Package 'seqtest'

October 14, 2022

Type Package

Title Sequential Triangular Test

Version 0.1-0

Date 2016-03-26

Author Takuya Yanagida [cre, aut]

Maintainer Takuya Yanagida <takuya.yanagida@univie.ac.at>

Description Sequential triangular test for the arithmetic mean in one- and two-
samples, proportions in one- and two-samples, and the Pearson's correlation
coefficient.

License GPL-3

LazyLoad yes

LazyData true

Depends R (>= 3.2.0)

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-26 17:09:02

R topics documented:

Index

---

descript   *Print descriptive statistics*

Description

This function prints descriptive statistics for the seqtest object

Usage

```
descript(x, digits = 2, output = TRUE)
```

Arguments

x	seqtest object.
digits	integer indicating the number of decimal places to be displayed.
output	logical: if TRUE, output is shown.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \leq \rho 0$. *Statistical Papers, 56*, 689-699.

See Also

[seqtest.mean](), [seqtest.prop](), [seqtest.cor](), [plot.seqtest](), [descript]()

Examples

```
#-------------------------------------
# Sequential triangular test for the arithmetic mean in one sample

seq.obj <- seqtest.mean(56, mu = 50, theta = 0.5,
                        alpha = 0.05, beta = 0.2)

seq.obj <- update(seq.obj, x = c(54, 52, 46, 49))

descript(seq.obj)

#-------------------------------------
# Sequential triangular test for the proportion in one sample

seq.obj <- seqtest.prop(c(1, 1, 0, 1), pi = 0.5, delta = 0.2,
                        alpha = 0.05, beta = 0.2)

seq.obj <- update(seq.obj, x = c(1, 1, 1, 1, 1, 0, 1, 1, 1))

descript(seq.obj)

#-------------------------------------
# Sequential triangular test for Pearson's correlation coefficient

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2,
                       alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, c(0.56, 0.76, 0.56, 0.52))

descript(seq.obj)
```

---

plot.seqtest *Plot seqtest*

Description

This function plots the seqtest object

Usage

```
## S3 method for class 'seqtest'
plot(x, ...)
```

Arguments

x   seqtest object

...   further arguments passed to or from other methods

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \leq \rho0$. *Statistical Papers, 56*, 689-699.

See Also

seqtest.mean, seqtest.prop, seqtest.cor, print.seqtest, descript

Examples

```
#-------------------------------------
# Sequential triangular test for the arithmetic mean in one sample

seq.obj <- seqtest.mean(56, mu = 50, theta = 0.5,
                        alpha = 0.05, beta = 0.2)
plot(seq.obj)

#-------------------------------------
# Sequential triangular test for the proportion in one sample

seq.obj <- seqtest.prop(c(1, 1, 0, 1), pi = 0.5, delta = 0.2,
                        alpha = 0.05, beta = 0.2)
plot(seq.obj)

#-------------------------------------
# Sequential triangular test for Pearson's correlation coefficient

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2,
                       alpha = 0.05, beta = 0.2)

plot(seq.obj)
```

---

plot.sim.seqtest.cor   *Plot sim.seqtest*

Description

This function plots the sim.seqtest.cor object

Usage

```
## S3 method for class 'sim.seqtest.cor'
plot(x, plot.lines = TRUE, plot.nom = TRUE,
     ylim = NULL, type = "b", pch = 19, lty = 1, lwd = 1, ...)
```

Arguments

x	sim.seqtest.cor object.
plot.lines	plot lines connecting points withe the x- and y-axis.
plot.nom	plot line at the nominal alpha.
ylim	the y limits of the plot.
type	what type of plot should be drawn ("p" for points, "l" for lines and "b" for both).
pch	plotting character.
lty	line type.
lwd	line width.
...	further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \le \rho 0$. *Statistical Papers, 56*, 689-699.

