create the density map, are used "Kernel descriptors", to perform these calculations is required the MASS package.

The parameter Div is very important because a very large value will cause the creation of the slow density map, and a very small value would create a ineffective density map.

One way to get a vector of X and Y coordinate from modules and azimuths of the origin position and end position (coordinates X, Y of the error), it is using the LoadData function.

Value

This function returns no value, creates a graph that represents a density map of the input values.

Author(s)

Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Maria Eugenia Polo Garcia <mepolo@unex.es>, Pablo Garcia Rodriguez <pablogr@unex.es>

References

Dixon K. R., Chapman J. A. (1980) Harmonic mean measure of animal activity areas. Ecology 61, 1040-1044.

Website http://fegtepinta.byethost11.com/

See Also

DrawHistogram, DrawDistribution, DrawPoints, DrawModuleAndAzimuthDistribution, DrawVectors.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
X_coordinate<-dat[,3]
Y_coordinate<-dat[,4]
DrawDensityMap(X_coordinate, Y_coordinate, PaintPoint = TRUE, Div=200)
```

DrawDistribution *Graphic representation of the Distribution of the azimuths*

Description

This function creates a graph that represents the azimuths distribution. Every azimuth is plotted as a blue point in a unit circle. The color red is representation of the Mean Azimuth that is in the center of the confidence interval

Usage

```
DrawDistribution(azimuths)
DrawDistribution(azimuths, Direction = 2)
```

Arguments

azimuths	Vector containing the values of the azimuths
Direction	Integer value that indicates the direction of the data. Direction 1 = Mathematical direction (0 ° E anticlockwise), direction 2 = Topographical direction (0 ° N clockwise). The default value is 2

Details

It is very important to work always with the same direction, because if the data are loaded in Topographical direction, and Graphic representation of the distribution in Mathematical direction, the resulting graph is wrong.

With this function you can see in detail where the largest and smallest concentrations of points are.

One way to get a set of azimuths from X and Y coordinates of the origin position and end position (coordinates X, Y of the vector), it is using the `LoadData` function.

Value

This function returns no value, creates a graph that represents the distribution of the input data.

Author(s)

Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Maria Eugenia Polo Garcia <mepolo@unex.es>, Pablo Garcia Rodriguez <pablogr@unex.es>

References

Website http://fegtepinta.byethost11.com/

See Also

`DrawHistogram`, `DrawPoints`, `DrawModuleAndAzimuthDistribution`, `DrawDensityMap`, `DrawVectors`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
DrawDistribution(azimuths, Direction = 2)
```

DrawHistogram *Graphic representation of the histogram of the azimuth distribution*

Description

This function creates a graph, that represents a circumference divided in sectors of different radius. The radius represents the percentage of azimuth data which belongs to each sector. Each portion represents the azimuths that exist between the angles formed by the sides of the portion.

Usage

```
DrawHistogram(azimuths)
DrawHistogram(azimuths, TamClasses = 20)
DrawHistogram(azimuths, TamClasses = 15, Direction = 2)
```

Arguments

`azimuths`	Vector containing the values of the azimuths
`TamClasses`	Integer value that represents the size of each portion. The default value is 15.
`Direction`	Integer value that indicates the direction of the data. Direction 1 = Mathematical direction (0 ° E anticlockwise), direction 2 = Topographical direction (0 ° N clockwise). The default value is 2

Details

It is very important to work always with the same direction, because if the data are loaded in Topographical direction, and Graphic representation of the Histogram in Mathematical direction, the resulting graph is wrong.

This function allows seeing where the more concentrated points are.

One way to get a set of azimuths from X and Y coordinates of the origin position and end position (coordinates X, Y of the vector), it is using the `LoadData` function.

Value

This function returns no value, creates a graph that represents a circular histogram of the input data.

