

Package: UAHDataScienceSF (via r-universe)

February 17, 2025

Type Package

Version 1.0.0

Title Interactive Statistical Learning Functions

Description An educational toolkit for learning statistical concepts through interactive exploration. Provides functions for basic statistics (mean, variance, etc.) and probability distributions with step-by-step explanations and interactive learning modes. Each function can be used for simple calculations, detailed learning with explanations, or interactive practice with feedback.

License MIT + file LICENSE

Imports stats, magick (>= 2.7.3), crayon (>= 1.3.4)

Suggests knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Config/pak/sysreqs libmagick++-dev gsfontr libssl-dev

Repository <https://andriyprotsak5.r-universe.dev>

RemoteUrl <https://github.com/andriyprotsak5/UAHDataScienceSF>

RemoteRef HEAD

RemoteSha 13007d1e3a15292d31ac5d8a6396a61055e188dd

Contents

absolute_acum_frequency	2
absolute_frequency	3
average_deviation	4
binomial_	5

chisquared	6
covariance	7
cv	8
drawVector	8
fisher	9
geometric_mean	10
getUserAction	11
harmonic_mean	11
initImages	12
laplace	12
mean_	13
median_	14
mode_	15
normal	15
pearson	16
percentile	17
poisson_	18
quartile	19
relative_acum_frequency	19
relative_frequency	20
standard_deviation	21
tstudent	22
variance	23

Index 24

absolute_acum_frequency
Absolute Accumulated Frequency Function

Description

This function calculates the absolute accumulated frequency of a value in a numeric vector.

Usage

```
absolute_acum_frequency(v = NULL, x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

v	Optional numeric vector (not needed for interactive mode)
x	Optional numeric value to count (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The absolute accumulated frequency of x in v (for non-interactive mode)

Examples

```
data <- c(1,4,3,3,2,5,7,12,1,2,3,12)
value <- 12

# Simple calculation
absolute_acum_frequency(data, value)

# Learning mode
absolute_acum_frequency(data, value, learn = TRUE)

# Interactive mode
if(interactive()){
  absolute_acum_frequency(interactive = TRUE)
}
```

absolute_frequency *Absolute Frequency Function*

Description

This function calculates the absolute frequency of a value in a numeric vector.

Usage

```
absolute_frequency(v = NULL, x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

v	Optional numeric vector (not needed for interactive mode)
x	Optional numeric value to count (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The absolute frequency of x in v (for non-interactive mode)

Examples

```
data <- c(1,4,3,3,2,5,7,12,1,2,3,12)
value <- 12

# Simple calculation
absolute_frequency(data, value)

# Learning mode
absolute_frequency(data, value, learn = TRUE)
```

```
# Interactive mode
if(interactive()){
  absolute_frequency(interactive = TRUE)
}
```

average_deviation *Average Absolute Deviation Function*

Description

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

Usage

```
average_deviation(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The average absolute deviation of the vector (for non-interactive mode)

Examples

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

# Simple calculation
average_deviation(data)

# Learning mode
average_deviation(data, learn = TRUE)

# Interactive mode
if(interactive()){
  average_deviation(interactive = TRUE)
}
```

`binomial_`*Binomial Distribution Function*

Description

This function calculates the binomial distribution probability.

Usage

```
binomial_(n = NULL, x = NULL, p = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

<code>n</code>	Optional number of trials (not needed for interactive mode)
<code>x</code>	Optional number of successes (not needed for interactive mode)
<code>p</code>	Optional probability of success (not needed for interactive mode)
<code>learn</code>	Logical, if TRUE shows step-by-step explanation
<code>interactive</code>	Logical, if TRUE enables interactive practice mode

Value

The binomial probability (for non-interactive mode)

Examples

```
n <- 3
x <- 2
p <- 0.7

# Simple calculation
binomial_(n, x, p)

# Learning mode
binomial_(n, x, p, learn = TRUE)

# Interactive mode
if(interactive()){
  binomial_(interactive = TRUE)
}
```

chisquared	<i>Chi-squared Distribution Function</i>
------------	--

Description

This function calculates the chi-squared statistic between two groups.

Usage

