Package ‘DanielBiostatistics10th’

August 8, 2022

Type Package

Title Functions for Wayne W. Daniel's Biostatistics, Tenth Edition

Version 0.1.7

Date 2022-08-04

Author Tingting Zhan [aut, cre, cph]

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Description Functions to accompany Wayne W. Daniel's Biostatistics: A Foundation for Analysis in the Health Sciences, Tenth Edition.

License GPL-2

Encoding UTF-8

Depends ggplot2

Imports e1071, ggrepel, latex2exp, methods, pracma, scales

Suggests car, DescTools

Language en-US

RoxygenNote 7.2.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-08 18:30:21 UTC

R topics documented:

`DanielBiostatistics10th-package` *Functions for Wayne W. Daniel's* Biostatistics *(Tenth Edition)*

Description

Functions and examples to accompany Wayne W. Daniel's *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition, Wiley, ISBN: 978-1-119-62550-6.

`https://www.wiley.com/en-us/Biostatistics:+A+Foundation+for+Analysis+in+the+Health+Sciences,+10th+Edition-p-9781119625506`

Data sets from 10th edition `https://bcs.wiley.com/he-bcs/Books?action=resource&bcsId=7849&itemId=1118302796&resourceId=30373`.

Resources from 11th edition `https://bcs.wiley.com/he-bcs/Books?action=index&bcsId=11491&itemId=1119496578`, with errata of data.

`binom2pois` *Binomial Approaching Poisson*

Description

Binomial Approaching Poisson

Usage

```
binom2pois(x, lambda, size = c(10L, 100L))
```

Arguments

`x`	integer scalar, observed number of responses
`lambda`	positive numeric scalar, parameter λ of Poisson distribution
`size`	integer vector, parameter n of binomial distribution

Details

binom2pois shows how binomial density approaches Poisson density when $n \to \infty$ and $p \to 0$, while holding a constant product $np = \lambda$.

Value

binom2pois returns a binom2pois object, for which a print method, an autolayer and an autoplot method are defined.

See Also

dbinom dpois

Examples

```
binom2pois(x = 4L, lambda = 6, size = seq.int(10L, 50L, by = 10L))
```

`BooleanRisk-class` *Boolean Risk-&-Disease Table*

Description

Boolean Risk-&-Disease Table

Slots

`.Data` two-by-two integer matrix, a Boolean risk-&-disease contingency table with layout

	Disease $(+)$	Disease $(-)$
Risk Factor $(+)$	x_{++}	x_{+-}
Risk Factor $(-)$	x_{-+}	x_{--}

`BooleanTest-class` *Boolean Test-&-Disease Table*

Description

Boolean Test-&-Disease Table

Slots

`.Data` two-by-two integer matrix, a Boolean test-&-disease contingency table with layout

	Disease $(+)$	Disease $(-)$

Test $(+)$	x_{++}	x_{+-}
Test $(-)$	x_{-+}	x_{--}

Chapter01 *Chapter 1*

Description

Functions and examples for Chapter 1, *Introduction to Biostatistics*.

Usage

```
sampleRow(x, size, replace = FALSE, prob = NULL)
```

Arguments

`x`	a data.frame
`size`	positive integer scalar, number of rows to be selected
`replace`	logical scalar, whether sampling should be with replacement (default `FALSE`)
`prob`	numeric vector of probability weights for each row of input `x` being sampled. Default `NULL` indicates simple random sampling

Value

sampleRow returns a data.frame, a simple random sample from the input.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

sample.int

Examples

```
library(DanielBiostatistics10th)
# To run a line of code, use shortcut
# Command + Enter: Mac and RStudio Cloud
# Control + Enter: Windows, Mac and RStudio Cloud
# To clear the console
# Control + L: Mac and RStudio Cloud

# Page 8, Example 1.4.1
d141 = read.csv(system.file('extdata', 'EXA_C01_S04_01.csv', package = 'DanielBiostatistics10th'))
```

```
# to read in the data for Example 1.4.1, as provided by Publisher in .csv format
# ?read.csv # invoke the help files of an R 'function'
class(d141) # `d141` is a 'data.frame' (a specific class defined in R)
dim(d141) # dimension, number-row and number-column
head(d141, n = 8L) # first `n` rows of a 'data.frame'
names(d141) # column names of a 'data.frame'
d141$AGE # use `$` to obtain one column from a 'data.frame'
sampleRow(d141, size = 10L, replace = FALSE) # to answer Example 1.4.1

# Page 11, Example 1.4.2
d141[seq.int(from = 4L, to = 166L, by = 18L), ]
```

Chapter02 *Chapter 2*

Description

Functions and examples for Chapter 2, *Descriptive Statistics*.

Usage

```
print_stats(x, na.rm = TRUE)

print_freqs(x, breaks, include.lowest = TRUE, right = TRUE)
```

Arguments

`x` numeric vector, the observations. In print_freqs function, this argument can also be a factor

`na.rm` logical scalar, whether to remove the missing observations (default `TRUE`)

`breaks` numeric vector, see cut.default

`include.lowest` logical scalar, default `TRUE`. See cut.default

`right` logical scalar, see cut.default

Details

print_freqs prints the (relative) frequencies and cumulative (relative) frequencies, from a numeric input vector, specified interval breaks as well as open/close status of the ends of the intervals.

print_stats prints the simple statistics of the input observations, such as sample size, mean, median, (smallest) mode, variance, standard deviation, coefficient of variation (if all observations are non-negative), quartiles, inter-quartile range (IQR), range, skewness and kurtosis. A histogram is also printed.

