

Package: statistics (via r-universe)

October 21, 2026

Version 0.14.1
Title Miscellanea Functions For Intro Stats Classes
Depends R (>= 3.5.0)
License GPL (>= 3)
Imports admisc (>= 0.25), declared (>= 0.26.11)
Description Utilities for introductory statistics labs: plot tail or interval areas for the normal or t distributions, highlight binomial probabilities, and summarise discrete random variables. Calculate pooled standard deviations and Welch-Satterthwaite degrees of freedom, and run teaching-friendly tests such as Fligner-Killeen plus one-way ANOVA, and t tests with an Ansari-Bradley variance check alongside pooled and Welch variants.
Repository <https://dusadrian.r-universe.dev>
Date/Publication 2026-08-10 04:28:29 UTC
RemoteUrl <https://github.com/dusadrian/statistics>
RemoteRef HEAD
RemoteSha b8519a81e38181a17a9160f8fb3fc8d87512b766

Contents

About the statistics package	2
anovahv	2
daria	5
dariat	6
dbinoms	7
dfcalc and spooled	9
histc	10
mbinom and sbinom	11
t.testhv	11
Index	15

About the statistics package

Utility functions for an intro stats class.

Description

Plot tail or interval areas for the normal or t distributions, and draw binomial distributions with highlighted outcome probabilities. Summarise discrete random variables via their mean and standard deviation. Compute pooled standard deviations and Welch-Satterthwaite degrees of freedom, then run teaching-friendly tests such as a Fligner-Killeen variance check plus one-way ANOVA, and t tests that pair an Ansari-Bradley variance check with both pooled and Welch variants.

Details

Package: statistics
 Type: Package
 Version: 0.14.1
 Date: 2026-08-10
 License: GPL-v3

Author(s)

Adrian Dusa

Maintainer: Adrian Dusa (dusa.adrian@unibuc.ro)

See Also

dnorm, pnorm, dbinom

anovahv

ANOVA with a homogeneity of variances check

Description

Performs one-way ANOVA together with a Fligner-Killeen test for the homogeneity of variances. When the group variances are not considered homogeneous, the print method displays a Welch approximation table.

Usage

```

anovahv(x, ...)

## Default S3 method:
anovahv(
  x, y, var.equal = NULL, conf.level = 0.95, numsum = FALSE, ...
)

## S3 method for class 'formula'
anovahv(
  formula, data, subset, na.action = na.omit,
  var.equal = NULL, conf.level = 0.95, numsum = FALSE,
  ...
)

```

Arguments

x	A numeric response vector.
y	A grouping variable. Character variables and categorical variables of class "declared" are coerced to factors. Other non-factor grouping variables are also treated as factors.
formula	A formula of the form values ~ group.
data	An optional data frame, matrix or list containing the variables in formula.
subset	An optional vector specifying a subset of observations.
na.action	A function specifying how missing values should be handled by the formula method.
var.equal	Logical or NULL. If TRUE, the classical ANOVA table is printed. If FALSE, the Welch approximation table is printed. If NULL, the default, the Fligner-Killeen test is used to decide which table to print.
conf.level	Numeric value between 0 and 1. When var.equal = NULL, the homogeneity test is evaluated against 1 - conf.level.
numsum	Logical. If TRUE, calculate and print the mean, standard deviation, number of valid values and number of missing values for each category of the grouping variable.
...	Additional arguments passed to methods.

Details

The function is a teaching-oriented wrapper around [aov](#), [lm](#), [oneway.test](#) and [fligner.test](#). It always stores both the classical ANOVA result and the Welch approximation result. The print method chooses which one to display.

The argument `var.equal` follows the same general idea as in [t.testhv](#). If it is left as NULL, the function performs a Fligner-Killeen homogeneity of variances test and prints the classical ANOVA table when the variances are considered homogeneous. If `var.equal = TRUE` or `var.equal = FALSE`, the user explicitly chooses the classical ANOVA or Welch output, respectively.

Grouping variables of class "declared" are detected and coerced with `as.factor()` before fitting the models. This allows declared categorical variables to be used directly as group variables in both the default and formula interfaces.

