Package ‘LearningRlab’
August 31, 2023

Version 2.4
Date 2023-8-30
Title Statistical Learning Functions
Author Carlos Javier Hellin Asensio [aut, cre],
Jose Manuel Gomez Caceres [aut],
Dennis Monheimius [aut],
Eduardo Benito [aut],
Juan Jose Cuadrado [aut],
Universidad de Alcala de Henares [aut]
Maintainer Carlos Javier Hellin Asensio <carlos.hellin@uah.es>
Depends magick, crayon
Suggests knitr, rmarkdown
Description Aids in learning statistical functions incorporating the result of calculus done with each function and how they are obtained, that is, which equation and variables are used. Also for all these equations and their related variables detailed explanations and interactive exercises are also included. All these characteristics allow to the package user to improve the learning of statistics basics by means of their use.
License Unlimited
VignetteBuilder knitr
NeedsCompilation yes
Type Package
Repository CRAN
Date/Publication 2023-08-30 22:10:03 UTC
RoxygenNote 7.1.1
Encoding UTF-8

R topics documented:

averageDeviation_ ................................................................. 3
binomial_ ............................................................................... 4
chisquared_ ......................................................................... 6
topics documented:

covariance_ ................................................................. 7
cv ................................................................. 8
drawVector ................................................................. 9
explain.absolute_acum_frecuency ........................................ 10
explain.absolute_frecuency ........................................ 11
explain.averageDeviation ........................................ 12
explain.binomial ........................................ 13
explain.chisquared ........................................ 14
explain.covariance ........................................ 15
explain.cv ................................................................. 16
explain.fisher ................................................................. 17
explain.geometricMean ........................................ 18
explain.harmonicMean ........................................ 19
explain.laplace ................................................................. 20
explain.mean ................................................................. 21
explain.median ................................................................. 22
explain.mode ................................................................. 23
explain.normal ................................................................. 24
explain.poisson ................................................................. 25
explain.percentile ................................................................. 26
explain.poisson ................................................................. 27
explain.quartile ................................................................. 28
explain.relative_acum_frecuency ........................................ 29
explain.relative_frecuency ........................................ 30
explain.standardDeviation ........................................ 31
explain.tstudent ................................................................. 32
explain.variance ................................................................. 33
fisher ................................................................. 34
frecuency_abs ................................................................. 35
frecuency_absolute_acum ........................................ 36
frecuency_relative ................................................................. 37
frecuency_relative_acum ........................................ 39
geometricMean_ ................................................................. 40
getUserAction ................................................................. 41
harmonicMean_ ................................................................. 42
initImages ................................................................. 43
interactive.absolute_acum_frecuency ........................................ 44
interactive.absolutefrecuency ........................................ 44
interactive.averageDeviation ........................................ 45
interactive.binomial ................................................................. 46
interactive.chisquared ................................................................. 47
interactive.covariance ................................................................. 48
interactive.cv ................................................................. 48
interactive.fisher ................................................................. 49
interactive.geometricMean ................................................ 50
interactive.harmonicMean ................................................ 51
interactive.laplace ................................................................. 52
interactive.mean ................................................................. 52
averageDeviation_  

Description

This function calculates the average absolute deviation of a numbers vector.

Usage

averageDeviation_(x)

Arguments

x     Should be a numbers vector
Details

To calculate the average deviation, the user should give a numbers vector. The result is the sum of the differences in absolute value between each vector element and the mean, divided by the number of elements. The average absolute deviation formula is the following:

\[
\sigma = \frac{\sum_{i}^{N} |X_i - \bar{X}|}{N}
\]

Value

Numeric, the average absolute deviation of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
# data creation
data <- c(1:20)
result = averageDeviation_(data)
```

---

### Binomial Distribution Calculus Function

**Description**

This function calculates the binomial distribution of experiment.

**Usage**

```
binomial_(n,x,p)
```
Arguments

x Should be a numbers.
n Should be a numbers.
p Should be a numbers.

Details

To calculate the binomial distribution, the user should give three number (the number of trials, probability of success and binomial random variable). The result is a discrete probability distribution that counts the number of successes in a sequence of n independent Bernoulli trials with a fixed probability p of occurrence of success between trials. The binomial distribution formula is the following:

\[
\text{Binomial Distribution} = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}
\]

Value

Numeric, the binomial distribution of three variables.

Note

Each variable is a number. Example: n <- 3 | x <- 2 | p <- 0.7

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

#data creation
n = 3
x = 2
p = 0.7

binomial_(n,x,p)
Description
This function calculates the chisquared distribution of two vectors of numbers.

Usage
chisquared\_ (x, y)

Arguments
- x: Should be a vector.
- y: Should be a vector.

Details
To calculate the chisquared distribution, the user should give two vectors of numbers. The result is a sum of the squares of k independent standard normal random variables. The chisquared distribution formula is the following:

\[
\text{Chi-Squared} = \frac{\sum (f_e - f_o)^2}{f_e}
\]

Value
Numeric, the chisquared distribution of two vectors of numbers.

Note
A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)
Jose Manuel Gomez Caceres, <josemanuel.gomczc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

```r
# data creation
data = c(1,4,3,2,5,7,12,1,2,3,12)
data2 = c(1,2,4,4,6,5,11,2,10,5,6,1)
chisquared_(data, data2)
```

---

**Covariance Calculus Function**

**Description**

This function calculates the covariance of two vectors of numbers.

**Usage**

```r
covariance_(x, y)
```

**Arguments**

- `x`: Should be a vector
- `y`: Should be a vector

**Details**

To calculate the covariance, the user should give two vectors of numbers. The result is a measure of the joint variability of two vectors of numbers. The covariance formula is the following:

\[
\text{Covariance} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n}
\]

**Value**

Numeric, the covariance of two vectors of numbers.

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5
Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
# data creation
data = c(1,4,3,2,5,7,12,1,2,3,12)
data2 = c(1,2,4,6,5,11,2,10,5,6,1)
covariance_(data, data2)
```

---

**cv_**

**Coefficient of Variation Calculus Function**

Description

This function calculates the coefficient of variation of a numbers vector.

Usage

`cv_(x)`

Arguments

- **x**
  Should be a numbers vector

Details

To calculate the coefficient of variation, the user should give a numbers vector. The result is defined as the ratio of the standard deviation to the mean. The coefficient of variation formula is the following:

\[
CV = \frac{\text{std dev}}{\text{mean}} = \frac{s}{\bar{y}}
\]

Value

Numeric, the coefficient of variation of the numbers vector.
drawVector

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples

```r
data = c(1,4,3,3,2,5,7,12,1,2,3,12)

drawVector(data)
```

---

drawVector | Draw Vector Function

Description

This function prints all the elements of a vector

Usage

```r
drawVector(buffer)
```

Arguments

buffer | A vector of elements

Value

There isn’t return value, prints on screen

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5 or c(true,false,false) creates a vector with the booleans: true, false, true

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares
Examples

```r
{  
  # data creation  
  data <- c(1:12)  
  drawVector(data)  
}
```

---

**explain.absolute_acum_frecuency**  
*Absolute Accumulated Frequency Calculus Explained*

---

**Description**

Step by step demonstration of the absolute accumulated frequency calculus

**Usage**

```r
explain.absolute_acum_frecuency(v, x)
```

**Arguments**

- `v` Should be a vector
- `x` Should be a number

**Details**

To calculate the absolute accumulated frequency, the user should give a vector and a number. We can saw the absolute accumulated frecuency formule in the frecuency_acum_absolute help document.

