

# Package ‘LearningRlab’

June 18, 2022

**Version** 2.3

**Date** 2022-6-17

**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@edu.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** 2022-06-17 23:40:09 UTC

**RoxygenNote** 7.1.1

**Encoding** UTF-8

## R topics documented:

averageDeviation_ . . . . .	3
binomial_ . . . . .	4
chisquared_ . . . . .	6

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.pearson . . . . .	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.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

<code>averageDeviation_</code>	3
<code>interactive.median</code>	53
<code>interactive.mode</code>	54
<code>interactive.normal</code>	55
<code>interactive.pearson</code>	55
<code>interactive.percentile</code>	56
<code>interactive.poisson</code>	57
<code>interactive.quartile</code>	58
<code>interactive.relative_acum_frecuency</code>	58
<code>interactive.relative_frecuency</code>	59
<code>interactive.standardDeviation</code>	60
<code>interactive.tstudent</code>	61
<code>interactive.variance</code>	62
<code>laplace_</code>	62
<code>LearningRlab</code>	64
<code>meanC</code>	64
<code>mean_</code>	65
<code>median_</code>	66
<code>mode_</code>	68
<code>normal_</code>	69
<code>pearson_</code>	70
<code>percentile_</code>	71
<code>poisson_</code>	72
<code>quartile_</code>	73
<code>standardDeviation_</code>	74
<code>tstudent_</code>	75
<code>variance_</code>	77
<b>Index</b>	<b>78</b>

---

`averageDeviation_`      *Average Absolute Deviation Function*

---

### 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 elemets. The average absolute deviation formule is the following:

$$\sigma = \frac{\sum_i^N |X_i - \bar{X}|}{N}$$

®LearningRlab

**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.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)
result = averageDeviation_(data)
```

---

 binomial\_

*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 formule is the following:

$$\mathbf{Binomial} \\ \mathbf{Distribution} = \frac{n!}{x!(n-x)!} p^x (1-p)^{(n-x)}$$

LearningRlab ®

**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)
```

chisquared\_

*Chisquared Distribution Calculus Function***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 formule is the following:

$$\mathbf{Chi-Squared} = \frac{\sum (f_e - f_o)^2}{f_e}$$

*LearningRlab* ®

**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.gomezc@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)
data2 = c(1,2,4,4,6,5,11,2,10,5,6,1)
chisquared_(data, data2)
```

---

covariance\_

*Covariance Calculus Function*

---

## Description

This function calculates the covariance of two vectors of numbers.

## Usage

```
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 formule is the following:

$$\mathbf{Covariance} = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{n}$$

*LearningRlab* ®

## 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.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)
data2 = c(1,2,4,4,4,6,5,11,2,10,5,6,1)
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 formule is the following:

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

LearningRlab ®

**Value**

Numeric, the coefficient of variation 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**

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

cv_(data)
```

---

drawVector

*Draw Vector Function*

---

**Description**

This function prints all the elements of a vector

**Usage**

```
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.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(1:12)  
  drawVector(data)  
}
```

---

explain.absolute\_acum\_frecuency

*Absolute Accumulated Frecuency Calculus Explained*

---

**Description**

Step by step demonstration of the absolute accumulated frecuency calculus

**Usage**

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

**Arguments**

v	Should be a vector
x	Should be a number

**Details**

To calculate the absolute accumulated frecuency, 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

**Examples**

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

---

explain.absolute\_frekuensi

*Absolute Frekuensi Calculus Explained*

---

**Description**

Step by step demonstration of the absolute frekuensi calculus

**Usage**

```
explain.absolute_frekuensi(v,x)
```

**Arguments**

v	Should be a vector
x	Should be a number

**Details**

To calculate the absolute frekuensi, the user should give a vector and a number. We can see the absolute frekuensi formule in the frekuensi\_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_frecuency(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.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(7,2,5,7,1,4,12)

explain.averageDeviation(data)
```

---

explain.binomial      *Binomial Distribution Function Explained*

---

**Description**

Step by step demonstration of the binomial distribution calculus.

**Usage**

```
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 n 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

## Examples

```
#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

```
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 see the chisquared distribution formula 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

**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

```
#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

```
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 saw the coefficient of variation formule 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**

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

explain.cv(data)
```

---

explain.fisher            *F Fisher Distribution Function Explained*

---

**Description**

Step by step demonstration of the fisher distribution calculus.

