Package: cnefetools (via r-universe)

November 1, 2026

Title Access and Analysis of Brazilian CNEFE Address Data

Version 0.3.0.9000

Description Download, cache and read municipality-level address data from the Cadastro Nacional de Enderecos para Fins Estatisticos (CNEFE) of the 2022 Brazilian Census, published by the Instituto Brasileiro de Geografia e Estatistica (IBGE) <https://ftp.ibge.gov.br/Cadastro_Nacional_de_Enderecos_para_Fins_Estatisticos/>. Beyond data access, provides spatial aggregation of addresses, computation of land-use mix indices, and dasymetric interpolation of census tract variables using CNEFE dwelling points as ancillary data. Results can be produced on 'H3' hexagonal grids or user-supplied polygons, and heavy operations leverage a 'DuckDB' backend with extensions for fast, in-process execution.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL https://github.com/pedreirajr/cnefetools, https://pedreirajr.github.io/cnefetools/

BugReports https://github.com/pedreirajr/cnefetools/issues

Suggests ggplot2, kableExtra, knitr, leafsync, mapview, odbr, rmarkdown, scales, testthat (>= 3.0.0), zip

Config/testthat/edition 3

Depends R (>= 4.4.0)

Imports arrow, dplyr, sf, geobr (>= 2.0.0), lifecycle, rlang, h3jsr, tidyr, DBI, duckdb, duckspatial (>= 1.0.0), cli (>= 3.6.0), checkmate, fs, httr2, piggyback, withr

LazyData true

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev git make libicu-dev libuv1-dev libxml2-dev libzstd-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libnode-dev

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

Date/Publication 2026-10-02 13:30:45 UTC

RemoteUrl https://github.com/pedreirajr/cnefetools

RemoteRef HEAD

RemoteSha bbc10d6a629216c79e85b953aac6af16c7a4b666

Contents

clear_cache_muni 2
clear_cache_tracts 3
cnefe_counts 4
cnefe_dictionary 7
cnefe_doc 7
cnefe_export 8
compute_lumi 9
read_cnefe 12
tracts_to_h3 14
tracts_to_polygon 16
tracts_variables_ref 18

Index **20**

`clear_cache_muni` *Delete cached CNEFE data files*

Description

`clear_cache_muni()` removes CNEFE data files stored in the user cache directory by `cnefe_counts()`, `compute_lumi()`, `tracts_to_h3()`, and related functions.

The cache holds gzipped CSVs (`.csv.gz`). Archives left by versions before 0.3.0, which cached the ZIP as published by IBGE, are removed as well.

Usage

```
clear_cache_muni(
  code_muni = "all",
  verbose = TRUE,
  cache_dir = NULL,
  year = NULL
)
```

Arguments

code_muni	Integer or "all". If "all" (default), every cached CNEFE file is deleted. If a seven-digit IBGE municipality code is provided, only the file for that municipality is deleted.
verbose	Logical; if TRUE (default), reports the number of files deleted and the space freed.
cache_dir	Character. Directory to use for cached downloads. If NULL (default), the CNEFETOOLS_CACHE_DIR environment variable is used when it is set, otherwise tools::R_user_dir() with which = "cache". Use this to point large downloads at a secondary drive or a shared volume.
year	Integer. Restrict the deletion to one CNEFE edition. NULL (default) clears every edition, which is the previous behaviour.

Value

Invisibly, the character vector of deleted file paths.

Examples

```
# Delete every cached CNEFE file
clear_cache_muni()

# Delete only the file for Lauro de Freitas-BA
clear_cache_muni(2919207)
```

---

clear_cache_tracts *Delete cached census tract Parquet files*

Description

clear_cache_tracts() removes census tract Parquet files stored in the user cache directory by tracts_to_h3() and tracts_to_polygon().

