Package 'DanielBiostatistics10th'

April 11, 2024

Type Package

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

Version 0.2.0

Date 2024-04-11

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 R (>= 4.3)

Imports ggplot2, e1071, ggrepel, latex2exp, methods, pracma, scales

Suggests BSDA, car, caret, DescTools, ggfortify, mblm, psych, reshape2, robslopes, survival

Language en-US

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.1

NeedsCompilation no

Author Tingting Zhan [aut, cre, cph] (<https://orcid.org/0000-0001-9971-4844>)

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Repository CRAN

Date/Publication 2024-04-11 14:40:29 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.

Author(s)

Maintainer: Tingting Zhan <tingtingzhan@gmail.com> (ORCID) [copyright holder]

`Chapter01` *Chapter 1: Introduction to Biostatistics*

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

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

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: Windows, Mac and RStudio Cloud

# Example 1.4.1; Page 8 (10th ed), Page 7 (11th ed)
class(EXA_C01_S04_01) # `EXA_C01_S04_01` is a 'data.frame' (a specific class defined in R)
dim(EXA_C01_S04_01) # dimension, number-row and number-column
head(EXA_C01_S04_01, n = 8L) # first `n` rows of a 'data.frame'
names(EXA_C01_S04_01) # column names of a 'data.frame'
EXA_C01_S04_01$AGE # use `$` to obtain one column from a 'data.frame'
sampleRow(EXA_C01_S04_01, size = 10L, replace = FALSE) # to answer Example 1.4.1

# Example 1.4.2; Page 11 (10th ed), Page 10 (11th ed)
EXA_C01_S04_01[seq.int(from = 4L, to = 166L, by = 18L), ]
```

Chapter02 *Chapter 2: Descriptive Statistics*

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 function print_freqs, 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

Function 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.

Function 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

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

Function print_stats does not have a returned value.

See Also

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

Examples

```
library(DanielBiostatistics10th)
library(ggplot2)

# Example 2.2.1; Page 20 (10th ed), Page 19 (11th ed)
head(EXA_C01_S04_01)
class(age <- EXA_C01_S04_01$AGE) # 'integer'
sort(age) # Table 2.2.1

# Example 2.3.1; Page 23 (10th ed); Page 22 (11th ed)
(r231 = print_freqs(age, breaks = seq.int(from=30, to=90, by=10), right = FALSE)) # Table 2.3.2
# The open/close of interval ends is determined by textbook using 30-39, 40-49, etc.
autoplot(r231) + labs(title = 'Figure 2.3.2 (10th ed)')

# Example 2.4.1-2.4.6; Page 38-42 (10th ed); Page 30-34 (11th ed)
# Example 2.5.1-2.5.3; Page 44-46 (10th ed); Page 35-41 (11th ed)
print_stats(age) # or some other data input

# Example 2.5.4 (omitted); Page 49 (10th ed), Page 41 (11th ed)
```

```
# Example 2.5.5; Page 50 (10th ed)
head(EXA_C02_S05_05)
boxplot(EXA_C02_S05_05$GRF, main = c('Example 2.5.5 (10th ed)'))
print_stats(EXA_C02_S05_05$GRF)
print_freqs(EXA_C02_S05_05$GRF, breaks = seq.int(10, 45, by = 5))
```

Chapter03 *Chapter 3: Some Basic Probability Concepts*

Description

Examples in Chapter 3, *Some Basic Probability Concepts*.

Value

No function defined for Chapter 3.

Examples

```
library(DanielBiostatistics10th)

# Example 3.4.1-3.4.9; Page 69-75 (10th ed), Page 61-67 (11th ed)
(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)
addProbs(d341, margin = 1L)
addProbs(d341, margin = 2L)

# Example 3.5.1; Page 81 (10th ed), Page 72 (11th ed)
(d351 = matrix(c(436L, 14L, 5L, 495L), nrow = 2L, dimnames = list(
  Test = c('Positive', 'Negative'), Alzheimer = c('Yes', 'No'))))
summary(BooleanTable(t(d351)), prevalence = .113)
```

Chapter04 *Chapter 4: Probability Distributions*

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

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

Value

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

See Also

dbinom dpois

Examples

```
binomBar(size = 25L, prob = .1)
poisBar(lambda = 12, xlim = 30L)

library(DanielBiostatistics10th)

# Example 4.2.1-4.2.7; Page 93-97 (10th ed), Page 81-85 (11th ed)
d421 = rep(1:8, times = c(62L, 47L, 39L, 39L, 58L, 37L, 4L, 11L))
print_freqs(d421) # Table 4.2.1, 4.2.2, Table 4.2.3

