

# Package ‘tablecompare’

May 8, 2026

**Type** Package

**Title** Compare Data Frames

**Version** 0.1.1

**Description** A toolbox for comparing two data frames. This package is defunct. I recommend you use the ``versus" package instead.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** glue, magrittr, rlang (>= 0.4.3), tidymodels (>= 0.4.3), purrr

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://github.com/eutwt/tablecompare>

**BugReports** <https://github.com/eutwt/tablecompare/issues>

**Depends** data.table (>= 1.14.2)

**NeedsCompilation** no

**Author** Ryan Dickerson [aut, cre]

**Maintainer** Ryan Dickerson <fresh.tent5866@fastmail.com>

**Repository** CRAN

**Date/Publication** 2023-11-14 05:00:02 UTC

## Contents

contents	2
count_dupes	2
count_values	3
tblcompare	4

<b>Index</b>	<b>7</b>
--------------	----------

---

contents	<i>Show the contents of a data frame</i>
----------	--

---

**Description**

Show the contents of a data frame

**Usage**

```
contents(.data)
```

**Arguments**

`.data` A data frame or data table

**Value**

A data table with one row per column in `.data` and columns "column": The name of the column in `.data`, "class": the names of classes the column inherits from (as returned by `class()`), collapsed into a single string.

**Examples**

```
contents(ToothGrowth)
```

---

count_dupes	<i>Check for duplicate rows</i>
-------------	---------------------------------

---

**Description**

`count_dupes()` returns values of by variables for which the `.data` has multiple rows, along with the number of rows for each combination of values.

`assert_unique()` throws an error if there are multiple rows for any combination of by variable values

**Usage**

```
count_dupes(.data, by, setkey = FALSE)
```

```
assert_unique(.data, by, data_chr, by_chr)
```

**Arguments**

.data	A data frame or data table
by	tidy-select. Columns in .data
setkey	Logical. Should the output be keyed by by cols?
data_chr	optional. character. You can use this argument to manually specify the name of data shown in error messages. Useful when using these functions as checks inside other functions.
by_chr	optional. character. You can use this argument to manually specify the name of by shown in error messages. Useful when using these functions as checks inside other functions.

**Value**

count\_dupes() A data.table with the (filtered) by columns and an additional column "n\_rows" which shows the number of rows in .data having the combination of by values shown in the output row.

assert\_unique() No return value. Called to throw an error depending on the input.

**Examples**

```
df <- read.table(text = "
x y z
1 6 1
2 6 2
3 7 3
3 7 4
4 3 5
4 3 6
", header = TRUE)

count_dupes(df, c(x, y))

## Not run:
assert_unique(df, c(x, y))

## End(Not run)
```

---

count\_values

*Check for existence of multiple values per group*


---

**Description**

count\_values() returns values of by variables for which the .data has multiple unique rows, along with the number of unique rows for each combination of values, only considering columns in col.

assert\_single\_value() throws an error if there are multiple unique rows for any combination of by variable values, only considering columns in col.

**Usage**

```
count_values(.data, col, by, setkey = FALSE)
```

```
assert_single_value(.data, col, by)
```

**Arguments**

<code>.data</code>	A data frame or data table
<code>col</code>	tidy-select. Columns in <code>.data</code> . When counting the number of unique rows, only the columns specified in <code>col</code> are considered.
<code>by</code>	tidy-select. Columns in <code>.data</code> .
<code>setkey</code>	Logical. Should the output be keyed by <code>by</code> cols?

**Value**

`count_values()` A `data.table` with the (filtered) `by` columns and an additional column "n\_vals" which shows the number of unique rows in `.data` having the combination of `by` values shown in the output row.

`assert_single_value()` No return value. Called to throw an error depending on the input.

**Examples**

```
df <- read.table(text = "
x y z
a 1 3
a 1 3
a 2 4
a 2 4
a 2 2
b 1 1
b 1 2
", header = TRUE)

count_values(df, z, by = c(x, y))

## Not run:
assert_single_value(df, z, by = c(x, y))

## End(Not run)
```

**Description**

Compare two data frames. Using a key-column common to both tables, see which rows are common and highlight differing values by column.

**Usage**

```
tblcompare(
  .data_a,
  .data_b,
  by,
  allow_bothNA = TRUE,
  ncol_by_out = 3,
  coerce = TRUE
)

value_diffs(comparison, col)

## S3 method for class 'tblcmp_compare'
value_diffs(comparison, col)

all_value_diffs(comparison)

## S3 method for class 'tblcmp_compare'
all_value_diffs(comparison)
```

**Arguments**

<code>.data_a</code>	A data frame or data table
<code>.data_b</code>	A data frame or data table
<code>by</code>	tidy-select. Selection of columns to use when matching rows between <code>.data_a</code> and <code>.data_b</code> . Both data frames must be unique on <code>by</code> .
<code>allow_bothNA</code>	Logical. If TRUE a missing value in both data frames is considered as equal
<code>ncol_by_out</code>	Number of by-columns to include in <code>col_diffs</code> and <code>unmatched_rows</code> output
<code>coerce</code>	Logical. If False only columns with the same class are compared.
<code>comparison</code>	An object of class "tblcmp_compare" (the output of a <code>tblcompare::tblcompare()</code> call)
<code>col</code>	tidy-select. A single column

**Value**

`tblcompare()` A "tblcmp\_compare"-class object, which is a list of data.table's having the following elements:

**tables** A data.table with one row per input table showing the number of rows and columns in each.

**by** A data.table with one row per by column showing the class of the column in each of the input tables.

**summ** A `data.table` with one row per column common to `.data_a` and `.data_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.table` showing the rows in each input table where values are unequal, the values in each table, and one column for each of the first `ncol_by_out` by columns for the identified rows in the input tables.

**unmatched\_cols** A `data.table` 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.table` which, for each row present in one input table but not the other, contains the columns `"table"`: which table the row appears in, `"i"` the row number of the input row, and one column for each of the first `ncol_by_out` by columns for each row.

`value_diffs()` A `data.table` with one row for each element of `col` found to be unequal between the input tables (`.data_a` and `.data_b` from the original `tblcompare()` call) The output table has columns `"i_a"/"i_b"`: the row number of the element in the input tables, `"val_a"/"val_b"`: the value of `col` in the input tables, and one column for each of the first `ncol_by_out` by columns for the identified rows in the input tables.

`all_value_diffs()` A `data.table` of the `value_diffs()` output for all columns having at least one value difference, combined row-wise into a single table. To facilitate this combination into a single table, the `"val_a"` and `"val_b"` columns are coerced to character.

# Index

`all_value_diffs (tblcompare)`, 4  
`assert_single_value (count_values)`, 3  
`assert_unique (count_dupes)`, 2  
  
`contents`, 2  
`count_dupes`, 2  
`count_values`, 3  
  
`tblcompare`, 4  
  
`value_diffs (tblcompare)`, 4