Package: generalRSS (via r-universe)

May 11, 2026

Type Package

Title Statistical Tools for Balanced and Unbalanced Ranked Set Sampling

Version 0.2.0

Maintainer Soohyun Ahn <shahn@ajou.ac.kr>

Description Ranked Set Sampling (RSS) is a stratified sampling method known for its efficiency compared to Simple Random Sampling (SRS). When sample allocation is equal across strata, it is referred to as balanced RSS (BRSS) whereas unequal allocation is called unbalanced RSS (URSS), which is particularly effective for asymmetric or skewed distributions. This package offers practical statistical tools and sampling methods for both BRSS and URSS, emphasizing flexible sampling designs and inference for population means, medians, proportions, and Area Under the Curve (AUC). It incorporates parametric and nonparametric tests, including empirical likelihood ratio (LR) methods. The package provides ranked set sampling methods from a given population, including sampling with imperfect ranking using auxiliary variables. Furthermore, it provides tools for efficient sample allocation in URSS, ensuring greater efficiency than SRS and BRSS. For more details, refer e.g. to Chen et al. (2003) <doi:10.1007/978-0-387-21664-5>, Ahn et al. (2022) <doi:10.1007/978-3-031-14525-4_3>, and Ahn et al. (2024) <doi:10.1111/insr.12589>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

NeedsCompilation no

Imports methods, stats, emplik, rootSolve

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://shahn63.r-universe.dev

Date/Publication 2025-09-12 07:47:38 UTC

RemoteUrl https://github.com/shahn63/generalrss

RemoteRef HEAD

RemoteSha 8a148772e177e5675bdbafe35daaa21b2d8fec33

Contents

`rss.AUC.test` *RSS empirical likelihood ratio (ELR) test in two-sample comparison*

Description

The rss.AUC.test function conducts an empirical likelihood ratio test to compare the Area Under the Curve (AUC) between two groups using RSS. It supports both balanced and unbalanced RSS designs.

Usage

```
rss.AUC.test(data1, data2, alpha = 0.05, delta0 = 0.5)
```

Arguments

`data1`	A numeric data frame of ranked set samples with columns `rank` for ranks and `y` for data values.
`data2`	A numeric data frame of ranked set samples with columns `rank` for ranks and `y` for data values.
`alpha`	A numeric value specifying the confidence level for the interval.
`delta0`	A numeric value indicating the hypothesized value of the AUC.

Details

This function performs an empirical likelihood ratio test to compare the Area Under the Curve (AUC) between two groups using Ranked Set Sampling (RSS). The test is equivalent to the Mann-Whitney U test, which evaluates whether there is a significant difference between the distributions of two groups. The function supports both Balanced RSS (BRSS) and unbalanced RSS (URSS), as described by Moon et al. (2022). Given two data frames of RSS data (`data1` and `data2`) with `rank` and `y` columns, the function calculates the empirical likelihood ratio test statistic, confidence interval, and p-value based on the hypothesized AUC value `delta0`. Test for `delta0`=0.5 corresponds to the null hypothesis of the Mann-Whitney U test, which asserts that there is no difference between the two groups, implying that any randomly selected observation from one group is just as likely to be greater or smaller than an observation from the other group.

Value

`RSS_AUC`	The RSS AUC estimate.
`CI`	The confidence interval for the AUC.
`-2*Log.LR`	The empirical log likelihood ratio test statistics.
`p.value`	The p-value for the test.

References

C. Moon, X. Wang, and J. Lim. (2022) Empirical likelihood inference for area under the receiver operating characteristic curve using ranked set samples. Pharmaceutical Statistics, 21(6), 1219–1245.

See Also

`rss.simulation`: used for simulating Ranked Set Samples (RSS), which can serve as input.

`rss.sampling`: used for sampling Ranked Set Samples (RSS) from a population data set, providing input data.

