Package: pacta.portfolio.import (via r-universe)

September 9, 2024

Title pacta.portfolio.import

Version 0.0.2

Description For more information visit <https://rmi.org/>.

License MIT + file LICENSE

Depends R (>= 3.5)

Imports cli, countrycode, dplyr, fs, progress, purrr, readr (>= 2.1.0), stringi, stringr, vctrs, wand

Suggests covr, devtools, lifecycle, lintr, pkgdown, roxygen2, spelling, styler, testthat (>= 3.0.0), tibble, withr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

Config/Needs/website rmi-pacta/pacta.pkgdown.rmitemplate

URL https://rmi-pacta.github.io/pacta.portfolio.import, https://github.com/RMI-PACTA/pacta.portfolio.import

BugReports https://github.com/RMI-PACTA/pacta.portfolio.import/issues

Repository https://rmi-pacta.r-universe.dev

RemoteUrl https://github.com/rmi-pacta/pacta.portfolio.import

RemoteRef HEAD

RemoteSha 83bb2b2564fc1fdfcb77e39763c7a7fd81796cef

Contents

`determine_headers` *Determine the headers of a portfolio CSV to import*

Description

This function will return a named vector giving the names of the headers in the portfolio CSV that match the proper header names expected by pacta.portfolio.analysis. The name of each element will be the proper column name it matches to.

Usage

```
determine_headers(filepath)
```

Arguments

`filepath`	A character vector containing an absolute or relative path to a single portfolio CSV

Value

A named character vector containing the names of the headers in the portfolio CSV that match the proper header names expected by pacta.portfolio.analysis. The name of each element will be the proper column name it matches to.

`get_csv_specs` *Get a data frame of CSV specifications*

Description

This function will return a data frame with numerous specifications for every CSV file passed in the `files` argument.

Usage

```
get_csv_specs(
  files,
 expected_colnames = c("Investor.Name", "Portfolio.Name", "ISIN", "MarketValue",
    "Currency")
)
```

Arguments

`files` A character vector containing absolute or relative paths to portfolio CSVs, or a directory containing portfolio CSVs

`expected_colnames` A character vector containing the names of the columns expected in the portfolio CSVs

Value

A data frame (invisibly) containing one row per portfolio CSV with columns for each identified specification

guess_delimiter *Guess the delimiter of a delimited file for a vector of filenames or filepaths*

Description

This function will guess the delimiter of a delimited file for a vector of filenames or filepaths and return the delimiter as a string. It defaults to the following delimiters in order if the others are not valid: ",", ";", tab, "|", ":". If the file is inaccessible or binary, it will return `NA` for that element. If you pass anything that is not a character vector or a single column `data.frame` to the `filepaths` argument, this function will give an error.

Usage

```
guess_delimiter(filepaths)
```

Arguments

`filepaths` A character vector

Value

A character vector the same length as `filepaths`.

`guess_file_encoding` *Guess the file encoding for a vector of filenames or filepaths*

Description

This function will guess the file encoding of a vector of filenames or filepaths and return the file encoding as a string. It primarily uses `stringi::stri_enc_detect()` to guess the encoding. Additionally, it searches for known CP850 and CP1252 characters and will return the appropriate encoding if found, because ICU/stringi cannot detect them. If a file is a binary file, it will return `"binary"`. If a file is inaccessible it will return `NA` for that element.

Usage

```
guess_file_encoding(filepaths, threshold = 0.2)
```

Arguments

`filepaths`	A character vector
`threshold`	A single element numeric (minimum confidence level of the guess [0-1])

Value

A character vector the same length as `filepaths`.

`guess_numerical_mark` *Guess the numerical marks in the* `market_value` *column of a portfolio CSV*

Description

This function will guess the numerical marks in the `market_value` column of a portfolio CSV. It will return a single character string containing the guessed decimal or thousands grouping mark, depending on the value passed to `type`, for each portfolio CSV passed in `filepaths`.

Usage

```
guess_numerical_mark(filepaths, type = "decimal")
```

Arguments

`filepaths`	A character vector
`type`	A single character string, either "decimal" or "grouping"

Value

A character vector the same length as `filepaths` containing a single character string defining the guessed numerical mark for each portfolio CSV

`is_file_accessible` *Validate a vector of filenames or filepaths*

Description

This function validates that a vector of filenames or filepaths are accessible files that: are a file, exist, have read access, and are not empty. Dropbox files that are visible but not downloaded locally will be empty files, and they will not pass this validation.

Usage

```
is_file_accessible(filepaths)
```

Arguments

`filepaths`	A character vector

Value

A logical vector the same length as `filepaths`.

`is_readable_file` *Validate read access to files in a vector of filenames or filepaths*

Description

This function validates that files in a vector of filenames or filepaths are readable files and returns `TRUE` or `FALSE` for each one.

Usage

```
is_readable_file(filepaths)
```

Arguments

`filepaths`	A character vector

Value

A logical vector the same length as `filepaths`.

is_text_file *Guess if a file is a text file for a vector of filenames or filepaths*

Description

This function will guess if a file is a text file for a vector of filenames or filepaths and return `TRUE` or `FALSE` for each. It guesses that a file is text if it doesn't find any nul bytes in the first 2048 bytes of the file. This is an imperfect guess, but it is very likely that a binary/non-text file will have a nul byte near the beginning of the file. This might guess that a file that is not intended to be read in as text is a text file, but at least you will likely be able to read in the file as text without error. If the file is inaccessible, either because it is empty, you don't have permission to read it, it's a directory, or it doesn't exist, this function will return `FALSE`. If you pass anything that is not a character vector or a single column data.frame to the `filepaths` argument, this function will give an error.

Usage

```
is_text_file(filepaths)
```

Arguments

`filepaths`	A character vector

Value

A logical vector the same length as `filepaths`.

is_valid_currency_code *Validate a vector of currency codes*

Description

This function validates that a vector of currency codes are valid currency codes that exist in the ISO 4217 alpha code specification.

Usage

```
is_valid_currency_code(currency_codes)
```

Arguments

`currency_codes`	A character vector

Value

A logical vector the same length as `currency_codes`.

is_valid_cusip *Validate a vector of CUSIPs*

Description

This function validates that a vector of CUSIPs are valid CUSIP codes.

Usage

```
is_valid_cusip(cusips)
```

Arguments

cusips	A character or numeric vector (automatically collapses a single column data.frame to a vector)

Value

A logical vector the same length as `cusips`.

is_valid_isin *Validate a vector of ISINs*

Description

This function validates that a vector of ISINs are valid codes that conform to the ISO 6166 specification with `TRUE` or `FALSE`. It checks the basic structure (2 alpha characters, 9 alpha-numeric characters, 1 check digit) and also validates the check digit using the Luhn algorithm.

Usage

```
is_valid_isin(isins)
```

Arguments

isins	A character vector

Value

A logical vector the same length as `isins`.

read_portfolio_csv — *Read in portfolio CSV/s, working around a number of non-standard issues*

Description

This function will read in one more portfolio CSVs. It works around a number of common issues, like alternate column names, alternate delimiter, alternate decimal and grouping marks, file encodings besides ASCII or UTF-8, etc.

Usage

```
read_portfolio_csv(filepaths, combine = TRUE)
```

Arguments

`filepaths`	A character vector or single column data frame (strings should be valid file paths to CSV files or a directory that contains CSV files)
`combine`	A single element logical (default `TRUE`)

Value

If `combine` is `TRUE`, returns a tbl_df with all of the readable data from the portfolio CSVs combined. If `combine` is `FALSE`, returns a list of tbl_dfs, one for each readable portfolio CSV.

Index