Package: sage (via r-universe)

October 15, 2026

Type Package

Title Retrieve Slices of the Small-Area Global Elections Archive

Version 0.2.0

Description Downloads and subsets slices of the Small-Area Global Elections (SAGE) archive, a harmonized collection of polling-station and precinct-level election results, without having to read monolithic country-level files. Reads partitioned 'Parquet' from a public bucket (configurable through the 'SAGE_SOURCE' environment variable, the 'sage_set_source()' function, or a local copy of the archive) and exposes a 'sage_load(country, years, columns)' function returning a 'tibble', plus helpers for the accompanying boundary polygons and candidate-level vote sidecars. See <https://github.com/noahdasanaike/sage> for the archive itself.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.0)

Imports arrow (>= 14.0.0), dplyr, rlang, tibble, utils

Suggests curl, sf, sfarrow, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

URL https://github.com/noahdasanaike/sage

BugReports https://github.com/noahdasanaike/sage/issues

NeedsCompilation no

Author Noah Dasanaike [aut, cre, cph], Dmitry Shkolnik [ctb]

Maintainer Noah Dasanaike <dasanaike.noah@gmail.com>

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

Date/Publication 2026-10-15 16:40:02 UTC

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

RemoteRef HEAD

RemoteSha 79d90bd492b55db9891565b726073f09b2427973

Contents

sage 2
sage_columns 3
sage_countries 4
sage_load 4
sage_polygons 5
sage_preference_votes 6
sage_set_source 7
sage_years 8

Index **9**

`sage` *Small-Area Global Elections (SAGE) Data Retrieval*

Description

These helpers fetch partitioned Parquet from the SAGE public release (default: a Google Cloud Storage bucket; override with `sage_set_source()` or the SAGE_SOURCE environment variable) and let users pull (country, years, columns) slices without reading the giant per-country RDS files.

Details

Remote reads go over plain HTTPS (one partition file at a time, guided by the `_index.csv` manifest), so the package works with any build of **arrow** – it does not require Arrow's S3/GCS cloud filesystem to be compiled in.

When the source is the public bucket and **arrow** does have GCS support compiled in, `sage_columns()` reads Parquet schemas through Arrow's random-access reader instead of downloading whole partitions. That is an optimization only: it falls back to the HTTPS download path whenever the cloud filesystem or an individual read is unavailable.

Source layout

The release is a hive-partitioned Parquet dataset:

```
<source>/
  country=Afghanistan/year=2014/part-0.parquet
  country=Afghanistan/year=2019/part-0.parquet
  country=Albania/year=2009/part-0.parquet
  ...
  _index.csv
```

so a (country, year) slice is a single partition read.

Offline excerpt

Every example in this package runs against a small excerpt of the archive (Iceland and Samoa) shipped under `extdata/mini_archive`, so no network access is required to try the interface:

```
sage_set_source(system.file("extdata", "mini_archive", package = "sage"))
```

Point `sage_set_source()` back at `"gs://sage-archive/parquet"` to use the full release.

---

sage_columns *List the columns available in the dataset*

Description

Not every column is populated for every country; this returns the union of columns over the largest partition of each of several countries, which is what surfaces the optional deep-hierarchy columns (`NAME4`, `NAME5`, ...). The answer is cached for the session, and is discarded when `sage_set_source()` changes the source.

Usage

```
sage_columns()
```

Value

A character vector of column names, beginning with `"country"` and `"year"`.

See Also

`sage_load()`, whose `columns` argument takes these names.

Examples

```
sage_set_source(system.file("extdata", "mini_archive", package = "sage"))
sage_columns()
```

sage_countries — *List countries available in the current source*

Description

Reads the `_index.csv` manifest that ships with the archive, so this costs one small request rather than a scan of the bucket.

Usage

```
sage_countries()
```

Value

A character vector of country names, sorted.

See Also

`sage_years()`, `sage_load()`

Examples

```
sage_set_source(system.file("extdata", "mini_archive", package = "sage"))
sage_countries()
```

sage_load — *Load a (country, years, columns) slice of SAGE*

Description

Reads only the partitions the request needs, one file per (country, year), and returns them as a single tibble.