Examples

```
## balanced RSS with a set size 3 and different sample sizes of 6 or 8 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data1<-rss.simulation(H=3,nsamp=c(6,6,6),dist="normal", rho=0.8,delta=0)
rss.data2<-rss.simulation(H=3,nsamp=c(8,8,8),dist="normal", rho=0.8,delta=0.2)

## RSS empirical likelihood ratio test
rss.AUC.test(data1=rss.data1, data2=rss.data2, alpha=0.05, delta0=0.5)

## Unbalanced RSS with a set size 3 and different sample sizes of 6, 10, and 8 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data1<-rss.simulation(H=3,nsamp=c(6,10,8),dist="normal", rho=0.8,delta=0)
rss.data2<-rss.simulation(H=3,nsamp=c(6,10,8),dist="normal", rho=0.8,delta=0.2)

## RSS empirical likelihood ratio test
rss.AUC.test(data1=rss.data1, data2=rss.data2, alpha=0.05, delta0=0.5)
```

`rss.design` *Calculate efficient sample allocations for ranked set sampling*

Description

The rss.design function calculates three efficient sample allocations for each stratum using given either initial RSS data or design. These allocations include Integer Neyman, adjusted Neyman, and local ratio consistent allocations for quantitative data and optimal Neyman for binary data to improve sampling efficiency.

Usage

```
rss.design(data = NULL, H = NULL, org.n = NULL, var.h = NULL, prop = FALSE)
```

Arguments

`data`	A numeric data frame of ranked set samples with columns 'rank' for ranks and 'y' for data values.
`H`	The set size
`org.n`	The initial sample allocation vector
`var.h`	A vector of strata variances
`prop`	logical indicating the sampling design for proportions

Details

This function offers a systematic approach for determining efficient sample allocations in unbalanced RSS. Given either an initial RSS data or design (but not both), the function optimizes sample allocation to ensure higher efficiency compared to the initial RSS design. For quantitative data, the function calculates the integer Neyman allocation proposed by Wright (2012), as well as, the adjusted Neyman allocation and local ratio consistent (LRC) allocation introduced by Ahn et al. (2022), which guarantees efficiency improvements compared to SRS and BRSS. For binary data, it computes the optimal Neyman allocation by Chen et al. (2006).

Value

`original.n`	The initial RSS sample allocation.
`Integer.Neyman`	The integer Neyman allocation.
`Adj.Neyman`	The adjusted Neyman allocation.
`LRC.allocation`	The local ratio consistent allocation.
`Neyman.proportion`	The optimal Neyman allocation for proportions.

References

S. Ahn, X. Wang, and J. Lim. (2022) Efficient sample allocation by local adjustment for unbalanced ranked set sampling. In Recent Advances on Sampling Methods and Educational Statistics, Springer.

H. Chen, E.A. Stasny, and D.A. Wolfe. (2006). Unbalanced ranked set sampling for estimating a population proportion. Biometrics, 62(1), 150-158.

T. Wright (2012) The equivalence of Neyman optimum allocation for sampling and equal proportions for apportioning the U.S. house of representatives. The American Statistician, 66(4), 217-224.

See Also

`rss.simulation`: used for simulating Ranked Set Samples (RSS), which can serve as input.

`rss.sampling`: used for sampling Ranked Set Samples (RSS) from a population data set, providing input data.

Examples

```
## Unbalanced RSS with a set size 3 and different sample sizes of 3, 10, and 5 for each stratum,
## using perfect ranking from a t distribution with a mean of 0.
rss.data=rss.simulation(H=3,nsamp=c(3,10,5), dist="t", rho=1,delta=0)

# Check the structure of the RSS data
colnames(rss.data) # Should include "y" and "rank"
head(rss.data$y)
head(rss.data$rank)

## RSS allocation calculation for a given pilot RSS data
rss.design(rss.data)


## Unbalanced RSS with a set size 3 and different sample sizes of 10, 15, and 15 for each stratum,
## using perfect ranking for proportions with a population proportion of 0.5.
rss.prop.data=rss.prop.simulation(H=3,nsamp=c(10,15,20),p=0.5)

## RSS allocation calculation for a given pilot RSS binary data
rss.design(rss.prop.data,prop=TRUE)
```

`rss.ELR.test` *RSS empirical likelihood ratio (ELR) test for one-sample population mean*

Description

The rss.ELR.test function conducts a one-sample empirical likelihood ratio test on ranked set sample data to assess the population mean.

