

Grande Omega

1.0.0

Generated by Doxygen 1.8.14

Contents

1	README	2
2	Todo List	2
3	Namespace Index	2
3.1	Namespace List	2
4	Hierarchical Index	2
4.1	Class Hierarchy	2
5	Class Index	3
5.1	Class List	3
6	Namespace Documentation	4
6.1	DataTools Namespace Reference	4
6.2	DataTools.classification Namespace Reference	5
6.3	DataTools.clustering Namespace Reference	5
6.4	DataTools.correlation Namespace Reference	5
6.5	DataTools.regression Namespace Reference	5
6.6	DataTools.utils Namespace Reference	5
6.7	Highcharts Namespace Reference	6
6.8	Tree Namespace Reference	6

7	Class Documentation	6
7.1	Chart Class Reference	6
7.2	ClusterPoint Class Reference	7
7.2.1	Detailed Description	7
7.2.2	Constructor & Destructor Documentation	7
7.3	Correlation Class Reference	8
7.4	DataSet Class Reference	8
7.5	Dbscan Class Reference	9
7.5.1	Detailed Description	9
7.5.2	Constructor & Destructor Documentation	9
7.5.3	Member Function Documentation	10
7.6	Empty< T > Class Template Reference	11
7.7	GenericVector Class Reference	11
7.7.1	Detailed Description	12
7.8	Highchart Class Reference	12
7.9	ICchartsList Interface Reference	12
7.10	ITmplModel Interface Reference	12
7.11	ITree< T > Interface Template Reference	13
7.12	KdTree< T > Class Template Reference	13
7.13	Kmeans Class Reference	13
7.13.1	Detailed Description	14
7.13.2	Constructor & Destructor Documentation	14
7.13.3	Member Function Documentation	15
7.14	KnearestClassification Class Reference	16
7.14.1	Detailed Description	17
7.14.2	Constructor & Destructor Documentation	17
7.14.3	Member Function Documentation	17
7.15	LinearRegression Class Reference	18
7.16	MatrixUtils Class Reference	19
7.16.1	Detailed Description	19
7.17	PearsonCorrelation Class Reference	19
7.18	PolynomialRegression Class Reference	20
7.19	PriorityQue< T > Class Template Reference	20
7.20	Queltem< T > Class Template Reference	21
7.21	Replacer Class Reference	21
7.22	Resources Class Reference	21
7.22.1	Detailed Description	22
7.23	SpearmanCorrelation Class Reference	22
7.24	TmplEngine Class Reference	22
7.25	Utils Class Reference	23
7.26	Vector2 Class Reference	23
7.26.1	Detailed Description	23

Index	25
-----------------------	----

1 README

GrandeOmega Data Analyses

2 Todo List

Class [KnearestClassification](#)

In order de reduce the time complexity, a KD Tree could be implemented. Searching a KD Tree for nearest neighbours has a time complexity of $O(\log n)$

3 Namespace Index

3.1 Namespace List

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

DataTools	4
DataTools.classification	5
DataTools.clustering	5
DataTools.correlation	5
DataTools.regression	5
DataTools.utils	5
Highcharts	6
Tree	6

4 Hierarchical Index

4.1 Class Hierarchy

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

ClusterPoint	7
Correlation	8
PearsonCorrelation	19

SpearmanCorrelation	22
Dbscan	9
GenericVector	11
Highchart	12
IChartsList	12
ITmplModel	12
Chart	6
DataSeries	8
ITree< T >	13
Empty< T >	11
KdTree< T >	13
Kmeans	13
KnearestClassification	16
LinearRegression	18
MatrixUtils	19
PolynomialRegression	20
PriorityQue< T >	20
QueItem< T >	21
Replacer	21
Resources	21
TmplEngine	22
Utils	23
Vector2	23

5 Class Index

5.1 Class List

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

Chart	6
ClusterPoint	
Vector that is assigned to a cluster	7
Correlation	8

DataSeries	8
Dbscan	
Class featuring the DB Scan algorithm	9
Empty< T >	11
GenericVector	
N dimensional vector	11
Highchart	12
IChartsList	12
ITmplModel	12
ITree< T >	13
KdTree< T >	13
Kmeans	
Class featuring de KMeans algorithm	13
KnearestClassification	
Classification algorithm using K Nearest	16
LinearRegression	18
MatrixUtils	
Utilities for Math algebra	19
PearsonCorrelation	19
PolynomialRegression	20
PriorityQue< T >	20
Queltem< T >	21
Replacer	21
Resources	
Resource manager for the Assembly class	21
SpearmanCorrelation	22
TmplEngine	22
Utils	23
Vector2	
Two dimensional vector	23

