

Package ‘reproducibleCalculationTools’

March 10, 2021

Type Package

Title reproducibleCalculationTools: Utilities to compare consecutive data evaluation in R

Version 1.0

Date 2021-03-10

Author Thomas Braschler

Maintainer Thomas Braschler <thomas.braschler@gmail.com>

Description This package contains utilities to compare rda files produced in consecutive data evaluation.

Depends R (≥ 3.5.0)

License GPL-3 + file LICENSE

LazyLoad yes

R topics documented:

reproducibleCalculationTools-package	1
compare_rda_contents	2
value_comparison	5
Index	7

reproducibleCalculationTools-package
reproducibleCalculationTools

Description

Provides utilities to compare subsequent data evaluations. At present, this involves [compare_rda_contents](#), a function to compare the contents of rda files, and [value_comparison](#), to compare individual variables.

Author(s)

Thomas Braschler

Maintainer: Thomas Braschler <thomas.braschler@gmail.com>

compare_rda_contents	<i>compare_rda_contents</i>
----------------------	-----------------------------

Description

Compares the variables contained in two distinct .rda files to see if they contain the same data

Usage

```
compare_rda_contents(path1,path2,numerical_tolerance=0,verbose=TRUE,exclude_variables=vector(mod
```

Arguments

path1	Path to the first R-Data file (a .rda or .RData file that can be opened with load)
path2	Path to the second R-Data file (a .rda or .RData file that can be opened with load)
numerical_tolerance	Allow for some numerical tolerance. This is the relative deviation, as compared to the mean value of two corresponding entries
verbose	Should the function output what it is doing or remain silent
exclude_variables	Variables that should be excluded from comparison, for example locally varying access paths or randomly drawn variables. It is also possible to use a syntax of the type "variable/column" to exclude individual columns for variables that do have column names such as data.frame variables.
relative_tolerance_evaluation	Should numerical precision in comparison of corresponding values be check at the level of the absolute differences (the default) or relative to the mean absolute value of the two numbers being compared (pass TRUE) for this).

Details

The function checks that the two R-Data files contain objects having the same name, and also performs some checks on the objects contained. These checks include whether variables (functions are not analyzed) have the same column names, and whether the values are identical (to within `numerical_tolerance` if `numerical_tolerance>0`). The function is meant to validate equality, so it doesn't deal with differently structured variables, and also not with functions. Lists are compared element by element.

Regarding numerical precision, this can be either checked on an absolute level (the default, `relative_tolerance_evaluation`) or relative to the mean absolute value of pairs of numbers being compared (when passing `relative_tolerance_evaluation=TRUE`). The function also attempts to apply comparison within numerical precision by conversion of strings to numerical values, but not within [data.frame](#) columns that explicitly have string column type as this is considered to signal maintenance of string type.

The reason for implementation of explicit exclusion mechanism in our case is that when

running the same scripts on different operating systems or computers, most values remain very close or identical, but there are a number of operating-system or system specific values, typically file paths, that are not generally matching when comparing calculation output. This information is valuable and often necessary for processing, but needs to be excluded from comparison when assessing numerical reproducibility.

Value

TRUE indicating that the two .rda files contain identical variables (for as far as the tests go) or FALSE, indicating that there is some differences.

Author(s)

