Package: osutils (via r-universe)

September 14, 2024

Title Useful Functions for OpenSAFELY

Version 0.0.0.9000

Description Contains functions that are often needed when using the OpenSAFELY platform <https://www.opensafely.org/>, such as redaction and low-memory processing.

License MIT + file LICENSE

URL https://github.com/wjchulme/osutils

BugReports https://github.com/wjchulme/osutils/issues/

Imports broom, data.table, dplyr, forcats, fs, gt, jsonlite, lifecycle, magrittr, plyr, purrr, readr, rlang, stats, stringr, tibble, tidyr, tidyselect, yaml

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

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

RemoteUrl https://github.com/wjchulme/osutils

RemoteRef HEAD

RemoteSha ed6804e952d4a8ef783b1897966cdca512f69505

Contents

`action_names_to_txt` *Put action names in a txt file —-*

Description

Put action names in a txt file —-

Usage

```
action_names_to_txt(action_list, filepath = NULL)
```

Arguments

`action_list`	list of project actions
`filepath`	file path and name where .txt file should be saved. If not provided, then prints to console!

Details

grab all action names and send to a txt file. "action_list" should be the "actions" list entry in the "project_list" object (i.e., `project_list$actions`)

`c_action` *Create comment object*

Description

Create comment object

Usage

```
c_action(...)
```

Arguments

`...`	a collection of actions and lists of actions.

Details

Use this to combine action objects before passing to `project_list()`. This ensures that the list of actions has the correct structure. Do not use `list(...)` or similar!

Value

A list of actions.

`gt_cat` *Convert output of categorical tabulation (redacted_summary_cat) to gt object*

Description

Convert output of categorical tabulation (redacted_summary_cat) to gt object

Usage

```
gt_cat(x, var_name = "", pct_decimals = 1)
```

Arguments

`x`	The data.frame produced by redacted_summary_cat.
`var_name`	The variable name.
`pct_decimals`	Decimal precision for percentages.

Details

This function takes the output of redacted_summary_cat and converts it to a gt object (as from the gt package) for outputting to html/pdf.

Value

A gt object.

`gt_catcat` *Convert output of categorical cross-tabulation (redacted_summary_catcat) to gt object*

Description

Convert output of categorical cross-tabulation (redacted_summary_catcat) to gt object

Usage

```
gt_catcat(
  x,
  var1_name = "",
  var2_name = "",
  title = NULL,
  source_note = NULL,
  pct_decimals = 1
)
```

Arguments

`x`	The data.frame produced by redacted_summary_catcat.
`var1_name`	The name of the first categorical variable.
`var2_name`	The name of the second categorical variable.
`title`	The title of the table.
`source_note`	A footnote.
`pct_decimals`	Decimal precision for percentages.

Details

This function takes the output of redacted_summary_catcat and converts it to a gt object (as from the gt package) for outputting to html/pdf.

Value

A gt object.

gt_catnum — *Convert output of categorical-numeric cross-tabulation (redacted_summary_catnum) to gt object*

Description

Convert output of categorical-numeric cross-tabulation (redacted_summary_catnum) to gt object

Usage

```
gt_catnum(x, cat_name = "", num_name = "", num_decimals = 1, pct_decimals = 1)
```

Arguments

`x`	The data.frame produced by redacted_summary_catnum.
`cat_name`	The categorical variable name.
`num_name`	The numeric variable name.
`num_decimals`	Decimal precision for numbers.
`pct_decimals`	Decimal precision for percentages.

Details

This function takes the output of redacted_summary_catnum and converts it to a gt object (as from the gt package) for outputting to html/pdf.

Value

A gt object.

gt_num — *Convert output of numeric tabulation (redact_summary_num) to gt object*

Description

Convert output of numeric tabulation (redact_summary_num) to gt object

Usage

```
gt_num(x, var_name = "", num_decimals = 1, pct_decimals = 1)
```

Arguments

`x` The data.frame produced by `redact_summary_num`

`var_name` The variable name

`num_decimals` Decimal precision for numbers

`pct_decimals` Decimal precision for percentages

Details

This function takes the output of `redact_summary_num` and converts it to a gt object (as from the `gt` package) for outputting to html/pdf.

