Twitter Tools

1.1.0.

Generated by Doxygen 1.8.6

Wed Apr 16 2014 16:20:50

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 Package Twitter_tools

Classes

- class alpha_fitter

  This class reads two series from files and determines the constant factor which minimizes the sum of squared differences of the two dataset by multipliing the first one.
- class correlation

  Class reads geolocations of users and makes random user-pairs. Then it calculates the distance distribution of the result graph.
- class dist_distribution

  Calculates the cumulative distance distribution of the mutual mention graph.
- class dist_distribution_2

  Class calculates the distance distribution of the mutual follower graph.
- class dist_distribution_short_range

  Class reads geolocations of users and makes random user-pairs which are close to eah other compared to the radius of Earth. Then it calculates the distance distribution of the result graph by an approximated distance formula.
- class Distribution< T >

  This class represents both Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF) of any numeric data with variable bin size.
- class geoPoint

  This class represents a point on the surface of the Earth or a point on a unit sphere
- class gnuplot_caller

  Test object o call GNUPLOT from a C# console application.
- class hashtag_parser

  This class reads all of the tweets from a database and parses all of the hashtags out of it.
- class ListAndCNT

  This class stores certain properties of a particular hashtag. Contains all of the coordinates of the occurances.
- class Pairs

  This class stores certain properties of a particular hashtag. Contans only the standard deviance about the location information.
- class PairsLowMem

  The minimalist version of the Pars class. Contains only the number of occurances and the standard deviance with memory friendly types.
- class hashtag_sigma

  This class reads all the hashtags from a file grouped by tags. It calculates the average and standard deviance of the particula tags.
- class Hist1D< T >

This class represents a 1 dimensional histogram. Similar to the class Distribution, but bin size is constant and is not normalized.

- class Hist2D< T, U >

  This class represents a 2 dimensional histogram. Similar to the class Distribution, but 2 dimensional, bin size is constant and is not normalized.

- class Program

  Contains the Main function. Only initializes an instance of an other object and call the "run()" method of it.

- class rank_dist

  Class reads links with weights above a certain threshold from the mutual mention graph and calculates the rank distribution of it.

- class ratio

  This class reads the counts of hashtags and calculatesthe average ratio of get-tagged and all hashtags.

- class ratio_dist

  Class reads the ratios of back and forth mention counts and makes the PDF of it.

- class scatter

  Class generates a scatterplot and histogram of the users in mutual follower degree-mutual mention degree space.

- class sigma_tester

  Dummy class for testing new methods. Contents often change.

- class time_measurer

  Ths class executes a particular query with different numbers of lines and measures the ecexution time in seconds of each execution.

- class TwitterBase

  This class is intended to be the base class all of the other classes which read from an SQL database and proces Twitter data.

Chapter 5

Class Documentation

5.1 Twitter_tools.alpha_fitter Class Reference

This class reads two series from files and determines the constant factor which minimizes the sum of squared differences of the two dataset by multipliing the first one.

Public Member Functions

- void run ()

  The 'main' function of the class. We call this method when we use the class. It's contents can change.
- void read (string filename1, string filename2)

  Reads two data files and finds the constant factor to multiply the first with.
- double S (List< double > list1, List< double > list2, double alpha)

  Sum of squared differences of alpha ∗ list1 and list2.
- double Diff (List< double > list1, List< double > list2, double alpha, S_delegate f)

  First order numerical derivative of the S_delegate type function f.
- double Diff_2 (List< double > list1, List< double > list2, double alpha, S_delegate f)

  Second order numerical derivative of the S_delegate type function f.
- delegate double S_delegate (List< double > list1, List< double > list2, double alpha)

  A delegate function which fits for S, the sum of squared differences of alpha ∗ list1 and list2.
- double Newton (S_delegate S, List< double > list1, List< double > list2)

  Minimizes the S_delegate type function S by changing it's parameter alpha.

5.1.1 Detailed Description

This class reads two series from files and determines the constant factor which minimizes the sum of squared differences of the two dataset by multipliing the first one.

This class contains methods to find a constant factor which minimizes the sum of squared differences between two datasets. It uses the Newton-Raphson method, and contains methods for first and second order derivatives.

5.1.2 Member Function Documentation

5.1.2.1 double Twitter_tools.alpha_fitter.Diff (List< double > *list1,* List< double > *list2,* double *alpha,* S_delegate *f*) [inline]

First order numerical derivative of the S_delegate type function f.

Parameters

list1	First double List type parameter of the function f.
list2	Second double List type parameter of the function f.
alpha	The double type parameter of the function f.
f	An S_delegate type function. The method will calculate the first derivative of this.

Returns

the first order numerical derivative of the S_delegate type function f at the given parameters.

5.1.2.2 double Twitter_tools.alpha_fitter.Diff_2 (List< double > *list1*, List< double > *list2*, double *alpha*, S_delegate *f*) [inline]

Second order numerical derivative of the S_delegate type function f.

Parameters

list1	First double List type parameter of the function f.
list2	Second double List type parameter of the function f.
alpha	The double type parameter of the function f.
f	An S_delegate type function. The method will calculate the second derivative of this.

Returns

the second order numerical derivative of the S_delegate type function f at the given parameters.

5.1.2.3 double Twitter_tools.alpha_fitter.Newton (S_delegate *S*, List< double > *list1*, List< double > *list2*) [inline]

Minimizes the S_delegate type function S by changing it's parameter alpha.

Finds the minimum of S by using the Newton-Raphson method to find the root of the first derivative of S.

Parameters

S	An S_delegate type function.
list1	First double List type parameter of the function S.
list2	Second double List type parameter of the function S.

Returns

a double value which minimizes S_delegate type S as the alpha parameter of it with the given parameters list1 and list2.

5.1.2.4 void Twitter_tools.alpha_fitter.read (string *filename1*, string *filename2*) [inline]

Reads two data files and finds the constant factor to multiply the first with.

It reads the files from the path \QSO Documents\ and appends .dat extension to them.

Parameters

filename1	The first filename without path and extension. The program multiplies this by the fit factor.
filename2	The second filename without path and extension.

5.1.2.5 void Twitter_tools.alpha_fitter.run ( ) [inline]

The 'main' function of the class. We call this method when we use the class. It's contents can change.

5.1.2.6 double Twitter_tools.alpha_fitter.S (List< double > *list1*, List< double > *list2*, double *alpha*) [inline]

Sum of squared differences of alpha ∗ list1 and list2.

This method finds the ninimum of the sizes of list1 and list2 and sums the squared differences over the common section.

Parameters

list1	List object of double variables. Program will multiply this List by the parameter alpha.
list2	List object of double variables.
alpha	A constant double factor the method multiplies the first List with.

Returns

a double value whish is the sum of the squared differences of alpha ∗ list1 and list2.

5.1.2.7 delegate double Twitter_tools.alpha_fitter.S_delegate (List< double > *list1*, List< double > *list2*, double *alpha*)

A delegate function which fits for S, the sum of squared differences of alpha ∗ list1 and list2.

Parameters

list1	First double List type parameter of the function f.
list2	Second double List type parameter of the function f.
alpha	The double type parameter of the function f.

Returns

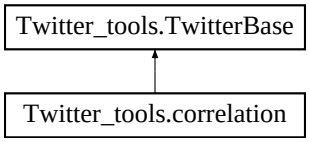
The documentation for this class was generated from the following file:

- alpha_fitter.cs

5.2 Twitter_tools.correlation Class Reference

Class reads geolocations of users and makes random user-pairs. Then it calculates the distance distribution of the result graph.