Usage

```
rss.ELR.test(data, alpha = 0.05, mu0)
```

Arguments

data	A numeric data frame of ranked set samples with columns rank for ranks and y for data values.
alpha	A numeric value specifying the confidence level for the interval.
mu0	A numeric value indicating the hypothesized value of the mean.

Details

This function performs a one-sample empirical likelihood ratio (ELR) test on ranked set sample data using the method introduced by Ahn et al. (2024). Given a data frame of RSS data data with rank and y columns, the function calculates the empirical likelihood ratio test statistic, confidence interval, and p-value based on the hypothesized mean value mu0.

Value

RSS_mean	The RSS mean estimate.
CI	The confidence interval for the population mean.
-2*Log.LR	The empirical log likelihood ratio test statistic.
p.value	The p-value for the test.

References

S. Ahn, X. Wang, C. Moon, and J. Lim. (2024) New scheme of empirical likelihood method for ranked set sampling: Applications to two one sample problems. International Statistical Review.

See Also

rss.simulation: used for simulating Ranked Set Samples (RSS), which can serve as input.

rss.sampling: used for sampling Ranked Set Samples (RSS) from a population data set, providing input data.

Examples

```
## Unbalanced RSS with a set size 3 and different sample sizes of 6, 10, and 8 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data<-rss.simulation(H=3,nsamp=c(6,10,8),dist="normal", rho=0.8,delta=0)

## RSS empirical likelihood ratio test

rss.ELR.test(data=rss.data, alpha=0.05, mu0=0)
```

`rss.prop.sampling` *Generate ranked set samples for proportions*

Description

The rss.prop.sampling function generates ranked set samples for proportions by performing ranked set sampling directly on a given population data set using an auxiliary variable (X) and subject IDs. Sampling is conducted under the infinite population framework, i.e., sets are drawn with replacement across cycles.

Usage

```
rss.prop.sampling(ID, X, H, nsamp)
```

Arguments

`ID`	A numeric vector of subject IDs from the population. IDs must be unique.
`X`	A numeric vector of auxiliary variable used for ranking. Must have the same length as ID.
`H`	The RSS set size
`nsamp`	A numeric vector specifying the sample allocation for each stratum.

Details

This function performs balanced or unbalanced ranked set sampling for proportions from a given data set. The length of the sample allocation vector (nsamp) must match the set size (H). The subject ID and auxiliary variable (X) must have the same length. In line with the infinite population framework, each cycle is generated independently with replacement across cycles.

Value

A data frame with the following columns:

`ID`	The sampled subjects' IDs.
`rank`	The rank information assigned to each sample.

Examples

```
## Example 1: Balanced RSS with equal sample sizes.
data(iris)
id=1:nrow(iris)
X=ifelse(iris$Sepal.Length<5.8,0,1)
rss.prop.data=rss.prop.sampling(ID=id, X=X, H=3,nsamp=c(6,6,6))

## Example 2: Unbalanced RSS with different sample sizes.
rss.prop.data=rss.prop.sampling(ID=id, X=X, H=3, nsamp=c(6,10,8))
```

```
# Check the structure of the RSS data
colnames(rss.prop.data) # include "ID", "rank", and "Y"
head(rss.prop.data$ID)
head(rss.prop.data$rank)
```

`rss.prop.simulation` *Generate example ranked set samples for proportions*

Description

The rss.prop.simulation function generates ranked set samples for proportions by simulating data with options to adjust the population proportion (p).

Usage

```
rss.prop.simulation(H, nsamp, p)
```

Arguments

`H`	The RSS set size.
`nsamp`	A numeric vector specifying the sample allocation for each stratum.
`p`	The true population proportion.

Details

This function simulates balanced or unbalanced RSS sampling for proportions. The length of the sample allocation vector (nsamp) must match the set size (H). The p parameter adjusts the true population proportion. The simulations assumes perfect ranking.

Value

`rank`	The rank information assigned to each sample.
`y`	The generated ranked set samples (binary values) based on the population proportion.

