Package ‘compareDF’

January 6, 2021

Type Package

Title Do a Git Style Diff of the Rows Between Two Dataframes with Similar Structure

Version 2.3.1

Date 2021-01-06

Description Compares two dataframes which have the same column structure to show the rows that have changed. Also gives a git style diff format to quickly see what has changed in addition to summary statistics.

License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports dplyr (>= 1.0.0), data.table (>= 1.12.8), magrittr (>= 1.5), htmlTable (>= 1.5), openxlsx (>= 4.1), tidyr (>= 1.1.0), stringr (>= 1.4.0), tibble (>= 3.0.1)

Suggests testthat, futile.logger, covr

LazyData TRUE

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Alex Joseph [aut, cre]

Maintainer Alex Joseph <alexsanjoseph@gmail.com>

Repository CRAN

Date/Publication 2021-01-06 19:40:02 UTC

R topics documented:

compare_df *Compare Two dataframes*

Description

Do a git style comparison between two data frames of similar columnar structure

Usage

```
compare_df(
  df_new,
  df_old,
  group_col,
  exclude = NULL,
  tolerance = 0,
  tolerance_type = "ratio",
  stop_on_error = TRUE,
  keep_unchanged_rows = FALSE,
  keep_unchanged_cols = TRUE,
  change_markers = c("+", "-", "="),
  round_output_to = 3
)
```

Arguments

`df_new` The data frame for which any changes will be shown as an addition (green)

`df_old` The data frame for which any changes will be shown as a removal (red)

`group_col` A character vector of a string of character vector showing the columns by which to group_by.

`exclude` The columns which should be excluded from the comparison

`tolerance` The amount in fraction to which changes are ignored while showing the visual representation. By default, the value is 0 and any change in the value of variables is shown off. Doesn't apply to categorical variables.

`tolerance_type` Defaults to 'ratio'. The type of comparison for numeric values, can be 'ratio' or 'difference'

`stop_on_error` Whether to stop on acceptable errors on not

`keep_unchanged_rows` whether to preserve unchanged values or not. Defaults to `FALSE`

`keep_unchanged_cols` whether to preserve unchanged values or not. Defaults to `TRUE`

`change_markers` what the different change_type nomenclature should be eg: c("new", "old", "unchanged").

`round_output_to` Number of digits to round the output to. Defaults to 3.

Examples

```
old_df = data.frame(var1 = c("A", "B", "C"),
                    val1 = c(1, 2, 3))
new_df = data.frame(var1 = c("A", "B", "C"),
                    val1 = c(1, 2, 4))
ctable = compare_df(new_df, old_df, c("var1"))
print(ctable$comparison_df)
ctable$html_output
```

`create_output_table` *Create human readable output from the comparison_df output*

Description

Currently ‘html‘ and ‘xlsx‘ are supported

Usage

```
create_output_table(
  comparison_output,
  output_type = "html",
  file_name = NULL,
  limit = 100,
  color_scheme = c(addition = "#52854C", removal = "#FC4E07", unchanged_cell =
    "#999999", unchanged_row = "#293352"),
  headers = NULL,
  change_col_name = "chng_type",
  group_col_name = "grp"
)
```

Arguments

`comparison_output`
Output from the comparison Table functions

`output_type` Type of comparison output. Defaults to ‘html‘

`file_name` Where to write the output to. Default to NULL which output to the Rstudio viewer (not supported for ‘xlsx‘)

`limit` maximum number of rows to show in the diff. >1000 not recommended for HTML

`color_scheme` What color scheme to use for the output. Should be a vector/list with named_elements. Default - `c("addition" = "green","removal" = "red","unchanged_cell" = "gray","unchanged_row" = "deepskyblue")`

`headers` A character vector of column names to be used in the table. Defaults to `colnames`.

`change_col_name`
Name of the change column to use in the table. Defaults to `chng_type`.

`group_col_name` Name of the group column to be used in the table (if there are multiple grouping vars). Defaults to `grp`.

`create_wide_output` *Convert to wide format*

Description

Easier to compare side-by-side

Usage

```
create_wide_output(comparison_output, suffix = c("_new", "_old"))
```

Arguments

`comparison_output`
Output from the comparison Table functions

`suffix` Nomenclature for the new and old dataframe

`results_2010` *Data set created set to show off the package capabilities - Results of students for 2010*

Description

A manually created dataset showing the hypothetical scores of two divisions of students

- Division The division to which the student belongs
- Student Name of the Student
- Maths, Physics, Chemistry, Art Scores of the student across different subjects
- Discipline, PE Grades of the students across different subjects

Usage

```
results_2010
```

Format

A data frame 12 rows and 8 columns

`results_2011` *Data set created set to show off the package capabilities - Results of students for 2011*

Description

A manually created dataset showing the hypothetical scores of two divisions of students

- Division The division to which the student belongs
- Student Name of the Student
- Maths, Physics, Chemistry, Art Scores of the student across different subjects
- Discipline, PE Grades of the students across different subjects

Usage

```
results_2011
```

Format

A data frame 13 rows and 8 columns

`view_html` *View Comparison output HTML*

Description

Some versions of Rstudio doesn't automatically show the html pane for the html output. This is a workaround

Usage

```
view_html(comparison_output)
```

Arguments

`comparison_output`
output from the comparisonDF compare function

Examples

```
old_df = data.frame(var1 = c("A", "B", "C"),
                    val1 = c(1, 2, 3))
new_df = data.frame(var1 = c("A", "B", "C"),
                    val1 = c(1, 2, 4))
ctable = compare_df(new_df, old_df, c("var1"))
# Not Run::
# view_html(ctable)
```

Index