Thomas Braschler

Examples

```
# Example 1: Compare identical rda files
path_base = system.file("data_to_compare.rda", package="reproducibleCalculationTools")
identical_comparison=compare_rda_contents(path_base,path_base)
identical_comparison # Anticipated: TRUE

# Example 2: Rda file 2 has one more variable
path_base = system.file("data_to_compare.rda", package="reproducibleCalculationTools")
path_one_more_variable = system.file("one_more_variable.rda", package="reproducibleCalculationTools")
comparison_with_one_more_variable=compare_rda_contents(path_base,path_one_more_variable)
comparison_with_one_more_variable # Anticipated: FALSE

# Example 3: Rda file 2 has one more variable, excluded during comparison
path_base = system.file("data_to_compare.rda", package="reproducibleCalculationTools")
path_one_more_variable = system.file("one_more_variable.rda", package="reproducibleCalculationTools")
comparison_with_one_more_variable=compare_rda_contents(path_base,path_one_more_variable,exclude_variables=)
comparison_with_one_more_variable # Anticipated: TRUE

#Example 4: rda file 2 has a different entry for a numerical variable
path_base = system.file("data_to_compare.rda", package="reproducibleCalculationTools")
path_different_number = system.file("different_single_number.rda", package="reproducibleCalculationTools")
comparison_with_different_number=compare_rda_contents(path_base,path_different_number)
comparison_with_different_number # Anticipated: FALSE

#Example 5: rda file 2 has a different entry for a numerical variable, but allow for different levels of absolute
path_base = system.file("data_to_compare.rda", package="reproducibleCalculationTools")
path_different_number = system.file("different_single_number.rda", package="reproducibleCalculationTools")
comparison_with_different_number_large_tolerance=compare_rda_contents(path_base,path_different_number,numerical_tolerance=)
comparison_with_different_number_large_tolerance # Anticipated: TRUE

comparison_with_different_number_small_tolerance=compare_rda_contents(path_base,path_different_number,numerical_tolerance=)
comparison_with_different_number_small_tolerance # Anticipated: FALSE

#Example 6: rda file 2 has a different entry for a numerical variable, but allow for different levels of relative
path_base = system.file("data_to_compare.rda", package="reproducibleCalculationTools")
path_different_number = system.file("different_single_number.rda", package="reproducibleCalculationTools")
comparison_with_different_number_large_tolerance=compare_rda_contents(path_base,path_different_number,numerical_tolerance=)
comparison_with_different_number_large_tolerance # Anticipated: TRUE

comparison_with_different_number_small_tolerance=compare_rda_contents(path_base,path_different_number,numerical_tolerance=)
```

```
comparison_with_different_number_small_tolerance # Anticipated: FALSE
```

```
#Example 7: rda file 2 has a different entry for a numerical variable, but allow for different levels of relative
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_different_number = system.file("different_single_number.rda",package="reproducibleCalculationTools")
comparison_with_different_number_large_tolerance=compare_rda_contents(path_base,path_different_number,numerical_tolerance=0.01)
comparison_with_different_number_large_tolerance # Anticipated: TRUE
```

```
comparison_with_different_number_small_tolerance=compare_rda_contents(path_base,path_different_number,numerical_tolerance=0.01)
comparison_with_different_number_small_tolerance # Anticipated: FALSE
```

```
# Example 8: rda file 2 has a different string value
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_different_string = system.file("different_string.rda",package="reproducibleCalculationTools")
comparison_with_different_string=compare_rda_contents(path_base,path_different_string)
comparison_with_different_string # Anticipated: FALSE
# The string cannot be converted to numerical but this shouldn't interfere
comparison_with_different_string=compare_rda_contents(path_base,path_different_string,numerical_tolerance=0.01)
comparison_with_different_string # Anticipated: FALSE
```

```
# Example 9: rda file 2 has a different string value in its dataframe example
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_different_string_dataframe = system.file("string_change_in_dataframe.rda",package="reproducibleCalculationTools")
comparison_with_different_string_dataframe=compare_rda_contents(path_base,path_different_string_dataframe)
comparison_with_different_string_dataframe # Anticipated: FALSE
# The string cannot be converted to numerical but this shouldn't interfere
comparison_with_different_string_dataframe=compare_rda_contents(path_base,path_different_string_dataframe,numerical_tolerance=0.01)
comparison_with_different_string_dataframe # Anticipated: FALSE
```

```
# Example 10: rda file 2 has a different string value in its dataframe example, but the column is explicitly excluded
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_different_string_dataframe = system.file("string_change_in_dataframe.rda",package="reproducibleCalculationTools")
comparison_with_different_string_dataframe=compare_rda_contents(path_base,path_different_string_dataframe,exclude_variables="aNumber")
comparison_with_different_string_dataframe # Anticipated: TRUE
```

```
# Example 11: rda file 2 has a different string value in its dataframe example, exclusion of an unrelated column
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_different_string_dataframe = system.file("string_change_in_dataframe.rda",package="reproducibleCalculationTools")
comparison_with_different_string_dataframe=compare_rda_contents(path_base,path_different_string_dataframe,exclude_variables="aNumber")
comparison_with_different_string_dataframe # Anticipated: FALSE
```

```
# Example 12: rda file 2 has one more column in the dataframe
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_extra_column_dataframe = system.file("extra_column_in_dataframe.rda",package="reproducibleCalculationTools")
comparison_extra_column_in_dataframe=compare_rda_contents(path_base,path_extra_column_dataframe)
comparison_extra_column_in_dataframe # Anticipated: FALSE
```

```
# Example 13: rda file 2 has one more column in the dataframe, but this difference is explicitly excluded from comparison
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_extra_column_dataframe = system.file("extra_column_in_dataframe.rda",package="reproducibleCalculationTools")
comparison_extra_column_in_dataframe=compare_rda_contents(path_base,path_extra_column_dataframe,exclude_variables="aNumber")
comparison_extra_column_in_dataframe # Anticipated: TRUE
```

```
# Example 14: rda file 2 has the number variable "aNumber" replaced by corresponding string (i.e. "5.3942" instead of 5.3942)
path_base = system.file("data_to_compare.rda",package="reproducibleCalculationTools")
path_number_as_string = system.file("number_as_string.rda",package="reproducibleCalculationTools")
comparison_number_as_string=compare_rda_contents(path_base,path_number_as_string)
```

```
comparison_number_as_string # Anticipated: TRUE
```

