|  |
| --- |
|  |
| SeesharpTools.JXI.  Mathematics |
| user manual |
|  |
|  |
|  |

|  |
| --- |
|  |

**文档版本**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订说明 | 修订人 |
| 2017/09/20 | 1.0.0 | SeesharpTool.JXI.Mathematics | 刘玉辉 |

**目 录**

[1 CurveFitting 1](#_Toc31122)

[1.1 CurveFitting概述 1](#_Toc28391)

[1.2 LinearFit 1](#_Toc14776)

[1.3 ExponentialFit 2](#_Toc23049)

[1.4 PolynomialFit 3](#_Toc13114)

[2 ProbabilityStatistics 5](#_Toc1005)

[2.1 ProbabilityStatistics概述 5](#_Toc7407)

[2.2 Mean 5](#_Toc4296)

[2.3 GeometricMean 6](#_Toc6154)

[2.4 HarmonicMean 6](#_Toc15248)

[2.5 TrimmedMean 7](#_Toc16401)

[2.6 Median 8](#_Toc21639)

[2.7 Mode 8](#_Toc9126)

[2.8 RootMeanSquare 9](#_Toc31090)

[2.9 Variance 10](#_Toc13444)

[2.10 StandardDeviation 10](#_Toc31774)

[2.11 MeanSquaredError 11](#_Toc28754)

[2.12 Histogram 12](#_Toc21082)

## 

CurveFitting

CurveFitting概述

CurveFitting类包括线性拟合、指数拟合、多项式拟合。

注：若报错并提示“试图加载格式不正确的程序”，请将程序生成的目标平台改成x86。

LinearFit

函数描述

线性拟合的拟合原型如下：



函数接口

线性拟合类的多态方法：

void LinearFit(double[] x, double[] y, double interestX, out double fittedY)

void LinearFit(double[] x, double[] y, double interestX, out double fittedY, out double slope, out double intercept)

void LinearFit(double[] x, double[] y, double[] interestX, ref double[] fittedY)

void LinearFit(double[] x, double[] y, double[] interestX, ref double[] fittedY, out double slope, out double intercept)

其中，x为输入数据的横坐标，y为输入数据的纵坐标，interestX为用户感兴趣的x值，fittedY为interestX对应的拟合值，slope为斜率，intercept为截距。

**注意：x，y的数组长度必须一致，interestX，fittedY的数组长度必须一致!**

范例

double[] x = { 1, 2, 3, 4, 5 };

double[] y = { 3, 5, 7, 9, 11 };

double slop = 0;

double interception = 0;

double[] data = new double[5];

EasyCurveFitting.LinearFit(x, y, x, ref data, out slop, out interception);

结果：

data={3, 5, 7, 9, 11 }；

slop=2;

Interception=1;

ExponentialFit

函数描述

指数拟合的拟合原型如下：



函数接口

指数拟合类的多态方法：

void ExponentialFit(double[] x, double[] y, double interestX, out double fittedY)

void ExponentialFit(double[] x, double[] y, double interestX, out double fittedY, out double amplitude, out double damping, out double offset)

void ExponentialFit(double[] x, double[] y, double[] interestX, ref double[] fittedY)

void ExponentialFit(double[] x, double[] y, double[] interestX, ref double[] fittedY, out double amplitude, out double damping, out double offset)

其中，x为输入数据的横坐标，y为输入数据的纵坐标，interestX为用户感兴趣的x值，fittedY为interestX对应的拟合值，amplitude为幅值，damping为衰减，offset为偏移量（目前为0，即没有偏移量）。

**注意：x，y的数组长度必须一致，interestX，fittedY的数组长度必须一致!**

范例

double[] x = { 1, 2, 3, 4, 5 };

double[] y = new double[5];

for (int i = 0; i < 5; i++)

{

y[i] = System.Math.Exp(1.7 \* x[i]) \* (2.6);

}

double amplitude = 0;

double damping = 0;

double offset = 0;

double[] data = new double[5];

EasyCurveFitting.ExponentialFit(x, y, x, ref data, out amplitude, out damping, out offset);

结果：

data={14.232263218490726, 77.906660123232257, 426.45695897974457, 2334.4029582910866, 12778.398984777754 }；

amplitude=2.600000000000001;

damping=1.7;

Offset=0；

PolynomialFit

函数描述

多项式拟合的拟合原型如下：



函数接口

多项式拟合类的多态方法：

void PolynomialFit(double[] x, double[] y, int order, double interestX, out double fittedY)

void PolynomialFit(double[] x, double[] y, int order, double interestX, out double fittedY, ref double[] coefficients)

void PolynomialFit(double[] x, double[] y, int order, double[] interestX, ref double[] fittedY)

void PolynomialFit(double[] x, double[] y, int order, double[] interestX, ref double[] fittedY, ref double[] coefficients)

其中，x为输入数据的横坐标，y为输入数据的纵坐标，order为拟合阶数，interestX为用户感兴趣的x值，fittedY为interestX对应的拟合值，coefficients为多项式拟合系数。

**注意：x，y的数组长度必须一致，interestX，fittedY的数组长度必须一致!**

范例

double[] x = { 1, 2, 3, 4, 5 };

double[] y = new double[5];

for (int i = 0; i < 5; i++)

{

y[i] = 10 + -10 \* x[i] -5 \* System.Math.Pow(x[i], 2) -4 \* System.Math.Pow(x[i], 3);

}

double[] coe = new double[5];

double[] data = new double[5];

int order = 3;

EasyCurveFitting.PolynomialFit(x, y, order, x, ref data, ref coe);

