

Package ‘ezpickr’

November 17, 2019

Title Easy Data Import Using GUI File Picker and Seamless Communication Between an Excel and R

Version 2.0.0

Description Choosing any rectangular data file using interactive GUI dialog box, and seamlessly manipulating tidy data between an 'Excel' window and R session.

License GPL-3

VignetteBuilder knitr

URL <https://github.com/jooyoungseo/ezpickr>

BugReports <https://github.com/jooyoungseo/ezpickr/issues>

Encoding UTF-8

LazyData true

Imports haven, jsonlite, readxl, magrittr, textreadr, tibble, stringr, writexl, purrr, mboxr, vroom, rmarkdown, utils

RoxygenNote 7.0.0.9000

Suggests testthat, knitr, striptrf, xml2, dplyr, rvest, antiword, covr

NeedsCompilation no

Author JooYoung Seo [aut, cre] (<<https://orcid.org/0000-0002-4064-6012>>),
Soyoung Choi [aut] (<<https://orcid.org/0000-0002-0998-3352>>),
hyun seung Lee [ctb],
Jonghoen Kim [ctb]

Maintainer JooYoung Seo <jooyoung@psu.edu>

Repository CRAN

Date/Publication 2019-11-17 05:20:03 UTC

R topics documented:

pick	2
picko	3
viewxl	5
Index	6

 pick

Choosing any rectangular data file using interactive GUI dialog box.

Description

You can alternatively use this function for choosing *.csv, *.csv2, *.tsv, *.txt, *.xls, *.xlsx, *.json, *.html, *.htm, *.php, *.pdf, *.doc, *.docx, *.rtf, *.RData, *.Rda, *.RDS, *.sav (SPSS), *.por, *.sas7bdat, *.sas7bcat, *.dta, *.xpt, *.mbox, and *.Rmd files in an interactive GUI mode. A file choose dialog box will be prompted.

Usage

```
pick(file = NULL, mode = NULL, ...)
```

Arguments

file	Either a path to a file, a connection, or literal data (either a single string or a raw vector). The default is NULL, which pops up an interactive GUI file choose dialogue box for users unless an explicit path/to/filename is given. Each corresponding function depending upon a file extension will be automatically matched and applied once you pick up your file using either the GUI-file-chooser dialog box or explicit path/to/filename.
mode	Character value for session locale and encoding; available values are: "ko1" for "CP949"; "ko2" for "UTF-8" while both change R locale into Korean (default is the current locale and encoding of your R session).
...	Any additional arguments available for each file type and extension: vroom for 'CSV' (Comma-Separated Values); 'CSV2' (Semicolon-Separated Values); 'TSV' (Tab-Separated Values); 'txt' (plain text) files; read_excel for 'Excel' files; read_spss for 'SPSS' files; read_stata for 'Stata' files; read_sas for 'SAS' files; read_document for 'Microsoft Word', 'PDF', 'RTF', 'HTML', 'HTM', and 'PHP' files; fromJSON for 'JSON' files; read_mbox for 'mbox' files; render for 'Rmd' files; source for 'R' files; readRDS for 'RDS' files; load for 'RDA' and 'RDATA' files.

Details

pick
See example below.

Value

tibble (table data.frame) object of the chosen rectangular data file will be returned.

Author(s)

JooYoung Seo, <jooyoung@psu.edu>
Soyoung Choi, <sxc940@psu.edu>

See Also

[picko](#) for Korean users.

Examples

```
# Choosing file and saving it into a variable
## Scenario 1: Picking up a file using interactive GUI dialog box:
if(interactive()) {
  library(ezpickr)
  ## Use either `pick(mode="ko1")` or `pick(mode="ko2")` for Korean R users.
  data <- pick()
}

## Scenario 2: Picking up a file using an explicit file name ("test.sav" in the example below;
## however, you can feed other files through this function
## such as *.SAS, *.DTA, *.csv, *.csv2, *.tsv, *.xlsx, *.txt,
## *.html, webpage URL, *.doc, *.docx, *.pdf, *.rtf, *.json, *.Rda, *.Rdata, and more):
library(ezpickr)
test <- system.file("extdata", "airquality.sav", package = "ezpickr")
## Use either `pick(test, mode="ko1")` or `pick(test, mode="ko2")` for Korean R users.
data <- pick(test)

# Now you can use the imported file as a tibble.
str(data)
```

picko	<i>Choosing any rectangular Korean data file using interactive GUI dialog box.</i>
-------	--

Description

The basic functionality is exactly the same as [pick](#), but optimized for Korean R users.