```
chisquared(x = NULL, y = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional first vector (not needed for interactive mode)
y	Optional second vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The chi-squared statistic (for non-interactive mode)

Examples

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

# Simple calculation
chisquared(x, y)

# Learning mode
chisquared(x, y, learn = TRUE)

# Interactive mode
if(interactive()){
  chisquared(interactive = TRUE)
}
```

covariance	<i>Covariance Function</i>
------------	----------------------------

Description

This function calculates the covariance between two vectors of numbers.

Usage

```
covariance(x = NULL, y = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional first numeric vector (not needed for interactive mode)
y	Optional second numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The covariance between the two vectors (for non-interactive mode)

Examples

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

# Simple calculation
covariance(data, data2)

# Learning mode
covariance(data, data2, learn = TRUE)

# Interactive mode
if(interactive()){
  covariance(interactive = TRUE)
}
```

cv *Coefficient of Variation Function*

Description

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

Usage

```
cv(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The coefficient of variation of the vector (for non-interactive mode)

Examples

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

# Simple calculation
cv(data)

# Learning mode
cv(data, learn = TRUE)

# Interactive mode
if(interactive()){
  cv(interactive = TRUE)
}
```

drawVector *Draw Vector Function*

Description

This function prints all the elements of a vector

Usage

```
drawVector(buffer)
```


Arguments

buffer A vector of elements to be printed

Value

No return value, prints to screen

Examples

```
## Not run:  
data <- c(1:12)  
drawVector(data)  
  
## End(Not run)
```

fisher

Fisher's F Distribution Function

Description

This function calculates the F statistic between two groups.

Usage

```
fisher(x = NULL, y = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x Optional first vector (not needed for interactive mode)
y Optional second vector (not needed for interactive mode)
learn Logical, if TRUE shows step-by-step explanation
interactive Logical, if TRUE enables interactive practice mode

Value

The F statistic (for non-interactive mode)

Examples

```
x <- c(70,75,74,72,68,59)  
y <- c(74,77,70,80,72,76)  
  
# Simple calculation  
fisher(x, y)  
  
# Learning mode  
fisher(x, y, learn = TRUE)
```

```
# Interactive mode
if(interactive()){
  fisher(interactive = TRUE)
}
```

geometric_mean

Geometric Mean Function

Description

This function calculates the geometric mean of a numeric vector. Can be used in three modes: simple calculation, learning mode with step-by-step explanation, or interactive mode for practice.

Usage

```
geometric_mean(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The geometric mean of the vector (for non-interactive mode)

Examples

```
data <- c(5,21,12,7,3,9,1)
# Simple calculation
geometric_mean(data)

# Learning mode
geometric_mean(data, learn = TRUE)

# Interactive mode
if(interactive()){
  geometric_mean(interactive = TRUE)
}
```

getUserAction	<i>Get User Action Function</i>
---------------	---------------------------------

Description

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

Usage

```
getUserAction()
```

Value

A vector entered by the user

Examples

```
## Not run:  
vector <- getUserAction()  
  
## End(Not run)
```

harmonic_mean	<i>Harmonic Mean Function</i>
---------------	-------------------------------

Description

This function calculates the harmonic mean of a numbers vector.

Usage

```
harmonic_mean(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The harmonic mean of the vector (for non-interactive mode)

Examples

```
data <- c(1,4,3,3,2,5,7,12,1,2,3,12)

# Simple calculation
harmonic_mean(data)

# Learning mode
harmonic_mean(data, learn = TRUE)

# Interactive mode
if(interactive()){
  harmonic_mean(interactive = TRUE)
}
```

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

Description

This function displays an image from the package resources.

Usage

```
initImages(image_name)
```

Arguments

image_name Name of the image file in inst/images

Value

No return value

laplace	<i>Laplace's Rule Function</i>
---------	--------------------------------

Description

This function calculates Laplace's Rule for a probability experiment.