6 Namespace Documentation

6.1 DataTools Namespace Reference

Namespaces

- 1

Classes

- class [GenericVector](#)
N dimensional vector.
- class [MatrixUtils](#)
Utilities for Math algebra.
- class [Vector2](#)
Two dimensional vector.

6.2 DataTools.classification Namespace Reference

Classes

- class [KnearestClassification](#)
Classification algorithm using K Nearest.

6.3 DataTools.clustering Namespace Reference

Classes

- class [ClusterPoint](#)
Vector that is assigned to a cluster.
- class [Dbscan](#)
Class featuring the DB Scan algorithm.
- class [Kmeans](#)
Class featuring de KMeans algorithm.

6.4 DataTools.correlation Namespace Reference

Classes

- class [Correlation](#)
- class [PearsonCorrelation](#)
- class [SpearmanCorrelation](#)

6.5 DataTools.regression Namespace Reference

Classes

- class [LinearRegression](#)
- class [PolynomialRegression](#)

6.6 DataTools.utils Namespace Reference

Classes

- class [PriorityQue](#)
- class [QuelItem](#)
- class [Utils](#)

6.7 Highcharts Namespace Reference

Classes

- class [Chart](#)
- class [DataSeries](#)
- class [Highchart](#)
- interface [IChartsList](#)
- interface [ITmplModel](#)
- class [Replacer](#)
- class [Resources](#)
Resource manager for the Assembly class.
- class [TmplEngine](#)

6.8 Tree Namespace Reference

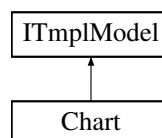
Classes

- class [Empty](#)
- interface [ITree](#)
- class [KdTree](#)

7 Class Documentation

7.1 Chart Class Reference

Inheritance diagram for Chart:



Public Member Functions

- **Chart** ([Highchart](#) chartType)
- void **SetDivId** (string divid)
- void **SetChartType** (string type)
- void **SetTitle** (string title)
- void **SetSubtitle** (string subtitle)
- void **SetXlabel** (string xlabel)
- void **SetYlabel** (string ylabel)
- void **SetXtooltip** (string xtooltip)
- void **SetYtooltip** (string ytooltip)
- string **CreateTemplate** ()
- void **AddDataSeries** ([DataSeries](#) data)
- List< Tuple< string, string > > **GetReplacers** ()

Properties

- string **Template** = "scatterplot" [get]

Private Attributes

- readonly Dictionary< [Replacer](#), string > **_replacers** = new Dictionary<[Replacer](#), string>()

7.2 ClusterPoint Class Reference

Vector that is assigned to a cluster.

Package Functions

- [ClusterPoint](#) ([GenericVector](#) vector, int cluster=-1)

Properties

- bool **Visited** [get, set]
- bool **Noise** [get, set]
- [GenericVector](#) **Vector** [get]
- int **Cluster** [get, set]

7.2.1 Detailed Description

Vector that is assigned to a cluster.

A ClusterPoint is a adapter that holds a GenericVector, but adds some clustering related properties to it.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 ClusterPoint()

```
ClusterPoint (
    GenericVector vector,
    int cluster = -1 ) [package]
```

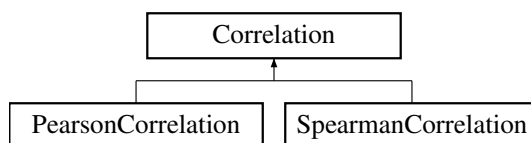
Constructor for ClusterPoint

Parameters

<i>vector</i>	Vector that needs to be hold
<i>cluster</i>	Cluster the vector belongs to. Default is -1

7.3 Correlation Class Reference

Inheritance diagram for Correlation:



Public Member Functions

- abstract double **GetCorrelationCoefficient** ()

Protected Member Functions

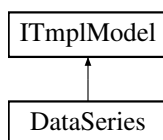
- **Correlation** (IEnumerable< [Vector2](#) > data)