Value

A gt object

`pipeline_action` *Create action object*

Description

Create action object

Usage

```
pipeline_action(
  name,
  run,
  arguments = NULL,
  needs = NULL,
  highly_sensitive = NULL,
  moderately_sensitive = NULL,
  ...
)
```

Arguments

`name` The name of the action. Must be a 1-d character

`run` The run command. Must be a 1-d character

`arguments` A character vector of arguments to be appended to the run command. Note that all arguments are parsed as strings / characters, so should be converted in-script if needed

`needs` A character vector of names of action dependencies

`highly_sensitive` A named character vector (or named list) of highly sensitive outputs from the action

`moderately_sensitive`
: A named character vector (or named list) of moderately sensitive outputs from the action

`...`
: other possible key:value pairs for action types with special parameters

Details

A named list of length one containing all information needed to define the action and turn it into a yaml chunk. This function can be used a a one-off to create single actions, or used to generate functions that create more specific actions with repeated patterns. All action objects created by this function should be then put together using the `pipeline_list()` function, for instance `pipeline_list(action(...), action(...), action(...), ...)`. If combining 2 or more actions before passing to `pipeline_list()`, use the helper function `c_action()` (similar to `purrr::splice(...)` or `purrr::list_flatten(list(...))`). This ensures that the list of actions has the correct structure. Do not use `list(...)` or similar!

Value

list

`pipeline_comment` *Create comment object*

Description

Create comment object

Usage

```
pipeline_comment(...)
```

Arguments

`...`
: character or -character-convertible objects

Details

key:value list element that will be converted to a comment block in yaml when `project_list_to_yaml()` is run. Each comment will be prefixed by "## " and suffixed by " ##". These comments are first converted to `'': '## your comment here ##'` in yaml, and then tidied up to `## your comment here ##` before saving.

Value

A list

pipeline_list *Create entire pipeline list*

Description

Create entire pipeline list

Usage

```
pipeline_list(..., .version = "3.0", .population_size = 1000L)
```

Arguments

`...` all actions and comments that go into the entire project pipeline. These can be provided as a mixture of single actions (from `pipeline_action()` function) or as lists of actions (from `c_action()` function.)

`.version` version of opensafely to use

`.population_size` size of dummy data expectations

Details

This function is used to put all actions together in the entire project list, as well as specifying the project frontmatter (version and expectations).

Value

A list

project_list_to_yaml *Convert list to yaml and save*

Description

Convert list to yaml and save

Usage

```
project_list_to_yaml(project_list, filepath = NULL)
```

Arguments

`project_list` list object containing all actions (created using action function) and comment-actions (created using comment_action function) and front-matter.

`filepath` file path and name where yaml file should be saved. If not provided, then prints to console!

Details

Convert list to yaml string and then prints or saves the results. This also does some reformatting of comment blocks, whitespace, etc.

`readtype_csv` *Read a csv file into a tibble, and type columns using a separate json file.*

Description

Read a csv file into a tibble, and type columns using a separate json file.

Usage

```
readtype_csv(
  file,
  suffix = "",
  delim,
  quote = "\"",
  escape_backslash = FALSE,
  escape_double = TRUE,
  locale = default_locale(),
  na = c("", "NA"),
  quoted_na = TRUE,
  comment = "",
  trim_ws = FALSE
)
```

Arguments

`file` Delimited file location.

`suffix` The suffix used in the name of the json file, which is appended to the delimited file name. Defaults to `""` (no suffix), so that the file name is the same as the delimited file name (excluding filetype extensions).

`delim` Single character used to separate fields within a record.

`quote` Single character used to quote strings.

`escape_backslash`
Does the file use backslashes to escape special characters? This is more general than `escape_double` as backslashes can be used to escape the delimiter character, the quote character, or to add special characters like `\\n`.

`escape_double` Does the file escape quotes by doubling them? i.e. If this option is `TRUE`, the value `""""` represents a single quote, `\"`.

`locale` The locale controls defaults that vary from place to place. The default locale is US-centric (like R), but you can use `locale()` to create your own locale that controls things like the default time zone, encoding, decimal mark, big mark, and day/month names.

na	Character vector of strings to interpret as missing values. Set this option to `character()` to indicate no missing values.
quoted_na	**[Deprecated]** Should missing values inside quotes be treated as missing values (the default) or strings. This parameter is soft deprecated as of readr 2.0.0.
comment	A string used to identify comments. Any text after the comment characters will be silently ignored.
trim_ws	Should leading and trailing whitespace (ASCII spaces and tabs) be trimmed from each field before parsing it?