Author(s)

Juan Carlos Ruiz Cuetos <bilba_t@hotmail.com>, Maria Eugenia Polo Garcia <mepolo@unex.es>, Pablo Garcia Rodriguez <pablogr@unex.es>

References

Website `http://fegtepinta.byethost11.com/`

See Also

`DrawDistribution`, `DrawPoints`, `DrawModuleAndAzimuthDistribution`, `DrawDensityMap`, `DrawVectors`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
DrawHistogram(azimuths, TamClasses=15)
```

`DrawModuleAndAzimuthDistribution` *Graphic representation of Module and Azimuth Distribution*

Description

This function creates a graph, that allows us to check the angular and linear magnitudes simultaneously, also indicated the Mean Azimuth (red color).

Usage

```
DrawModuleAndAzimuthDistribution(data_x, data_y)
DrawModuleAndAzimuthDistribution(data_x, data_y, Direction = 2)
```

Arguments

`data_x`	Vector containing the values of the X coordinate
`data_y`	Vector containing the values of the Y coordinate
`Direction`	Integer value that indicates the direction of the data. Direction 1 = Mathematical direction (0 ° E anticlockwise), direction 2 = Topographical direction (0 ° N clockwise). The default value is 2

Details

The graph draws a line for each of the samples. The lines will have origin in (0,0) and end at coordinates (X, Y). The concentric circumferences allow us to identify linear magnitudes, and the legend of the angles allows to know the angular magnitudes value.

It is very important to work always with the same direction, because if the data are loaded in Topographical direction, and Graphic representation of the module and azimuth distribution in Mathematical direction, the resulting graph is wrong.

One way to get a vector of X and Y coordinate from modules and azimuths of the origin position and end position (coordinates X, Y of the error), it is using the `LoadData` function.

Value

This function returns no value, creates a graph that represents module and azimuth distribution

Author(s)

Juan Carlos Ruiz Cuetos `<bilba_t@hotmail.com>`, Maria Eugenia Polo Garcia `<mepolo@unex.es>`, Pablo Garcia Rodriguez `<pablogr@unex.es>`

References

Website `http://fegtepinta.byethost11.com/`

See Also

`DrawHistogram`, `DrawDistribution`, `DrawPoints`, `DrawDensityMap`, `DrawVectors`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
X_coordinate<-dat[,3]
Y_coordinate<-dat[,4]
DrawModuleAndAzimuthDistribution(X_coordinate, Y_coordinate, Direction=2)
```

`DrawPoints` *Graphic representation of Points*

Description

This function creates a graph, that represents one point for each sample in the final coordinates of each vector assuming that initials are at the origin (0,0), also shows a percentage of discordant points (red color), selected from the module or Harmonic Mean, called "outliers".

Usage

```
DrawPoints(data_x, data_y)
DrawPoints(data_x, data_y, PercentajeOutliers = 3)
DrawPoints(data_x, data_y, HarmonicMean = TRUE)
DrawPoints(data_x, data_y, PercentajeOutliers = 3, HarmonicMean = TRUE)
DrawPoints(data_x, data_y, PercentajeOutliers = 5, HarmonicMean = FALSE)
...
```

Arguments

`data_x` Vector containing the values of the X coordinate

`data_y` Vector containing the values of the Y coordinate

`PercentajeOutliers`
Integer value that indicates the percentage of outliers. The default value is 5

`HarmonicMean` Logical value, if HarmonicMean=FALSE the module is used for the calculation of outliers, if HarmonicMean=TRUE the HarmonicMean is used for the calculation of outliers. The default value is FALSE

Details

All vectors are moved to a common origin (0,0) without changing its azimuth and module The graph represents the situation of the end nodes of the vectors.

One way to get a vector of X and Y coordinate from modules and azimuths of the origin position and end position (coordinates X, Y of the error), it is using the `LoadData` function.

Value

This function returns no value, creates a graph that represents the end nodes of the vectors.