Usage

```
laplace(x = NULL, y = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional first vector (not needed for interactive mode)
y	Optional second vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The probability according to Laplace's Rule (for non-interactive mode)

Examples

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

# Simple calculation
laplace(data, data2)

# Learning mode
laplace(data, data2, learn = TRUE)

# Interactive mode
if(interactive()){
  laplace(interactive = TRUE)
}
```

 mean_

Statistical Mean Function

Description

This function calculates the arithmetic mean of a numeric vector. Can be used in three modes: simple calculation, learning mode with step-by-step explanation, or interactive mode for practice.

Usage

```
mean_(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The arithmetic mean of the vector

Examples

```
# Simple calculation
data <- c(1,2,2,5,10,4,2)
mean_(data)

# Learning mode
mean_(data, learn = TRUE)

# Interactive mode
if(interactive()){
  mean_(interactive = TRUE)
}
```

median_

Median Function

Description

This function calculates the median of a numbers vector.

Usage

```
median_(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The median of the vector (for non-interactive mode)

Examples

```
data <- c(1,3,2,5,12,4,4,2,9)

# Simple calculation
median_(data)

# Learning mode
median_(data, learn = TRUE)

# Interactive mode
if(interactive()){
  median_(interactive = TRUE)
}
```

mode_	<i>Mode Function</i>
-------	----------------------

Description

This function calculates the mode of a numbers vector.

Usage

```
mode_(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The mode of the vector (for non-interactive mode)

Examples

```
data <- c(5,21,12,7,3,9,1)

# Simple calculation
mode_(data)

# Learning mode
mode_(data, learn = TRUE)

# Interactive mode
if(interactive()){
  mode_(interactive = TRUE)
}
```

normal	<i>Normal Distribution Function</i>
--------	-------------------------------------

Description

This function calculates the normal distribution probability density.

Usage

```
normal(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric value (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The normal probability density (for non-interactive mode)

Examples

```
x <- 0.1

# Simple calculation
normal(x)

# Learning mode
normal(x, learn = TRUE)

# Interactive mode
if(interactive()){
  normal(interactive = TRUE)
}
```

pearson

Pearson Correlation Function

Description

This function calculates the Pearson correlation coefficient between two vectors of numbers.

Usage

```
pearson(x = NULL, y = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional first numeric vector (not needed for interactive mode)
y	Optional second numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The Pearson correlation coefficient between the two vectors (for non-interactive mode)

Examples

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

# Simple calculation
pearson(data, data2)

# Learning mode
pearson(data, data2, learn = TRUE)

# Interactive mode
if(interactive()){
  pearson(interactive = TRUE)
}
```

percentile

Percentile Function

Description

This function calculates the percentiles of a numeric vector.

Usage

```
percentile(x = NULL, p = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
p	Numeric value between 0 and 1 for percentile calculation (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The percentile value (for non-interactive mode)

Examples

```
data <- c(1,4,3,3,2,5,7,12)

# Simple calculation
percentile(data, 0.3)

percentile(data, 0.3, learn = TRUE)
```

```
if(interactive()){  
  percentile(interactive = TRUE)  
}
```

poisson_

Poisson Distribution Function

Description

This function calculates the Poisson distribution probability.

Usage

```
poisson_(k = NULL, lam = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

k	Optional number of occurrences (not needed for interactive mode)
lam	Optional expected value lambda (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The Poisson probability (for non-interactive mode)

Examples

```
lam <- 2  
k <- 3  
  
# Simple calculation  
poisson_(k, lam)  
  
# Learning mode  
poisson_(k, lam, learn = TRUE)  
  
# Interactive mode  
if(interactive()){  
  poisson_(interactive = TRUE)  
}
```

`quartile`*Quartiles Function*

Description

This function calculates the quartiles of a numeric vector.

Usage

```
quartile(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

<code>x</code>	Optional numeric vector (not needed for interactive mode)
<code>learn</code>	Logical, if TRUE shows step-by-step explanation
<code>interactive</code>	Logical, if TRUE enables interactive practice mode

Value

The quartiles of the vector (for non-interactive mode)

Examples

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

# Simple calculation
quartile(data)

# Learning mode
quartile(data, learn = TRUE)

# Interactive mode
if(interactive()){
  quartile(interactive = TRUE)
}
```

`relative_acum_frequency`*Relative Accumulated Frequency Function*

Description

This function calculates the relative accumulated frequency of a value in a numeric vector.