**Value**

A demonstration of the calculus process

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>  
Eduardo Benito, <eduardo.benito@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcala de Henares
explain.absolute_frequency

Absolute Frequency Calculus Explained

Description
Step by step demonstration of the absolute frequency calculus

Usage
explain.absolute_frequency(v, x)

Arguments
v Should be a vector
x Should be a number

Details
To calculate the absolute frequency, the user should give a vector and a number. We can saw the absolute frequency formula in the frequency_abs help document.

Value
A demonstration of the calculus process

Note
A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)
Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples
{
  #data creation
data <- c(1,2,2,5,10,4,2)
value = 2
#function execution
explain.absolute_acum_frequency(data, value)
}
explain.averageDeviation

Average Absolute Deviation Function Explained

Description
Step by step demonstration of the average absolute deviation calculus.

Usage
explain.averageDeviation(x)

Arguments
x
Should be a numbers vector

Details
To calculate the average absolute deviation, the user should give a numbers vector. The result is the explained process to calculate the average absolute deviation, with the data of the dataset provided like argument. We can saw the average absolute deviation formule in the averageDeviation_ help document.

Value
Numeric, the average absolute deviation of the numbers vector.

Note
A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)
Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
# Examples

```r
# data creation
data <- c(7,2,5,7,1,4,12)

explain.averageDeviation(data)
```

## Description

Step by step demonstration of the binomial distribution calculus.

## Usage

```r
explain.binomial(n,x,p)
```

## Arguments

- `x`  
  Should be a numbers.

- `n`  
  Should be a numbers.

- `p`  
  Should be a numbers.

## Details

To calculate the binomial distribution, the user should give three number (the number of trials, probability of success and binomial random variable). The result is a discrete probability distribution that counts the number of successes in a sequence of independent Bernoulli trials with a fixed probability p of occurrence of success between trials. We can saw the binomial distribution formule in the binomial_ help document.

## Value

Numeric result and the process of this calculus explained.

## Note

Each variable is a number. Example: `n <- 3 | x <- 2 | p <- 0.7`

## Author(s)

Jose Manuel Gomez Caceres, `<josemanuel.gomezc@edu.uah.es>`
Juan Jose Cuadrado, `<jjcg@uah.es>`
Universidad de Alcala de Henares
explain.chisquared

Examples

```r
# data creation
n = 3
x = 2
p = 0.7

explain.binomial(n,x,p)
```

explain.chisquared  Chisquared Distribution Function Explained

Description

Step by step demonstration of the chisquared distribution calculus.

Usage

```r
explain.chisquared(x,y)
```

Arguments

- `x`  Should be a vector.
- `y`  Should be a vector.

Details

To calculate the chisquared distribution, the user should give two vectors of numbers. The result is a sum of the squares of k independent standard normal random variables. We can saw the chisquared distribution formule in the chisquared help document.

Value

Numeric result and the process of this calculus explained.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
explain.covariance

Examples

# data creation
data <- c(10,4,5,7,3,4,1)
data2 <- c(1,8,3,4,4,5,7)

explain.chisquared(data, data2)

---

explain.covariance Covariance Function Explained

Description

Step by step demonstration of the covariance calculus.

Usage

explain.covariance(x, y)

Arguments

x Should be a vector
y Should be a vector

Details

To calculate the covariance, the user should give two vectors of numbers. The result is the explained process to calculate the covariance, with the data of the datasets provided like argument. We can saw the harmonic mean formule in the covariance_help document.

Value

Numeric result and the process of this calculus explained.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

```r
# data creation
data <- c(10, 4, 5, 7, 3, 4, 1)
data2 <- c(1, 8, 3, 4, 4, 5, 7)

explain.covariance(data, data2)
```

---

### explain.cv

**Coefficient of Variation Function Explained**

---

**Description**

Step by step demonstration of the coefficient of variation calculus.

**Usage**

```r
explain.cv(x)
```

**Arguments**

- `x` Should be a numbers vector

**Details**

To calculate the coefficient of variation, the user should give a numbers vector. The result is defined as the ratio of the standard deviation to the mean. We can see the coefficient of variation formula in the `cv` help document.

**Value**

Numeric result and the process of this calculus explained.

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
### Examples

```r
# data creation
data <- c(10,4,5,7,3,4,1)
explain.cv(data)
```

### Description

Step by step demonstration of the fisher distribution calculus.

### Usage

```r
explain.fisher(x,y)
```

### Arguments

- **x** Should be a vector
- **y** Should be a vector

### Details

To calculate the fisher distribution, the user should give two vectors of numbers. The result is a continuous probability distribution that arises frequently as the null distribution of a test statistic. We can saw fisher distribution formule in the fisher_help document.

### Value

Numeric result and the process of this calculus explained.

### Note

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

### Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Explain the geometric mean.

Examples

```r
# data creation
data <- c(10, 4, 5, 7, 3, 4, 1)
data2 <- c(1, 8, 3, 4, 4, 5, 7)

explain.fisher(data, data2)
```

---

**explain.geometricMean**  
*Geometric Mean Function Explained*

**Description**

Step by step demonstration of the geometric mean calculus.

**Usage**

```r
explain.geometricMean(x)
```

**Arguments**

- `x`  
  Should be a numbers vector

**Details**

To calculate the geometric mean of a dataset, the user should give a vector. The result is the explained process to calculate the geometric mean, with the data of the dataset provided like argument. We can see the geometric mean formula in the geometricMean help document.

**Value**

Numeric result and the process of this calculus explained.

**Note**

A vector is created by `c()`, like `c(1, 2, 3, 4, 5)` creates a vector with the numbers: 1, 2, 3, 4, 5

**Author(s)**

Dennis Monheimius, <dennis.monheimius@edu.uah.es>  
Eduardo Benito, <eduardo.benito@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcala de Henares
**explain.harmonicMean**

**Examples**

```r
{  # data creation
data <- c(5,21,12,7,3,9,1)
   explain.geometricMean(data)
}
```

---

**explain.harmonicMean**  *Harmonic Mean Function Explained*

**Description**

Step by step demonstration of the harmonic mean calculus.

**Usage**

`explain.harmonicMean(x)`

**Arguments**

- **x**
  Should be a numbers vector

**Details**

To calculate the harmonic mean, the user should give a numbers vector. The result is the explained process to calculate the harmonic mean, with the data of the dataset provided like argument. We can saw the harmonic mean formule in the harmonicMean help document.

**Value**

Numeric result and the process of this calculus explained.

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

# data creation
data <- c(10,4,5,7,3,4,1)

explain.harmonicMean(data)

explain.laplace  Laplace's Rule Function Explained

Description

Step by step demonstration of the Laplace's rule calculus.

Usage

explain.laplace(x,y)

Arguments

x Should be a vector
y Should be a vector

Details

To calculate the Laplace's rule, the user should give two vector (unfavorable cases/favorable cases). The result is the quotient between the number of favorable cases to A, and that of all possible results of the experiment. We can saw the Laplace's rule correlation formule in the laplace help document.

Value

Numeric result and the process of this calculus explained.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
explain.mean

Examples

```r
#data creation
data <- 3
data2 <- c(1,2,3,4,5,6)

explain.laplace(data, data2)
```

---

**Mean Function Explained**

**Description**

Step by step demonstration of the arithmetic mean calculus.

