Package 'nomisdata'

May 9, 2026

Type Package

Title Access 'Nomis' UK Labour Market Data and Statistics

Version 0.1.2

Description Interface to the 'Nomis' database (<https://www.nomisweb.co.uk>), maintained by Durham University on behalf of the Office for National Statistics (ONS). Provides access to UK labour market statistics including census data, benefit claimant counts, and employment surveys. Supports automatic pagination, optional disk caching, spatial data via 'sf', and tidy data output. Independent implementation unaffiliated with ONS or Durham University.

License MIT + file LICENSE

URL https://github.com/cherylisabella/nomisdata

BugReports https://github.com/cherylisabella/nomisdata/issues

Imports cli (>= 3.6.0), dplyr (>= 1.1.0), httr2 (>= 1.0.0), jsonlite (>= 1.8.0), rlang (>= 1.1.0), tibble (>= 3.2.0), utils, digest, methods

Suggests cachem (>= 1.0.0), ggplot2 (>= 3.4.0), janitor (>= 2.2.0), knitr (>= 1.42), memoise (>= 2.0.0), rappdirs (>= 0.3.0), readr (>= 2.1.0), rmarkdown (>= 2.20), rsdmx (>= 0.6.0), scales (>= 1.2.0), sf (>= 1.0.0), testthat (>= 3.1.0), withr (>= 2.5.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Config/Needs/website tidyverse/tidytemplate

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

NeedsCompilation no

Author Cheryl Isabella Lim [aut, cre] (ORCID:
<https://orcid.org/0009-0004-5766-1392>)

Maintainer Cheryl Isabella Lim <cheryl.academic@gmail.com>

Repository CRAN

Date/Publication 2026-02-10 14:40:02 UTC

Contents

`add_geography_names` *Join Geography Names*

Description

Adds human-readable geography names to data.

Usage

```
add_geography_names(data, dataset_id = "NM_1_1")
```

Arguments

`data`	Data frame with GEOGRAPHY_CODE column
`dataset_id`	Dataset to get geography names from

Value

Data frame with GEOGRAPHY_NAME added

Examples

```
data <- fetch_nomis("NM_1_1", time = "latest", geography = "TYPE499")
data_with_names <- add_geography_names(data)
```

---

aggregate_geography *Aggregate Data by Geography Level*

Description

Aggregates data to higher geography levels.

Usage

```
aggregate_geography(data, to_type, value_col = "OBS_VALUE", fun = sum)
```

Arguments

data	Data frame with geography codes
to_type	Target geography TYPE code
value_col	Column containing values to aggregate (default: "OBS_VALUE")
fun	Aggregation function (default: sum)

Value

A tibble with aggregated data grouped by specified variables.

Examples

```
data(jsa_sample)
aggregated <- aggregate_geography(jsa_sample, "TYPE499", "OBS_VALUE")
head(aggregated)
```

aggregate_time *Aggregate Time Series*

Description

Aggregates data over time periods.

Usage

```
aggregate_time(
  data,
  period = c("year", "quarter", "month"),
  value_col = "OBS_VALUE",
  fun = mean
)
```

Arguments

data	Data frame with DATE column
period	Aggregation period: "year", "quarter", "month"
value_col	Column containing values to aggregate
fun	Aggregation function (default: mean)

Value

A tibble with PERIOD column and aggregated values.

Examples

```
data(jsa_sample)

if ("DATE" %in% names(jsa_sample)) {
  yearly_data <- aggregate_time(jsa_sample, "year", "OBS_VALUE")
}
```

browse_dataset *Browse Dataset Online*

Description

Opens the Nomis web interface for a dataset in your browser.

Usage

```
browse_dataset(id, page = c("dataset", "download", "metadata"))
```

Arguments

id	Dataset ID (e.g., "NM_1_1")
page	Which page to open: "dataset", "download", "metadata"

Value

Invisible TRUE if successful. Called for side effects (opening browser).

Examples

```
## Not run:
browse_dataset("NM_1_1")
browse_dataset("NM_1_1", page = "download")

## End(Not run)
```

---

clear_cache *Clear All Caches*

Description

Removes all cached data.

