Package 'wmwAUC'

July 31, 2026

Title Test of No Group Discrimination Using the WMW Statistic

Version 1.0.0

Date 2026-07-21

Description
Implements a wmwAUC test of H0: AUC = 1/2 for continuous, discrete, or mixed random variables, based on the Wilcoxon-Mann-Whitney (WMW) statistic. The classic WMW test is calibrated under H0: {(F, G): F = G} which does not match the set {(F, G): AUC = 1/2}, implied by the test statistic, and consequently leads to erroneous inferences. wmwAUC is calibrated under the correct null and implements two finite-sample corrected p-value methods: an Exact Unbiased (EU) method and a Bias-Corrected (BC) method, both valid for any tie pattern. Methods are described in M. Grendar (2025) ``Wilcoxon-Mann-Whitney Test of No Group Discrimination" <doi:10.48550/arXiv.2511.20308>.

License MIT + file LICENSE

URL https://github.com/grendar/wmwAUC

BugReports https://github.com/grendar/wmwAUC/issues

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.0.0)

Suggests testthat (>= 3.0.0), ggsci, gemR, knitr, rmarkdown, ggbeeswarm, ggplot2, rlang, rankFD, patchwork

Config/testthat/edition 3

NeedsCompilation no

Author Marian Grendar [aut, cre] (ORCID: <https://orcid.org/0000-0002-6712-3457>)

Maintainer Marian Grendar <marian.grendar@gmail.com>

Repository CRAN

Date/Publication 2026-07-31 10:40:02 UTC

Contents

`plot.wmwAUC_test` *Plot Method for wmwAUC_test Objects*

Description

Creates a combined plot ("diplot"): a boxplot with beeswarm overlay of the raw group values, alongside the empirical ROC curve (eROC) with confidence band (when available) and AUC confidence interval. Optionally adds a third panel with overlaid group density curves.

Usage

```
## S3 method for class 'wmwAUC_test'
plot(x, show_density = FALSE, combine_plots = TRUE, ...)
```

Arguments

`x`	Object of class 'wmwAUC_test' returned by `wmwAUC_test`.
`show_density`	Logical; if TRUE, adds a third panel with overlaid kernel density estimates for the two groups. Default FALSE.
`combine_plots`	Logical; if TRUE (default), returns a single combined plot via **patchwork**; if FALSE, returns a named list of the individual **ggplot2** objects instead.
`...`	Additional arguments (not currently used).

Value

If `combine_plots = TRUE`, a combined **patchwork** object. If `combine_plots = FALSE`, a named list with elements `box_plot`, `roc_plot`, and (if `show_density = TRUE`) `density_plot`.

print.wmwAUC_test *Print Method for wmwAUC_test Objects*

Description

Prints summary of wmwAUC test results.

Usage

```
## S3 method for class 'wmwAUC_test'
print(x, digits = 3, ...)
```

Arguments

x	Object of class 'wmwAUC_test' returned by `wmwAUC_test()`
digits	Integer, number of digits to display for numeric results (default: 3)
...	Additional arguments (not currently used)

Value

Invisibly returns the input object x (of class "wmwAUC_test"). Called primarily for side effects to print a formatted summary of the wmwAUC test results to the console.

simulation1 *Synthetic data*

Description

Synthetic data

Usage

```
data(simulation1)
```

Format

A list containing simulation results (N=10000, n=1000):

eauc Empirical AUC values

pval_wt Traditional wilcox.test p-values

pval_wmwAUC wmwAUC p-values under H0: AUC = 0.5

simulation2 *Synthetic data*

Description

Synthetic data

Usage

```
data(simulation2)
```

Format

A list containing simulation results (N=10000, n=1000):

eauc Empirical AUC values

pval_wt Traditional wilcox.test p-values

pval_wmwAUC wmwAUC p-values under H0: AUC = 0.5

wmwAUC-deprecated *Deprecated Function Names (wmwAUC 1.0.0)*

Description

These functions have been renamed in wmwAUC 1.0.0, alongside argument and behavior changes in some cases. Calling them raises an informative error pointing to the current function and to NEWS.md, rather than attempting to forward the call – for functions with argument-name or argument-value changes (see Details), a silent passthrough could otherwise produce a different result than the old function used to, without any indication that anything had changed.

Usage

```
wmw_test(...)

wmw_pvalue(...)

wmw_pvalue_ties(...)

pseudomedian_ci(...)
```

Arguments

...	Not used; calling any of these functions always raises an error.