See Also

sim.seqtest.cor, seqtest.cor

Examples

```
## Not run:

#-------------------------------------------
# Determine optimal k and nominal type-II-risk
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.01, beta = 0.05, delta = 0.25

# Step 1: Determine the optimal size of subsamples (k)

sim.obj.1 <- sim.seqtest.cor(rho.sim = 0.3, k = seq(4, 16, by = 1), rho = 0.3,
                             alternative = "greater",
                             delta = 0.25, alpha = 0.05, beta = 0.05,
                             runs = 10000)

plot(sim.obj.1)
```

```
# Step 2: Determine the optimal nominal type-II-risk based on
#         the optimal size of subsamples (k) from step 1

sim.obj.2 <- sim.seqtest.cor(rho.sim = 0.55, k = 16, rho = 0.3,
                             alternative = "greater",
                           delta = 0.25, alpha = 0.05, beta = seq(0.05, 0.15, by = 0.01),
                             runs = 10000)

plot(sim.obj.2)

## End(Not run)
```

---

print.seqtest   *Print seqtest*

Description

This function prints the seqtest object

Usage

```
## S3 method for class 'seqtest'
print(x, ...)
```

Arguments

x	seqtest object.
...	further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \le \rho0$. *Statistical Papers, 56*, 689-699.

See Also

seqtest.mean, seqtest.prop, seqtest.cor, plot.seqtest, descript

Examples

```
#-------------------------------------
# Sequential triangular test for the arithmetic mean in one sample

seq.obj <- seqtest.mean(56, mu = 50, theta = 0.5,
                        alpha = 0.05, beta = 0.2, output = FALSE)

print(seq.obj)

#-------------------------------------
# Sequential triangular test for the proportion in one sample

seq.obj <- seqtest.prop(c(1, 1, 0, 1), pi = 0.5, delta = 0.2,
                        alpha = 0.05, beta = 0.2, output = FALSE)

print(seq.obj)

#-------------------------------------
# Sequential triangular test for Pearson's correlation coefficient

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2,
                       alpha = 0.05, beta = 0.2, output = FALSE)

print(seq.obj)
```

---

  print.sim.seqtest.cor *Print sim.seqtest*

Description

This function prints the `sim.seqtest.cor` object

Usage

```
## S3 method for class 'sim.seqtest.cor'
print(x, ...)
```

Arguments

x   `sim.seqtest.cor` object.

...   further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \leq \rho0$. *Statistical Papers, 56*, 689-699.

See Also

[sim.seqtest.cor](), [plot.sim.seqtest.cor]()

Examples

```
## Not run:

#-------------------------------------------
# Determine optimal k and nominal type-II-risk
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.01, beta = 0.05, delta = 0.25

# Step 1: Determine the optimal size of subsamples (k)

sim.obj <- sim.seqtest.cor(rho.sim = 0.3, k = seq(4, 16, by = 1), rho = 0.3,
                           alternative = "greater",
                           delta = 0.25, alpha = 0.05, beta = 0.05,
                           runs = 10000, output = FALSE)

print(sim.obj)

# Step 2: Determine the optimal nominal type-II-risk based on
#         the optimal size of subsamples (k) from step 1

sim.obj <- sim.seqtest.cor(rho.sim = 0.55, k = 16, rho = 0.3,
                           alternative = "greater",
                           delta = 0.25, alpha = 0.05, beta = seq(0.05, 0.15, by = 0.01),
                           runs = 10000, output = FALSE)

print(sim.obj)

## End(Not run)
```

---

print.size   *Print size object*

Description

This function prints the `size` object

Usage

```
## S3 method for class 'size'
print(x, ...)
```

Arguments

x	size object.
...	further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

See Also

size.mean, size.prop, size.cor

Examples

```
#------------------------------------
# Two-sided one-sample test
# theta = 0.5
# alpha = 0.05, beta = 0.2

n <- size.mean(theta = 0.5, sample = "one.sample",
               alternative = "two.sided", alpha = 0.05, beta = 0.2)

print(n)

#------------------------------------
# Two-sided one-sample test
# H0: pi = 0.5, H1: pi != 0.5
# alpha = 0.05, beta = 0.2, delta = 0.2

n <- size.prop(delta = 0.2, pi = 0.5, sample = "one.sample",
               alternative = "two.sided", alpha = 0.05, beta = 0.2)

print(n)

#------------------------------------
# H0: rho = 0.3, H1: rho != 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2

n <- size.cor(delta = 0.2, rho = 0.3, alpha = 0.05, beta = 0.2)

print(n)
```

seqtest.cor	*Sequential triangular test for Pearson's correlation coefficient*

Description

This function performs the sequential triangular test for Pearson's correlation coefficient

Usage

```
seqtest.cor(x, k, rho, alternative = c("two.sided", "less", "greater"),
            delta, alpha = 0.05, beta = 0.1, output = TRUE, plot = FALSE)
```

Arguments

x	initial data, i.e., Pearson's correlation coefficient in a sub-sample of k observations.
k	number of observations in each sub-sample.
rho	a number indicating the correlation coefficient under the null hypothesis, $\rho.0$.
alternative	a character string specifying the alternative hypothesis,
delta	minimum difference to be detected, δ.
alpha	type-I-risk, α.
beta	type-II-risk, β.
output	logical: if TRUE, output is shown.
plot	logical: if TRUE, an initial plot is generated.

