Package 'r2dii.match'

June 30, 2020

Title Tools to Match Financial Portfolios with Climate Data

Version 0.0.3

Description These tools implement in R a fundamental part of the software 'PACTA' (Paris Agreement Capital Transition Assessment), which is a free tool that calculates the alignment between financial portfolios and climate scenarios (<https://2degrees-investing.org/>). Financial institutions use 'PACTA' to study how their capital allocation impacts the climate. This package matches data from financial portfolios to asset level data from market-intelligence databases (e.g. power plant capacities, emission factors, etc.). This is the first step to assess if a financial portfolio aligns with climate goals.

License GPL-3

URL https://2degreesinvesting.github.io/r2dii.match, https://github.com/2DegreesInvesting/r2dii.match

BugReports https://github.com/2DegreesInvesting/r2dii.match/issues

Depends R (>= 3.4)

Imports dplyr (>= 0.8.5), glue, magrittr, purrr, r2dii.data, rlang, stringdist, stringi, tibble, tidyr, tidyselect

Suggests covr, rmarkdown, spelling, testthat (>= 2.1.0)

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Mauro Lepore [aut, cre, ctr] (<https://orcid.org/0000-0002-1986-7988>), Jackson Hoffart [aut] (<https://orcid.org/0000-0002-8600-5042>), Klaus Hagedorn [aut], Florence Palandri [aut], Evgeny Petrovsky [aut], 2 Degrees Investing Initiative [cph, fnd]

Maintainer Mauro Lepore <maurolepore@gmail.com>

Repository CRAN

Date/Publication 2020-06-30 14:30:02 UTC

R topics documented:

`match_name` *Match a loanbook and asset-level datasets (ald) by the* name_* *columns*

Description

`match_name()` scores the match between names in a loanbook dataset (columns can be `name_direct_loantaker`, name_intermediate_parent* and `name_ultimate_parent`) with names in an asset-level dataset (column `name_company`). The raw names are first internally transformed, and aliases are assigned. The similarity between aliases in each of the loanbook and ald datasets is scored using `stringdist::stringsim()`.

Usage

```
match_name(
  loanbook,
  ald,
  by_sector = TRUE,
  min_score = 0.8,
  method = "jw",
  p = 0.1,
  overwrite = NULL
)
```

Arguments

`loanbook, ald`	data frames structured like r2dii.data::loanbook_demo and r2dii.data::ald_demo.
`by_sector`	Should names only be compared if companies belong to the same `sector`?
`min_score`	A number between 0-1, to set the minimum `score` threshold. A `score` of 1 is a perfect match.
`method`	Method for distance calculation. One of `c("osa","lv","dl","hamming","lcs","qgram","cosine","` See stringdist::stringdist-metrics.
`p`	Penalty factor for Jaro-Winkler distance. The valid range for `p` is `0 <= p <= 0.25`. If `p=0` (default), the Jaro-distance is returned. Applies only to `method='jw'`.

`overwrite` A data frame used to overwrite the `sector` and/or `name` columns of a particular direct loantaker or ultimate parent. To overwrite only `sector`, the value in the `name` column should be `NA` and vice-versa. This file can be used to manually match loanbook companies to ald.

Value

A data frame with the same groups (if any) and columns as `loanbook`, and the additional columns:

- `id_2dii` - an id used internally by `match_name()` to distinguish companies
- `level` - the level of granularity that the loan was matched at (e.g `direct_loantaker` or `ultimate_parent`)
- `sector` - the sector of the `loanbook` company
- `sector_ald` - the sector of the `ald` company
- `name` - the name of the `loanbook` company
- `name_ald` - the name of the `ald` company
- `score` - the score of the match (manually set this to `1` prior to calling `prioritize()` to validate the match)
- `source` - determines the source of the match. (equal to `loanbook` unless the match is from `overwrite`

The returned rows depend on the argument `min_value` and the result of the column `score` for each loan: * If any row has `score` equal to 1, `match_name()` returns all rows where `score` equals 1, dropping all other rows. * If no row has `score` equal to 1,`match_name()` returns all rows where `score` is equal to or greater than `min_score`. * If there is no match the output is a 0-row tibble with the expected column names – for type stability.