Examples

```
## Balanced RSS with a set size 3 and equal sample sizes of 6 for each stratum,
## using perfect ranking with true proportion of 0.6.
rss.prop.data=rss.prop.simulation(H=3,nsamp=c(6,6,6),p=0.6)

## Unbalanced RSS with a set size 3 and different sample sizes of 6, 10, and 8 for each stratum,
## using perfect ranking with true proportion of 0.2.
rss.prop.data=rss.prop.simulation(H=3,nsamp=c(6,10,8),p=0.2)

# Check the structure of the RSS data
colnames(rss.prop.data) # Should include "y" and "rank"
```

```
head(rss.prop.data$y)
head(rss.prop.data$rank)
```

`rss.prop.test` *RSS proportion test*

Description

The rss.prop.test function performs the population proportion test on ranked set sample data, supporting both balanced and unbalanced RSS designs.

Usage

```
rss.prop.test(data, alpha = 0.05, alternative = "two.sided", p0)
```

Arguments

`data`	A numeric data frame of ranked set samples with columns `rank` for ranks and `y` for data values.
`alpha`	A numeric value specifying the confidence level for the interval.
`alternative`	A character string specifying the alternative hypothesis. Must be one of "two.sided" (default), "greater", or "less".
`p0`	A numeric value indicating the hypothesized proportion for the one-sample test.

Details

This function performs a proportion test on ranked set samples. It uses the method introduced by Chen et al. (2006), Zamanzade and Mahdizadeh (2020), and Ahn. et al. (2022) . Provide `data` as a data frame with columns `rank` and `y`. The function calculates the test statistic, confidence intervals, and p-value based on the RSS data.

Value

`RSS_prop`	The RSS proportion estimate.
`CI`	The confidence interval for the population proportion.
`pstat`	The test statistic for the proportion test.
`p.value`	The p-value for the test.

References

Chen, H., Stasny, E. A., & Wolfe, D. A. (2006). Unbalanced ranked set sampling for estimating a population proportion. Biometrics, 62(1), 150-158.

Zamanzade, E., & Mahdizadeh, M. (2020). Using ranked set sampling with extreme ranks in estimating the population proportion. Statistical methods in medical research, 29(1), 165-177.

Ahn, S., Wang, X., Wang, M., & Lim, J. (2022). On continuity correction for RSS-structured cluster randomized designs with binary outcomes. Metron, 80(3), 383-397.

See Also

rss.prop.simulation: used for simulating Ranked Set Samples (RSS) for proportions, which can serve as input.

rss.prop.sampling: used for sampling Ranked Set Samples (RSS) from a population data set for proportions, providing input data.

Examples

```
## Unbalanced RSS with a set size 3 and different sample sizes of 12, 9, 6 for each stratum,
## with a population proportion of 0.6.
rss.prop.data=rss.prop.simulation(H=3,nsamp=c(12,9,6),p=0.6)

## RSS proportion test
rss.prop.test(data=rss.prop.data, alpha=0.05, alternative="two.sided", p0=0.2)
```

---

rss.sampling *Generate ranked set samples*

Description

The rss.sampling function generates ranked set samples by performing ranked set sampling directly on a given population data set using an auxiliary variable (X) and subject IDs. The outcome variable (Y) is optional. If Y=NULL, the function returns only the sampled IDs and ranks. Sampling is conducted under the infinite population framework, i.e., sets are drawn with replacement across cycles.

Usage

```
rss.sampling(ID, Y = NULL, X, H, nsamp)
```

Arguments

ID	A numeric vector of subject IDs from the population. IDs must be unique.
Y	A numeric vector of interested outcome variable. If Y=NULL (default), only IDs and ranks are returned.
X	A numeric vector of auxiliary variable used for ranking. Must have the same length as ID.
H	The RSS set size
nsamp	A numeric vector specifying the sample allocation for each stratum.

Details

This function performs balanced or unbalanced ranked set sampling from a given data set. The length of the sample allocation vector (nsamp) must match the set size (H). The subject ID, outcome variable (Y), and auxiliary variable (X) must have the same length. If Y is not provided (Y=NULL), the function returns only the sampled IDs and their ranks without generating Y values. In line with the infinite population framework, each cycle is generated independently with replacement across cycles.

Value

A data frame with the following columns:

`ID`	The sampled subjects' IDs.
`rank`	The rank information assigned to each sample.
`y`	The generated ranked set samples of the outcome variable Y. If Y=NULL, this column is not included.