Details

Null and alternative hypothesis is specified using arguments rho and delta. Note that the argument k (i.e., number of observations in each sub-sample) has to be specified. At least k = 4 is needed. The optimal value of k should be determined based on statistical simulation using sim.seqtest.cor function.

In order to specify a one-sided test, argument alternative has to be used (i.e., two-sided tests are conducted by default). That is, alternative = "less" specifies the null hypothesis, H0: $\rho \geq \rho.0$ and the alternative hypothesis, H1: $\rho < \rho.0$; alternative = "greater" specifies the null hypothesis, H0: $\rho \leq \rho.0$ and the alternative hypothesis, H1: $\rho > \rho.0$.

The main characteristic of the sequential triangular test is that there is no fixed sample size given in advance. That is, for the most recent sampling point, one has to decide whether sampling has to be continued or either the null- or the alternative hypothesis can be accepted given specified precision requirements (i.e. type-I-risk, type-II-risk and an effect size). The sequence of data pairs must we split into sub-samples of length k >= 4 each. The (cumulative) test statistic Z.m on a Cartesian coordinate system produces a "sequential path" on a continuation area as a triangle. As long as the statistic remains within that triangle, additional data have to be sampled. If the path touches or exceeds the borderlines of the triangle, sampling is completed. Depending on the particular borderline, the null-hypothesis is either accepted or rejected.

Value

Returns an object of class seqtest, to be used for later update steps. The object has following entries:

call	function call
type	type of the test (i.e., correlation coefficient)
spec	specification of function arguments
tri	specification of triangular
dat	data
res	list with results

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \leq \rho0$. *Statistical Papers, 56*, 689-699.

See Also

update.seqtest, sim.seqtest.cor, seqtest.mean, seqtest.prop, print.seqtest, plot.seqtest, descript

Examples

```
#-------------------------------------
# H0: rho = 0.3, H1: rho != 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2,
                       alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, c(0.56, 0.76, 0.56, 0.52))

#-------------------------------------
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3,
                       alternative = "greater", delta = 0.2,
                       alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, c(0.56, 0.76, 0.66))
```

---

seqtest.mean	*Sequential triangular test for the arithmetic mean*

Description

This function performs the sequential triangular test for the arithmetic mean in one- or two-samples

Usage

```
seqtest.mean(x, y = NULL, mu = NULL, alternative = c("two.sided", "less", "greater"),
             sigma = NULL, delta = NULL, theta = NULL, alpha = 0.05, beta = 0.1,
             output = TRUE, plot = FALSE)
```

Arguments

x	initial data for group x, at least one entry.
y	initial data for group y, at least one entry for a two-sample test.
mu	a number indicating the true value of the mean in case of the one-sample test, $\mu.0$.
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
sigma	standard deviation in the population, σ.
delta	absolute minimum difference to be detected, δ.
theta	relative minimum difference to be detected, θ.
alpha	type-I-risk, α.
beta	type-II-risk, β.
output	logical: if TRUE, output is shown.
plot	logical: if TRUE, a plot is generated.

Details

For the one-sample test, arguments x, mu and the minimum difference to be detected has to be specified (i.e., argument y must not be specified). For the two-sample test, arguments x, y, and the minimum difference to be detected has to be specified. There are two options to specify the minimum difference to be detected: (1) using arguments mu, sigma and delta or (2) using arguments mu and theta. Note that it is not a requirement to know sigma in advance, i.e., theta can be specified directly. For example, theta = 1 specifies a relative minimum difference to be detected of one standard deviation.

In order to specify a one-sided test, argument alternative has to be used (i.e., two-sided tests are conducted by default). For the one-sample test, alternative = "less" specifies the null hypothesis, H0: $\mu >= \mu.0$ and the alternative hypothesis, H1: $\mu < \mu.0$; alternative = "greater" specifies the null hypothesis, H0: $\mu <= \mu.0$ and the alternative hypothesis, H1: $\mu > \mu.0$. For the two-sample

test alternative = "less" specifies the null hypothesis, H0: $\mu.1 >= \mu.2$ and the alternative hypothesis, H1: $\mu.1 < \mu.2$; alternative = "greater" specifies the null hypothesis, H0: $\mu.1 <= \mu.2$ and the alternative hypothesis, H1: $\mu.1 > \mu.2$.