Details

Based on the readr::read_csv function. Requires csv files to be saved using writetype_csv, which will also create the json file containing the typing info. Datetime and time classes are not supported.

Value

A `tibble()`.

readtype_delim *Read a delimited file (including CSV and TSV) into a tibble, and type columns using a separate json file*

Description

Read a delimited file (including CSV and TSV) into a tibble, and type columns using a separate json file

Usage

```
readtype_delim(
  file,
  suffix = "",
  delim,
  quote = "\"",
  escape_backslash = FALSE,
  escape_double = TRUE,
  locale = default_locale(),
  na = c("", "NA"),
  quoted_na = TRUE,
  comment = "",
  trim_ws = FALSE
)
```

Arguments

`file` Delimited file location.

`suffix` The suffix used in the name of the json file, which is appended to the delimited file name. Defaults to `""` (no suffix), so that the file name is the same as the delimited file name (excluding filetype extensions).

`delim` Single character used to separate fields within a record.

`quote` Single character used to quote strings.

`escape_backslash`
Does the file use backslashes to escape special characters? This is more general than `escape_double` as backslashes can be used to escape the delimiter character, the quote character, or to add special characters like `\\n`.

`escape_double` Does the file escape quotes by doubling them? i.e. If this option is `TRUE`, the value `""""` represents a single quote, `\"`.

`locale` The locale controls defaults that vary from place to place. The default locale is US-centric (like R), but you can use `locale()` to create your own locale that controls things like the default time zone, encoding, decimal mark, big mark, and day/month names.

`na` Character vector of strings to interpret as missing values. Set this option to `character()` to indicate no missing values.

`quoted_na` **[Deprecated]** Should missing values inside quotes be treated as missing values (the default) or strings. This parameter is soft deprecated as of readr 2.0.0.

`comment` A string used to identify comments. Any text after the comment characters will be silently ignored.

`trim_ws` Should leading and trailing whitespace (ASCII spaces and tabs) be trimmed from each field before parsing it?

Details

Based on the readr::read_delim function. Requires delimited files to be saved using writetype_delim, which will also create the json file containing the typing info. Datetime and time classes are not supported.

Value

A `tibble()`.

`redacted_summary_cat` *Summarise a categorical variable and redact if necessary*

Description

Summarise a categorical variable and redact if necessary

Usage

```
redacted_summary_cat(
  x,
  threshold = 5L,
  precision = 1L,
  .missing_name = "(missing)",
  .redacted_name = "redacted"
)
```

Arguments

`x` The vector to summarise and redact.

`threshold` The redaction threshold. All values less than or equal to this threshold will be redacted (and possibly more; see the `redactor` function)

`precision` The precision of any rounding that is to be applied to frequency values. Defaults to 1 (no rounding).

`.missing_name` The string used to replace `NA` categories.

`.redacted_name` The string used to replace redacted values.

Details

This function takes a categorical vector (or something that can be coerced to a categorical vector), computes value frequencies and proportions, and redacts according to the rules in `redactor`.

Value

A table of redacted frequencies and proportions.

`redacted_summary_catcat` *Categorical by categorical cross-tabulation, with redaction if necessary*

Description

Categorical by categorical cross-tabulation, with redaction if necessary

Usage

```
redacted_summary_catcat(
  x1,
  x2,
  threshold = 5L,
  precision = 1L,
  .missing_name = "(missing)",
  .redacted_name = "redacted",
  .total_name = NULL
)
```

Arguments

x1 The first categorical variable.

x2 The second categoical variable.

threshold The redaction threshold. All values less than or equal to this threshold will be redacted (and possibly more; see the redactor function)

precision The precision of any rounding that is to be applied to frequency values. Defaults to 1 (no rounding).

.missing_name The string used to replace NA categories.

.redacted_name The string used to replace redacted values.

.total_name The string used to the label the marginal totals. If NULL, no marginal totals are reported.

Details

This function takes two categorical vectors (or vectors that can be coerced to a categorical vectors), performs a cross-tabulation, and redacts according to the rules in redactor. proportions are based on x1 totals.

Value

A table of redacted frequencies and proportions, arranged in long-format.