Value

An object inheriting from classes "anovahv", "aov" and "lm". The underlying object is the classical ANOVA model, allowing standard model generics such as `summary()`, `TukeyHSD()`, `predict()`, `residuals()` and `fitted()` to dispatch to their existing "aov" or "lm" methods. For compatibility, it also contains:

- `homog_test`: the Fligner-Killeen homogeneity of variances test;
- `test`: the classical ANOVA model produced by `aov`;
- `welch`: the Welch one-way test produced by `oneway.test`;
- `output_table`: a compact Welch approximation table;
- `numsum`: the grouped numerical summaries requested with `numsum = TRUE`, or `NULL`;
- `var.equal` and `conf.level`: the analysis settings.

Author(s)

Adrian Dusa

See Also

[aov](#), [anova](#), [oneway.test](#), [fligner.test](#), [t.testhv](#)

Examples

```
values <- c(
  15, 8, 17, 7, 26, 12, 8, 11, 16, 9, 16,
  24, 20, 19, 9, 17, 11, 8, 15, 6, 14
)
groups <- rep(1:3, each = 7)

anovahv(values, groups)
anovahv(values ~ groups)

# Include numerical summaries for every group
anovahv(values, groups, numsum = TRUE)

# Force a specific output, bypassing the variance decision
anovahv(values, groups, var.equal = TRUE)
anovahv(values, groups, var.equal = FALSE)

# Using a dataset
vgdf <- data.frame(values, groups)

anovahv(values ~ groups, data = vgdf)
```

```
using(
  vgd,
  anovahv(values ~ groups)
)

# Declared categorical grouping variables are detected
vgdf$declared_groups <- declared(
  vgd$groups,
  labels = c(A = 1, B = 2, C = 3)
)

anovahv(values ~ declared_groups, data = vgd)
anovahv(vgd$values, vgd$declared_groups)

# Class example
cls <- data.frame(
  values = c(
    22, 27, 32, 30, 29, 27, 33, 24, 24, 30,
    28, 22, 24, 18, 21, 26, 25, 20, 24, 28,
    20, 28, 31, 26, 26, 30, 21, 25, 29, 27
  ),
  groups = rep(1:3, each = 10)
)

using(
  cls,
  anovahv(values ~ groups)
)

result <- anovahv(values ~ groups, data = cls)
TukeyHSD(result)

# Post-hoc test, for example Bonferroni
using(
  cls,
  pairwise.t.test(values, groups, p.adj = "bonferroni")
)
```

daria*Calculate and draw the area under the normal curve z*

Description

The function "daria" - 'd'raws the 'area' under the normal curve for certain values of z.

Usage

```
daria(area, z1, z2, draw = FALSE)
```

Arguments

area	The required area
z1	First z value, in the interval +/- 4
z2	Second z value, in the interval +/- 4
draw	Logical; if TRUE, draw the area

Details

In the argument area, the function accepts:

"l", "u", "left" and "under" for the area to the left of z,

"r", "o", "a", "right" "over" and "above" for the area to the right of z

"b" and "between" for the area between two z values.

z values smaller than -4 and greater than +4 are truncated to these values.

Author(s)

Adrian Dusa

See Also

[pnorm](#), [qnorm](#)

Examples

```
daria("between", -1.96, 1.96)
```

```
daria("over", -1)
```

```
daria("under", -1)
```

```
daria("over", 2, draw = TRUE)
```

dariat

Calculate and draw the area under the t distribution

Description

A function similar to `daria`, with the only difference it uses the `t` instead of the normal `z` distribution. In addition, the function expects an additional parameter for the degrees of freedom.

Usage

```
dariat(area, t1, t2, df, draw = FALSE)
```

Arguments

area	The required area
t1	First t value, in the interval +/- 4
t2	Second t value, in the interval +/- 4
df	Degrees of freedom
draw	Logical; if TRUE, draw the area

Details

In the argument area, the function accepts:

"l", "u", "left" and "under" for the area to the left of z,

"r", "o", "a", "right" "over" and "above" for the area to the right of z

"b" and "between" for the area between two z values.

z values smaller than -4 and greater than +4 are truncated to these values.