The main characteristic of the sequential triangular test is that there is no fixed sample size given in advance. That is, for the most recent sampling point, one has to decide whether sampling has to be continued or either the null- or the alternative hypothesis can be accepted given specified precision requirements (i.e. type-I-risk, type-II-risk and a minimum difference to be detected). The (cumulative) test statistic Z.m on a Cartesian coordinate system produces a "sequential path" on a continuation area as a triangle. As long as the statistic remains within that triangle, additional data have to be sampled. If the path touches or exceeds the borderlines of the triangle, sampling is completed. Depending on the particular borderline, the null-hypothesis is either accepted or rejected.

Value

Returns an object of class seqtest, to be used for later update steps. The object has following entries:

call	function call
type	type of the test (i.e., arithmetic mean)
spec	specification of function arguments
tri	specification of the triangular
dat	data
res	list with results

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

See Also

update.seqtest, seqtest.prop, seqtest.cor, print.seqtest, plot.seqtest, descript

Examples

```
#-------------------------------------
# Two-sided one-sample test
# H0: mu = 50, H1: mu != 50
# alpha = 0.05, beta = 0.2, theta = 0.5
```

```
seq.obj <- seqtest.mean(56, mu = 50, theta = 0.5,
                        alpha = 0.05, beta = 0.2, plot = TRUE)

# alternative specifiation using sigma and delta
seq.obj <- seqtest.mean(56, mu = 50, sigma = 10, delta = 5,
                        alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, x = c(54, 52, 46, 49))
seq.obj <- update(seq.obj, x = c(46, 49, 51, 45))
seq.obj <- update(seq.obj, x = c(51, 42, 50, 53))
seq.obj <- update(seq.obj, x = c(50, 53, 49, 53))

#------------------------------------
# One-sided one-sample test
# H0: mu <= 50, H1: mu > 50
# alpha = 0.05, beta = 0.2, theta = 0.5

seq.obj <- seqtest.mean(c(56, 53), mu = 50, alternative = "greater",
                        theta = 0.5, alpha = 0.05, beta = 0.2, plot = TRUE)

# alternative specifiation using sigma and delta
seq.obj <- seqtest.mean(c(56, 53), mu = 50, alternative = "greater",
                        sigma = 10, delta = 5, alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, x = c(67, 52, 48, 59))
seq.obj <- update(seq.obj, x = c(53, 57, 54, 62))
seq.obj <- update(seq.obj, x = 58)

#------------------------------------
# Two-sided two-sample test
# H0: mu.1 = mu.2, H1: mu.1 != mu.2
# alpha = 0.01, beta = 0.1, theta = 1

seq.obj <- seqtest.mean(53, 45, theta = 1,
                        alpha = 0.01, beta = 0.01, plot = TRUE)

# alternative specifiation using sigma and delta
seq.obj <- seqtest.mean(57, 45, sigma = 10, delta = 10,
                        alpha = 0.01, beta = 0.01, plot = TRUE)

seq.obj <- update(seq.obj, x = c(58, 54, 56), y = c(45, 41, 42))
seq.obj <- update(seq.obj, x = c(56, 50, 49), y = c(42, 45, 50))
seq.obj <- update(seq.obj, x = c(62, 57, 59))
seq.obj <- update(seq.obj, y = c(41, 39, 46))
seq.obj <- update(seq.obj, x = 67)
seq.obj <- update(seq.obj, y = 40)
seq.obj <- update(seq.obj, y = 36)

#------------------------------------
# One-sided two-sample test
# H0: mu.1 <= mu.2, H1: mu.1 > mu.2
# alpha = 0.01, beta = 0.1, theta = 1
```

```
seq.obj <- seqtest.mean(53, 45, alternative = "greater", theta = 1,
                          alpha = 0.01, beta = 0.01, plot = TRUE)

# alternative specifiation using sigma and delta
seq.obj <- seqtest.mean(57, 45, alternative = "greater",sigma = 10, delta = 10,
                          alpha = 0.01, beta = 0.01, plot = TRUE)

seq.obj <- update(seq.obj, x = c(58, 54, 56), y = c(45, 41, 42))
seq.obj <- update(seq.obj, x = c(56, 50, 49), y = c(42, 45, 50))
seq.obj <- update(seq.obj, x = c(62, 57, 59))
seq.obj <- update(seq.obj, y = c(41, 39, 46))
```

