Package ‘versus’

January 12, 2024

Title Compare Data Frames
Version 0.3.0
Description A toolset for interactively exploring the differences between two data frames.
License MIT + file LICENSE
Suggests testthat (>= 3.0.0)
Config/testthat/edition 3
Encoding UTF-8
RoxygenNote 7.2.3
Imports rlang (>= 1.1.0), cli, dplyr (>= 1.1.0), glue, tidyselect (>= 1.2.0), vctrs (>= 0.6.4), tibble, pillar, purrr, collapse (>= 2.0.9), data.table
BugReports https://github.com/eutwt/versus/issues
Depends R (>= 4.1.0)
LazyData true
Config/Needs/website markdown
NeedsCompilation yes
Author Ryan Dickerson [aut, cre, cph]
Maintainer Ryan Dickerson <fresh.tent5866@fastmail.com>
Repository CRAN
Date/Publication 2024-01-12 00:30:02 UTC

R topics documented:

compare ................................................. 2
example_df_a ......................................... 3
example_df_b ......................................... 4
slice_diffs .......................................... 4
slice_unmatched ..................................... 5
value_diffs .......................................... 6
weave_diffs_long .................................... 7
**compare**

Index

| compare | Compare two data frames |

**Description**

`compare()` creates a representation of the differences between two tables, along with a shallow copy of the tables. This output is used as the `comparison` argument when exploring the differences further with other versus functions e.g. `slice_*()` and `weave_*()`.

**Usage**

```r
compare(table_a, table_b, by, allow_both_NA = TRUE, coerce = TRUE)
```

**Arguments**

- `table_a`: A data frame
- `table_b`: A data frame
- `by`: <tidy-select>. Selection of columns to use when matching rows between `.data_a` and `.data_b`. Both data frames must be unique on `by`.
- `allow_both_NA`: Logical. If `TRUE` a missing value in both data frames is considered as equal
- `coerce`: Logical. If `FALSE` and columns from the input tables have differing classes, the function throws an error.

**Value**

`compare()` A list of data frames having the following elements:

- **tables**: A data frame with one row per input table showing the number of rows and columns in each.
- **by**: A data frame with one row per by column showing the class of the column in each of the input tables.
- **intersection**: A data frame with one row per column common to `table_a` and `table_b` and columns "n_diffs" showing the number of values which are different between the two tables, "class_a"/"class_b" the class of the column in each table, and "value_diffs" a (nested) data frame showing the the values in each table which are unequal and the by columns
- **unmatched_cols**: A data frame with one row per column which is in one input table but not the other and columns "table": which table the column appears in, "column": the name of the column, and "class": the class of the column.
- **unmatched_rows**: A data frame which, for each row present in one input table but not the other, contains the column "table" showing which table the row appears in and the by columns for that row.
data.table inputs

If the input is a data.table, you may want \texttt{compare()} to make a deep copy instead of a shallow copy so that future changes to the table don’t affect the comparison. To achieve this, you can set \texttt{options(versus.copy_data_table = TRUE)}.

Examples

\texttt{compare(example\_df\_a, example\_df\_b, by = \texttt{car})}

---

\begin{verbatim}
example\_df\_a
Modified version of datasets::mtcars - version a
\end{verbatim}

Description

A version of mtcars with some values altered and some rows/columns removed. Not for informational purposes, used only to demonstrate the comparison of two slightly different data frames. Since some values were altered at random, the values do not necessarily reflect the true original values. The variables are as follows:

Usage

\texttt{example\_df\_a}

Format

A data frame with 9 rows and 9 variables:

- **car** The rowname in the corresponding \texttt{datasets::mtcars} row
- **mpg** Miles/(US) gallon
- **cyl** Number of cylinders
- **disp** Displacement (cu.in.)
- **hp** Gross horsepower
- **drat** Rear axle ratio
- **wt** Weight (1000 lbs)
- **vs** Engine (0 = V-shaped, 1 = straight)
- **am** Transmission (0 = automatic, 1 = manual)

Source

**example_df_b**  
*Modified version of datasets::mtcars - version b*

**Description**
A version of mtcars with some values altered and some rows/columns removed. Not for informational purposes, used only to demonstrate the comparison of two slightly different data frames. Since some values were altered at random, the values do not necessarily reflect the true original values. The variables are as follows:

**Usage**
```r
example_df_b
```

**Format**
A data frame with 9 rows and 9 variables:
- **car** The rowname in the corresponding datasets::mtcars row
- **wt** Weight (1000 lbs)
- **mpg** Miles/(US) gallon
- **hp** Gross horsepower
- **cyl** Number of cylinders
- **disp** Displacement (cu.in.)
- **carb** Number of caruretors
- **drat** Rear axle ratio
- **vs** Engine (0 = V-shaped, 1 = straight)

**Source**

---

**slice_diffs**  
*Get rows with differing values*

**Description**
Get rows with differing values

**Usage**
```r
slice_diffs(comparison, table, column = everything())
```
**slice_unmatched**

**Arguments**

- **comparison**
  The output of `compare()`

- **table**
  One of "a" or "b" indicating which of the tables used to create `comparison` should be sliced

- **column**
  `<tidy-select>`. A row will be in the output if the comparison shows differing values for any columns matching this argument

**Value**

The input table is filtered to the rows for which `comparison` shows differing values for one of the columns selected by `column`

**Examples**

```r
comp <- compare(example_df_a, example_df_b, by = car)
comp |> slice_diffs("a", mpg)
comp |> slice_diffs("b", mpg)
comp |> slice_diffs("a", c(mpg, disp))
```

---

**Description**

Get rows in only one table

**Usage**

```r
slice_unmatched(comparison, table)
slice_unmatched_both(comparison)
```

**Arguments**

- **comparison**
  The output of `compare()`

- **table**
  One of "a" or "b" indicating which of the tables used to create `comparison` should be sliced

**Value**

- **slice_unmatched()**
  The table identified by `table` is filtered to the rows `comparison` shows as not appearing in the other table

- **slice_unmatched_both()**
  The output of `slice_unmatched()` for both input tables row-stacked with a column `table` indicating which table the row is from. The output contains only columns present in both tables.
value_diffs

Examples

```r
comp <- compare(example_df_a, example_df_b, by = car)
comp |> slice_unmatched("a")
comp |> slice_unmatched("b")

# slice_unmatched(comp, "a") output is the same as
example_df_a |> dplyr::anti_join(example_df_b, by = comp$by$column)

comp |> slice_unmatched_both()
```

---

value_diffs

*Get the differing values from a comparison*

Description

Get the differing values from a comparison

Usage

```r
value_diffs(comparison, column)
value_diffs_stacked(comparison, column = everything())
```

Arguments

- **comparison**: The output of `compare()`
- **column**: `<tidy-select>`. The output will show the differing values for the provided columns.

Value

- **value_diffs()**: A data frame with one row for each element of col found to be unequal between the input tables (table_a and table_b from the original compare() output) The output table has the column specified by column from each of the input tables, plus the by columns.
- **value_diffs_stacked()**, **value_diffs_all()**: A data frame containing the value_diffs() outputs for the specified columns combined row-wise using dplyr::bind_rows(). If dplyr::bind_rows() is not possible due to incompatible types, values are converted to character first. value_diffs_all() is the same as value_diffs_stacked() with column = everything().

Examples

```r
comp <- compare(example_df_a, example_df_b, by = car)
value_diffs(comp, disp)
value_diffs_stacked(comp, c(disp, mpg))
```
weave_diffs_long  Get differences in context

**Description**

Get differences in context

**Usage**

```r
weave_diffs_long(comparison, column = everything())
weave_diffs_wide(comparison, column = everything())
```

**Arguments**

- **comparison**
  
  The output of `compare()`

- **column**

  `<tidy-select>`. A row will be in the output if the comparison shows differing values for any columns matching this argument

**Value**

- **weave_diffs_wide()**

  The input `table_a` filtered to rows where differing values exist for one of the columns selected by `column`. The selected columns with differences will be in the result twice, one for each input table.

- **weave_diffs_long()**

  Input tables are filtered to rows where differing values exist for one of the columns selected by `column`. These two sets of rows (one for each input table) are interleaved row-wise.

**Examples**

```r
comp <- compare(example_df_a, example_df_b, by = car)
comp |> weave_diffs_wide(disp)
comp |> weave_diffs_wide(c(mpg, disp))
comp |> weave_diffs_long(disp)
comp |> weave_diffs_long(c(mpg, disp))
```
Index

* datasets
  example_df_a.3
  example_df_b.4

compare.2

example_df_a.3
example_df_b.4

slice_diffs.4
slice_unmatched.5
slice_unmatched_both(slice_unmatched).5

value_diffs.6
value_diffs_stacked(value_diffs).6

weave_diffs_long.7
weave_diffs_wide(weave_diffs_long).7