Value

print_freqs returns a freqs object, for which a show method, an autolayer and an autoplot method are defined.

print_stats does not have a returned value.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

cut.default table cumsum mean.default median.default Mode var sd quantile skewness kurtosis

Examples

```
library(DanielBiostatistics10th)

# Page 20, Example 2.2.1
d141 = read.csv(system.file('extdata', 'EXA_C01_S04_01.csv', package = 'DanielBiostatistics10th'))
head(d141)
class(d141$AGE) # 'integer'
class(age <- as.numeric(d141$AGE)) # 'numeric'
sort(age) # Page 21, Table 2.2.1 # 'ordered vector'

# Page 23, Example 2.3.1
(ageB = seq.int(from = 30, to = 90, by = 10))
(r231 = print_freqs(age, breaks = ageB, right = FALSE)) # Page 25, Table 2.3.2
# The open/close of interval ends is determined by textbook using 30-39, 40-49, etc.
autoplot(r231, title = 'Page 27, Figure 2.3.2')

# Page 38-42, Example 2.4.1 - Example 2.4.6
# Page 44-46, Example 2.5.1 - Example 2.5.3
print_stats(age) # or some other data input

# Page 49, Example 2.5.4 (omitted)

# Page 50, Example 2.5.5
d255 = read.csv(system.file('extdata', 'EXA_C02_S05_05.csv', package = 'DanielBiostatistics10th'))
head(d255)
boxplot(d255$GRF, main = c('GRF from Page 50, Example 2.5.5'))
print_stats(d255$GRF)
print_freqs(d255$GRF, breaks = seq.int(10, 45, by = 5))
```

Chapter03 *Chapter 3*

Description

Functions for Chapter 3, *Some Basic Probability Concepts*.

Usage

```
addProbs(A)

predictiveValues(
  A,
  sensitivity = A[1, 1]/sum(A[, 1]),
  specificity = A[2, 2]/sum(A[, 2]),
  prevalence = stop("must provide prevalence")
)
```

Arguments

`A` integer matrix, two-dimensional contingency table. For predictiveValues function, this must be a BooleanTest object, or a two-by-two integer matrix of the same with layout as outlined in BooleanTest

`sensitivity, specificity`
numeric scalars, sensitivity and specificity of a test. By default, these are calculated by the test-disease contingency table `A`

`prevalence` numeric scalar or vector, prevalence(s) of disease

Details

addProbs provides the joint, marginal and conditional probabilities of a contingency table.

predictiveValues provides the predictive values based on the sensitivity, specificity of a test, as well as the disease prevalence.

Value

addProbs returns an addProbs object, which is a list consisting of a noquote matrix of joint probabilities, and two noquote matrix of conditional probabilities. A print method is defined for addProbs object.

predictiveValues returns a predictiveValues object, which is a list of three double vector elements named `'Prevalence'`, `'PVP'` and `'PVN'`. A print method and an autoplot method are defined for predictiveValues object.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

addmargins rowSums colSums proportions

Examples

```
library(DanielBiostatistics10th)

# Page 69-75, Example 3.4.1 - Example 3.4.8
(d341 = matrix(c(28L, 19L, 41L, 53L, 35L, 38L, 44L, 60L), ncol = 2L, dimnames = list(
  FamilyHx = c('none', 'Bipolar', 'Unipolar', 'UniBipolar'),
  Onset = c('Early', 'Late'))))
class(d341) # 'matrix', i.e., a two-dimensional 'array'
addProbs(d341)

# Page 81, Example 3.5.1
(d351 = matrix(c(436L, 14L, 5L, 495L), nrow = 2L, dimnames = list(
  Test = c('Positive', 'Negative'), Alzheimer = c('Yes', 'No'))))
predictiveValues(d351, prevalence = .113)
predictiveValues(d351, prevalence = c(.005, .98))
```

Chapter04 *Chapter 4*

Description

Functions for Chapter 4, *Probability Distributions*.

Usage

```
binomBar(size, prob, xlim = size, title)

poisBar(lambda, xlim, title)
```

Arguments

size	non-negative integer scalar, number of trials for binomial distribution
prob	numeric scalar between 0 and 1, probability of success on each trial for binomial distribution
xlim	length-two numeric vector, horizontal limit of the figure
title	character scalar, title of the figure
lambda	positive numeric scalar, mean of Poisson distribution

Details

binomBar and poisBar generate bar plots of binomial and Poisson distributions.

Value

binomBar and poisBar returns a 'discreteDistBar' object, for which a print method, an autolayer and an autoplot method are defined.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

dbinom dpois

Examples

```
library(DanielBiostatistics10th)

# Page 93-97, Example 4.2.1 - Example 4.2.7
d421 = rep(1:8, times = c(62L, 47L, 39L, 39L, 58L, 37L, 4L, 11L))
(fq421 = print_freqs(factor(d421))) # Page 94, Table 4.2.1 and 4.2.2; Page 96, Table 4.2.3
autoplot(fq421, type = 'density', title = 'Page 95, Figure 4.2.1')
autoplot(fq421, type = 'distribution', title = 'Page 96, Figure 4.2.2')

# ?dbinom # 'd' for binomial 'density'; calculate Prob(X = x)
# ?pbinom # 'p' for binomial 'probability'
# `lower.tail = TRUE` (default), calculate Prob(X <= x)
# `lower.tail = FALSE`, calculate Prob(X > x)

# Page 99, Example 4.3.1
dbinom(x = 3L, size = 5L, prob = .858)
# Page 103, Example 4.3.2
dbinom(x = 4L, size = 10L, prob = .14)
# Page 103, Example 4.3.3
(pL = pbinom(q = 5L, size = 25L, prob = .1, lower.tail = TRUE)) # (a) including!
(pU = pbinom(q = 5L, size = 25L, prob = .1, lower.tail = FALSE)) # (b) excluding!
pL + pU # R makes sure they add up to 1
binomBar(size = 25L, prob = .1)
# Page 103, Example 4.3.4
dbinom(x = 7L, size = 12L, prob = .55)
pbinom(q = 5L, size = 12L, prob = .55)
pbinom(q = 7L, size = 12L, prob = .55, lower.tail = FALSE)

# Page 110, Example 4.4.1
dpois(x = 3L, lambda = 12)
# Page 110, Example 4.4.2
ppois(2L, lambda = 12, lower.tail = FALSE)
poisBar(lambda = 12, xlim = 30L)
# Page 110, Example 4.4.3
ppois(1L, lambda = 2)
# Page 111, Example 4.4.4
dpois(3L, lambda = 2)
# Page 112, Example 4.4.5
ppois(5L, lambda = 2, lower.tail = FALSE)

# Page 119. Example 4.6.1
pnorm(2)
# Page 120. Example 4.6.2
```

```
pnorm(2.55) - pnorm(-2.55)
1 - 2 * pnorm(-2.55) # alternative solution
# Page 121. Example 4.6.3
pnorm(1.53) - pnorm(-2.74)
# Page 121. Example 4.6.4
pnorm(2.71, lower.tail = FALSE)
# Page 122. Example 4.6.5
pnorm(2.45) - pnorm(.84)

# Page 122. Example 4.7.1
pnorm(q = 3, mean = 5.4, sd = 1.3)
pnorm(q = (3-5.4)/1.3) # manual solution
# Page 125. Example 4.7.2
pnorm(649, mean = 491, sd = 119) - pnorm(292, mean = 491, sd = 119)
# Page 122. Example 4.7.3
1e4L * pnorm(8.5, mean = 5.4, sd = 1.3, lower.tail = FALSE)
```