value_comparison	<i>value_comparison</i>
------------------	-------------------------

Description

Compares two variables for identity to within specified precision

Usage

```
value_comparison(v1,v2,verbose=TRUE,numerical_tolerance=0,relative_tolerance_evaluation=FALSE)
```

Arguments

v1	First variable
v2	Second variable
numerical_tolerance	Allow for some numerical tolerance. This is the relative deviation, as compared to the mean value of two corresponding entries
verbose	Should the function output what it is doing or remain silent
relative_tolerance_evaluation	Should numerical precision in comparison of corresponding values be check at the level of the absolute differences (the default) or relative to the mean absolute value of the two numbers being compared (pass TRUE for this).

Details

This function looks for identity of content, not identity of memory storage. Further, this function has been designed as an aid to [compare.rda.contents](#). The variables should be reasonable, for instance, numbers, strings, vectors, or at most [data.frame](#). The function is not designed to handle variables pointing to functions or lists with various types.

Value

TRUE indicating that the variables are identical or at least sufficiently similar (for as far as the tests go) or FALSE, indicating that there is some differences.

Author(s)

Thomas Braschler

Examples

```
# Example 1: Identical string values
value_comparison("the same","the same",verbose=FALSE) #Anticipated: TRUE

# Example 2: Non-identical string values
value_comparison("the same","not the same",verbose=FALSE) # Anticipated: FALSE

# Example 3: Vectors
value_comparison(rep("the same",5),rep("the same",5),verbose=FALSE) # Anticipated: TRUE
value_comparison(rep("the same",5),rep("the same",4),verbose=FALSE) # Anticipated: FALSE
value_comparison(rep("the same",5),c(rep("the same",4),"not the same"),verbose=FALSE) # Anticipated: FALSE

# Example 4: Attempt at numerical conversion
value_comparison(4,"4") # Anticipated TRUE, the function tries to see whether the values could not be converted

#Example 5: Difference in a dataframe
dataframe1=data.frame(col_A=c(0,1,2,3), col_B=c("A","B","C","D"))
dataframe2=data.frame(col_A=c(0,1,2,4), col_B=c("A","B","C","D"))
value_comparison(dataframe1,dataframe1) # Anticipated TRUE
value_comparison(dataframe1,dataframe2) # Anticipated FALSE

#Example 6: Numerical tolerance
value_comparison(5,"5.00001",numerical_tolerance=1e-3) # Anticipated: TRUE
value_comparison("5.0002","5.00001",numerical_tolerance=1e-3) # Anticipated: TRUE
value_comparison("5.1002","5.00001",numerical_tolerance=1e-3) # Anticipated: FALSE
```

Index

- * **misc**
 - compare_rda_contents, [2](#)
 - value_comparison, [5](#)
- * **package**
 - reproducibleCalculationTools-package,
[1](#)
- compare_rda_contents, [1](#), [2](#), [5](#)
- data.frame, [2](#), [5](#)
- load, [2](#)
- reproducibleCalculationTools
(reproducibleCalculationTools-package),
[1](#)
- reproducibleCalculationTools-package,
[1](#)
- value_comparison, [1](#), [5](#)