Details

- wmw_test() -> wmwAUC_test. The special_case argument was renamed pseudomedian, and the ci_method = 'hanley' option was removed (not renamed), and replaced by the DeLong et al. (1988) method.
- wmw_pvalue() -> wmwAUC_pvalue_BC.
- wmw_pvalue_ties() -> wmwAUC_pvalue_EU.
- pseudomedian_ci() -> wmwAUC_pseudomedian_ci.

---

wmwAUC_pseudomedian_ci

Confidence Interval for the AUC-Equalizing Shift (Hodges-Lehmann Pseudomedian) via Test Inversion

---

Description

Computes a confidence interval for the pseudomedian by inverting the test $\mathrm{H}_0 : \mathrm{AUC}(X, Y + \delta) = 0.5$.

Usage

```
wmwAUC_pseudomedian_ci(
  x,
  y,
  conf.level = 0.95,
  pvalue_method = c("EU", "BC"),
  n_grid = 1000
)
```

Arguments

x	numeric vector, first sample
y	numeric vector, second sample
conf.level	confidence level (default 0.95)
pvalue_method	character, either 'EU' or 'BC'
n_grid	number of grid points for search (default 1000)

Details

The pseudomedian δ is defined generally, for any pair of distributions F, G, as the shift solving $P(X < Y + \delta) = 0.5$ — i.e. the value of δ that equalizes the AUC between X and the shifted $Y + \delta$. This definition does not require a location-shift model relating F and G. Under location-shift ($F(t) = G(t - \Delta)$), this δ coincides with the classical Hodges-Lehmann pseudomedian and equals the location difference Δ; this is a special case, not the definition. Outside location-shift, δ remains well-defined as the AUC-equalizing shift, but should not be read as "the median difference" between the two groups.

The confidence interval is obtained by grid search over candidate δ values, retaining those for which the AUC=0.5 test applied to $(X, Y + \delta)$ is not rejected at level `1 - conf.level`. The search range is set heuristically from ± 3 times the robust MAD scale of the pairwise differences around the point estimate; if the true interval extends beyond this range, or if the acceptance region is not contiguous (which can occur in small, imbalanced samples), a warning is issued. See Warnings.

EU and BC pseudomedian CIs are computed on independent calls, and may occasionally be reported as numerically identical at coarse n_grid even though the underlying p-value curves genuinely differ — both methods' crossing points can land in the same discrete grid cell.

At small `min(length(x), length(y))`, BC's near-zero power (see `wmwAUC_pvalue_BC`) can make its pseudomedian interval very wide and largely uninformative rather than merely conservative, since the grid search rarely rejects anywhere in the searched range. EU is recommended in that regime; see the warning issued below.

When `pvalue_method = 'EU'` and `length(x) + length(y) < 20`, each grid point's p-value may come from Monte Carlo permutation (see `wmwAUC_pvalue_EU`), making this function's result stochastic at small sample sizes. For reproducible results, call `set.seed()` immediately beforehand.

Value

list with conf.int, estimate and conf.level

Warnings

Two diagnostic warnings are issued when the grid search may be unreliable:

- **Non-contiguous acceptance region**: the interval is reported as `[min(accepted), max(accepted)]`, which assumes the set of non-rejected δ values is a single contiguous block. If it is not, the reported interval may be wider than the true acceptance region and should be inspected directly.
- **Search range too narrow**: if the p-value has not dropped below α at either edge of the search grid, the true endpoint may lie outside the searched range; consider widening the range or increasing `n_grid`.

`wmwAUC_pvalue_BC` *P-value for the Bias-Corrected (BC) Test of AUC*

Description

Tests $H_0 : \mathrm{AUC} = A_0$ vs the specified alternative, using a bias-corrected finite-sample variance estimator with the mid-rank kernel.

Usage

```
wmwAUC_pvalue_BC(
  x,
  y,
  alternative = "two.sided",
  A0 = 0.5,
  min_n_warn_threshold = 10
)
```

Arguments

x	Numeric vector of cases (group 1) values.
y	Numeric vector of reference/control (group 2) values.
alternative	Character: "two.sided", "greater", or "less".
A0	Numeric null value of AUC $= P(X < Y)$. Defaults to 0.5.
min_n_warn_threshold	Integer; if min(length(x), length(y)) is below this threshold, a warning is issued that power may be very low at this sample size. Default 10.

Details

BC estimates $\mathrm{Var}(\hat{A})$ by correcting each placement-variance component for its $O(1/n)$ upward bias, using a plug-in estimate of the bias subtracted from the naive placement variance; each corrected component is floored independently at a small $\epsilon > 0$ if it would otherwise go negative. The mid-rank kernel $h(x, y) = 1\{x < y\} + \frac{1}{2}1\{x = y\}$ is used throughout, for both the point estimate and the variance components.

