

Package ‘microdatasus’

July 29, 2026

Title Download and Process 'DataSUS' Files

Version 3.0.0

URL <https://github.com/rfsaldanha/microdatasus>,
<https://rfsaldanha.github.io/microdatasus/>

BugReports <https://github.com/rfsaldanha/microdatasus/issues>

Description Downloads and processes health microdata from Brazilian Unified Health System ('DataSUS') information systems. It handles the compressed 'DBC' format internally and provides functions to organize variables and add labels to categorical fields from mortality, live births, hospital admissions, outpatient care, health facilities, and notifiable diseases data. For details, see Saldanha et al. (2019)
<[doi:10.1590/0102-311x00032419](https://doi.org/10.1590/0102-311x00032419)>.

License MIT + file LICENSE

Copyright See file inst/COPYRIGHTS.

Encoding UTF-8

LazyData true

LazyDataCompression xz

Imports checkmate, cli, curl, data.table, dplyr (>= 1.2.0), foreign,
magrittr, rlang, stringi, tibble, utils, zip

Suggests lubridate, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1.0)

Config/Needs/website rmarkdown

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Raphael Saldanha [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-0652-8466>>),
Sidney da Silva Pereira Bissoli [ctb, cph] (DBC decompression code),
Mark Adler [ctb, cph] (blast decompressor)

Maintainer Raphael Saldanha <raphael.saldanha@fiocruz.br>

Repository CRAN

Date/Publication 2026-07-29 13:40:02 UTC

Contents

microdatasus-package	3
cnes_pf_sample	4
cnes_st_sample	4
equipe	5
fetch_cadger	5
fetch_datasus	6
fetch_sigtab	8
paisnet	9
process_cnes	9
process_sia	10
process_sih	12
process_sim	13
process_sinan_chagas	14
process_sinan_chikungunya	15
process_sinan_dengue	16
process_sinan_leishmaniose_tegumentar	17
process_sinan_leishmaniose_viscerar	18
process_sinan_malaria	19
process_sinan_zika	20
process_sinasc	21
read_dbc	22
sia_pa_sample	23
sigtab	23
sih_rd_sample	23
sim_do_sample	24
sinan_chagas_sample	24
sinan_chikungunya_sample	24
sinan_dengue_sample	25
sinan_leishmaniose_tegumentar_sample	25
sinan_leishmaniose_viscerar_sample	25
sinan_malaria_sample	26
sinan_zika_sample	26
sinasc_sample	26
tabCBO	27
tabMun	27
tabNaturalidade	28
tabOcupacao	28

Index

29

microdatasus-package *microdatasus: Download and prepare DataSUS microdata*

Description

Provides a reproducible workflow for DataSUS microdata: `fetch_datasus()` discovers, downloads, and combines published DBC files; `read_dbc()` reads a local DBC file; and the `process_*`() functions convert system-specific fields into analysis-ready values and labels.

Details

Start with `fetch_datasus()` and then use the processor corresponding to the selected system, such as `process_sim()`, `process_sinasc()`, `process_sih()`, `process_sia()`, `process_cnes()`, or one of the `process_sinan_*`() functions. `fetch_cadger()` and `fetch_sigtab()` retrieve current auxiliary tables used by CNES and SIA processing.

For concepts, coverage, data flows, and caveats of each Brazilian health information system, see Saldanha (2026), *Sistemas de Informação em Saúde no Brasil*.

Author(s)

Maintainer: Raphael Saldanha <raphael.saldanha@fiocruz.br> ([ORCID](#))

Authors:

- Raphael Saldanha <raphael.saldanha@fiocruz.br> ([ORCID](#))

Other contributors:

- Sidney da Silva Pereira Bissoli (DBC decompression code) [contributor, copyright holder]
- Mark Adler (blast decompressor) [contributor, copyright holder]

See Also

Useful links:

- <https://github.com/rfsaldanha/microdatasus>
- <https://rfsaldanha.github.io/microdatasus/>
- Report bugs at <https://github.com/rfsaldanha/microdatasus/issues>

cnes_pf_sample	<i>CNES-PF sample</i>
----------------	-----------------------

Description

CNES-PF sample with the first 100 rows from AC 2016-6 file.

