

# Package: tnl.Test (via r-universe)

June 11, 2026

**Type** Package

**Title** Non-Parametric Tests for the Two-Sample Problem

**Version** 0.1.0

**Description** Performing the hypothesis tests for the two sample problem based on order statistics and power comparisons. Calculate the test statistic, density, distribution function, quantile function, random number generation and others.

**License** GPL-3

**URL** <https://github.com/ihababusaif/tnl.Test>

**BugReports** <https://github.com/ihababusaif/tnl.Test/issues>

**Imports** partitions, plyr

**Suggests** covr, knitr, rmarkdown, roxygen2, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Ihab Abusaif [cre, aut]

(<https://www.researchgate.net/profile/Ihab-Abusaif>), Sümeýra Sert [aut]

(<https://www.researchgate.net/profile/Suemeyra-Sert>), Coşkun Kuş [aut] (<https://www.researchgate.net/profile/Coskun-Kus>), Kadir Karakaya [aut]

(<https://www.researchgate.net/profile/Kadir-Karakaya-2>), Hon Keung Tony Ng [aut], Haikady N. Nagaraja [aut]

**Maintainer** Ihab Abusaif <censtat@gmail.com>

**Config/pak/sysreqs** libgmp3-dev

**Repository** <https://cranhaven.r-universe.dev>

**Date/Publication** 2026-06-11 03:02:00 UTC

**RemoteUrl** <https://github.com/cranhaven/cranhaven.r-universe.dev>

**RemoteRef** package/tnl.Test

**RemoteSha** 0a597972b669fe14bd0c5982b7c56627a8eb3b82

**RemoteSubdir** tnl.Test

## Contents

tnl.test . . . . .	2
<b>Index</b>	<b>6</b>

---

tnl.test	<i>Non-parametric tests for the two-sample problem based on order statistics and power comparisons</i>
----------	--

---

### Description

`tnl.test` performs a nonparametric test for two sample test on vectors of data.

`ptnl` gives the distribution function of  $T_n^{(\ell)}$  against the specified quantiles.

`dtnl` gives the density of  $T_n^{(\ell)}$  against the specified quantiles.

`qtnl` gives the quantile function of  $T_n^{(\ell)}$  against the specified probabilities.

`rtnl` generates random values from  $T_n^{(\ell)}$ .

`tnl_mean()` gives an expression for  $E(T_n^{(\ell)})$  under  $H_0 : F = G$ .

`ptnl.lehmann` gives the distribution function of  $T_n^{(\ell)}$  under Lehmann alternatives.

`dtnl.lehmann` gives the density of  $T_n^{(\ell)}$  under Lehmann alternatives.

`qtnl.lehmann` gives the quantile function of  $T_n^{(\ell)}$  against the specified probabilities under Lehmann alternatives.

`rtnl.lehmann` generates random values from  $T_n^{(\ell)}$  under Lehmann alternatives.

### Usage

```
tnl.test(x, y, l, exact = "NULL")
```

```
ptnl(q, n, m, l, exact = "NULL", trial = 1e+05)
```

```
dtnl(k, n, m, l, exact = "NULL", trial = 1e+05)
```

```
qtnl(p, n, m, l, exact = "NULL", trial = 1e+05)
```

```
rtnl(N, n, m, l)
```

```
tnl_mean(n., m., l)
```

```
ptnl.lehmann(q, n., m., l, gamma)
```

```
dtnl.lehmann(k, n., m., l, gamma)
```

```
qtnl.lehmann(p, n., m., l, gamma)
```

```
rtnl.lehmann(N, n., m., l, gamma)
```

### Arguments

x	the first (non-empty) numeric vector of data values.
y	the second (non-empty) numeric vector of data values.
l	class parameter of $T_n^{(\ell)}$ .
exact	the method that will be used. "NULL" or a logical indicating whether an exact should be computed. See 'Details' for the meaning of NULL.
n, m	samples size.
trial	number of trials for simulation.
k, q	vector of quantiles.
p	vector of probabilities.
N	number of observations. If length(N) > 1, the length is taken to be the number required.
n., m.	samples size.
gamma	parameter of Lehmann alternative.

### Details

A non-parametric two-sample test is performed for testing null hypothesis  $H_0 : F = G$  against the alternative hypothesis  $H_1 : F \neq G$ . The assumptions of the  $T_n^{(\ell)}$  test are that both samples should come from a continuous distribution and the samples should have the same sample size.

