Package: rpxl (via r-universe)

October 24, 2024

Type Package

Title Read Password-Protected Excel Files

Version 0.1.0.9000

Description Uses the Python library msoffcrypto-tool to decrypt password-protected Excel files, which can then be read into R using readxl, readxlsb, or other packages.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports reticulate, readxl, readxlsb

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/epicentre-msf/rpxl

BugReports https://github.com/epicentre-msf/rpxl/issues

Repository https://epicentre-msf.r-universe.dev

RemoteUrl https://github.com/epicentre-msf/rpxl

RemoteRef HEAD

RemoteSha e68f4f228c4bf34f237eeaeeb4ddbf2a208702c9

Contents

decrypt_wb *Decrypt a password-protected Excel file*

Description

Uses the Python library msoffcrypto-tool to decrypt password-protected Excel files (either .xlsx, .xlsm, or .xlsb), which can then be read in with e.g. the readxl or readxlsb R packages.

Usage

```
decrypt_wb(path, password, path_out = tempfile(fileext = get_ext(path)))
```

Arguments

path	Path to an Excel workbook file (either .xlsx, .xlsm, or .xlsb)
password	Workbook password
path_out	Output path for decrypted workbook

Value

Path to the decrypted workbook file (same as argument path_out)

Examples

```
path_xlsx <- system.file("extdata", "xltest.xlsx", package = "rpxl")
decrypt_wb(path_xlsx, password = "1234")
```

install_rpxl *Install Python dependencies for the rpxl package*

Description

Install Python dependencies for the rpxl package

Usage

```
install_rpxl(
  envname = "r-rpxl",
  new_env = identical(envname, "r-rpxl"),
  method = "auto",
  ...
)
```

Arguments

envname	Name of Python environment to install within. Defaults to "r-rpxl".
new_env	Logical indicating whether to remove an existing virtual environment, if it has the name as envname, before creating the new virtual environment and installing the required Python packages.
method	Installation method. Defaults to "auto" to automatically find a method that will work in the local environment.
...	Additional arguments passed to reticulate::py_install

---

rp_xlsb *Read a protected .xlsb file*

Description

A wrapper to readxlsb::read_xlsb, with an initial call to decrypt_wb to decrypt the password-protected workbook

Usage

```
rp_xlsb(
  path,
  password,
  sheet = NULL,
  range = NULL,
  col_names = TRUE,
  col_types = NULL,
  na = "",
  trim_ws = TRUE,
  skip = 0,
  ...
)
```

Arguments

path	Path to the xlsb workbook
password	Workbook password
sheet	Name or index of the sheet to read. If the sheet name is specified as part of the range, this parameter is ignored
range	A named range or a string representing an excel range (of the form Sheet!A1:D10) or a cellranger object
col_names	TRUE uses the first row as the column name, FALSE sets names to column.#, or a character vector
col_types	NULL to imply type from spreadsheet or one of ignore/logical/numeric/date/string per column

na	Single string or array of strings to interpret as missing
trim_ws	Trim whitespace from strings
skip	Number of rows to skip before reading data
...	Additional options. Pass debug = TRUE to return xlsb environment

Value

A data frame

Examples

```
path_xlsb <- system.file("extdata", "xltest.xlsb", package = "rpxl")
rp_xlsb(path_xlsb, password = "1234", sheet = 1)
```

---

rp_xlsx *Read a protect .xlsx or .xlsm file*

Description

A wrapper to readxl::read_excel with an initial call to decrypt_wb to decrypt the password-protected workbook

Usage

```
rp_xlsx(
  path,
  password,
  sheet = NULL,
  range = NULL,
  col_names = TRUE,
  col_types = NULL,
  na = "",
  trim_ws = TRUE,
  skip = 0,
  n_max = Inf,
  guess_max = min(1000, n_max),
  progress = readxl_progress(),
  .name_repair = "unique"
)
```

Arguments

path	Path to the xls/xlsx file.
password	Workbook password

sheet	Sheet to read. Either a string (the name of a sheet), or an integer (the position of the sheet). Ignored if the sheet is specified via range. If neither argument specifies the sheet, defaults to the first sheet.
range	A cell range to read from, as described in cell-specification. Includes typical Excel ranges like "B3:D87", possibly including the sheet name like "Budget!B2:G14", and more. Interpreted strictly, even if the range forces the inclusion of leading or trailing empty rows or columns. Takes precedence over skip, n_max and sheet.
col_names	TRUE to use the first row as column names, FALSE to get default names, or a character vector giving a name for each column. If user provides col_types as a vector, col_names can have one entry per column, i.e. have the same length as col_types, or one entry per unskipped column.
col_types	Either NULL to guess all from the spreadsheet or a character vector containing one entry per column from these options: "skip", "guess", "logical", "numeric", "date", "text" or "list". If exactly one col_type is specified, it will be recycled. The content of a cell in a skipped column is never read and that column will not appear in the data frame output. A list cell loads a column as a list of length 1 vectors, which are typed using the type guessing logic from col_types = NULL, but on a cell-by-cell basis.
na	Character vector of strings to interpret as missing values. By default, readxl treats blank cells as missing data.
trim_ws	Should leading and trailing whitespace be trimmed?
skip	Minimum number of rows to skip before reading anything, be it column names or data. Leading empty rows are automatically skipped, so this is a lower bound. Ignored if range is given.
n_max	Maximum number of data rows to read. Trailing empty rows are automatically skipped, so this is an upper bound on the number of rows in the returned tibble. Ignored if range is given.
guess_max	Maximum number of data rows to use for guessing column types.
progress	Display a progress spinner? By default, the spinner appears only in an interactive session, outside the context of knitting a document, and when the call is likely to run for several seconds or more. See readxl_progress() for more details.
.name_repair	Handling of column names. Passed along to tibble::as_tibble(). readxl's default is '.name_repair = "unique", which ensures column names are not empty and are unique.

Value

A tibble-style data frame

Examples

```
path_xlsx <- system.file("extdata", "xltest.xlsx", package = "rpxl")
rp_xlsx(path_xlsx, password = "1234")
```

Index