Package: knbsdata (via r-universe)

November 25, 2026

Title Discover, Retrieve and Cite Official Kenyan Statistics

Version 0.5.0

Description Provides a consistent interface for discovering and retrieving official data and statistical products of the Kenya National Bureau of Statistics (KNBS): its survey archive <https://statistics.knbs.or.ke/nada/>, publications library, census, consumer price and economic data portals <https://opendata.knbs.or.ke/beta/databrowser/>, the open FinAccess survey microdata <https://finaccess.knbs.or.ke/>, and the Kenya Demographic and Health Survey indicators published through the DHS Program API <https://api.dhsprogram.com/>. Standardises them into documented schemas, preserves provenance metadata such as release dates and checksums, and generates reproducible citations. This is an independent open-source project and is not an official KNBS product.

License MIT + file LICENSE

URL https://github.com/Mustapha-Wasseja/knbsdata, https://Mustapha-Wasseja.github.io/knbsdata/

BugReports https://github.com/Mustapha-Wasseja/knbsdata/issues

Depends R (>= 4.1)

Imports cli, digest, httr2 (>= 1.0.0), jsonlite, lifecycle, readxl, rlang (>= 1.1.0), tibble, tools, utils, vctrs, yaml

Suggests covr, dplyr, haven, knitr, rmarkdown, spelling, testthat (>= 3.0.0), tidyr, withr, writexl

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

NeedsCompilation no

Author Mustapha Wasseja Mohammed [aut, cre], Pascal Ochieng Owiti [aut] (affiliation: Kenya National Bureau of Statistics)

Maintainer Mustapha Wasseja Mohammed <muswaseja@gmail.com>

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

Date/Publication 2026-11-25 17:28:34 UTC

RemoteUrl https://github.com/cran/knbsdata

RemoteRef HEAD

RemoteSha fc46040ca8d090891caa37ce0bfc7061f4907353

Contents

knbs_abstract_table . . . 3
knbs_abstract_tables . . . 4
knbs_cache_clear . . . 5
knbs_cache_dir . . . 5
knbs_cache_info . . . 6
knbs_cache_prune . . . 7
knbs_cite . . . 7
knbs_county_code . . . 8
knbs_download_finaccess . . . 9
knbs_download_microdata . . . 11
knbs_download_publication . . . 12
knbs_finaccess_files . . . 13
knbs_geographies . . . 15
knbs_get_indicator . . . 16
knbs_indicators . . . 18
knbs_match_counties . . . 19
knbs_metadata . . . 20
knbs_option . . . 21
knbs_options . . . 21
knbs_publications . . . 23
knbs_read_microdata . . . 24
knbs_release_series . . . 25
knbs_releases . . . 26
knbs_retail_prices . . . 27
knbs_search . . . 29
knbs_search_surveys . . . 30
knbs_sources . . . 31
knbs_status . . . 32
knbs_survey_files . . . 33
knbs_survey_metadata . . . 34
knbs_variables . . . 35
knbs_write . . . 36

Index **39**

knbs_abstract_table *Retrieve one Statistical Abstract table as tidy observations*

Description

[Experimental]

Downloads the table's chapter workbook from the KNBS website (cached locally), extracts the table by its curated recipe, verifies the layout anchors and pinned reference values, and returns one row per cell of the printed table.

Usage

```
knbs_abstract_table(table, cache = knbs_option("cache"), refresh = FALSE)
```

Arguments

table	A table id from knbs_abstract_tables() (for example "2.2"), or enough of a table title to identify one unambiguously.
cache	Whether to use the local cache.
refresh	Whether to re-download the workbook and re-extract.

Details

period is the integer year. status carries the source's own year flags, "provisional" (*) or "revised" (+), and is NA for final years. Sectioned tables (the production accounts, the income accounts) carry their section as an extra column named for what it is — industry or account. A row the source prints without figures (such as "Less: Subsidies" in the production accounts) is kept with NA values, because that is what the source shows.

Tables that print totals beside their parts (GDP by activity, expenditure on GDP, capital formation) carry a parent column naming the row each item is a component of: industries have "All industries" (or "All industries at basic prices") as their parent, sub-industries have their industry, and headline totals and memo items have NA. Summing rows that share a parent never double counts; summing the whole table does. For level and share tables the parts are checked to add up to their parent in every year, within rounding. Imports enter GDP with a negative sign, and the metadata says so. The accounts tables (2.1, 2.8, 2.9) are linked by accounting identities rather than parts and totals, and have no parent column.

County tables (construction by county, school enrolment, registered births and deaths) carry geo_code, each row's census frame code, so they join knbs_geographies(), the census tables and anything passed through knbs_county_code(); their total row is "00". Such a table is refused if its rows no longer account for the 47 counties exactly once. Tables that print a unit per row (production of selected commodities) carry it in a unit column. The sheets' footnotes and source lines are in knbs_metadata()'s notes.

The extraction is deliberately strict: if the workbook's layout, sections or pinned reference values no longer match what was verified at curation time, the function stops with knbs_error_schema_changed or knbs_error_validation_failed rather than guessing — wrong tidy data would be worse than an honest error.

Value

A knbs_abstract_table tibble. knbs_metadata() returns its provenance, including the workbook file and its SHA-256; knbs_cite() cites it.

See Also

knbs_abstract_tables()

Examples

```
gdp <- knbs_abstract_table("2.2")
expenditure <- knbs_abstract_table("expenditure on the gross domestic")
```

---

knbs_abstract_tables *List the curated Statistical Abstract tables*

Description

[Experimental]

The Statistical Abstract publishes its chapter tables as Excel workbooks on the KNBS website. This function lists the tables the package has *curated*: each one has a hand-verified extraction recipe — sheet, layout anchors, section labels, unit and pinned reference values — so that knbs_abstract_table() can return it as clean tidy observations, or refuse with a clear error if the workbook has changed.

Curated so far, from the 2025 edition: all fifteen tables of Chapter 2 (National Accounts) — the production accounts, GDP by activity at current and constant prices, expenditure on GDP, capital formation, national income and saving — and 33 tables from twelve more chapters: wage employment, insurance and savings, exports, crop sales and prices, forests and wildlife, electricity and fuel, producer prices and industrial production, construction, tourism, transport, education enrolment, and registered births and deaths. Six are county tables.

Usage

```
knbs_abstract_tables()
```

Value

A knbs_catalogue tibble with columns table_id, title, chapter, unit, sheet and workbook.