# ?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)

# Example 4.2.8; Page 98 (10th ed), Page 85 (11th ed)
mean(d421)
sd(d421)
var(d421)

# Example 4.3.1; Page 99 (10th ed)
dbinom(x = 3L, size = 5L, prob = .858)
# Example 4.3.1; Page 87 (11th ed)
dbinom(x = 3L, size = 5L, prob = .899)

# Example 4.3.2; Page 103 (10th ed), Page 90 (11th ed)
dbinom(x = 4L, size = 10L, prob = .14)
```

```
# Example 4.3.3; Page 103 (10th ed), Page 91 (11th ed)
(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 # = 1

# Example 4.3.4; Page 105 (10th ed), Page 92 (11th ed)
dbinom(x = 7L, size = 12L, prob = .55)
pbinom(q = 5L, size = 12L, prob = .55)
pbinom(q = 7L, size = 12L, prob = .55, lower.tail = FALSE)

# Example 4.4.1; Page 110 (10th ed), Page 97 (11th ed)
dpois(x = 3L, lambda = 12)

# Example 4.4.2; Page 110 (10th ed), Page 98 (11th ed)
ppois(2L, lambda = 12, lower.tail = FALSE)

# Example 4.4.3; Page 110 (10th ed), Page 98 (11th ed)
ppois(1L, lambda = 2)

# Example 4.4.4; Page 111 (10th ed), Page 98 (11th ed)
dpois(3L, lambda = 2)

# Example 4.4.5; Page 112 (10th ed), Page 98 (11th ed)
ppois(5L, lambda = 2, lower.tail = FALSE)

# Example 4.6.1; Page 119 (10th ed), Page 106 (11th ed)
pnorm(2)

# Example 4.6.2; Page 120 (10th ed), Page 106 (11th ed)
pnorm(2.55) - pnorm(-2.55)
1 - 2 * pnorm(-2.55) # alternative solution

# Example 4.6.3; Page 121 (10th ed), Page 107 (11th ed)
pnorm(1.53) - pnorm(-2.74)

# Example 4.6.4; Page 121 (10th ed), Page 107 (11th ed)
pnorm(2.71, lower.tail = FALSE)

# Example 4.6.5; Page 122 (10th ed), Page 107 (11th ed)
pnorm(2.45) - pnorm(.84)

# Example 4.7.1; Page 122 (10th ed), Page 109 (11th ed)
pnorm(q = 3, mean = 5.4, sd = 1.3)
pnorm(q = (3-5.4)/1.3) # manual solution

# Example 4.7.2; Page 125 (10th ed), Page 111 (11th ed)
pnorm(649, mean = 491, sd = 119) - pnorm(292, mean = 491, sd = 119)

# Example 4.7.3; Page 122 (10th ed), Page 111 (11th ed)
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 functions 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_0^2, $(\sigma_{10}^2, \sigma_{20}^2)$, or $\sigma_{10}^2/\sigma_{20}^2$) for function 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 function 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

Function 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.

Function 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.

Function 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.