Protected Attributes

- readonly IEnumerable< [Vector2](#) > **Data**

7.4 DataSeries Class Reference

Inheritance diagram for DataSeries:



Public Member Functions

- **DataSeries** ([Highchart](#) type, [IChartsList](#) data, string name, bool marker=true, bool tracking=false)
- void **SetType** (string type)
- void **SetName** (string name)
- void **SetMarker** (bool marker)
- void **SetMouseTracking** (bool tracking)
- void **SetData** ([IChartsList](#) data)
- string **ToTmplString** ()
- string **CreateTemplate** ()
- List< Tuple< string, string > > **GetReplacers** ()

Properties

- string **Template** = "dataseries" [get]

Private Attributes

- readonly Dictionary< [Replacer](#), string > **_replacers** = new Dictionary<[Replacer](#), string>()

7.5 Dbscan Class Reference

Class featuring the DB Scan algorithm.

Public Member Functions

- [Dbscan](#) (float eps, int minPoints, IEnumerable< [GenericVector](#) > data)

Properties

- Dictionary< int, IEnumerable< [GenericVector](#) > > **DataClusters** [get]

Private Member Functions

- void [Run](#) ()
- void [ExpandCluster](#) ([ClusterPoint](#) point, IEnumerable< [ClusterPoint](#) > neighbours, int cluster)
- List< [ClusterPoint](#) > [RegionQuery](#) ([GenericVector](#) point)

Private Attributes

- readonly float **_radius**
- readonly int **_minPoints**
- readonly List< [ClusterPoint](#) > **_dataSet**

7.5.1 Detailed Description

Class featuring the DB Scan algorithm.

The DB Scan algorithm is capable of clustering vectors of n dimensions. As input, it needs a radius within neighbours should be together, the minimum amount of point in a cluster and ofcourse the dataset, in this case, a `IEnumerable<GenericVector>`. As output it produces a Dictionary `DataClusters` with the key as cluster and the value the vectors belonging to the cluster.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 Dbscan()

```
Dbscan (
    float eps,
    int minPoints,
    IEnumerable< GenericVector > data )
```

Constructor for DB Scan

Parameters

<i>eps</i>	Radius within neighbours should be
<i>minPoints</i>	Min amount of points in a cluster
<i>data</i>	Dataset to cluster

7.5.3 Member Function Documentation**7.5.3.1 ExpandCluster()**

```
void ExpandCluster (
    ClusterPoint point,
    IEnumerable< ClusterPoint > neighbours,
    int cluster ) [private]
```

Expand the cluster from point, by visiting all neighbours of his neighbours.

Parameters

<i>point</i>	Point to expand
<i>neighbours</i>	Neighbours of point
<i>cluster</i>	Current cluster to expand

7.5.3.2 RegionQuery()

```
List<ClusterPoint> RegionQuery (
    GenericVector point ) [private]
```

Compute all points withing Point's radius-neighbourhood

Parameters

<i>point</i>	Point to compute neighbours for
--------------	---------------------------------

Returns

All points within point's radius, including point

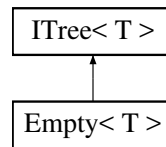
7.5.3.3 Run()

```
void Run ( ) [private]
```

Run the DB Scan algorithm on the provided dataset

7.6 Empty< T > Class Template Reference

Inheritance diagram for Empty< T >:



Public Attributes

- bool **IsEmpty** => false

Properties

- int **Dimension** [get]
- [ITree< T >](#) **Left** [get]
- [ITree< T >](#) **Right** [get]
- T **Value** [get]

7.7 GenericVector Class Reference

N dimensional vector.

Public Member Functions

- **GenericVector** (int size)
- **GenericVector** (params double[] args)
- double [] **ToArray** ()
- override string **ToString** ()
- [Vector2](#) **ToVector2** (int indexOne=0, int indexTwo=1)

Static Public Member Functions

- static [GenericVector](#) **Sum** ([GenericVector](#) a, [GenericVector](#) b)
- static [GenericVector](#) **Devide** ([GenericVector](#) a, int devider)
- static double **Distance** ([GenericVector](#) a, [GenericVector](#) b)

Public Attributes

- int **Size** => _points.Length
- double **BiggestPoint** => _points.Max()
- double **this[int x]** => _points[x]

Private Attributes

- readonly double [] **_points**

7.7.1 Detailed Description

N dimensional vector.

