Package 'opencis'

September 7, 2026

Type Package

Title Import Data from Spanish Sociological Research Center (CIS)

Version 0.1.2

Description Search and import data directly to R from the Spanish Sociological Research Center (CIS) <https://www.cis.es/inicio>. The CIS is a public institution that conducts electoral and sociological research studies on the Spanish society. The CIS has a large database of surveys that can be accessed through its website. The package includes functions to search for surveys, survey questions and timeseries, and import the data directly to R.

License GPL (>= 3)

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports httr (>= 1.4.7), digest (>= 0.6.0), cachem (>= 1.1.0), tibble (>= 3.0.0), purrr (>= 1.0.0), haven (>= 2.5.3), magrittr (>= 2.0.0), rvest (>= 1.0.0), stringr (>= 1.0.0), memoise (>= 2.0.0)

URL https://opencis.spainelectoralproject.com, https://CRAN.R-project.org/package=opencis, https://github.com/hmeleiro/opencis

BugReports https://github.com/hmeleiro/opencis/issues

Suggests knitr, testthat (>= 3.0.0), rmarkdown

VignetteBuilder knitr

Config/Needs/website hmeleiro/spainelectoraltheme

Config/testthat/edition 3

NeedsCompilation no

Author Héctor Meleiro [aut, cre] (ORCID: <https://orcid.org/0009-0003-4468-2197>)

Maintainer Héctor Meleiro <hmeleiros@gmail.com>

Repository CRAN

Date/Publication 2026-09-07 18:30:14 UTC

Contents

browse_pdf 2
cache_info 3
clear_cache 3
download_study 4
get_data_dictionary 5
get_metadata 6
read_cis 6
search_all_cis 7
search_cis 8

Index **10**

`browse_pdf` *Open the questionnaire PDF of a CIS study*

Description

Opens a PDF document from a CIS study in the default browser.

Usage

```
browse_pdf(study_code, wanted_file = "cues")
```

Arguments

`study_code`	A string with the study code.
`wanted_file`	A keyword used to match the PDF filename inside the ZIP. Use `"cues"` (default) for the questionnaire or `"ft"` for the technical sheet.

Details

CIS study ZIP files typically contain two PDF documents:

- The **questionnaire** (cuestionario): use `wanted_file = "cues"`.
- The **technical sheet** (ficha técnica): use `wanted_file = "ft"`.

Value

Called for its side effect of opening the PDF in the browser. Returns `NULL` invisibly.

Examples

```
if (interactive()) {
# Open the questionnaire (cuestionario) for study 3328
browse_pdf("3328")

# Open the technical sheet (ficha técnica) for study 3328
browse_pdf("3328", wanted_file = "ft")
}
```

`cache_info` *Inspect the opencis HTTP cache*

Description

Reports the active disk-cache configuration and current usage. The cache is enabled by default and stores successful CIS responses for 24 hours.

Usage

```
cache_info()
```

Details

Configure it with the options `opencis.cache`, `opencis.cache_dir`, `opencis.cache_max_age` (seconds), and `opencis.cache_max_size` (bytes).

Value

A named list containing the cache state, directory, limits, number of entries, and current size in bytes.

`clear_cache` *Clear the opencis session cache*

Description

Clears the in-memory cache used by `search_cis` and `read_cis`, the persistent HTTP disk cache, and the HTTP sessions and their cookies. Call this to force fresh data or a fresh CIS anti-bot verification.

Usage

```
clear_cache(disk = TRUE)
```

Arguments

disk	Logical. If TRUE (the default), also remove all entries from the persistent HTTP cache. Set it to FALSE to retain downloaded responses between sessions.

Value

NULL invisibly.

---

download_study *Download a CIS study ZIP file to disk*

Description

Downloads the data ZIP file for a CIS study to a specified directory, instead of a temporary folder. Useful for projects that need to keep the raw data files.

Usage

```
download_study(study_code, destdir = ".")
```

Arguments

study_code	A string with the study code.
destdir	A string with the directory where the ZIP file will be saved. Defaults to the current working directory.

Value

The path to the saved ZIP file, invisibly.

Examples

```
if (interactive()) {
# Save the ZIP file to a temporary directory
path <- download_study("3328", destdir = tempdir())
cat("Saved to:", path, "\n")
}
```

`get_data_dictionary` *Extract a data dictionary from a CIS study data frame*

Description

Returns a tibble listing each variable in the data along with its variable label and value labels, as loaded by haven.

Usage

```
get_data_dictionary(data)
```

Arguments

`data`	A data.frame loaded from a CIS `.sav` file, typically the output of `read_cis`.

Value

A tibble with columns:

variable Variable name.

label Variable label, or `NA` if none.

value_labels A named numeric vector of value labels, or `NULL` for unlabelled variables (list-column).