**Usage**

```
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 see fisher distribution formulae 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

## Examples

```
#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

```
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 saw the geometric mean formule 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.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(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 as the quotient between the number of favorable cases to A, and that of all possible results of the experiment. We can see the Laplace's rule correlation formula 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

**Examples**

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

explain.laplace(data, data2)
```

---

explain.mean

*Mean Function Explained*

---

**Description**

Step by step demonstration of the arithmetic mean calculus.

**Usage**

```
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 saw the arithmetic mean formule 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.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(1,2,2,5,10,4,2)  
  
  explain.mean(data)  
}
```

---

explain.median	<i>Median Function Explained</i>
----------------	----------------------------------

---

**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.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(1,2,2,5,10,4,2)  
  
  explain.median(data)  
}
```

---

explain.mode	<i>Mode Function Explained</i>
--------------	--------------------------------

---

**Description**

Step by step demonstration of the mode calculus.

**Usage**

```
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.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(1,1,2,5,2,3,1,4,1)

  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 see the normal distribution correlation formulae 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**

```
#data creation
x = 0.1

explain.normal(x)
```

---

explain.pearson	<i>Pearson Correlation Function Explained</i>
-----------------	---

---

**Description**

Step by step demonstration of the pearson correlation calculus.

**Usage**

```
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 see the pearson correlation formula 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**

```
#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**

```
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 formulae 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**

<code>k</code>	Should be a numbers
<code>lam</code>	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**

```
#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**

```
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

**Examples**

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

  explain.quartile(data)
}
```

---

explain.relative\_acum\_frecuency

*Relative Accumulated Frecuency Calculus Explained*

---

**Description**

Step by step demonstration of the relative accumulated frecuency calculus

**Usage**

```
explain.relative_acum_frecuency(v, x)
```

**Arguments**

v	Should be a vector
x	Should be a numebr of the vector

**Details**

To calculate the relative accumulated frecuency, the user should give a vector and a number. We can saw the relative accumulated frecuency formule in the frecuency\_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.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(1,2,2,5,10,4,2)
  value = 2
  #function execution
  explain.relative_acum_frecuency(data, value)
}
```

---

explain.relative\_frecuency

*Relative Frecuency Calculus Explained*

---

**Description**

Step by step demonstration of the relative frecuency calculus

**Usage**

```
explain.relative_frecuency(v,x)
```

**Arguments**

v	Should be a vector
x	Should be a number

**Details**

To calculate the relative frecuency, the user should give a vector and a number. We can saw the relative frecuency 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

**Examples**

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

---

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 saw the standard deviation formule 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.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(1,5,3,7,10,4,2)

explain.standardDeviation(data)
```

---

explain.tstudent      *T-Student Distribution Function Explained*

---

## Description

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

## Usage

```
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



**Examples**

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

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

---

explain.variance	<i>Variance Function Explained</i>
------------------	------------------------------------

---

**Description**

Step by step demonstration of the variance calculus.

**Usage**

```
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**

```
#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**

```
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 formule is the following:

$$\text{Fisher Distribution} = \frac{S_x^2}{S_w^2}$$

LearningRlab®

**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.gomezc@edu.uah.es>  
Juan Jose Cuadrado, <jjcg@uah.es>  
Universidad de Alcala de Henares

**Examples**

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

fisher_(x,y)
```

---

frecuency\_abs

*Absolute Frecuency Calculus*

---

**Description**

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

**Usage**

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

**Arguments**

v                    Should be a vector  
x                    Should be a number

**Details**

The absolute frecuency formula is the following:

**Absolute frequency =** number of aparitions of  
the examined element

*LearningRlab* ®

**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.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  
  frequency_abs(data, value)  
}
```

---

frequency\_absolute\_acum

*Accumulated Absolute Frequency 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**

```
frequency_absolute_acum(v, x)
```

**Arguments**

v	Should be a vector
x	Should be a number

**Details**

The accumulated absolute frequency formula is the following:

$$\text{Absolute accumulated frequency (X)} = \sum F_i \text{ where } i \leq X$$

*LearningRlab* ®

**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.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_absolute_acum(data, value)
}
```

---

frecuency\_relative      *Relative Frequency Calculus*

---

**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**

```
frequency_relative(v,x)
```

**Arguments**

v	Should be a vector
x	Should be a number

**Details**

The relative frequency formula is the following:

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

*LearningRlab* ®

**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  
  frequency_relative(data, value)  
}
```