Inheritance diagram for Twitter_tools.correlation:



Public Member Functions

- void run ()

  The 'main' function of the class. Reads geolocations of users and makes random user-pairs. Then it calculates the cumulative distance distribution of the result graph and writes it to a file named 'tavolsageloszlas_vilag_korrelacio' at defaultPath.

Additional Inherited Members

5.2.1 Detailed Description

Class reads geolocations of users and makes random user-pairs. Then it calculates the distance distribution of the result graph.

5.2.2 Member Function Documentation

5.2.2.1 void Twitter_tools.correlation.run ( ) [inline]

The 'main' function of the class. Reads geolocations of users and makes random user-pairs. Then it calculates the cumulative distance distribution of the result graph and writes it to a file named 'tavolsageloszlas_vilag_korrelacio' at defaultPath.

The documentation for this class was generated from the following file:

- correlation.cs

5.3 Twitter_tools.Distribution< T >.DFpoint Struct Reference

This struct represents a point of a Distribution Function. Just an x-y pair of variables. Type of x is T, and y is double.

Public Attributes

- T x

  Element of the domain of DF.

- double y

  Element of the codomain of DF. Output to the element x.

5.3.1 Detailed Description

This struct represents a point of a Distribution Function. Just an x-y pair of variables. Type of x is T, and y is double.

5.3.2 Member Data Documentation

5.3.2.1 T Twitter_tools.Distribution< T >.DFpoint.x

Element of the domain of DF.

5.3.2.2 double Twitter_tools.Distribution< T >.DFpoint.y

Element of the codomain of DF. Output to the element x.

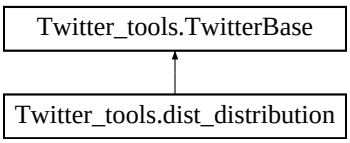
The documentation for this struct was generated from the following file:

- distribution.cs

5.4 Twitter_tools.dist_distribution Class Reference

Calculates the cumulative distance distribution of the mutual mention graph.

Inheritance diagram for Twitter_tools.dist_distribution:



Public Member Functions

- void run ()

  "Main" function of the class. Calculates the cumulative distance distribution of the mutual mention graph and writes it to a .dat file named tavolsageloszlas_vilag_PDF_scalar-product_nocold at the defaultPath.

Additional Inherited Members

5.4.1 Detailed Description

Calculates the cumulative distance distribution of the mutual mention graph.

5.4.2 Member Function Documentation

5.4.2.1 void Twitter_tools.dist_distribution.run ( ) [inline]

"Main" function of the class. Calculates the cumulative distance distribution of the mutual mention graph and writes it to a .dat file named tavolsageloszlas_vilag_PDF_scalar-product_nocold at the defaultPath.

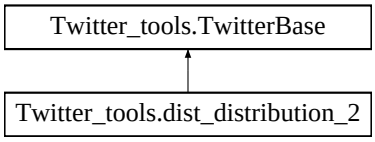
The documentation for this class was generated from the following file:

- dist_distribution.cs

5.5 Twitter_tools.dist_distribution_2 Class Reference

Class calculates the distance distribution of the mutual follower graph.

Inheritance diagram for Twitter_tools.dist_distribution_2:



Public Member Functions

- void run ()

  "Main" function of the object. Calculates the cumulative distance distribution of the mutual follower graph and writes it to a .dat file named tavolsageloszlas_vilag_PDF_follow_mutual at defaultPath.

Additional Inherited Members

5.5.1 Detailed Description

Class calculates the distance distribution of the mutual follower graph.

5.5.2 Member Function Documentation

5.5.2.1 void Twitter_tools.dist_distribution_2.run ( ) [inline]

"Main" function of the object. Calculates the cumulative distance distribution of the mutual follower graph and writes it to a .dat file named tavolsageloszlas_vilag_PDF_follow_mutual at defaultPath.

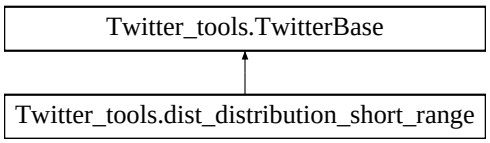
The documentation for this class was generated from the following file:

- dist_distribution_2.cs

5.6 Twitter_tools.dist_distribution_short_range Class Reference

Class reads geolocations of users and makes random user-pairs which are close to eah other compared to the radius of Earth. Then it calculates the distance distribution of the result graph by an approximated distance formula.

Inheritance diagram for Twitter_tools.dist_distribution_short_range:



Public Member Functions

- void run ()

  "Main" function of the object. Calculates the distance distribution of the random graph of Twitter users by an approximated distance formula and writes it to a .dat file named tavolsageloszlas_vilag_korrelacio_SR at defaultPath. Can be easily modified to read points only from the USA or Great Britain.

Additional Inherited Members

5.6.1 Detailed Description

Class reads geolocations of users and makes random user-pairs which are close to eah other compared to the radius of Earth. Then it calculates the distance distribution of the result graph by an approximated distance formula.

5.6.2 Member Function Documentation

5.6.2.1 void Twitter_tools.dist_distribution_short_range.run ( ) [inline]

"Main" function of the object. Calculates the distance distribution of the random graph of Twitter users by an approximated distance formula and writes it to a .dat file named tavolsageloszlas_vilag_korrelacio_SR at defaultPath. Can be easily modified to read points only from the USA or Great Britain.

The documentation for this class was generated from the following file:

- dist_distribution_short_range.cs

5.7 Twitter_tools.Distribution< T > Class Template Reference

This class represents both Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF) of any numeric data with variable bin size.

Classes

- struct DFpoint

  This struct represents a point of a Distribution Function. Just an x-y pair of variables. Type of x is T, and y is double.

Public Member Functions

- Distribution (List< T > data, int binNumber_arg)

  Initializes a distribution instance. Puts elements of data to binNumber_arg variable sized bins.
- void writeToFile (string fileName, string path=TwitterBase.defaultPath, string dataName="data", string comment="")

  Writes the CDF and PDF data to a .dat file at the given path with the given filename.
- void writeToFileNoLog (string fileName, string path=TwitterBase.defaultPath, string dataName="data", string comment="")

  Writes the CDF and PDF data to a .dat file at the given path with the given filename, but without their logarithms.

Public Attributes

- List< T > data

  Stores a set of T type comparable values. Class puts these elements to different sized bins.
- int binNumber

  Number of the bins in the DFs.
- int sampleNumber

  Number of the samples (= values) stored in one bin.
- List< DFpoint > CDF = new List<DFpoint>()

  The Cumulative Distribution Function itself as a List of DFpoint structs.
- List< DFpoint > PDF = new List<DFpoint>()

  The Probability Distributon Function itself as a List of DFpoint structs.

5.7.1 Detailed Description

This class represents both Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF) of any numeric data with variable bin size.

Both of the PDF and CDF functions are normalized to 1.

Template Parameters

T	Numeric type. This class will put this type of numbers to bins. Must implement IComparable interface and must be able to be converted to double.

Type Constraints

T* : *IComparable

5.7.2 Constructor & Destructor Documentation

5.7.2.1 Twitter_tools.Distribution< T >.Distribution (List< T > *data*, int *binNumber_arg*) [inline]

Initializes a distribution instance. Puts elements of data to binNumber_arg variable sized bins.

This method sorts the List data, and splits it to (approximately) binNumber_arg parts. Each part becomes a bin, so the size of the bin is variable, but the y size of the bin is (apprximately) fixed. This is the Probability Distribution Function. After it, the method numerically derives the PDF so we get the Cumulative distribution function, CDF.