---

redacted_summary_catnum

Categorical by numeric cross-tabulation, with redaction if necessary

---

Description

Categorical by numeric cross-tabulation, with redaction if necessary

Usage

```
redacted_summary_catnum(
  variable_cat,
  variable_num,
  threshold = 5L,
  .missing_name = "(missing)",
  .redacted_name = "redacted"
)
```

Arguments

`variable_cat` The categorical vector (or will be coerced to one)

`variable_num` The numeric vector

`threshold` The redaction threshold. If the length of `x` is less than or equal to this threshold, then no summary values will be reported.

`.missing_name` The string used to replace `NA` categories.

`.redacted_name` The string used to replace redacted values.

Details

This function takes a categorical vector and a numeric vector of the same length, and performs a cross-tabulation. Summary statistics are redacted according to the rules in `redactor`.

Value

A table of summary statistics for the numeric variable, stratified by the categorical variable

`redacted_summary_date` *Redact a date vector*

Description

Redact a date vector

Usage

```
redacted_summary_date(x, threshold = 5L, .redacted_name = "redacted")
```

Arguments

`x` The date variable.

`threshold` The redaction threshold. If the length of `x` is less than or equal to this threshold, then no summary values will be reported.

`.redacted_name` The string used to replace redacted values.

Details

This function takes a date vector (or something that can be coerced to one), and summarises it. Summary statistics are redacted according to the rules in `redactor`.

Value

A table of summary statistics for the variable.

`redacted_summary_num` *Summarise a numeric vector and redact if necessary*

Description

Summarise a numeric vector and redact if necessary

Usage

```
redacted_summary_num(x, threshold = 5L, .redacted_name = "redacted")
```

Arguments

`x` The numeric variable.

`threshold` The redaction threshold. If the length of `x` is less than or equal to this threshold, then no summary values will be reported.

`.redacted_name` The string used to replace redacted values.

Details

This function takes a numeric vector (or something that can be coerced to one), and summarises it. Summary statistics are redacted according to the rules in `redactor`.

Value

A table of summary statistics for the variable.

`redactor` *Indicates which values to redact from a vector of frequencies*

Description

Indicates which values to redact from a vector of frequencies

Indicates which values to redact from a vector of frequencies

Usage

```
redactor(n, threshold)

redactor(n, threshold)
```

Arguments

`n` A vector of integer frequencies or counts from a 1-dimension frequency distribution.

`threshold` The redaction threshold. All values (and possibly more; see details) less than or equal to this threshold will be redacted.

Details

Given a vector of frequencies `n`, this function returns a logical vector of frequencies to be redacted. All frequencies less than or equal to the threshold are redacted. If the sum the redacted frequencies is also less than or equal to the threshold, then the smallest unredacted frequency is also redacted.

Given a vector of frequencies `n`, this function returns a logical vector of frequencies to be redacted. All frequencies less than or equal to the threshold are redacted. If the sum the redacted frequencies is also less than or equal to the threshold, then the smallest unredacted frequency is also redacted.

Value

A logical vector the same length as `n`.

A logical vector the same length as `n`.

`redactor2` *Redact values in a vector based on frequency values*

Description

Redact values in a vector based on frequency values

Usage

```
redactor2(n, threshold, x = NULL)
```

Arguments

`n` A vector of integer frequencies or counts from a 1-dimension frequency distribution.

`threshold` The redaction threshold. All values (and possibly more; see details) less than or equal to this threshold will be redacted.

`x` Values to redact. If `x` is `NULL` then `x` redacts values of `n`.

Details

If `x` is `NULL`, then this function redacts values in `n` and returns the redacted vector. If `x` is not `NULL`, values in `x` are redacted according to frequencies in `n`. Values are redacted as follows: all frequencies less than or equal to the threshold are redacted; if the sum the redacted frequencies is also less than or equal to the threshold, then the smallest unredacted frequency is also redacted.

Value

A vector the same length as `n`.

`redact_tblsummary` *Redact tbl_summary object*

Description

Redact tbl_summary object

Usage

```
redact_tblsummary(x, threshold, redact_chr = NA_character_)
```

Arguments

`x`	A tbl_summary object created by the `gt` package.
`threshold`	The redaction threshold. All values less than or equal to this threshold will be redacted.
`redact_chr`	The character string used to replace redacted values. Default is "NA".