Usage

```
clear_cache_tracts(uf = "all", verbose = TRUE, cache_dir = NULL, year = NULL)
```

Arguments

uf	"all", a two-letter UF abbreviation (e.g. "BA"), a two-digit numeric state code (e.g. 29L), or a seven-digit IBGE municipality code (e.g. 2919207). If "all" (default), all cached Parquet files are deleted. Otherwise, only the file for the resolved state is deleted.
verbose	Logical; if TRUE (default), reports the number of files deleted and the space freed.

cache_dir	Character. Directory to use for cached downloads. If NULL (default), the CNEFETOOLS_CACHE_DIR environment variable is used when it is set, otherwise tools::R_user_dir() with which = "cache". Use this to point large downloads at a secondary drive or a shared volume.
year	Integer. Restrict the deletion to one CNEFE edition. NULL (default) clears every edition, which is the previous behaviour.

Value

Invisibly, the character vector of deleted file paths.

Examples

```
# Delete all cached census tract Parquets
clear_cache_tracts()

# Delete only the Parquet for Bahia (several equivalent calls)
clear_cache_tracts("BA")
clear_cache_tracts(29)
clear_cache_tracts(2919207)  # municipality code → state resolved automatically
```

---

cnefe_counts *Count CNEFE address species on a spatial grid*

Description

cnefe_counts() reads CNEFE records for a given municipality, assigns each address point to spatial units (either H3 hexagonal cells or user-provided polygons), and returns per-unit counts of COD_ESPECIE as addr_type1 to addr_type8.

Usage

```
cnefe_counts(
  code_muni,
  year = 2022,
  polygon_type = lifecycle::deprecated(),
  polygon = NULL,
  crs_output = NULL,
  h3_resolution = 9,
  verbose = TRUE,
  cache = TRUE,
  cache_dir = NULL,
  backend = c("duckdb", "r")
)
```

Arguments

code_muni	Integer. Seven-digit IBGE municipality code.
year	Integer. The CNEFE data year. Currently only 2022 is supported. Defaults to 2022.
polygon_type	**[Deprecated]** The aggregation mode is now inferred from polygon: leave it NULL for an H3 grid, or pass an sf::sf object for user polygons. Passing polygon_type still works and warns.
polygon	An sf::sf object with polygon geometries. Supplying it switches the output from an H3 grid to these polygons. A warning is issued reporting the percentage of CNEFE points covered by the polygon area. If no CNEFE points fall within the polygon, an error is raised.
crs_output	The CRS for the output object. Only used when polygon is supplied. Default is NULL, which uses the original CRS of the polygon argument. Can be an EPSG code (e.g., 4326, 31983) or any CRS object accepted by sf::st_transform().
h3_resolution	Integer. H3 grid resolution (default: 9). Only used for the H3 grid, so it is ignored when polygon is supplied.
verbose	Logical; if TRUE, prints messages and timing information.
cache	Logical. If TRUE (default), the downloaded data is stored as a gzipped CSV in the user cache directory and reused in future calls. If FALSE, a temporary file is used and deleted after the call.
cache_dir	Character. Directory to use for cached downloads. If NULL (default), the CNEFETOOLS_CACHE_DIR environment variable is used when it is set, otherwise tools::R_user_dir() with which = "cache". Use this to point large downloads at a secondary drive or a shared volume.
backend	Character. "duckdb" (default) uses DuckDB with the H3 extension, and the spatial extension as well when polygon is supplied. "r" uses h3jsr and sf in R instead, and needs no DuckDB extension. "r" exists for environments where DuckDB extensions cannot be installed, such as some restricted computing clusters. It is **not** the lighter option: it materialises the filtered address table in R memory, so its footprint grows with the municipality, while DuckDB aggregates in a streaming fashion and stays nearly flat. On São Paulo (5.7 million addresses) the measured peak is about 9 GB under "r" against 0.7 GB under "duckdb", alongside being roughly 13 times slower. If the constraint is memory rather than installability, keep the DuckDB backend and cap it with the cnefetools.duckdb_config option instead. See ?cnefetools for that option, and the benchmark article at https://pedreirajr.github.io/cnefetools/articles/bench_duckdb.html for the measurements.