Usage

```
sage_load(
  country,
  years = NULL,
  columns = NULL,
  election_type = NULL,
  min_match_confidence = NULL,
  drop_all_na = TRUE
)
```

Arguments

`country`	Single country name; required.
`years`	Optional integer vector of years (default: all available).
`columns`	Optional character vector of column names (default: all).
`election_type`	Optional character filter on `election_type` (e.g., `"Presidential"`).
`min_match_confidence`	If non-NULL, restricts to rows whose `match_confidence` is at least this strict (`"high"` keeps only high; `"medium"` keeps high+medium; etc.).
`drop_all_na`	Drop columns that are entirely NA for the slice. The union schema carries every column any partition has, so a country slice would otherwise show columns that never apply to it.

Value

A tibble, with `country` and `year` as the first two columns.

See Also

`sage_columns()` for the available columns, `sage_polygons()` for boundaries.

Examples

```
sage_set_source(system.file("extdata", "mini_archive", package = "sage"))
is_2020 <- sage_load("Iceland", years = 2020)
dim(is_2020)
names(is_2020)

# Restrict to a few columns; country and year are always kept.
sage_load("Iceland", years = 2020, columns = c("party", "votes"))

## Not run:
# Against the full release (requires network access).
sage_set_source("gs://sage-archive/parquet")
de_2021 <- sage_load("Germany", years = 2021, columns = c("party", "votes", "NAME3"))
all_de_high <- sage_load("Germany", min_match_confidence = "high")

## End(Not run)
```

---

sage_polygons *Load the polygon set for one country*

Description

SAGE ships a parallel `polygons/` tree (one geoparquet per country) alongside the main parquet release; the polygons carry only the `NAME`-key + `year` + `geometry` columns and are deduplicated to one row per unique polygon. Use this helper when you need boundaries for choropleth mapping; the main `sage_load()` only returns lat/long.

Usage

```
sage_polygons(country, years = NULL)
```

Arguments

country	Country name.
years	Optional integer vector of years.

Details

Requires the **sf** and **sfarrow** packages.

Value

An sf data frame, or NULL invisibly if the source has no polygons for the country.

See Also

sage_load() for the vote data these boundaries join to.

Examples

```
## Not run:
# Requires network access, plus the sf and sfarrow packages; polygons are
# not part of the offline excerpt.
sage_set_source("gs://sage-archive/parquet")
is_poly <- sage_polygons("Iceland", years = 2021)

## End(Not run)
```

---

sage_preference_votes *Load the per-candidate vote sidecar*

Description

Some countries report votes for individual candidates alongside the party totals in the main release. This reads that sidecar. It is available for Germany (Erststimme by Wahlkreis) and the Netherlands (preference votes by stembureau); any other country is an error.

Usage

```
sage_preference_votes(country)
```

Arguments

country	Country name; one of "Germany" or "Netherlands".

Value

A tibble of candidate-level votes.

See Also

sage_load() for the main party-level release.

Examples

```
## Not run:
# Requires network access; the sidecar is not part of the offline excerpt.
sage_set_source("gs://sage-archive/parquet")
de_cand <- sage_preference_votes("Germany")

## End(Not run)
```

---

sage_set_source *Set the source URL or path for the SAGE archive*

Description

Sets the archive every other function in the package reads from. The setting is global to the session and persists until it is changed again; unset, the package falls back to the SAGE_SOURCE environment variable and then to the public bucket.

Usage

```
sage_set_source(source)
```

Arguments

source	A local path or remote URI (e.g., "gs://bucket/path", "https://example.com/sage_parquet", or a local directory). A trailing slash on a remote URI is ignored.

Value

The source, invisibly.

See Also

sage_load() to read a slice from the source that is set here.

Examples

```
# Point the package at the offline excerpt that ships with it.
sage_set_source(system.file("extdata", "mini_archive", package = "sage"))
sage_countries()

## Not run:
# The full public release (requires network access).
sage_set_source("gs://sage-archive/parquet")

## End(Not run)
```

`sage_years` *List election years available for a country*

Description

List election years available for a country

Usage

```
sage_years(country)
```

Arguments

`country`	Country name (e.g., `"Germany"`), as returned by `sage_countries()`.

Value

An integer vector of years, sorted. A country with no partitions in the source gives a zero-length vector.

See Also

`sage_countries()`, `sage_load()`

Examples

```
sage_set_source(system.file("extdata", "mini_archive", package = "sage"))
sage_years("Iceland")
```

Index

sage, 2
sage_columns, 3
sage_columns(), *5*
sage_countries, 4
sage_countries(), *8*
sage_load, 4
sage_load(), *3, 4*, *6–8*
sage_polygons, 5
sage_polygons(), *5*
sage_preference_votes, 6
sage_set_source, 7
sage_set_source(), *3*
sage_years, 8
sage_years(), *4*