Package: xldiff (via r-universe)

October 7, 2026

Title Compare excel sheets

Version 0.2.0

Description `xldiff` provides tools to compare excel sheets, broadly inspired by ``diff"-type functions. Provided functions can read sheets of two excel files and produce a third file that highlights cells that have changed. In the case of numeric changes, the direction of change is highlighted. These tools do not account for structural changes in the sheets (e.g., the addition of a column), but are useful in tracking changed values in tables or parameter files. Utility functions developed to streamline formatting output files are also more broadly useful in programmatically formatting excel files using openxlsx.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests openxlsx, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports cli, dplyr, flextable, glue, lifecycle, openxlsx2 (>= 1.22.0.9000), purrr, readxl, rlang, tibble, tidyr, tidyxl

URL https://framverse.r-universe.dev/xldiff, https://github.com/cbedwards-dfw/xldiff, https://cbedwards-dfw.github.io/xldiff/

Remotes JanMarvin/openxlsx2

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev

Repository https://framverse.r-universe.dev

Date/Publication 2026-02-03 23:36:54 UTC

RemoteUrl https://github.com/cbedwards-dfw/xldiff

RemoteRef HEAD

RemoteSha cfebb1a9cf7f440a10a08d0dcfbbdd34f6a75126

Contents

add_cell_borders . . . 2
add_changed_formats . . . 3
cell_range_translate . . . 4
cells_stylize . . . 5
diff_to_table . . . 6
excel_diff . . . 7
excel_diff_table . . . 8
excel_diff_tibble . . . 10
formula_diff . . . 11
mat_to_a1 . . . 12
present_rows_changed . . . 12
sheet_comp . . . 13
sheet_comp_basic . . . 15
wb_get_col_widths . . . 15

Index **17**

`add_cell_borders` **[Deprecated]** *Adds cell borders to openxlsx spreadsheet*

Description

Update: `excel_diff` now includes formatting! This function should hopefully be less necessary.

Usage

```
add_cell_borders(
  wb,
  sheet,
  block_ranges,
  sheet_start = "A1",
  every_cell = FALSE,
  border_col = "black",
  border_thickness = "medium"
)
```

Arguments

wb	openxlsx workbook object
sheet	character corresponding to sheet name of openxlsx workbook object wb.
block_ranges	One or more cell ranges specified in excel format (e.g. c("A1:D5, "B6", "A8:D8"))
sheet_start	Optional. If wb$sheet corresponds to an excel sheet in which the wb$sheet entries were read starting on a cell other than "A1" (e.g. readxl::read_excel with range specified or skip provided), provide the top left cell that was read into R in order to handle the offsetting, so that you can specify cell ranges based on the original excel file.
every_cell	Do we want borders around each individual cell in each cell block (TRUE), or just around the outer edges of the block (FALSE). Defaults to FALSE.
border_col	Color for border. See ?openxlsx::createStyle for details. Defaults to "black".
border_thickness	Thickness for border. See ?openxlsx::createStyle for details. Common choices: "thin", "thick".

Details

When calling sheet_diff(), creating a new workbook for the diff contents, and then coloring to highlight changed cells, the original spreadsheet formatting is lost. To facilitate interpretting the diff, it can be useful to recreate the major components of the original formatting, especially cell borders. This function adds cell borders, and is designed for ease of use when replicating formatting from the original excel file. Blocks of cells to give borders to can be specified in the original excel format (e.g. "A1:D5"). For only outside borders around each block (default), use argument every_cell = FALSE. To add all the cell borders within each block to generate a grid appearance, set every_cell = TRUE. Note that non-border formatting of each cell will not be maintained, but border formatting will be overwritten. When adding thin boundaries between inner cells and a thick outer border for a block of cells, first use add_cell_borders to with every_cell = TRUE, and appropriate border arguments (usually border_thickness = "thin") and then use again with every_cell = FALSE and appropriate border arguments (usually border_thickness = "medium").