Author(s)

Juan Carlos Ruiz Cuetos `<bilba_t@hotmail.com>`, Maria Eugenia Polo Garcia `<mepolo@unex.es>`, Pablo Garcia Rodriguez `<pablogr@unex.es>`

References

Website `http://fegtepinta.byethost11.com/`

See Also

`DrawHistogram`, `DrawDistribution`, `DrawModuleAndAzimuthDistribution`, `DrawDensityMap`, `DrawVectors`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
X_coordinate<-dat[,3]
Y_coordinate<-dat[,4]
DrawPoints(X_coordinate, Y_coordinate, PercentajeOutliers=8)
```

DrawVectors — *Graphic representation of Vectors*

Description

This function creates a graph, which shows an arrow for each sample, where the origin of each of the arrows is the measured position and the destination is the real position. In order to represent this graph, the input data must be of the measured and real type (type=3).

Usage

```
DrawVectors(measured_data_, real_data_)
```

Arguments

`measured_data_`	Matrix containing the coordinates measured
`real_data_`	Matrix containing the coordinates real

Details

The measured values and real values can not be calculated, must be introduced as input data, although it is advisable to use `LoadData` for obtain vector of X coordinate and Y coordinate and zimuths of measured values.

Value

This function returns no value, creates a graph that represents the vectors of the input data.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`DrawHistogram`, `DrawDistribution`, `DrawPoints`, `DrawModuleAndAzimuthDistribution`, `DrawDensityMap`.

Examples

```
FileName<-system.file("data/MeasuredRealData.txt", package="VecStat")
dat<-LoadData(FileName, Type=3, Direction=2)
measured_data_<-dat[,6:7]
real_data_<-dat[,8:9]
DrawVectors(measured_data_, real_data_)
```

KurtosisAzimuthCoefficient
Calculation of Kurtosis Coefficient azimuths

Description

This function calculates the Kurtosis Coefficient a set of input azimuths.

Usage

```
KurtosisAzimuthCoefficient(azimuths)
```

Arguments

azimuths	Vector containing the values of the azimuths

Details

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the LoadData function.

Value

The Value Kurtosis Coefficient of all input azimuths.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

MeanModule, MeanAzimuth, CircularVariance, CircularStandardDeviation, CircularDispersal, VonMisesParameter, SkewnessAzimuthCoefficient, AllAzimuthStatisti

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
KurtosisAzimuthCoefficient(azimuths)
```

`KurtosisModuleCoefficient` *Calculation of Kurtosis Coefficient modules*

Description

This function calculates the Kurtosis coefficient, which characterizes the shape of the distribution from a set of input modules.

Usage

```
KurtosisModuleCoefficient(modules)
```

Arguments

`modules`	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

The Kurtosis coefficient indicates how of sharp or flat is a distribution.

Value

The value Kurtosis coefficient of all input modules

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `Range`, `ModuleSum`, `MeanArithmetic`, `StandardError`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviationModulePopulation`, `SkewnessModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
KurtosisModuleCoefficient(modules)
```

LoadData	*Data loading and conversion of polar coordinates, rectangular coordinates, and measured and real coordinates*

Description

This function reads data from file "txt" containing data of polar type, rectangular, or measured and real and returns a matrix with all data conversions possible according Mathematical direction (0 º E anticlockwise) or Topographical direction (0 ° N clockwise)

Usage

```
LoadData(FileName)
LoadData(FileName, Type=2)
LoadData(FileName, Type = 1, Direction = 2)
```

Arguments

`FileName`	File name to load data
`Type`	Integer value that indicates the type of data contained in the file. Type 1 = Rectangular, type 2= Polar, type 3 = Measured and Real
`Direction`	Integer value that indicates the direction of the data. Direction 1 = Mathematical direction (0 º E anticlockwise), direction 2 = Topographical direction (0 ° N clockwise)

Details

Input files must be in a certain format for the data type. By default the type is Rectangular with Topographicl direction.

The rectangular type contains two columns, the first column is the X coordinate and the second column is the Y coordinate, and each row corresponds to one of the data. The X coordinate and Y coordinate are obtained from the difference of the coordinates measured with respect to the coordinates real.

The polar type contains two columns, the first column represents the module and the second column represents the azimuth, and each row corresponds to one of the data.

The measured and real type contains four columns, the first two columns represent the X and Y coordinates of the measured data and the last two columns represents the coordinates of the actual data, and each row corresponds to one of the data.

It is very important to choose the correct sense of direction, because the output data depend on this variable. The Mathematical direction, chooses the source of the degrees in the X coordinate axis, growing anticlockwise. The Topographical direction, chooses the source of the degrees in the Y coordinate axis, growing clockwise. Mathematical direction and Topographical direction follow sexagesimal angular system.

Value

The function returns a matrix nx9 size , where n is the number of data. The column 1 represents the module,the column 2 represents the azimuth, the column 3 represents the X coordinate, the column 4 represents the Y coordinate, the column 5 represents the direction and type of input data, the column 6 and 7 represent the X and Y coordinates of the measured data, the column 8 and 9 represent the X and Y coordinates of the real data.

The columns 6,7,8 and 9 will only have value if the input data are of type 3 (measured and real).

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

`DrawHistogram`, `DrawModuleAndAzimuthDistribution`, `DrawDensityMap`, `AllAzimuthStatistic`
`AllModuleStatistics`, `ValuationOfUniformity`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
LoadData(FileName, Type=1, Direction=2)
```