See Also

knbs_abstract_table(), knbs_publications()

Examples

```
knbs_abstract_tables()
```

knbs_cache_clear *Empty the cache*

Description

Removes every cached response, and the Statistical Abstract workbooks the cache holds.

Usage

```
knbs_cache_clear(quiet = knbs_option("quiet"))
```

Arguments

quiet	Suppress the summary message.

Value

The number of entries removed, invisibly.

See Also

knbs_cache_prune() to remove only expired entries.

Examples

```
# A throwaway cache, so the example leaves yours alone
op <- options(knbsdata.cache_dir = tempfile("knbs-cache-"))
knbs_cache_clear()
options(op)
```

knbs_cache_dir *Where knbsdata stores cached responses*

Description

The cache lives in tools::R_user_dir() unless the knbsdata.cache_dir option points elsewhere. Nothing is written until something is cached, and the cache looks after itself: the first write of each session removes entries that have expired, entries an earlier version of the package wrote, and Statistical Abstract workbooks no longer in use. Set options(knbsdata.cache = FALSE) to cache nothing at all, or point knbsdata.cache_dir at tempdir() to keep the cache for one session only.

Usage

```
knbs_cache_dir(create = FALSE)
```

Arguments

`create`	Whether to create the directory if it does not exist.

Value

The cache directory path, invisibly when `create = TRUE`.

See Also

`knbs_cache_info()`, `knbs_cache_clear()`, `knbs_cache_prune()`

Examples

```
knbs_cache_dir(create = FALSE)
```

---

knbs_cache_info *What is in the cache*

Description

What is in the cache

Usage

```
knbs_cache_info()
```

Value

A tibble with one row per cached entry: `cache_key`, `source_id`, `operation`, `resource_id`, `retrieved_at`, `expires_at`, `expired` and `size_bytes`. Zero rows if nothing is cached.

See Also

`knbs_cache_dir()`, `knbs_cache_clear()`

Examples

```
# A throwaway cache, so the example leaves yours alone
op <- options(knbsdata.cache_dir = tempfile("knbs-cache-"))
knbs_cache_info()
options(op)
```

knbs_cache_prune *Remove expired cache entries*

Description

Remove expired cache entries

Usage

```
knbs_cache_prune(quiet = knbs_option("quiet"))
```

Arguments

quiet	Suppress the summary message.

Value

The number of entries removed, invisibly.

See Also

knbs_cache_clear() to remove everything.

Examples

```
# A throwaway cache, so the example leaves yours alone
op <- options(knbsdata.cache_dir = tempfile("knbs-cache-"))
knbs_cache_prune()
options(op)
```

knbs_cite *Cite a knbsdata object*

Description

Builds a citation from the object's provenance metadata. Where the source supplies its own required citation, that text is used and the package adds only the retrieval information.

Usage

```
knbs_cite(
  x,
  style = c("plain", "bibtex", "ris"),
  include_retrieval_date = TRUE,
  ...
)
```

Arguments

x	A knbsdata object or a knbs_metadata object.
style	"plain", "bibtex" or "ris".
include_retrieval_date	Whether to record when the data were retrieved. Retrieval date matters for reproducibility whenever a "latest" release was requested.
...	Passed to methods.

Value

A knbs_citation object; print it, or use format() to get a string.

See Also

knbs_metadata()

Examples

```
# Any object knbsdata returns can be cited; this catalogue ships with it.
files <- knbs_finaccess_files(year = 2024, content = "data")
knbs_cite(files)
knbs_cite(files, style = "bibtex")
```

---

knbs_county_code *Official county codes for county names written any way*

Description

[Experimental]

Turns county names — or county numbers — as they appear in anyone's data into the official two-digit codes of the 2019 census frame, the codes knbs_geographies() and the census tables use. Built for mutate():

```
my_data |> mutate(geo_code = knbs_county_code(county))
```

KNBS's own sources disagree on names: the census writes "Murang'a", "Tharaka-Nithi" and "Nairobi City", the Data Portal "MURANGA", "THARAKA NITHI" and "NAIROBI". Both, and the usual variants elsewhere, resolve to the same codes.

Usage

```
knbs_county_code(
  x,
  vintage = 2019L,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

x	County names or numbers: a character, numeric or factor vector.
vintage	The census frame to match against. Only 2019 is available.
cache	Whether to use the local cache for the reference names.
refresh	Whether to re-read the reference names.

Details

A name matches when it is the official name once case, punctuation, spacing and apostrophes are set aside and the words "County", "City", "Government", "of" and "the" are dropped: "Taita Taveta", "TAITA/TAVETA" and "Taita-Taveta County" are all `"06"`, "Nairobi" and "Nairobi City County" both `"47"`. A number from 1 to 47, written 1, `"1"` or `"01"`, is taken as the county number. "Kenya" and "National" give the census code for the whole country, `"00"`.

Nothing else is matched. There is no list of alternative names, so a misspelling stays unmatched rather than being guessed at; the warning names each unmatched value and, where one official name is close, says which. `knbs_match_counties()` shows how every value was matched.

The official names are read from the census frame through `knbs_geographies()`, once, and cached. If the census service cannot be reached, the 47 counties' statutory names and numbers built into the package are used instead; they are the census frame's own, and the package's tests hold them identical.

Value

A character vector the length of `x`, holding two-digit codes, and `NA` where a value did not match.

See Also

`knbs_match_counties()`, `knbs_geographies()`

Examples

```
knbs_county_code(c("Murang'a", "MURANGA", "Nairobi", "taita/taveta", "47"))
#> [1] "21" "21" "47" "06" "47"
```

---

knbs_download_finaccess

Download FinAccess survey files

Description

[Experimental]

Downloads files listed by knbs_finaccess_files() from the FinAccess portal, under their official names, and proves they are the files the package catalogued. Before a transfer, a sixteen-byte request confirms the file's served name and size, so a file that has changed is refused before anything large moves. After it, the size is checked again and the SHA-256 must equal the catalogued one. A <file>.metadata.json sidecar beside each file records where it came from, the producers and their terms, so that knbs_read_microdata() and knbs_cite() can say so later.

Usage

