

# Package ‘dataCompare’

May 8, 2026

**Title** A 'shiny' App to Compare Two Data Frames

**Version** 1.0.5

**Description** A tool developed with the 'Golem' framework which provides an easier way to check cells differences between two data frames. The user provides two data frames for comparison, selects IDs variables identifying each row of input data, then clicks a button to perform the comparison. Several 'R' package functions are used to describe the data and perform the comparison in the server of the application. The main ones are `comparedf()` from 'arsenal' and `skim()` from 'skimr'. For more details see the description of `comparedf()` from the 'arsenal' package and that of `skim()` from the 'skimr' package.

**License** MIT + file LICENSE

**URL** <https://github.com/seewe/dataCompare>

**BugReports** <https://github.com/seewe/dataCompare/issues>

**Imports** arsenal, config, data.table, devtools, dplyr, DT, explore, golem, htmltools, magrittr, shiny, shinycssloaders, shinydashboard, shinydashboardPlus, shinyWidgets, skimr, tools, utils

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Sergio Ewane Ebouele [aut, cre]

**Maintainer** Sergio Ewane Ebouele <info@dataforknow.com>

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2024-11-28 08:00:02 UTC

## Contents

compare_data_frame_object . . . . .	2
compare_data_frame_object_report . . . . .	3
data_table_formatter . . . . .	4
run_data_compare_app . . . . .	4
same_variables . . . . .	5
skim_char . . . . .	5
skim_num . . . . .	6
<b>Index</b>	<b>7</b>

---

compare\_data\_frame\_object

*Function which perform the comparison of dataframe*

---

### Description

Function which perform the comparison of dataframe

### Usage

```
compare_data_frame_object(df1, df2, id_var)
```

### Arguments

df1	The first dataframe of the comparison
df2	The second dataframe of the comparison
id_var	The character vector containing id variables which identify the observations in dataframe 'df1' and datafram 'df2'

### Value

An object of class "comparedf" as made by the 'comparedf' S3 method is returned.

### Examples

```
library(dplyr)
compare_data_frame_object(
  iris %>% dplyr::mutate(ID = row_number()),
  iris %>% dplyr::mutate(ID = row_number()),
  'ID')
```

---

```
compare_data_frame_object_report
    run the rmarkdown report of the data comparison and save it in an
    external directory
```

---

## Description

run the rmarkdown report of the data comparison and save it in an external directory

## Usage

```
compare_data_frame_object_report(  
  df1,  
  df2,  
  ids,  
  report_title = "Comparator report",  
  report_author = "Author name here",  
  report_context = "Add a small text here to explain the context."  
)
```

## Arguments

df1	The first dataframe of the comparison
df2	The second dataframe of the comparison
ids	The character vector containing id variables which identify the observations in dataframe 'df1' and datafram 'df2'
report_title	A character string which contains the title of the report
report_author	A character string which contains the name of the report author
report_context	A character string containing the context of the report

## Value

null.

## Examples

```
library(dplyr)  
compare_data_frame_object_report(  
  df1 = iris %>% dplyr::mutate(ID = row_number()),  
  df2 = iris %>% dplyr::mutate(ID = row_number()),  
  ids = 'ID',  
  report_title = "Iris data Comparator report",  
  report_author = "Sergio Ewane",  
  report_context = "i'm checking if the two dataframe contains the same values"  
)
```

data\_table\_formatter *Datatable formatter, to print on the screen*

---

**Description**

Datatable formatter, to print on the screen

**Usage**

```
data_table_formatter(df, n_page = 5)
```

**Arguments**

df	dataframe to format
n_page	number of rows to display per page

**Value**

An object of class "htmlwidget" containing a formatted data.frame to print on app UI

**Examples**

```
data_table_formatter(iris, 10)
```

---

run\_data\_compare\_app *Run the dataCompare Shiny Application*

---

**Description**

Run the dataCompare Shiny Application

**Usage**

```
run_data_compare_app(...)
```

**Arguments**

...	list of arguments
-----	-------------------

**Value**

No return value, launch the app

---

same_variables	<i>Detect common variables in two dataset</i>
----------------	---

---

**Description**

Detect common variables in two dataset

**Usage**

```
same_variables(df1, df2)
```

**Arguments**

df1	the first dataset to use
df2	The second dataset to use

**Value**

a Character vector containing all variables names in both df1 and df2

**Examples**

```
same_variables(iris, iris)
same_variables(mtcars, mtcars)
```

---

skim_char	<i>Skim a dataset and return only characters variables characteristics</i>
-----------	--

---

**Description**

Skim a dataset and return only characters variables characteristics

**Usage**

```
skim_char(the_data)
```

**Arguments**

the_data	Data on which the skim function will apply the description on character variables
----------	---

**Value**

a data.frame object containing description of all character (factor, character or date) variable in the input data.

**Examples**

```
skim_char(iris)
skim_char(mtcars)
```

---

`skim_num`*Skim a dataset and return only numeric variables characteristics*

---

**Description**

Skim a dataset and return only numeric variables characteristics

**Usage**

```
skim_num(the_data)
```

**Arguments**

`the_data` Data on which the skim function will apply the description on numeric variables

**Value**

a data.frame object containing description of all numeric (double or integer) variable in the input data.

**Examples**

```
skim_num(iris)
skim_num(mtcars)
```

# Index

`compare_data_frame_object`, [2](#)  
`compare_data_frame_object_report`, [3](#)  
`data_table_formatter`, [4](#)  
`run_data_compare_app`, [4](#)  
`same_variables`, [5](#)  
`skim_char`, [5](#)  
`skim_num`, [6](#)