Exceptions

System.InvalidOperationException	Thrown when method can not convert T to double.

Parameters

data	List of T type objects. T must implement System.IComparabe interface, and can be converted to double. If not, this method throws a System.InvalidOperationException exception.
binNumber_arg	The number of the bins, i.e. the number of parts we split the data into.

5.7.3 Member Function Documentation

5.7.3.1 void Twitter_tools.Distribution< T >.writeToFile (string *fileName*, string *path* = **TwitterBase.defaultPath**, string *dataName* = "data", string *comment* = "") [inline]

Writes the CDF and PDF data to a .dat file at the given path with the given filename.

This method uses the System.TextWriter class to write to a file. The first two lines are comments, beginning with the '#' mark for further working with the file in GNUPLOT. User can name the data so in the comments that name will be shown. Each line of the file represents a bin. The method writes the index of the bin, the element of data at the bin limits (= the x value of the bin), PDF and CDF value (= the y value) of the bin, and the logarithm of each respectively for easier data visualization. Method will write 0s instead of the log of the zero values. Base of logarithms is 10.

Exceptions

System.InvalidOperationException	Thrown when method can not convert T to double.

Parameters

fileName	Method will name the .dat file afterthis at the given path. Must not contain extension.
path	Method will save the file at this path. Can be omitted. If omitted, method will save file to TwitterBase.defaultPath.
dataName	Name of the data. Appear in the dat description line of the file. Default value is "data".
comment	Optional comment line. Method will write this string at the first line of the file after a '#' mark. If omitted, first line is left uot entirely.

5.7.3.2 void Twitter_tools.Distribution< T >.writeToFileNoLog (string *fileName*, string *path* = TwitterBase.defaultPath, string *dataName* = "data", string *comment* = "") [inline]

Writes the CDF and PDF data to a .dat file at the given path with the given filename, but without their logarithms.

This method uses the System.TextWriter class to write to a file. The first two lines are comments, beginning with the '#' mark for further working with the file in GNUPLOT. User can name the data so in the comments that name will be shown. Each line of the file represents a bin. The method writes the index of the bin, the element of data at the bin limits (= the x value of the bin), PDF and CDF value (= the y value) of the bin

Parameters

fileName	Method will name the .dat file afterthis at the given path. Must not contain extension.
path	Method will save the file at this path. Can be omitted. If omitted, method will save file to TwitterBase.defaultPath.
dataName	Name of the data. Appear in the dat description line of the file. Default value is "data".
comment	Optional comment line. Method will write this string at the first line of the file after a '#' mark. If omitted, first line is left uot entirely.

5.7.4 Member Data Documentation

5.7.4.1 int Twitter_tools.Distribution< T >.binNumber

Number of the bins in the DFs.

5.7.4.2 List<DFpoint> Twitter_tools.Distribution< T >.CDF = new List<DFpoint>()

The Cumulative Distribution Function itself as a List of DFpoint structs.

5.7.4.3 List<T> Twitter_tools.Distribution< T >.data

Stores a set of T type comparable values. Class puts these elements to different sized bins.

5.7.4.4 List<DFpoint> Twitter_tools.Distribution< T >.PDF = new List<DFpoint>()

The Probability Distributon Function itself as a List of DFpoint structs.

5.7.4.5 int Twitter_tools.Distribution< T >.sampleNumber

Number of the samples (= values) stored in one bin.

The documentation for this class was generated from the following file:

- distribution.cs

5.8 Twitter_tools.geoPoint Class Reference

This class represents a point on the surface of the Earth or a point on a unit sphere

Public Member Functions

- double norm ()

  Returns the norm (length) of the x, y, z vector.
- void normalize ()

  Normalizes the vector. Changes the length to 1 and leaves the direction.
- geoPoint ()

  Initializes a geoPoint instance with 0, 0, 0 coordinates.
- geoPoint (double setlon, double setlat)

  Initializes a geoPoint instance with the givel longitude and latitude coordinates,
- geoPoint (double x, double y, double z)

  Initializes a geoPoint instance with the given x,y,z coordinates.
- geoPoint (geoPoint GP)

  Initializes a geoPoint instance equals to the givel geoPoint instance.
- geoPoint (Random random)

  Generates a geoPoint on a random point of the unit sphere with uniform spherical distribution.
- void setLatLon ()

  Sets the latitude and longitude coordinates based on the x, y, z coordinates.
- void setXYZ ()

  Sets the x, y, z coordinates based on tha latitude and longitude coordinates.
- override string ToString ()

  Returns the String representation of the geoPoint instance.
- string getLatLonString ()

  Returns the String representation of the geoPoint instance.

Public Attributes

- double cx

  The x coordinate of the unit vector representation of the geoPoint.
- double cy

  The y coordinate of the unit vector representation of the geoPoint.
- double cz

  The z coordinate of the unit vector representation of the geoPoint.
- double lat

  The latitude coordinate of the geoPoint.
- double lon

  The longitude coordinate of the geoPoint.

5.8.1 Detailed Description

This class represents a point on the surface of the Earth or a point on a unit sphere

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Twitter_tools.geoPoint.geoPoint ( ) [inline]

Initializes a geoPoint instance with 0, 0, 0 coordinates.

5.8.2.2 Twitter_tools.geoPoint.geoPoint (double *setlon,* double *setlat*) [inline]

Initializes a geoPoint instance with the givel longitude and latitude coordinates,

Parameters

setlon	The longitude coordinate.
setlat	The latitude coordinate.

5.8.2.3 Twitter_tools.geoPoint.geoPoint (double *x*, double *y*, double *z*) [inline]

Initializes a geoPoint instance with the given x,y,z coordinates.

Parameters

x	The x coordinate of the geoPoint.
y	The y coordinate of the geoPoint.
z	The z coordinate of the geoPoint.

5.8.2.4 Twitter_tools.geoPoint.geoPoint (geoPoint *GP*) [inline]

Initializes a geoPoint instance equals to the givel geoPoint instance.

Parameters

GP	

5.8.2.5 Twitter_tools.geoPoint.geoPoint (Random *random*) [inline]

Generates a geoPoint on a random point of the unit sphere with uniform spherical distribution.

Mathematical background at http://mathworld.wolfram.com/SpherePointPicking.html

Parameters

random	Random class of the System.Random library.

5.8.3 Member Function Documentation

5.8.3.1 string Twitter_tools.geoPoint.getLatLonString ( ) [inline]

Returns the String representation of the geoPoint instance.

Returns the longitude and latitude coordinates in String.

Returns

5.8.3.2 double Twitter_tools.geoPoint.norm ( ) [inline]

Returns the norm (length) of the x, y, z vector.

Returns

5.8.3.3 void Twitter_tools.geoPoint.normalize ( ) [inline]

Normalizes the vector. Changes the length to 1 and leaves the direction.

5.8.3.4 void Twitter_tools.geoPoint.setLatLon ( ) [inline]

Sets the latitude and longitude coordinates based on the x, y, z coordinates.

5.8.3.5 void Twitter_tools.geoPoint.setXYZ ( ) [inline]

Sets the x, y, z coordinates based on tha latitude and longitude coordinates.

5.8.3.6 override string Twitter_tools.geoPoint.ToString ( ) [inline]

Returns the String representation of the geoPoint instance.

Returns the x, y, z coordinates in String.

Returns

5.8.4 Member Data Documentation

5.8.4.1 double Twitter_tools.geoPoint.cx

The x coordinate of the unit vector representation of the geoPoint.

5.8.4.2 double Twitter_tools.geoPoint.cy

The y coordinate of the unit vector representation of the geoPoint.

