Package: nomesbr (via r-universe)

February 26, 2026

Type Package

Title Limpa e Simplifica Nomes de Pessoas (Name Cleaner and Simplifier)

Version 0.1.0

Description Limpa e simplifica nomes de pessoas para auxiliar no pareamento de banco de dados na ausência de chaves únicas não ambíguas. Detecta e corrige erros tipográficos mais comuns, simplifica opcionalmente termos sujeitos eventualmente a omissão em cadastros, e simplifica foneticamente suas palavras, aplicando variação própria do algoritmo metaphoneBR. (Cleans and simplifies person names to assist in database matching when unambiguous unique keys are unavailable. Detects and corrects common typos, optionally simplifies terms prone to omission in records, and applies phonetic simplification using a custom variation of the metaphoneBR algorithm.) Mation (2025) <doi:10.6082/uchicago.15104>.

License MIT + file LICENSE

Encoding UTF-8

Language pt

LazyData true

RoxygenNote 7.3.2

Imports data.table, dplyr, httr2, stringr, tictoc

Suggests testthat (>= 3.0.0), DBI, duckdb, digest, mockery, knitr, rmarkdown

Depends R (>= 4.3.0)

Config/testthat/edition 3

URL https://github.com/ipeadata-lab/nomesbr, https://ipeadata-lab.github.io/nomesbr/

BugReports https://github.com/ipeadata-lab/nomesbr/issues

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libssl-dev

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

Date/Publication 2025-12-26 18:46:06 UTC

RemoteUrl https://github.com/ipea/nomesbr

RemoteRef HEAD

RemoteSha fd0e281cc787016a5f1ba00164200bc72976c38b

Contents

`consulta_nome_em_central` *Consulta Nomes em uma Base de Dados DuckDB*

Description

Realiza uma consulta a uma tabela de nomes em um banco de dados DuckDB, retornando todas as colunas para os nomes que correspondem à lista de entrada.

Usage

```
consulta_nome_em_central(nomes, mestre, usar_hash = TRUE)
```

Arguments

`nomes`	Um vetor de caracteres (character vector) contendo os nomes ou hashes a serem consultados.
`mestre`	Uma string com o caminho para o banco de dados DuckDB (arquivo '.duckdb').
`usar_hash`	Logico. Se `TRUE` (default), a consulta vai ser feita na coluna `'nome_original_hash'`. Se `FALSE`, a consulta vai ser feita na coluna `'nome_original'`.

Details

A função se conecta a um banco de dados DuckDB especificado pelo caminho em ‘mestre‘. A consulta é otimizada para buscar múltiplos nomes de uma vez, gerando uma instrução SQL com parâmetros para evitar injeção de SQL.

O parâmetro ‘usar_hash‘ permite escolher a coluna para a busca:

- Se `TRUE` (padrão), a busca é feita na coluna `'nome_original_hash'`. Isso é ideal se os nomes na tabela estão armazenados como hashes (ex: SHA-256), pois pode ser mais rápido e seguro para comparações exatas.
- Se `FALSE`, a busca é feita na coluna `'nome_original'`, que deve conter os nomes em formato de texto.

A fun\ç\ão gerencia automaticamente a conex\ão com o banco de dados, garantindo que ela seja fechada ao final da execu\ç\ão, mesmo que ocorra um erro.

Value

Um `data.frame` contendo os resultados da consulta. Se nenhum nome for encontrado, retorna um `data.frame` com zero linhas e as colunas da tabela `nomes_limpos`.

Examples

```
## Not run:
# Exemplo de uso com hash (padrão)
# Suponha que 'caminho/para/meu_banco.duckdb' existe e tem a tabela 'nomes_limpos'
# com uma coluna 'nome_original_hash'.
hashes_para_buscar <- c("a1b2c3...", "d4e5f6...")
resultados_hash <- consulta_nome_em_central(
  nomes = hashes_para_buscar,
  mestre = "caminho/para/meu_banco.duckdb"
)

# Exemplo de uso com texto
# Suponha que a tabela 'nomes_limpos' também tem uma coluna 'nome_original'.
nomes_para_buscar <- c("João da Silva", "Maria Oliveira")
resultados_texto <- consulta_nome_em_central(
  nomes = nomes_para_buscar,
  mestre = "caminho/para/meu_banco.duckdb",
  usar_hash = FALSE
)

## End(Not run)
```

identificar_adicionar_nome_proprio *Adiciona Nome Próprio Validado de 'nomes_proprios_compostos' .*

Description

Adiciona Nome Próprio Validado de 'nomes_proprios_compostos' .

Usage

```
identificar_adicionar_nome_proprio(dt, s)

add_nome_proprio_to_word1_and_word2p(dt, s)
```

Arguments

dt	Um 'data.table'.
s	Nome da coluna (string) base para derivação das colunas de palavras (por exemplo, se 's = "nome_simpl"', espera 'nome_simpl1', 'nome_simpl2p').

Value

O 'data.table' 'dt' com colunas '_v2' adicionadas.

Examples

```
dt_nomes <- data.table::data.table(nome=c("MARIA DO SOCORRO SILVA",
"ANA PAULA DE OLIVEIRA","JOSE DAS FLORES"))
dt_nomes <- identificar_adicionar_nome_proprio(dt_nomes,"nome")
print(dt_nomes)
```

limpar_nomes *Limpa e Analisa Nomes em um data.table*

Description

Processa uma coluna de nomes em um 'data.table', aplicando uma série de regras de limpeza para identificar e corrigir/marcar problemas comuns como menções a "FALECIDO", "CARTORIO", erros de digitação, espaços indevidos, etc.

