

# Package ‘nomisr’

January 9, 2019

**Type** Package

**Title** Access 'Nomis' UK Labour Market Data

**Version** 0.4.1

**Description** Access UK official statistics from the 'Nomis' database. 'Nomis' includes data from the Census, the Labour Force Survey, DWP benefit statistics and other economic and demographic data from the Office for National Statistics, based around statistical geographies. See <https://www.nomisweb.co.uk/api/v01/help> for full API documentation.

**URL** <https://github.com/ropensci/nomisr>,  
<https://docs.evanodell.com/nomisr>

**BugReports** <https://github.com/ropensci/nomisr/issues>

**License** MIT + file LICENSE

**Depends** R (>= 3.4.0)

**Imports** jsonlite, tibble, readr, dplyr, httr, utils, rsdmx, rlang

**RoxygenNote** 6.1.1

**Suggests** knitr, rmarkdown, testthat, purrr, tidyr, magrittr

**VignetteBuilder** knitr

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Evan Odell [aut, cre] (<<https://orcid.org/0000-0003-1845-808X>>),  
Paul Egeler [rev, ctb],  
Christophe Dervieux [rev] (<<https://orcid.org/0000-0003-4474-2498>>)

**Maintainer** Evan Odell <[evanode1191@gmail.com](mailto:evanode1191@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-01-09 15:20:06 UTC

## R topics documented:

nomisr . . . . .	2
nomis_api_key . . . . .	2
nomis_codelist . . . . .	3
nomis_content_type . . . . .	4
nomis_data_info . . . . .	5
nomis_get_data . . . . .	6
nomis_get_metadata . . . . .	9
nomis_overview . . . . .	11
nomis_search . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

nomisr	<i>nomisr: Access Nomis UK Labour Market Data with R</i>
--------	--

---

### Description

Access UK official statistics from the Nomis database. Nomis includes data from the Census, the Labour Force Survey, DWP benefit statistics and other economic and demographic data from the Office for National Statistics.

### Details

The package provides functions to find what data is available, metadata, including the variables and query options for different datasets and a function for downloading data.

The full API documentation and optional registration for an API key is available at <https://www.nomisweb.co.uk/api/v01/help>.

---

nomis_api_key	<i>Nomis API Key</i>
---------------	----------------------

---

### Description

Assign or reassign API key for Nomis.

### Usage

```
nomis_api_key(check_env = FALSE)
```

### Arguments

check_env	If TRUE, will check the environment variable NOMIS_API_KEY first before asking for user input.
-----------	--

## Details

The Nomis API has an optional key. Using the key means that 100,000 rows can be returned per call, which can speed up larger data requests and reduce the chances of being rate limited or having requests timing out.

By default, `nomisr` will look for the environment variable `NOMIS_API_KEY` when the package is loaded. If found, the API key will be stored in the session option `nomisr.API.key`. If you would like to reload the API key or would like to manually enter one in, this function may be used.

You can sign up for an API key [here](#).

---

<code>nomis_codelist</code>	<i>Nomis codelists</i>
-----------------------------	------------------------

---

## Description

Nomis uses its own internal coding for query concepts. `nomis_codelist` returns the codes for a given concept in a tibble, given a dataset ID and a concept name. Note that some codelists, particularly geography, can be very large.

## Usage

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

## Arguments

<code>id</code>	A string with the ID of the particular dataset. Must be specified.
<code>concept</code>	A string with the variable concept to return options for. If left empty, returns all the variables for the dataset specified by <code>id</code> . Codes are not case sensitive and must be specified.
<code>search</code>	Search for codes that contain a given string. The wildcard character <code>*</code> can be added to the beginning and/or end of each search string. Search strings are not case sensitive. Defaults to <code>NULL</code> . Note that the search function is not very powerful for some datasets.

## Value

A tibble with the codes used to query specific concepts.

## See Also

[nomis\\_data\\_info\(\)](#)  
[nomis\\_get\\_metadata\(\)](#)  
[nomis\\_overview\(\)](#)

## Examples

```
x <- nomis_codelist("NM_1_1", "item")

# Searching for codes ending with "london"
y <- nomis_codelist("NM_1_1", "geography", search = "*london")

z <- nomis_codelist("NM_161_1", "cause_of_death")
```

---

nomis_content_type	<i>Nomis dataset content types</i>
--------------------	------------------------------------

---

## Description

Nomis content type metadata is included in annotation tags, in the form of `contenttype/<contenttype>` in the `annotationtitle` column in the `annotations.list`-column returned from `nomis_data_info()`. For example, the content types returned from dataset "NM\_1658\_1", using `nomis_data_info("NM_1658_1")`, are "geoglevel", "2001census" and "sources".