`Chapter05to07` *Chapter 5, 6 and 7*

Description

Functions for Chapter 5, *Some Important Sampling Distributions*, Chapter 6, *Estimation* and Chapter 7, *Hypothesis Testing*.

Usage

```
aggregated_z(
  xbar,
  n,
  sd,
  null.value,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  ...
)

aggregated_t(
  xbar,
  xsd,
  n,
  null.value,
  var.equal = FALSE,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  ...
)
```

```
prop_CLT(
  x,
  n,
  bool_obs,
  xbar = x/n,
  null.value,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  ...
)

aggregated_var(
  xsd,
  n,
  null.value,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  ...
)
```

Arguments

`xbar` numeric scalar or length-two vector. Sample mean(s) for numeric variable(s) $\bar{x}$ or $(\bar{x}_1, \bar{x}_2)$. Sample proportion(s) for binary (i.e., logical) variable(s) $\hat{p}$ or $(\hat{p}_1, \hat{p}_2)$. In the case of two-sample tests, this could also be a numeric scalar indicating the difference in sample means $\bar{x}_1 - \bar{x}_2$ or sample proportions $\hat{p}_1 - \hat{p}_2$

`n` integer scalar n or length-two vector. Sample size(s) n or (n_1, n_2)

`sd` numeric scalar or length-two vector. population standard deviation(s) σ or (σ_1, σ_2)

`null.value` (optional) numeric scalar or length-two vector. Null value(s) of the population mean(s) (μ_0, (μ_{10}, μ_{20}), or $\mu_{10} - \mu_{20}$) for aggregated_z and aggregated_t. Null value(s) of the population proportion(s) (p_0, (p_{10}, p_{20}), or $p_{10} - p_{20}$) for prop_CLT. Null value(s) of the population variance(s) (ratio) (σ^2_0, $(\sigma^2_{10}, \sigma^2_{20})$, or $\sigma^2_{10}/\sigma^2_{20}$) for aggregated_var. If missing, only the confidence intervals will be computed.

`alternative` character scalar, alternative hypothesis, either `'two.sided'` (default), `'greater'` or `'less'`

`conf.level` numeric scalar, confidence level, default 0.95

`...` potential arguments, not in use currently

`xsd` numeric scalar or length-two vector. Sample standard deviation(s) $\sigma_{\bar{x}}$ or $(\sigma_{\bar{x}_1}, \sigma_{\bar{x}_2})$

`var.equal` logical scalar, whether to treat the two population variances as being equal (default `FALSE`) in aggregated_t

`x` integer scalar or length-two vector, number of positive count(s) of binary (i.e., logical) variable(s)

`bool_obs` logical vector of Boolean observations, used in one-sample z-test on proportion

Details

aggregated_z performs one- or two-sample z-test using the aggregated statistics of sample mean(s) and sample size(s) when `null.value` is provided. Otherwise, only the confidence interval based on z-distribution is computed.

aggregated_t performs one- or two-sample t-test using the aggregated statistics of sample mean(s), sample standard deviation(s) and sample size(s) when `null.value` is provided. Otherwise, only the confidence interval based on t-distribution is computed.

prop_CLT performs one- or two-sample z-test on proportion(s), using Central Limit Theorem when `null.value` is provided. Otherwise, only the confidence interval based on z-distribution is computed.

aggregated_var performs one-sample χ^2-test on variance, or two-sample F-test on variances, using the aggregated statistics of sample standard deviation(s) and sample size(s) when `null.value` is provided. Otherwise, only the confidence interval based on χ^2- or F-distribution is computed.