Function 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

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

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

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

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

See Also

t.test prop.test var.test

Examples

```
library(DanielBiostatistics10th)

# Example 5.3.2; Page 142,
aggregated_z(xbar = 190, sd = 12.7, n = 10L, null.value = 185.6, alternative = 'greater')

# Example 5.3.3; Page 143,
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

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

# Example 5.4.2; Page 148,
aggregated_z(xbar = 20, sd = c(15, 20), n = c(35L, 40L), null.value = c(45, 30),
  alternative = 'greater')

# Example 5.5.1; Page 150,
prop_CLT(xbar = .4, n = 150L, null.value = .357, alternative = 'greater')

# Example 5.5.2; Page 152,
prop_CLT(xbar = .45, n = 200L, null.value = .51, alternative = 'less')

# Example 5.6.1; Page 155,
prop_CLT(xbar = .1, null.value = c(.28, .21), n = c(100L, 100L), alternative = 'greater')

# Example 5.6.2; Page 155,
prop_CLT(xbar = .05, null.value = c(.34, .26), n = c(250L, 200L), alternative = 'less')

# Example 6.2.1; Page 166 (10th ed), Page 147 (11th ed)
aggregated_z(xbar = 22, n = 10L, sd = sqrt(45))

# Example 6.2.2; Page 168 (10th ed), Page 149 (11th ed)
aggregated_z(xbar = 84.3, n = 15L, sd = sqrt(144), conf.level = .99)
```

```
# Example 6.2.3; Page 168 (10th ed), Page 150 (11th ed)
aggregated_z(xbar = 17.2, n = 35L, sd = 8, conf.level = .9)

# Example 6.2.4; Page 169 (10th ed), Page 150 (11th ed)
head(EXA_C06_S02_04)
aggregated_z(xbar = mean(EXA_C06_S02_04$ACTIVITY), n = nrow(EXA_C06_S02_04), sd = sqrt(.36))

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

# Example 6.4.1; Page 177,
aggregated_z(xbar = c(4.5, 3.4), sd = sqrt(c(1, 1.5)), n = c(12L, 15L))

# Example 6.4.2; Page 178,
aggregated_z(xbar = c(4.3, 13), sd = c(5.22, 8.97), n = c(328L, 64L), conf.level = .99)

# Example 6.4.3; Page 180,
aggregated_t(xbar = c(4.7, 8.8), xsd = c(9.3, 11.5), n = c(18L, 10L), var.equal = TRUE)

# Example 6.4.4; Page 181,
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

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

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

# Example 6.7.1; Page 190,
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))

# Example 6.8.1; Page 192,
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))

# Example 6.9.1; Page 196,
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)))

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

# Example 7.2.1; Page 222 (10th ed); Page 201 (11th ed)
aggregated_z(xbar = 27, sd = sqrt(20), n = 10L, null.value = 30)

# Example 7.2.2; Page 226 (10th ed); Page 204 (11th ed)
```

```
aggregated_z(xbar = 27, sd = sqrt(20), n = 10L, null.value = 30, alternative = 'less')

# Example 7.2.3; Page 228 (10th ed); Page 206 (11th ed)
head(EXA_C07_S02_03)
t.test(EXA_C07_S02_03$DAYS, mu = 15)

# Example 7.2.4; Page 231 (10th ed); Page 209 (11th ed)
aggregated_z(xbar = 146, sd = 27, n = 157L, null.value = 140, alternative = 'greater')

# Example 7.2.5; Page 232 (10th ed); Page 210 (11th ed)
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)

# Example 7.3.1; Page 237 (10th ed), Page 213 (11th ed)
aggregated_z(xbar = c(4.5, 3.4), sd = sqrt(c(1, 1.5)), n = c(12L, 15L), null.value = 0)

# Example 7.3.2; Page 239 (10th ed), Page 215 (11th ed)
head(EXA_C07_S03_02)
with(EXA_C07_S03_02, t.test(x = CONTROL, y = SCI, alternative = 'less', var.equal = TRUE))

# Example 7.3.3; Page 240 (10th ed), Page 217 (11th ed)
aggregated_t(xbar = c(19.16, 9.53), xsd = c(5.29, 2.69), n = c(15L, 30L), null.value = 0)

# Example 7.3.4; Page 242 (10th ed), Page 219 (11th ed)
aggregated_z(xbar = c(59.01, 46.61), sd = c(44.89, 34.85), n = c(53L, 54L), null.value = 0,
  alternative = 'greater')

# Example 7.4.1; Page 251 (10th ed), Page 226 (11th ed)
head(EXA_C07_S04_01)
with(EXA_C07_S04_01, t.test(x = POSTOP, y = PREOP, alternative = 'greater', paired = TRUE))

# Example 7.5.1; Page 258 (10th ed), Page 232 (11th ed)
prop_CLT(x = 24L, n = 301L, null.value = .063, alternative = 'greater')

# Example 7.6.1; Page 261 (10th ed), Page 235 (11th ed)
prop_CLT(x = c(24L, 11L), n = c(44L, 29L), null.value = 0, alternative = 'greater')

# Example 7.7.1; Page 264 (10th ed), Page 238 (11th ed)
head(EXA_C07_S07_01)
aggregated_var(xsd = sd(EXA_C07_S07_01$mass), n = 16L, null.value = 600)

# Example 7.8.1; Page 268 (10th ed), Page 242 (11th ed)
aggregated_var(xsd = c(30.62, 11.37), n = 6L, null.value = 1, alternative = 'greater')

# Example 7.8.2; Page 270,
with(EXA_C07_S03_02, 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,
  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
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

Function 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

Function power_z returns a 'power_z' object, which inherits from 'power.htest' class.

See Also

power.t.test

Examples

```
library(DanielBiostatistics10th)
library(ggplot2)

# Example 7.9.1; Page 272 (10th ed), Page 245 (11th ed)
(p791 = power_z(seq.int(from = 16, to = 19, by = .5), null.value = 17.5, sd = 3.6, n = 100L))
```