Examples

```
## Example 1: Balanced RSS with equal sample sizes.
data(iris)
id=1:nrow(iris)
rss.data=rss.sampling(ID=id, Y=iris$Sepal.Length, X=iris$Petal.Length, H=3,nsamp=c(6,6,6))

## Example 2: Unbalanced RSS with different sample sizes.
rss.data=rss.sampling(ID=id, Y=iris$Sepal.Length, X=iris$Petal.Length, H=3, nsamp=c(6,10,8))

# Check the structure of the RSS data
colnames(rss.data) # include "ID", "rank", and "Y"
head(rss.data$ID)
head(rss.data$rank)

## Example 3: If Y is not available, retrieve sampled IDs and ranks only.
rss.data=rss.sampling(ID=id, X=iris$Petal.Length, H=3,nsamp=c(6,10,8))

# Check the structure of the RSS data
colnames(rss.data) # include "ID" and "rank"
head(rss.data$ID)
head(rss.data$rank)
head(rss.data$y)
```

---

rss.sign.test *RSS Sign test*

Description

The rss.sign.test function performs Sign test on ranked set sample data, supporting both balanced and unbalanced RSS designs.

Usage

```
rss.sign.test(data, alpha = 0.05, alternative = "two.sided", median0)
```

Arguments

data	A numeric data frame of ranked set samples with columns rank for ranks and y for data values.
alpha	A numeric value specifying the confidence level for the interval.
alternative	A character string specifying the alternative hypothesis. Must be one of "two.sided" (default), "greater", or "less".
median0	A numeric value indicating the hypothesized median for the one-sample test.

Details

This function performs a one-sample sign test on ranked set samples. For balanced RSS (BRSS), it uses the method introduced by Hettmansperger (1995), while for unbalanced RSS (URSS), it follows the approach described by Barabesi (2001). Provide data as a data frame with columns rank and y. The function calculates the sign statistic, confidence intervals, and p-value based on the RSS data.

Value

RSS_median	The RSS median estimate.
sign	The sign-statistic for the test.
CI	The confidence interval for the population median.
z	The z-statistic for the test.
p.value	The p-value for the test.

References

T. P. Hettmansperger. (1995) The ranked-set sample sign test. Journal of Non-parametric Statistics, 4:263–270.

L. Barabesi. (2001) The unbalanced ranked-set sample sign test. Journal of Non-parametric Statistics, 13(2):279–289.

Chen, Z., Bai Z., Sinha B. K. (2003). Ranked Set Sampling: Theory and Application. New York: Springer.

S. Ahn, X. Wang, C. Moon, and J. Lim. (2024) New scheme of empirical likelihood method for ranked set sampling: Applications to two one sample problems. Internation Statistical Review.

See Also

rss.simulation: used for simulating Ranked Set Samples (RSS), which can serve as input.

rss.sampling: used for sampling Ranked Set Samples (RSS) from a population data set, providing input data.

Examples

```
## Unbalanced RSS with a set size 3 and different sample sizes of 12, 9, 6 for each stratum,
## using imperfect ranking from a lognormal distribution with a mean of 0.
rss.data=rss.simulation(H=3,nsamp=c(12,9,6),dist="lognormal", rho=0.8,delta=0)

## RSS sign-test
rss.sign.test(data=rss.data, alpha=0.05, alternative="two.sided", median0=0)
```

`rss.simulation` *Generate example ranked set samples*

Description

The rss.simulation function generates ranked set samples by simulating data from a specified population distribution: a standard normal distribution, a t-distribution with 5 degrees of freedom, or a log-normal sample with log-scale mean 0 and log-scale standard deviation $\sqrt{0.481}$, centered by subtracting 1.27 so the resulting data have approximately mean 0 and variance 1 ('normal', 't', or 'lognormal', respectively) with options to adjust the mean (`delta`) and control ranking quality (`rho`).

Usage

```
rss.simulation(H, nsamp, dist, rho, delta)
```

Arguments

`H`	The RSS set size.
`nsamp`	A numeric vector specifying the sample allocation for each stratum.
`dist`	A character string specifying the distribution to generate ranked set samples. Must be one of 'normal', 't', 'lognormal'.
`rho`	The ranking quality or accuracy indicating the correlation between outcome and auxiliary variable. Must be between 0 and 1 for imperfect ranking, or exactly 1 for perfect ranking.
`delta`	The true population mean.