Usage

```
limpar_nomes(d, s)

find_and_clean_NAnames_and_extra_spaces(d, s)
```

Arguments

d	Um objeto ‘data.table‘.
s	O nome da coluna (em string) dentro de ‘d‘ que contém os nomes a serem processados.

Details

A função executa os seguintes passos principais:

1. Cria uma cópia da coluna de nomes para limpeza.
2. Detecta e trata menções a "FALECIDO(A)".
3. Detecta e trata menções a "CARTORIO" e nomes de cidades comuns em registros.
4. Corrige espaçamento perto de caracteres especiais com ‘limpa_espaco_acento_til_apostrofe‘.
5. Identifica e trata nomes contendo termos problemáticos como "PAI", "MAE", "SEM", "NAO", exceto em contextos aceitáveis.
6. Identifica e trata casos de "NADA CONSTA" e variações.
7. Corrige E, DA, DE e variantes com caracter prévio indevido (ex: "EDAS" para "DAS" se aplicável).
8. Remove saudações como "SR.", "SRA.".
9. Remove termos como "IGNORADO", "DESCONHECIDO".
10. Remove repetições de partículas de ligação (ex: "DE DE").
11. Limpa letras repetidas no início ou meio de palavras.

Value

Um ‘data.table‘ modificado, contendo a coluna original, uma nova coluna com sufixo "_clean" com os nomes limpos, e colunas booleanas indicando a detecção de cada tipo de problema (ex: ‘falecido‘, ‘cartorio‘).

Examples

```
# Supondo que 'meu_DT' é um data.table com uma coluna 'nome_sujo'
DT_exemplo <- data.table::data.table(
id = 1:3,
nome_sujo = c("MARIA FALECIDA SSILVA", "CARTORIO DE PAZ", "JOAO D ARC")
)
DT_limpo <- limpar_nomes(DT_exemplo, "nome_sujo")
print(DT_limpo)
```

`remove_PARTICULAS_AGNOMES` *Remove Partículas, Agnomes e algumas Patentes de Nomes*

Description

Remove Partículas, Agnomes e algumas Patentes de Nomes

Usage

```
remove_PARTICULAS_AGNOMES(s)
```

Arguments

`s`	Vetor de caracteres contendo nomes.

Value

Vetor de caracteres com nomes simplificados.

Examples

```
vct_nomes <- c("JOAO DA SILVA FILHO","CORONEL JACINTO")
remove_PARTICULAS_AGNOMES(vct_nomes)
```

`segmentar_nomes` *Adiciona Colunas com Partes do Nome (w1, w2, w3, w2p, w12p)*

Description

Adiciona Colunas com Partes do Nome (w1, w2, w3, w2p, w12p)

Usage

```
segmentar_nomes(dt, s)

add_string_w1_w2_w3_and_w2p(dt, s)
```

Arguments

`dt`	Um ‘data.table‘.
`s`	Nome da coluna (string) em ‘dt‘ contendo os nomes.

Value

O ‘data.table‘ ‘dt‘ modificado por referência, com novas colunas.

Examples

```
dt_nomes <- data.table::data.table(nome=c("MARIA DO SOCORRO SILVA",
"ANA PAULA DE OLIVEIRA"))
dt_nomes <- segmentar_nomes(dt_nomes,"nome")
print(dt_nomes)
```

`simplifica_PARTICULAS_AGNOMES_PATENTES` *Cria coluna com agnomes, algumas patentes/cargos as remove, remove partículas*

Description

Cria coluna com agnomes, algumas patentes/cargos as remove, remove partículas

Usage

```
simplifica_PARTICULAS_AGNOMES_PATENTES(d, s = "nome_clean")
```

Arguments

d	um objeto 'data.table'
s	string com nome da coluna de caracteres contendo nomes para simplificar. Por padrão, "nome_clean".

Value

data.table com novas colunas de nome simplificado e de marca agnomes_titulos

Examples

```
dt_nomes <- data.table::data.table(nome = c("JOAO DA SILVA FILHO",
"CORONEL JACINTO"))
dt_nomes <- simplifica_PARTICULAS_AGNOMES_PATENTES(d=dt_nomes,s="nome")
print(dt_nomes)
```

`tabular_problemas_em_nomes` *Tabula Problemas Detectados nos Nomes*

Description

Cria uma tabela resumo contabilizando o número de ocorrências para cada tipo de problema detectado pela função ‘marcar_problemas_e_limpar_nomes‘.

Usage

```
tabular_problemas_em_nomes(d, s)

tabulate_name_poblems(d, s)
```

Arguments

d	O ‘data.table‘ retornado por ‘marcar_problemas_e_limpar_nomes‘.
s	O nome da coluna original (string) que foi processada.

Value

Um ‘data.table‘ com as colunas:

- ‘condition‘: O nome da condição/problema verificado.
- ‘N_detected‘: Número de vezes que a condição foi detectada.
- ‘N_made_NA‘: Número de detecções que resultaram na limpeza para ‘NA‘.
- ‘N_replaced‘: Número de detecções onde o nome foi alterado (não para ‘NA‘).

Examples

```
DT_limpo <- data.table::data.table(nome = c("JOSEE SILVA",
"RAIMUNDA DA DA SILVA"), nome_clean = c("JOSE SILVA",
"RAIMUNDA DA SILVA"),
falecido = NA, cartorio = NA,
espaco_TilAcentoApostrofe = NA,
nome_P_M_S_N = NA, nada_nao = NA,
nada_nao_consta2 = NA, final_missing = NA, Xartigo = NA, sr_sra = NA,
ignorado = NA, dededada = 1, letra_repetida = 1)
sumario <- tabular_problemas_em_nomes(DT_limpo, "nome")
print(sumario)
```

Index