

Package: infosigasp (via r-universe)

October 21, 2026

Type Package

Title Download and Import Traffic Crash Data from 'INFOSIGA-SP'

Version 0.2.0

Description Provides a programmatic interface to the open data published by the Sao Paulo State Traffic Accident Information and Management System ('INFOSIGA-SP'), maintained by the Sao Paulo State Department of Motor Vehicles ('DETRAN-SP'). Functions download and import tidy data frames of traffic crash events ('sinistros'), victims ('pessoas') and vehicles ('veiculos') from 2015 onward, handling the source encoding, decimal marks, date formats and on-disk caching. See <https://infosiga.detran.sp.gov.br/> for the original data portal.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports cli, dplyr (>= 1.2.0), readr (>= 2.0.0), sf, stringr, tibble, tools, utils

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/Needs/website reactable, htmltools

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL <https://github.com/viniciusoike/infosigasp>,
<https://viniciusoike.github.io/infosigasp/>

BugReports <https://github.com/viniciusoike/infosigasp/issues>

VignetteBuilder knitr

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository <https://viniciusoike.r-universe.dev>
Date/Publication 2026-09-20 22:15:07 UTC
RemoteUrl <https://github.com/viniciusoike/infosigasp>
RemoteRef HEAD
RemoteSha e3a6f8c884da8446ae966a1e5f6b2f1d9ce6e74d

Contents

clear_infosiga_cache	2
dictionary_infosiga	3
infosiga_cache_info	3
read_infosiga	4
Index	7

clear_infosiga_cache	<i>Clear the INFOSIGA-SP cache</i>
----------------------	------------------------------------

Description

Removes package-managed cache entries. By default, only canonical cleaned datasets are removed; the downloaded source archive is retained. Files outside the package’s managed cache paths are never touched.

Usage

```
clear_infosiga_cache(processed = TRUE, source = FALSE, quiet = FALSE)
```

Arguments

- processed Logical. Remove all canonical cleaned datasets.
- source Logical. Also remove the downloaded source ZIP archive.
- quiet Logical. Suppress the completion message.

Value

Invisibly, the paths that were selected for removal.

See Also

[infosiga_cache_info\(\)](#) and [read_infosiga\(\)](#).

Examples

```
temporary_cache <- tempfile("infosigasp-example-")  
old_options <- options(infosigasp.cache_dir = temporary_cache)  
clear_infosiga_cache(quiet = TRUE)  
options(old_options)
```

dictionary_infosiga *Open the INFOSIGA-SP data dictionary*

Description

Opens the package's searchable online data dictionary. Set `source = "official"` to open the INFOSIGA-SP source website instead.

Usage

```
dictionary_infosiga(
  dataset = NULL,
  source = c("online", "official"),
  open = interactive()
)
```

Arguments

dataset	Optional dataset to link to: one of "sinistros", "pessoas" or "veiculos". The online dictionary opens at that dataset's section. This argument has no effect when <code>source = "official"</code> .
source	Which documentation to open. "online" (default) uses the searchable dictionary on the package website. "official" uses the INFOSIGA-SP website, where DETRAN-SP publishes the original files.
open	Logical. If TRUE (the default in interactive sessions), open the URL in a browser.

Value

The selected URL, invisibly.

Examples

```
dictionary_infosiga(open = FALSE)
dictionary_infosiga("sinistros", open = FALSE)
dictionary_infosiga(source = "official", open = FALSE)
```

infosiga_cache_info *Inspect the INFOSIGA-SP cache*

Description

Lists the source archive and canonical cleaned datasets currently stored in the package cache. Merely inspecting the cache does not create any files or directories.

Usage

```
infosiga_cache_info()
```

Value

A tibble with the cache entry type, dataset, absolute path, size in MiB and modification time. Returns an empty tibble when the cache is empty.

See Also

`read_infosiga()` and `clear_infosiga_cache()`.

Examples

```
infosiga_cache_info()
```

read_infosiga	<i>Import an INFOSIGA-SP dataset</i>
---------------	--------------------------------------

Description

Downloads (if necessary) and imports one of the three INFOSIGA-SP datasets as a tidy tibble. The first interactive call asks before downloading about 120 MB to the user's local cache; subsequent calls read from disk.

