Package 'ibger'

October 14, 2026

Title Access the 'IBGE' Aggregate Data API from 'R'

Version 0.2.1

Description 'Tidyverse'-friendly interface to the Brazilian Institute of Geography and Statistics ('IBGE') aggregate data 'API' <https://servicodados.ibge.gov.br/api/docs/agregados?versao=3>. Query aggregates, variables, localities, periods, and metadata from surveys and censuses conducted by 'IBGE'.

License MIT + file LICENSE

URL https://docs.ropensci.org/ibger/, https://github.com/ropensci/ibger

BugReports https://github.com/ropensci/ibger/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports httr2 (>= 1.0.0), curl (>= 6.0.0), cli (>= 3.6.0), tibble, purrr, dplyr, rlang, glue

Suggests knitr, rmarkdown, pkgdown, ggplot2, tidyr, shiny (>= 1.8.0), DT, bslib (>= 0.6.0), bsicons, rstudioapi, testthat (>= 3.0.0), httptest2, withr

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author André Leite [aut, cre] (ORCID: <https://orcid.org/0000-0002-4718-9766>), Marcos Wasiliew [aut] (ORCID: <https://orcid.org/0009-0004-4694-3159>), Hugo Vasconcelos [aut] (ORCID: <https://orcid.org/0000-0001-6249-0920>), Carlos Amorim [aut] (ORCID: <https://orcid.org/0000-0001-6315-8305>), Diogo Bezerra [aut] (ORCID: <https://orcid.org/0000-0002-1216-8674>), Allan Quadros [rev] (ORCID: <https://orcid.org/0000-0003-3250-5380>, Reviewed the package for rOpenSci, see <https://github.com/ropensci/software-review/issues/787>), Diana García-Cortés [rev] (ORCID:

<https://orcid.org/0000-0003-1562-7319>, Reviewed the package for rOpenSci, see <https://github.com/ropensci/software-review/issues/787>)

Maintainer André Leite <leite@castlab.org>

Repository CRAN

Date/Publication 2026-10-14 12:22:13 UTC

Contents

fetch_sidra_url . . . 2
ibge_aggregates . . . 3
ibge_clear_cache . . . 4
ibge_explorer . . . 5
ibge_localities . . . 6
ibge_metadata . . . 7
ibge_periods . . . 7
ibge_subjects . . . 8
ibge_surveys . . . 9
ibge_survey_metadata . . . 10
ibge_survey_periods . . . 12
ibge_variables . . . 13
parse_ibge_value . . . 15
parse_sidra_url . . . 16

Index **17**

`fetch_sidra_url` *Fetch data from a SIDRA API URL*

Description

Parses a SIDRA API URL and fetches the data using `ibge_variables()`, returning the same tidy tibble format.

Usage

```
fetch_sidra_url(url, validate = TRUE)
```

Arguments

`url`	Character string. A SIDRA API URL, typically starting with `https://apisidra.ibge.gov.br/values/`
`validate`	Logical. If `TRUE` (default), validates parameters against aggregate metadata before querying.

Value

A tibble in tidy (long) format, same as `ibge_variables()`.

Examples

```
url <- paste0(
  "https://apisidra.ibge.gov.br/values",
  "/t/7060/n1/all/v/63/p/last%2012/c315/7169"
)
fetch_sidra_url(url)

# Pipe-friendly: inspect then fetch
url |> parse_sidra_url()
url |> fetch_sidra_url()
```

`ibge_aggregates` *List IBGE aggregates*

Description

Retrieves the set of available aggregates (tables), grouped by survey. Each aggregate corresponds to a SIDRA table. Results are cached in memory per unique combination of parameters, so repeated calls with the same filters are instant.

Usage

```
ibge_aggregates(
  period = NULL,
  subject = NULL,
  classification = NULL,
  periodicity = NULL,
  level = NULL
)
```

Arguments

`period`	Period of interest, as a periodicity code followed by one or more period ids in brackets: `"P5[202001]"` (January 2020, monthly aggregates), `"P1[2019,2020]"` (2019 and 2020, annual aggregates).
`subject`	Numeric subject code (e.g. `70` for animal slaughter). Use `ibge_subjects()` to look up codes.
`classification`	Numeric classification code (e.g. `12026`).
`periodicity`	Periodicity code (see Details): `"P1"` (annual), `"P5"` (monthly), `"P8"` (semi-annual), `"P9"` (quarterly), `"P13"` (rolling quarter), etc.
`level`	Geographic level: `"N1"` (Brazil), `"N2"` (region), `"N3"` (state), `"N6"` (municipality), etc.