`MaxValue` *Calculation of Max Value modules*

Description

This function calculates the Max Value of a set of modules

Usage

```
MaxValue(modules)
```

Arguments

`modules`	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The max value of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

`NumberOfElements`, `MinValue`, `Range`, `ModuleSum`, `MeanArithmetic`, `StandardError`,
`StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviatior`
`SkewnessModuleCoefficient`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
MaxValue(modules)
```

`MeanArithmetic` *Calculation of Mean Arithmetic modules*

Description

This function calculates the Mean Arithmetic of a set of modules

Usage

```
MeanArithmetic(modules)
```

Arguments

`modules`	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The arithmetic mean value of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `Range`, `ModuleSum`, `StandardError`, `StandardDeviatio`
`VarianceModule`, `VarianceModulePopulation`, `StandardDeviationModulePopulation`,
`SkewnessModuleCoefficient`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
MeanArithmetic(modules)
```

`MeanAzimuth` *Calculation of Mean Azimuth azimuths*

Description

This function calculates the Mean Azimuth a set of input azimuths.

Usage

```
MeanAzimuth(azimuths)
```

Arguments

`azimuths`	Vector containing the values of the azimuths

Details

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The Value Mean Azimuth of all input azimuths. The value is expressed in sexagesimal.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`MeanModule`, `CircularVariance`, `CircularStandardDeviation`, `CircularDispersal`, `KurtosisAzimuthCoefficient`, `VonMisesParameter`, `SkewnessAzimuthCoefficient`, `AllAzimuthStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
MeanAzimuth(azimuths)
```

`MeanModule` *Calculation of Mean Module azimuths*

Description

This function calculates the Mean Module through a set of input azimuths.

Usage

```
MeanModule(azimuths)
```

Arguments

`azimuths`	Vector containing the values of the azimuths

Details

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The Value Mean module of all input azimuths.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`MeanAzimuth`, `CircularVariance`, `CircularStandardDeviation`, `CircularDispersalOfSample`, `KurtosisAzimuthCoefficient`, `VonMisesParameter`, `SkewnessAzimuthCoefficient`, `AllAzimuthStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
MeanModule(azimuths)
```

MinValue — *Calculation of Min Value modules*

Description

This function calculates the Min Value of a set of modules

Usage

```
MinValue(modules)
```

Arguments

`modules`	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The min value of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

`NumberOfElements`, `MaxValue`, `Range`, `ModuleSum`, `MeanArithmetic`, `StandardError`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviation`
`SkewnessModuleCoefficient`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
MinValue(modules)
```

`ModuleSum` *Calculation of Module Sum*

Description

This function calculates the sum of all modules.

Usage

```
ModuleSum(modules)
```

Arguments

`modules`	Vector containing the values of the module

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The sum of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `Range`, `MeanArithmetic`, `StandardError`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviatio`
`SkewnessModuleCoefficient`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
ModuleSum(modules)
```

NumberOfElements *Calculation of Number of Elements modules*

Description

This function calculates the Number of Elements of a set of modules

Usage

```
NumberOfElements(modules)
```

Arguments

modules	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the LoadData function.