Details

The counts in the columns addr_type1 to addr_type8 correspond to:

- addr_type1: Private household (Domicílio particular)
- addr_type2: Collective household (Domicílio coletivo)
- addr_type3: Agricultural establishment (Estabelecimento agropecuário)

- addr_type4: Educational establishment (Estabelecimento de ensino)
- addr_type5: Health establishment (Estabelecimento de saúde)
- addr_type6: Establishment for other purposes (Estabelecimento de outras finalidades)
- addr_type7: Building under construction or renovation (Edificação em construção ou reforma)
- addr_type8: Religious establishment (Estabelecimento religioso)

All eight types are reported. In particular, addr_type7 is retained here, whereas compute_lumi() excludes it when computing land-use mix indices.

Value

An sf::sf object containing:

- id_hex (when polygon is NULL): H3 cell identifier
- Original columns from polygon (when polygon is supplied)
- addr_type1 ... addr_type8: counts per address type
- geometry: polygon geometry

When polygon is supplied, the output CRS matches the original polygon CRS (or crs_output if specified).

See Also

compute_lumi() for land-use mix indices on the same spatial units.

Examples

```
# Count addresses per H3 hexagon (resolution 9)
hex_counts <- cnefe_counts(code_muni = 2929057, cache = FALSE)

# Count addresses per user-provided polygon (neighborhoods of Lauro de Freitas-BA)
# Using geobr to download neighborhood boundaries
library(geobr)
nei_ldf <- subset(
  read_neighborhood(year = 2022),
  code_muni == 2919207
)
nei_counts <- cnefe_counts(
  code_muni = 2919207,
  polygon = nei_ldf,
  cache = FALSE
)
```

cnefe_dictionary *Open the official CNEFE data dictionary*

Description

Opens the bundled Excel data dictionary in the system's default spreadsheet viewer (e.g., Excel, LibreOffice).

Usage

```
cnefe_dictionary(year = 2022)
```

Arguments

year	Integer. The CNEFE data year. Currently only 2022 is supported.

Value

Invisibly, the path to the Excel file inside the installed package.

Examples

```
cnefe_dictionary()
```

cnefe_doc *Open the official CNEFE methodological note*

Description

Opens the bundled PDF methodological document in the system's default PDF viewer.

Usage

```
cnefe_doc(year = 2022)
```

Arguments

year	Integer. The CNEFE data year. Currently only 2022 is supported.

Value

Invisibly, the path to the PDF file inside the installed package.

Examples

```
cnefe_doc()
```

`cnefe_export` *Export CNEFE data to a persistent, optimised file*

Description

`cnefe_export()` downloads a municipality (or reuses the cache) and writes it to a location and format of your choosing, so the data no longer depends on the package cache being present or on the IBGE server being reachable.

Usage

```
cnefe_export(
  code_muni,
  path,
  format = c("parquet", "csv", "csv.gz"),
  year = 2022,
  overwrite = FALSE,
  cache = TRUE,
  cache_dir = NULL,
  verbose = TRUE
)
```

Arguments

`code_muni`	Integer. Seven-digit IBGE municipality code.
`path`	Character. Directory to write into. Created if missing.
`format`	Character. `"parquet"` (default), `"csv"` or `"csv.gz"`.
`year`	Integer. The CNEFE data year. Currently only 2022 is supported.
`overwrite`	Logical. Whether to replace an existing file. Defaults to `FALSE`, which errors instead, since these files are expensive to produce.
`cache`	Logical. Whether to use the package cache for the download.
`cache_dir`	Character. Directory to use for cached downloads. If `NULL` (default), the `CNEFETOOLS_CACHE_DIR` environment variable is used when it is set, otherwise `tools::R_user_dir()` with `which = "cache"`.
`verbose`	Logical. Whether to print progress.

Details

The package cache is designed to be transient: it lives in a directory the package manages, it holds the ZIP exactly as IBGE published it, and `clear_cache_muni()` is expected to empty it. That is the wrong shape for a reproducible analysis that must still run in a year.

This function fills that gap. Point it at a project directory, a shared volume or an external drive, choose a format, and use the resulting file directly:

```
path <- cnefe_export(2919207, "data/cnefe")
cnefe <- read_cnefe(file = path)
```

`read_cnefe()` accepts any file this function writes, and also the raw ZIP as distributed by IBGE, so a file obtained by other means can be read without the download step at all.