**Usage**

```r
explain.mean(x)
```

**Arguments**

- `x`: Should be a numbers vector

**Details**

To calculate the arithmetic mean of a dataset, the user should give a vector. The result is the explained process to calculate the arithmetic mean, with the data of the dataset provided like argument. We can see the arithmetic mean formula in the `mean` help document.

**Value**

Numeric result and the process of this calculus explained.

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

- Dennis Monheimius, <dennis.monheimius@edu.uah.es>
- Eduardo Benito, <eduardo.benito@edu.uah.es>
- Juan Jose Cuadrado, <jjcg@uah.es>

Universidad de Alcala de Henares
explain.median

Examples
{
  #data creation
  data <- c(1,2,2,5,10,4,2)

  explain.mean(data)
}

---

explain.median  

Median Function Explained

Description

Step by step demonstration of the median calculus.

Usage

explain.median(x)

Arguments

x
  Should be a numbers vector

Details

To calculate the median, the user should give a numbers vector. The result is the explained process to calculate the median, with the data of the dataset provided like argument. We can saw the median formule in the median_help document.

Value

Numeric result and the process of this calculus explained.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares
**explain.mode**

**Examples**

```r
{  
    # data creation  
    data <- c(1,2,2,5,10,4,2)  
    
    explain.median(data)  
}
```

---

**explain.mode**  
*Mode Function Explained*

**Description**

Step by step demonstration of the mode calculus.

**Usage**

```r
explain.mode(x)
```

**Arguments**

- `x`  
  Should be a numbers vector

**Details**

To calculate the mode, the user should give a numbers vector. The result is the explained process to calculate the mode, with the data of the dataset provided like argument. We can saw the mode formule in the mode_ help document.

**Value**

Numeric result and the process of this calculus explained.

**Note**

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

Dennis Monheimius, <dennis.monheimius@edu.uah.es>  
Eduardo Benito, <eduardo.benito@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcala de Henares
Examples
{
    # data creation
    data <- c(1, 1, 2, 5, 2, 3, 1, 4, 1)
    explain.mode(data)
}

explain.normal

Normal Distribution Function Explained

Description
Step by step demonstration of the normal distribution calculus.

Usage
explain.normal(x)

Arguments
x Should be a number.

Details
To calculate the normal distribution, the user should give a number. The result is a type of continuous probability distribution for a real-valued random variable. We can saw the normal distribution correlation formula in the normal_help document.

Value
Numeric result and the process of this calculus explained.

Note
The variable is a number. Example: x <- 0.1

Author(s)
Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

```r
#data creation
x = 0.1
explain.normal(x)
```

---

**explain.pearson**  
*Pearson Correlation Function Explained*

### Description

Step by step demonstration of the pearson correlation calculus.

### Usage

```r
explain.pearson(x, y)
```

### Arguments

- **x**  
  Should be a vector

- **y**  
  Should be a vector

### Details

To calculate the pearson correlation, the user should give two vectors of numbers. The result is the covariance of the two vectors of numbers divided by the product of their standard deviations. We can saw the pearson correlation formule in the pearson_ help document.

### Value

Numeric result and the process of this calculus explained.

### Note

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

### Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcala de Henares
Examples

```r
# data creation
data <- c(10,4,5,7,3,4,1)
data2 <- c(1,8,3,4,4,5,7)

explain.pearson(data, data2)
```

---

**explain.percentile**  
*Percentiles Calculus Explained*

**Description**

Step by step demonstration of the percentiles calculus

**Usage**

```r
explain.percentile(x)
```

**Arguments**

- `x` Should be a vector

**Details**

To calculate the percentiles, the user should give a vector. We can see the percentile formula in the `percentile` help document.

**Value**

A demonstration of the calculus process

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

{
    #data creation
    data <- c(1,2,2,5,10,4,2)
    explain.percentile(data)
}

explain.poisson  Poisson Distribution Function Explained

Description

Step by step demonstration of the Poisson distribution calculus.

Usage

explain.poisson(k,lam)

Arguments

k            Should be a numbers
lam          Should be a numbers

Details

To calculate the Poisson distribution, the user should give two number ( the number of times the phenomenon and the number of occurrences). The result is a discrete probability distribution that expresses, from a mean frequency of occurrence, the probability that a certain number of events will occur during a certain period of time. We can saw the Poisson distribution correlation formule in the poisson_ help document.

Value

Numeric result and the process of this calculus explained.

Note

Each variable is a number. Example: lam <- 2 | k <- 3

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

```r
#data creation
lam = 2
k = 3
explain.poisson(k, lam)
```

---

**explain.quartile**  
*Quartiles Calculus Explained*

**Description**

Step by step demonstration of the quartiles calculus

**Usage**

```r
explain.quartile(x)
```

**Arguments**

- `x` Should be a vector

**Details**

To calculate the quartiles, the user should give a vector. We can saw the quartile formule in the quartile help document.

**Value**

A demonstration of the calculus process

**Note**

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
explain.relative_acum_frequency

Relative Accumulated Frequency Calculus Explained

Description
Step by step demonstration of the relative accumulated frequency calculus

Usage
explain.relative_acum_frequency(v,x)

Arguments
v Should be a vector
x Should be a number of the vector

Details
To calculate the relative accumulated frequency, the user should give a vector and a number. We can saw the relative accumulated frequency formula in the frequency_acum_relative help document.

Value
A demonstration of the calculus process

Note
A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)
Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples
{
    #data creation
data <- c(1,2,2,5,10,4,2)
value = 2
#function execution
explain.relative_acum_frecuency(data, value)
}

explain.relative_frecuency

Relative Frequency Calculus Explained

Description
Step by step demonstration of the relative frequency calculus

Usage
explain.relative_frecuency(v,x)

Arguments
v
    Should be a vector
x
    Should be a number

Details
To calculate the relative frequency, the user should give a vector and a number. We can saw the
relative frequency formule in the frecuency_relative help document.

Value
A demonstration of the calculus process

Note
A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)
Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
explain.standardDeviation

Standard Deviation Function Explained

Description

Step by step demonstration of the standard deviation calculus.

Usage

explain.standardDeviation(x)

Arguments

x

Should be a numbers vector

Details

To calculate the standard deviation, the user should give a numbers vector. The result is the explained process to calculate the standard deviation, with the data of the dataset provided like argument. We can see the standard deviation formula in the standardDeviation_help document.

Value

Numeric result and the process of this calculus explained.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.mohonimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples

{
  # data creation
  data <- c(1,2,2,5,10,4,2)
  value = 2
  # function execution
  explain.relative_frequency(data, value)
}
Examples

```r
# data creation
data <- c(1,5,3,7,10,4,2)

explain.standardDeviation(data)
```

---

### T-Student Distribution Function Explained

#### Description

Step by step demonstration of the T-Student distribution calculus.

#### Usage

```r
explain.tstudent(x,u,s,n)
```

#### Arguments

- `x`: Should be a number
- `u`: Should be a number
- `s`: Should be a number
- `n`: Should be a number

#### Details

To calculate the T-Student distribution, the user should give four number (sample mean, population mean, population standard deviation and sample size). The result is a probability distribution that arises from the problem of estimating the mean of a normally distributed population when the sample size is small. We can saw the T-Student distribution formule in the tstudent help document.

#### Value

Numeric result and the process of this calculus explained.