## Usage

```
nomis_content_type(content_type, id = NULL)
```

## Arguments

`content_type` A string with the content type to return metadata on.  
`id` A string with an optional `content_type` id.

## Value

A tibble with metadata on a given content type.

## See Also

[nomis\\_search\(\)](#)  
[nomis\\_data\\_info\(\)](#)

## Examples

```
a <- nomis_content_type("sources")

tibble::glimpse(a)

b <- nomis_content_type("sources", id = "census")
```

```
tibble::glimpse(b)
```

---

nomis_data_info	<i>Nomis data structures</i>
-----------------	------------------------------

---

## Description

Retrieve metadata on the structure and available variables for all available data sets or the information available in a specific dataset based on its ID.

## Usage

```
nomis_data_info(id)
```

## Arguments

`id` Dataset ID. If empty, returns data on all available datasets. If the ID of a dataset, returns metadata for that particular dataset.

## Value

A tibble with all available datasets and their metadata.

## See Also

```
nomis_get_data()  
nomis_get_metadata()  
nomis_overview()  
nomis_codelist()
```

## Examples

```
# Get info on all datasets  
x <- nomis_data_info()  
  
tibble::glimpse(x)  
  
# Get info on a particular dataset  
y <- nomis_data_info('NM_1658_1')  
  
tibble::glimpse(y)
```

---

nomis_get_data	<i>Retrieve Nomis datasets</i>
----------------	--------------------------------

---

## Description

Retrieves specific datasets from Nomis, based on their ID. To find dataset IDs, use `nomis_data_info()`. Datasets are retrieved in csv format and parsed with the `read_csv` function from the `readr` package into a tibble, with all columns parsed as character columns.

To find the code options for a given dataset, use `nomis_get_metadata()` for specific codes, and `nomis_codelist()` for code values.

This can be a slow process if querying significant amounts of data. Guest users are limited to 25,000 rows per query, although `nomisr` identifies queries that will return more than 25,000 rows, sending individual queries and combining the results of those queries into a single tibble. In interactive sessions, `nomisr` will warn you if guest users are requesting more than 350,000 rows of data, and if registered users are requesting more than 1,500,000 rows.

Note the difference between the time and date parameters. The time and date parameters should not be used at the same time. If they are, the function will retrieve data based on the the date parameter. If given more than one query, time will return all data available between those queries, inclusively, while date will only return data for the exact queries specified. So `time = c("first", "latest")` will return all data, while `date = c("first", "latest")` will return only the earliest and latest data published.

## Usage

```
nomis_get_data(id, time = NULL, date = NULL, geography = NULL,
  sex = NULL, measures = NULL, additional_queries = NULL,
  exclude_missing = FALSE, select = NULL, tidy = FALSE,
  tidy_style = "snake_case", query_id = NULL, ...)
```

## Arguments

<code>id</code>	A string containing the ID of the dataset to retrieve, in "nm_***_*" format. The <code>id</code> parameter is not case sensitive.
<code>time</code>	Parameter for selecting dates and date ranges. Accepts either a single date value, or two date values and returns all data between the two date values, There are two styles of values that can be used to query time. The first is one or two of "latest" (returns the latest available data), "previous" (the date prior to "latest"), "prevyear" (the date one year prior to "latest") or "first" (the oldest available data for the dataset). The second style is to use or a specific date or multiple dates, in the style of the time variable codelist, which can be found using the <code>nomis_get_metadata()</code> function. Values for the time and date parameters should not be used at the same time. If they are, the function will retrieve data based on the the date parameter. Defaults to NULL.