结果：

data={-9.0000000000094, -61.999999999993193, -172.99999999999784, -5.00000000002835, -664.99999999999807}；

coe={9.99999999993561，-9.9999999999196483，-5.00000000002835，-3.9999999999970135}；

ProbabilityStatistics

ProbabilityStatistics概述

ProbabilityStatistics概率统计类包括均值、中位数、众数、方差、均方误差、标准差、均方根的计算及一维数组的直方图统计。

Mean

函数描述

算数平均值（arithmetic mean）的定义为：



函数接口

**double Mean(double[] nums)**

其中，nums为输入数据。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4, 5 };

double mean = ProbabilityStatistics.Mean(nums );

结果：

mean=3

GeometricMean

函数描述

几何平均值（geometric mean）是指n个观察值连乘积的n次方根：



函数接口

**double GeometricMean(double[] nums)**

其中，nums为输入数据。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4, 5 };

double mean = ProbabilityStatistics.GeometricMean(nums);

结果：

mean=2.6051710846973521

HarmonicMean

函数描述

调和平均值（harmonic mean）又称倒数平均数，是总体各统计变量倒数的算术平均数的倒数：



函数接口

**double HarmonicMean(double[] nums)**

其中，nums为输入数据。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4, 5 };

double mean = ProbabilityStatistics.HarmonicMean(nums, meanType);

结果：

mean=2.18978102189781

TrimmedMean

函数描述

切尾均值（trimmed mean），在排序去掉两端极端值后按照剩余数据计算算数平均值。

函数接口

**double TrimmedMean(double[] nums, double trimmedPercent = 0)**

其中，nums为输入数据，trimmedPercent为切尾百分比。

**注意：输入序列不能为空；切尾百分比为0~100之间的数**

范例

double[] nums = { 1, 2, 3, 4, 5 };

double mean = ProbabilityStatistics.TrimmedMean(nums,40);

结果：

mean=3

Median

函数描述

中位数（Median）又称中值，为输入序列大小排序后的中间值。先对输入序列X排序，如输入序列包含奇数个元素，中值为中间元素；如输入序列包含偶数个元素，中值为中间两个元素的平均数。

函数接口

**double Median(double[] nums)**

nums为输入数据；

函数返回值为nums的中位数。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4 };

double median = ProbabilityStatistics.Median(nums);

结果：

median= 2.5

Mode

函数描述

众数是样本观测值在频数分布表中频数最多的那一组的组中值，即输入序列中出现频次最高的一个或多个数

函数接口

**double[] Mode(double[] nums)**

nums为输入数据；

函数返回值为nums的众数。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4, 3 };

double[] mode = ProbabilityStatistics.Mode(nums);

结果：

mode ={3 }

RootMeanSquare

函数描述

均方根（Root Mean Square, RMS）有时也称方均根，为输入序列的平方的算数平均值的算数平方根，数学表达式为：



函数接口

**double RootMeanSquare(double[] nums)**

nums为输入数据；

函数返回值为nums的均方根。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4, 5 };

double rms = ProbabilityStatistics.RootMeanSquare(nums);

结果：

rms= 3.3166247903554

Variance

函数描述

方差（variance)是输入序列中各个数据与算数平均值之差的平方的算数平均值，数学表达式为：



函数接口

**double Variance(double[] nums)**

nums为输入数据；

函数返回值为nums的均方根。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4, 5 };

double variance = ProbabilityStatistics.Variance(nums);

结果：

variance=2

StandardDeviation

函数描述

标准差（Standard Deviation）即方差的算数平方根，数学表达式为：



函数接口

**double StandardDeviation(double[] nums)**

nums为输入数据；

函数返回值为nums的标准差。

**注意：输入序列不能为空**

范例

double[] nums = { 1, 2, 3, 4, 5};

double standardDeviation = ProbabilityStatistics.StandardDeviation(nums);

结果：

standardDeviation=1.4142...

MeanSquaredError

函数描述

两组输入序列之间的均方误差（Mean Squared Error, MSE）定义为



函数接口

**double MeanSquaredError(double[] x, double[] y)**

x、y为输入序列；

函数返回值为x与y的均方误差。

**注意：两输入序列不能为空且必须等长**

范例

double[] x = { 1, 2, 3, 4, 5 };

double[] y = { 1.1, 1.9, 3.2, 4.1, 5 };

double mse = ProbabilityStatistics.MeanSquaredError(x, y);

结果：

mse=0.014

Histogram

函数描述

分析输入序列落入指定划分的各区间的频率计数。

直方图分析频率区间宽度的为：



其中numberOfIntervals是要求的区间数。区间的中点由下列等式确定



函数接口

**void Histogram(double[] nums, ref int[] histogram, ref double[] intervals, IntervalType intervalType = IntervalType.LeftClosed)**

nums为输入序列；

直方图分析区间数由histogram和intervals长度决定，两者长度必须相等；

intervalType为区间开闭方式。若选择为**IntervalType.LeftClosed**则统计过程区间按照左闭右开方式处理（最后一个区间作为闭区间处理）；若选择为**IntervalType.RightClosed**则统计过程区间按照左开右闭方式处理（第一个区间作为闭区间处理）；

函数通过out关键字将直方图分析结果histgram和区间中点序列inervals返回。

范例

double[] data = new double[] { 0, 1, 2, 3, 4, 1, 2, 3, 2 };

int nbrOfIntervals = 4;

int[] histogram = new int[nbrOfIntervals];

double[] intervals = new double[nbrOfIntervals];

IntervalType intervalType = IntervalType.LeftClosed;

ProbabilityStatistics.Hstogram(data, ref histogram, ref intervals, intervalType);

结果

histogram 为{1, 2, 3, 3}

intervals为{0.5, 1.5, 2.5, 3.5}