This class is a vector with N dimensions. The amount of dimensions is created upon initialization and can not be changed afterwards. It implements some basic vector algebra like summation, multiplication and deviding.

7.8 Highchart Class Reference

Static Public Attributes

- static readonly [Highchart](#) **Scatterplot** = new [Highchart](#)("scatter")
- static readonly [Highchart](#) **Regression** = new [Highchart](#)("line")

Properties

- string **Value** [get]

Private Member Functions

- **Highchart** (string value)

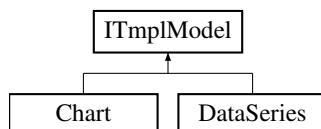
7.9 IChartsList Interface Reference

Public Member Functions

- string **ToChartsList** ()

7.10 ITmplModel Interface Reference

Inheritance diagram for ITmplModel:



Public Member Functions

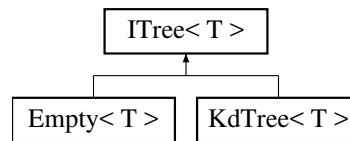
- string **CreateTemplate** ()
- List< Tuple< string, string > > **GetReplacers** ()

Properties

- string **Template** [get]

7.11 ITree< T > Interface Template Reference

Inheritance diagram for ITree< T >:

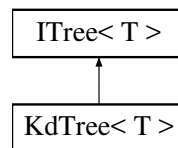


Properties

- int **Dimension** [get]
- ITree< T > **Left** [get]
- ITree< T > **Right** [get]
- T **Value** [get]
- bool **IsEmpty** [get]

7.12 KdTree< T > Class Template Reference

Inheritance diagram for KdTree< T >:



Public Attributes

- bool **IsEmpty** => true

Protected Member Functions

- **KdTree** (ITree< T > left, T value, ITree< T > right, int dimension)

Properties

- int **Dimension** [get]
- ITree< T > **Left** [get]
- ITree< T > **Right** [get]
- T **Value** [get]

7.13 Kmeans Class Reference

Class featuring de KMeans algorithm.

Public Member Functions

- [Kmeans](#) (int k, int iterations, IEnumerable< [GenericVector](#) > dataSet)

Properties

- Dictionary< int, IEnumerable< [GenericVector](#) > > **DataClusters** [get]

Private Member Functions

- void [Run](#) ()
- void [RecalculateClusters](#) ()
- void [RecalculateCentroids](#) ()
- int [GetNearestCluster](#) ([GenericVector](#) v)
- Dictionary< int, [GenericVector](#) > [GenerateRandomCentroids](#) (int kAmount)
- [GenericVector](#) [GetRandomVector](#) ()

Static Private Member Functions

- static bool [IsChangedCluster](#) (IEnumerable< int > a, IReadOnlyList< int > b)

Private Attributes

- readonly Random **_random** = new Random()
- readonly int **_clusters**
- readonly int **_iterations**
- readonly List< [ClusterPoint](#) > **_dataSet**
- Dictionary< int, [GenericVector](#) > **_centroids**

7.13.1 Detailed Description

Class featuring de KMeans algorithm.

KMeans is a clustering algorithm. As input it needs *k* (the amount of clusters), the max amount of iterations and the dataset to cluster. It outputs the clustered values.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 Kmeans()

```
Kmeans (
    int k,
    int iterations,
    IEnumerable< GenericVector > dataSet )
```

Constructor for Kmeans

Parameters

<i>k</i>	Amount of clusters
<i>iterations</i>	Max amount of iterations
<i>dataSet</i>	The dataset to cluster

7.13.3 Member Function Documentation

7.13.3.1 GenerateRandomCentroids()

```
Dictionary<int, GenericVector> GenerateRandomCentroids (
    int kAmount ) [private]
```

Generate random centroids to start with

Parameters

<i>kAmount</i>	Amount of centroid to generate
----------------	--------------------------------

Returns

Centroids

7.13.3.2 GetNearestCluster()

```
int GetNearestCluster (
    GenericVector v ) [private]
```

Get the cluster wherefrom the centroid is nearest to the point

Parameters

<i>vector</i>	Vector to get the nearest cluster for
---------------	---------------------------------------