Author(s)

Adrian Dusa

See Also

[pt](#), [qt](#)

Examples

```
# for 100 degrees of freedom
dariat("between", -1.96, 1.96, df = 100)

dariat("over", -1, df = 100)

dariat("under", -1, df = 100)

dariat("over", 2, df = 100, draw = TRUE)
```

dbinoms

Calculate probabilities and draw graphics for a binomial distribution

Description

This function draws graphics for a certain number of repetitions of an experiment, at a certain probability of success, and calculates the probability of obtaining one or more values from a random variable.

Usage

```
dbinoms(x, size, prob, log = FALSE, draw = FALSE,  
        zoom = FALSE, new = FALSE, text = FALSE)
```

Arguments

x	Number of favourable outcomes: a value or a vector of values
size	Number of repetitions
prob	Probability of success
log	Logical; if TRUE, the probability is returned as log(p)
draw	Logical; if TRUE, draws the binomial distribution
zoom	Logical; if TRUE, eliminates from the graphic all numbers with probability equal to zero
new	Logical; if TRUE, a new window will be created for each graphic
text	Logical; if TRUE, display the probability above each bar

Author(s)

Adrian Dusa

See Also

[dbinom](#)

Examples

```
# 8 repetitions, with a 0.5 probability of success, calculate the  
# probability of obtaining between 2 and 4 favourable outcomes  
dbinoms(2:4, 8, 0.5)  
  
# less than 7 favourable outcomes  
dbinoms(0:6, 8, 0.5)  
  
# at most 7 favourable outcomes  
dbinoms(0:7, 8, 0.5)  
  
# above 5 favourable outcomes  
dbinoms(6:8, 8, 0.5)  
  
# at least 5 favourable outcomes  
dbinoms(5:8, 8, 0.5)  
  
# exactly 6 favourable outcomes  
dbinoms(6, 8, 0.5)  
  
# 1, 3 or 6 favourable outcomes  
dbinoms(c(1, 3, 6), 8, 0.5)  
  
# same, drawing the graphic
```

```
dbinoms(c(1, 3, 6), 8, 0.5, draw = TRUE)

# same, drawing the probabilities in the graphic
dbinoms(c(1, 3, 6), 8, 0.5, draw = TRUE, text = TRUE)
```

dfcalc and spooled	<i>Calculates the degrees of freedom and the pooled variation for a t test</i>
--------------------	--

Description

The function `dfcalc` is used only for two samples t test, when the group variations are **not** equal. For small and independent samples, and unknown but equal population variances, the variances of the two samples are used. As the sample variances are never equal, this function calculates their pooled variance based on the two standard deviations and their respective sample sizes.

Usage

```
dfcalc(x, y, n1, n2)
spooled(x, y, n1, n2)
```

Arguments

<code>x</code>	The values of the standard deviation for the first group
<code>y</code>	The values of the standard deviation for the second group
<code>n1</code>	Size of the first group
<code>n2</code>	Size of the second group

Author(s)

Adrian Dusa

Examples

```
group1 <- c(13, 14, 9, 12, 8, 10, 5, 10, 9, 12, 16)
group2 <- c(16, 18, 11, 19, 14, 17, 13, 16, 17, 18, 22, 12)
sd1 <- sd(group1)
sd2 <- sd(group2)
n1 <- length(group1)
n2 <- length(group2)

# more direct
dfcalc(group1, group2)

# if the standard deviations and group sizes are known
dfcalc(sd1, sd2, n1, n2)

# the pooled standard deviation
```

```
spooled(sd1, sd2, n1, n2)

# more direct
spooled(group1, group2)
```

histc*Histogram with a superimposed normal curve*

Description

Draws a histogram with a normal curve that approximates the distribution. When `ylim` is not provided, it is chosen to accommodate both the histogram bars and the superimposed curve.

