Package: infosigasp (via r-universe)

June 21, 2026

Type Package

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

Version 0.1.0

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, readr (>= 2.0.0), tibble, tools, utils

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

Config/testthat/edition 3

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

LazyData false

Config/pak/sysreqs libx11-dev

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

Date/Publication 2026-06-21 13:39:41 UTC

RemoteUrl https://github.com/viniciusoike/infosigasp

RemoteRef HEAD

RemoteSha a7e19dc8ba9640783ce00816d5870af00d523fd1

Contents

`clean_infosiga` *Clean and process an INFOSIGA-SP dataset*

Description

Applies the standard processing that `read_infosiga()` performs by default (`clean = TRUE`). Use this directly only when you imported a dataset with `clean = FALSE` (the raw version) and want to process it afterwards.

Usage

```
clean_infosiga(data, dataset = c("sinistros", "pessoas", "veiculos"))
```

Arguments

`data`	A data frame imported with `read_infosiga()` (typically with `clean = FALSE`).
`dataset`	Which dataset `data` corresponds to: `"sinistros"`, `"pessoas"` or `"veiculos"`. Determines which columns are processed.

Details

The processing is deliberately light: it standardises missing values, fixes source formatting artefacts and assigns meaningful types to columns whose published representation is inconvenient (ordinal text, binary flags, year-month strings). It never renames columns, recodes category labels or drops rows, so the result stays a faithful, analysis-ready view of the source.

The following steps are applied, in order. Every step is idempotent, so `clean_infosiga()` can be called again on an already-processed dataset without changing it.

1. **Whitespace.** Leading and trailing whitespace is trimmed from every text column. Some source fields are space-padded to a fixed width (for example `nacionalidade` is published as `"BRASILEIRA "`); without trimming, comparisons, grouping and joins on those columns silently fail.
2. **Missing values.** The literal `"NAO DISPONIVEL"` ("not available") marker is replaced by `NA` in every text column. Trimming happens first so that space-padded markers are also caught.
3. **Ordered factors.** Ordinal columns are converted to **ordered factors** with their natural order:
   - `dia_da_semana`: `Domingo` < ... < `Sabado` (the Brazilian week starts on Sunday).
   - `turno`: `MADRUGADA` < `MANHA` < `TARDE` < `NOITE`.
   - `gravidade_lesao` (in `pessoas`): `LEVE` < `GRAVE` < `FATAL`.
   - `faixa_etaria_demografica`, `faixa_etaria_legal` (in `pessoas`): age bands in increasing order.
4. **Year-month dates.** Year-month columns (`ano_mes_sinistro`, `ano_mes_obito`), published as `"YYYY/MM"` strings, are parsed to first-of-month `Date` values, matching the `Date` class already used for the full-date columns.
5. **Crash-type flags** (`sinistros`). The binary `tp_sinistro_*` columns – which mark whether a crash involved a given event type and are published as `"S"` (yes) or empty (no) – become **logical** (`TRUE` / `FALSE`). The categorical `tp_sinistro_primario` (the primary crash type, e.g. `"COLISAO"`) is *not* a flag and is left as text.
6. **Days to death** (`pessoas`). `tempo_sinistro_obito`, the number of days between the crash and the victim's death (published as a numeric string), becomes **integer**.
7. **Street numbers** (`sinistros`). `numero_logradouro` is kept as text (house numbers may contain letters), but a spurious trailing `".0"` from the source export (`"193.0"`) is stripped to `"193"`.
8. **Coordinates** (`sinistros`). `latitude`/`longitude` are validated as a pair against the bounding box of the state of Sao Paulo. Points outside the box – mis-encoded values and `(0, 0)` "null island" placeholders – have both coordinates set to `NA`. This affects roughly 7% of records; no rows are dropped. Use `clean = FALSE` if you need the raw coordinates.

Nominal text columns (such as `municipio`, `tipo_via` or `sexo`) are left as character vectors. Numeric columns that are already well typed – notably `idade` (the victim's age, in `pessoas`) – are passed through unchanged and are *not* range-checked: missing ages are `NA`, and ages of `0` (infants) are kept. In the current upstream data `idade` ranges from 0 to about 102, but the package does not enforce any bound, so validate it yourself if your analysis is sensitive to outliers.