Missing values are silently omitted from  $x$  and  $y$ .

Exact and simulated p-values are available for the  $T_n^{(\ell)}$  test. If exact="NULL" (the default) the p-value is computed based on exact distribution when the sample size is less than 11. Otherwise, p-value is computed based on a Monte Carlo simulation. If exact="TRUE", an exact p-value is computed. If exact="FALSE", a Monte Carlo simulation is performed to compute the p-value. It is recommended to calculate the p-value by a Monte Carlo simulation (use exact="FALSE"), as it takes too long to calculate the exact p-value when the sample size is greater than 10.

The probability mass function (pmf), cumulative density function (cdf) and quantile function of  $T_n^{(\ell)}$  are also available in this package, and the above-mentioned conditions about exact="NULL", exact="TRUE" and exact="FALSE" is also valid for these functions.

Exact distribution of  $T_n^{(\ell)}$  test is also computed under Lehman alternative.

Random number generator of  $T_n^{(\ell)}$  test statistic are provided under null hypothesis in the library.

**Value**

`tnl.test` returns a list with the following components

`statistic`: the value of the test statistic.

`p.value`: the p-value of the test.

`ptnl` returns a list with the following components

`method`: The method that was used (exact or simulation). See 'Details'.

`cdf`: distribution function of  $T_n^{(\ell)}$  against the specified quantiles.

`dtnl` returns a list with the following components

`method`: The method that was used (exact or simulation). See 'Details'.

`pmf`: density of  $T_n^{(\ell)}$  against the specified quantiles.

`qtnl` returns a list with the following components

`method`: The method that was used (exact or simulation). See 'Details'.

`quantile`: quantile function against the specified probabilities.

`rtnl` return  $N$  of the generated random values.

`tnl_mean()` return the mean of  $T_n^{(\ell)}$ .

`ptnl.lehmann` return vector of the distribution under Lehmann alternatives against the specified gamma.

`dtnl.lehmann` return vector of the density under Lehmann alternatives against the specified gamma.

`qtnl.lehmann` returns a quantile function against the specified probabilities under Lehmann alternatives.

`rtnl.lehmann` return  $N$  of the generated random values under Lehmann alternatives.

**References**

Karakaya, K., Sert, S., Abusaif, I., Kuş, C., Ng, H. K. T., & Nagaraja, H. N. (2023). A Class of Non-parametric Tests for the Two-Sample Problem based on Order Statistics and Power Comparisons. Submitted paper.

Aliev, F., Özbek, L., Kaya, M. F., Kuş, C., Ng, H. K. T., & Nagaraja, H. N. (2022). A nonparametric test for the two-sample problem based on order statistics. *Communications in Statistics-Theory and Methods*, 1-25.

**Examples**

```
require(stats)
x <- rnorm(7, 2, 0.5)
y <- rnorm(5, 0, 1)
tnl.test(x, y, l = 2)
ptnl(q = c(2, 5), n = 6, m = 5, l = 2, trial = 100000)
dtnl(k = c(1, 3, 6), n = 7, m = 5, l = 2)
qtnl(p = c(.3, .9), n = 4, m = 5, l = 1)
```

```
rtnl(N = 20, n = 7, m = 10, l = 1)
require(base)
tnl_mean(n. = 11, m. = 8, l = 1)
ptnl.lehmann(q = 3, n. = 5, m. = 7, l = 2, gamma = 1.2)
dtnl.lehmann(k = 3, n. = 6, m. = 5, l = 2, gamma = 0.8)
qtnl.lehmann(p = c(.1, .5, .9), n. = 7, m. = 5, l = 1, gamma = 0.5)
rtnl.lehmann(N = 15, n = 7,m=7, l = 2, gamma = 0.5)
```

# Index

`dtnl`, 2, 4

`dtnl (tnl.test)`, 2

`dtnl.lehmann`, 2, 4

`ptnl`, 2, 4

`ptnl (tnl.test)`, 2

`ptnl.lehmann`, 2, 4

`qtnl`, 2, 4

`qtnl (tnl.test)`, 2

`qtnl.lehmann`, 2, 4

`rtnl`, 2, 4

`rtnl (tnl.test)`, 2

`rtnl.lehmann`, 2, 4

`tnl.test`, 2, 2, 4

`tnl_mean (tnl.test)`, 2

`tnl_mean()`, 2, 4