Parquet is the default because it is columnar, typed and compressed, which makes it markedly smaller and faster to read than the published CSV, and because it is the format Arrow and DuckDB both read natively.

Value

The path to the written file, invisibly.

See Also

`read_cnefe()`, which reads the result back through its `file` argument, and `clear_cache_muni()` for the transient cache.

Examples

```
# Write a municipality to a project directory as Parquet
path <- cnefe_export(2929057, path = tempdir(), cache = FALSE, overwrite = TRUE)

# Read it back without touching the network
cnefe <- read_cnefe(file = path)
```

---

compute_lumi *Compute land-use mix indicators on a spatial grid*

Description

`compute_lumi()` reads CNEFE records for a given municipality, assigns each address point to spatial units (either H3 hexagonal cells or user-provided polygons), and computes the residential proportion (`p_res`) and land-use mix indices, such as the Entropy Index (`ei`), the Herfindahl-Hirschman Index (`hhi`), the Balance Index (`bal`), the Index of Concentration at Extremes (`ice`), the adapted HHI (`hhi_adp`), and the Bidirectional Global-centered Balance Index (`bgbi`), following the methodology proposed in Pedreira Junior et al. (2025, 2026). The 2026 article introduces the BGBI, and the adapted HHI is documented in the 2025 preprint.

Usage

```
compute_lumi(
  code_muni,
  year = 2022,
  polygon_type = lifecycle::deprecated(),
  polygon = NULL,
```

```
  crs_output = NULL,
  h3_resolution = 9,
  verbose = TRUE,
  cache = TRUE,
  cache_dir = NULL,
  backend = c("duckdb", "r")
)
```

Arguments

code_muni	Integer. Seven-digit IBGE municipality code.
year	Integer. The CNEFE data year. Currently only 2022 is supported. Defaults to 2022.
polygon_type	**[Deprecated]** The aggregation mode is now inferred from polygon: leave it NULL for an H3 grid, or pass an sf::sf object for user polygons. Passing polygon_type still works and warns.
polygon	An sf::sf object with polygon geometries. Supplying it switches the output from an H3 grid to these polygons. A warning is issued reporting the percentage of CNEFE points covered by the polygon area. If no CNEFE points fall within the polygon, an error is raised.
crs_output	The CRS for the output object. Only used when polygon is supplied. Default is NULL, which uses the original CRS of the polygon argument. Can be an EPSG code (e.g., 4326, 31983) or any CRS object accepted by sf::st_transform().
h3_resolution	Integer. H3 grid resolution (default: 9). Only used for the H3 grid, so it is ignored when polygon is supplied.
verbose	Logical; if TRUE, prints messages and timing information.
cache	Logical. If TRUE (default), the downloaded data is stored as a gzipped CSV in the user cache directory and reused in future calls. If FALSE, a temporary file is used and deleted after the call.
cache_dir	Character. Directory to use for cached downloads. If NULL (default), the CNEFETOOLS_CACHE_DIR environment variable is used when it is set, otherwise tools::R_user_dir() with which = "cache". Use this to point large downloads at a secondary drive or a shared volume.
backend	Character. "duckdb" (default) uses DuckDB with the H3 extension, reading the cached gzipped CSV directly. "r" computes H3 in R using h3jsr instead, and needs no DuckDB extension. "r" exists for environments where DuckDB extensions cannot be installed, such as some restricted computing clusters. It is **not** the lighter option: it materialises the filtered address table in R memory, so its footprint grows with the municipality, while DuckDB aggregates in a streaming fashion and stays nearly flat. On São Paulo (5.7 million addresses) the measured peak is about 8.4 GB under "r" against 0.6 GB under "duckdb", alongside being roughly 15 times slower. If the constraint is memory rather than installability, keep the DuckDB backend and cap it with the cnefetools.duckdb_config option instead. See ?cnefetools for that option, and the benchmark article at https://pedreirajr.github.io/cnefetools/articles/bench_duckdb.html for the measurements.