Usage

```
relative_acum_frequency(v = NULL, x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

v	Optional numeric vector (not needed for interactive mode)
x	Optional numeric value to count (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The relative accumulated frequency of x in v (for non-interactive mode)

Examples

```
data <- c(1,4,3,3,2,5,7,12,1,2,3,12)
value <- 12

# Simple calculation
relative_acum_frequency(data, value)

# Learning mode
relative_acum_frequency(data, value, learn = TRUE)

# Interactive mode
if(interactive()){
  relative_acum_frequency(interactive = TRUE)
}
```

relative_frequency *Relative Frequency Function*

Description

This function calculates the relative frequency of a value in a numeric vector.

Usage

```
relative_frequency(v = NULL, x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

v	Optional numeric vector (not needed for interactive mode)
x	Optional numeric value to count (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The relative frequency of x in v (for non-interactive mode)

Examples

```
data <- c(1,4,3,3,2,5,7,12,1,2,3,12)
value <- 12

# Simple calculation
relative_frequency(data, value)

# Learning mode
relative_frequency(data, value, learn = TRUE)

# Interactive mode
if(interactive()){
  relative_frequency(interactive = TRUE)
}
```

standard_deviation *Standard Deviation Function*

Description

This function calculates the standard deviation of a numbers vector.

Usage

```
standard_deviation(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The standard deviation of the vector (for non-interactive mode)

Examples

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

# Simple calculation
standard_deviation(data)
```

```
# Learning mode
standard_deviation(data, learn = TRUE)

# Interactive mode
if(interactive()){
  standard_deviation(interactive = TRUE)
}
```

tstudent

Student's t Distribution Function

Description

This function calculates the t-statistic for sample data.

Usage

```
tstudent(
  x = NULL,
  u = NULL,
  s = NULL,
  n = NULL,
  learn = FALSE,
  interactive = FALSE
)
```

Arguments

x	Optional sample mean (not needed for interactive mode)
u	Optional population mean (not needed for interactive mode)
s	Optional standard deviation (not needed for interactive mode)
n	Optional sample size (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The t-statistic (for non-interactive mode)

Examples

```
x <- 52.9
u <- 50
s <- 3
n <- 10
```

```
# Simple calculation
tstudent(x, u, s, n)

# Learning mode
tstudent(x, u, s, n, learn = TRUE)

# Interactive mode
if(interactive()){
  tstudent(interactive = TRUE)
}
```

variance

Variance Function

Description

This function calculates the variance of a numbers vector.

Usage

```
variance(x = NULL, learn = FALSE, interactive = FALSE)
```

Arguments

x	Optional numeric vector (not needed for interactive mode)
learn	Logical, if TRUE shows step-by-step explanation
interactive	Logical, if TRUE enables interactive practice mode

Value

The variance of the vector (for non-interactive mode)

Examples

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

# Simple calculation
variance(data)

# Learning mode
variance(data, learn = TRUE)

# Interactive mode
if(interactive()){
  variance(interactive = TRUE)
}
```

Index

absolute_acum_frequency, [2](#)
absolute_frequency, [3](#)
average_deviation, [4](#)

binomial_, [5](#)

chisquared, [6](#)
covariance, [7](#)
cv, [8](#)

drawVector, [8](#)

fisher, [9](#)

geometric_mean, [10](#)
getUserAction, [11](#)

harmonic_mean, [11](#)

initImages, [12](#)

laplace, [12](#)

mean_, [13](#)
median_, [14](#)
mode_, [15](#)

normal, [15](#)

pearson, [16](#)
percentile, [17](#)
poisson_, [18](#)

quartile, [19](#)

relative_acum_frequency, [19](#)
relative_frequency, [20](#)

standard_deviation, [21](#)

tstudent, [22](#)

variance, [23](#)