Uses one-tier approach with $\hat{\sigma}^2_{\mathrm{adj}}$.

BC is a conservative test: observed size stays below nominal across a wide range of sample sizes, heteroskedasticity, and tie proportions, at a real cost in power for small or imbalanced samples. See min_n_warn_threshold and its warning text; the EU method (wmwAUC_pvalue_EU) is recommended when min(n1, n2) is small.

x is taken to represent cases and y the reference/control group, matching the convention of wilcox.test(). Internally, the test statistic and variance components are computed in the $P(X < Y)$ framework.

Value

Numeric p-value.

---

wmwAUC_pvalue_EU *P-value for the Exact Unbiased (EU) Test of AUC*

Description

Tests H_0: AUC $= A_0$ vs the specified alternative, using an exact finite-sample unbiased variance estimator with the mid-rank kernel.

Usage

```
wmwAUC_pvalue_EU(
  x,
  y,
  alternative = "two.sided",
  A0 = 0.5,
  max_exact = 10000,
  n_perm = 2000
)
```

Arguments

x	Numeric vector of cases (group 1) values.
y	Numeric vector of reference/control (group 2) values.
alternative	Character: "two.sided", "greater", or "less".
A0	Numeric null value of AUC $= P(X < Y)$. Defaults to 0.5. Only supported when length(x) + length(y) >= 20 (see Details); an error is raised otherwise.
max_exact	Integer; the permutation branch (used when length(x) + length(y) < 20) enumerates all permutations exactly when their count is at most max_exact, and falls back to Monte Carlo sampling above that. Default 10000.
n_perm	Integer; number of Monte Carlo permutation replicates used when exact enumeration is not feasible. Default 2000.

Details

Uses two-tier approach: studentized permutation for length(x) + length(y) < 20 and the exact finite-sample unbiased estimator for length(x) + length(y) >= 20.

For length(x) + length(y) < 20, a studentized permutation test is used: the same t-statistic is recomputed on each permuted split of the pooled data, and the p-value is the proportion of permuted statistics at least as extreme as the observed one. This permutation scheme relies on group-relabeling exchangeability, which preserves H_0 only when $A_0 = 0.5$; general A0 is therefore not currently supported in this small-sample regime and will raise an error.

For length(x) + length(y) >= 20, EU estimates $\mathrm{Var}(\hat{A})$ by the exact finite-sample unbiased combination derived from the Hoeffding decomposition of the mid-rank kernel, with Welch–Satterthwaite degrees of freedom.

x is taken to represent cases and y the reference/control group, matching the convention of wilcox.test(). Internally, the test statistic and variance components are computed in the $P(X < Y)$ framework.

Value

Numeric p-value.

---

wmwAUC_test *wmwAUC Test of No Group Discrimination*

Description

Performs the wmwAUC test of $\mathrm{H}_0 : \mathrm{AUC} = 0.5$ based on the Wilcoxon-Mann-Whitney statistic.

Usage

```
wmwAUC_test(
  formula,
  data,
  ref_level = NULL,
  pvalue_method = c("EU", "BC"),
  alternative = c("two.sided", "greater", "less"),
  ci_method = "delong",
  conf_level = 0.95,
  pseudomedian = FALSE,
  n_grid = 1000,
  ...
)
```

Arguments

formula	Formula of the form response ~ group
data	Data frame containing continuous response variable and grouping factor
ref_level	Character, reference level of grouping factor (if NULL, uses first level)
pvalue_method	Character, method ('EU', 'BC') used for computing p-values (default 'EU')
alternative	Character, alternative hypothesis is c("two.sided", "greater", "less")
ci_method	Character, confidence interval method for eAUC: c('delong', 'boot', 'none')
conf_level	Numeric, confidence level for intervals (default 0.95)
pseudomedian	Logical; if TRUE, additionally reports the AUC-equalizing shift (pseudomedian) and its confidence interval (default FALSE)
n_grid	Numeric, number of grid points for search in wmwAUC_pseudomedian_ci() (default 1000)
...	Additional arguments passed to roc_with_ci()

Details

The function tests the null hypothesis $\mathrm{H}_0\colon \mathrm{AUC} = 0.5$ against $\mathrm{H}_1\colon \mathrm{AUC} \neq 0.5$, where AUC represents the Area Under the ROC Curve.

The Exact Unbiased ('EU') method is used by default for computing p-values. Bias-Corrected ('BC') method is available through pvalue_method = 'BC' and is markedly conservative at small or imbalanced sample sizes; EU is recommended unless BC's specific properties are wanted (see wmwAUC_pvalue_BC).