Usage

cnes_pf_sample

Format

An object of class `data.frame` with 100 rows and 39 columns.

Details

The variable `CPF_PROF` was removed due non-ASCII characters.

cnes_st_sample	<i>CNES-ST sample</i>
----------------	-----------------------

Description

CNES-ST sample with the first 100 rows from AC 2016-6 file.

Usage

cnes_st_sample

Format

An object of class `data.frame` with 100 rows and 201 columns.

equipe	<i>Equipe table</i>
--------	---------------------

Description

Equipe table, based on INE_EQUIPE_BR.dbf file from SIM TabWIN definitions file.

Usage

equipe

Format

A data frame with 2428 rows and 2 variables:

COD Code

equipe_ref Label

fetch_cadger	<i>Download the current CADGER table</i>
--------------	--

Description

Downloads and reads the current CNES establishment-name table distributed by DataSUS. [process_cnes\(\)](#) can use this table to add establishment names.

Usage

```
fetch_cadger(timeout = 240)
```

Arguments

timeout A positive numeric scalar. Download and connection timeout, in seconds.

Value

A data frame with character columns CNES (establishment code) and FANTASIA (trade name).

Network access

This function downloads the current TAB_CNES.zip archive from DataSUS. Transfer progress is displayed by default. The temporary archive and extracted files are removed before the function returns or aborts.

References

Saldanha, R. F. (2026). [CNES – Cadastro Nacional de Estabelecimentos de Saúde](#).

See Also

[process_cnes\(\)](#), [fetch_datasus\(\)](#)

fetch_datasus

Download DataSUS microdata

Description

Downloads published DBC files from DataSUS, reads them with [read_dbc\(\)](#), and combines the records in deterministic period, state, and file-part order.

Usage

```
fetch_datasus(
  year_start,
  month_start = NULL,
  year_end,
  month_end = NULL,
  uf = "all",
  information_system,
  vars = NULL,
  stop_on_error = FALSE,
  timeout = 240,
  track_source = FALSE,
  quiet = FALSE
)
```

Arguments

year_start, year_end	Numeric scalars giving the first and last requested years, inclusive.
month_start, month_end	Numeric scalars giving the first and last requested months, inclusive. Months are required for SIH, SIA, and CNES systems and ignored, with a warning, for annual systems.
uf	A Brazilian state abbreviation, a character vector of abbreviations, or "all". "all" cannot be combined with individual states. A warning alert is displayed when this argument is ignored for systems published only as national files.
information_system	A single system identifier listed in Supported systems .
vars	NULL, or a character vector of column names to retain. Selection is applied to each file before the files are combined.
stop_on_error	Logical scalar. If TRUE, abort after any listing, download, or read failure. If FALSE, warn and return the files that could be read successfully.
timeout	A positive numeric scalar giving the connection and transfer timeout, in seconds, for each network attempt.

track_source	Logical scalar. If TRUE, append a source column with the original DBC file name. This column is retained even when vars is supplied. The function aborts if the downloaded data already contain a column named source.
quiet	Logical scalar. If FALSE (the default), display the transfer progress reported by <code>curl::curl_download()</code> and announce each file before downloading it. If TRUE, suppress status messages, per-file announcements, and progress meters. Warnings and errors are not suppressed.

Details

The function first lists the relevant DataSUS directories and downloads only files present in those listings. When more than one publication represents the same system, period, state, and file part, definitive/current data take precedence over preliminary data, and current data take precedence over historical copies.

Downloads are sequential. Unless quiet = TRUE, transfer progress is displayed by `curl::curl_download()`. Transient network failures are retried up to two times; missing, empty, invalid DBC, and incompatible-schema files are not retried. Partial files and other temporary files are removed before the function returns or aborts.