---

`frecuency_relative_acum`*Accumulated Relative Frecuency Calculus*

---

**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 frequency formula is the following:

$$\text{Relative accumulated frequency (X)} = \sum f_i \text{ where } i \leq X$$

*LearningRlab* ®

**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
  frecency_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 formule is the following:

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

©LearningRlab

**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



**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)
```

---

<code>getUserAction</code>	<i>Get User Action Funcion</i>
----------------------------	--------------------------------

---

**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)  
}
```

---

harmonicMean\_      *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 formule is the following:

$$\text{Harmonic Mean} = \frac{N}{\frac{1}{X_1} + \frac{1}{X_2} + \dots + \frac{1}{X_N}}$$

*LearningRlab* ®

### 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**

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

harmonicMean_(data)
```

---

initImages	<i>Init Images Function</i>
------------	-----------------------------

---

**Description**

This function is used to display an image.

**Usage**

```
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**

```
{

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

  ## End(Not run)
}
```

---

`interactive.absolute_acum_frecuency`*User Interactive Absolute Accumulated Frecuency Calculus*

---

**Description**

Interactive function for absolute accumulated frecuency 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 frecuency

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

---

`interactive.absolute_frecuency`*User Interactive Absolute Frecuency Calculus*

---

**Description**

Interactive function for absolute frecuency calculus.

**Usage**

```
interactive.absolute_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 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 Alcala de Henares

**Examples**

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

---

interactive.averageDeviation

*User Interactive Average Absolute Deviation Calculus*

---

**Description**

Interactive function for average absolute deviation calculus.

**Usage**

```
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 average absolute deviation formule, apart from the averageDeviation\_ help document.

**Value**

An interactive process to calculate the average absolute deviation

**Author(s)**

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

**References**

[https://en.wikipedia.org/wiki/Average\\_absolute\\_deviation](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\_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

**Examples**

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

---

```
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 formule, 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.chisquared()  
  
## 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 formule, 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**

```
## 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()
```



**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 formule, 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 Alcala de Henares

**Examples**

```
## 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 formule, apart from the fisher\_ help document.

**Value**

An interactive process to calculate the fisher 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.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 formule, apart from the `geometricMean_ help` document.

**Value**

An interactive process to calculate the geometric mean.

**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:  
interactive.geometricMean()  
  
## End(Not run)
```

---

```
interactive.harmonicMean  
                                  User Interactive Harmonic Mean Calculus
```

---

**Description**

Interactive function for harmonic mean calculus.

**Usage**

```
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 formule, 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**

```
## 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 datas. The function itself will provide the Laplace's rule formule, 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**

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

---

interactive.mean     *User Interactive Mean Calculus*

---

**Description**

Interactive function for arithmetic mean calculus.

**Usage**

```
interactive.mean()
```

**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 formule, apart from the mean\_ help document.

**Value**

An interactive process to calculate the arithmetic mean.

**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:  
  interactive.mean()  
  
  ## End(Not run)  
}
```

---

interactive.median      *User Interactive Median Calculus*

---

**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 formule, apart from the median\_ help document.

**Value**

An interactive process to calculate the median

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

---

interactive.mode	<i>User Interactive Mode Calculus</i>
------------------	---------------------------------------

---

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

---

interactive.normal      *User Interactive Normal Distribution Calculus*

---

**Description**

Interactive function for normal distribution calculus.

**Usage**

```
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 formule, apart from the normal\_ help document.

**Value**

An interactive process to calculate the normal distribution.

**Author(s)**

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

**Examples**

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

---

interactive.pearson      *User Interactive Pearson Correlation Calculus*

---

**Description**

Interactive function for pearson correlation calculus.

**Usage**

```
interactive.pearson()
```

**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 formule, 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, <jjcg@uah.es>  
Universidad de Alcala de Henares

**Examples**

```
## 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



**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:  
interactive.percentile()  
  
## End(Not run)
```

---

interactive.poisson     *User Interactive Poisson Distribution Calculus*

---

**Description**

Interactive function for Poisson distribution calculus.

**Usage**

```
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 datas. The function itself will provide the Poisson distribution formule, 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**

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

---

`interactive.quartile`    *User Interactive Quartiles Calculus*

---

**Description**

Interactive function for quartiles calculus.