---

add_changed_formats *Format* openxlsx2 *worksheet based on changes*

Description

Highlights cells that changed, coloring differently for increasing, decreasing, and non-numeric cells. Typically used on a worksheet that contains the $sheet.diff dataframe from the same sheet comparison as provided in the cur_sheet argument. In some cases it may be useful to define custom color schemes (e.g., if increasing numbers are good and decreasing numbers are bad, you may want green and red foregrounds for those types of changes). Individual colors can be provided for increasing, decreasing, or text changes. Similarly, increasing profits and decreasing costs logically should both show the same color. Optional arguments rows_invert, cols_invert, and df_invert allow you to specify individual regions of the dataframe to reverse the color scheme.

Usage

```
add_changed_formats(
  wb,
  cur_sheet,
  sheet_comp,
  rows_invert = NULL,
  cols_invert = NULL,
  df_invert = NULL,
  change_color = "#E1C0FF",
  pos_color = "#FFC7CE",
  neg_color = "#C6EFCE"
)
```

Arguments

wb	openxlsx2 workbook to make on which to make changes.
cur_sheet	sheet name in openxlsx2 workbook on which to make changes.
sheet_comp	list of comparison dataframes generated by `sheet_comp()`
rows_invert	Optional vector of row numbers to invert color scheme for increase vs decrease.
cols_invert	Optional vector of column numbers to invert color scheme for increase vs decrease.
df_invert	Optional data frame with `$row` and `$col` entries to identify individual cells to invert color schemes for increase vs decrease. Defaults to NULL.
change_color	Hex code for color to highlight cells that changed (and were not numerics). Defaults to lavendar ("#E1C0FF").
pos_color	As `change_color`, but for numeric cells that increase in value. Defaults to coral ("#E1C0FF").
neg_color	As `change_color`, but for numeric cells that decrease in value. Defaults to light green ("#C6EFCE")

---

cell_range_translate *Translates from excel cell address to rows and columns*

Description

Translates from excel cell address to rows and columns

Usage

```
cell_range_translate(x, expand = TRUE, start = "A1")
```

Arguments

x	Single string of individual cell or cell range (e.g. "D6" or "D6:AC8")
expand	If TRUE (default), provides the row and column for all cells in the range. If False, provides just the row and column of the start and end cells.
start	Optional argument to account for offset when matching cells in an excel file to a dataframe when the dataframe was generated by reading the excel file starting at a location other than "A1".

Value

dataframe of addresses for each cell in the range, where $row gives the row number and $col gives the column number.

Examples

```
cell_range_translate("D6")
cell_range_translate("A2:H3")
```

---

cells_stylize **[Deprecated]** *Apply style to worksheet based on one or more excel-style cell ranges*

Description

[Deprecated] Apply style to worksheet based on one or more excel-style cell ranges

Usage

```
cells_stylize(wb, sheet, style, block_ranges, stack = TRUE)
```

Arguments

wb	openxlsx workbook object
sheet	character corresponding to sheet name of openxlsx workbook object wb.
style	openxlsx cell style, created with openxlsx::createStyle(). This can include text size, bolding or italics, text wrapping, foreground color, text color, etc. See ?openxlsx::createStyle for details.
block_ranges	One or more cell ranges specified in excel format (e.g. c("A1:D5, "B6", "A8:D8"))
stack	Should style be appended to existing styles (TRUE) or replace existing styles (FALSE). Defaults to TRUE.

`diff_to_table` *Compares two dataframes and returns diff as flextable*

Description

Workhorse function for `excel_diff_table()`. Can be used to diff any pair of dataframes or tibbles. Diff is applied cell by cell, so results will only be meaningful if the shape of the two objects is the same.

Usage

```
diff_to_table(
  df1,
  df2,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  digits_show = 6,
  dfnames = NULL,
  excelify_col_names = FALSE
)
```

Arguments

`df1`	first dataframe or tibble
`df2`	second dataframe or tibble
`proportional_threshold`	Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. `proportional_threshold` will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).
`absolute_threshold`	Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override `proportional_threshold`. Numeric, defaults to NULL.
`digits_show`	When there is a change in number values, how many digits should be shown in `## ---> ##`? Numeric, defaults to 6. Recommend not making this so small that flagged changes don't get printed (e.g., if this is 2 and `proportional_threshold` is 0.001, 0.1% changes will get flagged, but only the first two digits will get shown).
`dfnames`	Character vector with names of the two dataframes to use for table title. Optional.
`excelify_col_names`	Should columns be identified by excel letter convention? Defaults to FALSE. Set to TRUE when comparing dataframes extracted from excel sheets.