Returns

Nearest cluster

7.13.3.3 GetRandomVector()

```
GenericVector GetRandomVector ( ) [private]
```

Pick random vector from the dataset

7.13.3.4 IsChangedCluster()

```
static bool IsChangedCluster (
    IEnumerable< int > a,
    IReadOnlyList< int > b ) [static], [private]
```

Check if the clusters are changed by comparing the two provided collections

Parameters

<i>a</i>	Collection A
<i>b</i>	Collection B

Returns

/c True if the cluster is changed otherwise /c false

7.13.3.5 RecalculateCentroids()

```
void RecalculateCentroids ( ) [private]
```

Recompute the new centroids of the clusters

7.13.3.6 RecalculateClusters()

```
void RecalculateClusters ( ) [private]
```

Recalculate the cluster foreach point regarding the current centroid.

7.13.3.7 Run()

```
void Run ( ) [private]
```

Run the algorithm on the provided dataset

7.14 KnearestClassification Class Reference

Classification algorithm using K Nearest.

Public Member Functions

- [KnearestClassification](#) (Dictionary< int, IEnumerable< [GenericVector](#) >> trainingData, int k)
- int [ClassifyPoint](#) ([GenericVector](#) point)

Static Private Member Functions

- static int [GetBiggestCluster](#) ([PriorityQue](#)< [ClusterPoint](#) > nearestPoints)

Private Attributes

- readonly Dictionary< int, IEnumerable< [GenericVector](#) > > **_trainingData**
- readonly int **_k**

7.14.1 Detailed Description

Classification algorithm using K Nearest.

K Nearest is a classification algorithm that uses a training set in order to classify new points. To do so, K Nearest computes the distance from the new point to all the points in the training set. With a time complexity of $O(m*n)$ where n is the length of the training set and m is the length of the set of new points, this is a CPU heavy operation.

Todo In order to reduce the time complexity, a KD Tree could be implemented. Searching a KD Tree for nearest neighbours has a time complexity of $O(\log n)$

7.14.2 Constructor & Destructor Documentation

7.14.2.1 KnearestClassification()

```
KnearestClassification (
    Dictionary< int, IEnumerable< GenericVector > > trainingData,
    int k )
```

Constructor for KnearestClassification

Parameters

<i>trainingData</i>	Set of pre clustered data
<i>k</i>	Amount of neighbours a new point is verified with

7.14.3 Member Function Documentation

7.14.3.1 ClassifyPoint()

```
int ClassifyPoint (
    GenericVector point )
```

Classify a point

Parameters

<i>point</i>	Point to classify
--------------	-------------------

Returns

int Cluster the point belongs to

7.14.3.2 GetBiggestCluster()

```
static int GetBiggestCluster (
    PriorityQueue< ClusterPoint > nearestPoints ) [static], [private]
```

Get the biggest cluster a que

Parameters

<i>nearestPoints</i>	Points nearest to the classified point
----------------------	--

Returns

int Biggest cluster

7.15 LinearRegression Class Reference**Public Member Functions**

- **LinearRegression** (IEnumerable< [Vector2](#) > data)
- IEnumerable< [GenericVector](#) > **GetLinearRegressionLine** ()

Public Attributes

- double **Slope** => GetSlope()
- double **YIntercept** => MeanY - (Slope * MeanX)

Properties

- double **MeanX** [get]
- double **MeanY** [get]

Private Member Functions

- double **GetSlope** ()

Static Private Member Functions

- static double **Mean** (IEnumerable< double > sample)

Private Attributes

- readonly IEnumerable< [Vector2](#) > **_data**

7.16 MatrixUtils Class Reference

Utilities for Math algebra.

Static Public Member Functions

- static double [][] **MatrixInverse** (double[][] matrix)
- static double [][] **MatrixCreate** (int rows, int cols)

Static Private Member Functions

- static double [][] **MatrixDecompose** (double[][] matrix, out int[] perm, out int toggle)
- static double [] **HelperSolve** (double[][] luMatrix, double[] b)
- static double [][] **MatrixDuplicate** (double[][] matrix)

7.16.1 Detailed Description

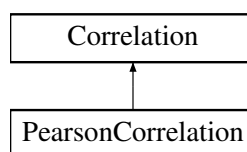
Utilities for Math algebra.