Details

Binary land-use classification:
The indices computed here rest on a binary split. An address is counted as residential when `COD_ESPECIE == 1` (private household), and as non-residential otherwise. This follows the formulation of the indices as published in Pedreira Junior et al. (2026), where the measures are defined and empirically validated on that two-category basis.

Exclusion of buildings under construction:
`compute_lumi()` drops records with `COD_ESPECIE == 7` (building under construction or renovation), because such records describe a transitional state rather than a realised land use. Note that `cnefe_counts()` does **not** apply this exclusion and reports these records as `addr_type7`.

The citywide baseline P:
Two indices use a citywide residential share P: the `bgbi` index, which is referenced against it, and the Balance Index (`bal`), which uses it through r = P / (1 - P). The other indices are computed entirely within each spatial unit. Two properties of P are worth stating.
First, P is computed from CNEFE address-type counts rather than from census population, so it describes the distribution of address types and not the distribution of residents.
Second, P is always computed over the full municipality, including when `polygon` is supplied, so it does not adapt to the area the supplied polygons happen to cover. This is intended, as P describes the context the addresses sit in, which is the municipality, and a sub-area of a city is still part of that wider context. A baseline recomputed over the sub-area would measure something different, namely mix relative to the sub-area itself rather than relative to the city.

Value

An `sf::sf` object containing:

When `polygon` is `NULL` (H3 grid):
- `id_hex`: H3 cell identifier
- `p_res`, `ei`, `hhi`, `bal`, `ice`, `hhi_adp`, `bgbi`: land-use mix indicators
- `geometry`: hexagon geometry (CRS 4326)

When `polygon` is supplied:
- Original columns from `polygon`
- `p_res`, `ei`, `hhi`, `bal`, `ice`, `hhi_adp`, `bgbi`: land-use mix indicators
- `geometry`: polygon geometry (in the original or `crs_output` CRS)

References

Pedreira Junior, J. U.; Louro, T. V.; Assis, L. B. M.; Brito, P. L.; Bomfim, F. G. (2026). BGBI: A citywide-referenced and bidirectional land use mix index for planning and policy evaluation. *Land Use Policy*, 169, 108135. https://doi.org/10.1016/j.landusepol.2026.108135

Pedreira Junior, J. U.; Louro, T. V.; Assis, L. B. M.; Brito, P. L. (2025). Measuring land use mix with address-level census data. *engrXiv* preprint. https://engrxiv.org/preprint/view/5975 (where the adapted HHI (`hhi_adp`) is documented)

Massey, D. S. (2001). The prodigal paradigm returns: ecology comes back to sociology. In A. Booth & A. C. Crouter (Eds.), *Does It Take a Village? Community Effects on Children, Adolescents, and Families*. Lawrence Erlbaum.

Song, Y.; Merlin, L.; Rodriguez, D. (2013). Comparing measures of urban land use mix. *Computers, Environment and Urban Systems*, 42, 1–13. https://doi.org/10.1016/j.compenvurbsys.2013.08.001

Examples

```
# Compute land-use mix indices on H3 hexagons
lumi <- compute_lumi(code_muni = 2929057, cache = FALSE)

# Compute land-use mix indices on user-provided polygons (neighborhoods of Lauro de Freitas-BA)
# Using geobr to download neighborhood boundaries
library(geobr)
nei_ldf <- subset(
  read_neighborhood(year = 2022),
  code_muni == 2919207
)
lumi_poly <- compute_lumi(
  code_muni = 2919207,
  polygon = nei_ldf,
  cache = FALSE
)
```

`read_cnefe` *Read CNEFE data for a given municipality*

Description

Downloads and reads the CNEFE CSV file for a given IBGE municipality code, using the official IBGE FTP structure. The function relies on an internal index linking municipality codes to the corresponding ZIP URLs. Data are returned either as an Arrow Table (default) or as an sf object with SIRGAS 2000 coordinates.

