Package 'writexl'

<div align="center">October 4, 2024</div>

Type Package

Title Export Data Frames to Excel 'xlsx' Format

Version 1.5.1

Description Zero-dependency data frame to xlsx exporter based on 'libxlsxwriter'
<https://libxlsxwriter.github.io>. Fast and no Java or Excel required.

License BSD_2_clause + file LICENSE

Encoding UTF-8

URL <https://ropensci.r-universe.dev/writexl>
<https://docs.ropensci.org/writexl/>

BugReports <https://github.com/ropensci/writexl/issues>

RoxygenNote 7.0.2

Suggests spelling, readxl, nycflights13, testthat, bit64

Language en-US

SystemRequirements zlib

NeedsCompilation yes

Author Jeroen Ooms [aut, cre] (<https://orcid.org/0000-0002-4035-0289>),
John McNamara [cph] (Author of libxlsxwriter (see AUTHORS and COPYRIGHT
files for details))

Maintainer Jeroen Ooms <jeroenooms@gmail.com>

Repository CRAN

Date/Publication 2024-10-04 09:00:02 UTC

Contents

---

lxw_version   *Version*

Description

Shows version of bundled libxlsxwriter.

Usage

```
lxw_version()
```

---

write_xlsx   *Export to xlsx*

Description

Writes a data frame to an xlsx file. To create an xlsx with (multiple) named sheets, simply set x to
a named list of data frames.

Usage

```
write_xlsx(
  x,
  path = tempfile(fileext = ".xlsx"),
  col_names = TRUE,
  format_headers = TRUE,
  use_zip64 = FALSE
)
```

Arguments

x	data frame or named list of data frames that will be sheets in the xlsx
path	a file name to write to
col_names	write column names at the top of the file?
format_headers	make the col_names in the xlsx centered and bold
use_zip64	use zip64 to enable support for 4GB+ xlsx files. Not all platforms can read this.

Details

Currently supports strings, numbers, booleans and dates. Formatting options may be added in future
versions.

Examples

```
# Roundtrip example with single excel sheet named 'mysheet'
tmp <- write_xlsx(list(mysheet = iris))
readxl::read_xlsx(tmp)
```

---

xl_formula   *Excel Types*

Description

Create special column types to write to a spreadsheet

Usage

```
xl_formula(x)

xl_hyperlink(url, name = NULL)
```

Arguments

x	character vector to be interpreted as formula
url	character vector of URLs
name	character vector of friendly names

Examples

```
df <- data.frame(
  name = c("UCLA", "Berkeley", "Jeroen"),
  founded = c(1919, 1868, 2030),
 website = xl_hyperlink(c("http://www.ucla.edu", "http://www.berkeley.edu", NA), "homepage")
)
df$age <- xl_formula('=(YEAR(TODAY()) - INDIRECT("B" & ROW()))')
write_xlsx(df, 'universities.xlsx')

# cleanup
unlink('universities.xlsx')
```

Index