5.8.4.3 double Twitter_tools.geoPoint.cz

The z coordinate of the unit vector representation of the geoPoint.

5.8.4.4 double Twitter_tools.geoPoint.lat

The latitude coordinate of the geoPoint.

5.8.4.5 double Twitter_tools.geoPoint.lon

The longitude coordinate of the geoPoint.

The documentation for this class was generated from the following file:

- geoPoint.cs

5.9 Twitter_tools.gnuplot_caller Class Reference

Test object o call GNUPLOT from a C# console application.

Public Member Functions

- void run ()

  The "main" funciotn of the object.

5.9.1 Detailed Description

Test object o call GNUPLOT from a C# console application.

5.9.2 Member Function Documentation

5.9.2.1 void Twitter_tools.gnuplot_caller.run ( ) [inline]

The "main" funciotn of the object.

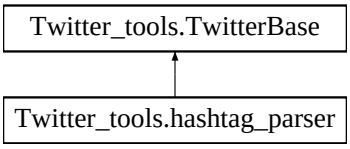
The documentation for this class was generated from the following file:

- gnuplot_caller.cs

5.10 Twitter_tools.hashtag_parser Class Reference

This class reads all of the tweets from a database and parses all of the hashtags out of it.

Inheritance diagram for Twitter_tools.hashtag_parser:



Public Member Functions

- void run ()

  The "main" function of the class. Reads all of the tweets from a database and parses all of the hashtags out of it.

Additional Inherited Members

5.10.1 Detailed Description

This class reads all of the tweets from a database and parses all of the hashtags out of it.

This class is derived from the class SQL_reader. It reads all the tweets from the database 'future1' ansd seeks for all the hashtags.

5.10.2 Member Function Documentation

5.10.2.1 void Twitter_tools.hashtag_parser.run ( ) [inline]

The "main" function of the class. Reads all of the tweets from a database and parses all of the hashtags out of it.

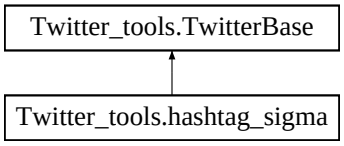
The documentation for this class was generated from the following file:

- hashtag_parser.cs

5.11 Twitter_tools.hashtag_sigma Class Reference

This class reads all the hashtags from a file grouped by tags. It calculates the average and standard deviance of the particula tags.

Inheritance diagram for Twitter_tools.hashtag_sigma:



Public Member Functions

- void run ()

  The "main" function of the class. Reads all the hashtags from a txt file grouped by tags. It calculates the average and standard deviance of the particular tags and writes them to files.

Additional Inherited Members

5.11.1 Detailed Description

This class reads all the hashtags from a file grouped by tags. It calculates the average and standard deviance of the particula tags.

5.11.2 Member Function Documentation

5.11.2.1 void Twitter_tools.hashtag_sigma.run ( ) [inline]

The "main" function of the class. Reads all the hashtags from a txt file grouped by tags. It calculates the average and standard deviance of the particular tags and writes them to files.

Works in the directory stored at defaultPath. Reads data from a file called 'hashtag.txt' and writes 2 dimensional histogram to 'hashtag-graf_sigma_CNT-gt_hist_linlinlin.dat', 'average to hashtag-graf_avg-sigma_VS_CNT-gt_linlinlin.-dat', and detailed hashtag info to 'hashtag-graf_sigma_VS_count_linlin_2.dat'.

The documentation for this class was generated from the following file:

- hashtag_sigma.cs

5.12 Twitter_tools.Hist1D< T > Class Template Reference

This class represents a 1 dimensional histogram. Similar to the class Distribution, but bin size is constant and is not normalized.

Public Member Functions

- int whichBin (T data)

  Returns the index of the bin in the histogram which contains the 'data' if the bin size is given.
- void addData (T data)

  Adds 1 to the bin data belongs to.

- void fillWith (int value)

  Fills all of the bins with the given value.
- Hist1D (List< T > dataSet, double binSize_, int binNumber_)

  Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements of dataSet.
- void writeToFile (string fileName, string path=TwitterBase.defaultPath, string dataName="data", string comment="")
- void writeToFile (string fileName, double substitute, string path=TwitterBase.defaultPath, string dataName="data", string comment="")

Public Attributes

- int binNumber

  Number of bins in the histogram.
- double binSize

  Size (= width) of one bin. "Right side" of the histogram is binNumber ∗ binSize.
- int[] hist

  Class stores the values in this array. This is the histogram itself.

5.12.1 Detailed Description

This class represents a 1 dimensional histogram. Similar to the class Distribution, but bin size is constant and is not normalized.

Template Parameters

T	Class puts this type of data to bins. Must be able to be converted to double.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 Twitter_tools.Hist1D< T >.Hist1D (List< T > *dataSet*, double *binSize_*, int *binNumber_*) [inline]

Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements of dataSet.

Parameters

dataSet	List of T type objects. Elements must be able to be converted to double.
*binSize_*	Size (= width) of one bin.
*binNumber_*	Number of the binSize_ sized bins.

5.12.3 Member Function Documentation

5.12.3.1 void Twitter_tools.Hist1D< T >.addData (T *data*) [inline]

Adds 1 to the bin data belongs to.

Parameters

data	The numerical data which we want to put in a bin. Type must be able to be converted to double.

5.12.3.2 void Twitter_tools.Hist1D< T >.fillWith (int *value*) [inline]

Fills all of the bins with the given value.

Parameters

value	Sets all of the bin values to this value.

5.12.3.3 int Twitter_tools.Hist1D< T >.whichBin (T *data*) [inline]

Returns the index of the bin in the histogram which contains the 'data' if the bin size is given.

Exceptions

System.InvalidOperationException	Thrown when method can not convert T to double.

Parameters

data	The numerical data which we want to put in a bin. Type must be able to be converted to double.

Returns

an integer which is the index of the bin data belongs to.

5.12.3.4 void Twitter_tools.Hist1D< T >.writeToFile (string *fileName*, string *path* = **TwitterBase.defaultPath**, string *dataName* = "data", string *comment* = "") [inline]

This method uses the System.TextWriter class to write to a file. The first two lines are comments, beginning with the '#' mark for further working with the file in GNUPLOT. User can name the data so in the comments that name will be shown. Each line of the file represents a bin. The method writes the index of the bin, the bin limit, and the integer value of the bin.

Parameters

fileName	Method will name the .dat file afterthis at the given path. Must not contain extension.
path	Method will save the file at this path. Can be omitted. If omitted, method will save file to TwitterBase.defaultPath.
dataName	Name of the data. Appear in the dat description line of the file. Default value is "data".
comment	Optional comment line. Method will write this string at the first line of the file after a '#' mark. If omitted, first line is left uot entirely.

5.12.3.5 void Twitter_tools.Hist1D< T >.writeToFile (string *fileName*, double *substitute*, string *path* = **TwitterBase.defaultPath**, string *dataName* = "data", string *comment* = "") [inline]

This method uses the System.TextWriter class to write to a file. The first two lines are comments, beginning with the '#' mark for further working with the file in GNUPLOT. User can name the data so in the comments that name will be shown. Each line of the file represents a bin. The method writes the index of the bin, the bin limit, and the integer value of the bin.

Parameters

fileName	Method will name the .dat file afterthis at the given path. Must not contain extension.
substitute	In the file, method writes this value in the place of zeros. It is useful if we draw data in logscale.

path	Method will save the file at this path. Can be omitted. If omitted, method will save file to TwitterBase.defaultPath.
dataName	Name of the data. Appear in the dat description line of the file. Default value is "data".
comment	Optional comment line. Method will write this string at the first line of the file after a '#' mark. If omitted, first line is left uot entirely.