Value

The number of elements of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

MaxValue, MinValue, Range, ModuleSum, MeanArithmetic, StandardError, StandardDeviationMo
VarianceModule, VarianceModulePopulation, StandardDeviationModulePopulation, SkewnessModuleCoefficient, KurtosisModuleCoefficient, AllModuleStatistics.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
NumberOfElements(modules)
```

`Range` *Calculation of Range modules*

Description

This function calculates the Range (Difference between maximum and minimum value of the module) of a set of modules

Usage

```
Range(modules)
```

Arguments

`modules`	Vector containing the values of the module

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The range value of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `ModuleSum`, `MeanArithmetic`, `StandardError`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviation`
`SkewnessModuleCoefficient`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
Range(modules)
```

RaoTest *Rao Test. Formal test of uniformity*

Description

This function does the Rao test. This test is presented as more robust than the Rayleigh test.

Usage

```
RaoTest(azimuths)
RaoTest(azimuths, Alpha = 0.01)
```

Arguments

`azimuths`	Vector containing the values of the azimuths
`Alpha`	Value used to obtain the Rao Value of the Rao Table. The values can be 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 0.9. The default is 0.01.

Details

For a uniform distribution, the space between points should be about `360°/n`. If the space deviates much from this value, the probability that the data belong to a uniform distribution is reduced.

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

Returns the probability value, and indicates whether or not to accept the hypothesis of uniformity.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`ValuationOfUniformity`, `RayleighTest`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
RaoTest(azimuths, Alpha = 0.01)
```

`RayleighTest` *Rayleigh Test. Formal test of uniformity*

Description

This function does the Rayleigh test, a formal tests of uniformity simplest and most used.

Usage

```
RayleighTest(azimuths)
```

Arguments

`azimuths`	Vector containing the values of the azimuths

Details

We are supposed that we not must to know the mean address specified . The hypothesis of uniformity is rejected if the mean length of the resultant vector of the sample is very large. This test assumes that a value the mean module most large implies a concentration more around the mean, and therefore less probability that the data is uniformly distributed.

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

Returns the value of the probability of accepting the hypothesis of uniformity.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

`ValuationOfUniformity`, `RaoTest`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
RayleighTest(azimuths)
```

SkewnessAzimuthCoefficient

Calculation of Skewness Coefficient azimuths

Description

This function calculates the Skewness Coefficient a set of input azimuths.

Usage

```
SkewnessAzimuthCoefficient(azimuths)
```

Arguments

azimuths	Vector containing the values of the azimuths

Details

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the LoadData function.

Value

The Value Skewness Coefficient of all input azimuths.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

MeanModule, MeanAzimuth, CircularVariance, CircularStandardDeviation, CircularDispersal, KurtosisAzimuthCoefficient, VonMisesParameter, AllAzimuthStatisti

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
SkewnessAzimuthCoefficient(azimuths)
```

`SkewnessModuleCoefficient`
Calculation of Skewness Coefficient modules

Description

This function calculates the skewness coefficient, and indicates the symmetry of the distribution from a set of input modules.

Usage

```
SkewnessModuleCoefficient(modules)
```

Arguments

`modules`	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

The normal distribution is symmetric if the skewness is 0.

Value

The value skewnwess coefficient of all input modules

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `Range`, `ModuleSum`, `MeanArithmetic`, `StandardError`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviationModulePopulation`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
SkewnessModuleCoefficient(modules)
```

StandardDeviationModule
Calculation of Standard Deviation modules

Description

This function calculates the standard deviation of a set of input modules.

Usage

```
StandardDeviationModule(modules)
```

Arguments

modules	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the LoadData function.

Value

The standard deviation of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

NumberOfElements, MaxValue, MinValue, Range, ModuleSum, MeanArithmetic, StandardError, VarianceModule, VarianceModulePopulation, StandardDeviationModulePopu SkewnessModuleCoefficient, KurtosisModuleCoefficient, AllModuleStatistics.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
StandardDeviationModule(modules)
```

`StandardDeviationModulePopulation`
Calculation of Population Standard Deviation modules

Description

This function calculates the population standard deviation of a set of input modules.

Usage

```
StandardDeviationModulePopulation(modules)
```

Arguments

`modules`	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The population standard deviation of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `Range`, `ModuleSum`, `MeanArithmetic`, `StandardError`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `SkewnessModuleCoefficient`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
StandardDeviationModulePopulation(modules)
```

`StandardError` *Calculation of Standard Error modules*

Description

This function calculates the standard error of the arithmetic mean of a set of input modules.