---

seqtest.prop   *Sequential triangular test for the proportion*

Description

This function performs the sequential triangular test for the proportion in one- or two-samples

Usage

```
seqtest.prop(x, y = NULL, pi = NULL, alternative = c("two.sided", "less", "greater"),
              delta, alpha = 0.05, beta = 0.1, output = TRUE, plot = FALSE)
```

Arguments

x	initial data for group x, at least one entry.
y	initial data for group y, at least one entry for a two-sample test.
pi	a number indicating the true value of the probability of success in group x, $\pi.0$.
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "less" or "greater".
delta	minimum difference to be detected, δ.
alpha	type-I-risk, α.
beta	type-II-risk, β.
output	logical: if TRUE, output is shown.
plot	logical: if TRUE, a plot is generated.

Details

For the one-sample test, arguments x, pi, and delta has to be specified (i.e., argument y must not be specified). For the two-sample test, arguments x, y, pi, and delta has to be specified

In order to specify a one-sided test, argument alternative has to be used (i.e., two-sided tests are conducted by default). For the one-sample test, alternative = "less" specifies the null hypothesis, H0: $\pi \geq \pi.0$ and the alternative hypothesis, H1: $\pi < \pi.0$; alternative = "greater" specifies the null hypothesis, H0: $\pi \leq \pi.0$ and the alternative hypothesis, H1: $\pi > \pi.0$. For the two-sample

test `alternative = "less"` specifies the null hypothesis, H0: $\pi.1 >= \pi.2$ and the alternative hypothesis, H1: $\pi.1 < \pi.2$; `alternative = "greater"` specifies the null hypothesis, H0: $\pi.1 <= \pi.2$ and the alternative hypothesis, H1: $\pi.1 > \pi.2$.

The main characteristic of the sequential triangular test is that there is no fixed sample size given in advance. That is, for the most recent sampling point, one has to decide whether sampling has to be continued or either the null- or the alternative hypothesis can be accepted given specified precision requirements (i.e. type-I-risk, type-II-risk and an effect size). The (cumulative) test statistic `Z.m` on a Cartesian coordinate system produces a "sequential path" on a continuation area as a triangle. As long as the statistic remains within that triangle, additional data have to be sampled. If the path touches or exceeds the borderlines of the triangle, sampling is completed. Depending on the particular borderline, the null-hypothesis is either accepted or rejected.

Value

Returns an object of class `seqtest`, to be used for later update steps. The object has following entries:

call	function call
type	type of the test (i.e., proportion)
spec	specification of function arguments
tri	specification of the triangular
dat	data
res	list with results

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

See Also

`update.seqtest`, `seqtest.mean`, `seqtest.cor`, `print.seqtest`, `plot.seqtest`, `descript`