Details

All filters are optional. Their format is checked before the request: the IBGE API silently ignores filters it cannot parse (returning the whole catalog) or answers HTTP 500, so malformed values are rejected here with an informative error. A well-formed filter that matches no aggregate returns an empty tibble with a warning.

Periodicity codes used by the API (as observed in the catalog):

Code	Periodicity
P1	Annual
P5	Monthly
P7	Every three years
P8	Semi-annual
P9	Quarterly
P11	Every two years
P13	Rolling quarter (PNAD Contínua)
P16	Every six years

Value

A tibble with columns: `survey_id`, `survey_name`, `aggregate_id`, `aggregate_name`

Examples

```
ibge_aggregates()
ibge_aggregates(periodicity = "P5")
ibge_aggregates(level = "N6")
ibge_aggregates(subject = 70)
ibge_aggregates(period = "P5[202001]")
```

---

ibge_clear_cache *Clear metadata cache*

Description

Removes all cached metadata (aggregates and surveys), forcing fresh API calls on subsequent requests.

Usage

```
ibge_clear_cache()
```

Value

No return value, called for side effects.

Examples

```
ibge_clear_cache()
```

`ibge_explorer` *Interactive aggregate explorer*

Description

Launches a Shiny app for browsing and filtering the full catalog of IBGE aggregates. The app displays value boxes with summary counts, a searchable table with column-level filters, and lets you filter by survey or search aggregate names.

Usage

```
ibge_explorer(launch.browser = TRUE)
```

Arguments

`launch.browser`	Logical or function. If `TRUE` (the default), opens the app in the default browser. If `FALSE`, opens in the RStudio Viewer pane when running inside RStudio. You can also pass a function such as `shiny::paneViewer()`.

Details

Clicking a row shows a notification with the aggregate ID and the corresponding `ibge_metadata()` call. A CSV download button is also available.

Value

This function is called for its side effect (launching the app). Returns the value of `shiny::runApp()` invisibly.

Examples

```
# Open in RStudio Viewer
ibge_explorer(launch.browser = FALSE)

# Open in browser
ibge_explorer()
```

`ibge_localities` *Localities for an aggregate*

Description

Retrieves available localities for an aggregate at one or more geographic levels. Validates the requested level(s) against the aggregate metadata before querying.

Usage

```
ibge_localities(aggregate, level = "N6", validate = TRUE)
```

Arguments

`aggregate`	Numeric aggregate identifier.
`level`	Geographic level. Use `"N1"` (Brazil), `"N2"` (region), `"N3"` (state), `"N6"` (municipality), `"N7"` (metropolitan area), etc. For multiple levels, use a vector: `c("N6", "N7")`.
`validate`	Logical. If `TRUE` (default), validates level against aggregate metadata. Use `FALSE` to skip.

Details

When several levels are requested, one request is made per level and the results are combined. (The API's own multi-level endpoint returns an empty list when any of the requested levels has no localities for the aggregate, which would silently hide the other levels.)

Value

A tibble with columns: `id`, `name`, `level_id`, `level_name`

Examples

```
ibge_localities(1437, level = "N1")
ibge_localities(1437, level = c("N2", "N3"))
```

ibge_metadata *Metadata for an aggregate*

Description

Retrieves complete metadata for an aggregate: name, survey, subject, periodicity, territorial levels, variables, and classifications with all their categories.

Usage

```
ibge_metadata(aggregate)
```

Arguments

aggregate	Numeric aggregate identifier.