Following the convention of wilcox.test() AUC equals the probability $P(X > Y)$ that a randomly selected observation from the first group exceeds a randomly selected observation from the second group. For response ~ group, observations from the non-reference group constitute X, while observations from the reference group (specified by ref_level) constitute Y. Thus AUC = P(non-reference > reference). If ref_level is not specified, the first factor level is used as reference. The U-statistic and the resulting empirical AUC (eAUC) are calculated consistently with this group assignment.

The test statistic is eAUC, which estimates the true AUC. The empirical ROC curve (eROC) is constructed by varying the classification threshold across all observed values and computing sensitivity and 1-specificity at each threshold.

When `pseudomedian = TRUE`, the function additionally reports the AUC-equalizing shift δ, defined as the value solving $P(X < Y + \delta) = 0.5$; see `wmwAUC_pseudomedian_ci` for details.

Confidence intervals for the true AUC are computed using either the DeLong et al. (1988) structural-components method based on asymptotic normality, or bootstrap resampling. If bootstrap resampling is selected, it is also used for constructing the confidence band for the ROC curve.

This function can call two independent sources of randomness: bootstrap resampling (`ci_method = 'boot'`), and, when `pseudomedian = TRUE` with a small sample (`n1 + n2 < 20`), `wmwAUC_pvalue_EU`'s Monte Carlo permutation fallback, called once per grid point inside `wmwAUC_pseudomedian_ci`. For reproducible results, call `set.seed()` immediately before `wmwAUC_test()` rather than relying on the ambient RNG state; a single seed covers both sources, since they draw from the same stream in a fixed order.

Value

Object of class 'wmwAUC_test' containing:

`pseudomedian_requested`
: Logical indicating whether the pseudomedian was computed

`n`
: Named vector with components n1, n2 giving sample sizes for each group

`U_statistic`
: U statistic

`p_value`
: P-value for testing H0: AUC = 0.5

`alternative`
: Alternative hypothesis specification

`pvalue_method`
: Character string describing the test method

`data_name`
: Character string giving the name of the data

`pseudomedian`
: AUC-equalizing shift estimate (when pseudomedian = TRUE)

`pseudomedian_conf_int`
: Confidence interval for AUC-equalizing shift (when pseudomedian = TRUE)

`pseudomedian_conf_level`
: Confidence level for the pseudomedian interval (when pseudomedian = TRUE)

`ci_method`
: Method used to compute confidence interval for AUC

`roc_object`
: ROC analysis object returned by `roc_with_ci()`

`auc`
: Empirical AUC (eAUC), the standardized U statistic

`auc_conf_int`
: Confidence interval for true AUC using DeLong et al. or bootstrap method

`x_vals`
: Numeric vector of observations from non-reference group

`y_vals`
: Numeric vector of observations from reference group

`groups`
: Character vector of group labels from original data

`group_levels`
: Character vector of factor levels for grouping variable

`group_ref_level`
: Character string indicating which level corresponds to reference group

References

Grendar, M. (2025). Wilcoxon-Mann-Whitney test of no group discrimination. arXiv:2511.20308. (Full bibliography, including all methods and sources cited throughout this package, is given there.)

See Also

print.wmwAUC_test for formatted output of wmwAUC_test(). plot.wmwAUC_test for plot of output of wmwAUC_test(). wmwAUC_pvalue_BC for details on computing p-values using the 'BC' method. wmwAUC_pvalue_EU for details on computing p-values using the 'EU' method. wmwAUC_pseudomedian_ci for details on computing confidence intervals for the pseudomedian. wilcox.test for the Wilcoxon-Mann-Whitney test in base R. rank.two.samples in package **rankFD** for an implementation of the Brunner-Munzel test.

Examples

```
library('wmwAUC')

library('gemR')
data(MS)
da <- MS
# preparing data frame
class(da$proteins) <- setdiff(class(da$proteins), "AsIs")
df <- as.data.frame(da$proteins)
df$MS <- da$MS
# wmwAUC test
wmd <- wmwAUC_test(P19099 ~ MS, data = df, ref_level = 'no')
wmd
plot(wmd)
# compute pseudomedian
set.seed(123L)
wmd_pm <- wmwAUC_test(P19099 ~ MS, data = df, ref_level = 'no', pseudomedian = TRUE)
wmd_pm
# compute confint for AUC by bootstrap
set.seed(123L)
wmd_ci_boot <- wmwAUC_test(P19099 ~ MS, data = df, ref_level = 'no', ci_method = 'boot')
wmd_ci_boot
plot(wmd_ci_boot)
# BC method
wmd_bc <- wmwAUC_test(P19099 ~ MS, data = df, ref_level = 'no', pvalue_method = 'BC')
wmd_bc
```

Index