Value

aggregated_z returns an `'htest'` object when `null.value` is provided, otherwise returns a length-two numeric vector.

aggregated_t returns an htest object when `null.value` is provided, otherwise returns a length-two numeric vector.

prop_CLT returns an htest object when `null.value` is provided, otherwise returns a length-two numeric vector.

aggregated_var returns an htest object when `null.value` is provided, otherwise returns a length-two numeric vector.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

t.test prop.test var.test

Examples

```
library(DanielBiostatistics10th)

# Page 142, Example 5.3.2
aggregated_z(xbar = 190, sd = 12.7, n = 10L, null.value = 185.6, alternative = 'greater')
# Page 143, Example 5.3.3
pnorm(125, mean = 120, sd = 15/sqrt(50)) - pnorm(115, mean = 120, sd = 15/sqrt(50))
aggregated_z(125, sd = 15, n = 50L, null.value = 120, alternative = 'less')$p.value -
  aggregated_z(115, sd = 15, n = 50L, null.value = 120, alternative = 'less')$p.value

# Page 145, Example 5.4.1
aggregated_z(xbar = c(92, 105), sd = 20, n = 15L, null.value = 0, alternative = 'less')
# Page 148, Example 5.4.2
```

```
aggregated_z(xbar = 20, sd = c(15, 20), n = c(35L, 40L), null.value = c(45, 30),
  alternative = 'greater')

# Page 150, Example 5.5.1
prop_CLT(xbar = .4, n = 150L, null.value = .357, alternative = 'greater')
# Page 152, Example 5.5.2
prop_CLT(xbar = .45, n = 200L, null.value = .51, alternative = 'less')

# Page 155, Example 5.6.1
prop_CLT(xbar = .1, null.value = c(.28, .21), n = c(100L, 100L), alternative = 'greater')
# Page 155, Example 5.6.2
prop_CLT(xbar = .05, null.value = c(.34, .26), n = c(250L, 200L), alternative = 'less')

# Page 166, Example 6.2.1
aggregated_z(xbar = 22, n = 10L, sd = sqrt(45))
# Page 168, Example 6.2.2
aggregated_z(xbar = 84.3, n = 15L, sd = sqrt(144), conf.level = .99)
# Page 168, Example 6.2.3
aggregated_z(xbar = 17.2, n = 35L, sd = 8, conf.level = .9)
# Page 169, Example 6.2.4
d624 = read.csv(system.file('extdata', 'EXA_C06_S02_04.csv', package = 'DanielBiostatistics10th'))
head(d624)
aggregated_z(xbar = mean(d624$ACTIVITY), n = nrow(d624), sd = sqrt(.36))

# Page 173, Example 6.3.1
aggregated_t(xbar = 250.8, xsd = 130.9, n = 19L)

# Page 177, Example 6.4.1
aggregated_z(xbar = c(4.5, 3.4), sd = sqrt(c(1, 1.5)), n = c(12L, 15L))
# Page 178, Example 6.4.2
aggregated_z(xbar = c(4.3, 13), sd = c(5.22, 8.97), n = c(328L, 64L), conf.level = .99)
# Page 180, Example 6.4.3
aggregated_t(xbar = c(4.7, 8.8), xsd = c(9.3, 11.5), n = c(18L, 10L), var.equal = TRUE)
# Page 181, Example 6.4.4
aggregated_t(xbar = c(4.7, 8.8), xsd = c(9.3, 11.5), n = c(18L, 10L))
# Welch slightly different from Cochran; textbook explained on Page 182

# Page 185, Example 6.5.1
prop_CLT(xbar = .18, n = 1220L)

# Page 187, Example 6.6.1
prop_CLT(x = c(31L, 53L), n = c(68L, 255L), conf.level = .99)

# Page 190, Example 6.7.1
n_671 = uniroot(f = function(n, sd, level = .95) {
   qnorm(1-(1-level)/2) * sd/sqrt(n) - 5 # half-width of CI <= 5 grams
}, interval = c(0, 2e2), sd = 20)
sprintf('Example 6.7.1 requires a sample size of %d.', ceiling(n_671$root))

# Page 192, Example 6.8.1
n_681 = uniroot(f = function(n, p, level = .95) {
  qnorm(1-(1-level)/2) * sqrt(p*(1-p)/n) - .05
}, interval = c(0, 1e3), p = .35)
```

```
sprintf('Example 6.8.1 requires a sample size of %d.', ceiling(n_681$root))

# Page 196, Example 6.9.1
d691 = c(9.7, 12.3, 11.2, 5.1, 24.8, 14.8, 17.7)
sqrt(aggregated_var(xsd = sd(d691), n = length(d691)))

# Page 200, Example 6.10.1
aggregated_var(xsd = c(8.1, 5.9), n = c(16L, 4L))

# Page 222, Example 7.2.1
aggregated_z(xbar = 27, sd = sqrt(20), n = 10L, null.value = 30)
# Page 226, Example 7.2.2
aggregated_z(xbar = 27, sd = sqrt(20), n = 10L, null.value = 30, alternative = 'less')
# Page 228, Example 7.2.3
d723 = read.csv(system.file('extdata', 'EXA_C07_S02_03.csv', package = 'DanielBiostatistics10th'))
head(d723)
t.test(d723$DAYS, mu = 15)
# Page 231, Example 7.2.4
aggregated_z(xbar = 146, sd = 27, n = 157L, null.value = 140, alternative = 'greater')
# Page 232, Example 7.2.5
d725 = c(33.38, 32.15, 34.34, 33.95, 33.46, 34.13, 33.99, 34.10, 33.85,
  34.23, 34.45, 34.19, 33.97, 32.73, 34.05)
t.test(d725, mu = 34.5)

# Page 237, Example 7.3.1
aggregated_z(xbar = c(4.5, 3.4), sd = sqrt(c(1, 1.5)), n = c(12L, 15L), null.value = 0)
# Page 239, Example 7.3.2
d732 = read.csv(system.file('extdata', 'EXA_C07_S03_02.csv', package = 'DanielBiostatistics10th'))
head(d732)
with(d732, t.test(x = CONTROL, y = SCI, alternative = 'less', var.equal = TRUE))
# Page 240, Example 7.3.3
aggregated_t(xbar = c(19.16, 9.53), xsd = c(5.29, 2.69), n = c(15L, 30L), null.value = 0)
# Page 242, Example 7.3.4
aggregated_z(xbar = c(59.01, 46.61), sd = c(44.89, 34.85), n = c(53L, 54L), null.value = 0,
  alternative = 'greater')

# Page 251, Example 7.4.1
d741 = read.csv(system.file('extdata', 'EXA_C07_S04_01.csv', package = 'DanielBiostatistics10th'))
head(d741)
with(d741, t.test(x = POSTOP, y = PREOP, alternative = 'greater', paired = TRUE))

# Page 258, Example 7.5.1
prop_CLT(x = 24L, n = 301L, null.value = .063, alternative = 'greater')

# Page 261, Example 7.6.1
prop_CLT(x = c(24L, 11L), n = c(44L, 29L), null.value = 0, alternative = 'greater')

# Page 264, Example 7.7.1
d771 = read.csv(system.file('extdata', 'EXA_C07_S07_01.csv', package = 'DanielBiostatistics10th'))
aggregated_var(xsd = sd(d771$mass), n = 16L, null.value = 600)

# Page 268, Example 7.8.1
aggregated_var(xsd = c(30.62, 11.37), n = 6L, null.value = 1, alternative = 'greater')
```

```
# Page 270, Example 7.8.2
with(d732, var.test(x = CONTROL, y = SCI))
```