Value

A tibble with the same columns as `data`, with the processing described in *Details* applied.

See Also

`read_infosiga()`, which calls this function when `clean = TRUE`.

Examples

```
# Process the bundled raw sample
raw <- readr::read_delim(
  system.file("extdata", "pessoas_sample.csv", package = "infosigasp"),
```

```
  delim = ";", show_col_types = FALSE
)
clean <- clean_infosiga(raw, "pessoas")
levels(clean$gravidade_lesao)
```

`infosiga_cache` *Manage the infosigasp on-disk cache*

Description

INFOSIGA-SP ships its data as a single archive of roughly 120 MB (uncompressed, over 700 MB). To avoid repeated downloads, infosigasp stores the archive in a per-user cache directory and reuses it across sessions. These functions inspect and manage that cache.

Usage

```
infosiga_cache_dir()

infosiga_cache_list()

infosiga_cache_clear(confirm = interactive())
```

Arguments

`confirm`	Logical. If `TRUE` (the default in interactive sessions), ask for confirmation before deleting cached files. Set to `FALSE` to delete without prompting (e.g. in scripts).

Details

The cache location defaults to the operating-system specific user cache directory returned by `tools::R_user_dir()` (`"infosigasp"`, `"cache"`). You can override it for the current session with the `infosigasp.cache_dir` option, e.g. `options(infosigasp.cache_dir = "~/my-cache")`, or permanently through your `.Rprofile`.

Value

- `infosiga_cache_dir()` returns the cache directory path (a string). It is a pure accessor with no side effects: the directory itself is created lazily the first time data is written (e.g. by `infosiga_download()`), so the reported path may not yet exist.
- `infosiga_cache_list()` returns a character vector of cached file paths (possibly empty).
- `infosiga_cache_clear()` invisibly returns the paths it removed.

Examples

```
# Where does infosigasp cache its files?
infosiga_cache_dir()

# What is currently cached?
infosiga_cache_list()
```

`infosiga_datasets` *List the available INFOSIGA-SP datasets*

Description

Returns a small tibble describing the datasets that `read_infosiga()` can import, including their grain (what one row represents) and key columns.

Usage

```
infosiga_datasets()
```

Value

A tibble with columns `dataset`, `description`, `grain` and `keys`.

Examples

```
infosiga_datasets()
```

`infosiga_dictionary` *Download the INFOSIGA-SP data dictionary*

Description

Downloads the official INFOSIGA-SP data dictionary, a set of PDF documents (one per dataset) describing every column and its accepted values. The archive is saved to the cache and the extracted PDF paths are returned.

Usage

```
infosiga_dictionary(
  dest = file.path(infosiga_cache_dir(), "dictionary"),
  overwrite = FALSE,
  quiet = FALSE
)
```

Arguments

dest	Directory in which to extract the PDF files. Defaults to a dictionary sub-folder of infosiga_cache_dir().
overwrite	Logical. Re-download even if the dictionary archive is already cached. Defaults to FALSE.
quiet	Logical. Suppress progress messages. Defaults to FALSE.

Value

A character vector of paths to the extracted PDF files, invisibly.

Examples

```
## Not run:
pdfs <- infosiga_dictionary()
# Open the dictionary for the crash-events dataset
browseURL(grep("sinistros", pdfs, value = TRUE))

## End(Not run)
```

---

infosiga_download *Download the INFOSIGA-SP source archive*

Description

Downloads the consolidated INFOSIGA-SP data archive (dados_infosiga.zip) from DETRAN-SP into the local cache. Most users do not need to call this directly: read_infosiga() downloads the archive on demand. Use this function when you want to pre-fetch the data (for example, before going offline) or to force a refresh.

Usage

```
infosiga_download(overwrite = FALSE, quiet = FALSE, timeout = 3600)
```

Arguments

overwrite	Logical. If FALSE (default) and the archive is already cached, the existing file is kept and returned. Set to TRUE to download again and replace it.
quiet	Logical. If FALSE (default), report progress with informative messages.
timeout	Download timeout in seconds. The archive is large (around 120 MB), so the default temporarily raises options()$timeout to 3600. Pass a larger value on slow connections.