Usage

```
histc(x, from, to, size = 15, ...)
```

Arguments

<code>x</code>	Numeric vector
<code>from</code>	Starting point on the horizontal axis.
<code>to</code>	End point on the horizontal axis.
<code>size</code>	Size of the graphic, in centimeters.
<code>...</code>	Other parameters, specific to the base <code>hist()</code> function.

Author(s)

Adrian Dusa

Examples

```
x <- sample(18:93, 150, replace = TRUE)

histc(x)

histc(x, 10, 100)

histc(x, 10, 100, xlab = "Age", ylab = "Frequency",
      main = "Histogram for age in years")
```

mbinom and sbinom	<i>Calculates the mean and the standard deviation of a discrete random variable</i>
-------------------	---

Description

The function expects a tabel (a data frame or a matrix) with just two columns: the first containing the values of a random variable, and the associated probabilities in the second column.

Usage

```
mbinom(x)
sbinom(x)
```

Arguments

x The data table.

Details

If the sum of the probabilities on the second columns is not equal to 1, the function interprets them as absolute values and recalculates the relative frequencies.

Author(s)

Adrian Dusa

Examples

```
data <- matrix(c(0:4, 0.015, 0.235, 0.425, 0.245, 0.080), ncol = 2)
mbinom(data)
sbinom(data)

data <- data.frame(X = 0:4, P_X = c(12, 188, 340, 196, 64))
mbinom(data)
sbinom(data)
```

t.testhv	<i>Student's t test with a homogeneity of variances check</i>
----------	---

Description

Performs one-sample, two-sample and paired t tests, with additional support for choosing between the classical Student t test and the Welch t test. For two independent samples, the default behavior is to first run the Ansari-Bradley test for the homogeneity of variances.

Usage

```
## S3 method for class 'testhv'
t(x, ...)

## Default S3 method:
t.testhv(
  x, y = NULL,
  alternative = c("two.sided", "less", "greater"),
  mu = 0, paired = FALSE, var.equal = NULL, conf.level = 0.95,
  ...
)

## S3 method for class 'formula'
t.testhv(
  formula, data, subset, na.action = na.pass, ...
)

## S3 method for class 'Pair'
t.testhv(
  x,
  alternative = c("two.sided", "less", "greater"),
  mu = 0, var.equal = NULL, conf.level = 0.95,
  ...
)
```

Arguments

x	A numeric vector, a Pair object for paired tests, or the first argument dispatched by the generic method.
y	An optional second numeric vector. If y is a categorical object of class "declared", it is treated as a grouping variable and x is split into two samples.
formula	A formula. Use values ~ group for two independent samples, values ~ 1 for a one-sample test, or Pair(before, after) ~ 1 for a paired test.
data	An optional data frame, matrix or list containing the variables in formula.
subset	An optional vector specifying a subset of observations.
na.action	A function specifying how missing values should be handled by the formula method.
alternative	Character string specifying the alternative hypothesis. See Details.
mu	A number indicating the true value of the mean, difference in means, or mean difference under the null hypothesis.
paired	Logical, indicating whether the two supplied numeric vectors are paired. This argument is not accepted by the formula method; use Pair(before, after) ~ 1 instead.
var.equal	Logical or NULL. If TRUE, the classical Student t test with pooled variance is printed. If FALSE, the Welch test is printed. If NULL, the default, an Ansari-Bradley test is used to decide which of the two tests to print.

<code>conf.level</code>	Numeric value between 0 and 1, giving the confidence level of the interval.
<code>...</code>	Additional arguments passed to the relevant method.

Details

The main difference from base R's `t.test` is the default treatment of `var.equal`. In `t.test()`, this argument is logical and defaults to `FALSE`, meaning Welch's unequal-variance test is used unless the user explicitly requests the pooled-variance Student test.

In `t.testhv()`, `var.equal` defaults to `NULL`. This leaves the choice to the function: for two independent samples it first applies `ansari.test` as a homogeneity of variances test, then prints the pooled-variance Student test when the variances are considered homogeneous, or the Welch test otherwise. Users can still set `var.equal = TRUE` or `var.equal = FALSE` to mimic the explicit behavior of `t.test`.

