Package 'csvy'

May 8, 2026

Type Package

Title Import and Export CSV Data with a YAML Metadata Header

Version 0.3.0

Date 2018-07-31

Description
Support for import from and export to the CSVY file format. CSVY is a file format that combines the simplicity of CSV (comma-separated values) with the metadata of other plain text and binary formats (JSON, XML, Stata, etc.) by placing a YAML header on top of a regular CSV.

URL https://github.com/leeper/csvy

BugReports https://github.com/leeper/csvy/issues

Imports tools, data.table, jsonlite, yaml

Suggests testthat, datasets

License GPL-2

RoxygenNote 6.0.1

NeedsCompilation no

Author Thomas J. Leeper [aut, cre] (ORCID:
<https://orcid.org/0000-0003-4097-6326>),
Alexey N. Shiklomanov [aut] (ORCID:
<https://orcid.org/0000-0003-4022-5979>),
Jonathan Carroll [aut] (ORCID: <https://orcid.org/0000-0002-1404-5264>)

Maintainer Thomas J. Leeper <thosjleeper@gmail.com>

Repository CRAN

Date/Publication 2018-08-01 04:40:02 UTC

Contents

colclass_dict 2
csvy 2
get_yaml_header 2
read_csvy 3

read_metadata 4
write_csvy 4
write_metadata 5

Index **6**

`colclass_dict` *Dictionary of column classes for reading data*

Description

Dictionary of column classes for reading data

Usage

```
colclass_dict
```

Format

An object of class `character` of length 7.

`csvy` *Import and Export CSV Data With a YAML Metadata Header*

Description

CSVY is a file format that combines the simplicity of CSV (comma-separated values) with the metadata of other plain text and binary formats (JSON, XML, Stata, etc.). The CSVY file specification is simple: place a YAML header on top of a regular CSV. The csvy package implements this format using two functions: `write_csvy` and `read_csvy`.

`get_yaml_header` *Retrieve YAML header from file*

Description

Note that this assumes only one Yaml header, starting on the first line of the file.

Usage

```
get_yaml_header(file, yaml_rxp = "^\\#*---[[:space:]]*$", verbose = TRUE)
```

Arguments

file	A character string or R connection specifying a file.
yaml_rxp	Regular expression for parsing YAML header
verbose	Logical. If TRUE, print warning if no header found.

Value

Character vector of lines containing YAML header, or 'NULL' if no YAML header found.

---

read_csvy *Import CSVY data*

Description

Import CSVY data as a data.frame

Usage

```
read_csvy(file, metadata = NULL, stringsAsFactors = FALSE,
  detect_metadata = TRUE, ...)
```

Arguments

file	A character string or R connection specifying a file.
metadata	Optionally, a character string specifying a YAML (".yaml") or JSON (".json") file containing metadata (in lieu of including it in the header of the file).
stringsAsFactors	A logical specifying whether to treat character columns as factors. Passed to read.csv or fread depending on the value of method. Ignored for method = 'readr' which never returns factors.
detect_metadata	A logical specifying whether to auto-detect a metadata file if none is specified (and if no header is found).
...	Additional arguments passed to fread.

See Also

write_csvy

Examples

```
read_csvy(system.file("examples", "example1.csvy", package = "csvy"))
```

read_metadata *Read metadata*

Description

Read csvy metadata from an external `.yml/.yaml` or `.json` file

Usage

```
read_metadata(file)
```

Arguments

`file`	full path of file from which to read the metadata.

Value

the metadata as a list

write_csvy *Export CSVY data*

Description

Export data.frame to CSVY

Usage

```
write_csvy(x, file, metadata = NULL, sep = ",", dec = ".",
  comment_header = if (is.null(metadata)) TRUE else FALSE,
  name = deparse(substitute(x)), metadata_only = FALSE, ...)
```

Arguments

`x`	A data.frame.
`file`	A character string or R connection specifying a file.
`metadata`	Optionally, a character string specifying a YAML (".yaml") or JSON (".json") file to write the metadata (in lieu of including it in the header of the file).
`sep`	A character string specifying a between-field separator. Passed to `fwrite`.
`dec`	A character string specifying a within-field separator. Passed to `fwrite`.
`comment_header`	A logical indicating whether to comment the lines containing the YAML front matter. Default is `TRUE`.
`name`	A character string specifying a name for the dataset.
`metadata_only`	A logical indicating whether only the metadata should be produced (no CSV component).
`...`	Additional arguments passed to `fwrite`.

See Also

write_csvy

Examples

```
library("datasets")
write_csvy(head(iris))

# write yaml w/o comment charaters
write_csvy(head(iris), comment_header = FALSE)
```

`write_metadata` *Write csvy metadata*

Description

Write csvy metadata to an external `.yml`/`.yaml` or `.json` file

Usage

```
write_metadata(metadata_list = NULL, file = NULL)
```

Arguments

`metadata_list`	metadata to be stored. Must be valid as per `yaml::as.yaml()` or `jsonlite::write_json()` for that particular output type.
`file`	full path of file in which to save the metadata.

Value

`NULL` (invisibly)

Index

∗ **datasets**
colclass_dict, 2

colclass_dict, 2
csvy, 2
csvy-package *(*csvy*)*, 2

fread, *3*
fwrite, *4*

get_yaml_header, 2

jsonlite::write_json(), *5*

read.csv, *3*
read_csvy, *2*, 3
read_metadata, 4

write_csvy, *2, 3*, 4, *5*
write_metadata, 5

yaml::as.yaml(), *5*