Value

flextable object

Examples

```
df1 <- mtcars
df2 <- mtcars
df2[2, 2] <- -8
df2[5, c(3, 4)] <- c(11, 15)
diff_to_table(df1, df2)
```

`excel_diff` *Minimal spreadsheet comparison function*

Description

Compares a single sheet between two files, creating a new file that uses the formatting of the existing sheets (except highlighting) and highlights cells that differed. Small numeric differences between cell values can be the result of happenstance ("decimal dust") - the `digits_signif` and `proportional.diff` arguments can control behavior to ignore minor numeric differences.

Usage

```
excel_diff(
  file_1,
  file_2,
  results_name,
  sheet_name,
  sheet_name_file_2 = NULL,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  digits_show = 6,
  verbose = FALSE,
  extra_width = NULL
)
```

Arguments

`file_1`	Filename (including path) for first file to compare
`file_2`	Filename (including path) for second file to compare
`results_name`	Name (including path) for file to save comparison to. Must end in ".xlsx"
`sheet_name`	Character string of sheet to compare. Can provide vector of character strings to produce comparisons of multiple sheets.
`sheet_name_file_2`	OPTIONAL. Matching sheet names to `sheet_name` but for file 2. Use only if the two files have matching sheets with different names. Defaults to NULL.

`proportional_threshold`
: Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. `proportional_threshold` will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).

`absolute_threshold`
: Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override `proportional_threshold`. Numeric, defaults to NULL.

`digits_show`
: When there is a change in number values, how many digits should be shown in `## ---> ##`? Numeric, defaults to 6. Recommend not making this so small that flagged changes don't get printed (e.g., if this is 2 and `proportional_threshold` is 0.001, 0.1% changes will get flagged, but only the first two digits will get shown).

`verbose`
: Should sheet names be listed as they are diffed? Logical, defaults to TRUE

`extra_width`
: How much extra width should be added to columns that changed? Helpful to improve readability, since changed cells have longer entries. Numeric, defaults to 0.4.

See Also

`excel_diff_table()`, `excel_diff_tibble()`

Examples

```
## Not run:
filename_1 <- "Chin1124.xlsx"
filename_2 <- "Chin2524.xlsx"

excel_diff(
  file_1 = filename_1,
  file_2 = filename_2,
  results.name = "Chin1124 vs Chin 2524.xlsx",
  sheet_name = "ER_ESC_Overview_New"
)

## End(Not run)
```

`excel_diff_table` *Diff excel sheets and show table*

Description

Similar to excel_diff, but returns flextable that can be displayed and navigated in the Rstudio viewer. Only shows rows that have changed. For every row with changes, provides a row of before and after, highlighting changed vlaues (red for the value in file_1, green for the value in file_2)."ROWS"

column identifies excel row number and columns identify excel column names. Defaults to flagging changes of at least 0.1% from `file_1` to `file_2` (`proportional.diff = TRUE`, `digits.signif = 3`).

Usage

```
excel_diff_table(
  file_1,
  file_2,
  sheet_name,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  digits_show = 6
)
```

Arguments

`file_1`	Filename (including path) for first file to compare
`file_2`	Filename (including path) for second file to compare
`sheet_name`	Character string of a single excel sheet to compare between the files. (Unlike `excel_diff`, only one sheet can be compared at a time)
`proportional_threshold`	Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. `proportional_threshold` will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).
`absolute_threshold`	Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override `proportional_threshold`. Numeric, defaults to NULL.
`digits_show`	When there is a change in number values, how many digits should be shown in `## ---> ##`? Numeric, defaults to 6. Recommend not making this so small that flagged changes don't get printed (e.g., if this is 2 and `proportional_threshold` is 0.001, 0.1% changes will get flagged, but only the first two digits will get shown).