Details

This function redacts all statistics based on counts less than the threshold (including means, medians, etc) it also removes potentially disclosive items from the object, namely:

- `x$inputs$data` which contains the input data
- `x$inputs$meta_data` which contains the raw summary table for the table

Value

A redacted tbl_summary object

`reformat_codelists` *Converts a json file of codelist names and URLs into an HTML table*

Description

Converts a json file of codelist names and URLs into an HTML table

Usage

```
reformat_codelists(import_json_from = "./codelists/codelists.json", export_to)
```

Arguments

`import_json_from`
A character containing the path of the json file containing the codelists. defaults to `./codelists/codelists.json` which is the OpenSAFELY standard

`export_to` The path to which the file should be saved

Details

This function currently only exports an HTML file but it can be adapted to output text, markdown, etc. Ideally this would be an in-built OpenSAFELY feature rather than written externally in R.

`round_km` *Rounded Kaplan-Meier curves*

Description

Rounded Kaplan-Meier curves

Usage

```
round_km(data, time, event, strata = NULL, threshold = 6)
```

Arguments

`data` A data frame containing the required survival times

`time` Event/censoring time variable, supplied as a character. Must be numeric >0

`event` Event indicator variables supplied as a character. Censored (`0/FALSE`) or not (`1/TRUE`). Must be logical or integer with values zero or one

`strata` names of stratification / grouping variables, supplied as a character vector of variable names

`threshold` Redact threshold to apply

Details

This function rounds Kaplan-Meier survival estimates by delaying events times until at least `threshold` events have occurred.

Value

A tibble with rounded numbers of at risk, events, censored, and derived survival estimates, by strata

`sample_nonoutcomes_n` *Sample patients (or other observational units) based on patient IDs, depending on occurrence of an event or not*

Description

Sample patients (or other observational units) based on patient IDs, depending on occurrence of an event or not

Usage

```
sample_nonoutcomes_n(had_outcome, id, n)
```

Arguments

`had_outcome` A logical indicating if the patient has experienced the outcome or not

`id` An integer patient identifier with the following properties:

- consistent between cohort extracts
- unique
- completely randomly assigned (no correlation with practice ID, age, registration date, etc etc) which should be true as it based on hash of true IDs
- strictly greater than zero

`n` The number of patients (amongst all those who did not experience the event) to be sampled

Details

If `had_outcome` is `TRUE` then result is always `TRUE`. If `had_outcome` is `FALSE`, then result is `TRUE` with probability `max(1,n/sum(1-had_outcome))` and `FALSE` with probability `min(0, 1 - n/sum(1-had_outcome)).` Patients are selected in ascending order of patient ID until the sampling number is met. Warns (does not fail) if `n` is greater than `sum(1-had_outcome)`.

Value

A logical vector indicating whether the patient has been sampled or not

`sample_nonoutcomes_prop` *Sample patients (or other observational units) based on patient IDs, depending on occurrence of an event or not*

Description

Sample patients (or other observational units) based on patient IDs, depending on occurrence of an event or not

Usage

```
sample_nonoutcomes_prop(had_outcome, id, proportion)
```

Arguments

`had_outcome` A logical indicating if the patient has experienced the outcome or not

`id` An integer patient identifier with the following properties:

- consistent between cohort extracts
- unique
- completely randomly assigned (no correlation with practice ID, age, registration date, etc etc) which should be true as it based on hash of true IDs
- strictly greater than zero

`proportion` The proportion of patients (amongst all those who did not experience the event) to be sampled

Details

If `had_outcome` is `TRUE` then result is always `TRUE`. If `had_outcome` is `FALSE`, then result is `TRUE` with probability `proportion` and `FALSE` with probability `1 - proportion`. Patients are selected in ascending order of patient ID until the sampling proportion is met.

Value

A logical vector indicating whether the patient has been sampled or not

sample_random_n *Sample n patients (or other observational units) based on patient IDs.*

Description

Sample n patients (or other observational units) based on patient IDs.

Usage

```
sample_random_n(id, n)
```

Arguments

`id`	An integer patient identifier with the following properties: • consistent between cohort extracts • unique • completely randomly assigned (no correlation with practice ID, age, registration date, etc etc) which should be true as it based on hash of true IDs • strictly greater than zero
`n`	The number of patients (amongst all those who did not experience the event) to be sampled

Details

Result is `TRUE` with probability `max(1,n/length(id))` and `FALSE` with probability `min(0, 1 - n/length(id))`. Patients are selected in ascending order of patient ID until the sampling number is met. Warns (does not fail) if `n` is greater than `length(id)`.