```
knbs_download_finaccess(
  file,
  destination,
  overwrite = FALSE,
  verify = TRUE,
  quiet = knbs_option("quiet")
)
```

Arguments

file	File ids from knbs_finaccess_files(), such as "hh2024-stata", or that function's result.
destination	Directory to write into; created if needed. There is no default: nothing is written anywhere you have not named.
overwrite	Whether to replace a file already in destination.
verify	Whether to hold each file to the catalogue. FALSE downloads whatever the link serves now, under the name it is served with: for when the producers have revised a file since the catalogue was made. A web page is refused either way.
quiet	Whether to suppress messages and the progress bar.

Details

Terms. The portal states no licence. FSD Kenya's terms for the 2013 survey on Harvard Dataverse restrict the data to statistical and scientific research, forbid redistributing it without FSD Kenya's written agreement, rule out investigating specific individuals or organisations, and ask publications to cite the source. Do not assume wider reuse is allowed: ask the producers first. The package never bundles these files. The first download in a session says so.

Already downloaded. A file already in destination is not downloaded again if it is byte-identical to the catalogued one; its record is returned instead. That makes knbs_read_microdata(knbs_download_finaccess("hh20 "data")) safe to re-run. A file there that differs is left alone, and the call stops unless overwrite = TRUE.

Large files. The 2024 files are 160 to 290 MB. Downloads stream to disk and show progress unless quiet. They are not held to the package's request timeout; a transfer is abandoned only once it has stalled for that long. Google Drive serves a file too large for its virus scan after a notice saying it could not be scanned; the package requests the file directly, as the notice's own download button does, and relies on the checks above.

Registration. Asking for the 2013 survey's Stata copy, which Harvard Dataverse holds behind a registration form, raises `knbs_error_authentication_required` with the page to visit. knbsdata does not automate sign-in, licence acceptance or approval workflows.

Value

Invisibly, a tibble with one row per file: `resource_id`, `file_id`, `original_name`, `local_path`, `download_url`, `downloaded_at`, `size_bytes`, `sha256`, `content_type`, `package_version` and `source_id`. Pass it to `knbs_read_microdata()`.

See Also

`knbs_finaccess_files()`, `knbs_read_microdata()`

Examples

```
dir <- file.path(tempdir(), "finaccess")

# The 2024 household survey, Stata format (160 MB)
fa <- knbs_read_microdata(knbs_download_finaccess("hh2024-stata", dir))

# The questionnaires and manuals
knbs_finaccess_files(content = c("questionnaire", "manual")) |>
  knbs_download_finaccess(dir)
```

`knbs_download_microdata`

Download public-use microdata

Description

[Experimental]

The Kenya National Data Archive gates every download behind an account, and this package does not automate sign-in, licence acceptance or approval workflows, so the function reports what is required and stops.

Usage

```
knbs_download_microdata(
  survey,
  files = NULL,
  destination,
  overwrite = FALSE,
  verify = TRUE,
  manifest = TRUE,
  quiet = FALSE
)
```

Arguments

survey	A study identifier.
files	Files to download.
destination	Directory to write to. Unused: nothing is downloaded.
overwrite	Whether to overwrite existing files.
verify	Whether to verify downloads.
manifest	Whether to write a download manifest.
quiet	Whether to suppress progress messages.

Details

It exists so that the refusal is explicit and points at the official process, rather than the operation appearing to be unsupported by oversight.

Microdata that their producers publish for anyone are another matter: the FinAccess household and MSE Tracker surveys are listed by `knbs_finaccess_files()` and downloaded by `knbs_download_finaccess()`. Files obtained through the archive's own process can be read with `knbs_read_microdata()`.

Value

Never returns; raises `knbs_error_authentication_required`.

See Also

`knbs_survey_files()` to see what a study holds and on what terms; `knbs_finaccess_files()` for open microdata.

Examples

```
# Raises an error explaining the official access process; contacts nothing.
try(knbs_download_microdata("KEN-KNBS-KIHBS-2015-2016-v01"))
```

---

knbs_download_publication

Download a publication and verify what arrived

Description

[Experimental]

Downloads one publication file, preserving its official filename, verifying that a document (not an error page) arrived, checksumming it, and writing a `<file>.metadata.json` provenance sidecar next to it.

Usage

```
knbs_download_publication(
  publication_id,
  destination,
  overwrite = FALSE,
  verify = TRUE,
  sidecar = TRUE,
  source = NULL,
  quiet = knbs_option("quiet")
)
```

Arguments

publication_id	A publication_id from knbs_publications().
destination	Directory to write into; created if needed. There is no default: nothing is written anywhere you have not named.
overwrite	Whether to replace an existing file of the same name.
verify	Whether to refuse a response that is a web page rather than a document.
sidecar	Whether to write the provenance sidecar.
source	Source id.
quiet	Whether to suppress the progress message.

Value

Invisibly, a one-row tibble describing the download: local_path, download_url, downloaded_at, size_bytes, sha256 and the identifying fields.

See Also

knbs_publications() to find identifiers, knbs_cite() to cite what you downloaded.

Examples

```
pubs <- knbs_publications("economic survey", year = 2026)
knbs_download_publication(pubs$publication_id[[1]], tempdir())
```

---

knbs_finaccess_files *List the FinAccess survey files*

Description

[Experimental]

The FinAccess household survey, Kenya's survey of financial inclusion run by the Central Bank of Kenya, KNBS and FSD Kenya, publishes its anonymised, weighted microdata openly, with no account needed: six rounds from 2006 to 2024, and two waves of the FinAccess MSE Tracker Survey of micro and small enterprises (2022 and 2023). This lists every file, with the questionnaires, manuals and reports, from a catalogue kept with the package; nothing is contacted.

Download files with `knbs_download_finaccess()` and read them with `knbs_read_microdata()`.

Usage

```
knbs_finaccess_files(year = NULL, survey = NULL, content = NULL, format = NULL)
```

Arguments

`year`	Survey years to keep.
`survey`	`"household"` or `"mse_tracker"`.
`content`	Any of `"data"`, `"questionnaire"`, `"manual"` and `"report"`.
`format`	Any of `"stata"`, `"spss"`, `"excel"` and `"pdf"`. For an archive, the format of the data inside it.

Details

The catalogue records what each link on the portal actually serves, which is not always what its label says: the MSE Tracker Wave II "SPSS" link serves the Stata file and the reverse, the Wave I "Stata" link serves the SPSS file, and the 2006 and 2009 "Questionnaire" archives hold the survey data too. The `notes` column says where a label was corrected. The portal's 2019 heading lists no files.

The 2013 survey's Stata copy is held on Harvard Dataverse behind a registration form; it is listed with `access = "registration"` and never downloaded. Its SPSS copy is open.

Value

A `knbs_catalogue` tibble with one row per file: `file_id`, `title`, `survey`, `year`, `wave`, `content`, `format`, `file_name`, `size_bytes`, `sha256`, `archive`, `access`, `url` and `notes`.

See Also

`knbs_download_finaccess()`, `knbs_read_microdata()`

Examples