Examples

```
# Create a small labelled data frame
df <- data.frame(
  SEXO = haven::labelled(c(1, 2, 1), labels = c(Hombre = 1, Mujer = 2)),
  EDAD = c(34, 51, 29)
)
attr(df$SEXO, "label") <- "Sexo"
attr(df$EDAD, "label") <- "Edad"

# Inspect its variable dictionary
dict <- get_data_dictionary(df)
print(dict)

# Find variables with a specific keyword in their label
dict[grepl("sexo", dict$label, ignore.case = TRUE), ]

# Inspect value labels for a specific variable
sex_var <- match("SEXO", dict$variable)
if (!is.na(sex_var)) {
  dict$value_labels[[sex_var]]
}
```

get_metadata *Get metadata of a CIS study*

Description

Retrieves the technical metadata of a CIS study from its detail page, including study dates, type, country, author, and thematic indices.

Usage

```
get_metadata(study_code)
```

Arguments

study_code	A string with the study code.

Value

A tibble with two columns: `field` and `value`.

Examples

```
if (interactive()) {
# Get metadata for study 3328
meta <- get_metadata("3328")
print(meta)

# Access a specific field
meta$value[meta$field == "Tipo de estudio"]
}
```

read_cis *Import a CIS study*

Description

Download and import the data of a CIS study.

Usage

```
read_cis(study_code)
```

Arguments

study_code	A string with the study code.

Value

A data.frame with the study data.

Examples

```
if (interactive()) {
# If you know the study code you can just read it into R
df <- read_cis("3328")
print(df)

# If you dont know the study code, you can search for a study using search_cis() function:
studies <- search_cis(q = "gastronomia")
print(studies)

df <- read_cis(studies$study[1])
print(df)
}
```

`search_all_cis` *Search all CIS results with automatic pagination*

Description

Calls `search_cis` repeatedly, incrementing the page index until no more results are returned, and returns all results in a single tibble.

Usage

```
search_all_cis(
  q = "",
  from = NULL,
  to = NULL,
  sort = "relevance",
  catalogo = "estudio",
  ...
)
```

Arguments

`q`	String. The search query. Default is an empty string.
`from`	Date or NULL. The start date for filtering results. Default is NULL. The date format must be "YYYY-MM-DD".
`to`	Date or NULL. The end date for filtering results. Default is NULL. The date format must be "YYYY-MM-DD".
`sort`	String. The sorting order for the results (`"publishDate-"`, `"publishDate+"`, `"relevance"`). Default is `"relevance"`.
`catalogo`	String. The catalog type (`"estudio"`, `"pregunta"`, `"serie"`). Default is `"estudio"`.
`...`	Additional parameters passed to `search_cis`.

Value

A tibble with all search results across all pages.

Examples

```
if (interactive()) {
# Retrieve all postelectoral studies (all pages)
all_studies <- search_all_cis(q = "postelectoral")
print(nrow(all_studies))

# Filter by date range
studies_2010_2020 <- search_all_cis(
  q    = "ideologia",
  from = "2010-01-01",
  to   = "2020-12-31"
)
print(studies_2010_2020)
}
```

---

search_cis *Search for CIS studies.*

Description

Searches for CIS studies using the CIS search engine.

Usage

```
search_cis(
  start = 1,
  q = "",
  from = NULL,
  to = NULL,
  sort = "relevance",
  catalogo = "estudio",
  ...
)
```

Arguments

start	Integer. The starting page for the search results. Default is 1, iterate to get more results.
q	String. The search query. Default is an empty string.
from	Date or NULL. The start date for filtering results. Default is NULL. The date format must be "YYYY-MM-DD".
to	Date or NULL. The end date for filtering results. Default is NULL. The date format must be "YYYY-MM-DD".

sort	String. The sorting order for the results ("publishDate-", "publishDate+", "relevance"). Default is "relevance".
catalogo	String. The catalog type ("estudio", "pregunta", "serie"). Default is "estudio".
...	Additional parameters (not used).

Value

A data.frame with the search results.

Examples

```
if (interactive()) {
# Search by search terms
studies <- search_cis(q = "postelectoral")
print(studies)

# Narrow the search by dates
studies <- search_cis(q = "postelectoral",
                          from = "2011-01-01",
                          to = "2020-01-01")
print(studies)

# Use the catalogo parameter to search for questions ("pregunta") or data series ("serie")
studies <- search_cis(q = "ideologia",
                          from = "2011-01-01",
                          to = "2020-01-01",
                          catalogo = "serie")
print(studies)
}
```

Index

browse_pdf, 2

cache_info, 3
clear_cache, 3

download_study, 4

get_data_dictionary, 5
get_metadata, 6

read_cis, *3*, *5*, 6

search_all_cis, 7
search_cis, *3*, *7*, 8