Value

flextable object.

See Also

`excel_diff()`, `excel_diff_tibble()`

Examples

```
## Not run:
excel_diff_table(
  file_1 = "C:/Repos/test file 1.xlsx",
  file_2 = "C:/Repos/test file 2.xlsx",
  sheet = "Sheet1"
)

## End(Not run)
```

`excel_diff_tibble` *Diff excel sheet and return tibble*

Description

Compares sheet from two excel files and returns a tibble of the diff.

Usage

```
excel_diff_tibble(
  file_1,
  file_2,
  sheet_name,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  digits_show = 6,
  trim_cols = FALSE,
  diff_only = FALSE
)
```

Arguments

`file_1`	Filename (including path) for first file to compare
`file_2`	Filename (including path) for second file to compare
`sheet_name`	Character string of a single excel sheet to compare between the files. (Unlike `excel_diff`, only one sheet can be compared at a time)
`proportional_threshold`	Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. `proportional_threshold` will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).
`absolute_threshold`	Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override `proportional_threshold`. Numeric, defaults to NULL.

digits_show	When there is a change in number values, how many digits should be shown in ## ---> ##? Numeric, defaults to 6. Recommend not making this so small that flagged changes don't get printed (e.g., if this is 2 and proportional_threshold is 0.001, 0.1% changes will get flagged, but only the first two digits will get shown).
trim_cols	Remove unchanged columns? Useful with wide dataframes when viewing results in the console. Defaults to FALSE.
diff_only	Show only the changed values? defaults to FALSE.

See Also

excel_diff(), excel_diff_table()

---

formula_diff *Compare two sheets to see if formula use changed*

Description

Identifies if one or more cells changed from having a formula to not between two excel sheets.

Usage

```
formula_diff(file_1, file_2, sheet_name)
```

Arguments

file_1	Filename (including path) for first file to compare
file_2	Filename (including path) for second file to compare
sheet_name	Character string of sheet to compare. Can provide vector of character strings to produce comparisons of multiple sheets.

Value

invisibly returns dataframe of the cell address and before/after formulas for cases in which a formula was present in both files but differed.

Examples

```
## Not run:
formula_diff(
  file_1 = "CohoPugetSoundTAMMInputTemplate2025.xlsx",
  file_2 = "CohoPugetSoundTAMMInputs 2025 v1.5.xlsx",
  sheet_name = "Input_Harvestnew"
)

## End(Not run)
```

mat_to_a1 *Provide Excel addresses from matrix of logicals*

Description

Useful when working with functions that need excel formats (e.g. `openxlsx2` functions). Does not try to simplify blocks of TRUEs into block addresses.

Usage

```
mat_to_a1(mat)
```

Arguments

`mat`	matrix of logical values

Value

vector of characters containing addresses in excel "A1" format.

Examples

```
mat <- matrix(FALSE, ncol = 4, nrow = 3)
mat[1, 1] <- TRUE
mat[2, 3] <- TRUE
mat
mat_to_a1(mat)
```

present_rows_changed *Summarize the rows changed between two dataframes*

Description

When given two dataframes, uses sheet_comp to compare the two dataframes, then presents the rows that have changed: prints the row numbers to the console, and then returns the "diff" of those rows. Optionally, can ignore columns that have not changed and can show only changed values.

Usage

```
present_rows_changed(
  t1,
  t2,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  digits_show = 6,
  trim_cols = FALSE,
  diff_only = FALSE
)
```

Arguments

t1	First dataframe
t2	Second dataframe, same dimensions as first.
proportional_threshold	Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. proportional_threshold will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).
absolute_threshold	Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override proportional_threshold. Numeric, defaults to NULL.
digits_show	When there is a change in number values, how many digits should be shown in ## ---> ##? Numeric, defaults to 6. Recommend not making this so small that flagged changes don't get printed (e.g., if this is 2 and proportional_threshold is 0.001, 0.1% changes will get flagged, but only the first two digits will get shown).
trim_cols	Remove unchanged columns? Useful with wide dataframes when viewing results in the console. Defaults to FALSE.
diff_only	Show only the changed values? defaults to FALSE.