#### Note

Each variable is a number. Example: `x <- 2 | y <- 4`

#### Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
explain.variance

Examples

```r
# data creation
x = 52.9
u = 50
s = 3
n = 10

explain.tstudent(x,u,s,n)
```

---

**explain.variance**  Variance Function Explained

**Description**

Step by step demonstration of the variance calculus.

**Usage**

```r
explain.variance(x)
```

**Arguments**

- **x**  Should be a numbers vector

**Details**

To calculate the variance, the user should give a numbers vector. The result is the explained process to calculate the variance, with the data of the dataset provided like argument. We can saw the variance formule in the variance_help document.

**Value**

Numeric result and the process of this calculus explained.

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

**Author(s)**

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

```r
# data creation
data <- c(10,4,5,7,3,4,1)

explain.variance(data)
```

**fisher_.**  

*F Fisher Distribution Calculus Function*

**Description**

This function calculates the fisher distribution of a numbers vector.

**Usage**

```r
fisher_(x, y)
```

**Arguments**

- `x` Should be a vector
- `y` Should be a vector

**Details**

To calculate the fisher distribution, the user should give two vectors of numbers. The result is a continuous probability distribution that arises frequently as the null distribution of a test statistic. The fisher distribution formula is the following:

\[
Fisher Distribution = \frac{S_x^2}{S_w^2}
\]

**Value**

Numeric, the fisher distribution.

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5
Author(s)
Jose Manuel Gomez Caceres, <josemanuel.gomez@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
# data creation
x <- c(70, 75, 74, 72, 68, 59)
y <- c(74, 77, 70, 80, 72, 76)

frecuency_abs(x, y)
```

---

### frecuency_abs

**Absolute Frequency Calculus**

**Description**

This function calculates the number of times that a specific number appears in the data set.

**Usage**

```r
frecuency_abs(v, x)
```

**Arguments**

- `v` Should be a vector
- `x` Should be a number

**Details**

The absolute frequency formula is the following:

\[
\text{Absolute frequency} = \text{number of aparitions of the examined element}
\]

**Value**

An integer that represents the number of times that the value appears in the vector
Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```
#data creation
data = c(1,4,3,3,2,5,7,12,1,2,3,12)
value = 12
#function execution
frecuency_abs(data, value)
```

frecuency_absolute_acum

*Accumulated Absolute Freucency Calculus*

Description

This function calculate the number of times that a specific number appears in the data set. The value depends on the elements that are lower than itself

Usage

`frecuency_absolute_acum(v,x)`

Arguments

- `v` Should be a vector
- `x` Should be a number
Details

The accumulated absolute frequency formula is the following:

\[
\text{Absolute accumulated frequency (X)} = \sum \text{Fi where } i \leq X
\]

Value

A double that represents the number of times that the value appears in the vector regarding the total of elements

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
#data creation
data = c(1,4,3,3,2,5,7,12,1,2,3,12)
value = 12
#function execution
frecuency_absolute_acum(data, value)
```

Description

This function calculate the number of times that a specific number appears in the data set divided by the total length of the vector.
Usage

frecuency_relative(v,x)

Arguments

v  Should be a vector
x  Should be a number

Details

The relative frequency formula is the following:

\[
\text{Relative frequency} = \frac{\text{absolute frequency}}{\sum \text{all frequencies}}
\]

Value

A double that represents the number of times that the value appears in the vector regarding the total of elements

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```
{  #data creation
data <- c(1,4,3,3,2,5,7,12,1,2,3,12)
value = 12
frecuency_relative(data, value)
}
```
Description

This function calculate the number of times that a specific number appears in the data set divided by the total length of the vector. The value depends on the elements that are lower than itself.

Usage

frecuency_relative_acum(v,x)

Arguments

v  Should be a vector
x  Should be a number

Details

The accumulated relative frecuency formula is the following:

\[
\text{Relative accumulated frequency (X)} = \sum_{i \leq X} f_i
\]

Value

A double that represents the number of times that the value appears in the vector regarding the total of elements

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

{  
    #data creation
    data = c(1,4,3,3,2,5,7,12,1,2,3,12)
    value = 12
    #function execution
    frecuency_relative_acum(data, value)
}

geometricMean_  

Geometric Mean Function

Description

This function calculates the geometric mean of a numbers vector.

Usage

geometricMean_(x)

Arguments

x  

Should be a numbers vector

Details

To calculate the geometric mean of a dataset, the user should give a numbers vector. The result is the product of all vector elements raise to 1 divided by the number of elements. The arithmetic mean formula is the following:

\[ MG = \sqrt[n]{(X_1)(X_2)(X_3)...(X_n)} \]

Value

A numeric, the geometric mean of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5
getUserAction

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

#data creation
data = c(1:20)

geometricMean_(data)

---

getUserAction Get User Action Function

Description

This function get the buffer introduced by the user. Typically a numerical vector.

Usage

getUserAction()

Value

A vector

Note

The process is interactive with the user.

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

{
  ## Not run:
  vector <- getUserAction()

  ## End(Not run)
  
}
Harmonic Mean Function

Description

This function calculates the harmonic mean of a numbers vector.

Usage

harmonicMean_ (x)

Arguments

x  Should be a numbers vector

Details

To calculate the harmonic mean, the user should give a numbers vector. The result is calculated by dividing the number of observations by the reciprocal of each number in the vector. The harmonic mean formula is the following:

\[
	ext{Harmonic Mean} = \frac{N}{\frac{1}{X_1} + \frac{1}{X_2} + \ldots + \frac{1}{X_N}}
\]

Value

Numeric, the harmonic mean of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

```r
# data creation
data = c(1,4,3,3,2,5,7,12,1,2,3,12)
harmonicMean_(data)
```

### initImages

<table>
<thead>
<tr>
<th>initImages</th>
<th>Init Images Function</th>
</tr>
</thead>
</table>

Description

This function is used to display an image.

Usage

```r
initImages(path)
```

Arguments

- `path` : An url of an image

Value

There isn’t return value

Note

The path should be toward an image

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
{
  ## Not run:
  path = "https://i.imgur.com/8237YhzJ.png"
  initImages(path)

  ## End(Not run)
}
```
interactive.absolute_acum_frecuency

User Interactive Absolute Accumulated Frequency Calculus

Description
Interactive function for absolute accumulated frequency calculus.

Usage
interactive.absolute_acum_frecuency()

Details
The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the formula.

Value
An interactive process to calculate the absolute accumulated frequency

Author(s)
Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples

```r
## Not run:
interactive.absolute_acum_frecuency()
## End(Not run)
```

interactive.absolute_frecuency

User Interactive Absolute Frequency Calculus

Description
Interactive function for absolute frequency calculus.

Usage
interactive.absolute_frecuency()
**interactive.averageDeviation**

**Description**

Interactive function for average absolute deviation calculus.

**Usage**

```r
interactive.averageDeviation()
```

**Details**

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the formula.

**Value**

An interactive process to calculate the average absolute deviation
Author(s)

Dennis Monheimius, <dennis.monhemius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>

References

https://en.wikipedia.org/wiki/Average_absolute_deviation

Examples

## Not run:
interactive.averageDeviation()

## End(Not run)

---

**interactive.binomial**  
*User Interactive Binomial Distribution Calculus*

Description

Interactive function for binomial distribution calculus.

Usage

interactive.binomial()

Details

The user provides the values when the function needs it. After that, the function will ask what is the correct result for this datas. The function itself will provide the binomial distribution formule, apart from the binomial_<doc> help document.

Value

An interactive process to calculate the binomial distribution.

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
interactive.chisquared

User Interactive Chisquared Distribution Calculus

Description

Interactive function for chisquared distribution calculus.

Usage

interactive.chisquared()

Details

The user provides the datasets when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the chisquared distribution formula, apart from the chisquared_ help document.

Value

An interactive process to calculate the chisquared distribution

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

## Not run:
interactive.binomial()

## End(Not run)
interactive.covariance

User Interactive Covariance Calculus

Description

Interactive function for covariance calculus.

Usage

interactive.covariance()

Details

The user provides the datasets when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the covariance formula, apart from the covariance help document.

Value

An interactive process to calculate the covariance

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
## Not run:
interactive.covariance()

## End(Not run)
```

interactive.cv

User Interactive Coefficient of Variation Calculus

Description

Interactive function for Coefficient of Variation calculus.