Chapter07_power *Chapter 7 (Power Curve)*

Description

Functions for Chapter 7, *Hypothesis Testing*.

Usage

```
power_z(
  x,
  null.value,
  sd,
  n,
  std.err = sd/sqrt(n),
  alternative = c("two.sided", "less", "greater"),
  sig.level = 0.05
)
```

Arguments

`x`	numeric vector, mean parameter(s) μ_1 in the alternative hypothesis
`null.value`	numeric scalar, mean parameter μ_0 in the null hypothesis
`sd`	numeric scalar, population standard deviation σ
`n`	integer scalar, sample size n
`std.err`	numeric scalar, standardized error. For one-sample z-test, this is $\sigma/\sqrt{n}$. Be aware of the name clash with stderr
`alternative`	character scalar, alternative hypothesis, either `'two.sided'` (default), `'greater'` or `'less'`
`sig.level`	numeric scalar, significance level (i.e., Type-I-error rate), default .05

Details

power_z calculates the powers at each element of the alternative parameters μ_1, for one-sample z-test

- $H_0 : \mu = \mu_0$ vs. $H_A : \mu \neq \mu_0$, if `alternative = 'two.sided'`
- $H_0 : \mu \leq \mu_0$ vs. $H_A : \mu > \mu_0$, if `alternative = 'greater'`
- $H_0 : \mu \geq \mu_0$ vs. $H_A : \mu < \mu_0$, if `alternative = 'less'`

Value

power_z returns a power object, for which a show method, an autolayer and an autoplot method are defined for power_z object.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

power.t.test

Examples

```
library(DanielBiostatistics10th)

# Page 272, Example 7.9.1
(p791 = power_z(seq.int(from = 16, to = 19, by = .5), null.value = 17.5, sd = 3.6, n = 100L))
# Page 275, Table 7.9.1
autoplot(p791, title = 'Page 275, Figure 7.9.2')

# Page 276, Example 7.9.2
(p792 = power_z(seq.int(from = 50, to = 80, by = 5), null.value = 65, sd = 15, n = 20L,
                sig.level = .01, alternative = 'less'))
autoplot(p792, title = 'Page 277, Figure 7.9.4')
autoplot(p792, all.alternative = TRUE, title = '1-sided vs. 2-sided test')

# Page 278, Example 7.10.1
(n_d7101 <- uniroot(f = function(x) {
  power_z(55, null.value = 65, sd = 15, n = x, sig.level = .01, alternative = 'less') - .95
}, interval = c(0, 50))$root)
(C_d7101 = qnorm(p = .01, mean = 65, sd = 15/sqrt(ceiling(n_d7101)), lower.tail = TRUE))
```

`Chapter09` *Chapter 9*

Description

Functions for Chapter 9, *Simple Linear Regression and Correlation*.

Usage

```
predict_lm(object, newx, level = 0.95, ...)
```

Arguments

`object`	lm object, with one and only one numeric predictor
`newx`	(optional) numeric scalar or vector, new x-value(s) for which the fitted response(s) are to be reported
`level`	numeric scalar, tolerance/confidence level, default .95
`...`	potential arguments, not in use currently

Value

predict_lm returns a predict_lm object, for which a print method, an autolayer and an autoplot method are defined.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

predict.lm

Examples

```
library(DanielBiostatistics10th)

# Page 417, Example 9.3.1
d931 <- read.csv(system.file('extdata', 'EXA_C09_S03_01.csv', package = 'DanielBiostatistics10th'))
head(d931)
names(d931)[2:3] = c('Waist', 'AT')
plot(AT ~ Waist, data = d931, xlab = 'Waist circumference (cm), X',
     ylab = 'Deep abdominal AT area (cm2), Y', main = 'Page 419, Figure 9.3.1')

# Page 436, Example 9.4.2
summary(m931 <- lm(AT ~ Waist, data = d931))
cor(d931[2:3]); cor.test(~ AT + Waist, data = d931)
confint(m931) # confidence interval of regression coefficients
anova(m931)

# Page 440, Example 9.4.3
plot(m931, which = 1, main = 'Page 440, Figure 9.4.8')

# Page 441, Section 9.5
autoplot(predict_lm(m931), xlab = 'Waist circumference (cm), X',
         ylab = 'Deep abdominal AT area (cm2), Y',
         title = 'Page 422, Figure 9.3.3; Page 442, Figure 9.5.1')

# Page 447, Example 9.7.1
d971 = read.csv(system.file('extdata', 'EXA_C09_S07_01.csv', package = 'DanielBiostatistics10th'))
head(d971)
summary(mod_971 <- lm(CV ~ HEIGHT, data = d971))
autoplot(predict_lm(mod_971), xlab = 'Height (cm)', ylab = 'Cv (units)',
         title = 'Page 449, Figure 9.7.2')

# Page 452, Example 9.7.2
cor(d971); cor.test(~ CV + HEIGHT, data = d971) # Page 451, Figure 9.7.4, Figure 9.7.5

# Page 453, When the Hypothesized rho Is a Nonzero Value
# R does not have a function to do this
```

Chapter11 *Chapter 11*

Description

Functions for Chapter 11, *Regression Analysis: Some Additional Techniques*.

