Package 'compareDF'

May 21, 2018

Type Package

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

Version 1.5.0

Date 2018-05-21

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 changes in addition to summary statistics.

License MIT + file LICENSE

Depends R (>= 3.3.0)

Imports dplyr (>= 0.4.3), htmlTable (>= 1.5), tidyr (>= 0.4.1), stringr (>= 1.0.0)

Suggests testthat

LazyData TRUE

RoxygenNote 6.0.1

NeedsCompilation no

Author Alex Joseph [aut, cre]

Maintainer Alex Joseph <alexsanjoseph@gmail.com>

Repository CRAN

Date/Publication 2018-05-21 16:03:53 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, limit_html = 100,
  tolerance = 0, stop_on_error = TRUE, keep_unchanged = FALSE,
  color_scheme = c(addition = "green", removal = "red", unchanged_cell =
  "gray", unchanged_row = "deepskyblue"))
```

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
limit_html	maximum number of rows to show in the html diff. >1000 not recommended
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.
stop_on_error	Whether to stop on acceptable errors on not
keep_unchanged	whether to preserve unchanged values or not. Defaults to FALSE
color_scheme	What color scheme to use for the HTML output. Should be a vector/list with named_elements. Default - c("addition" = "green", "removal" = "red", "unchanged_cell" = "gray", "unchanged_row" = "deepskyblue")

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
```

`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