Package ‘waldo’

April 16, 2020

Title Find Differences Between R Objects

Version 0.1.0

Description Compare complex R objects and reveal the key
differences. Designed particularly for use in testing packages where
being able to quickly isolate key differences makes understanding test
failures much easier.

License MIT + file LICENSE

URL https://github.com/r-lib/waldo

BugReports https://github.com/r-lib/waldo/issues

Imports cli, diffobj, fansi, glue, methods, rematch2, rlang, tibble

Suggests testthat, covr, R6

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Hadley Wickham [aut, cre],
RStudio [cph]

Maintainer Hadley Wickham <hadley@rstudio.com>

Repository CRAN

Date/Publication 2020-04-16 09:20:03 UTC

R topics documented:

  compare ................................................................. 2

Index 4
Description

This function is an alternative to `all.equal()` that attempts to provide a description of the differences that is more immediately understandable. It:

- Orders the differences from most important to least important.
- Displays the values of atomic vectors that are actually different.
- Carefully uses colour to emphasise changes (while still being readable when colour isn’t available).
- Uses R code (not a text description) to show where differences arise.
- Where possible, it compares elements by name, rather than by position.
- Errs on the side of producing too much output, rather than too little.

Usage

```r
compare(
  x,
  y,
  x_arg = "x",
  y_arg = "y",
  tolerance = NULL,
  ignore_srcref = TRUE,
  ignore_attr = FALSE,
  ignore_encoding = TRUE
)
```

Arguments

- `x, y` Objects to compare. `y` is treated as the reference object so messages describe how `x` is different to `y`.
- `x_arg, y_arg` Name of `x` and `y` arguments, used when generated paths to internal components.
- `tolerance` If non-NULL, used as threshold for ignoring small floating point difference when comparing numeric vectors. Setting to any non-NULL value will cause integer and double vectors to be compared based on their values, rather than their types. It uses the same algorithm as `all.equal()`, i.e., first we generate `x_diff` and `y_diff` by subsetting `x` and `y` to look only locations with differences. Then we check that \( \text{mean}(\text{abs}(x_{\text{diff}} - y_{\text{diff}})) / \text{mean}(\text{abs}(y_{\text{diff}})) \) (or just \( \text{mean}(\text{abs}(x_{\text{diff}} - y_{\text{diff}})) \) if `y_diff` is small) is less than `tolerance`.
- `ignore_srcref` Ignore differences in function `srcref`? TRUE by default since the `srcref` does not change the behaviour of a function, only its printed representation.
ignore_attr  Ignore all differences in attributes? Only provided for backward compatibility with all.equal(). Using TRUE is not generally recommended because it will ignore many important functional differences.

ignore_encoding  Ignore string encoding? TRUE by default, because this is R’s default behaviour. Use FALSE when specifically concerned with the encoding, not just the value of the string.

Value

A character vector with class "waldo_compare". If there are no differences it will have length 0; otherwise each element is contains the description of a single difference.

Examples

# Thanks to diffobj package comparison of atomic vectors shows differences
# with a little context
compare(letters, c("z", letters[-26]))
compare(c(1, 2, 3), c(1, 3))
compare(c(1, 2, 3), c(1, 3, 4, 5))
compare(c(1, 2, 3), c(1, 2, 5))

# More complex objects are traversed, stopping only when the types are different
compare(
  list(x = list(y = list(structure(1, z = 2)))),
  list(x = list(y = list(structure(1, z = "a"))))
)

# Where possible, recursive structures are compared by name
compare(iris, rev(iris))

compare(list(x = "x", y = "y"), list(y = "y", x = "x"))
# Otherwise they're compared by position
compare(list("x", "y"), list("x", "z"))
compare(list(x = "x", x = "y"), list(x = "x", y = "z"))
Index

all.equal(), 2

compare, 2