```
knbs_finaccess_files()
knbs_finaccess_files(year = 2024, content = "data")
knbs_finaccess_files(survey = "mse_tracker")
```

knbs_geographies *Kenya's geographic reference codes*

Description

[Experimental]

Returns the coded territorial hierarchy used by the 2019 Population and Housing Census: Kenya, its 47 counties, and their sub-counties, with names and parent codes. These are the join keys the census tables themselves use, read live from the same source (CC BY 4.0), never shipped as a snapshot that could drift from it.

Usage

```
knbs_geographies(
  level = NULL,
  vintage = NULL,
  parent = NULL,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

level	Keep only these levels: "national", "county", "subcounty".
vintage	Reference frame year. Only 2019 is available; the argument exists so that code written today keeps meaning the same thing if other vintages arrive later.
parent	Keep only units whose parent_code is one of these, for example "01" for Mombasa's sub-counties.
source	Source id.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Details

Codes are character and keep their leading zeros: "00" is Kenya, two digits are counties, four digits are sub-counties, and a sub-county's first two digits are its county. vintage is 2019 throughout — these are the census-frame units, and boundaries or units may have changed since. No other vintage, and no crosswalk to other coding systems, is offered, because none has been verified.

Value

A knbs_geography tibble with columns geo_code, geo_name, geo_level, parent_code, parent_name, vintage and source_id.

See Also

knbs_get_indicator(), whose observations carry the same codes.

Examples

```
counties <- knbs_geographies(level = "county")
knbs_geographies(parent = "01")
```

---

knbs_get_indicator *Retrieve an indicator as tidy observations*

Description

[Experimental]

Returns one row per observation, in one schema whichever source serves the indicator: a census table, a consumer price series, a Data Portal dataflow, a republished dataset or a DHS Program survey indicator.

Usage

```
knbs_get_indicator(
  indicator,
  start = NULL,
  end = NULL,
  geo_level = NULL,
  geo_code = NULL,
  frequency = NULL,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

indicator	An indicator id (see knbs_indicators()), or enough of an indicator name to identify one unambiguously.
start, end	Keep only observations in this period range. A year, a "YYYY-MM" month, or a date.
geo_level	Keep only these levels. Census tables: "national", "county", "subcounty". CPI series: "national", "region". DHS Program indicators: "national", "region", "county".
geo_code	Keep only these codes, as character, in the indicator's own geography.

frequency	Keep only series of this frequency: "annual", "quarterly" or "monthly". Only the Open Data for Africa source accepts it — its datasets may carry annual and sub-annual series side by side, and a mixed dataset asks you to choose rather than picking one silently.
source	Source id, to disambiguate an indicator served by more than one source.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Details

The period convention follows the source's own frequency. Monthly series (the CPI family) carry period as a Date at the first of the month plus a period_label like "2026-06"; census tables carry the census year as an integer period. start and end accept a year (2020), a month ("2020-01"), or a full date, and apply to both conventions.

Geography differs by family and the two systems must not be mixed. Census tables use territory codes (geo_level of "national", "county", "subcounty") that join knbs_geographies(). The CPI series use the CPI collection strata (geo_level of "national" or "region": the Nairobi income bands, Rest of Urban, and the former provinces) — their codes are the source's own and must never be joined against census territories. Six listed CPI regions, the former provinces, carry no data in any period; asking for them returns zero rows because that is the source's answer.

For the CPI family the default is the national series; ask for geo_level = "region" or name geo_codes for more. The CPI portal states no licence and no index base period; knbs_metadata() on the result says so. If the CPI Explorer is down, the error names the KNBS Data Portal series that stands in for the one asked for, and how closely: the portal's index has matched it, its rounded inflation rates have not always.

DHS Program indicators (source = "dhs_program") come with a row for every survey that measured them, 1989 to 2022, with period the survey year and survey naming it ("2022 DHS", "2020 MIS"). characteristic_type and characteristic say which group a value describes: the "Total", a background characteristic such as "Wealth quintile" or "Residence" (for Kenya as a whole), or a "Region". Region rows are the former provinces, or malaria zones, at geo_level "region" with no code; from 2014 the 47 counties are geo_level "county" with 2019 census codes, so they join the census tables. Where the DHS Program gives a value for more than one reference period, every value is kept and preferred marks the one it prefers: keep preferred rows for one value per survey and group.

Every other dimension of a table becomes a pair of columns, <name> holding the source's labels and <name>_code its codes, the way geo_name and geo_code pair up. Census tables use short curated names (marital_status, age, religion, ...); Data Portal flows use their SDMX dimension id in lower case (na_indic, commodity); Open Data for Africa datasets use the portal's dimension id (table, indicator). A Sex dimension is always sex, in lower case. The census age table mixes single years with five-year groups, so it carries age_type ("total", "single_year", "age_group", "other"): keep one type before summing.

Value

A knbs_indicator tibble. knbs_metadata() returns its provenance; knbs_cite() cites it.

See Also

knbs_indicators(), knbs_geographies()

Examples

```
cpi <- knbs_get_indicator("CPI_ALL_ITEMS", start = "2020-01")
inflation <- knbs_get_indicator("CPI_INFLATION_YOY", geo_level = "region")
pop <- knbs_get_indicator("POPCEN0001", geo_level = "county")
gcp <- knbs_get_indicator("jhbzkjc") # Gross County Product
stunting <- knbs_get_indicator("CN_NUTS_C_HA2", geo_level = "county") # KDHS
```

---

knbs_indicators *List available aggregate indicators*

Description

[Experimental]

Lists the indicators served by every verified indicator source: the 2019 Population and Housing Census tables (CC BY 4.0), the monthly consumer price series from the KNBS CPI Explorer, the KNBS Data Portal's SDMX dataflows, the datasets republished by Kenya Open Data for Africa, and the Kenya Demographic and Health Survey and Malaria Indicator Survey indicators published by the DHS Program. The `source_id` column says which source serves each row.

Usage

```
knbs_indicators(
  topic = NULL,
  search = NULL,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

`topic`	Keep only these topics, for example `"Population"` or `"prices"`.
`search`	Free text matched against indicator names.
`source`	Source id. Defaults to every verified source that lists indicators.
`cache`	Whether to use the local cache.
`refresh`	Whether to bypass any cached response.

Details

`frequency` and `unit` are filled only where the source states them: the CPI series are monthly, in index points or percent; a census table is not a recurring series, so its `frequency` and `unit` are `NA` rather than invented.

If one source cannot be reached, its rows are missing and a warning says so; the error comes only when no source answers.