Value

A list of class `ibge_metadata` with:

- `id`, `name`, `url`, `survey`, `subject`
- `periodicity`: list with frequency, start and end
- `territorial_level`: list with administrative, special and ibge
- `variables`: tibble with id, name, unit
- `classifications`: tibble with id, name and list-column `categories`, where each element is a tibble with category_id, category_name, category_unit and category_level

Examples

```
meta <- ibge_metadata(7060)
meta$variables
meta$classifications
tidyr::unnest(meta$classifications, categories)
```

ibge_periods *Periods for an aggregate*

Description

Retrieves available periods for an aggregate.

Usage

```
ibge_periods(aggregate)
```

Arguments

aggregate	Numeric aggregate identifier.

Value

A tibble with columns: id, literal, modification

Examples

```
ibge_periods(1705)
```

---

ibge_subjects *IBGE subject codes lookup*

Description

Returns a tibble with IBGE subject (theme) codes and descriptions. These codes can be used to understand the thematic organization of IBGE surveys and aggregates.

Usage

```
ibge_subjects(pattern = NULL, ignore_case = TRUE)
```

Arguments

pattern	Optional character string. A regular expression used to filter subject descriptions. If NULL (default), returns the full table. Matching is case-insensitive by default.
ignore_case	Logical. If TRUE (default), pattern matching ignores case.

Details

The table is built-in (no API call is made) and contains all subject codes currently defined by IBGE.

Value

A tibble with columns:

id Integer subject code.

name Character subject description.

Examples

```
# All subjects
ibge_subjects()

# Find sanitation-related themes
ibge_subjects("saneamento|esgoto|lixo")

# Find themes mentioning "internet"
ibge_subjects("internet")
```

`ibge_surveys` *List IBGE surveys with metadata*

Description

Retrieves the full catalog of IBGE surveys (statistical operations) that have associated metadata in the IBGE Metadata API, including status, category, collection and publication frequency, and thematic classifications.

Usage

```
ibge_surveys(thematic_classifications = TRUE)
```

Arguments

`thematic_classifications`
: Logical. If `TRUE` (default), includes a list-column `thematic_classifications` with the classification details for each survey. Set to `FALSE` for a simpler flat tibble.

Details

This function queries a **different API** from the other `ibge_*` functions. While `ibge_aggregates()` and `ibge_metadata()` use the Aggregates API (v3), this function uses the Metadata API (v2), which provides institutional and methodological documentation about surveys.

Value

A tibble with columns:

id Survey code (e.g. `"SC"`, `"CD"`)

name Survey name in Portuguese

name_en Survey name in English (may be `NA`)

status Survey status: Ativa, Encerrada, etc.

category Conjuntural, Estrutural, or Especial

collection_frequency Data collection frequency (Mensal, Anual, etc.)

publication_frequency Publication frequency

thematic_classifications List-column of tibbles with thematic classification details (name, domain, description). Only if `thematic_classifications = TRUE`.

See Also

`ibge_survey_periods()`, `ibge_survey_metadata()`

Examples

```
# Full catalog
ibge_surveys()

# Flat table without classifications
ibge_surveys(thematic_classifications = FALSE)

# Filter active conjunctural surveys
library(dplyr)
ibge_surveys(thematic_classifications = FALSE) |>
  filter(status == "Ativa", category == "Conjuntural")
```

---

ibge_survey_metadata *Get survey metadata for a specific period*

Description

Retrieves detailed institutional and methodological metadata for a survey in a given reference period. Before querying, both the survey code and the year/month combination are validated against the IBGE catalog, so invalid inputs produce clear errors without wasting an API call.

Usage

```
ibge_survey_metadata(survey, year, month = NULL, order = 0)
```

Arguments

`survey`	Character string. The survey code (e.g. `"SC"`, `"CD"`). Must match a code from `ibge_surveys()`.
`year`	Integer. Reference year. Must be a year with available metadata for the survey (see `ibge_survey_periods()`).
`month`	Integer or `NULL`. Reference month (1–12). Required for conjunctural (sub-annual) surveys; omit for structural (annual+) surveys. If omitted for a conjunctural survey, a warning is issued listing the available months.
`order`	Integer. Publication order within the period (default `0`, the most recent).

Details

The structure of `occurrences` varies by survey and may include fields such as objective, data collection method, sample design, geographic scope, reference period, and more.