```
# Table 7.9.1
autoplot(p791) + labs(title = 'Figure 7.9.2')

# Example 7.9.2; Page 276 (10th ed), Page 248 (11th ed)
(p792 = power_z(seq.int(from = 50, to = 70, by = 5), null.value = 65, sd = 15, n = 20L,
                sig.level = .01, alternative = 'less'))
autoplot(p792) + labs(title = 'Figure 7.9.4')

# Example 7.10.1; Page 278,
(n_d7101 <- uniroot(f = function(x) {
 power_z(55, null.value = 65, sd = 15, n = x, sig.level = .01, alternative = 'less')$power - .95
}, interval = c(0, 50))$root)
power_z(55, null.value = 65, sd = 15, n = ceiling(n_d7101), sig.level = .01, alternative = 'less')
```

Chapter08 *Chapter 8: Analysis of Variance*

Description

Examples for Chapter 8, *Analysis of Variance*.

Value

No function defined for Chapter 8.

Examples

```
library(DanielBiostatistics10th)
library(reshape2)

# Example 8.2.1; Page 318 (10th ed), Page 280 (11th ed)
head(EXA_C08_S02_01)
d821 = within(EXA_C08_S02_01, expr = {
  type = factor(type)
})
boxplot(selenium ~ type, data = d821, main = 'Figure 8.2.7')
(aov_d821 = aov(selenium ~ type, data = d821))
# ?stats::aov # analysis-of-variance model
anova(aov_d821)
# ?stats::anova # ANOVA table

# Example 8.2.2; Page 325 (10th ed), Page 286 (11th ed)
(tukey_d822 <- TukeyHSD(aov_d821, conf.level = 0.95)) # Figure 8.2.8
plot(tukey_d822)

# Example 8.3.1; Page 339 (10th ed), Page 298 (11th ed)
head(EXA_C08_S03_01)
head(d831 <- within(EXA_C08_S03_01, expr = {
 ageGroup = structure(ageGroup, levels = c('<20', '20s', '30s', '40s', '>50'), class = 'factor')
}))
```

```
(aov_831 = aov(time ~ method + ageGroup, data = d831))
anova(aov_831)

# Example 8.4.1; Page 348 (10th ed), Page 307 (11th ed)
head(EXA_C08_S04_01)
head(d841 <- within(EXA_C08_S04_01, expr = {
  SUBJ = factor(SUBJ)
 TIME = structure(TIME, levels = c('Baseline', '1-Mon', '3-Mon', '6-Mon'), class = 'factor')
}))
(aov_841 = aov(FUNC ~ SUBJ + TIME, data = d841))
anova(aov_841)

# Example 8.4.2; Page 352 (10th ed), Page 310 (11th ed)
# (optional; out of the scope of this course)
head(EXA_C08_S04_02)
names(EXA_C08_S04_02)[3:6] = c('baseline', '2wk', '4wk', '6wk')
head(d842a <- within(EXA_C08_S04_02, expr = {
  subject = factor(subject)
  treatment = structure(treatment, levels = c('placebo', 'aloe_juice'), class = 'factor')
}))
head(d842b <- reshape2::melt(d842a, id.vars = c('subject', 'treatment'),
                             variable.name = 'time', value.name = 'OralScores'))
# Hypothesis:
# Main effect of 'treatment';
# Main effect of 'time';
# Interaction between 'treatment' and 'time'
(aov_842 = aov(OralScores ~ treatment * time + Error(subject), data = d842b))
class(aov_842)
summary(aov_842)
# Section 'Error: subject' in R output
# .. is Figure 8.4.4 'Tests of Between-Subjects Effects' (without the row of 'Intercept')
# .. 'treatment' row: effect of treatment at **baseline** (i.e. reference time),
# ... degree-of-freedom (dof) = 2-1 = 1
# .. 'Residuals' row: residual at baseline, dof = (25-1) - (2-1) = 23
# .. It's important to note that 'treatment' is a **between-subject factor**.
# Section 'Error: Within' in R output
# .. is Figure 8.4.4 'Tests of Within-Subjects Effects'
# .. 'time' row: effect of time within subject for placebo (i.e. reference treatment), dof = 4-1 = 3
# .. 'treatment:time' row: interation of treatment and time, dof = (2-1)*(4-1) = 3
# .. 'Residuals' row: residual at 2wk, 4wk and 6wk, dof = (4-1)*23 = 69
# ... [(4-1) timepoints, 23 dof at each timepoints]
# Analysis Interpretation
# .. No signif. diff. detected between placebo vs. aloe at baseline (p = .815)
# .. No signif. diff. detected in the trends over time between placebo vs. aloe (p = .974)
# .. Signif. diff. detected among the four timepoints, for either placebo or aloe pts (p = 3e-7)
# R code below creates an equivalent ANOVA model
anova(aov(OralScores ~ treatment * time + subject, data = d842b))
# .. 'subject' is considered as a block factor