Value

A `knbs_catalogue` tibble with columns `indicator_id`, `indicator_name`, `topic`, `frequency`, `unit` and `source_id`.

See Also

`knbs_get_indicator()`

Examples

```
knbs_indicators()
knbs_indicators(topic = "prices")
knbs_indicators(search = "marital")
```

---

knbs_match_counties *See how county names were matched*

Description

[Experimental]

The diagnostic companion to `knbs_county_code()`: one row per distinct value of `x`, saying which county it matched, how, and — for a value that matched nothing — the official name it most resembles, if any.

Usage

```
knbs_match_counties(
  x,
  vintage = 2019L,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

`x`	County names or numbers: a character, numeric or factor vector.
`vintage`	The census frame to match against. Only 2019 is available.
`cache`	Whether to use the local cache for the reference names.
`refresh`	Whether to re-read the reference names.

Value

A tibble with columns `input`, `geo_code`, `geo_name` (the official name), `matched_by` (`"name"`, `"number"`, `"national"`, or `NA`) and `suggestion` (for unmatched values only).

See Also

`knbs_county_code()`

Examples

```
knbs_match_counties(c("Nairobi City County", "Kakamenga", "12"))
```

`knbs_metadata` *Inspect the provenance of a knbsdata object*

Description

Returns the metadata attached to an object: where it came from, when it was retrieved, which release it represents and how it should be cited.

Usage

```
knbs_metadata(x, ...)
```

Arguments

`x`	A knbsdata object.
`...`	Passed to methods.

Value

A `knbs_metadata` object.

See Also

`knbs_cite()`

Examples

```
# Metadata accompany every object the package returns.
meta <- knbs_metadata(knbs_finaccess_files())
meta$producer
meta$licence
```

knbs_option *Get a knbsdata option*

Description

Retrieves a single package option. Options are stored as ordinary R options prefixed with `knbsdata.`, so `knbs_option("timeout")` reads `getOption("knbsdata.timeout")` and falls back to the package default.

Usage

```
knbs_option(name)
```

Arguments

`name`	Option name, without the `knbsdata.` prefix. One of `"cache"`, `"cache_dir"`, `"timeout"`, `"retries"`, `"user_agent"` or `"quiet"`.

Value

The option value. `NULL` for options that are unset and have no default (`cache_dir`, `user_agent`).

See Also

`knbs_options()` to set options.

Examples

```
knbs_option("timeout")
knbs_option("cache")
```

knbs_options *Set knbsdata options*

Description

Sets package options for the current session. Arguments left as `NULL` are unchanged. Called with no arguments, returns the current values.

Usage

```
knbs_options(
  cache = NULL,
  cache_dir = NULL,
  timeout = NULL,
  retries = NULL,
  user_agent = NULL,
  quiet = NULL
)
```

Arguments

cache	TRUE or FALSE. Whether retrieval functions may read from and write to the local cache.
cache_dir	Path to the cache directory. NULL uses tools::R_user_dir("knbsdata", "cache").
timeout	Request timeout in seconds.
retries	Maximum number of retries for retryable HTTP failures.
user_agent	Override for the HTTP user agent string. NULL uses a transparent default identifying the package and version.
quiet	TRUE or FALSE. Whether to suppress progress messages.

Details

To restore an option to its default, use options() directly, for example options(knbsdata.cache_dir = NULL), or withr::with_options() for a temporary change.

Value

When setting, the previous values as a named list, invisibly. When called with no arguments, the current values as a named list.

See Also

knbs_option() to read a single option.

Examples

```
old <- knbs_options(timeout = 60)
knbs_option("timeout")

# restore
knbs_options(timeout = old$timeout)
```

knbs_publications *Search official KNBS publications*

Description

[Experimental]

Searches the documents in the KNBS website's media library: statistical releases, survey reports, the Economic Survey and other published files.

Usage

```
knbs_publications(
  query = NULL,
  year = NULL,
  format = NULL,
  limit = 100,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

query	Free-text search over titles and descriptions.
year	Keep only publications whose title names this year, or a range c(from, to).
format	Keep only these file formats: any of "pdf", "xlsx", "xls", "docx", "csv", "zip".
limit	Maximum number of records to return (at most 100).
source	Source id. Defaults to the first verified publications source.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Details

Two honesty notes about this source. First, release_date is the date the file was uploaded to the site, not the publication's reference period; the year column is read from the title, and is NA when the title names none. Second, the media library also holds speeches, launch remarks and forms — narrowing to what you mean is the job of query and format.

The source ignores its own file-type parameter (audit F-22), so the format filter is applied by the package — and because a page of recent uploads may contain no matches at all, supplying format makes the search page back through older uploads until it has limit matches or the matches are exhausted. Expect a format-filtered search to take a few seconds longer than a plain one.

The reference-year filter is applied to the title-derived year, after retrieval. With a narrow query this is exact; with a broad one it filters only within the records retrieved.

Value

A knbs_catalogue tibble with columns publication_id, title, year, release_date, modified, description, format, file_size, file_name, download_url, landing_page, source_id, checksum and access.

See Also

knbs_download_publication()

Examples

```
knbs_publications("economic survey")
knbs_publications("consumer price", format = "xlsx")
knbs_publications("census", year = c(2019, 2023))
```

---

knbs_read_microdata *Read a microdata file into R*

Description

[Experimental]

Reads a Stata (.dta), SPSS (.sav, .zsav, .por) or Excel (.xlsx, .xls) file, or the data file inside a .zip archive, keeping its variable and value labels. Pass it a path or what knbs_download_finaccess() returns. When the file has a knbsdata provenance sidecar beside it, the result carries that provenance: knbs_metadata() says where it came from and knbs_cite() cites it.

Usage

```
knbs_read_microdata(x, file = NULL, ...)
```

Arguments

x	A path, or a download record from knbs_download_finaccess() holding one data file.
file	For an archive, the entry to read. Needed only when it holds more than one data file.
...	Passed to the reader: haven::read_dta(), haven::read_sav(), haven::read_por() or readxl::read_excel(). col_select reads only some variables of a large Stata or SPSS file, and n_max only its first rows.

Details

Stata and SPSS files are read with **haven**, which must be installed. Value labels arrive as `haven::labelled()` columns: `haven::as_factor()` turns them into factors and `haven::zap_labels()` drops them. Each column's question text is in its `"label"` attribute.

An archive is unpacked to a temporary directory, which is removed once the file is read. If it holds more than one data file, name the one to read with `file`.