5.12.4 Member Data Documentation

5.12.4.1 int Twitter_tools.Hist1D< T >.binNumber

Number of bins in the histogram.

5.12.4.2 double Twitter_tools.Hist1D< T >.binSize

Size (= width) of one bin. "Right side" of the histogram is binNumber ∗ binSize.

5.12.4.3 int [ ] Twitter_tools.Hist1D< T >.hist

Class stores the values in this array. This is the histogram itself.

The documentation for this class was generated from the following file:

- hist1D.cs

5.13 Twitter_tools.Hist2D< T, U > Class Template Reference

This class represents a 2 dimensional histogram. Similar to the class Distribution, but 2 dimensional, bin size is constant and is not normalized.

Public Member Functions

- int whichBin< V > (V data, double binSize)

  Returns the index of the bin in the histogram which contains the 'data' if the bin size is given.
- void addData (Tuple< T, U > data)

  Adds 1 to the bin data belongs to.
- void addData (T x_data, U y_data)

  Adds 1 to the bin data belongs to.
- void fillWith (int value)

  Fills all of the bins with the given value.
- Hist2D (List< Tuple< T, U >> dataSet, double x_binSize_, int x_binNumber_, double y_binSize_, int y_binNumber_)

  Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements of dataSet.
- Hist2D (double x_binSize_, int x_binNumber_, double y_binSize_, int y_binNumber_)

  Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements of zeros.
- Hist2D (List< Hist1D< T >> hists, double x_binSize_)

  Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements stored in a list of Hist1D objects.
- void writeToFile (string fileName, string path=TwitterBase.defaultPath, string x_dataName="x", string y_dataName="y", string comment="", bool transposed=false, double substitute=0.0)

  Writes the data stored in the histogram to a .dat file with the given name at the given path in matrix format.

Public Attributes

- int x_binNumber

  Number of x bins in the histogram.
- double x_binSize

  Size (= width) of one x bin. "Right side" of the histogram is x_binNumber ∗ x_binSize.
- int y_binNumber

  Number of y bins in the histogram.
- double y_binSize

  Size (= width) of one y bin. "Bottom" of the histogram is y_binNumber ∗ y_binSize.
- int[,] hist

  Class stores the values in this array. This is the histogram itself.

5.13.1 Detailed Description

This class represents a 2 dimensional histogram. Similar to the class Distribution, but 2 dimensional, bin size is constant and is not normalized.

Template Parameters

T	Class puts this type of x data to bins. Must be able to be converted to double.
U	Class puts this type of y data to bins. Must be able to be converted to double.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 Twitter_tools.Hist2D< T, U >.Hist2D (List< Tuple< T, U >> *dataSet*, double *x_binSize_*, int *x_binNumber_*, double *y_binSize_*, int *y_binNumber_*) [inline]

Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements of dataSet.

Parameters

dataSet	List of T, U type Tuples. T and U types must be able to be converted to double.
*x_binSize_*	Size (= width) of one x bin.
*x_binNumber_*	Number of the x_binSize_ sized bins.
*y_binSize_*	Size (= width) of one y bin.
*y_binNumber_*	Number of the y_binSize_ sized bins.

5.13.2.2 Twitter_tools.Hist2D< T, U >.Hist2D (double *x_binSize_*, int *x_binNumber_*, double *y_binSize_*, int *y_binNumber_*) [inline]

Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements of zeros.

Parameters

*x_binSize_*	Size (= width) of one x bin.
*x_binNumber_*	Number of the x_binSize_ sized bins.
*y_binSize_*	Size (= width) of one y bin.
*y_binNumber_*	Number of the y_binSize_ sized bins.

5.13.2.3 Twitter_tools.Hist2D< T, U >.Hist2D (List< Hist1D< T >> *hists*, double *x_binSize_*) [inline]

Initializes a new instance of Hist1D with tha given bin size and bin number, filled with the elements stored in a list of Hist1D objects.

Each column (y) will be one of the Hist1D objects of the hists parameter. If the Hist1D objects have different binSize attibutes, the y_binSize attribute of this object will be the maximum of them. The missing elements will be initialized as zeros. The y_binSize of this object will be the binSize of the binSize first Hist1D object, no matter what are the others. The x_binNumber of this object will be the number of the Hist1D object hists parameter contains.

Parameters

hists	List of Hist1D objects. Each object of this List will be one of the columns of the Hist2D object.
*x_binSize_*	The size (= width) of the x bins in the Hist2D object.

5.13.3 Member Function Documentation

5.13.3.1 void Twitter_tools.Hist2D< T, U >.addData (Tuple< T, U > *data*) [inline]

Adds 1 to the bin data belongs to.

Parameters

data	The numerical data which we want to put in a bin. Expects a tuple with the x and y values. Both's type must be able to be converted to double.

5.13.3.2 void Twitter_tools.Hist2D< T, U >.addData (T *x_data*, U *y_data*) [inline]

Adds 1 to the bin data belongs to.

Parameters

x_data	The numerical x data which we want to put in a bin. Type T must be able to be converted to double.
x_data	The numerical y data which we want to put in a bin. Type U must be able to be converted to double.

5.13.3.3 void Twitter_tools.Hist2D< T, U >.fillWith (int *value*) [inline]

Fills all of the bins with the given value.

Parameters

value	Sets all of the bin values to this value.

5.13.3.4 int Twitter_tools.Hist2D< T, U >.whichBin< V > (V *data*, double *binSize*) [inline]

Returns the index of the bin in the histogram which contains the 'data' if the bin size is given.

Exceptions

System.InvalidOperationException	Thrown when method can not convert T to double.

Parameters

data	The numerical data which we want to put in a bin. Type must be able to be converted to double.

Returns

an integer which is the index of the bin data belongs to.

5.13.3.5 void Twitter_tools.Hist2D< T, U >.writeToFile (string *fileName*, string *path* = **TwitterBase.defaultPath**, string *x_dataName* = "x", string *y_dataName* = "y", string *comment* = "", bool *transposed* = false, double *substitute* = 0.0) [inline]

Writes the data stored in the histogram to a .dat file with the given name at the given path in matrix format.

This method uses the System.TextWriter class to write to a file. The first two lines are comments, beginning with the '#' mark for further working with the file in GNUPLOT. User can name the data so in the comments that name will be shown. The method writes the integer values of the bin in matrix format.

Parameters

fileName	Method will name the .dat file after this at the given path. Must not contain extension.
path	Method will save the file at this path. Can be omitted. If omitted, method will save file to TwitterBase.defaultPath.
x_dataName	Name of the x data. Appear in the dat description line of the file. Default value is "x".
y_dataName	Name of the y data. Appear in the dat description line of the file. Default value is "y".
comment	Optional comment line. Method will write this string at the first line of the file after a '#' mark. If omitted, first line is left uot entirely.
transposed	If true, rows will represent the fixed x values, and columns are fixed y values. If false, rows are ys, columns are xs. By default, transposed is false.
substitute	If not 0, the method will substitute zero values with the given double number. Useful when plotting in logscale. Equals to 0 if omitted.

5.13.4 Member Data Documentation

5.13.4.1 int [,] Twitter_tools.Hist2D< T, U >.hist

Class stores the values in this array. This is the histogram itself.

5.13.4.2 int Twitter_tools.Hist2D< T, U >.x_binNumber

Number of x bins in the histogram.

5.13.4.3 double Twitter_tools.Hist2D< T, U >.x_binSize

Size (= width) of one x bin. "Right side" of the histogram is x_binNumber $*$ x_binSize.