Usage

```
read_cnefe(
  code_muni = NULL,
  year = 2022,
  verbose = TRUE,
  cache = TRUE,
  cache_dir = NULL,
  output = c("arrow", "sf"),
  file = NULL
)
```

Arguments

`code_muni`	Integer. Seven-digit IBGE municipality code. Omit it when reading a local file through `file`.
`year`	Integer. The CNEFE data year. Currently only 2022 is supported. Defaults to 2022.

verbose — Logical; if TRUE, print informative messages about download, extraction, and reading steps.

cache — Logical; if TRUE, keep the downloaded data as a gzipped CSV in the cache folder (see cache_dir). If FALSE, a temporary file is used and removed after reading.

cache_dir — Character. Directory to use for cached downloads. If NULL (default), the CNEFETOOLS_CACHE_DIR environment variable is used when it is set, otherwise tools::R_user_dir() with which = "cache". Use this to point large downloads at a secondary drive or a shared volume.

output — Character. Output format. "arrow" (default) returns an arrow::Table, whereas "sf" returns an sf point object with coordinates built from LONGITUDE / LATITUDE in CRS 4674.

file — Character. Path to a CNEFE file already on disk, read instead of downloading. Accepts .zip as published by IBGE, .csv, .csv.gz and .parquet, which is what cnefe_export() writes. Mutually exclusive with code_muni, and it makes the function independent of the IBGE server.

Details

When output = "arrow" (default), the function does not perform any spatial conversion and simply returns the Arrow table. When output = "sf", the function converts the result to an sf point object using the LONGITUDE and LATITUDE columns, with CRS EPSG:4674 (SIRGAS 2000), keeping these columns in the final object (remove = FALSE).

Value

If output = "arrow", an arrow::Table containing all CNEFE records for the given municipality.

If output = "sf", an sf object with point geometry in EPSG:4674 (SIRGAS 2000), using the LONGITUDE and LATITUDE columns.

Caching

When cache = TRUE (the default), the municipality is downloaded once and kept as a gzipped CSV, so later calls, in this session or any other, reuse it instead of downloading again. The cache folder is cache_dir when given, otherwise the CNEFETOOLS_CACHE_DIR environment variable when set, otherwise tools::R_user_dir() with which = "cache".

When cache = FALSE, the file is stored in a temporary location and removed when the function exits.

The cache is meant to be cleared (see clear_cache_muni()). For a copy that stays put, use cnefe_export(). The "Cache and exported copies" article on the package website covers both.

See Also

cnefe_export() to write a municipality to a persistent, optimised file that this function can read back through file.

Examples

```
# Read CNEFE data as an Arrow table
cnefe <- read_cnefe(code_muni = 2929057, cache = FALSE)

# Read a local file instead, with no network access. overwrite = TRUE because
# cnefe_export() refuses to clobber an existing export by default.
path <- cnefe_export(2929057, path = tempdir(), cache = FALSE, overwrite = TRUE)
cnefe_local <- read_cnefe(file = path)

# Read as an sf spatial object
cnefe_sf <- read_cnefe(code_muni = 2929057, output = "sf", cache = FALSE)
```

---

tracts_to_h3 *Convert census tract aggregates to an H3 grid using CNEFE points*

Description

`tracts_to_h3()` performs a dasymetric interpolation with the following steps:

1. census tract totals are allocated to CNEFE dwelling points inside each tract;
2. allocated values are aggregated to an H3 grid at a user-defined resolution.

The function uses DuckDB with the spatial and H3 extensions for the heavy work.

Unlike `cnefe_counts()` and `compute_lumi()`, this function does not expose a `backend` argument and relies on DuckDB exclusively. The dominant cost here is a spatial overlay between the full CNEFE point set of the municipality and the census tract polygons, which is a different workload from the tabular aggregation those other functions perform. Running that overlay in R would take prohibitively long in medium and large municipalities, so a pure-R fallback would offer users a path that does not finish rather than a slower one.