A first variable name that begins with the remnant of a byte-order mark, an encoding slip that puts a stray character or two in front of a name such as `interview__key`, is repaired, and the repair is noted in the metadata.

Value

A tibble. With a provenance sidecar, a `knbs_microdata` tibble whose metadata records the source, the producers, the file's checksum and its terms.

See Also

`knbs_download_finaccess()`, `knbs_cite()`

Examples

```
# Any Stata or SPSS file; labels survive the round trip
path <- tempfile(fileext = ".dta")
haven::write_dta(
  tibble::tibble(id = 1:3,
                 sex = haven::labelled(c(1, 2, 1), c(Male = 1, Female = 2))),
  path
)
knbs_read_microdata(path)
knbs_read_microdata(path, col_select = "sex")


# A FinAccess survey, downloaded, verified and read with its provenance
fa <- knbs_read_microdata(
  knbs_download_finaccess("mse2023-stata", file.path(tempdir(), "finaccess"))
)
knbs_cite(fa)
```

---

knbs_release_series *KNBS's regular release series*

Description

[Experimental]

The series `knbs_releases()` recognises, with the id to ask for each by. Curated from the titles in the KNBS website's publications library.

Usage

```
knbs_release_series()
```

Value

A tibble with columns `series_id`, `series` and `frequency`.

See Also

`knbs_releases()`

Examples

```
knbs_release_series()
```

`knbs_releases` *What KNBS has released lately*

Description

[Experimental]

Lists the documents KNBS has added to its website, newest first, with each one's release series where it belongs to one — the monthly CPI and inflation release, Leading Economic Indicators, the quarterly GDP and balance of payments reports, the Economic Survey and the rest — and the call that downloads it. Tenders, job adverts and other notices are left out unless asked for.

```
knbs_releases()                    # the last 30 days
knbs_releases(series = "cpi")      # every CPI release, latest first
knbs_releases(series = "cpi")[1, ] # has this month's come out?
```

Usage

```
knbs_releases(
  since = NULL,
  until = NULL,
  series = NULL,
  notices = FALSE,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

since, until	Keep releases on or after since and on or before until: dates, or "YYYY-MM-DD" strings. By default the last 30 days, or, when series is given, the series' whole history in the library.
series	Keep only these series: ids or names from knbs_release_series(), matched without regard to case ("cpi", "lei", "Economic Survey").
notices	Whether to include tenders, adverts and other notices.
cache	Whether to use the local cache.
refresh	Whether to re-list the library.

Details

The date is the day the document was uploaded to the website, which is the day it was released for regular publications. It is not the period covered, and older releases were uploaded in bulk when the library was set up in mid-2023. period gives the month ("2026-09") or quarter ("2026-Q2") a title names, and is NA where the title names neither.

A document's series is read from its title by a curated list of KNBS's regular releases, see knbs_release_series(); titles vary ("Kenya Residential Proprety Price Index", "Leading Economic Indicators- March 2026"), so the patterns allow for that. A document matching no series has series NA. Notices are titles naming a tender, procurement, framework agreement, advert, vacancy or employment form.

The publications library is listed once a day and cached, as for knbs_search().

Value

A knbs_catalogue tibble: release_date, series, period, title, format, publication_id, landing_page, and get — the call that downloads the document.

See Also

knbs_release_series(), knbs_publications(), knbs_search()

Examples

```
knbs_releases(since = "2026-09-01")
knbs_releases(series = c("cpi", "lei"), since = "2026-01-01")
```

---

knbs_retail_prices *National average retail prices of the CPI basket commodities*

Description

[Experimental]

Returns the monthly national average retail prices the KNBS CPI Explorer publishes for the 22 commodities it tracks — maize flour, bread, milk, petrol, house rent and the rest — one row per commodity per month, in Kenya Shillings per the stated unit and quantity.

Usage

```
knbs_retail_prices(
  commodity = NULL,
  start = NULL,
  end = NULL,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

commodity	Keep only these commodities, by CPI001-style code (exact) or by name (matched as text).
start, end	Keep only these months. A year, a "YYYY-MM" month, or a date.
source	Source id.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Details

These are national averages only: the source holds no regional retail prices. The portal states no licence; knbs_metadata() on the result says so. A range query retrieves the full history once (a few seconds, then cached); an exact single month is fetched directly.

Value

A knbs_prices tibble with columns commodity_code, commodity, unit, quantity, period, period_label, price and source_id.

See Also

knbs_get_indicator() for the price *indices* the same portal serves.

Examples

```
knbs_retail_prices(commodity = "petrol", start = 2024)
knbs_retail_prices(start = "2026-06", end = "2026-06")
```

knbs_search *Search everything the package can retrieve*

Description

[Experimental]

One search across the indicator catalogues of every verified source, the survey catalogue of the Kenya National Data Archive, the open FinAccess survey microdata, the publications in the KNBS website's media library, and the curated Statistical Abstract tables. Results come back ranked, in one schema, each with the exact call that retrieves it — so finding something and getting it are one copy and paste apart.

Usage

```
knbs_search(
  query,
  type = c("indicator", "table", "survey", "microdata", "publication"),
  limit = 50,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

query	Words to search for.
type	Which kinds of resource to search: any of "indicator", "table", "survey", "microdata" and "publication". All by default. A "microdata" result is one survey round whose data can be downloaded openly; its get call downloads and reads it.
limit	Maximum number of results.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached catalogue.

Details

Matching is done by the package, over each catalogue in full, so that it means the same thing everywhere: every word of query must appear in a result's title, id or topic. Words match by stem — "index" finds "Indices", "prices" finds "Price" — and a word of four letters or more also matches as the start of a longer one ("agric" finds "Agriculture"). A few acronyms common in Kenyan statistics are understood both ways: CPI, GDP, GCP, GVA, PPI, BOP, KDHS, KIHBS and LEI. Common words such as "of" and "the" are ignored.

Results are ranked in tiers: an exact id, then an exact title, then the query as a phrase in the title ("consumer price index" is the phrase in "Consumer Price Indices and Inflation Rates"), then every word somewhere in the title, then a match elsewhere. Within a tier, data comes before documents and newer releases before older ones.

Catalogues are cached for a day. The first search of the day lists the whole publications library, around 16 requests at the site's polite rate; later searches are instant. If a source cannot be reached its results are missing and a warning names it; the error comes only when no catalogue answers.

Geographic codes are not searched here; see knbs_geographies().

Value

A knbs_catalogue tibble with one row per result: resource_type, resource_id, title, topic, period, release_date, access, source_id, url, and get — the R call that retrieves the resource.

See Also

knbs_indicators(), knbs_search_surveys(), knbs_finaccess_files(), knbs_publications(), knbs_abstract_tables()

Examples