# Example 8.5.2; Page 364 (10th ed), Page 321 (11th ed)
head(EXA_C08_S05_02)
head(d852 <- within(EXA_C08_S05_02, expr = {
 A = structure(A, levels = c('Cardiac', 'Cancer', 'CVA', 'Tuberculosis'), class = 'factor')
```

```
  B = structure(B, levels = c('20s', '30s', '40s', '50+'), class = 'factor')
}))
(aov_852 = aov(HOME ~ A * B, data = d852))
anova(aov_852)
summary(lm(HOME ~ A * B, data = d852))
# produces alpha, beta and (alpha beta)'s in the formulation
```

Chapter09 *Chapter 9: Simple Linear Regression and Correlation*

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

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

See Also

predict.lm

Examples

```
library(DanielBiostatistics10th)
library(ggplot2)

# Example 9.3.1; Page 417 (10th ed), Page 358 (11th ed)
head(EXA_C09_S03_01)
names(EXA_C09_S03_01)[2:3] = c('Waist', 'AT')
plot(AT ~ Waist, data = EXA_C09_S03_01, xlab = 'Waist circumference (cm), X',
     ylab = 'Deep abdominal AT area (cm2), Y', main = 'Figure 9.3.1')

# Example 9.4.1-9.4.2; Page 432-436 (10th ed), Page 372-375 (11th ed)
summary(m931 <- lm(AT ~ Waist, data = EXA_C09_S03_01))
```

```
cor(EXA_C09_S03_01[2:3]); cor.test(~ AT + Waist, data = EXA_C09_S03_01)
confint(m931) # confidence interval of regression coefficients
anova(m931)

# (omitted) Example 9.4.3; Page 376 (11th ed)

# Example 9.4.3; Page 440 (10th ed)
# Example 9.4.4; Page 379 (11th ed)
plot(m931, which = 1, main = 'Figure 9.4.8 (10th) or 9.4.9 (11th)')

# Section 9.5; Page 441 (10th ed), Page 380 (11th ed)
autoplot(predict_lm(m931)) + labs(
  xlab = 'Waist circumference (cm), X', ylab = 'Deep abdominal AT area (cm2), Y',
  title = 'Figure 9.5.1')

# Example 9.7.1; Page 447 (10th ed), Page 386 (11th ed)
head(EXA_C09_S07_01)
summary(mod_971 <- lm(CV ~ HEIGHT, data = EXA_C09_S07_01))
autoplot(predict_lm(mod_971)) + labs(
 xlab = 'Height (cm)', ylab = 'Cv (units)', title = 'Figure 9.7.2 (10th ed); 9.7.1 (11th ed)')

# Example 9.7.2; Page 452 (10th ed), Page 390 (11th ed)
cor(EXA_C09_S07_01); cor.test(~ CV + HEIGHT, data = EXA_C09_S07_01) # Figure 9.7.4, 9.7.5

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

`Chapter10` *Chapter 10: Multiple Regression and Correlation*

Description

Examples for Chapter 10, *Multiple Regression and Correlation*.

Value

No function defined for Chapter 10.

Examples

```
library(DanielBiostatistics10th)

# Example 10.3.1; Page 493 (10th ed), Page 419 (11th ed)
head(EXA_C10_S03_01)
pairs(EXA_C10_S03_01, main = 'Figure 10.3.1')
summary(mod_1031 <- lm(CDA ~ AGE + EDLEVEL, data = EXA_C10_S03_01))

# Example 10.4.1; Page 502 (10th ed), Page 428 (11th ed)
# .. see 'Multiple R-squared' (not 'Adjusted R-squared')
```

```
# Example 10.4.2; Page 504 (10th ed), Page 429 (11th ed)
# .. see 'F-statistic'

# Example 10.4.3; Page 505 (10th ed), Page 430 (11th ed)
# .. see 'Coefficients:'

# confidence interval for beta's; Page 506 (10th ed), Page 431 (11th ed)
confint(mod_1031)

# Example 10.5.1; Page 509 (10th ed), Page 434 (11th ed)
(newd_1031 = data.frame(AGE = 68, EDLEVEL = 12))
predict(mod_1031, newdata = newd_1031, interval = 'prediction')
predict(mod_1031, newdata = newd_1031, interval = 'confidence')

# Example 10.6.1; Page 511 (10th ed), Page 436 (11th ed)
head(EXA_C10_S06_01)
pairs(EXA_C10_S06_01, main = 'Scatter Plot Matrix of Example 10.6.1')
summary(mod_1061 <- lm(W ~ P + S, data = EXA_C10_S06_01))
confint(mod_1061)

# Example 10.6.2; Page 515 (10th ed), Page 440 (11th ed)
psych::partial.r(EXA_C10_S06_01, x = 2:3, y = 1L)
```