Usage

interactive.cv()
**interactive.fisher**  

**Details**  

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the coefficient of variation formula, apart from the cv help document.

**Value**  

An interactive process to calculate the average absolute deviation

**Author(s)**  

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcalá de Henares

**Examples**  

```r  
## Not run:  
interactive.cv()  
## End(Not run)
```

---

**interactive.fisher**  

*User Interactive F Fisher Distribution Calculus*

**Description**  

Interactive function for fisher distribution calculus.

**Usage**  

`interactive.fisher()`

**Details**  

The user provides the datasets when the function needs it. After that, the function will ask what is the correct result for this datasets. The function itself will provide the fisher distribution formula, apart from the fisher help document.

**Value**  

An interactive process to calculate the fisher distribution
interactive.geometricMean

Author(s)
José Manuel Gómez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan José Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples

## Not run:
interactive.fisher()

## End(Not run)

---

interactive.geometricMean

User Interactive Geometric Mean Calculus

Description
Interactive function for geometric mean calculus.

Usage
interactive.geometricMean()

Details
The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the geometric mean formula, apart from the geometricMean_.help document.

Value
An interactive process to calculate the geometric mean.

Author(s)
Dennis Monheimius, <dennis.monhemius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan José Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares
### Description

Interactive function for harmonic mean calculus.

### Usage

```r
interactive.harmonicMean()
```

### Details

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the harmonic mean formula, apart from the harmonicMean_ help document.

### Value

An interactive process to calculate the harmonic mean

### Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

### Examples

```r
## Not run:
interactive.harmonicMean()
## End(Not run)
```
interactive.laplace  User Interactive Laplace’s Rule Calculus

Description

Interactive function for Laplace’s rule calculus.

Usage

interactive.laplace()

Details

The user provides the values when the function needs it. After that, the function will ask what is the correct result for this data. The function itself will provide the Laplace’s rule formula, apart from the laplace_help document.

Value

An interactive process to calculate the Laplace’s rule.

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
## Not run:
interactive.laplace()
## End(Not run)
```

interactive.mean  User Interactive Mean Calculus

Description

Interactive function for arithmetic mean calculus.

Usage

interactive.mean()
interactive.median

**Details**

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the arithmetic mean formula, apart from the mean_ help document.

**Value**

An interactive process to calculate the arithmetic mean.

**Author(s)**

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

**Examples**

```r

## Not run:
interactive.median()
## End(Not run)
```

**Description**

Interactive function for median calculus.

**Usage**

`interactive.median()`

**Details**

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the median formula, apart from the median_ help document.

**Value**

An interactive process to calculate the median
Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
## Not run:
interactive.mode()
## End(Not run)
```

---

**interactive.mode**  
*User Interactive Mode Calculus*

Description

Interactive function for mode calculus.

Usage

`interactive.mode()`

Details

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset.

Value

An interactive process to calculate the mode.

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples

```r
## Not run:
interactive.mode()
## End(Not run)
```
**interactive.normal**  
*User Interactive Normal Distribution Calculus*

**Description**
Interactive function for normal distribution calculus.

**Usage**
```r
interactive.normal()
```

**Details**
The user provides the values when the function needs it. After that, the function will ask what is the correct result for this data. The function itself will provide the normal distribution formula, apart from the normal_ help document.

**Value**
An interactive process to calculate the normal distribution.

**Author(s)**
Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcala de Henares

**Examples**
```r
## Not run:  
interactive.normal()
## End(Not run)
```

---

**interactive.pearson**  
*User Interactive Pearson Correlation Calculus*

**Description**
Interactive function for pearson correlation calculus.

**Usage**
```r
interactive.pearson()
```
interactive.percentile

Details

The user provides the datasets when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the pearson correlation formula, apart from the pearson_ help document.

Value

An interactive process to calculate the pearson correlation.

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjc@uah.es>
Universidad de Alcala de Henares

Examples

```r
## Not run:
interactive.pearson()
## End(Not run)
```

---

interactive.percentile

User Interactive Percentile Calculus

Description

Interactive function for percentiles calculus.

Usage

interactive.percentile()

Details

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the formula.

Value

An interactive process to calculate the percentiles
interactive.poisson

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
## Not run:
interactive.percentile()

## End(Not run)
```

---

## Description

Interactive function for Poisson distribution calculus.

## Usage

```r
interactive.poisson()
```

## Details

The user provides the values when the function needs it. After that, the function will ask what is the correct result for this data. The function itself will provide the Poisson distribution formula, apart from the poisson_help document.

## Value

An interactive process to calculate the Poisson distribution.

## Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
## Not run:
interactive.poisson()

## End(Not run)
```
**interactive.quartile**  
*User Interactive Quartiles Calculus*

**Description**
Interactive function for quartiles calculus.

**Usage**
```r
interactive.quartile()
```

**Details**
The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the formula.

**Value**
An interactive process to calculate the quartiles

**Author(s)**
Dennis Monheimius, <dennis.monhemimius@edu.uah.es>  
Eduardo Benito, <eduardo.benito@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcala de Henares

**Examples**
```r
## Not run:
interactive.quartile()
## End(Not run)
```

---

**interactive.relative_acum_freqency**  
*User Interactive Relative Accumulated Frecuency Calculus*

**Description**
Interactive function for relative accumulated frecuency calculus.

**Usage**
```r
interactive.relative_acum_freqency()
```
interactive.relative_frecuency

Details
The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the formula.

Value
An interactive process to calculate the relative accumulated frequency

Author(s)
Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples
```r
## Not run:
interactive.relative_acum_frecuency()
```

## End(Not run)

---

interactive.relative_frecuency

*User Interactive Relative Frequency Calculus*

Description
Interactive function for relative frequency calculus.

Usage
```r
interactive.relative_frecuency()
```

Details
The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the formula.