Details

This function simulates balanced or unbalanced RSS sampling. The length of the sample allocation vector (`nsamp`) must match the set size (`H`). The dist parameter allows users to select one of three population distributions: a standard normal distribution, a t-distribution with 5 degrees of freedom, or a log-normal sample with log-scale mean 0 and log-scale standard deviation $\sqrt{0.481}$, centered by subtracting 1.27 so the resulting data have approximately mean 0 and variance 1. Additionally, the `delta` parameter adjusts the population mean. The `rho` parameter controls ranking accuracy by representing the correlation between the outcome and an auxiliary variable. A value of `rho` = 1 indicates perfect ranking, while values between 0 and 1 represent varying levels of imperfect

ranking using a linear ranking model which is defined as $X_i = Y_i + \epsilon_i$ for $i = 1, 2, \cdots, n$ where ϵ_i are independent random variables from $N(0, \sigma^2)$. The variance σ^2 is set to obtain a specific correlation between the outcome (Y) and auxiliary (X) variables.

Value

rank	The rank information assigned to each sample.
y	The generated ranked set samples based on the specified distribution.

Examples

```
## Balanced RSS with a set size 3 and equal sample sizes of 6 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data=rss.simulation(H=3,nsamp=c(6,6,6),dist="normal", rho=0.8,delta=0)

## Unbalanced RSS with a set size 3 and different sample sizes of 6, 10, and 8 for each stratum,
## using perfect ranking from a t distribution with a mean of 0.
rss.data=rss.simulation(H=3,nsamp=c(6,10,8),dist="t", rho=1,delta=0)

# Check the structure of the RSS data
colnames(rss.data) # Should include "y" and "rank"
head(rss.data$y)
head(rss.data$rank)
```

---

rss.t.test *RSS t-test for one-sample and two-sample problems*

Description

The rss.t.test function performs one- and two-sample t-tests on ranked set sample data using t approximations, with methods described by Ahn et al. (2014).

Usage

```
rss.t.test(
  data1,
  data2 = NULL,
  alpha = 0.05,
  alternative = "two.sided",
  mu0 = 0,
  method
)
```

Arguments

data1	A numeric data frame of ranked set samples with columns rank for ranks and y for data values.
data2	An optional numeric data frame of ranked set samples with columns rank for ranks and y for data values.
alpha	A numeric value specifying the confidence level for the interval.
alternative	A character string specifying the alternative hypothesis. Must be one of "two.sided" (default), "greater", or "less".
mu0	A numeric value indicating the hypothesized value of the mean (for a one-sample problem) or the mean difference (for a two-sample problem).
method	A character string specifying the method used to approximate the t-distribution. Must be either "sample" or "naive".

Details

This function performs a t-test on ranked set sample data for both one-sample and two-sample mean problems, using t approximations. For a one-sample test, provide data1 as a data frame with rank and y columns. For a two-sample test, provide both data1 and data2 with equal set sizes. The method parameter allows for two options to approximate the t-distribution: "sample" and "naive" as introduced by Ahn et al. (2014). The function compute the t-statistic, confidence interval, degrees of freedom, and p-value based on the provided RSS data and specified parameters.

Value

RSS_mean	The RSS mean estimate (for a one-sample problem) or a vector of RSS mean estimates for each group (for a two-sample problem).
CI	The confidence interval for the population mean (for a one-sample problem) or for the mean difference (for a two-sample problem).
t	The t-statistic for the test.
df	The degrees of freedom for the test.
p.value	The p-value for the test.

References

S. Ahn, J. Lim, and X. Wang. (2014) The student's t approximation to distributions of pivotal statistics from ranked set samples. Journal of the Korean Statistical Society, 43, 643–652.

See Also

rss.simulation: used for simulating Ranked Set Samples (RSS), which can serve as input.

rss.sampling: used for sampling Ranked Set Samples (RSS) from a population data set, providing input data.