date	<p>Parameter for selecting specific dates. Accepts one or more date values. If given multiple values, only data for the given dates will be returned, but there is no limit to the number of data values. For example, <code>date=c("latest", latestMINUS3, latestMINUS6")</code> will return the latest data, data from three months prior to the latest data and six months prior to the latest data. There are two styles of values that can be used to query time.</p> <p>The first is one or more of "latest" (returns the latest available data), "previous" (the date prior to "latest"), "prevyear" (the date one year prior to "latest") or "first" (the oldest available data for the dataset).</p> <p>The second style is to use or a specific date or multiple dates, in the style of the time variable codelist, which can be found using the <a href="#">nomis_get_metadata()</a> function.</p> <p>Values for the time and date parameters should not be used at the same time. If they are, the function will retrieve data based on the the date parameter.</p> <p>Defaults to NULL.</p>
geography	<p>The code of the geographic area to return data for. If NULL, returns data for all available geographic areas, subject to other parameters. Defaults to NULL. In the rare instance that a geographic variable does not exist, if not NULL, the function will return an error.</p>
sex	<p>The code for sexes/genders to include in the dataset. Accepts a string or number, or a vector of strings or numbers. <code>nomisr</code> automatically voids any queries for sex if it is not an available code in the requested dataset. Defaults to NULL and returns all available sex/gender data.</p> <p>There are two different codings used for sex, depending on the dataset. For datasets using "SEX", 7 will return results for males and females, 6 only females and 5 only males. Defaults to NULL, equivalent to <code>c(5,6,7)</code> for datasets where sex is an option. For datasets using "C_SEX", 0 will return results for males and females, 1 only males and 2 only females. Some datasets use "GENDER" with the same values as "SEX", which works with both <code>sex = &lt;code&gt;</code> and <code>gender = &lt;code&gt;</code> as a dot parameter.</p>
measures	<p>The code for the statistical measure(s) to include in the data. Accepts a single string or number, or a list of strings or numbers. If NULL, returns data for all available statistical measures subject to other parameters. Defaults to NULL.</p>
additional_queries	<p>Any other additional queries to pass to the API. See <a href="https://www.nomisweb.co.uk/api/v01/help">https://www.nomisweb.co.uk/api/v01/help</a> for instructions on query structure. Defaults to NULL. Deprecated in package versions greater than 0.2.0 and will eventually be removed in a future version.</p>
exclude_missing	<p>If TRUE, excludes all missing values. Defaults to FALSE.</p>
select	<p>A character vector of one or more variables to include in the returned data, excluding all others. <code>select</code> is not case sensitive.</p>
tidy	<p>Logical parameter. If TRUE, converts variable names to <code>snake_case</code>, or another style as specified by the <code>tidy_style</code> parameter. Defaults to FALSE. The default variable name style from the API is <code>SCREAMING_SNAKE_CASE</code>.</p>

tidy_style	The style to convert variable names to, if tidy = TRUE. Accepts one of "snake_case", "camelCase" and "period.case". Defaults to "snake_case".
query_id	Results can be labelled as belonging to a certain query made to the API. query_id accepts any value as a string, and will be included in every row of the tibble returned by nomis_get_data in a column labelled "QUERY_ID" in the default SCREAMING_SNAKE_CASE used by the API. Defaults to NULL.
...	Use to pass any other parameters to the API. Useful for passing concepts that are not available through the default parameters. Only accepts concepts identified in <a href="#">nomis_get_metadata()</a> and concept values identified in <a href="#">nomis_codelist()</a> . Parameters can be quoted or unquoted. Each parameter should have a name and a value. For example, CAUSE_OF_DEATH = 10300 when querying dataset "NM_161_1". Parameters are not case sensitive. Note that R using partial matching for function variables, and so passing a parameter with the same opening characters as one of the above-named parameters can cause an error unless the value of the named parameter is specified, including as NULL. See example below:

### Value

A tibble containing the selected dataset. By default, all tibble columns except for the "OBS\_VALUE" column are parsed as characters.

### See Also

[nomis\\_data\\_info\(\)](#)  
[nomis\\_get\\_metadata\(\)](#)  
[nomis\\_codelist\(\)](#)  
[nomis\\_overview\(\)](#)