Internally, both the pooled-variance and Welch tests are computed and stored in the result object. The print method decides which one to display, based on `var.equal` and, when `var.equal = NULL`, the p-value of the homogeneity test.

The `alternative` argument accepts the same standard values as `t.test`: `"two.sided"`, `"less"` and `"greater"`. In addition, `"!="` and `"two.tailed"` are accepted for a two-sided test, `"<"` and `"lower"` for a left-tailed test, and `">"`, `"higher"` and `"upper"` for a right-tailed test.

Formula calls support categorical grouping variables of class `"declared"`. Such variables are coerced to factors before splitting the response into two groups. The default method also supports the shortcut `t.testhv(values, group)` when `group` is a `"declared"` categorical variable with exactly two levels.

Paired t tests can be requested either with two numeric vectors and `paired = TRUE`, or by supplying a `Pair` object. In the formula method, paired tests should be written as `Pair(before, after) ~ 1`; `paired = TRUE` is deliberately rejected in formula calls.

Value

An object of class `"ttesthv"`, containing:

- `homogtest`: the Ansari-Bradley homogeneity of variances test, or `NULL` when it is not relevant;
- `ttest`: the pooled-variance Student t test;
- `ttestWelch`: the Welch t test;
- `paired`, `var.equal` and `conf.level`: the corresponding analysis settings.

Author(s)

Adrian Dusa

See Also

`t.test`, `ansari.test`, `Pair`

Examples

```

group1 <- c(13, 14, 9, 12, 8, 10, 5, 10, 9, 12, 16)
group2 <- c(16, 18, 11, 19, 14, 17, 13, 16, 17, 18, 22, 12)

t.testhv(group1, group2)

# Force the same variance choice as base R's t.test()
t.testhv(group1, group2, var.equal = FALSE)
t.testhv(group1, group2, var.equal = TRUE)

# Formula interface
dataset <- data.frame(
  values = c(group1, group2),
  group = c(rep(1, 11), rep(2, 12))
)

t.testhv(values ~ group, data = dataset)

using(
  dataset,
  t.testhv(values ~ group)
)

# Declared categorical grouping variables are detected
dataset$declared_group <- declared(
  dataset$group,
  labels = c(Group1 = 1, Group2 = 2)
)

t.testhv(values ~ declared_group, data = dataset)
t.testhv(dataset$values, dataset$declared_group)

# Paired tests
before <- c(8, 7, 6, 9, 10)
after <- c(7, 6, 7, 6, 8)

t.testhv(before, after, paired = TRUE)
t.testhv(Pair(before, after))

paired_data <- data.frame(before = before, after = after)
t.testhv(Pair(before, after) ~ 1, data = paired_data)

```

Index

* functions

histc, [10](#)

* misc

About the statistics package, [2](#)

anovahv, [2](#)

daria, [5](#)

dariat, [6](#)

dbinoms, [7](#)

dfcalc and spooled, [9](#)

mbinom and sbinom, [11](#)

t.testhv, [11](#)

About the statistics package, [2](#)

anova, [4](#)

anovaFK (anovahv), [2](#)

anovahv, [2](#)

ansari.test, [13](#)

aov, [3](#), [4](#)

daria, [5](#)

dariat, [6](#)

dbinom, [8](#)

dbinoms, [7](#)

dfcalc (dfcalc and spooled), [9](#)

dfcalc and spooled, [9](#)

fligner.test, [3](#), [4](#)

histc, [10](#)

lm, [3](#)

mbinom (mbinom and sbinom), [11](#)

mbinom and sbinom, [11](#)

oneway.test, [3](#), [4](#)

Pair, [13](#)

pnorm, [6](#)

pt, [7](#)

qnorm, [6](#)

qt, [7](#)

sbinom (mbinom and sbinom), [11](#)

spooled (dfcalc and spooled), [9](#)

statistics (About the statistics package), [2](#)

t.test, [13](#)

t.testAB (t.testhv), [11](#)

t.testhv, [3](#), [4](#), [11](#)