

Package ‘zmij’

August 4, 2026

Title Round-Trip-Safe Double-Precision Formatting

Version 0.1.0

Description Converts double-precision floating-point values to decimal strings with the shortest significands needed for round-trip recovery. Uses the 'zmij' algorithm through a vendored 'Rust' implementation. The method is described in Zverovich (2025), ``Faster Double-to-String Conversion" <<https://vitaut.net/posts/2025/faster-dtoa/>>. Provides vectorized formatting, a correctly rounded inverse parser, and explicit handling of missing and non-finite values.

License MIT + file LICENSE

Copyright file inst/COPYRIGHTS

URL <https://nanx.me/zmij/>, <https://github.com/nanxstats/zmij>

BugReports <https://github.com/nanxstats/zmij/issues>

Encoding UTF-8

Depends R (>= 4.2)

Suggests testthat (>= 3.0.0)

SystemRequirements Cargo (Rust's package manager), rustc >= 1.71.0, xz

Config/rextendr/version 0.5.0

Config/testthat/edition 3

RoxygenNote 8.0.0

NeedsCompilation yes

Author Nan Xiao [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-0250-5673>>),
David Tolnay [ctb, cph] (Author of the vendored Rust zmij crate),
Victor Zverovich [ctb, cph] (Author of the zmij algorithm and original
C++ implementation)

Maintainer Nan Xiao <me@nanx.me>

Repository CRAN

Date/Publication 2026-08-04 16:20:02 UTC

Contents

format_double	2
parse_double	3
Index	4

format_double	<i>Format Double-Precision Values with zmij</i>
---------------	---

Description

Convert numeric values to decimal strings with the shortest significands that recover the original IEEE 754 binary64 values when parsed using round-to-nearest, ties-to-even rounding.

Usage

format_double(x)

Arguments

x A numeric vector, matrix, or array.

Details

Integer inputs are converted to double precision before formatting. zmij uses fixed notation for decimal exponents from -5 through 15 and scientific notation outside that range. Other attributes, including classes, are not retained because the returned values are character strings.

Value

A character vector with the same length and shape as x. Names and dimensions are preserved. Missing values remain NA; NaN, positive infinity, and negative infinity become "NaN", "inf", and "-inf", respectively.

References

Zverovich V (2025). "Faster double-to-string conversion." <https://vitaut.net/posts/2025/faster-dtoa/>.

Tolnay D. "zmij: A double-to-string conversion algorithm based on Schubfach." <https://github.com/dtolnay/zmij>.

Examples

```
x <- c(pi, 0.1, -0, .Machine$double.xmin, .Machine$double.xmax)
formatted <- format_double(x)
formatted
identical(parse_double(formatted), x)

format_double(c(NA_real_, NaN, Inf, -Inf))
```

parse_double

Parse Double-Precision Strings

Description

Parse decimal strings into IEEE 754 binary64 values using Rust's correctly rounded parser. This is the inverse of `format_double()` for every finite double.

Usage

```
parse_double(x)
```

Arguments

`x` A character vector, matrix, or array.

Details

The parser accepts decimal and scientific notation, along with "NaN", "inf", and "-inf". Leading or trailing whitespace and other invalid strings cause an error that identifies the first invalid element.

Using `parse_double()` avoids platform-specific edge cases in R's built-in decimal parser near the limits of the binary64 range.

Value

A double vector with the same length and shape as `x`. Names and dimensions are preserved. Missing strings remain `NA_real_`.

Examples

```
x <- c(pi, 0.1, -0, .Machine$double.xmin, .Machine$double.xmax)
identical(parse_double(format_double(x)), x)

parse_double(c("1.5", "3e-1", NA_character_, "inf"))
```

Index

`format_double`, [2](#)
`format_double()`, [3](#)
`parse_double`, [3](#)