Usage

```
predict_glm_binomial(object, newx, level = 0.95, ...)
```

Arguments

`object` glm object with binomial link function, i.e., a logistic regression model, as well as one and only one numeric predictor

`newx` (optional) numeric scalar or vector, new x-value(s) for which the fitted response(s) are to be reported

`level` numeric scalar, tolerance/confidence level, default .95

`...` potential arguments, not in use currently

Value

predict_glm_binomial returns a predict_glm_binomial object, for which a print method, an autolayer and an autoplot method are defined.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

See Also

predict.glm

Examples

```
library(DanielBiostatistics10th)
library(car)
library(DescTools)

# Page 540, Example 11.1.1
d1111 = read.csv(system.file('extdata', 'EXA_C11_S01_01.csv', package = 'DanielBiostatistics10th'))
head(log(d1111$conc, base = 10))
head(d1111$logConc)

# Page 542, Example 11.1.2
d1112 = read.csv(system.file('extdata', 'EXA_C11_S01_02.csv', package = 'DanielBiostatistics10th'))
cor.test(~ sbp + weight, data = d1112)
cor.test(~ sbp + bmi, data = d1112)

# Page 545, Example 11.2.1
d1121 = read.csv(system.file('extdata', 'EXA_C11_S02_01.csv', package = 'DanielBiostatistics10th'))
d1121a = within(d1121, expr = {
  SMOKE = as.logical(SMOKE)
})
xlab1121 = 'Length of gestation (weeks)'; ylab1121 = 'Birth weight (grams)'
car::scatterplot(GRAMS ~ WEEKS | SMOKE, data = d1121a, regLine = FALSE, smooth = FALSE,
                 xlab = xlab1121, ylab = ylab1121, main = 'Page 547, Figure 11.2.1')
# Page 547, Figure 11.2.2: main model (without interaction)
summary(m1121_main <- lm(GRAMS ~ WEEKS + SMOKE, data = d1121a))
confint(m1121_main)
car::scatterplot(GRAMS ~ WEEKS | SMOKE, data = d1121a, regLine = FALSE, smooth = FALSE,
                 xlab = xlab1121, ylab = ylab1121, main = 'Page 548, Figure 11.2.3')
(cf_main = m1121_main$coefficients)
abline(a = cf_main[1L], b = cf_main[2L], col = 'blue') # regression line for non-smoking mothers
abline(a = cf_main[1L] + cf_main[3L], b = cf_main[2L], col = 'magenta')

# Page 551, Example 11.2.3
d1123 = read.csv(system.file('extdata', 'EXA_C11_S02_03.csv', package = 'DanielBiostatistics10th'))
d1123a = within(d1123, expr = {
 METHOD = factor(METHOD, levels = c('C', 'A', 'B')) # textbook designated 'C' as reference level
})
summary(mod_1123 <- lm(EFFECT ~ AGE * METHOD, data = d1123a)) # Page 555, Figure 11.2.5
confint(mod_1123)
car::scatterplot(EFFECT ~ AGE | METHOD, data = d1123a, smooth = FALSE,
           xlab = 'Age', ylab = 'Treatment effectiveness', main = 'Page 555, Figure 11.2.6')

# (optional) Page 561, Example 11.3.1
d1131 = read.csv(system.file('extdata', 'EXA_C11_S03_01.csv', package = 'DanielBiostatistics10th'))
head(d1131)
names(d1131) = c('JOBPER', 'ASRV', 'ENTH', 'AMB', 'COMM', 'PROB', 'INIT')
summary(mod_1131_raw <- lm(JOBPER ~ ASRV + ENTH + AMB + COMM + PROB + INIT, data = d1131))
# summary(mod_1131 <- MASS::stepAIC(mod_1131_raw, direction = 'backward'))
# the stepwise selection criterion used in MINITAB is not necessarily AIC

# Page 572, Example 11.4.1
addmargins(d1141 <- array(c(92L, 21L, 15L, 20L), dim = c(2L, 2L), dimnames = list(
  OCAD = c('Present', 'Absent'), Sex = c('Male', 'Female')))) # Page 572, Table 11.4.2
(d1141a = within(as.data.frame(as.table(d1141)), expr = {
  OCAD = (OCAD == 'Present')
  Sex = factor(Sex, levels = c('Female', 'Male'))
}))
(m1141 = glm(OCAD ~ Sex, family = binomial(link = 'logit'), weights = Freq, data = d1141a))
summary(m1141) # Page 573, Figure 11.4.1
exp(m1141$coefficients[2L]) # exp(beta_M)
exp(confint(m1141)) # confidence interval of exp(beta)
predict(m1141, newdata = data.frame(Sex = setNames(nm = c('Male', 'Female'))), type = 'response')

# Page 573, Example 11.4.2
d1142 = read.csv(system.file('extdata', 'EXA_C11_S04_02.csv', package = 'DanielBiostatistics10th'))
summary(mod_1142 <- glm(ATT ~ AGE, family = binomial, data = d1142)) # Page 575, Figure 11.4.2
```

```
exp(mod_1142$coefficients[2L])
exp(confint(mod_1142))
car::Anova(mod_1142) # Optional
autoplot(predict_glm_binomial(mod_1142, newx = c(50, 65, 80)), title = 'Page 576, Figure 11.4.3')

# (optional) Page 576, Example 11.4.3
d1143 = read.csv(system.file('extdata', 'REV_C11_24.csv', package = 'DanielBiostatistics10th'))
summary(glm(ONSET ~ HIAA + TRYPT, family = binomial(link = 'logit'), data = d1143))
# Page 577, Figure 11.4.4
# Predictor TRYPT should be removed from model due to p-value \approx 1
summary(glm(ONSET ~ HIAA, family = binomial(link = 'logit'), data = d1143))

# (optional) Page 578, Example 11.4.4
DescTools::PseudoR2(mod_1142, which = 'CoxSnell')
DescTools::PseudoR2(mod_1142, which = 'Nagelkerke')

# (optional) Page 579, Example 11.4.5 (same as Example 11.4.4)
```

Chapter12 *Chapter 12*

Description

Functions for Chapter 12, *The Chi-Square Distribution and The Analysis of Frequencies*.