**Usage**

```
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**

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

---

`interactive.relative_acum_frecuency`  
*User Interactive Relative Accumulated Frecuency Calculus*

---

**Description**

Interactive function for relative accumulated frecuency calculus.

**Usage**

```
interactive.relative_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 relative accumulated frecuency

**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:  
interactive.relative_acum_frecuency()  
  
## End(Not run)
```

---

```
interactive.relative_frecuency
```

*User Interactive Relative Frecuency Calculus*

---

**Description**

Interactive function for relative frecuency calculus.

**Usage**

```
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 frecuency

**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:  
interactive.relative_frecuency()  
  
## End(Not run)
```

---

interactive.standardDeviation  
*User Interactive Standard Deviation Calculus*

---

**Description**

Interactive function for standard deviation calculus.

**Usage**

```
interactive.standardDeviation()
```

**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 formule, apart from the standardDeviation\_help document.

**Value**

An interactive process to calculate the standard deviation

**Author(s)**

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

**References**

[https://en.wikipedia.org/wiki/Standard\\_deviation](https://en.wikipedia.org/wiki/Standard_deviation)

**Examples**

```
## 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 datas. The function itself will provide the T-Students distribution formule, 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 Alcala de Henares

**Examples**

```
## 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 formule, apart from the variance\_ help document.

**Value**

An interactive process to calculate the average absolute deviation

**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:  
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 formule is the following:

$$\text{LaPlace Rule} = \frac{\text{casos favorables}}{\text{casos posibles}}$$

LearningRlab ®

**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**

```
#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.

Package: LearningRlab  
Type: Package  
Version: 2.2  
Date: 2021-1-17  
License: Unlimited

### 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:

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

@LearningRlab

**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**

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

---

mean\_

*Mean Calculus Function*

---

**Description**

This function calculates the arithmetic mean of a numbers vector.

**Usage**

`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 formulè is the following:

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

@LearningRlab

**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 Alcalá de Henares

**Examples**

```
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 formule is the following:

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

© LearningRlab

**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.monhemimius@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\_

*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 formule is the following:

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

©LearningRlab

**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.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(1,2,2,3,4)

  mode_(data)
```

}

---

`normal_`*Normal Distribution Calculus Function*

---

**Description**

This function calculates the normal distribution of experiment.

**Usage**`normal_(x)`**Arguments**

`x` Should be a numbers.

**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:

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

LearningFlab®

**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

```
#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 formule is the following:

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

*LearningRlab* ®

## 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

**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(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)
```

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,  $0 \Rightarrow y \leq 1$

**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:

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

*LearningRlab* ®

**Value**

A vector sorted with the elements divided by 100 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.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 formule is the following:

$$\text{Poisson Distribution} = \frac{\lambda^x e^{-\lambda}}{X!}$$

LearningRlab ®

### Value

Numeric, the pearson correlation of two numbers.

### 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

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

---

quartile\_

*Quartiles Calculus*

---

### Description

Calculates the 3 Quartiles of a vector of data

### Usage

```
quartile_(x)
```

### Arguments

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:

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

LearningRlab ®

**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.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)

  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 elemets. The standard deviation formule is the following:

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

©LearningRlab

**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**

```
#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:

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

LearningRlab ®

**Value**

Numeric, the T-Student distribution.

**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

**Examples**

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

tstudent_(x,u,s,n)
```

---

**variance\_** *Variance Calculus Function*

---

**Description**

This function calculates the variance of a numbers vector.

**Usage**

```
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 expectation of the squared deviation of all numbers vector from its mean. The variance formule is the following:

$$\mathbf{Variance} = \frac{\sum (X - \bar{X})^2}{N - 1}$$

LearningRlab ®

**Value**

Numeric, the variance 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**

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

variance_(data)
```