```
# The microdata and table catalogues ship with the package: no request
knbs_search("financial inclusion", type = "microdata")
knbs_search("capital formation", type = "table")

knbs_search("consumer price index")
knbs_search("gdp", type = c("indicator", "table"))
hits <- knbs_search("county population")
hits$get[[1]] # the call that fetches the top result
```

---

knbs_search_surveys *Search official survey and census studies*

Description

[Experimental]

Searches the study catalogue of a supported survey archive and returns one row per study.

Usage

```
knbs_search_surveys(
  query = NULL,
  year = NULL,
  producer = NULL,
  access = NULL,
  limit = 100,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

query	Free-text search. NULL returns the whole catalogue.
year	A single year, or c(from, to) for a range. Matched against the study's reference years; a study spanning several years matches if any of them falls in the range. Studies the source records no reference year for cannot match, so they are excluded with a knbs_warning_missing_metadata warning that says how many, and names them in its excluded field.
producer	Keep only studies whose producing entity contains this text.
access	Keep only these access types; see the access_type column.
limit	Maximum number of studies to return, counted after the year, producer and access filters.
source	Source id. Defaults to the first verified source that supports survey search.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Details

Some catalogue fields — acronym, geographic_coverage and unit_of_analysis — are not carried in search results and come back as NA. They are held in the study record, which knbs_survey_metadata() retrieves. Filling them here would mean one extra request per row.

Value

A knbs_catalogue tibble with columns survey_id, title, acronym, reference_years, country, producer, geographic_coverage, unit_of_analysis, access_type, catalogue_url, metadata_updated and source_id.

See Also

knbs_survey_metadata(), knbs_variables(), knbs_survey_files()

Examples

```
knbs_search_surveys("household budget")
knbs_search_surveys(year = c(2015, 2020), access = "public_use")
```

---

knbs_sources *Supported KNBS sources*

Description

Lists the sources declared in the package source registry, together with the operations each one supports and how far it has been verified. No network request is made; this reports what the package believes about each source.

Usage

```
knbs_sources(operation = NULL, status = NULL)
```

Arguments

operation	Optional. Keep only sources supporting this operation, for example "search_surveys" or "get_indicator".
status	Optional. Keep only sources at these stability levels: one or more of "discovery", "experimental", "beta", "stable" or "deprecated".

Details

Sources whose base_url has not yet been verified are shown with stability "discovery" and are not usable for retrieval. Endpoint verification is the subject of the source-discovery audit in dev/source-discovery/.

Value

A tibble with one row per source and columns source_id, source_name, source_type, operations (a list-column), stability, authentication, last_verified and base_url.

Examples

```
knbs_sources()
knbs_sources(operation = "search_surveys")
```

---

knbs_status *Report on the health of supported sources*

Description

Answers two different questions depending on live. With live = FALSE it reports what the package knows about each source without contacting anything. With live = TRUE it asks each verified source to answer a small request and reports what happened.

Usage

```
knbs_status(source = NULL, live = interactive(), timeout = 10)
```

Arguments

source	Source ids to check. Defaults to every registered source.
live	Whether to contact the sources. Defaults to interactive(), so that automated runs never depend on a live service.
timeout	Request timeout in seconds, applied only to this call.

Details

The `status` column is one of:

`ok` The source answered.

`unavailable` The source was contacted and did not answer usefully. The `message` column says why.

`unverified` The source's endpoint has not been verified, so there is nothing to contact. It is never checked live.

`not_checked` The source could have been checked, but `live` was `FALSE`.

Value

A tibble with one row per source and columns `source_id`, `operation`, `status`, `http_status`, `response_time_ms`, `checked_at` and `message`.

See Also

`knbs_sources()` for what each source supports.

Examples

```
# Offline: reports what the registry knows, contacts nothing.
knbs_status(live = FALSE)

# Contacts each verified source.
knbs_status(live = TRUE)
```

---

knbs_survey_files *List the data files belonging to a study*

Description

[Experimental]

Reports what the archive holds and whether the package may retrieve it. At present no supported source permits programmatic download of microdata, so `downloadable` is `FALSE` throughout; use `catalogue_url` from `knbs_search_surveys()` to reach the official access process.

Usage

```
knbs_survey_files(
  survey,
  type = NULL,
  access = NULL,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

survey	A study identifier, or enough of its title to identify it.
type	Keep only files of this type.
access	Keep only files with these access types.
source	Source id.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Details

This lists data files only. Questionnaires, reports and technical documents are out of scope: the archive publishes no machine-readable listing of them, and retrieving them would mean parsing a web page. They are linked from the study's page on the archive, which `catalogue_url` points to.

Value

A tibble with columns `survey_id`, `file_id`, `file_name`, `file_type`, `format`, `size`, `case_count`, `var_count`, `access_type`, `downloadable`, `download_url`, `checksum` and `source_id`.

See Also

`knbs_variables()`, `knbs_download_microdata()`

Examples

```
knbs_survey_files("KEN-KNBS-KIHBS-2015-2016-v01")
```

---

knbs_survey_metadata *Retrieve the full metadata record for a study*

Description

[Experimental]

Retrieves the archive's full descriptive record for one study.

Usage

```
knbs_survey_metadata(
  survey,
 include = c("overview", "scope", "sampling", "collection", "processing", "access",
    "citation"),
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

survey	A study identifier, or enough of its title to identify it unambiguously.
include	Which descriptive sections to keep. The underlying records are DDI documents, in which sampling, collection and processing all live in one method section, so asking for any of them returns that section.
source	Source id.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Value

A knbs_survey metadata object. Use knbs_metadata() to inspect it and knbs_cite() to cite it.

See Also

knbs_search_surveys(), knbs_variables(), knbs_survey_files()

Examples

```
meta <- knbs_survey_metadata("KEN-KNBS-KIHBS-2015-2016-v01")
knbs_cite(meta)
```

---

knbs_variables *Search the variables described by a study*

Description

[Experimental]

Returns variable metadata only. No microdata are downloaded, and none can be: the package does not fetch gated files.