Value
An interactive process to calculate the relative frequency
Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
## Not run:
interactive.relative_frequency()

## End(Not run)
```

---

interactive.standardDeviaton

User Interactive Standard Deviation Calculus

Description

Interactive function for standard deviation calculation.

Usage

```r
interactive.standardDeviaton()
```

Details

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the standard deviation formula, apart from the standardDeviaton_help document.

Value

An interactive process to calculate the standard deviation

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>

References

Examples

```r
## Not run:
interactive.standardDeviation()

## End(Not run)
```

---

**interactive.tstudent**  
*User Interactive T-Student Distribution Calculus*

**Description**

Interactive function for T-Student distribution calculus.

**Usage**

`interactive.tstudent()`

**Details**

The user provides the values when the function needs it. After that, the function will ask what is the correct result for this data. The function itself will provide the T-Students distribution formula, apart from the tstudent_help document.

**Value**

An interactive process to calculate the T-Student distribution.

**Author(s)**

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcalá de Henares

**Examples**

```r
## Not run:
interactive.tstudent()

## End(Not run)
```
### interactive.variance  *User Interactive Variance Calculus*

**Description**

Interactive function for variance calculus.

**Usage**

`interactive.variance()`

**Details**

The user provides the dataset when the function needs it. After that, the function will ask what is the correct result for this dataset. The function itself will provide the variance formula, apart from the variance help document.

**Value**

An interactive process to calculate the average absolute deviation

**Author(s)**

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@iah.es>
Universidad de Alcala de Henares

**Examples**

```r
## Not run:
interactive.variance()
## End(Not run)
```

---

### laplace_  *Laplace’s Rule Calculus Function*

**Description**

This function calculates the Laplace’s rule of experiment.

**Usage**

`laplace_(x,y)`
Arguments

- `x` Should be a vector
- `y` Should be a vector

Details

To calculate the Laplace’s rule, the user should give two vector (unfavorable cases/favorable cases). The result is the quotient between the number of favorable cases to `A`, and that of all possible results of the experiment. The Laplace’s rule formula is the following:

\[
\text{LaPlace Rule} = \frac{\text{casos favorables}}{\text{casos posibles}}
\]

Value

Numeric, the pearson correlation.

Note

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
# data creation
data = 3
data2 = c(1,2,3,4,5,6)
laplace_(data, data2)
```
LearningRlab  Statistical Learning Functions

Description

Package used to teach basic statistics to students.

Details

This package pretends to serve the user as a method of learning basic statistical functions at secondary and baccalaureate courses. The content of the package incorporate a serie of statistical functions like the calculus of the arithmetic mean or the calculus of the frequencies. There is no only calculus functions, further more, there are incorporated interactive and explicative functions to help and guide the user in the learning process.

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Maintainer: Eduardo Benito <eduardo.benito@edu.uah.com>

meanC  Mean Function Developed in C

Description

This function calculates the arithmetic mean of a numbers vector.

Usage

meanC(x)

Arguments

x  Should be a numbers vector
Details

To calculate the arithmetic mean of a dataset, the user should give a numbers vector. The result is the addition of all vector elements divided by the number of elements. The arithmetic mean formule is the following:

\[ \bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \]

Value

A numeric, the arithmetic mean of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
#data creation
vector = c(1:10)
meanC(vector)
```

Description

This function calculates the arithmetic mean of a numbers vector.

Usage

```r
mean_(x)
```
Arguments

x  Should be a numbers vector

Details

To calculate the arithmetic mean of a dataset, the user should give a numbers vector. The result is the addition of all vector elements divided by the number of elements. The arithmetic mean formula is the following:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{X_1 + X_2 + \ldots + X_n}{n}$$

Value

A numeric, the arithmetic mean of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
vector <- c(2,4,6,8,10,12,14,16,18)
result = mean_(vector)
result
```

---

**median_**

Median Calculus Function

Description

This function calculates the median of a numbers vector.
Usage

median_(x)

Arguments

x: Should be a numbers vector

Details

To calculate the median, the user should give a numbers vector. The result is the value separating the higher half from the lower half of the dataset, it may be thought of as the middle value. The median formula is the following:

\[ \text{Median} = \frac{1}{2} (n + 1) \text{th value} \]

Value

A numeric, the median of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```
{ 
  result = median_(c(1,3,2,5,12,4,4,2,9))
  result
}
```
Mode Calculus Function

Description

This function calculates the mode of a numbers vector.

Usage

mode_(x)

Arguments

x Should be a numbers vector

Details

To calculate the mode of a dataset, the user should give a numbers vector. The result is the numeric value that appears most often. In other words, it's the value that is most likely to be sampled. The mode formula is the following:

\[
\text{Mode} = l_1 + \left( \frac{f_0 - f_{-1}}{2f_0 - f_{-1} - f_1} \right) \times c
\]

Value

Numeric, the mode of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples

{
    #data creation
    data = c(1,2,2,3,4)
    mode_(data)
normal_  

Normal Distribution Calculus Function

Description

This function calculates the normal distribution of experiment.

Usage

normal_(x)

Arguments

x  Should be a number.

Details

To calculate the normal distribution, the user should give a number. The result is a type of continuous probability distribution for a real-valued random variable. The normal distribution formula is the following:

\[ \text{Normal Distribution} = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \, dx \]

Value

Numeric, the normal distribution.

Note

The variable is a number. Example: x <- 0.1

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares
Examples

```r
#data creation
x = 0.1
normal_(x)
```

---

**pearson_**  
*Pearson Correlation Calculus Function*

**Description**

This function calculates the pearson correlation of two vectors of numbers.

**Usage**

`pearson_(x, y)`

**Arguments**

- `x`  
  Should be a vector

- `y`  
  Should be a vector

**Details**

To calculate the pearson correlation, the user should give two vectors of numbers. The result is the covariance of the two vectors of numbers divided by the product of their standard deviations. The pearson correlation formula is the following:

\[
\text{Pearson Correlation} = \frac{\text{Cov}(x, y)}{S_x \times S_y}
\]

**Value**

Numeric, the pearson correlation of two vectors of numbers.

**Note**

A vector is created by `c()`, like `c(1,2,3,4,5)` creates a vector with the numbers: 1,2,3,4,5
percentile_ Percentile Calculus Function

Description

This function calculate the percentiles of a vector of numbers

Usage

percentile_(x, p)

Arguments

x Should be a vector
p Should be a number, 0 => y <= 1

Details

To calculate the percentiles, the user should give a vector. This function divide the dataset in 100 parts as equal as possible. The formula is the following:

\[ P_i = \begin{cases} \frac{\text{elemento}(E + 1)}{2} & \text{para } D 
eq 0 \\ \frac{\text{elemento}(E) + \text{elemento}(E+1)}{2} & \text{para } D = 0 \end{cases} \]

Value

A vector sorted with the elements divided by 100 parts

Examples

#data creation
data = c(1,4,3,3,2,5,7,12,1,2,3,12)
data2 = c(1,2,4,4,6,5,11,2,10,5,6,1)
pearson_(data, data2)
Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```
{  
  #data creation  
  data = c(1,4,3,3,2,5,7,12,1,2,3,12)  
  percentile_(data,0.3)  
}
```

---

**poisson_**  
*Poisson Distribution Calculus Function*

Description

This function calculates the Poisson distribution of experiment.

Usage

```
poisson_(k, lam)
```

Arguments

```
k  Should be a numbers  
lam  Should be a numbers  
```

Details

To calculate the Poisson distribution, the user should give two number ( the number of times the phenomenon and the number of occurrences). The result is a discrete probability distribution that expresses, from a mean frequency of occurrence, the probability that a certain number of events
will occur during a certain period of time. The Poisson distribution formula is the following:

\[
\text{Poisson Distribution} \quad = \frac{\lambda^x e^{-\lambda}}{x!}
\]

Value

Numeric, the Pearson correlation of two numbers.