`Chapter11` *Chapter 11: Regression Analysis: Some Additional Techniques*

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

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

See Also

predict.glm

Examples

```
library(DanielBiostatistics10th)
library(ggplot2)
library(car)
library(DescTools)

# Example 11.1.1; Page 540,
head(EXA_C11_S01_01)
head(log(EXA_C11_S01_01$conc, base = 10))
head(EXA_C11_S01_01$logConc)

# Example 11.1.2; Page 542,
head(EXA_C11_S01_02)
cor.test(~ sbp + weight, data = EXA_C11_S01_02)
cor.test(~ sbp + bmi, data = EXA_C11_S01_02)

# Example 11.2.1; Page 545,
head(EXA_C11_S02_01)
d1121 = within(EXA_C11_S02_01, expr = {
  SMOKE = as.logical(SMOKE)
})
xlab1121 = 'Length of gestation (weeks)'; ylab1121 = 'Birth weight (grams)'
car::scatterplot(GRAMS ~ WEEKS | SMOKE, data = d1121, regLine = FALSE, smooth = FALSE,
                 xlab = xlab1121, ylab = ylab1121, main = 'Figure 11.2.1')
summary(m1121_main <- lm(GRAMS ~ WEEKS + SMOKE, data = d1121)) # Figure 11.2.2
confint(m1121_main)
car::scatterplot(GRAMS ~ WEEKS | SMOKE, data = d1121, regLine = FALSE, smooth = FALSE,
                 xlab = xlab1121, ylab = ylab1121, main = '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')

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

# Example 11.3.1; Page 561,
head(EXA_C11_S03_01)
names(EXA_C11_S03_01) = c('JOBPER', 'ASRV', 'ENTH', 'AMB', 'COMM', 'PROB', 'INIT')
summary(mod_1131_raw <- lm(JOBPER ~ ASRV + ENTH + AMB + COMM + PROB + INIT, data = EXA_C11_S03_01))
# summary(mod_1131 <- MASS::stepAIC(mod_1131_raw, direction = 'backward'))
# the stepwise selection criterion used in MINITAB is not necessarily AIC
```

```
# Example 11.4.1; Page 572,
addmargins(d1141 <- array(c(92L, 21L, 15L, 20L), dim = c(2L, 2L), dimnames = list(
  OCAD = c('Present', 'Absent'), Sex = c('Male', 'Female')))) # Table 11.4.2
(d1141a = within(as.data.frame(as.table(d1141)), expr = {
  OCAD = (OCAD == 'Present')
  Sex = factor(Sex, levels = c('Female', 'Male'))
}))
summary(m1141 <- glm(OCAD ~ Sex, family = binomial, weights = Freq, data = d1141a)) # 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')

# Example 11.4.2; Page 573,
head(EXA_C11_S04_02)
summary(mod_1142 <- glm(ATT ~ AGE, family = binomial, data = EXA_C11_S04_02)) # 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))) + labs(title = 'Figure 11.4.3')

# Example 11.4.3; Page 576,
head(REV_C11_24)
summary(glm(ONSET ~ HIAA + TRYPT, family = binomial, data = REV_C11_24)) # Figure 11.4.4
# Predictor TRYPT should be removed from model due to p-value \approx 1
summary(glm(ONSET ~ HIAA, family = binomial, data = REV_C11_24))

# Example 11.4.4-11.4.5; Page 578-579
DescTools::PseudoR2(mod_1142, which = 'CoxSnell')
DescTools::PseudoR2(mod_1142, which = 'Nagelkerke')
```

`Chapter12` *Chapter 12: χ^2 Distribution and the Analysis of Frequencies*

Description

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

Usage

```
print_OE(O, prob)
```

Arguments

`O`	integer vector, observed counts
`prob`	numeric vector, anticipated probability. If missing (default), an uniform distribution across all categories are used.

Value

Function 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.