5.13.4.4 int Twitter_tools.Hist2D< T, U >.y_binNumber

Number of y bins in the histogram.

5.13.4.5 double Twitter_tools.Hist2D< T, U >.y_binSize

Size (= width) of one y bin. "Bottom" of the histogram is y_binNumber $*$ y_binSize.

The documentation for this class was generated from the following file:

- Hist2D.cs

5.14 Twitter_tools.ListAndCNT Class Reference

This class stores certain properties of a particular hashtag. Contains all of the coordinates of the occurances.

Public Member Functions

- ListAndCNT (string tag_, List< geoPoint > List_, int CNT_, int CNT_gt_)

  Initializes a ListAndCNT instance with the given attributes.

Public Attributes

- string tag

  The string representation of the tag.
- List< geoPoint > List

  Stores a list of geo-coordinates where the hashtag was published.
- int CNT

  The count of the tag was tagged.
- int CNT_gt

  The count of the tag was tagged in tweets with coordinates.

5.14.1 Detailed Description

This class stores certain properties of a particular hashtag. Contains all of the coordinates of the occurances.

5.14.2 Constructor & Destructor Documentation

5.14.2.1 Twitter_tools.ListAndCNT.ListAndCNT (string *tag_*, List< geoPoint > *List_*, int *CNT_*, int *CNT_gt_*) [inline]

Initializes a ListAndCNT instance with the given attributes.

Parameters

*tag_*	The tag attribute of the ListAndCNT class.
*List_*	The List attribute of the ListAndCNT class.
*CNT_*	The CNT attribute of the ListAndCNT class.
*CNT_gt_*	The CNT_gt attribute of the ListAndCNT class.

5.14.3 Member Data Documentation

5.14.3.1 int Twitter_tools.ListAndCNT.CNT

The count of the tag was tagged.

Not all tweets are geo-tagged so it can be different from CNT_gt.

5.14.3.2 int Twitter_tools.ListAndCNT.CNT_gt

The count of the tag was tagged in tweets with coordinates.

Not all tweets are geo-tagged so it can be different from CNT.

5.14.3.3 List<geoPoint> Twitter_tools.ListAndCNT.List

Stores a list of geo-coordinates where the hashtag was published.

5.14.3.4 string Twitter_tools.ListAndCNT.tag

The string representation of the tag.

The 'tag' is case insensitive.

The documentation for this class was generated from the following file:

- hashtag_sigma.cs

5.15 Twitter_tools.Pairs Class Reference

This class stores certain properties of a particular hashtag. Contans only the standard deviance about the location information.

Public Member Functions

- Pairs (string tag_, double sigma_, int CNT_, int CNT_gt_)

  Initializes a Pairs instance with the given attributes.

Public Attributes

- string tag

  The string representation of the tag.
- double sigma

  The standard deviance of coordinates of the geotagged occurances of the hashtag.
- int CNT

  The count of the tag was tagged.
- int CNT_gt

  The count of the tag was tagged in tweets with coordinates.

5.15.1 Detailed Description

This class stores certain properties of a particular hashtag. Contans only the standard deviance about the location information.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 Twitter_tools.Pairs.Pairs (string *tag_*, double *sigma_*, int *CNT_*, int *CNT_gt_*) [inline]

Initializes a Pairs instance with the given attributes.

Parameters

*tag_*	The tag attribute of the Pairs class.
*sigma_*	The sigma attribute of the Pairs class.
*CNT_*	The CNT attribute of the Pairs class.
*CNT_gt_*	The CNT_gt attribute of the Pairs class.

5.15.3 Member Data Documentation

5.15.3.1 int Twitter_tools.Pairs.CNT

The count of the tag was tagged.

Not all tweets are geo-tagged so it can be different from CNT_gt.

5.15.3.2 int Twitter_tools.Pairs.CNT_gt

The count of the tag was tagged in tweets with coordinates.

Not all tweets are geo-tagged so it can be different from CNT.

5.15.3.3 double Twitter_tools.Pairs.sigma

The standard deviance of coordinates of the geotagged occurances of the hashtag.

5.15.3.4 string Twitter_tools.Pairs.tag

The string representation of the tag.

The 'tag' is case insensitive.

The documentation for this class was generated from the following file:

- hashtag_sigma.cs

5.16 Twitter_tools.PairsLowMem Class Reference

The minimalist version of the Pars class. Contains only the number of occurances and the standard deviance with memory friendly types.

Public Member Functions

- PairsLowMem (double sigma_, int CNT_gt_)

  Initializes a new PairsLowMem instance with the given attributes.

Public Attributes

- float sigma

  The standard deviance of coordinates of the geotagged occurances of the hashtag.
- ushort CNT_gt

  The count of the tag was tagged in tweets with coordinates.

5.16.1 Detailed Description

The minimalist version of the Pars class. Contains only the number of occurances and the standard deviance with memory friendly types.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Twitter_tools.PairsLowMem.PairsLowMem (double *sigma_*, int *CNT_gt_*) [inline]

Initializes a new PairsLowMem instance with the given attributes.

Parameters

*sigma_*	The sigma attribute of te PairsLowMem class.
*CNT_gt_*	The CNT_gt attribute of te PairsLowMem class.

5.16.3 Member Data Documentation

5.16.3.1 ushort Twitter_tools.PairsLowMem.CNT_gt

The count of the tag was tagged in tweets with coordinates.

5.16.3.2 float Twitter_tools.PairsLowMem.sigma

The standard deviance of coordinates of the geotagged occurances of the hashtag.

The documentation for this class was generated from the following file:

- hashtag_sigma.cs

5.17 Twitter_tools.Program Class Reference

Contains the Main function. Only initializes an instance of an other object and call the "run()" method of it.

5.17.1 Detailed Description

Contains the Main function. Only initializes an instance of an other object and call the "run()" method of it.

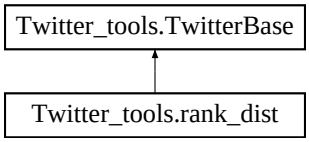
The documentation for this class was generated from the following file:

- Program.cs

5.18 Twitter_tools.rank_dist Class Reference

Class reads links with weights above a certain threshold from the mutual mention graph and calculates the rank distribution of it.

Inheritance diagram for Twitter_tools.rank_dist:



Public Member Functions

- void run (int[] thresholds)

  The "main" funtcion of the class. Reads links with weights above a certain threshold from the mutual mention graph and calculates the rank distribution of it, then it writes the results to .dat files called mention_mutual_num and mention_mutual_rank_dist_tr.

Additional Inherited Members

5.18.1 Detailed Description

Class reads links with weights above a certain threshold from the mutual mention graph and calculates the rank distribution of it.

5.18.2 Member Function Documentation

5.18.2.1 void Twitter_tools.rank_dist.run (int[] *thresholds*) [inline]

The "main" funtcion of the class. Reads links with weights above a certain threshold from the mutual mention graph and calculates the rank distribution of it, then it writes the results to .dat files called mention_mutual_num and mention_mutual_rank_dist_tr.

Saves results to .dat files at defaultPath. Saves the number of read lines (= links) for each threshold to 'mention_-mutual_num'. Saves the rank distribution for each threshold to 'mention_mutual_rank_dist'. Here each row represents a distribution for a particular rank, and each column is for a fixed threshold. Saves the transposed matrix of the distribution written to 'mention_mutual_rank_dist' to the file 'mention_mutual_rank_dist_tr'. Here each row has a fixed threshold and each column has a ficed rank. Writing both of the matrix and it's transposed matrix helps in data visualization.