Value

A logical vector indicating whether the patient has been sampled or not

sample_random_prop *Sample a proportion of patients (or other observational units) based on patient IDs*

Description

Sample a proportion of patients (or other observational units) based on patient IDs

Usage

```
sample_random_prop(id, proportion)
```

Arguments

`id` An integer patient identifier with the following properties:

- consistent between cohort extracts
- unique
- completely randomly assigned (no correlation with practice ID, age, registration date, etc etc) which should be true as it based on hash of true IDs
- strictly greater than zero

`proportion` The proportion of patients (amongst all those who did not experience the event) to be sampled

Details

Result is `TRUE` with probability `p` and `FALSE` with probability `1-p`. `p` is equal to `ceiling(length(id)*proportion)/length(id)`, which is equal to `proportion` when `length(id)*proportion` is an integer, and slightly higher otherwise. Patients are selected in ascending order of patient ID until the sampling proportion is met.

Value

A logical vector indicating whether the patient has been sampled or not

`sample_weights` *Derive sampling probabilities*

Description

Derive sampling probabilities

Usage

```
sample_weights(had_outcome, sampled)
```

Arguments

`had_outcome` A logical indicating if the patient has experienced the outcome or not

`sampled` A logical indicating if a patient was sampled or not

Value

A numeric vector of the sampling probability

writetype_csv — *Write a data frame to a csv file, and save typing information in a separate json file*

Description

Write a data frame to a csv file, and save typing information in a separate json file

Usage

```
writetype_csv(
  x,
  path,
  suffix = "",
  na = "NA",
  quote_escape = "double",
  eol = "\n"
)
```

Arguments

`x`	A data frame or tibble to write to disk.
`path`	File or connection to write to. (path is now deprecated in readr v1.4 for OpenSAFELY currently has older version, so use path for now).
`suffix`	The suffix used in the name of the json file, to be appended to the delimited file name. Defaults to `""` (no suffix), so that the file name is the same as the delimited file name (excluding filetype extensions).
`na`	String used for missing values. Defaults to `"NA"`. Missing values will never be quoted; strings with the same value as `na` will always be quoted.
`quote_escape`	The type of escaping to use for quoted values, one of "`double`", "`backslash`" or "`none`". You can also use `FALSE`, which is equivalent to "`none`". The default is "`double`", which is expected format for Excel.
`eol`	The end of line character to use. Most commonly either "`\n`" for Unix style newlines, or "`\r\n`" for Windows style newlines.

Details

Based on the readr::write_delim function. Additionally, this function saves a json file containing typing info for the data frame, which can be used to re-type the data when re-imported into R. Datetime and time classes are not supported.

Value

Returns the input invisibly.

writetype_delim *Write a data frame to a delimited file, and save typing information in a separate json file*

Description

Write a data frame to a delimited file, and save typing information in a separate json file

Usage

```
writetype_delim(
  x,
  path,
  suffix = "",
  delim = " ",
  na = "NA",
  quote_escape = "double",
  eol = "\n"
)
```

Arguments

x	A data frame or tibble to write to disk.
path	File or connection to write to. (path is now deprecated in readr v1.4 for OpenSAFELY currently has older version, so use path for now)
suffix	The suffix used in the name of the json file, to be appended to the delimited file name. Defaults to "" (no suffix), so that the file name is the same as the delimited file name (excluding filetype extensions).
delim	Delimiter used to separate values.
na	String used for missing values. Defaults to "NA". Missing values will never be quoted; strings with the same value as na will always be quoted.
quote_escape	The type of escaping to use for quoted values, one of "double", "backslash" or "none". You can also use FALSE, which is equivalent to "none". The default is "double", which is expected format for Excel.
eol	The end of line character to use. Most commonly either "\n" for Unix style newlines, or "\r\n" for Windows style newlines.

Details

Based on the readr::write_delim function. Additionally, this function saves a json file containing typing info for the data frame, which can be used to re-type the data when re-imported into R. Some further readr::write_delim options are deliberately unavailable as they won't make sense for files intended for re-importing. Datetime and time classes are not supported.

Value

Returns the input invisibly.

Index