### Examples

```
# Return data on Jobseekers Allowance for each country in the UK
jobseekers_country <- nomis_get_data(id = "NM_1_1", time = "latest",
                                   geography = "TYPE499",
                                   measures = c(20100, 20201), sex = 5)

tibble::glimpse(jobseekers_country)

# Return data on Jobseekers Allowance for Wigan
jobseekers_wigan <- nomis_get_data(id = "NM_1_1", time = "latest",
                                   geography = "1879048226",
                                   measures = c(20100, 20201), sex = "5")

tibble::glimpse(jobseekers_wigan)

# annual population survey - regional - employment by occupation
emp_by_occupation <- nomis_get_data(id = "NM_168_1", time = "latest",
```



```

        geography = "2013265925", sex = "0",
        select = c("geography_code",
        "C_OCCPUK11H_0_NAME", "obs_vAlUE"))

tibble::glimpse(emp_by_occupation)

# Deaths in 2016 and 2015 by three specified causes,
# identified with nomis_get_metadata()
death <- nomis_get_data("NM_161_1", date = c("2016", "2015"),
        geography = "TYPE480",
        cause_of_death = c(10300, 102088, 270))

tibble::glimpse(death)

# All causes of death in London in 2016
london_death <- nomis_get_data("NM_161_1", date = c("2016"),
        geography = "2013265927", sex = 1, age = 0)

tibble::glimpse(london_death)

## Not run:
# Results in an error because `measure` is mistaken for `measures`
mort_data1 <- nomis_get_data(id = "NM_161_1", date = "2016",
        geography = "TYPE464", sex = 0, cause_of_death = "10381",
        age = 0, measure = 6)

# Does not error because `measures` is specified
mort_data2 <- nomis_get_data(id = "NM_161_1", date = "2016",
        geography = "TYPE464", sex = 0, measures = NULL,
        cause_of_death = "10381", age = 0, measure = 6)

## End(Not run)

```

---

nomis\_get\_metadata      *Nomis metadata concepts and types*

---

## Description

Retrieve all concept code options of all Nomis datasets, concept code options for a given dataset, or the all the options for a given concept variable from a particular dataset. Specifying concept will return all the options for a given variable in a particular dataset.

If looking for a more detailed overview of all available metadata for a given dataset, see [nomis\\_overview\(\)](#).

## Usage

```
nomis_get_metadata(id, concept = NULL, type = NULL, search = NULL,
        additional_queries = NULL, ...)
```

## Arguments

<code>id</code>	The ID of the particular dataset. Returns no data if not specified.
<code>concept</code>	A string with the variable concept to return options for. If left empty, returns all the variables for the dataset specified by <code>id</code> . Codes are not case sensitive. Defaults to <code>NULL</code> .
<code>type</code>	A string with options for a particular code value, to return types of variables available for a given code. Defaults to <code>NULL</code> . If <code>concept == NULL</code> , <code>type</code> will be ignored.
<code>search</code>	A string or character vector of strings to search for in the metadata. Defaults to <code>NULL</code> . As in <code>nomis_search()</code> , the wildcard character <code>*</code> can be added to the beginning and/or end of each search string.
<code>additional_queries</code>	Any other additional queries to pass to the API. See <a href="https://www.nomisweb.co.uk/api/v01/help">https://www.nomisweb.co.uk/api/v01/help</a> for instructions on query structure. Defaults to <code>NULL</code> . Deprecated in package versions greater than 0.2.0 and will eventually be removed.
<code>...</code>	Use to pass any other parameters to the API.

## Value

A tibble with metadata options for queries using `nomis_get_data()`.

## See Also

`nomis_data_info()`  
`nomis_get_data()`  
`nomis_overview()`

## Examples

```
a <- nomis_get_metadata('NM_1_1')
tibble::glimpse(a)

b <- nomis_get_metadata('NM_1_1', 'geography')
tibble::glimpse(b)

# returns all types of geography
c <- nomis_get_metadata('NM_1_1', 'geography', 'TYPE')
tibble::glimpse(c)

# returns geography types available within Wigan
d <- nomis_get_metadata('NM_1_1', 'geography', '1879048226')
tibble::glimpse(d)
```

```
e <- nomis_get_metadata('NM_1_1', 'item', geography = 1879048226, sex = 5)
print(e)

f <- nomis_get_metadata('NM_1_1', 'item', search = "*married*")

tibble::glimpse(f)
```

---

nomis_overview	<i>Nomis dataset overview</i>
----------------	-------------------------------

---

## Description

Returns an overview of available metadata for a given dataset.

## Usage

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

## Arguments

<code>id</code>	The ID of the particular dataset. Returns no data if not specified.
<code>select</code>	A string or character vector of one or more overview parts to select, excluding all others. <code>select</code> is not case sensitive. The options for <code>select</code> are described below, and are taken from the <a href="#">Nomis API help page</a> .