Parameters

thresholds	An int list of the thresholds. The method will calculate the histogram for each threshold given in this List.

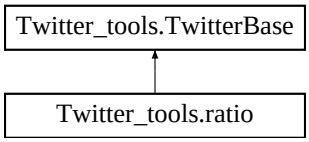
The documentation for this class was generated from the following file:

- rank_dist.cs

5.19 Twitter_tools.ratio Class Reference

This class reads the counts of hashtags and calculatesthe average ratio of get-tagged and all hashtags.

Inheritance diagram for Twitter_tools.ratio:



Public Member Functions

- void run ()

  The "main" function of the class. Reads the counts of hashtags and calculatesthe average ratio of get-tagged and all hashtags.

Additional Inherited Members

5.19.1 Detailed Description

This class reads the counts of hashtags and calculatesthe average ratio of get-tagged and all hashtags.

5.19.2 Member Function Documentation

5.19.2.1 void Twitter_tools.ratio.run ( ) [inline]

The "main" function of the class. Reads the counts of hashtags and calculatesthe average ratio of get-tagged and all hashtags.

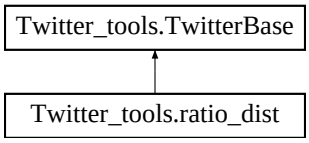
The documentation for this class was generated from the following file:

- ratio.cs

5.20 Twitter_tools.ratio_dist Class Reference

Class reads the ratios of back and forth mention counts and makes the PDF of it.

Inheritance diagram for Twitter_tools.ratio_dist:



Public Member Functions

- void run ()

  The "main" function of the class. Reads the ratios of back and forth mention counts and makes the PDF of their ratio.

Additional Inherited Members

5.20.1 Detailed Description

Class reads the ratios of back and forth mention counts and makes the PDF of it.

5.20.2 Member Function Documentation

5.20.2.1 void Twitter_tools.ratio_dist.run ( ) [inline]

The "main" function of the class. Reads the ratios of back and forth mention counts and makes the PDF of their ratio.

Saves results to a .dat file named 'ratio-dist' at the defaultPath.

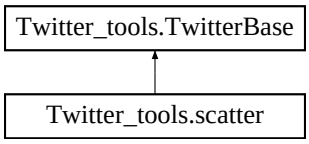
The documentation for this class was generated from the following file:

- ratio_dist.cs

5.21 Twitter_tools.scatter Class Reference

Class generates a scatterplot and histogram of the users in mutual follower degree-mutual mention degree space.

Inheritance diagram for Twitter_tools.scatter:



Public Member Functions

- void run ()

  The "main function of the class. Generates a scatterplot and histogram of the users in mutual follower degree-mutual mention degree space. Writes results to the .dat file called scatter_hist.

Additional Inherited Members

5.21.1 Detailed Description

Class generates a scatterplot and histogram of the users in mutual follower degree-mutual mention degree space.

5.21.2 Member Function Documentation

5.21.2.1 void Twitter_tools.scatter.run ( ) [inline]

The "main function of the class. Generates a scatterplot and histogram of the users in mutual follower degree-mutual mention degree space. Writes results to the .dat file called scatter_hist.

Saves results to a .dat file named 'scatter_hist' at defaultPath.

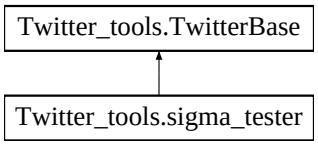
The documentation for this class was generated from the following file:

- scatter.cs

5.22 Twitter_tools.sigma_tester Class Reference

Dummy class for testing new methods. Contents often change.

Inheritance diagram for Twitter_tools.sigma_tester:



Public Member Functions

- void run ()

  The "main" function of the class.

Additional Inherited Members

5.22.1 Detailed Description

Dummy class for testing new methods. Contents often change.

5.22.2 Member Function Documentation

5.22.2.1 void Twitter_tools.sigma_tester.run ( ) [inline]

The "main" function of the class.

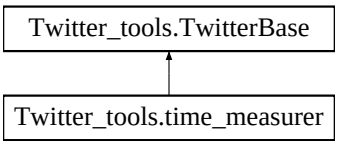
The documentation for this class was generated from the following file:

- sigma_tester.cs

5.23 Twitter_tools.time_measurer Class Reference

Ths class executes a particular query with different numbers of lines and measures the ecexution time in seconds of each execution.

Inheritance diagram for Twitter_tools.time_measurer:



Public Member Functions

- void run ()

  The "main" function of the class. Executes a particular query with different numbers of lines and measures the ecexution time in seconds of each execution. Saves the results to a .dat file called time_measurer at the default path.

Additional Inherited Members

5.23.1 Detailed Description

Ths class executes a particular query with different numbers of lines and measures the ecexution time in seconds of each execution.

5.23.2 Member Function Documentation

5.23.2.1 void Twitter_tools.time_measurer.run ( ) [inline]

The "main" function of the class. Executes a particular query with different numbers of lines and measures the ecexution time in seconds of each execution. Saves the results to a .dat file called time_measurer at the default path.

Saves results to the file 'time_measurer.dat' at defaultPath.

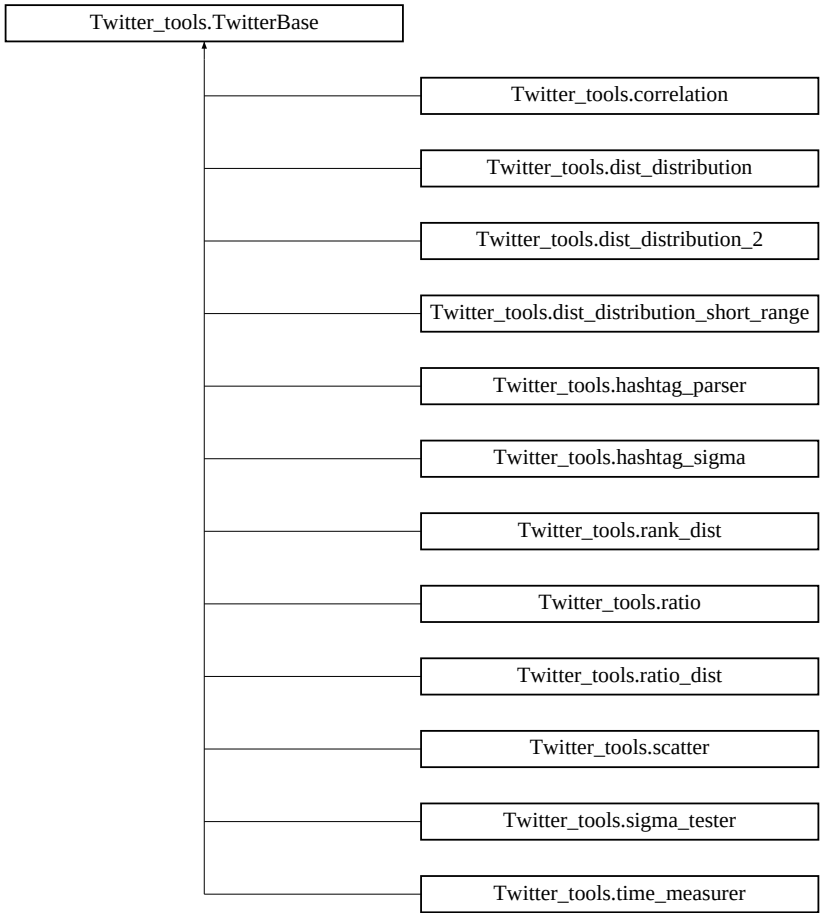
The documentation for this class was generated from the following file:

- time_measurer.cs

5.24 Twitter_tools.TwitterBase Class Reference

This class is intended to be the base class all of the other classes which read from an SQL database and proces Twitter data.