Details

The archive is updated monthly by DETRAN-SP and accumulates all records from 2015 onward. The download URL can be overridden with the `infosigasp.zip_url` option, which may be a character vector of mirror URLs tried in order until one succeeds. The default is the official DETRAN-SP endpoint followed by a GitHub-release mirror that serves a point-in-time snapshot when the official portal is unavailable. Override the option to add your own mirror or for testing.

Because DETRAN-SP overwrites the archive in place each month under the same file name, a cached copy can become stale silently. When a cached archive is reused that is older than the `infosigasp.stale_days` option (30 days by default; set to `Inf` to disable), a warning suggests refreshing it. The age is taken from the cached file's modification time.

Value

The path to the cached archive, invisibly.

See Also

`read_infosiga()` to import the data, and `infosiga_cache_dir()` to locate the cache.

Examples

```
## Not run:
# Pre-fetch the archive into the cache
infosiga_download()

# Force a refresh after a monthly update
infosiga_download(overwrite = TRUE)

## End(Not run)
```

---

read_infosiga *Import an INFOSIGA-SP dataset*

Description

Downloads (if necessary) and imports one of the three INFOSIGA-SP datasets as a tidy tibble. The source archive is cached locally, so the first call triggers a download and subsequent calls read from disk.

Usage

```
read_infosiga(
  dataset = c("sinistros", "pessoas", "veiculos"),
  clean = TRUE,
  year = NULL,
  download_if_missing = TRUE,
  quiet = FALSE,
  ...
)
```

Arguments

dataset	Which dataset to import. One of: "sinistros" Crash events (one row per event). "pessoas" Victims / people involved (one row per person). "veiculos" Vehicles involved (one row per vehicle).
clean	Logical. If TRUE (default), return a processed dataset: text is trimmed, the "NAO DISPONIVEL" marker becomes NA, ordinal columns become ordered factors, crash-type flags become logical, and impossible coordinates are dropped (see clean_infosiga() for the full list of steps). If FALSE, return the raw data exactly as published, with all text columns as character vectors.
year	Optional integer vector used to filter rows by year of the crash (ano_sinistro). If NULL (default), all available years are returned. For example, year = 2020:2023.
download_if_missing	Logical. If TRUE (default), download the archive when it is not already cached. If FALSE and the archive is missing, an informative error is raised.
quiet	Logical. If FALSE (default), report progress.
...	Additional arguments passed to infosiga_download() (for example overwrite = TRUE to force a refresh).

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. read_infosiga() handles all of these and returns UTF-8 text, Date columns and numeric coordinates. Each dataset is distributed across two period files inside the archive (2015-2021 and 2022 onward); they are read and row-bound transparently.

By default (clean = TRUE) the result is then processed by clean_infosiga(): text columns are whitespace-trimmed, the "NAO DISPONIVEL" ("not available") marker becomes NA, ordinal columns (dia_da_semana, turno, gravidade_lesao, the age bands) become **ordered factors**, the ano_mes_* year-month strings are parsed to first-of-month Dates, the binary tp_sinistro_* crash-type flags become **logical**, tempo_sinistro_obito becomes **integer**, and latitude/longitude values outside the bounding box of Sao Paulo state are dropped to NA. See clean_infosiga() for the complete, ordered list. Pass clean = FALSE to obtain the raw data exactly as published – every text column kept as a character vector, with "NAO DISPONIVEL" and the source's fixed-width whitespace padding preserved verbatim.

A small fraction of rows in the source contain data-quality issues (for example, an unescaped ; inside a street name, or mis-encoded coordinates). Any value that cannot be parsed to its declared column type is set to NA and recorded by readr::problems(). Empty fields are read as NA in both modes. In the raw data (clean = FALSE) the crash-type flag columns (tp_sinistro_*) hold "S" when the flag applies and NA otherwise; with clean = TRUE they are converted to logical.

Value

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

See Also

infosiga_download(), infosiga_cache_dir(), infosiga_dictionary().

Examples

```
## Not run:
# Import all crash events, processed (downloads the archive on first use)
sinistros <- read_infosiga("sinistros")
levels(sinistros$dia_da_semana)

# Only victims from 2022 and 2023
vitimas <- read_infosiga("pessoas", year = 2022:2023)

# The raw data, exactly as published
raw <- read_infosiga("sinistros", clean = FALSE)

## 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