Usage

```
relativeRisk(A)

oddsRatio(A)

print_OE(O, prob)
```

Arguments

A	a BooleanRisk object, or a two-by-two integer matrix of the same with layout as outlined in BooleanRisk
O	integer vector, observed counts
prob	numeric vector, anticipated probability. If missing (default), an uniform distribution across all categories are used.

Value

relativeRisk returns a 'logRelativeRisk' object, for which a vcov method and a print method are defined.

oddsRatio returns a 'logOddsRatio' object, for which a vcov method and a print method are defined.

print_OE prints a table with observed and expected frequencies, as well as the category-wise χ^2 statistics. A double vector of the category-wise χ^2 statistics is returned invisibly.

References

Wayne W. Daniel, *Biostatistics: A Foundation for Analysis in the Health Sciences*, Tenth Edition. Wiley, ISBN: 978-1-119-62550-6.

Examples

```
library(DanielBiostatistics10th)

# Page 605, Example 12.3.1
d1231_b = c(-Inf, seq(from = 125, to = 275, by = 25), Inf)
(d1231 = setNames( # Page 605, Table 12.3.1
  c(1L, 3L, 8L, 18L, 6L, 4L, 4L, 3L),
  nm = levels(cut(double(), breaks = d1231_b, right = FALSE, include.lowest = TRUE))))
chi1231 = print_OE(d1231, prob = diff.default(pnorm(q = d1231_b, mean = 198.67, sd = 41.31)))
pchisq(sum(chi1231), df = length(d1231) - 3L, lower.tail = FALSE)
# -3L: three restrictions (explained on Page 608)
# (1) making sum(xo) == sum(xe)
# (2) estimating mean
# (3) estimating sd

# Page 609, Example 12.3.2
# 100 doctors, 25 patients per doctor
d1232 = c(5L, 6L, 8L, 10L, 10L, 15L, 17L, 10L, 10L, 9L, 0L)
o1232 = setNames(c(sum(d1232[1:2]), d1232[-(1:2)]), nm = c('0-1', 2:9, '10 or more'))
(p1232 = sum((0:10) * d1232) / (25 * 100)) # binomial `prob`
chi1232 = print_OE(o1232, prob = c(
  pbinom(1L, size = 25L, prob = p1232),
  dbinom(2:9, size = 25L, prob = p1232),
  pbinom(9, size = 25L, prob = p1232, lower.tail = FALSE)))
pchisq(sum(chi1232), df = length(o1232) - 2L, lower.tail = FALSE)
# -2L: two restrictions (explained on Page 611)
# (1) making sum(o) == sum(e)
# (2) estimating p1232

# Page 611, Example 12.3.3
d1233 = c(5L, 14L, 15L, 23L, 16L, 9L, 3L, 3L, 1L, 1L, 0L)
o_1233 = setNames(c(d1233[1:8], sum(d1233[-(1:8)])), nm = c(0:7, '8 or more'))
p_1233 = c(dpois(0:7, lambda = 3), # lambda = 3 is provided by the textbook
           ppois(7L, lambda = 3, lower.tail = FALSE))
chi1233 = print_OE(o_1233, prob = p_1233)
pchisq(sum(chi1233), df = length(o_1233) - 1L, lower.tail = FALSE)
# -1L: one restrictions
# (1) making sum(xo) == sum(xe)
chisq.test(o_1233, p = p_1233) # equivalent # warning on any(E < 5)

# Page 614, Example 12.3.4
d1234 = c('Dec 05' = 62L, 'Jan 06' = 84L, 'Feb 06' = 17L, 'Mar 06' = 16L, 'Apr 06' = 21L)
chi1234 = print_OE(d1234)
pchisq(sum(chi1234), df = length(d1234) - 1L, lower.tail = FALSE)
chisq.test(d1234) # equivalent

# Page 616, Example 12.3.5
```

```
d1235 = c(dominant = 43L, heterozygous = 125L, recessive = 32L)
chi1235 = print_OE(d1235, prob = c(1, 2, 1))
pchisq(sum(chi1235), df = length(d1235) - 1L, lower.tail = FALSE)
chisq.test(d1235, p = c(1, 2, 1), rescale.p = TRUE) # equivalent

# Page 621, Example 12.4.1
addmargins(d1241 <- array(c(260L, 15L, 7L, 299L, 41L, 14L), dim = c(3L, 2L), dimnames = list(
  Race = c('White', 'Black', 'Other'),
  FolicAcid = c('TRUE', 'FALSE'))))
chisq.test(d1241) # ?stats::chisq.test

# Page 626, Example 12.4.2
addmargins(d1242 <- array(c(131L, 14L, 52L, 36L), dim = c(2L, 2L), dimnames = list(
  Type = c('Faller', 'Non-Faller'),
  LifestyleChange = c('TRUE', 'FALSE'))))
chisq.test(d1242, correct = FALSE)
chisq.test(d1242, correct = TRUE) # Page 627, Yates's Correction

# Page 631, Example 12.5.1
addmargins(d1251 <- array(c(21L, 19L, 75L, 77L), dim = c(2L, 2L), dimnames = list(
  Group = c('Narcoleptic', 'Healthy'),
  Migraine = c('TRUE', 'FALSE'))))
(chisq_1251 = chisq.test(d1251, correct = FALSE))
if (FALSE) {
  # (optional) using test on two proportions
  # only equivalent for 2*2 contingency table
  (clt_1251 = prop_CLT(x = c(21L, 19L), n = 96L, null.value = 0))
  all.equal.numeric(unname(clt_1251$statistic^2), unname(chisq_1251$statistic))
}

# Page 638, Example 12.6.1
addmargins(d1262 <- array(c(2L, 8L, 7L, 4L), dim = c(2L, 2L), dimnames = list(
  Group = c('PI_Naive', 'PA_Experienced'),
  Regimen2yr = c('TRUE', 'FALSE'))))
fisher.test(d1262)

# Page 644, Example 12.7.1
addmargins(d1271 <- array(c(22L, 18L, 216L, 199L), dim = c(2L, 2L),
 dimnames = list(Exercising = c('Extreme', 'No'), PretermLabor = c('TRUE', 'FALSE'))))
relativeRisk(d1271)
# textbook confidence interval (.65, 1.86) wrong (too many rounding in intermediate steps)

# Page 647, Example 12.7.2
addmargins(d1272 <- array(c(64L, 68L, 342L, 3496L), dim = c(2L, 2L), dimnames = list(
  SmkPregnancy = c('TRUE', 'FALSE'),
  Obesity = c('TRUE', 'FALSE'))))
oddsRatio(d1272)

# Page 650, Example 12.7.3
# Page 652, Example 12.7.4
(d1273 <- array(c(21L, 16L, 11L, 6L, 50L, 18L, 14L, 6L), dim = c(2L, 2L, 2L), dimnames = list(
  HTN = c('Present', 'Absent'), OCAD = c('Cases', 'Controls'),
  Age = c('<=55', '>55'))))
```

```
addmargins(d1273, margin = 1:2) # Page 651, Table 12.7.6
mantelhaen.test(d1273)
```

