Package ‘kendallknight’

February 20, 2025

Type Package

Title Efficient Implementation of Kendall's Correlation Coefficient Computation

Version 0.6.0

Imports stats

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Depends R(>= 3.5.0)

Description The computational complexity of the implemented algorithm for Kendall's correlation is O(n log(n)), which is faster than the base R implementation with a computational complexity of O(n^2). For small vectors (i.e., less than 100 observations), the time difference is negligible. However, for larger vectors, the speed difference can be substantial and the numerical difference is minimal. The references are
Knight (1966) <doi:10.2307/2282833>,
Abrevaya (1999) <doi:10.1016/S0165-1765(98)00255-9>,
Christensen (2005) <doi:10.1007/BF02736122> and
Emara (2024) <https://learningcpp.org/>.
This implementation is described in
Vargas Sepulveda (2024) <doi:10.48550/arXiv.2408.09618>.

License Apache License (>= 2)

BugReports https://github.com/pachadotdev/kendallknight/issues

URL https://pacha.dev/kendallknight/,
https://github.com/pachadotdev/kendallknight

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

LinkingTo cpp11

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

LazyData true

Author Mauricio Vargas Sepulveda [aut, cre]
(<https://orcid.org/0000-0003-1017-7574>),
Loader Catherine [ctb] (original stirlerr implementations in C (2000)),
Ross Ihaka [ctb] (original chebyshev_eval, gammafn and lgammacor
implementations in C (1998))

Maintainer Mauricio Vargas Sepulveda <m.sepulveda@mail.utoronto.ca>

Repository CRAN

Date/Publication 2025-02-20 22:10:01 UTC

Contents

`kendallknight-package` *kendallknight: Efficient Implementation of Kendall's Correlation Coefficient Computation*

Description

The computational complexity of the implemented algorithm for Kendall's correlation is O(n log(n)), which is faster than the base R implementation with a computational complexity of O(n^2). For small vectors (i.e., less than 100 observations), the time difference is negligible. However, for larger vectors, the speed difference can be substantial and the numerical difference is minimal. The references are Knight (1966) doi:10.2307/2282833, Abrevaya (1999) doi:10.1016/S01651765(98)00255-9, Christensen (2005) doi:10.1007/BF02736122 and Emara (2024) https://learningcpp.org/. This implementation is described in Vargas Sepulveda (2024) doi:10.48550/arXiv.2408.09618.

Author(s)

Maintainer: Mauricio Vargas Sepulveda <m.sepulveda@mail.utoronto.ca> (ORCID)

Other contributors:

- Loader Catherine (original stirlerr implementations in C (2000)) [contributor]
- Ross Ihaka (original chebyshev_eval, gammafn and lgammacor implementations in C (1998)) [contributor]

See Also

Useful links:

- https://pacha.dev/kendallknight/
- https://github.com/pachadotdev/kendallknight
- Report bugs at https://github.com/pachadotdev/kendallknight/issues

---

`cigarettes` *Life expectancy and cigarettes per day*

Description

A dataset containing life expectancy and cigarettes per day.

Usage

```
cigarettes
```

Format

A data frame with 15 rows and 2 variables:

life_expectancy Life expectancy in years.

cigarettes_per_day Cigarettes smoked per day.

Source

Real Statistics Using Excel (https://real-statistics.com/correlation/kendalls-tau-correlation/kendalls-correlation-testing-with-ties/).

Examples

```
cigarettes
```

kendall_cor *Kendall Correlation*

Description

kendall_cor() calculates the Kendall correlation coefficient between two numeric vectors. It uses the algorithm described in Knight (1966), which is based on the number of concordant and discordant pairs. The computational complexity of the algorithm is $O(n \log(n))$, which is faster than the base R implementation in stats::cor(..., method = "kendall") that has a computational complexity of $O(n^2)$. For small vectors (i.e., less than 100 observations), the time difference is negligible. However, for larger vectors, the difference can be substantial.

By construction, the implementation drops missing values on a pairwise basis. This is the same as using stats::cor(..., use = "pairwise.complete.obs").

Usage

```
kendall_cor(x, y = NULL)
```

Arguments

x	a numeric vector or matrix.
y	an optional numeric vector.

Value

A numeric value between -1 and 1.

References

Knight, W. R. (1966). "A Computer Method for Calculating Kendall's Tau with Ungrouped Data". Journal of the American Statistical Association, 61(314), 436–439.

Abrevaya J. (1999). Computation of the Maximum Rank Correlation Estimator. Economic Letters 62, 279-285.

Christensen D. (2005). Fast algorithms for the calculation of Kendall's Tau. Journal of Computational Statistics 20, 51-62.

Emara (2024). Khufu: Object-Oriented Programming using C++

Examples

```
# input vectors -> scalar output
x <- c(1, 0, 2)
y <- c(5, 3, 4)
kendall_cor(x, y)

# input matrix -> matrix output
x <- mtcars[, c("mpg", "cyl")]
kendall_cor(x)
```

kendall_cor_test *Kendall Correlation Test*

Description

`kendall_cor_test()` calculates p-value for the the Kendall correlation using the exact values when the number of observations is less than 50. For larger samples, it uses an approximation as in base R.

Usage

```
kendall_cor_test(x, y, alternative = c("two.sided", "greater", "less"))
```

Arguments

`x`	a numeric vector.
`y`	a numeric vector.
`alternative`	a character string specifying the alternative hypothesis. The possible values are `"two.sided"`, `"greater"`, and `"less"`.

Value

A list with the following components:

`statistic`	The Kendall correlation coefficient.
`p_value`	The p-value of the test.
`alternative`	A character string describing the alternative hypothesis.

References

Knight, W. R. (1966). "A Computer Method for Calculating Kendall's Tau with Ungrouped Data". Journal of the American Statistical Association, 61(314), 436–439.

Abrevaya J. (1999). Computation of the Maximum Rank Correlation Estimator. Economic Letters 62, 279-285.

Christensen D. (2005). Fast algorithms for the calculation of Kendall's Tau. Journal of Computational Statistics 20, 51-62.

Emara (2024). Khufu: Object-Oriented Programming using C++

Examples

```
x <- c(1, 0, 2)
y <- c(5, 3, 4)
kendall_cor_test(x, y)
```

Index