Usage

```
tracts_to_h3(
  code_muni,
  year = 2022,
  h3_resolution = 9,
  vars = c("pop_ph", "pop_ch"),
  cache = TRUE,
  cache_dir = NULL,
  verbose = TRUE
)
```

Arguments

code_muni	Integer. Seven-digit IBGE municipality code.
year	Integer. The CNEFE data year. Currently only 2022 is supported. Defaults to 2022.
h3_resolution	Integer. H3 resolution (0 to 15). Defaults to 9.
vars	Character vector. Names of tract-level variables to interpolate. Supported variables: • pop_ph: population in private households (*Domicílios particulares*). • pop_ch: population in collective households (*Domicílios coletivos*). • male: total male population. • female: total female population. • age_0_4, age_5_9, age_10_14, age_15_19, age_20_24, age_25_29, age_30_39, age_40_49, age_50_59, age_60_69, age_70m: population by age group. • race_branca, race_preta, race_amarela, race_parda, race_indigena: population by race/color (*cor ou raça*). • n_resp: number of household heads (*Pessoas responsáveis por domicílios*). • avg_inc_resp: average income of the household heads. For a reference table mapping these variable names to the official IBGE census tract codes and descriptions, see tracts_variables_ref. Allocation rules: • pop_ph is allocated only to private dwellings. • pop_ch is allocated only to collective dwellings. • n_resp is allocated only to private dwellings (same rule as pop_ph). • Demographic variables (male, female, age_*, race_*) are allocated to private dwellings when the tract has any; if the tract has zero private dwellings but has collective dwellings, they are allocated to collective. • avg_inc_resp is assigned (not split) to each private dwelling point; tracts with no private dwellings receive no allocation.
cache	Logical. Whether to use the package cache for the census tract assets and the CNEFE files.
cache_dir	Character. Directory to use for cached downloads. If NULL (default), the CNEFETOOLS_CACHE_DIR environment variable is used when it is set, otherwise tools::R_user_dir() with which = "cache". Use this to point large downloads at a secondary drive or a shared volume.
verbose	Logical. Whether to print step messages and timing.

Value

An sf object (CRS 4326) with an H3 grid and the requested interpolated variables.

Examples

```
# Interpolate population to H3 hexagons
hex_pop <- tracts_to_h3(
  code_muni = 2929057,
  vars = c("pop_ph", "pop_ch"),
  cache = FALSE
)
```

`tracts_to_polygon` *Convert census tract aggregates to user-provided polygons using CNEFE points*

Description

`tracts_to_polygon()` performs a dasymetric interpolation with the following steps:

1. census tract totals are allocated to CNEFE dwelling points inside each tract;
2. allocated values are aggregated to user-provided polygons (neighborhoods, administrative divisions, custom areas, etc.).

The function uses DuckDB with spatial extensions for the heavy work.

Unlike `cnefe_counts()` and `compute_lumi()`, this function does not expose a `backend` argument and relies on DuckDB exclusively. The dominant cost here is a spatial overlay between the full CNEFE point set of the municipality and the census tract polygons, which is a different workload from the tabular aggregation those other functions perform. Running that overlay in R would take prohibitively long in medium and large municipalities, so a pure-R fallback would offer users a path that does not finish rather than a slower one.

Usage

```
tracts_to_polygon(
  code_muni,
  polygon,
  year = 2022,
  vars = c("pop_ph", "pop_ch"),
  crs_output = NULL,
  cache = TRUE,
  cache_dir = NULL,
  verbose = TRUE
)
```