Examples

```
#------------------------------------
# Two-sided one-sample test
# H0: pi = 0.5, H1: pi != 0.5
# alpha = 0.05, beta = 0.2, delta = 0.2

seq.obj <- seqtest.prop(c(1, 1, 0, 1), pi = 0.5, delta = 0.2,
                        alpha = 0.05, beta = 0.2, plot = TRUE)
```

```
seq.obj <- update(seq.obj, x = c(1, 1, 1, 1, 1, 0, 1, 1, 1))
seq.obj <- update(seq.obj, x = c(0, 1, 1, 1))
seq.obj <- update(seq.obj, x = c(1, 1))

#------------------------------------
# One-sided one-sample test
# H0: pi <= 0.5, H1: pi > 0.5
# alpha = 0.05, beta = 0.2, delta = 0.2

seq.obj <- seqtest.prop(c(1, 1, 0, 1), pi = 0.5,
                        alternative = "greater", delta = 0.2,
                        alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, x = c(1, 1, 1, 1, 1, 0, 1, 1, 1))
seq.obj <- update(seq.obj, x = c(0, 1, 1, 1))

#------------------------------------
# Two-sided two-sample test
# H0: pi.1 = pi.2 = 0.5, H1: pi.1 != pi.2
# alpha = 0.01, beta = 0.1, delta = 0.2

seq.obj <- seqtest.prop(1, 0, pi = 0.5, delta = 0.2,
                        alpha = 0.01, beta = 0.1, plot = TRUE)

seq.obj <- update(seq.obj, x = c(1, 1, 1, 0), y = c(0, 0, 1, 0))
seq.obj <- update(seq.obj, x = c(0, 1, 1, 1), y = c(0, 0, 0, 0))
seq.obj <- update(seq.obj, x = c(1, 0, 1, 1), y = c(0, 0, 0, 1))
seq.obj <- update(seq.obj, x = c(1, 1, 1, 1), y = c(0, 0, 0, 0))
seq.obj <- update(seq.obj, x = c(0, 1, 0, 1))
seq.obj <- update(seq.obj, y = c(0, 0, 0, 1))
seq.obj <- update(seq.obj, x = c(1, 1, 1, 1))

#------------------------------------
# One-sided two-sample test
# H0: pi.1 <=  pi.1 = 0.5, H1: pi.1 > pi.2
# alpha = 0.01, beta = 0.1, delta = 0.2

seq.obj <- seqtest.prop(1, 0, pi = 0.5, delta = 0.2,
                        alternative = "greater",
                        alpha = 0.01, beta = 0.1, plot = TRUE)

seq.obj <- update(seq.obj, x = c(1, 1, 1, 0), y = c(0, 0, 1, 0))
seq.obj <- update(seq.obj, x = c(0, 1, 1, 1), y = c(0, 0, 0, 0))
seq.obj <- update(seq.obj, x = c(1, 0, 1, 1), y = c(0, 0, 0, 1))
seq.obj <- update(seq.obj, x = c(1, 1, 1), y = c(0, 0))
```

---

sim.seqtest.cor   *Simulation of the sequential triangular test for Pearson's correlation coefficient*

Description

This function performs a statistical simulation for the sequential triangular test for Pearson's correlation coefficient.

Usage

```
sim.seqtest.cor(rho.sim, k, rho, alternative = c("two.sided", "less", "greater"),
                delta, alpha = 0.05, beta = 0.1, runs = 1000,
                m.x = 0, sd.x = 1, m.y = 0, sd.y = 1,
                digits = 3, output = TRUE, plot = FALSE)
```

Arguments

rho.sim	simulated population correlation coefficient, ρ.
k	an integer or a numerical vector indicating the number of observations in each sub-sample.
rho	a number indicating the correlation coefficient under the null hypothesis, $\rho.0$.
alternative	a character string specifying the alternative hypothesis,
delta	minimum difference to be detected, δ.
alpha	type-I-risk, α.
beta	an integer or a numerical vector indicating the type-II-risk, β.
runs	numer of simulation runs.
m.x	population mean of simulated vector x.
sd.x	population standard deviation of simulated vector x.
m.y	population mean of simulated vector y.
sd.y	population standard deviation of simulated vector y.
digits	integer indicating the number of decimal places to be displayed.
output	logical: if TRUE, output is shown.
plot	logical: if TRUE, plot is shown.

Details

In order to determine the optimal k, simulation is conducted under the H0 condition, i.e., rho.sim = rho. Simulation is carried out for a sequence of k values to seek for the optimal k where the empirical alpha is as close as possible to the nominal alpha. In order to determine optimal beta (with fixed k), simulation is conudcted under the H1 condition, i.e., rho.sim = rho + delta or rho.sim = rho - delta. Simulation is carried out for a sequencen of beta values to seek for the optimal beta where the empirical beta is as close as possible to the nominal beta.

In order to specify a one-sided test, argument alternative has to be used (i.e., two-sided tests are conducted by default). Specifying argument alternative = "less" conducts the simulation for the null hypothesis, H0: $\rho >= \rho.0$ with the alternative hypothesis, H1: $\rho < \rho.0$; specifying argument alternative = "greater" conducts the simluation for the null hypothesis, H0: $\rho <= \rho.0$ with the alternative hypothesis, H1: $\rho > \rho.0$.

Value

Returns an object of class `sim.seqtest.cor` with following entries:

call	function call
spec	specification of function arguments
simres	list with results (for each k or beta) for each run
res	data.frame with results, i.e., k, alpha.nom (nominal alpha), alpha.emp (estimated empirical alpha), beta.nom (nomin

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \leq \rho 0$. *Statistical Papers, 56*, 689-699.