## Value

A tibble with two columns, one a character vector with the name of the metadata category, and the other a list column of values for each category.

## Overview part options

**DatasetInfo** General dataset information such as name, description, sub-description, mnemonic, restricted access and status

**Coverage** Shows the geographic coverage of the main geography dimension in this dataset (e.g. United Kingdom, England and Wales etc.)

**Keywords** The keywords allocated to the dataset

**Units** The units of measure supported by the dataset

**ContentTypes** The classifications allocated to this dataset

**DateMetadata** Information about the first release, last update and next update

**Contact** Details for the point of contact for this dataset

**Analyses** Show the available analysis breakdowns of this dataset

**Dimensions** Individual dimension information (e.g. sex, geography, date, etc.)

**Dimension-concept** Allows a specific dimension to be selected (e.g. dimension-geography would allow information about geography dimension). This is not used if "Dimensions" is specified too.

**Codes** Full list of selectable codes, excluding Geography, which as a list of Types instead. (Requires "Dimensions" to be selected too)

**Codes-concept** Full list of selectable codes for a specific dimension, excluding Geography, which as a list of Types instead. This is not used if "Codes" is specified too (Requires "Dimensions" or equivalent to be selected too)

**DimensionMetadata** Any available metadata attached at the dimensional level (Requires "Dimensions" or equivalent to be selected too)

**Make** Information about whether user defined codes can be created with the MAKE parameter when querying data (Requires "Dimensions" or equivalent to be selected too)

**DatasetMetadata** Metadata attached at the dataset level

### See Also

[nomis\\_data\\_info\(\)](#)

[nomis\\_get\\_metadata\(\)](#)

### Examples

```
library(dplyr)

q <- nomis_overview("NM_1650_1")

q %>% tidyr::unnest(name) %>% glimpse()

s <- nomis_overview("NM_1650_1", select = c("Units", "Keywords"))

s %>% tidyr::unnest(name) %>% glimpse()
```

---

nomis\_search

*Search Nomis datasets*

---

### Description

A function to search for datasets on given topics. In the case of multiple search parameters, returns metadata on all datasets matching one or more of the parameters. The wildcard character \* can be added to the beginning and/or end of each search string.

### Usage

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

**Arguments**

name	A string or character vector of strings to search for in available dataset names. Defaults to NULL.
description	A string or character vector of strings to search for in available dataset descriptions. Note that <code>description</code> looks for complete matches, so wildcards should be used at the start and end of each string. Defaults to NULL.
keywords	A string or character vector of strings to search for in available dataset keywords. Defaults to NULL.
content_type	A string or character vector of strings to search for in available dataset content types. <code>content_type</code> can include an optional ID for that content type. Defaults to NULL.
units	A string or character vector of strings to search for in available dataset units. Defaults to NULL.

**Value**

A tibble with details on all datasets matching the search query.

**See Also**

[nomis\\_content\\_type\(\)](#)

**Examples**

```
x <- nomis_search(name = '*seekers*')  
y <- nomis_search(keywords = 'Claimants')  
  
# Return metadata of all datasets with content_type "sources".  
a <- nomis_search(content_type = "sources")  
  
# Return metadata of all datasets with content_type "sources" and  
# source ID "aces"  
b <- nomis_search(content_type = "sources-aces")
```

# Index

`nomis_api_key`, [2](#)  
`nomis_codelist`, [3](#)  
`nomis_codelist()`, [5](#), [6](#), [8](#)  
`nomis_content_type`, [4](#)  
`nomis_content_type()`, [13](#)  
`nomis_data_info`, [5](#)  
`nomis_data_info()`, [3](#), [4](#), [6](#), [8](#), [10](#), [12](#)  
`nomis_get_data`, [6](#)  
`nomis_get_data()`, [5](#), [10](#)  
`nomis_get_metadata`, [9](#)  
`nomis_get_metadata()`, [3](#), [5–8](#), [12](#)  
`nomis_overview`, [11](#)  
`nomis_overview()`, [3](#), [5](#), [8–10](#)  
`nomis_search`, [12](#)  
`nomis_search()`, [4](#), [10](#)  
`nomisr`, [2](#)  
`nomisr-package (nomisr)`, [2](#)