Note

Each variable is a number. Example: \( \text{lam} \leftarrow 2 \mid k \leftarrow 3 \)

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomezc@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
# data creation
lam = 2
k = 3
poisson_(k,lam)
```

Quartiles Calculus

Description

Calculates the 3 quartiles of a vector of data.

Usage

\text{quartile}_x(x)

Arguments

- \text{x} Should be a vector
Details
To calculate the quartiles, the user should give a vector. This function divide the dataset in 4 parts as equal as possible. The formula is the following:

\[ Q_i = \frac{i(N)}{4} \text{ th value} \]

Value
A vector sorted with the elements divided by 4 parts

Note
A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)
Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples
{
    #data creation
    data = c(1:20)

    quartile_(data)
}

---

**standardDeviation_**

**Standard Deviation Calculus Function**

Description
This function calculates the standard deviation of a numbers vector.

Usage
standardDeviation_(x)

Arguments

x Should be a numbers vector
Details

To calculate the standard deviation, the user should give a numbers vector. The result is the square root of the sum of the differences between each vector element and the mean squared divided by the number of elements. The standard deviation formula is the following:

\[ s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2} \]

Value

Numeric, the standard deviation of the numbers vector.

Note

A vector is created by c(), like c(1,2,3,4,5) creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monhemimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

```r
# data creation
data = c(1,4,3,3,2,5,7,12,1,2,3,12)
standardDeviation_(data)
```

---

tstudent_  

**T-Student Distribution Calculus Function**

Description

This function calculates the T-Student distribution of experiment.

Usage

tstudent_(x,u,s,n)
Arguments

- **x**: Should be a number
- **u**: Should be a number
- **s**: Should be a number
- **n**: Should be a number

Details

To calculate the T-Student distribution, the user should give four number (sample mean, population mean, population standard deviation and sample size). The result is a probability distribution that arises from the problem of estimating the mean of a normally distributed population when the sample size is small. The T-Student distribution formula is the following:

\[
\text{T-Student Distribution} = \frac{x - \mu}{\sigma / \sqrt{n}}
\]

Value

Numeric, the T-Student distribution.

Note

Each variable is a number. Example: x <- 2 | y <- 4

Author(s)

Jose Manuel Gomez Caceres, <josemanuel.gomez@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcalá de Henares

Examples