See Also

seqtest.cor, plot.sim.seqtest.cor, print.sim.seqtest.cor

Examples

```
## Not run:

#------------------------------------------
# Determine optimal k and nominal type-II-risk
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.01, beta = 0.05, delta = 0.25

# Step 1: Determine the optimal size of subsamples (k)

sim.seqtest.cor(rho.sim = 0.3, k = seq(4, 16, by = 1), rho = 0.3,
                alternative = "greater",
                delta = 0.25, alpha = 0.05, beta = 0.05,
                runs = 10000, plot = TRUE)

# Step 2: Determine the optimal nominal type-II-risk based on
#         the optimal size of subsamples (k) from step 1

sim.seqtest.cor(rho.sim = 0.55, k = 16, rho = 0.3,
                alternative = "greater",
                delta = 0.25, alpha = 0.05, beta = seq(0.05, 0.15, by = 0.01),
                runs = 10000, plot = TRUE)

## End(Not run)
```

---

size.cor	*Sample size determination for testing Pearson's correlation coefficient*

Description

This function performs sample size computation for testing Pearson's correlation coefficient based on precision requirements (i.e., type-I-risk, type-II-risk and an effect size).

Usage

```
size.cor(rho = NULL, delta,
        alternative = c("two.sided", "less", "greater"),
        alpha = 0.05, beta = 0.1, output = TRUE)
```

Arguments

rho	a number indicating the correlation coefficient under the null hypothesis, $\rho.0$.
delta	minimum difference to be detected, δ.
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
alpha	type-I-risk, α.
beta	type-II-risk, β.
output	logical: if TRUE, output is shown.

Value

Returns an object of class size with following entries:

call	function call
type	type of the test (i.e., correlation coefficient)
spec	specification of function arguments
res	list with the result, i.e., optimal sample size

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

See Also

[seqtest.cor](), [size.mean](), [size.prop](), [print.size]()

Examples

```
#-----------------------------------
# H0: rho = 0.3, H1: rho != 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2

size.cor(rho = 0.3, delta = 0.2, alpha = 0.05, beta = 0.2)

#-----------------------------------
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2

size.cor(rho = 0.3, delta = 0.2, alternative = "greater", alpha = 0.05, beta = 0.2)
```

---

size.mean   *Sample size determination for testing the arithmetic mean*

Description

This function performs sample size computation for the one-sample and two-sample t-test based on precision requirements (i.e., type-I-risk, type-II-risk and an effect size).

Usage

```
size.mean(theta, sample = c("two.sample", "one.sample"),
          alternative = c("two.sided", "less", "greater"),
          alpha = 0.05, beta = 0.1, output = TRUE)
```

Arguments

theta	relative minimum difference to be detected, θ.
sample	a character string specifying one- or two-sample t-test, must be one of "two.sample" (default) or "one.sample".
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
alpha	type-I-risk, α.
beta	type-II-risk, β.
output	logical: if TRUE, output is shown.

Value

Returns an object of class size with following entries:

```
        call    function call
        type    type of the test (i.e., arithmetic mean)
        spec    specification of function arguments
        res     list with the result, i.e., optimal sample size
```

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

See Also

seqtest.mean, size.prop, size.cor, print.size

Examples

```
#-------------------------------------
# Two-sided one-sample test
# H0: mu = mu.0, H1: mu != mu.0
# alpha = 0.05, beta = 0.2, theta = 0.5

size.mean(theta = 0.5, sample = "one.sample",
          alternative = "two.sided", alpha = 0.05, beta = 0.2)

#-------------------------------------
# One-sided one-sample test
# H0: mu <= mu.0, H1: mu > mu.0
# alpha = 0.05, beta = 0.2, theta = 0.5

size.mean(theta = 0.5, sample = "one.sample",
          alternative = "greater", alpha = 0.05, beta = 0.2)

#-------------------------------------
# Two-sided two-sample test
# H0: mu.1 = mu.2, H1: mu.1 != mu.2
# alpha = 0.01, beta = 0.1, theta = 1

size.mean(theta = 1, sample = "two.sample",
          alternative = "two.sided", alpha = 0.01, beta = 0.1)

#-------------------------------------
# One-sided two-sample test
# H0: mu.1 <= mu.2, H1: mu.1 > mu.2
```

```
# alpha = 0.01, beta = 0.1, theta = 1

size.mean(theta = 1, sample = "two.sample",
          alternative = "greater", alpha = 0.01, beta = 0.1)
```

---

size.prop   *Sample size determination for testing the proportion*

Description

This function performs sample size computation for the one-sample and two-sample test for proportions based on precision requirements (i.e., type-I-risk, type-II-risk and an effect size).