`freqs-class` *S4 Class freqs*

Description

S4 Class freqs

Slots

`.Data` integer vector, frequency counts

`data.name` character integer, name of the data, only used in output

`Gosset_Welch` *Two-Sample Student's t-statistic and Welch–Satterthwaite Equation*

Description

To determine the degree of freedom, as well as the standard error, of two-sample t-statistic, with or without the equal-variance assumption.

Usage

```
Gosset_Welch(s1, s2, v1 = s1^2, v2 = s2^2, n1, n2, var.equal = FALSE)
```

Arguments

`s1, s2`	(optional) double vectors, sample standard deviations of the two samples
`v1, v2`	double vectors, sample variances of the two samples, default $v_1 = s_1^2$, $v_2 = s_2^2$.
`n1, n2`	integer vectors, sample sizes of the two samples
`var.equal`	logical scalar, whether to treat the two variances as being equal when calculating the degree of freedom and the standard error of the mean-difference. If `FALSE` (default), Welch–Satterthwaite equation is used. If `TRUE`, the original two-sample t-test from William Sealy Gosset is used.

Value

Gosset_Welch returns a numeric scalar of the degree of freedom, with a numeric scalar attribute `'std.err'` of the standard error of the mean-difference.

References

Student's t-test by William Sealy Gosset, doi:10.1093/biomet/6.1.1.

Welch–Satterthwaite equation by Bernard Lewis Welch and F. E. Satterthwaite doi:10.2307/3002019 and doi:10.1093/biomet/34.12.28.

See Also

t.test

Examples

```
x = rnorm(32L, sd = 1.6); y = rnorm(57L, sd = 2.1)
vx = var(x); vy = var(y); nx = length(x); ny = length(y)
t.test(x, y, var.equal = FALSE)[c('parameter', 'stderr')]
Gosset_Welch(v1 = vx, v2 = vy, n1 = nx, n2 = ny, var.equal = FALSE)
t.test(x, y, var.equal = TRUE)[c('parameter', 'stderr')]
Gosset_Welch(v1 = vx, v2 = vy, n1 = nx, n2 = ny, var.equal = TRUE)
```

`power-class` *S4 class power*

Description

S4 class power

Slots

`.Data` numeric scalar or vector, power(s) calculated at alternative parameter(s) μ_1

`x` numeric scalar or vector, alternative parameter(s) μ_1

`rr` RejectionRegion object

`RejectionRegion-class` *S4 class RejectionRegion*

Description

S4 class RejectionRegion

Slots

`.Data` numeric scalar for one-sided test, or length-two numeric vector for two-sided test

`null.value` numeric scalar, null value

`std.err` numeric scalar, standard error of sampling distribution

`parameter` numeric vector, additional parameters

`alternative` character scalar, alternative hypothesis

`sig.level` numeric scalar, significance level (Type I error probability)

`test` character scalar, type of test. Currently only `'z'` is supported

`show,BooleanRisk-method` *Show BooleanRisk Object*

Description

Show BooleanRisk object

Usage

```
## S4 method for signature 'BooleanRisk'
show(object)
```

Arguments

`object`	a BooleanRisk object

Value

The show method for BooleanRisk object does not have a returned value.

`show,BooleanTest-method` *Show BooleanTest Object*

Description

Show BooleanTest object

Usage

```
## S4 method for signature 'BooleanTest'
show(object)
```

Arguments

`object`	a BooleanTest object

Value

The show method for BooleanTest object does not have a returned value.

`show,freqs-method` *Show freqs Object*

Description

Show freqs object

Usage

```
## S4 method for signature 'freqs'
show(object)
```

Arguments

`object`	an freqs object

Value

The show method for freqs object does not have a returned value.

`show,power-method` *Show power Object*

Description

Show power object

Usage

```
## S4 method for signature 'power'
show(object)
```

Arguments

`object`	a power object

Value

The show method for power object does not have a returned value.

Index