Details

Alternatively, look at the diffdf package!

Value

A diff of the two dataframes, similar to $sheet_diff part of the return of sheet_comp. Includes a row_number column, and remaining columns have been labeled to match excel column naming conventions.

---

sheet_comp *Compare two dataframes of spreadsheets*

Description

Primary funtion for xldiff package. When cell values change between dataframe t1 and dataframe t2, the corresponding $sheet_diff entry will show [the first value] --> [the second value]. Note that because these changes are presenting as characters, changes in numbers with many digits can produce difficult-to-read cells. The proportional_threshold (or optionally absolute_threshold) can be used to determine how big a change should be to be flagged. digits_show controls how many significant digits should be used when showing the diffs. For example, proportional_threshold = 0.01 and digits_show = 5 will flag changes of at least 1%, and when those changes are present, the flagged cell will simplify the numbers to 5 significant digits when showing ## -> ##.

Usage

```
sheet_comp(
  t1,
  t2,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  digits_show = 6
)
```

Arguments

t1	First dataframe
t2	Second dataframe, same dimensions as first.
proportional_threshold	Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. proportional_threshold will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).
absolute_threshold	Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override proportional_threshold. Numeric, defaults to NULL.
digits_show	When there is a change in number values, how many digits should be shown in ## ---> ##? Numeric, defaults to 6. Recommend not making this so small that flagged changes don't get printed (e.g., if this is 2 and proportional_threshold is 0.001, 0.1% changes will get flagged, but only the first two digits will get shown).

Value

List of comparison data frames, including logical matrices used in formatting cells to highlight changes.

- $sheet_diff: cell entries for comparison
- $mat_changed logical matrix where TRUE corresponds to a cell that changed
- $mat_diff_decrease: logical matrix where TRUE corresponds to a cell of numeric values that decreased = mat_diff_increase: as above, but for increases.

Examples

```
## Not run:
## using palmerpenguins data to simulate spreadsheets
library(palmerpenguins)
t1 <- t2 <- head(penguins)
## change island variable to characters for easier modification
t2$island <- t1$island <- as.character(t1$island)
## change several entries in the second version
```

```
t2$island[3] <- "Scotland"
t2$flipper_length_mm[1] <- 18
sheet_comp(t1, t2)

## End(Not run)
```

`sheet_comp_basic` *Stripped down version of sheet_comp*

Description

Compares for perfect match, returns single matrix

Usage

```
sheet_comp_basic(t1, t2)
```

Arguments

`t1`	First dataframe
`t2`	Second dataframe, same dimensions as first.

Value

logical matrix

`wb_get_col_widths` **[Experimental]** *Get column widths of openxlsx2 Workbook sheet*

Description

Code by Jan Marvin. This function will likely be superceded by a function in openxlsx2 in the future.

Usage

```
wb_get_col_widths(wb, sheet = openxlsx2::current_sheet())
```

Arguments

`wb`	openxlsx2 workbook object
`sheet`	sheet name, defaults to current sheet

Value

Dataframe with $col (column names) and $width (column widths)

Examples

```
library(openxlsx2)
wb <- wb_workbook() |>
  wb_add_worksheet("Sheet1") |>
  wb_add_data(x = mtcars) |>
  wb_set_col_widths(cols = 2:4, widths = 12)

current_widths <- wb_get_col_widths(wb, sheet = "Sheet1")
current_widths
```

Index

add_cell_borders, 2
add_changed_formats, 3

cell_range_translate, 4
cells_stylize, 5

diff_to_table, 6

excel_diff, 7
excel_diff(), *9*, *11*
excel_diff_table, 8
excel_diff_table(), *8*, *11*
excel_diff_tibble, 10
excel_diff_tibble(), *8, 9*

formula_diff, 11

mat_to_a1, 12

present_rows_changed, 12

sheet_comp, 13
sheet_comp_basic, 15

wb_get_col_widths, 15