Value

A list of class `ibge_survey_metadata` with:

status Survey status (corrente, concluída, desativada, etc.)

category conjuntural or estrutural

type Type of statistical operation

area Responsible area

acronym Survey acronym

start_date Survey start date

deactivation_date Deactivation date, if applicable

sidra_url URL to SIDRA data

concla_url URL to CONCLA (CNAE classifications)

thematic_classifications Tibble of thematic classifications

occurrences List of metadata records for the period (structure varies by survey)

See Also

`ibge_surveys()`, `ibge_survey_periods()`

Examples

```
# Structural survey (no month needed)
ibge_survey_metadata("CD", year = 2022)

# Conjunctural survey (specify month)
ibge_survey_metadata("SC", year = 2023, month = 6)

# Inspect methodology fields
meta <- ibge_survey_metadata("CD", year = 2022)
names(meta$occurrences[[1]])

# Invalid code: clear error with suggestions
try(ibge_survey_metadata("PMS", year = 2024))
#> Error: Survey code "PMS" not found in the IBGE catalog.

# Invalid year: error with available range
try(ibge_survey_metadata("CD", year = 1800))
#> Error: Year 1800 not available for survey "CD".
#> i Available years: 1940 to 2022 (9 total).
```

`ibge_survey_periods` *List periods with metadata for a survey*

Description

Retrieves the periods (year/month combinations) for which a given survey has associated metadata records. Before querying, the survey code is validated against the IBGE catalog; invalid codes produce a helpful error with suggestions.

Usage

```
ibge_survey_periods(survey)
```

Arguments

`survey`	Character string. The survey code as returned by `ibge_surveys()` (e.g. `"SC"` for Pesquisa Mensal de Serviços, `"CD"` for Censo Demográfico). Invalid codes are caught before the request is sent, with suggestions for similar codes.

Value

A tibble with columns:

year Integer year with metadata available.

month Integer month (1–12) or `NA` for structural (annual or longer) surveys.

order Publication order within the period (0 = most recent).

See Also

`ibge_surveys()`, `ibge_survey_metadata()`

Examples

```
# Conjunctural survey (monthly periods)
ibge_survey_periods("SC")   # Pesquisa Mensal de Serviços

# Structural survey (annual periods)
ibge_survey_periods("CD")   # Censo Demográfico

# Invalid code: helpful error with suggestions
try(ibge_survey_periods("PMS"))
#> Error: Survey code "PMS" not found in the IBGE catalog.
#> i Did you mean one of these?
#>   * SC - Pesquisa Mensal de Serviços
#>   ...
```

ibge_variables *Get variable data from an aggregate*

Description

Main function of the package. Retrieves variable results from an IBGE aggregate for the specified localities, periods and classifications.

Usage

```
ibge_variables(
  aggregate,
  variable = NULL,
  periods = -6,
  localities = "BR",
  classification = NULL,
  view = NULL,
  validate = TRUE,
  chunk = TRUE
)
```

Arguments

aggregate	Numeric aggregate identifier (SIDRA table).
variable	Variable(s) to retrieve. Can be: • NULL (default): returns all standard variables • Numeric vector: specific IDs, e.g. c(284, 285) • "all": includes automatically generated percentage variables
periods	Period(s) to query. Can be: • Negative integer: last N periods, e.g. -6 (default) • Numeric vector: e.g. c(201701, 201702, 201703) • String with range: e.g. "201701-201712"
localities	Locality(ies) to query. Can be: • "BR" (default): Brazil • Level code: "N3" (all states), "N6" (all municipalities) • Named list for specific localities: list(N3 = c(33, 35)) (RJ and SP states), list(N6 = c(3550308, 3304557)) (SP and RJ municipalities)
classification	Classification(s) to filter results. Named list where names are classification IDs and values are category ID vectors. Use "all" for all categories. E.g. list("226" = c(4844, 96608), "218" = 4780)
view	Display mode: NULL (default), "OLAP" or "flat".
validate	Logical. If TRUE (default), validates parameters against aggregate metadata before querying. Use FALSE to skip.

chunk — Controls automatic splitting of large queries. The IBGE API rejects requests whose result is too large with an HTTP 500 error (the documented limit is 100,000 values, but in practice requests fail above about 50,000). Can be:

- `TRUE` (default): estimates the result size (variables x periods x localities x categories) and, when it exceeds 50,000 values, transparently splits the query into multiple smaller requests (by periods, then by localities) and combines the results.
- `FALSE`: always performs a single request.
- Positive number: same as `TRUE`, but with a custom per-request value limit.