Examples

```
## Balanced RSS with a set size 3 and equal sample sizes of 6 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data1=rss.simulation(H=3,nsamp=c(6,6,6),dist="normal", rho=0.8,delta=0)

## one-sample t-test using 'naive' method
rss.t.test(data1=rss.data1, data2=NULL, alpha=0.05,
alternative="two.sided", mu0=0, method="naive")

## one-sample t-test using 'sample' method
rss.t.test(data1=rss.data1, data2=NULL, alpha=0.05,
alternative="two.sided", mu0=0, method="sample")

## Unbalanced RSS with a set size 3 and different sample sizes of 6, 10, and 8 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data2<-rss.simulation(H=3,nsamp=c(6,8,10),dist="normal", rho=0.8,delta=0)

## two-sample t-test using 'naive' method
rss.t.test(data1=rss.data1, data2=rss.data2, alpha=0.05,
alternative="two.sided", mu0=0, method="naive")

## two-sample t-test using 'sample' method
rss.t.test(data1=rss.data1, data2=rss.data2, alpha=0.05,
alternative="two.sided", mu0=0, method="sample")
```

`rss.z.test` *RSS z-test for one-sample and two-sample problems*

Description

The rss.z.test function performs one- and two-sample z-tests on ranked set sample data using normal approximation, with options for specifying the confidence level, alternative hypothesis, and hypothesized mean or mean difference.

Usage

```
rss.z.test(
  data1,
  data2 = NULL,
  alpha = 0.05,
  alternative = "two.sided",
  mu0 = 0
)
```

Arguments

`data1`	A numeric data frame of ranked set samples with columns `rank` for ranks and `y` for data values.

data2	An optional numeric data frame of ranked set samples with columns rank for ranks and y for data values (for two-sample problem).
alpha	A numeric value specifying the confidence level for the interval.
alternative	A character string specifying the alternative hypothesis. Must be one of "two.sided" (default), "greater", or "less".
mu0	A numeric value indicating the hypothesized value of the mean (for a one-sample problem) or the difference in means (for a two-sample problem).

Details

This function performs a z-test on ranked set sample data for both one-sample and two-sample mean comparison problems, using normal approximation. For a one-sample test, only data1 is needed, provided as a data frame with columns rank and y. For a two-sample test, both data1 and data2 must be supplied, each as data frames with rank and y columns. The function computes the test statistic, confidence interval, and p-value based on the provided RSS data and specified parameters.

Value

RSS_mean	The RSS mean estimate for a one-sample problem or a vector of RSS mean estimates for each group in a two-sample problem.
CI	The confidence interval for the population mean for a one-sample problem or for the mean difference in a two-sample problem.
z	The z-statistic for the test.
p.value	The p-value for the test.

References

Chen, Z., Bai Z., Sinha B. K. (2003). Ranked Set Sampling: Theory and Application. New York: Springer.

S. Ahn, J. Lim, and X. Wang. (2014) The student's t approximation to distributions of pivotal statistics from ranked set samples. Journal of the Korean Statistical Society, 43, 643–652.

S. Ahn, X. Wang, C. Moon, and J. Lim. (2024) New scheme of empirical likelihood method for ranked set sampling: Applications to two one sample problems. International Statistical Review.

See Also

rss.simulation: used for simulating Ranked Set Samples (RSS), which can serve as input.

rss.sampling: used for sampling Ranked Set Samples (RSS) from a population data set, providing input data.

Examples

```
## Balanced RSS with a set size 3 and equal sample sizes of 6 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data1=rss.simulation(H=3,nsamp=c(6,6,6),dist="normal", rho=0.8,delta=0)

## one-sample z-test
```

```
rss.z.test(data1=rss.data1, data2=NULL, alpha=0.05,
alternative="two.sided", mu0=0)

## Unbalanced RSS with a set size 3 and different sample sizes of 6, 10, and 8 for each stratum,
## using imperfect ranking from a normal distribution with a mean of 0.
rss.data2<-rss.simulation(H=3,nsamp=c(6,8,10),dist="normal", rho=0.8,delta=0)

## two-sample z-test
rss.z.test(data1=rss.data1, data2=rss.data2, alpha=0.05,
alternative="two.sided", mu0=0)
```

Index