Usage

```
picko(file = NULL, ...)
```

Arguments

file	Either a path to a file, a connection, or literal data (either a single string or a raw vector). The default is NULL, which pops up an interactive GUI file choose dialogue box for users unless an explicit path/to/filename is given. Each corresponding function depending upon a file extension will be automatically matched and applied once you pick up your file using either the GUI-file-chooser dialog box or explicit path/to/filename.
...	Any additional arguments available for each file type and extension: vroom for 'CSV' (Comma-Separated Values); 'CSV2' (Semicolon-Separated Values); 'TSV' (Tab-Separated Values); 'txt' (plain text) files; read_excel for 'Excel'

files; [read_spss](#) for 'SPSS' files; [read_stata](#) for 'Stata' files; [read_sas](#) for 'SAS' files; [read_document](#) for 'Microsoft Word', 'PDF', 'RTF', 'HTML', 'HTM', and 'PHP' files; [fromJSON](#) for 'JSON' files; [read_mbox](#) for 'mbox' files; [render](#) for 'Rmd' files; [source](#) for 'R' files; [readRDS](#) for 'RDS' files; [load](#) for 'RDA' and 'RDATA' files.

Details

picko

See example below.

Value

tibble (table data.frame) object of the chosen rectangular data file will be returned.

Author(s)

JooYoung Seo, <jooyoung@psu.edu>

Soyoung Choi, <sxc940@psu.edu>

See Also

[pick](#) for more details on basic functionality.

Examples

```
# Choosing file and saving it into a variable
## Scenario 1: Picking up a file using interactive GUI dialog box:
if(interactive()) {
  library(ezpickr)
  data <- picko()
}

## Scenario 2: Picking up a file using an explicit file name ("test.sav" in the example below;
## however, you can feed other files through this function
## such as *.SAS, *.DTA, *.csv, *.csv2, *.tsv, *.xlsx, *.txt,
## *.html, webpage URL, *.doc, *.docx, *.pdf, *.rtf, *.json, *.Rda, *.Rdata, and more):
if(interactive()) {
  library(ezpickr)
  test <- system.file("extdata", "airquality.sav", package = "ezpickr")
  data <- picko(test)

# Now you can use the imported file as a tibble.
str(data)
}
```

viewxl	<i>Seamlessly manipulate any rectangular data file between an Excel window and R session.</i>
--------	---

Description

You can use this function for loading and manipulating any `data.frame`, `data_frame`, `tbl_df`, `matrix`, `table`, `vector`, or `DocumentTermMatrix` objects into your system-default spreadsheet software (e.g., Excel) in a real time. This function has employed `write_xlsx` under the hood.

Usage

```
viewxl(x, ...)
```

Arguments

<code>x</code>	An object of class <code>data.frame</code> , <code>matrix</code> , <code>table</code> , <code>vector</code> , or <code>DocumentTermMatrix</code> .
<code>...</code>	Any additional arguments available for <code>write_xlsx</code> .

Details

viewxl

See example below.

Value

Data object opened in a preferable spreadsheet application window which will in turn be called on your R session again.

Author(s)

JooYoung Seo, <jooyoung@psu.edu>

Soyoung Choi, <sxc940@psu.edu>

Examples

```
if(interactive()) {
  library(ezpickr)
  data(airquality)
  str(airquality)
  ## View your data object in your spreadsheet software:
  viewxl(airquality)
  # Then, when necessary, you can modify the opened data in the spreadsheet and save it as a new data.

  # You can pass a list object to the `view()` function like below:
  l <- list(iris = iris, mtcars = mtcars, chickwts = chickwts, quakes = quakes)
  viewxl(l)
  # Then, each list item will appear in your Excel window sheet by sheet.
}
```

Index

*Topic **picko**

picko, 3

*Topic **pick**

pick, 2

*Topic **viewxl**

viewxl, 5

fromJSON, 2, 4

load, 2, 4

pick, 2, 3, 4

picko, 3, 3

read_document, 2, 4

read_excel, 2, 3

read_mbox, 2, 4

read_sas, 2, 4

read_spss, 2, 4

read_stata, 2, 4

readRDS, 2, 4

render, 2, 4

source, 2, 4

viewxl, 5

vroom, 2, 3

write_xlsx, 5