```r
#data creation
x = 52.9
u = 50
s = 3
n = 10
tstudent_(x, u, s, n)
```
Description

This function calculates the variance of a numbers vector.

Usage

\texttt{variance\_}(x)

Arguments

\begin{itemize}
  \item \texttt{x} \quad \text{Should be a numbers vector}
\end{itemize}

Details

To calculate the variance, the user should give a numbers vector. The result is the expectation of the squared deviation of all numbers vector from its mean. The variance formule is the following:

\[
\text{Variance} = \frac{\sum (X - \bar{X})^2}{N - 1}
\]

Value

Numeric, the variance of the numbers vector.

Note

A vector is created by \texttt{c()}, like \texttt{c(1,2,3,4,5)} creates a vector with the numbers: 1,2,3,4,5

Author(s)

Dennis Monheimius, <dennis.monheimius@edu.uah.es>
Eduardo Benito, <eduardo.benito@edu.uah.es>
Juan Jose Cuadrado, <jjcg@uah.es>
Universidad de Alcala de Henares

Examples

\begin{verbatim}
#data creation
data = c(1,4,3,3,2,5,7,12,1,2,3,12)

variance\_\(data)\n\end{verbatim}
Index

* ~absoluta
  averageDeviation_, 3
  explain.averageDeviation, 12
  frequency_abs, 35
  frequency_absolute_acum, 36
  interactive.averageDeviation, 45
* ~absolute
  averageDeviation_, 3
  explain.absolute_acum_frecuency, 10
  explain.absolute_frecuency, 11
  explain.averageDeviation, 12
  frequency_abs, 35
  frequency_absolute_acum, 36
  interactive.absolute_acum_frecuency, 44
  interactive.averageDeviation, 45
* ~accumulated
  explain.absolute_acum_frecuency, 10
  explain.relative_acum_frecuency, 29
  frequency_absolute_acum, 36
  frequency_relative_acum, 39
  interactive.absolute_acum_frecuency, 44
  interactive.relative_acum_frecuency, 58
* ~acumulada
  explain.absolute_acum_frecuency, 10
  explain.relative_acum_frecuency, 29
  frequency_absolute_acum, 36
  frequency_relative_acum, 39
* ~arithmetic.mean
  mean_, 65
  meanC, 64
* ~armonica
  explain.harmonicMean, 19
  harmonicMean_, 42
  interactive.harmonicMean, 51
* ~average
  averageDeviation_, 3
  explain.averageDeviation, 12
  interactive.averageDeviation, 45
* ~binomial
  binomial_, 4
  explain.binomial, 13
  interactive.binomial, 46
* ~chicuadrado
  chisquared_, 6
  explain.chisquared, 14
  interactive.chisquared, 47
* ~chisquared
  chisquared_, 6
  explain.chisquared, 14
  interactive.chisquared, 47
* ~correlation
  explain.covariance, 15
* ~covariance
  covariance_, 7
  interactive.covariance, 48
  interactive.pearson, 55
* ~covarianza
  covariance_, 7
  explain.covariance, 15
  interactive.covariance, 48
  interactive.pearson, 55
* ~cuartiles
  interactive.quartile, 58
  quartile_, 73
* ~cv
  cv_, 8
  explain.cv, 16
  interactive.cv, 48
* ~desviacion
  averageDeviation_, 3
  explain.averageDeviation, 12
  explain.standardDeviation, 31
  interactive.averageDeviation, 45
  interactive.standardDeviation, 60
  standardDeviation_, 74
* ~deviation
  averageDeviation_, 3
  explain.averageDeviation, 12
  explain.standardDeviation, 31
  interactive.averageDeviation, 45
  interactive.standardDeviation, 60
  standardDeviation_, 74
* ~distribucion
  binomial_, 4
  explain.binomial, 13
  explain.fisher, 17
  explain.normal, 24
  explain.poisson, 27
  explain.tstudent, 32
  fisher_, 34
  interactive.binomial, 46
  interactive.fisher, 49
  interactive.normal, 55
  interactive.poisson, 57
  interactive.tstudent, 61
  normal_, 69
  poisson_, 72
  tstudent_, 75
* ~explicada
  explain.averageDeviation, 12
  explain.binomial, 13
  explain.chisquared, 14
  explain.fisher, 17
  explain.normal, 24
  explain.poisson, 27
  explain.tstudent, 32
  explain.variance, 33
* ~explicación
  averageDeviation_, 3
  explain.averageDeviation, 12
  explain.standardDeviation, 31
  explain.binomial, 13
  explain.chisquared, 14
  explain.fisher, 17
  explain.normal, 24
  explain.poisson, 27
  explain.tstudent, 32
  explain.variance, 33
explain.covariance, 15
explain.cv, 16
explain.fisher, 17
explain.geometricMean, 18
explain.harmonicMean, 19
explain.laplace, 20
explain.mean, 21
explain.median, 22
explain.mode, 23
explain.normal, 24
explain.poisson, 25
explain.standardDeviation, 31
explain.tstudent, 32
explain.variance, 33

* ~fisher
  explain.fisher, 17
  interactive.fisher, 49

* ~frecuencia
  frequency_abs, 35
  frequency_absolute_acum, 36
  frequency_relative, 37
  frequency_relative_acum, 39
  interactive.absolute_frecuency, 44
  interactive.relative_frecuency, 59

* ~frecuencia
  explain.absolute_acum_frecuency, 10
  explain.absolute_frecuency, 11
  explain.relative_acum_frecuency, 29
  explain.relative_frecuency, 30
  frequency_abs, 35
  frequency_absolute_acum, 36
  frequency_relative, 37
  frequency_relative_acum, 39
  interactive.absolute_acum_frecuency, 44
  interactive.absolute_frecuency, 44
  interactive.relative_acum_frecuency, 58
  interactive.relative_frecuency, 59

* ~geometrica
  explain.geometricMean, 18
  geometricMean_, 40
  interactive.geometricMean, 50

* ~geometric
  explain.geometricMean, 18
  geometricMean_, 40
  interactive.geometricMean, 50

* ~get
  getUserAction, 41

* ~harmonic
  explain.harmonicMean, 19
  harmonicMean_, 42
  interactive.harmonicMean, 51

* ~imagenes
  initImages, 43

* ~images
  initImages, 43

* ~iniciar
  initImages, 43

* ~init
  initImages, 43

* ~interactiva
  interactive.averageDeviation, 45
  interactive.binomial, 46
  interactive.chisquared, 47
  interactive.covariance, 48
  interactive.cv, 48
  interactive.fisher, 49
  interactive.geometricMean, 50
  interactive.harmonicMean, 51
  interactive.laplace, 52
  interactive.mean, 52
  interactive.median, 53
  interactive.mode, 54
  interactive.normal, 55
  interactive.poisson, 55
  interactive.poisson, 57
  interactive.standardDeviation, 60
  interactive.tstudent, 61
  interactive.variance, 62

* ~interactive
  interactive.absolute_acum_frecuency, 44
  interactive.absolute_frecuency, 44
  interactive.averageDeviation, 45
  interactive.binomial, 46
  interactive.chisquared, 47
  interactive.covariance, 48
  interactive.cv, 48
  interactive.fisher, 49
  interactive.geometricMean, 50
  interactive.harmonicMean, 51
  interactive.laplace, 52
  interactive.mean, 52
  interactive.median, 53
interactive.mode, 54
interactive.normal, 55
interactive.pearson, 55
interactive.percentile, 56
interactive.poisson, 57
interactive.quartile, 58
interactive.relative_acum_frequency, 58
interactive.relative_frequency, 59
interactive.standardDeviation, 60
interactive.tstudent, 61
interactive.variance, 62
* interactivo
interactive.geometricMean, 50
interactive.mean, 52
* laplace
explain.laplace, 20
interactive.laplace, 52
laplace_, 62
* meanC
meanC, 64
* mean
explain.geometricMean, 18
explain.harmonicMean, 19
explain.mean, 21
geometricMean_, 40
harmonicMean_, 42
interactive.geometricMean, 50
interactive.harmonicMean, 51
interactive.mean, 52
mean_, 65
* media.aritmetica
mean_, 65
meanC, 64
* media.explicada
explain.mean, 21
* media.interactiva
interactive.mean, 52
* mediaC
meanC, 64
* mediana.explicada
explain.median, 22
* mediana.interactiva
interactive.median, 53
* mediana
median_, 66
* median
explain.median, 22
interactive.median, 53
median_, 66
* media
explain.geometricMean, 18
explain.harmonicMean, 19
geometricMean_, 40
harmonicMean_, 42
interactive.geometricMean, 50
interactive.harmonicMean, 51
mean_, 65
* moda.explicada
explain.mode, 23
* moda.interactiva
interactive.mode, 54
* moda
mode_, 68
* mod
explain.mode, 23
interactive.mode, 54
mode_, 68
* normal
explain.normal, 24
interactive.normal, 55
normal_, 69
* pearson
explain.pearson, 25
pearson_, 70
* percentiles
interactive.percentile, 56
* percentile
explain.percentile, 26
interactive.percentile, 56
percentile_, 71
* percentil
interactive.percentile, 56
percentile_, 71
* pintar
drawVector, 9
* poisson
explain.poisson, 27
interactive.poisson, 57
poisson_, 72
* quantiles
interactive.quartile, 58
* quantile
quartile_, 73
* quartile
explain.quartile, 28
interactive.quartile, 58
 quartile_, 73
* ~regla
  explain.laplace, 20
 interactive.laplace, 52
 laplace_, 62
* ~relativa
  frecuency_relative, 37
 frecuency_relative_acum, 39
* ~relative
  explain.relative_acum_frequency, 29
 explain.relative_frequency, 30
 frecuency_relative, 37
 frecuency_relative_acum, 39
 interactive.relative_acum_frequency, 58
 interactive.relative_frequency, 59
* ~rule
  explain.laplace, 20
 interactive.laplace, 52
 laplace_, 62
* ~standar
  explain.standardDeviation, 31
 interactive.standardDeviation, 60
* ~standart
  standardDeviation_, 74
* ~tipica
  explain.standardDeviation, 31
 interactive.standardDeviation, 60
 standardDeviation_, 74
* ~tstudent
  explain.tstudent, 32
 fisher_, 34
 interactive.tstudent, 61
 tstudent_, 75
* ~usuario
  getUserAction, 41
* ~usario
  getUserAction, 41
* ~variaciion
  cv_, 8
 explain.cv, 16
 interactive.cv, 48
* ~varianza
  explain.variance, 33
 interactive.variance, 62
 variance_, 77
* ~variation
  cv_, 8
 explain.cv, 16
 interactive.cv, 48
* ~vector
  drawVector, 9
 averageDeviation_, 3
 binomial_, 4
 chisquared_, 6
 covariance_, 7
 cv_, 8
 drawVector, 9
 explain.absolute_acum_frequency, 10
 explain.absolute_frequency, 11
 explain.averageDeviation, 12
 explain.binomial, 13
 explain.chisquared, 14
 explain.covariance, 15
 explain.cv, 16
 explain.fisher, 17
 explain.geometricMean, 18
 explain.harmonicMean, 19
 explain.laplace, 20
 explain.mean, 21
 explain.median, 22
 explain.mode, 23
 explain.normal, 24
 explain.pearson, 25
 explain.percentile, 26
 explain.poisson, 27
 explain.quartile, 28
 explain.relative_acum_frequency, 29
 explain.relative_frequency, 30
 explain.standardDeviation, 31
 explain.tstudent, 32
 explain.variance, 33
 fisher_, 34
 frecuency_abs, 35
 frecuency_absolute_acum, 36
 frecuency_relative, 37
INDEX

frequency_relative_acum, 39  
harmonicMean_, 42  
initImages, 43  
interactive.absolute_acum_frecuency, 44  
interactive.absolute_frecuency, 44  
interactive.averageDeviation, 45  
interactive.binomial, 46  
interactive.chisquared, 47  
interactive.covariance, 48  
interactive.cv, 48  
interactive.fisher, 49  
interactive.geometricMean, 50  
interactive.harmonicMean, 51  
interactive.laplace, 52  
interactive.mean, 52  
interactive.median, 53  
interactive.mode, 54  
interactive.normal, 55  
interactive.pearson, 55  
interactive.percentile, 56  
interactive.poisson, 57  
interactive.quartile, 58  
interactive.relative_acum_frecuency, 58  
interactive.relative_frecuency, 59  
interactive.standardDeviation, 60  
interactive.tstudent, 61  
interactive.variance, 62  
laplace_, 62  
LearningRlab, 64  
LearningRlab-package (LearningRlab), 64  
mean_, 65  
meanC, 64  
median_, 66  
mode_, 68  

normal_, 69  

pearson_, 70  
percentile_, 71  
poisson_, 72  

quartile_, 73  
standardDeviation_, 74  
tstudent_, 75  

variance_, 77