NAME

StatisticsUtil

SYNOPSIS

use StatisticsUtil;

use Statistics qw(:all);

DESCRIPTION

StatisticsUtil module provides the following functions:

Average, AverageDeviation, Correlation, Covariance, Euclidean, Factorial, FactorialDivison, Frequency, GeometricMean, HarmonicMean, KLargest, KSmallest, Kurtosis, Maximum, Mean, Median, Minimum, Mode, PearsonCorrelation, Product, RSquare, Range, Skewness, StandardDeviation, StandardDeviationN, StandardError, StandardScores, StandardScoresN, Standardize, Sum, SumOfSquares, TrimMean, Variance, VarianceN

METHODS

Average

```
$Value = Average(\@DataArray);
```

Computes the mean of an array of numbers: SUM(x[i]) / n

AverageDeviation

```
$Value = AverageDeviation(\@DataArray);
```

Computes the average of the absolute deviation of an array of numbers: SUM(ABS(x[i] - Xmean)) / n

Correlation

```
$Value = Correlation(\@XDataArray, \@YDataArray);
```

Computes the Pearson correlation coefficient between two arrays of numbers: SUM((x[i] - Xmean)(y[i] - Ymean)) / SQRT(SUM((x[i] - Xmean)^2)(SUM((y[i] - Ymean)^2)))

Euclidean

```
$Return = Euclidean(\@DataArray);
```

Computes the euclidean distance of an array of numbers: SQRT(SUM(x[i] ** 2))

Covariance

```
$Value = Covariance(\@XDataArray, \@YDataArray);
```

Computes the covariance between two arrays of numbers: SUM((x[i] - Xmean) (y[i] - Ymean)) / n

Factorial

```
$Value = Factorial($Num);
```

Computes the factorial of a positive integer.

FactorialDivison

```
$Value = FactorialDivision($Numerator, $Denominator);
```

Compute the factorial divison of two positive integers.

Frequency

```
%FrequencyValues = Frequency(\@DataArray, [$NumOfBins]);
%FrequencyValues = Frequency(\@DataArray, [\@BinRange]);
```

A hash array is returned with keys and values representing range and frequency values, respectively. The frequency value for a specific key corresponds to all the values which are greater than the previous key and less than or equal to the current key. A key value representing maximum value is added for generating frequency distribution for specific number of bins, and whenever the maximum array value is greater than the maximum specified in bin range, it is also added to bin range.

GeometricMean

```
$Value = GeometricMean(\@DataArray);
```

Computes the geometric mean of an array of numbers: NthROOT(PRODUCT(x[i]))

HarmonicMean

```
$Value = HarmonicMean(\@DataArray);
```

Computes the harmonic mean of an array of numbers: 1 / (SUM(1/x[i]) / n)

KLargest

```
$Value = KLargest(\@DataArray, $KthNumber);
```

Returns the k-largest value from an array of numbers.

KSmallest

```
$Value = KSmallest(\@DataArray, $KthNumber);
```

Returns the k-smallest value from an array of numbers.

Kurtosis

```
$Value = Kurtosis(\@DataArray);
```

Computes the kurtosis of an array of numbers: [{n(n + 1)/(n - 1)(n - 2)(n - 3)} SUM{ ((x[i] - Xmean)/STDDEV)^4 }] - {3((n - 1)^2)}/{(n - 2)(n-3)}

Maximum

```
$Value = Maximum(\@DataArray);
```

Returns the largest value from an array of numbers.

Minimum

```
$Value = Minimum(\@DataArray);
```

Returns the smallest value from an array of numbers.

Mean

```
$Value = Mean(\@DataArray);
```

Computes the mean of an array of numbers: SUM(x[i]) / n

Median

```
$Value = Median(\@DataArray);
```

Computes the median value of an array of numbers. For an even number array, it's the average of two middle values.

For even values of n: Xsorted[(n - 1)/2 + 1] For odd values of n: (Xsorted[n/2] + Xsorted[n/2 + 1])/2

Mode

```
$Value = Mode(\@DataArray);
```

Returns the most frequently occuring value in an array of numbers.

PearsonCorrelation

```
$Value = Correlation(\@XDataArray, \@YDataArray);
```

Computes the Pearson correlation coefficient between two arrays of numbers: SUM((x[i] - Xmean)(y[i] - Ymean)) / SQRT(SUM((x[i] - Xmean)^2)(SUM((y[i] - Ymean)^2)))

Product

```
$Value = Product(\@DataArray);
```

Compute the product of an array of numbers.

Range

```
($Smallest, $Largest) = Range(\@DataArray);
```

Return the smallest and largest values from an array of numbers.

RSquare

```
$Value = RSquare(\@XDataArray, \@YDataArray);
```

Computes square of the Pearson correlation coefficient between two arrays of numbers.

Skewness

```
$Value = Skewness(\@DataArray);
```

Computes the skewness of an array of numbers: {n/(n - 1)(n - 2)} SUM{ ((x[i] - Xmean)/STDDEV)^3 }

StandardDeviation

```
$Value = StandardDeviation(\@DataArray);
```

Computes the standard deviation of an array of numbers. SQRT (SUM((x[i] - mean)^2) / (n - 1))

StandardDeviationN

```
$Value = StandardDeviationN(\@DataArray);
```

Computes the standard deviation of an array of numbers representing entire population: SQRT (SUM((x[i] - mean)^2) / n)

StandardError

```
$Value = StandardError($StandardDeviation, $Count);
```

Computes the standard error using standard deviation and sample size.

Standardize

```
$Value = Standardize($Value, $Mean, $StandardDeviation);
```

Standardizes the value using mean and standard deviation.

StandardScores

```
@Values = StandardScores(\@DataArray);
```

Computes the standard deviation above the mean for an array of numbers: (x[i] - mean) / (n - 1)

StandardScoresN

```
@Values = StandardScoresN(\@DataArray);
```

Computes the standard deviation above the mean for an array of numbers representing entire population: (x[i] - mean) / n

Sum

```
$Value = Sum(\@DataArray);
```

Compute the sum of an array of numbers.

SumOfSquares

```
$Value = SumOfSquares(\@DataArray);
```

Computes the sum of an array of numbers.

TrimMean

```
$Value = TrimMean(\@DataArray, $FractionToExclude));
```

Computes the mean of an array of numbers by excluding a fraction of numbers from the top and bottom of the data set.

Variance

```
$Value = Variance(\@DataArray);
```

Computes the variance of an array of numbers: SUM((x[i] - Xmean)^2 / (n - 1))

VarianceN

```
$Value = Variance(\@DataArray);
```

Compute the variance of an array of numbers representing entire population: SUM((x[i] - Xmean)^2 / n)

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

Constants.pm, ConversionsUtil.pm, MathUtil.pm

COPYRIGHT

Copyright (C) 2004-2012 Manish Sud. All rights reserved.

This file is part of MayaChemTools.

MayaChemTools is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.