# Index

- \* **~absoluta**
  - averageDeviation\_, 3
  - explain.averageDeviation, 12
  - frequency\_abs, 35
  - frequency\_absolute\_acum, 36
  - interactive.averageDeviation, 45
- \* **~absolute**
  - averageDeviation\_, 3
  - explain.absolute\_acum\_frequency, 10
  - explain.absolute\_frequency, 11
  - explain.averageDeviation, 12
  - frequency\_abs, 35
  - frequency\_absolute\_acum, 36
  - interactive.absolute\_acum\_frequency, 44
  - interactive.absolute\_frequency, 44
  - interactive.averageDeviation, 45
- \* **~accumulated**
  - explain.absolute\_acum\_frequency, 10
  - explain.relative\_acum\_frequency, 29
  - frequency\_absolute\_acum, 36
  - frequency\_relative\_acum, 39
  - interactive.absolute\_acum\_frequency, 44
  - interactive.relative\_acum\_frequency, 58
- \* **~acumulada**
  - explain.absolute\_acum\_frequency, 10
  - explain.relative\_acum\_frequency, 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
- \* **~chicadrado**
  - chisquared\_, 6
  - explain.chisquared, 14
  - interactive.chisquared, 47
- \* **~chisquared\_**
  - chisquared\_, 6
  - explain.chisquared, 14
- \* **~chisquared**
  - interactive.chisquared, 47
- \* **~coeficiente**
  - cv\_, 8
  - explain.cv, 16
  - interactive.cv, 48
- \* **~coeficiente**
  - cv\_, 8
  - explain.cv, 16
  - interactive.cv, 48
- \* **~coger**
  - getUserAction, 41
- \* **~correlacion**
  - explain.pearson, 25
  - pearson\_, 70
- \* **~correlacion**
  - explain.pearson, 25
  - pearson\_, 70
- \* **~covaiance**
  - 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
- \* **~distribution**
  - 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
- \* **~draw**
  - drawVector, 9
- \* **~explained**
  - explain.averageDeviation, 12
  - explain.standardDeviation, 31
- \* **~explain**
  - explain.absolute\_acum\_frecuency, 10
  - explain.absolute\_frecuency, 11
  - 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\_frecuency, 29
  - explain.relative\_frecuency, 30
  - explain.tstudent, 32
  - explain.variance, 33
- \* **~explicada**
  - 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.poisson, 27
- 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\_frequency, 44
  - interactive.relative\_frequency, 59
- \* **~frequency**
  - explain.absolute\_acum\_frequency, 10
  - explain.absolute\_frequency, 11
  - explain.relative\_acum\_frequency, 29
  - explain.relative\_frequency, 30
  - frequency\_abs, 35
  - frequency\_absolute\_acum, 36
  - frequency\_relative, 37
  - frequency\_relative\_acum, 39
  - interactive.absolute\_acum\_frequency, 44
  - interactive.absolute\_frequency, 44
  - interactive.relative\_acum\_frequency, 58
  - interactive.relative\_frequency, 59
- \* **~geometrica**
  - explain.geometricMean, 18
  - geometricMean\_, 40
  - interactive.geometricMean, 50
- \* **~geometric**
  - explain.geometricMean, 18
  - geometricMean\_, 40
  - interactive.geometricMean, 50
- 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.harmonicMean, 51
  - interactive.laplace, 52
  - interactive.median, 53
  - interactive.mode, 54
  - interactive.normal, 55
  - interactive.pearson, 55
  - interactive.poisson, 57
  - interactive.standardDeviation, 60
  - interactive.tstudent, 61
  - interactive.variance, 62
- \* **~interactive**
  - interactive.absolute\_acum\_frequency, 44
  - interactive.absolute\_frequency, 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
- \* **~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
- \* **~mode**
  - 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\_frecuency, 29
  - explain.relative\_frecuency, 30
  - frecuency\_relative, 37
  - frecuency\_relative\_acum, 39
  - interactive.relative\_acum\_frecuency, 58
  - interactive.relative\_frecuency, 59
- \* **~rule**
  - explain.laplace, 20
  - interactive.laplace, 52
  - laplace\_, 62
- \* **~standard**
  - 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
- \* **~user**
  - getUserAction, 41
- \* **~usuario**
  - getUserAction, 41
- \* **~variacion**
  - cv\_, 8
  - explain.cv, 16
  - interactive.cv, 48
- \* **~variance**
  - explain.variance, 33
  - interactive.variance, 62
  - variance\_, 77
- \* **~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\_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.pearson, 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

frequency\_relative\_acum, [39](#)

geometricMean\_, [40](#)  
getUserAction, [41](#)

harmonicMean\_, [42](#)  
harmonicMean\_ (harmonicMean\_), [42](#)

initImages, [43](#)  
interactive.absolute\_acum\_frequency,  
[44](#)  
interactive.absolute\_frequency, [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](#)

laplace\_, [62](#)  
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](#)