Examples

```
library(DanielBiostatistics10th)

# Example 12.3.1; Page 605,
d1231_b = c(-Inf, seq.int(from = 125, to = 275, by = 25), Inf)
(d1231 = setNames( # 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

# Example 12.3.2; Page 609,
# 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

# Example 12.3.3; Page 611,
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)

# Example 12.3.4; Page 614 (10th ed), Page 531 (11th ed)
d1234 = c('Dec 05' = 62L, 'Jan 06' = 84L, 'Feb 06' = 17L, 'Mar 06' = 16L, 'Apr 06' = 21L)
print_OE(d1234) # Figure 12.3.2
chisq.test(d1234)

# Example 12.3.5; Page 616 (10th ed), Page 533 (11th ed)
d1235 = c(dominant = 43L, heterozygous = 125L, recessive = 32L)
print_OE(d1235, prob = c(1, 2, 1))
chisq.test(d1235, p = c(1, 2, 1), rescale.p = TRUE)

# Example 12.4.1; Page 621
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

# Example 12.4.2; Page 626,
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) # Yates's Correction

# Example 12.5.1; Page 631,
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))
}

# Example 12.6.1; Page 638,
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)

# Example 12.7.1; Page 644,
(d1271 = array(c(22L, 18L, 216L, 199L), dim = c(2L, 2L), dimnames = list(
  Exercising = c('Extreme', 'No'), PretermLabor = c('TRUE', 'FALSE'))))
summary(BooleanTable(t(d1271)))
# textbook confidence interval (.65, 1.86) wrong (too many rounding in intermediate steps)

# Example 12.7.2; Page 647,
(d1272 = array(c(64L, 68L, 342L, 3496L), dim = c(2L, 2L), dimnames = list(
  SmkPregnancy = c('TRUE', 'FALSE'), Obesity = c('TRUE', 'FALSE'))))
summary(BooleanTable(t(d1272)))

# Example 12.7.3-12.7.4; Page 650-652,
(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) # Table 12.7.6
mantelhaen.test(d1273)
```

Chapter13 *Chapter 13: Nonparametric and Distribution-Free Statistics*

Description

Examples for Chapter 13, *Nonparametric and Distribution-Free Statistics*.

Value

No function defined for Chapter 13.

Examples

```
library(DanielBiostatistics10th)
library(reshape2); packageDate('reshape2')
library(BSDA); packageDate('BSDA')
library(robslopes); packageDate('robslopes')
library(mblm); packageDate('mblm')

# Example 13.3.1; Page 673,
d1331 = c(4, 5, 8, 8, 9, 6, 10, 7, 6, 6)
BSDA::SIGN.test(d1331, md = 5)

# Example 13.3.2; Page 677,
head(EXA_C13_S03_02)
with(EXA_C13_S03_02, BSDA::SIGN.test(x = X, y = Y, alternative = 'less'))

# Example 13.4.1; Page 683,
d1341 = c(4.91, 4.10, 6.74, 7.27, 7.42, 7.50, 6.56, 4.64, 5.98, 3.14, 3.23, 5.80, 6.17, 5.39, 5.77)
wilcox.test(d1341, mu = 5.05)

# Example 13.5.1; Page 686,
head(EXA_C13_S05_01)
(med1351 = median(unlist(EXA_C13_S05_01), na.rm = TRUE)) # common median
addmargins(t1351 <- with(EXA_C13_S05_01, expr = {
  tmp <- cbind(Urban = table(URBAN < med1351), Rural = table(RURAL < med1351, useNA = 'no'))
  rownames(tmp) <- paste('Number of scores', c('above', 'below'), 'median')
  tmp
})) # Table 13.5.2
chisq.test(t1351, correct = FALSE)

# Example 13.6.1; Page 691,
head(EXA_C13_S06_01)
with(EXA_C13_S06_01, wilcox.test(X, Y, exact = FALSE, alternative = 'less'))

# Example 13.7.1; Page 699,
head(EXA_C13_S07_01)
ks.test(EXA_C13_S07_01$GLUCOSE, y = pnorm, mean = 80, sd = 6)

# Example 13.8.1; Page 705 (10th ed), Page 611 (11th ed)
(d1381 = data.frame(Air = c(12.22, 28.44, 28.13, 38.69, 54.91),
                    Benzaldehyde = c(3.68, 4.05, 6.47, 21.12, 3.33),
                    Acetaldehyde = c(54.36, 27.87, 66.81, 46.27, 30.19))) # Table 13.8.1
with(melt(d1381, id.vars = NULL, value.name = 'Eosinophil', variable.name = 'Exposure'),
     expr = kruskal.test(x = Eosinophil, g = Exposure))

# Example 13.8.2; Page 708 (10th ed), Page 613 (11th ed)
head(EXA_C13_S08_02)
with(melt(EXA_C13_S08_02, id.vars = NULL, value.name = 'Value', variable.name = 'Hospital'),
     expr = kruskal.test(x = Value, g = Hospital))
```

```
# Example 13.9.1; Page 713 (10th ed); Page 618 (11th ed)
head(EXA_C13_S09_01)
m1391 = as.matrix(EXA_C13_S09_01[LETTERS[1:3]], rownames.force = TRUE)
names(dimnames(m1391)) = c('Therapist', 'Model')
m1391 # Table 13.9.1
friedman.test(m1391)

# Example 13.9.2; Page 715 (10th ed); Page 620 (11th ed)
head(EXA_C13_S09_02)
m1392 = as.matrix(EXA_C13_S09_02[LETTERS[1:4]], rownames.force = TRUE)
names(dimnames(m1392)) = c('Animal', 'Dose')
m1392 # Table 13.9.2
friedman.test(m1392)

# Example 13.10.1; Page 720,
head(EXA_C13_S10_01)
with(EXA_C13_S10_01, cor.test(X, Y, method = 'spearman'))

# Example 13.10.2; Page 722,
head(EXA_C13_S10_02)
with(EXA_C13_S10_02, cor.test(V2, V3, method = 'spearman', exact = FALSE))

# Example 13.11.1-13.11.2; Page 728-729,
testosterone <- c(230, 175, 315, 290, 275, 150, 360, 425)
citricAcid <- c(421, 278, 618, 482, 465, 105, 550, 750)
robslopes::TheilSen(x = citricAcid, y = testosterone)
# textbook uses *central* median of ordered pairs, while ?robslopes::TheilSen uses *upper* median
summary(mblm::mblm(testosterone ~ citricAcid, repeated = FALSE)) # Figure 13.11.1 (11th ed)
```