Removes all cached data from disk and clears memoised functions.

Usage

```
clear_cache()
```

```
clear_cache()
```

Value

Invisible TRUE. Called for side effects.

Invisible TRUE. Called for side effects (clearing cache files).

Examples

```
enable_cache()
clear_cache()


enable_cache(tempfile("nomis_cache"))

clear_cache()
```

dataset_overview *Get Dataset Overview*

Description

Get Dataset Overview

Usage

```
dataset_overview(id, select = NULL)
```

Arguments

id	Dataset ID (required)
select	Character vector of sections to return

Value

Tibble with overview information

Examples

```
dataset_overview("NM_1_1")
dataset_overview("NM_1_1", select = c("Keywords", "Units"))
```

describe_dataset *Describe Dataset Structure*

Description

Describe Dataset Structure

Usage

```
describe_dataset(id = NULL)
```

Arguments

id	Dataset ID (e.g., "NM_1_1"). If NULL, returns all datasets.

Value

Tibble with dataset metadata

Examples

```
describe_dataset("NM_1_1")
all_datasets <- describe_dataset()
```

---

enable_cache *Enable caching for API responses*

Description

Enable caching for API responses

Usage

```
enable_cache(path = NULL)
```

Arguments

path	Cache directory path. If NULL, uses an appropriate default location.

Value

Path to cache directory (invisibly)

Examples

```
# Use temporary directory for cache
enable_cache(tempfile("nomis_cache"))
```

---

explore_dataset *Interactive Dataset Explorer*

Description

Opens an interactive menu to explore dataset dimensions and codes. Only works in interactive R sessions.

Usage

```
explore_dataset(id)
```

Arguments

id	Dataset ID

Value

Selected codes as a list, or NULL if not interactive.

Examples

```
## Not run:
# Only works in interactive sessions
explore_dataset("NM_1_1")

## End(Not run)
```

`fetch_codelist` *Fetch Codelist*

Description

Fetch Codelist

Usage

```
fetch_codelist(id, concept, search = NULL)
```

Arguments

`id`	Dataset ID
`concept`	Concept name
`search`	Search term

Value

Tibble of codes

Examples

```
fetch_codelist("NM_1_1", "geography")
fetch_codelist("NM_1_1", "geography", "*manchester*")
```

`fetch_nomis` *Fetch Data from Nomis*

Description

Main function to download data from Nomis datasets.

Usage

```
fetch_nomis(
  id,
  time = NULL,
  date = NULL,
  geography = NULL,
  sex = NULL,
  measures = NULL,
  exclude_missing = FALSE,
  select = NULL,
  ...,
  .progress = interactive()
)
```

Arguments

`id`	Dataset ID (required)
`time`	Time range using keywords or specific dates
`date`	Specific dates (alternative to time)
`geography`	Geography code(s)
`sex`	Sex/gender code(s)
`measures`	Measure code(s)
`exclude_missing`	Remove missing values
`select`	Column names to include
`...`	Additional dimension filters
`.progress`	Show progress bar for multi-part queries

Value

Tibble with requested data

Examples

```
fetch_nomis(
  "NM_1_1",
  time = "latest",
  geography = "TYPE499",
  measures = 20100,
  sex = 7
)

fetch_nomis(
  "NM_1_1",
  date = c("latest", "prevyear"),
  geography = c("2092957697", "2092957698"),
  measures = 20100
)
```

---

fetch_spatial *Fetch Spatial Data*

Description

Downloads data in KML format with spatial boundaries.