Usage

```
knbs_variables(
  survey,
  search = NULL,
  file = NULL,
  name = NULL,
  concept = NULL,
  limit = Inf,
  details = FALSE,
  source = NULL,
  cache = knbs_option("cache"),
  refresh = FALSE
)
```

Arguments

survey	A study identifier, or enough of its title to identify it.
search	Free text matched against variable names and labels.
file	Keep only variables from this file, by file id or file name.
name	Exact variable name.
concept	Keep only variables with this concept. Requires details = TRUE, since concepts are only in the per-variable records.
limit	Maximum number of variables to return.
details	Whether to retrieve the full record for each variable. This costs one request per variable, so it is capped at 100 rows.
source	Source id.
cache	Whether to use the local cache.
refresh	Whether to bypass any cached response.

Details

Question text, universe, concept and value labels each require one request per variable, so they are NA unless details = TRUE.

Value

A tibble of variable metadata. A study catalogued without variable documentation returns zero rows, which is a normal answer rather than an error.

See Also

knbs_survey_files(), knbs_survey_metadata()

Examples

```
knbs_variables("KEN-KNBS-KIHBS-2015-2016-v01", search = "consumption")
```

---

knbs_write *Write KNBS data to a file, with its provenance*

Description

[Experimental]

Saves a table for colleagues who work in Excel, Stata, SPSS or anything that reads CSV, without losing where it came from. The format follows the file extension:

- .xlsx - an Excel workbook with the data on a data sheet and an about sheet carrying the title, source, licence, release date, retrieval time and a ready-made citation.

- `.csv` - UTF-8 CSV, with missing values as empty cells.
- `.dta` - Stata, with the title as the dataset label.
- `.sav` - SPSS, likewise.
- `.rds` - R, keeping the object whole, metadata included.

CSV, Stata and SPSS files get a `<file>.metadata.json` provenance sidecar beside them, the same sidecar `knbs_download_publication()` writes.

Usage

```
knbs_write(
  x,
  path,
  format = NULL,
  metadata = NULL,
  overwrite = FALSE,
  quiet = knbs_option("quiet")
)
```

Arguments

`x`	A data frame: a result of this package, or any table.
`path`	Where to write. The extension chooses the format.
`format`	One of `"xlsx"`, `"csv"`, `"dta"`, `"sav"` or `"rds"`, to override the extension.
`metadata`	The provenance to record. Defaults to `x`'s own.
`overwrite`	Whether to replace an existing file.
`quiet`	Whether to suppress the confirmation message.

Details

Codes stay text: `"01"` is written as text in every format, never as the number 1. In CSV that means a quoted string, which Excel may still turn into a number when it opens the file; for Excel, write `.xlsx`.

List columns (the value labels of a survey's variables, for example) have no equivalent in these formats; each cell is written as JSON text, and the provenance says which columns were converted.

Stata limits variable names to 32 characters. Longer names are shortened

- keeping a `_code` or `_type` ending, and staying unique - and the full name becomes the variable's label; the provenance lists every rename.

A table derived with `summarise()` or a pivot is plain data and carries no provenance of its own. Pass the source's with `metadata = knbs_metadata(original)`; without it the file is still written, with a warning, and the `about` sheet or sidecar says no provenance was attached.

Excel output needs the writexl package and Stata or SPSS output the haven package; you are asked to install them if they are missing.

Value

The path written, invisibly.

See Also

knbs_metadata(), knbs_cite()

Examples

```
# The FinAccess file catalogue ships with the package, provenance and all
files <- knbs_finaccess_files(content = "data")
path <- file.path(tempdir(), "finaccess-files.csv")
knbs_write(files, path)                       # + finaccess-files.csv.metadata.json

gdp <- knbs_abstract_table("2.2")
knbs_write(gdp, file.path(tempdir(), "gdp-by-activity.xlsx"))   # data + about sheet
knbs_write(gdp, file.path(tempdir(), "gdp-by-activity.dta"))

# A derived table keeps the source's provenance when you pass it on
totals <- dplyr::summarise(gdp, value = sum(value), .by = period)
knbs_write(totals, file.path(tempdir(), "totals.csv"), metadata = knbs_metadata(gdp))
```

Index

format(), *8*

haven::labelled(), *25*
haven::read_dta(), *24*
haven::read_por(), *24*
haven::read_sav(), *24*

interactive(), *32*

knbs_abstract_table, 3
knbs_abstract_table(), *4*
knbs_abstract_tables, 4
knbs_abstract_tables(), *3, 4*, *30*
knbs_cache_clear, 5
knbs_cache_clear(), *6, 7*
knbs_cache_dir, 5
knbs_cache_dir(), *6*
knbs_cache_info, 6
knbs_cache_info(), *6*
knbs_cache_prune, 7
knbs_cache_prune(), *5, 6*
knbs_cite, 7
knbs_cite(), *4*, *10*, *13*, *17*, *20*, *24, 25*, *35*, *38*
knbs_county_code, 8
knbs_county_code(), *3*, *19, 20*
knbs_download_finaccess, 9
knbs_download_finaccess(), *12*, *14*, *24, 25*
knbs_download_microdata, 11
knbs_download_microdata(), *34*
knbs_download_publication, 12
knbs_download_publication(), *24*, *37*
knbs_finaccess_files, 13
knbs_finaccess_files(), *10–12*, *30*
knbs_geographies, 15
knbs_geographies(), *3*, *8, 9*, *17, 18*, *30*
knbs_get_indicator, 16
knbs_get_indicator(), *16*, *19*, *28*
knbs_indicators, 18
knbs_indicators(), *16*, *18*, *30*
knbs_match_counties, 19
knbs_match_counties(), *9*
knbs_metadata, 20
knbs_metadata(), *3, 4*, *8*, *17*, *24*, *28*, *35*, *38*
knbs_option, 21
knbs_option(), *22*
knbs_options, 21
knbs_options(), *21*
knbs_publications, 23
knbs_publications(), *4*, *13*, *27*, *30*
knbs_read_microdata, 24
knbs_read_microdata(), *10–12*, *14*
knbs_release_series, 25
knbs_release_series(), *27*
knbs_releases, 26
knbs_releases(), *25, 26*
knbs_retail_prices, 27
knbs_search, 29
knbs_search(), *27*
knbs_search_surveys, 30
knbs_search_surveys(), *30*, *33*, *35*
knbs_sources, 31
knbs_sources(), *33*
knbs_status, 32
knbs_survey_files, 33
knbs_survey_files(), *12*, *31*, *35, 36*
knbs_survey_metadata, 34
knbs_survey_metadata(), *31*, *36*
knbs_variables, 35
knbs_variables(), *31*, *34, 35*
knbs_write, 36

options(), *22*

readxl::read_excel(), *24*

tempdir(), *5*
tibble, *6*, *32, 33*
tools::R_user_dir(), *5*

withr::with_options(), *22*