NOTE the code used in this class is written by James D. McCaffey. All credits goes to him. More info (<https://jamesmccaffrey.wordpress.com/2015/03/06/inverting-a-matrix-using-c/>)

In this class are included several Matrix functions, that supports the matrix inverse function.

7.17 PearsonCorrelation Class Reference

Inheritance diagram for PearsonCorrelation:



Public Member Functions

- **PearsonCorrelation** (IEnumerable< [Vector2](#) > data)
- override double **GetCorrelationCoefficient** ()

Private Member Functions

- double **PearsonCoefficient** ()

Additional Inherited Members

7.18 PolynomialRegression Class Reference

Public Member Functions

- **PolynomialRegression** (IEnumerable< [Vector2](#) > data, int polynomialDegree)
- double **PredictPoint** (double x)
- [GenericVector](#) [] **GetPolynomialPoints** ()

Properties

- int **PolynomialDegree** [get]

Private Member Functions

- double **ComputePolynomialPointY** (double x)
- void **ComputerMatrixCoefficients** ()
- void **ComputeMatrixYvalues** (int polynomialDegree)
- void **ComputeMatrixPoints** (int polynomialDegree)
- double **ComputeSigmaPowerXY** (int degree)
- double **ComputeSigmaPowerX** (int degree)

Private Attributes

- readonly double [][] **_matrixPoints**
- readonly double [] **_matrixCoefficients**
- readonly double [] **_matrixYvalues**
- readonly [Vector2](#) [] **_data**
- readonly Dictionary< int, double > **_sigmaDegreeCache** = new Dictionary<int, double>()

7.19 PriorityQueue< T > Class Template Reference

Public Member Functions

- **PriorityQue** (IEnumerable< Tuple< double, T >> entry)
- void **Insert** (double priority, T quelItem)
- [QuelItem](#)< T > **Peek** ()
- [QuelItem](#)< T > **Pop** ()
- List< Tuple< double, T > > **ToList** ()

Public Attributes

- int **Count** => _entries.Count
- bool **IsEmpty** => Count == 0

Private Attributes

- List< Tuple< double, T > > **_entries**

7.20 `Queltem< T >` Class Template Reference

Public Member Functions

- **Queltem** (Tuple< double, T > queltem)

Public Attributes

- double **Priority** => _queltem.Item1
- T **Item** => _queltem.Item2

Private Attributes

- readonly Tuple< double, T > **_queltem**

7.21 `Replacer` Class Reference

Static Public Attributes

- static readonly `Replacer` **DivId** = new `Replacer`("divid")
- static readonly `Replacer` **Chart** = new `Replacer`("chart")
- static readonly `Replacer` **Title** = new `Replacer`("title")
- static readonly `Replacer` **Subtitle** = new `Replacer`("subtitle")
- static readonly `Replacer` **Xlabel** = new `Replacer`("xlabel")
- static readonly `Replacer` **Ylabel** = new `Replacer`("ylabel")
- static readonly `Replacer` **Xtooltip** = new `Replacer`("xtooltip")
- static readonly `Replacer` **Ytooltip** = new `Replacer`("ytooltip")
- static readonly `Replacer` **Data** = new `Replacer`("data")
- static readonly `Replacer` **Type** = new `Replacer`("type")
- static readonly `Replacer` **Name** = new `Replacer`("name")
- static readonly `Replacer` **Marker** = new `Replacer`("marker")
- static readonly `Replacer` **Mousetracking** = new `Replacer`("mousetracking")

Properties

- string **Value** [get]

Private Member Functions

- **Replacer** (string value)

7.22 `Resources` Class Reference

Resource manager for the Assembly class.

Static Public Member Functions

- static Stream **LoadFile** (string resource)
- static string [] **GetResourcesNames** ()

Static Private Attributes

- static readonly Assembly **Assembly** = typeof(Highchart).GetTypeInfo().Assembly

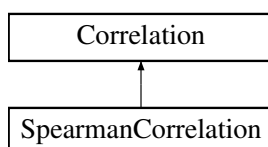
7.22.1 Detailed Description

Resource manager for the Assembly class.

This class manages the files that are copied in the assembly upon compilation.