Usage

```
fetch_spatial(
  id,
  time = NULL,
  date = NULL,
  geography = NULL,
  select = NULL,
  exclude_missing = FALSE,
  ...,
  parse_sf = TRUE
)
```

Arguments

id	Dataset ID
time	Time period selection (same as fetch_nomis)
date	Specific date selection (alternative to time)
geography	Geography code(s) to filter
select	Column names to include
exclude_missing	Remove missing values if TRUE
...	Additional query parameters (measures, sex, etc.)
parse_sf	If TRUE and sf is available, parse to sf object

Value

KML data as text or sf object (if parse_sf = TRUE)

Examples

```
spatial_data <- fetch_spatial(
  "NM_1_1",
  time = "latest",
  geography = "TYPE480",
  measures = 20100,
  sex = 7
)
```

---

get_codes *Get Concept Codes*

Description

Get Concept Codes

Usage

```
get_codes(id, concept = NULL, type = NULL, search = NULL, ...)
```

Arguments

id	Dataset ID (required)
concept	Concept name (e.g., "geography", "sex"). If NULL, returns all concepts.
type	Optional type filter
search	Search term (supports wildcards)
...	Additional query parameters

Value

Tibble with codes and descriptions

Examples

```
get_codes("NM_1_1")

get_codes("NM_1_1", "geography")

get_codes("NM_1_1", "geography", "TYPE499")

get_codes("NM_1_1", "geography", search = "*manchester*")
```

`jsa_sample` *Sample Jobseeker's Allowance Data*

Description

A small sample dataset from the Jobseeker's Allowance dataset (NM_1_1) for the UK, Great Britain, and England. Useful for offline examples and testing.

Usage

```
jsa_sample
```

Format

A tibble with 3 rows and 12 columns:

GEOGRAPHY_CODE ONS geography code

GEOGRAPHY_NAME Geography name (UK, GB, England)

SEX Sex code (7 = Total)

SEX_NAME Sex description

ITEM Item code

ITEM_NAME Item description

MEASURES Measure code (20100)

MEASURES_NAME Measure description

DATE Date code (YYYY-MM format)

DATE_NAME Date description

OBS_VALUE Observed value (number of claimants)

OBS_STATUS Observation status code

RECORD_COUNT Number of records in query

Source

Nomis API: https://www.nomisweb.co.uk

Examples

```
data(jsa_sample)
head(jsa_sample)
summary(jsa_sample$OBS_VALUE)
```

lookup_geography *Look up Geography Codes*

Description

Search for UK geography codes by name. Returns matching geographies from local authorities, regions, wards, and other levels.

Usage

```
lookup_geography(search_term, dataset_id = "NM_1_1", type = NULL)
```

Arguments

search_term	Name or partial name to search (e.g., "London", "Manchester")
dataset_id	Dataset to search in (default: "NM_1_1")
type	Optional geography TYPE code to filter results

Value

Tibble of matching geographies with codes and names

Examples

```
lookup_geography("London")
lookup_geography("Manchester")
lookup_geography("Birmingham", type = "TYPE464")  # Local authorities only
```

search_datasets *Search for Datasets*

Description

Search for Datasets

Usage

```
search_datasets(
  name = NULL,
  keywords = NULL,
  description = NULL,
  content_type = NULL,
  units = NULL
)
```

Arguments

name	Character vector of name search terms (supports wildcards *)
keywords	Character vector of keyword search terms
description	Character vector of description search terms
content_type	Character vector of content types
units	Character vector of units

Value

Tibble of matching datasets

Examples

```
search_datasets(name = "*employment*")
search_datasets(keywords = "census")
search_datasets(name = "*benefit*", keywords = "claimants")
```

---

set_api_key *Set API Key*

Description

Configure your Nomis API key for increased rate limits. Register at: https://www.nomisweb.co.uk/myaccount/userjoin.asp

Usage

```
set_api_key(key = NULL, persist = FALSE)
```

Arguments

key	API key string. If NULL, will prompt or check environment.
persist	If TRUE, saves to .Renviron for future sessions.

Value

Invisible TRUE if successful

Examples

```
set_api_key("your-key-here")
set_api_key("your-key-here", persist = TRUE)
```

`tidy_names` *Tidy Column Names*

Description

Tidy Column Names

Usage

```
tidy_names(df, style = "snake_case")
```

Arguments

`df`	Data frame
`style`	Naming style: "snake_case", "camelCase", "period.case"

Value

Data frame with tidied names

Examples

```
df <- data.frame(GEOGRAPHY_NAME = "UK", OBS_VALUE = 100)
tidy_names(df)
```

Index