Arguments

code_muni	Integer. Seven-digit IBGE municipality code.
polygon	An sf::sf object with polygon geometries (POLYGON or MULTIPOLYGON). The function will automatically align CRS and issue a warning reporting the percentage of the polygon area that falls outside the municipality.
year	Integer. The CNEFE data year. Currently only 2022 is supported. Defaults to 2022.
vars	Character vector. Names of tract-level variables to interpolate. Supported variables: • pop_ph: population in private households (*Domicílios particulares*). • pop_ch: population in collective households (*Domicílios coletivos*). • male: total male population. • female: total female population. • age_0_4, age_5_9, age_10_14, age_15_19, age_20_24, age_25_29, age_30_39, age_40_49, age_50_59, age_60_69, age_70m: population by age group. • race_branca, race_preta, race_amarela, race_parda, race_indigena: population by race/color (*cor ou raça*). • n_resp: number of household heads (*Pessoas responsáveis por domicílios*). • avg_inc_resp: average income of the household heads. For a reference table mapping these variable names to the official IBGE census tract codes and descriptions, see tracts_variables_ref. Allocation rules: • pop_ph is allocated only to private dwellings. • pop_ch is allocated only to collective dwellings. • n_resp is allocated only to private dwellings (same rule as pop_ph). • Demographic variables (male, female, age_*, race_*) are allocated to private dwellings when the tract has any; if the tract has zero private dwellings but has collective dwellings, they are allocated to collective. • avg_inc_resp is assigned (not split) to each private dwelling point; tracts with no private dwellings receive no allocation.
crs_output	The CRS for the output object. Default is NULL, which uses the original CRS of the polygon argument. Can be an EPSG code (e.g., 4326, 31983) or any CRS object accepted by sf::st_transform().
cache	Logical. Whether to use the package cache for the census tract assets and the CNEFE files.
cache_dir	Character. Directory to use for cached downloads. If NULL (default), the CNEFETOOLS_CACHE_DIR environment variable is used when it is set, otherwise tools::R_user_dir() with which = "cache". Use this to point large downloads at a secondary drive or a shared volume.
verbose	Logical. Whether to print step messages and timing.

Value

An sf object with the user-provided polygons and the requested interpolated variables. The output CRS matches the original polygon CRS (or crs_output if specified).

Examples

```
# Interpolate population to user-provided polygons (neighborhoods of Lauro de Freitas-BA)
# Using geobr to download neighborhood boundaries
library(geobr)
nei_ldf <- subset(
  read_neighborhood(year = 2022),
  code_muni == 2919207
)
poly_pop <- tracts_to_polygon(
  code_muni = 2919207,
  polygon = nei_ldf,
  vars = c("pop_ph", "pop_ch"),
  cache = FALSE
)
```

---

tracts_variables_ref *Reference table for tracts_to_* function variables*

Description

A data frame that maps variable names used in `tracts_to_h3()` and `tracts_to_polygon()` to the census tract variable codes and descriptions.

Usage

```
tracts_variables_ref
```

Format

A data frame with 22 rows and 4 columns:

var_cnefetools Variable name used in cnefetools functions.

code_var_ibge Variable code in the census tract aggregates, as used by censobr.

desc_var_ibge Official IBGE variable description in Portuguese.

table_ibge Census tract table where the variable is found (Domicilios, Pessoas, or ResponsavelRenda), which correspond to the censobr datasets Domicilio, Pessoas and ResponsavelRenda.

Details

The variable codes are the ones used by the censobr package, which repackages the IBGE census tract aggregates and is where the census tract assets take their attributes from (see `data-raw/sc_assets_build.R` in the package repository), and each table corresponds to a censobr dataset. They don't always match the file and column names on the IBGE FTP server.

Source

IBGE - Censo Demografico 2022, Agregados por Setores Censitarios, as repackaged by the censobr package.

Examples

```
# View the reference table
tracts_variables_ref

# Find the IBGE code for a specific variable
tracts_variables_ref[tracts_variables_ref$var_cnefetools == "pop_ph", ]
```

Index

∗ **datasets**
tracts_variables_ref, 18

arrow::Table, *13*

clear_cache_muni, 2
clear_cache_muni(), *8, 9*, *13*
clear_cache_tracts, 3
cnefe_counts, 4
cnefe_counts(), *2*, *11*, *14*, *16*
cnefe_dictionary, 7
cnefe_doc, 7
cnefe_export, 8
cnefe_export(), *13*
compute_lumi, 9
compute_lumi(), *2*, *6*, *14*, *16*

read_cnefe, 12
read_cnefe(), *9*

sf, *12, 13*
sf::sf, *5, 6*, *10, 11*, *17*
sf::st_transform(), *5*, *10*, *17*

Table, *12*
tools::R_user_dir(), *3–5*, *8*, *10*, *13*, *15*, *17*
tracts_to_h3, 14
tracts_to_h3(), *2, 3*, *18*
tracts_to_polygon, 16
tracts_to_polygon(), *3*, *18*
tracts_variables_ref, *15*, *17*, 18