Assigning aliases

The transformation process used to compare names between loanbook and ald datasets applies best practices commonly used in name matching algorithms:

- Remove special characters.
- Replace language specific characters.
- Abbreviate certain names to reduce their importance in the matching.
- Spell out numbers to increase their importance.

Handling grouped data

This function ignores but preserves existing groups.

See Also

Other user-oriented: `prioritize()`

Examples

```
library(dplyr, warn.conflicts = FALSE)
library(r2dii.data)

mini_loanbook <- sample_n(loanbook_demo, 10)

match_name(mini_loanbook, ald_demo)

match_name(
  mini_loanbook, ald_demo,
  min_score = 0.9,
  by_sector = TRUE
)
```

---

prioritize *Pick rows where* score *is 1 and* level *per loan is of highest* priority

---

Description

When multiple perfect matches are found per loan (e.g. a match at `direct_loantaker` level and `ultimate_parent` level), we must prioritize the desired match. By default, the highest `priority` is the most granular match (i.e. `direct_loantaker`).

Usage

```
prioritize(data, priority = NULL)
```

Arguments

`data`	A data frame like the validated output of `match_name()`. See *Details* on how to validate `data`.
`priority`	One of: • `NULL`: defaults to the default level priority as returned by `prioritize_level()`. • A character vector giving a custom priority. • A function to apply to the output of `prioritize_level()`, e.g. `rev`. • A quosure-style lambda function, e.g. `~ rev(.x)`.

Details

How to validate `data`

Write the output of `match_name()` into a .csv file with:

```
# Writting to current working directory
matched %>%
  readr::write_csv("matched.csv")
```

Compare, edit, and save the data manually:

- Open *matched.csv* with any spreadsheet editor (e.g. MS Excel, Google Sheets).
- Compare the columns `name` and `name_ald` manually to determine if the match is valid. Other information can be used in conjunction with just the names to ensure the two entities match (sector, internal information on the company structure, etc.)
- Edit the data:
  - If you are happy with the match, set the `score` value to `1`.
  - Otherwise set or leave the `score` value to anything other than `1`.
- Save the edited file as, say, *valid_matches.csv*.

Re-read the edited file (validated) with:

```
# Reading from current working directory
valid_matches <- readr::read_csv("valid_matches.csv")
```

Value

A data frame with a single row per loan, where `score` is 1 and priority level is highest.

Handling grouped data

This function ignores but preserves existing groups.

See Also

`match_name()`, `prioritize_level()`.

Other user-oriented: `match_name()`

Examples

```
library(dplyr)

# styler: off
matched <- tribble(
  ~sector, ~sector_ald,  ~score, ~id_loan,               ~level,
   "coal",      "coal",       1,     "aa",     "ultimate_parent",
   "coal",      "coal",       1,     "aa",    "direct_loantaker",
   "coal",      "coal",       1,     "bb", "intermediate_parent",
   "coal",      "coal",       1,     "bb",     "ultimate_parent",
)
# styler: on

prioritize_level(matched)

# Using default priority
prioritize(matched)

# Using the reverse of the default priority
prioritize(matched, priority = rev)

# Same
```

```
prioritize(matched, priority = ~ rev(.x))

# Using a custom priority
bad_idea <- c("intermediate_parent", "ultimate_parent", "direct_loantaker")

prioritize(matched, priority = bad_idea)
```

`prioritize_level` *Arrange unique* `level` *values in default order of* `priority`

Description

Arrange unique `level` values in default order of `priority`

Usage

```
prioritize_level(data)
```

Arguments

`data`	A data frame, commonly the output of `match_name()`.

Value

A character vector of the default level priority per loan.

Examples

```
matched <- tibble::tibble(
  level = c(
    "intermediate_parent_1",
    "direct_loantaker",
    "direct_loantaker",
    "direct_loantaker",
    "ultimate_parent",
    "intermediate_parent_2"
  )
)
prioritize_level(matched)
```

Index