Inheritance diagram for Twitter_tools.TwitterBase:



Public Member Functions

- double getDistance (geoPoint u, geoPoint v)

  Gives the distance of two geoPoint objects n the surface of the unit sphere.
- double getApproxDistance (geoPoint u, geoPoint v)

  Gives the approximate distance of two close geoPoint objects on the surface of the unit sphere.
- geoPoint GetAvg (List< geoPoint > gps)

  Gives the average of the given set of geoPoint objects.
- double GetStdDev (List< geoPoint > gps)

  Calculates the standard deviance of the given set of geoPoint objects.
- double GetAvgAbsDev (List< geoPoint > gps)

  Calculates the average absolute deviation of the given set of geoPoint objects.
- void showTime (Stopwatch stopWatch)

  Stops the given Stopwatch instance and writes it's current state to the screen.
- void initNoExe (string com)

  Sets all of the attributes of SQL_reader superclass to the most commonly used values. Does noot start to execute the query.
- void init (string com)

Sets all of the attributes of SQL_reader superclass to the most commonly used values.

- void close ()

  Closes the SqlConnection con connection property of the SQL_reader superclass (probably) opened in the init(string com) method.

Public Attributes

- SqlConnection con

  SqlConnection class for connecting to the Twitter database
- SqlConnectionStringBuilder csb = new SqlConnectionStringBuilder()

  SqlConnectionStringBuilder class for
- SqlCommand command

  Contains the SQL query.
- SqlDataReader reader

  Iterates through the database and reads the table line by line.
- const double R = 6372.797

  The average radius of Earth.
- const string defaultPath = "\\\\QSO\\Public\\rudolf\\My Documents\\"

  The default directory each class writes to and reads from.

5.24.1 Detailed Description

This class is intended to be the base class all of the other classes which read from an SQL database and proces Twitter data.

The class contains all of teh attributes we need in our particular project and methods which help to initialize them, and other useful attributes and methods used by many classes.

5.24.2 Member Function Documentation

5.24.2.1 void Twitter_tools.TwitterBase.close ( ) [inline]

Closes the SqlConnection con connection property of the SQL_reader superclass (probably) opened in the init(string com) method.

5.24.2.2 double Twitter_tools.TwitterBase.getApproxDistance (geoPoint *u*, geoPoint *v*) [inline]

Gives the approximate distance of two close geoPoint objects on the surface of the unit sphere.

Approximates the surface with a plan and gives the magnitude of the difference vector of the two given vectors as the distance of the points. The order of the two arguments does not matter.

Parameters

u	The first geoPoint object.
v	The second geoPoint object.

Returns

with the approximate distance of the two given close geoPoint objects in radian. The return type is double.

5.24.2.3 geoPoint Twitter_tools.TwitterBase.GetAvg (List< geoPoint > *gps*) [inline]

Gives the average of the given set of geoPoint objects.

Sums all the unit vectors and normalizes after, so the result vector will be on the surface of the unit sphere as well. The method initialzes a new geoPoint object with the resulting unt vector.

Parameters

gps	List object of geoPoint objects. Can contain any number of geoPoint objects.

Returns

a geoPoint object which is the "middle" of the set given as the parameter.

5.24.2.4 double Twitter_tools.TwitterBase.GetAvgAbsDev (List< geoPoint > *gps*) [inline]

Calculates the average absolute deviation of the given set of geoPoint objects.

Returns a double number which is the average absolute deviation of the given set of geoPoint objects. The method calculates the average of the geoPoint set, then it sums the differences and takes the average of differences. Returns -1 is the average is a NaN.

Parameters

gps	List object of geoPoint objects. Can contain any number of geoPoint objects.

Returns

5.24.2.5 double Twitter_tools.TwitterBase.getDistance (geoPoint *u*, geoPoint *v*) [inline]

Gives the distance of two geoPoint objects n the surface of the unit sphere.

Based on the scalar product of the two unit vectors. The order of the two arguments does not matter. May return NaN.

Parameters

u	The first geoPoint object.
v	The second geoPoint object.

Returns

with the distance of the two given geoPoint objects in radian. The return type is double.

5.24.2.6 double Twitter_tools.TwitterBase.GetStdDev (List< geoPoint > *gps*) [inline]

Calculates the standard deviance of the given set of geoPoint objects.

Returns a double number which is the standard deviance of the given set of geoPoint objects. The method calculates the average of the geoPoint set, then it sums the squared differences and takes the square root of the average of squared differences. Returns -1 is the average is a NaN.

Parameters

gps	List object of geoPoint objects. Can contain any number of geoPoint objects.

Returns

a double value which is the standard deviance of the given set of geoPoint objects in radian. The return value is -1 if sigma is NaN for some reason.

5.24.2.7 void Twitter_tools.TwitterBase.init (string *com*) [inline]

Sets all of the attributes of SQL_reader superclass to the most commonly used values.

Sets the datasource to 'future1', and the initial catalog to 'rudolf'. Integrated Security to 'true', ConnectionTimeOut to '100 000'. Sets the command to the given string in the parameter, opens the connection and starts to execute the query so after calling this method we only have to use the reader.Read() command.

Parameters

com	Expects a string variable containing a valid SQL query. reader will execute this query.

5.24.2.8 void Twitter_tools.TwitterBase.initNoExe (string *com*) [inline]

Sets all of the attributes of SQL_reader superclass to the most commonly used values. Does noot start to execute the query.

Sets the datasource to 'future1', and the initial catalog to 'rudolf'. Integrated Security to 'true', ConnectionTimeOut to '100 000'. Sets the command to the given string in the parameter, opens the connection, so after calling this method we only have to execute the query and use the reader.Read() command.

Parameters

com	Expects a string variable containing a valid SQL query. reader will execute this query.
noExecute	A boolean variable. The value of it does not have any effect on the behavior of the method. Only the existence of it prevents the function from executing the query.

5.24.2.9 void Twitter_tools.TwitterBase.showTime (Stopwatch *stopWatch*) [inline]

Stops the given Stopwatch instance and writes it's current state to the screen.

The method expects a System.Diagnostics.Stopwatch instance as a parameter and writes it's current Elapsed property to the screen in "Elapsed time: HH:MM:SS.SS" format.

Parameters

stopWatch	System.Diagnostics.Stopwatch object instance.

5.24.3 Member Data Documentation

5.24.3.1 SqlCommand Twitter_tools.TwitterBase.command

Contains the SQL query.

5.24.3.2 SqlConnection Twitter_tools.TwitterBase.con

SqlConnection class for connecting to the Twitter database

5.24.3.3 SqlConnectionStringBuilder Twitter_tools.TwitterBase.csb = new SqlConnectionStringBuilder()

SqlConnectionStringBuilder class for

5.24.3.4 const string Twitter_tools.TwitterBase.defaultPath = "\\\\QSO\\Public\\rudolf\\My Documents\\"

The default directory each class writes to and reads from.

Originally it is set to \\QSO\Public\rudolf\My Documents\

5.24.3.5 const double Twitter_tools.TwitterBase.R = 6372.797

The average radius of Earth.

This has nothing to do with databases, but almost all of the classes which implement SQL_reader use this constant so it is useful to teclare it here.

5.24.3.6 SqlDataReader Twitter_tools.TwitterBase.reader

Iterates through the database and reads the table line by line.

The documentation for this class was generated from the following file:

- TwitterBase.cs

Index