Details

Before querying the API, validates all parameters against the aggregate metadata. If any parameter is invalid, stops with a clear error message showing the allowed values.

Note that large requests can be slow on the server side: queries covering all municipalities (`localities = "N6"`, some 5,500 localities) may take one to several minutes *per request* to be answered by the IBGE API, even before any chunking. This is normal — the function is not hanging.

Value

A tibble in tidy (long) format with columns: `variable_id`, `variable_name`, `variable_unit`, classification columns (when present), `locality_id`, `locality_name`, `locality_level`, `period`, `value`

Examples

```
# IPCA in Brazil
ibge_variables(7060, localities = "BR")

# Specific variables for states (quarterly animal slaughter)
ibge_variables(1092, variable = c(284, 285), localities = "N3")

# Large query (all municipalities): split automatically into
# multiple requests to respect the API's 100,000 value limit
ibge_variables(1612, localities = "N6")

# Specific municipalities with classification
ibge_variables(
  aggregate      = 1712,
  variable       = 214,
  periods        = -3,
  localities     = list(N6 = c(3550308, 3304557)),
  classification = list("226" = c(4844, 96608))
)
```

`parse_ibge_value` *Parse IBGE value column*

Description

Converts the character `value` column returned by `ibge_variables()` to numeric, handling IBGE special value conventions.

Usage

```
parse_ibge_value(x)
```

Arguments

`x`	A character or numeric vector of IBGE values.

Details

According to IBGE's tabular presentation standards, the value column may contain special character codes instead of numbers:

Code	Meaning
`-`	Numeric value equal to zero (not from rounding)
`..`	Not applicable
`...`	Data not available
`X`	Suppressed to avoid identifying individual respondents

Value

A numeric vector where:

- `"-"` becomes `0`
- `".."`, `"..."` and `"X"` become `NA_real_`
- All other values are converted with `as.numeric()`

Examples

```
parse_ibge_value(c("1.5", "10", "-", "..", "...", "X", NA))
#> [1] 1.5  10.0  0.0   NA    NA    NA    NA


# Typical usage after ibge_variables()
library(dplyr)

ibge_variables(7060, localities = "BR") |>
  mutate(value = parse_ibge_value(value))
```

parse_sidra_url *Parse a SIDRA API URL into ibger parameters*

Description

Converts a SIDRA API URL (from the SIDRA Query Builder or sidrar package) into a human-readable breakdown of its parameters, enriched with names from the aggregate metadata.

Usage

```
parse_sidra_url(url)
```

Arguments

url	Character string. A SIDRA API URL, typically starting with https://apisidra.ibge.gov.br/values/

Value

A list of class sidra_query with:

- aggregate: list with id and name
- variables: tibble with id and name
- periods: character vector of period codes
- localities: list of level/locality pairs
- classifications: list of classification/category details
- ibger_call: string with the equivalent ibge_variables() call

Examples

```
url <- paste0(
  "https://apisidra.ibge.gov.br/values",
  "/t/5434/n1/all/v/4090/p/last%201/c888/47946,56623"
)
parse_sidra_url(url)
```

Index

as.numeric(), *15*

fetch_sidra_url, 2

ibge_aggregates, 3
ibge_clear_cache, 4
ibge_explorer, 5
ibge_localities, 6
ibge_metadata, 7
ibge_periods, 7
ibge_subjects, 8
ibge_subjects(), *3*
ibge_survey_metadata, 10
ibge_survey_metadata(), *10*, *12*
ibge_survey_periods, 12
ibge_survey_periods(), *10, 11*
ibge_surveys, 9
ibge_surveys(), *10–12*
ibge_variables, 13
ibge_variables(), *2*, *15*

parse_ibge_value, 15
parse_sidra_url, 16

shiny::paneViewer(), *5*
shiny::runApp(), *5*

tibble, *2*, *4*, *6*, *8, 9*, *12*, *14*