7.23 SpearmanCorrelation Class Reference

Inheritance diagram for SpearmanCorrelation:



Public Member Functions

- **SpearmanCorrelation** (IEnumerable< Vector2 > data)
- override double **GetCorrelationCoefficient** ()

Private Member Functions

- double **SpearmanCofficient** ()

Static Private Member Functions

- static double **CorrectionFactor** (IEnumerable< double > rankedData)
- static Dictionary< double, double > **ComputeRanking** (IEnumerable< double > values)

Additional Inherited Members

7.24 TmplEngine Class Reference

Static Public Member Functions

- static string **CreateTemplate** (ITmplModel model)

Private Attributes

- const string **Prefix** = "Highcharts.Files."
- const string **FileExtension** = ".tmpl"
- const string **LeftToken** = "<%"
- const string **RightToken** = "%>"

7.25 Utils Class Reference

Static Public Member Functions

- static void **Times** (this int count, System.Action action)

7.26 Vector2 Class Reference

Two dimensional vector.

Public Member Functions

- **Vector2** (double x, double y)

Properties

- double **X** [get, set]
- double **Y** [get, set]

7.26.1 Detailed Description

Two dimensional vector.

Two dimensional vector having a X and Y dimension.

Index

Chart, [6](#)
ClassifyPoint
 DataTools::classification::KnearestClassification, [17](#)
ClusterPoint, [7](#)
 DataTools::clustering::ClusterPoint, [7](#)
Correlation, [8](#)

DataSeries, [8](#)
DataTools, [4](#)
DataTools.classification, [5](#)
DataTools.clustering, [5](#)
DataTools.correlation, [5](#)
DataTools.regression, [5](#)
DataTools.utils, [5](#)
DataTools::classification::KnearestClassification
 ClassifyPoint, [17](#)
 GetBiggestCluster, [18](#)
 KnearestClassification, [17](#)
DataTools::clustering::ClusterPoint
 ClusterPoint, [7](#)
DataTools::clustering::Dbscan
 Dbscan, [9](#)
 ExpandCluster, [10](#)
 RegionQuery, [10](#)
 Run, [10](#)
DataTools::clustering::Kmeans
 GenerateRandomCentroids, [15](#)
 GetNearestCluster, [15](#)
 GetRandomVector, [15](#)
 IsChangedCluster, [15](#)
 Kmeans, [14](#)
 RecalculateCentroids, [16](#)
 RecalculateClusters, [16](#)
 Run, [16](#)
Dbscan, [9](#)
 DataTools::clustering::Dbscan, [9](#)

Empty< T >, [11](#)
ExpandCluster
 DataTools::clustering::Dbscan, [10](#)

GenerateRandomCentroids
 DataTools::clustering::Kmeans, [15](#)
GenericVector, [11](#)
GetBiggestCluster
 DataTools::classification::KnearestClassification, [18](#)
GetNearestCluster
 DataTools::clustering::Kmeans, [15](#)
GetRandomVector
 DataTools::clustering::Kmeans, [15](#)

Highchart, [12](#)
Highcharts, [6](#)
IChartsList, [12](#)
ITmplModel, [12](#)
ITree< T >, [13](#)
IsChangedCluster
 DataTools::clustering::Kmeans, [15](#)

KdTree< T >, [13](#)
Kmeans, [13](#)
 DataTools::clustering::Kmeans, [14](#)
KnearestClassification, [16](#)
 DataTools::classification::KnearestClassification, [17](#)

LinearRegression, [18](#)

MatrixUtils, [19](#)

PearsonCorrelation, [19](#)
PolynomialRegression, [20](#)
PriorityQue< T >, [20](#)

Queltem< T >, [21](#)

RecalculateCentroids
 DataTools::clustering::Kmeans, [16](#)
RecalculateClusters
 DataTools::clustering::Kmeans, [16](#)
RegionQuery
 DataTools::clustering::Dbscan, [10](#)
Replacer, [21](#)
Resources, [21](#)
Run
 DataTools::clustering::Dbscan, [10](#)
 DataTools::clustering::Kmeans, [16](#)

SpearmanCorrelation, [22](#)

TmplEngine, [22](#)
Tree, [6](#)

Utils, [23](#)

Vector2, [23](#)