Chapter14 *Chapter 14: Survival Analysis*

Description

Examples for Chapter 14, *Survival Analysis*.

Value

No function defined for Chapter 14.

Examples

```
library(DanielBiostatistics10th)
library(survival)
library(ggplot2)
library(ggfortify)

# Example 14.3.1; Page 756 (10th ed), Page 652 (11th ed)
```

```
head(EXA_C14_S03_01)
head(d1431 <- within(EXA_C14_S03_01, expr = {
  TIME = .difftime(TIME, units = 'months')
 OS = Surv(TIME, event = (VITAL != 'ned')) # ?survival::Surv # create a time-to-event variable
}))
class(d1431$OS) # 'Surv'
head(d1431$OS)
summary(sf_1431 <- survfit(OS ~ TUMOR, data = d1431)) # Table 14.3.2
# ?survival::survfit # Kaplan-Meier estimates for the survival function
autoplot(sf_1431) + xlim(0, 120) + labs(title = 'Figure 14.3.1')
# ?ggfortify:::autoplot.survfit # to plot the Kaplan-Meier curves

# Example 14.4.1; Page 764 (10th ed), Page 659 (11th ed)
survdiff(OS ~ TUMOR, data = d1431, rho = 0)
# ?survival::survdiff(..., rho = 0) # log rank test to compare survival functions

# Example 14.5.1; Page 769 (10th ed), Page 663 (11th ed)
head(EXA_C14_S05_01)
head(d1451 <- within(EXA_C14_S05_01, expr = {
  time = .difftime(time, units = 'weeks')
  PFS = Surv(time, status)
 drug = relevel(structure(drug, levels = c('Opiate', 'Other'), class = 'factor'), ref = 'Other')
}))
summary(model1_1451 <- coxph(PFS ~ drug + age, data = d1451))
# ?survival::coxph # Cox proportional hazard model
# 'Opiate' has higher hazard compared to Drug='Other' (hazard ratio (HR) = 9.407, p < .001)
# With 'proportional hazard' assumption,
# ... we don't need to discuss Opiate vs. Other at any specific time point
confint(model1_1451)
# 'age' is not significant (p = .772), so it should be removed from the model
summary(model2_1451 <- coxph(PFS ~ drug, data = d1451))
confint(model2_1451)
# 'Opiate' has higher hazard compared to Drug='Other' (hazard ratio (HR) = 9.923, p < .001)
autoplot(survfit(PFS ~ drug, data = d1451)) + labs(title = 'Figure 14.5.1')
```

Index