Usage

```
StandardError(modules)
```

Arguments

`modules`	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

The standard Error of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

`NumberOfElements`, `MaxValue`, `MinValue`, `Range`, `ModuleSum`, `MeanArithmetic`, `StandardDeviationModule`, `VarianceModule`, `VarianceModulePopulation`, `StandardDeviatio` `SkewnessModuleCoefficient`, `KurtosisModuleCoefficient`, `AllModuleStatistics`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
StandardError(modules)
```

`ValuationOfUniformity` *Valuation graph of the uniformity*

Description

This function generates a graph that allows you to see quickly if the data distribution can be considered uniform or if it departs conspicuously from the uniformity.

Usage

```
ValuationOfUniformity(azimuths)
```

Arguments

`azimuths`	Vector containing the values of the azimuths

Details

To create the graph are placed in increasing order all azimuths and are drawn in a rectangular coordinate system.

If the points drawn, are located on a line of about 45 degrees, passing through the origin, we can say that the data fit the uniform model.

The arbitrary choice of origin, it may produce that the graph of distribution of points to induces a misunderstanding in the interpretation. To solve this problem, adding the unit to each element of the pair of values of the first 20 percent of points, and subtract the unit to each element of the pair of values of the last 20 percent.

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the `LoadData` function.

Value

This function returns no value, what it does is display a chart to show if be accepted or rejected the hypothesis of uniform distribution.

Author(s)

Juan Carlos Ruiz Cuetos `<jruizcue@alumnos.unex.es>`

See Also

`RayleighTest`, `RaoTest`.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
ValuationOfUniformity(azimuths)
```

VarianceModule *Calculation of Variance modules*

Description

This function calculates the variance of a set of input modules.

Usage

```
VarianceModule(modules)
```

Arguments

modules	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the LoadData function.

Value

The variance of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

NumberOfElements, MaxValue, MinValue, Range, ModuleSum, MeanArithmetic, StandardError, StandardDeviationModule, VarianceModulePopulation, StandardDeviationM SkewnessModuleCoefficient, KurtosisModuleCoefficient, AllModuleStatistics.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
VarianceModule(modules)
```

VarianceModulePopulation
Calculation of Population Variance modules

Description

This function calculates the population variance of a set of input modules.

Usage

```
VarianceModulePopulation(modules)
```

Arguments

modules	Vector containing the values of the modules

Details

One way to get a vector of modules from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the LoadData function.

Value

The population variance of all input modules.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

NumberOfElements, MaxValue, MinValue, Range, ModuleSum, MeanArithmetic, StandardError, StandardDeviationModule, VarianceModule, StandardDeviationModulePopul SkewnessModuleCoefficient, KurtosisModuleCoefficient, AllModuleStatistics.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
modules<-dat[,1]
VarianceModulePopulation(modules)
```

VonMisesParameter *Calculation of Von Mises concentration parameter azimuths*

Description

This function calculates the Von Mises concentration parameter a set of input azimuths.

Usage

```
VonMisesParameter(azimuths)
```

Arguments

azimuths	Vector containing the values of the azimuths

Details

One way to get a vector of azimuths from X and Y coordinates of measured values and real values (coordinates (X, Y) of the error), It is using the LoadData function.

The Von Mises concentration parameter, measures the variation in the distribution in relation of a perfect circle. If this parameter tends to 0, the distribution converges to the uniform distribution, if tend to infinity, the distribution is concentrated in mean direction of the resultant vector.

Value

The Von Mises concentration parameter of all input azimuths.

Author(s)

Juan Carlos Ruiz Cuetos <jruizcue@alumnos.unex.es>

See Also

MeanModule, MeanAzimuth, CircularVariance, CircularStandardDeviation, CircularDispersal, KurtosisAzimuthCoefficient, SkewnessAzimuthCoefficient, AllAzimuthStatistics.

Examples

```
FileName<-system.file("data/RectangularData.txt", package="VecStat")
dat<-LoadData(FileName, Type=1, Direction=2)
azimuths<-dat[,2]
VonMisesParameter(azimuths)
```

Index