Usage

```
size.prop(pi = NULL, delta, sample = c("two.sample", "one.sample"),
          alternative = c("two.sided", "less", "greater"),
          alpha = 0.05, beta = 0.1, correct = FALSE, output = TRUE)
```

Arguments

pi	a number indicating the true value of the probability under the null hypothesis (one-sample test), $\pi.0$ or a number indicating the true value of the probability in group 1 (two-sample test), $\pi.1$.
delta	minimum difference to be detected, δ.
sample	a character string specifying one- or two-sample proportion test, must be one of "two.sample" (default) or "one.sample".
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "less" or "greater".
alpha	type-I-risk, α.
beta	type-II-risk, β.
correct	a logical indicating whether continuity correction should be applied.
output	logical: if TRUE, output is shown.

Value

Returns an object of class size with following entries:

call	function call
type	type of the test (i.e., proportion)
spec	specification of function arguments
res	list with the result, i.e., optimal sample size

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

Fleiss, J. L., Levin, B., & Paik, M. C. (2003). *Statistical methods for rates and proportions* (3rd ed.). New York: John Wiley & Sons.

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

See Also

seqtest.prop, size.mean, size.cor, print.size

Examples

```
#-----------------------------------
# Two-sided one-sample test
# H0: pi = 0.5, H1: pi != 0.5
# alpha = 0.05, beta = 0.2, delta = 0.2

size.prop(pi = 0.5, delta = 0.2, sample = "one.sample",
          alternative = "two.sided", alpha = 0.05, beta = 0.2)

#-----------------------------------
# One-sided one-sample test
# H0: pi <= 0.5, H1: pi > 0.5
# alpha = 0.05, beta = 0.2, delta = 0.2

size.prop(pi = 0.5, delta = 0.2, sample = "one.sample",
          alternative = "less", alpha = 0.05, beta = 0.2)

#-----------------------------------
# Two-sided two-sample test
# H0: pi.1 = pi.2 = 0.5, H1: pi.1 != pi.2
# alpha = 0.01, beta = 0.1, delta = 0.2

size.prop(pi = 0.5, delta = 0.2, sample = "two.sample",
          alternative = "two.sided", alpha = 0.01, beta = 0.1)

#-----------------------------------
# One-sided two-sample test
# H0: pi.1 <=  pi.1 = 0.5, H1: pi.1 > pi.2
# alpha = 0.01, beta = 0.1, delta = 0.2

size.prop(pi = 0.5, delta = 0.2, sample = "two.sample",
          alternative = "greater", alpha = 0.01, beta = 0.1)
```

---

update.seqtest *Update seqtest*

Description

This function updates the `seqtest` object

Usage

```
## S3 method for class 'seqtest'
update(object, x = NULL, y = NULL, initial = FALSE,
       output = TRUE, plot = TRUE, ...)
```

Arguments

object	`cor.seqtest` object.
x	data for group 1.
y	data for group 2.
initial	logical, used internally for creating a `seqtest` object
output	logical: if `TRUE`, output is shown.
plot	logical: if `TRUE`, plot is shown.
...	further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

Rasch, D., Pilz, J., Verdooren, L. R., & Gebhardt, G. (2011). *Optimal experimental design with R*. Boca Raton: Chapman & Hall/CRC.

Rasch, D., Kubinger, K. D., & Yanagida, T. (2011). *Statistics in psychology - Using R and SPSS*. New York: John Wiley & Sons.

Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient's null-hypothesis: $0 < \rho \leq \rho 0$. *Statistical Papers, 56*, 689-699.

See Also

[seqtest.mean](), [seqtest.prop](), [seqtest.cor](),

Examples

```
#-------------------------------------
# Sequential triangular test for the arithmetic mean in one sample

seq.obj <- seqtest.mean(56, mu = 50, theta = 0.5,
                        alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, x = c(54, 52, 46, 49))

#-------------------------------------
# Sequential triangular test for the proportion in one sample

seq.obj <- seqtest.prop(c(1, 1, 0, 1), pi = 0.5, delta = 0.2,
                        alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, x = c(1, 1, 1, 1, 1, 0, 1, 1, 1))

#-------------------------------------
# Sequential triangular test for Pearson's correlation coefficient

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2,
                        alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, c(0.56, 0.76, 0.56, 0.52))
```

Index