Usage

```
read_infosiga(
  dataset = c("sinistros", "pessoas", "veiculos"),
  processing = c("clean", "typed", "raw"),
  standardize = NULL,
  refresh = FALSE,
  quiet = FALSE,
  cache = TRUE
)
```

Arguments

dataset	Which dataset to import. "sinistros" Occurrence records: confirmed crashes and notifications (one row per record). "pessoas" Victims / people involved (one row per person). "veiculos" Vehicles involved (one row per vehicle).
processing	Level of processing to apply. "raw" imports every field as character and preserves empty strings, whitespace, sentinels and source representations. "typed" parses the documented column classes without applying additional cleaning. "clean" (default) adds the package's cleaning pipeline to the typed import.

standardize	Optional character vector selecting label harmonisation: "municipios" restores official municipality spellings and harmonises administrative-region names; "cores" merges duplicate vehicle-colour spellings; and "profissoes" applies consistent title case and missing-value markers to occupations. Use "all" for every option applicable to the selected dataset. Requires processing = "clean".
refresh	Logical. If TRUE, download the latest available source data before reading. If FALSE (default), reuse the copy in the local infosigasp cache, downloading it only when it is missing.
quiet	Logical. If FALSE (default), report progress.
cache	Logical. If TRUE (default), canonical results from processing = "clean" are stored in the package's managed user cache and reused on subsequent calls. Raw, typed and standardised variants are never stored as processed artifacts. Set to FALSE to bypass the processed cache; the downloaded source archive is managed separately.

Details

Source files are encoded in Latin-1 (ISO-8859-1), use ; as the field separator, , as the decimal mark and DD/MM/YYYY dates. Every mode decodes text to UTF-8, parses the CSV structure and row-binds the period files. The modes differ in what happens to the fields after that tabular import.

- processing = "raw" returns a lossless tabular representation: every field is character, including dates and numbers, while empty strings, whitespace, sentinels and malformed representations remain visible. This is not a byte-for-byte copy because encoding is decoded and period files are combined.
- processing = "typed" parses the documented column classes. Dates become Date, times become hms, numeric fields become integer or double, empty fields become NA, and identifiers remain character. Category labels, padding and explicit source sentinels otherwise remain unchanged.
- processing = "clean" starts from the typed import, then trims text, maps "NAO DISPONIVEL" to NA, orders ordinal columns, parses ano_mes_*, converts crash-type flags to logical, converts tempo_sinistro_obito to integer, removes a trailing ".0" from numero_logradouro, and validates coordinate pairs against the Sao Paulo state boundary with a 2 km buffer. Missing qtd_* counts remain NA.

Canonical clean results are cached separately from the source ZIP. Cache entries are keyed by the source archive checksum and an internal cleaning schema version. Obsolete entries for a dataset are removed after a new entry is written. Standardisation is always applied in memory after loading the canonical clean result. Use [infosiga_cache_info\(\)](#) to inspect disk use and [clear_infosiga_cache\(\)](#) to remove processed entries.

Before converting a closed-domain column, the cleaning step validates its observed values. If an ordinal column, crash-type flag or integer field contains an unexpected representation, the entire source column is preserved and a warning identifies the new values. This prevents upstream changes from becoming missing values or incorrect FALSE values silently.

Label harmonisation is selective and opt-in. standardize = "municipios" uses official IBGE municipality names keyed by cod_ibge and harmonises the spelling of INFOSIGA administrative regions. "cores" merges case and gender variants of basic vehicle colours, while preserving detailed

liveries and multi-tone values. "profissoes" applies consistent title case and known missing-value markers to occupation labels. These transformations preserve the rows and columns and never create broader analytical categories.

A small fraction of rows in the source contain data-quality issues (for example, an unescaped ; inside a street name, or mis-encoded coordinates). In typed and clean modes, values that cannot be parsed to their declared column type become NA and are recorded by `readr::problems()`. Raw mode preserves those field values as character. Structural CSV problems are recorded in every mode.

Coverage, field definitions and analytical caveats are documented in the searchable online data dictionary returned by `dictionary_infosiga()`. In particular, data from 2015–2018 cover fatal crashes only, while data from 2019 onward also include non-fatal crashes and notifications.

Value

A `tibble` with one row per record. The columns keep the original INFOSIGA-SP names (in Portuguese); see the package data dictionary via `dictionary_infosiga()`. The three datasets can be joined on `id_sinistro` (and `id_veiculo`, where present).

See Also

`dictionary_infosiga()`.

Examples

```
## Not run:
# Import all occurrence records, cleaned (downloads on first use)
sinistros <- read_infosiga("sinistros")
levels(sinistros$dia_da_semana)

# Import all victims / people involved
vitimas <- read_infosiga("pessoas")

# Lossless tabular import: every field is character
raw <- read_infosiga("sinistros", processing = "raw")

# Parse documented classes without further cleaning
typed <- read_infosiga("sinistros", processing = "typed")

## End(Not run)

# A bundled sample (no download required) illustrates the structure:
sample_path <- system.file(
  "extdata", "sinistros_sample.csv",
  package = "infosigasp"
)
if (nzchar(sample_path)) head(readr::read_delim(sample_path, ";"))
```

Index

`clear_infosiga_cache`, [2](#)
`clear_infosiga_cache()`, [4](#), [5](#)

`dictionary_infosiga`, [3](#)
`dictionary_infosiga()`, [6](#)

`infosiga_cache_info`, [3](#)
`infosiga_cache_info()`, [2](#), [5](#)

`read_infosiga`, [4](#)
`read_infosiga()`, [2](#), [4](#)
`readr::problems()`, [6](#)

`tibble`, [6](#)