Years and state abbreviations refer to DataSUS processing periods and places of processing, which may differ from dates or places of occurrence and residence contained in the records.

Value

A tibble containing all successfully read records, or NULL if no requested file could be read. No diagnostic attributes are added.

Supported systems

- **SIH:** "SIH-RD", "SIH-RJ", "SIH-SP", and "SIH-ER".
- **SIM:** "SIM-DO", "SIM-DOFET", "SIM-DOEXT", "SIM-DOINF", and "SIM-DOMAT".
- **SINASC:** "SINASC".
- **CNES:** "CNES-LT", "CNES-ST", "CNES-DC", "CNES-EQ", "CNES-SR", "CNES-HB", "CNES-PF", "CNES-EP", "CNES-RC", "CNES-IN", "CNES-EE", "CNES-EF", and "CNES-GM".
- **SIA:** "SIA-AB", "SIA-ABO", "SIA-ACF", "SIA-AD", "SIA-AN", "SIA-AM", "SIA-AQ", "SIA-AR", "SIA-ATD", "SIA-PA", "SIA-PS", and "SIA-SAD".
- **SINAN:** "SINAN-DENGUE", "SINAN-CHIKUNGUNYA", "SINAN-ZIKA", "SINAN-MALARIA", "SINAN-CHAGAS", "SINAN-LEISHMANIOSE-VISCERAL", "SINAN-LEISHMANIOSE-TEGUMENTAR", and "SINAN-LEPTOSPIROSE".

Network access

An Internet connection and FTP access to DataSUS are required. DataSUS may restrict FTP access from some countries.

References

Saldanha, R. F. (2026). *Sistemas de Informação em Saúde no Brasil*, especially the chapters on **SIM**, **SINASC**, **SIH**, **SIA**, **SINAN**, and **CNES**.

See Also

[read_dbc\(\)](#) for local DBC files; [process_sim\(\)](#), [process_sinasc\(\)](#), [process_sih\(\)](#), [process_sia\(\)](#), [process_cnes\(\)](#), and the `process_sinan_*`() functions for system-specific recoding.

Examples

```
sim <- fetch_datasus(
  year_start = 2014,
  year_end = 2014,
  uf = "AC",
  information_system = "SIM-D0",
  vars = c("CODMUNRES", "DFOBITO", "CAUSABAS")
)

sih <- fetch_datasus(
  year_start = 2014,
  month_start = 1,
  year_end = 2014,
  month_end = 2,
  uf = c("AC", "RR"),
  information_system = "SIH-RD"
)
```

 fetch_sigtab

Download the current SIGTAB table

Description

Downloads and reads the current SIA procedure table distributed by DataSUS. [process_sia\(\)](#) can use this table to add procedure descriptions.

Usage

```
fetch_sigtab(timeout = 240)
```

Arguments

`timeout` A positive numeric scalar. Download and connection timeout, in seconds.

Value

A data frame with character columns `COD` (procedure code) and `nome_proced` (procedure name).

Network access

This function downloads the current `TAB_SIA.zip` archive from DataSUS. Transfer progress is displayed by default. The temporary archive and extracted files are removed before the function returns or aborts.

References

Saldanha, R. F. (2026). *SIA – Sistema de Informações Ambulatoriais do SUS*.

See Also

[process_sia\(\)](#), [fetch_datasus\(\)](#)

paisnet	<i>paisnet table</i>
---------	----------------------

Description

paisnet table, based on PAISNET.DBF file from CNES TabWIN definitions file.

Usage

```
paisnet
```

Format

A data frame with 266 rows and 3 variables:

ID_PAIS Country code 1

COPAISINF Contry code 2

NM_PAIS Label

process_cnes	<i>Prepare CNES microdata</i>
--------------	-------------------------------

Description

Recodes supported CNES fields into descriptive values and normalizes escaped Unicode text. Establishment ("CNES-ST") and professional ("CNES-PF") records have different layouts and are processed accordingly.

Usage

```
process_cnes(
  data,
  information_system = c("CNES-ST", "CNES-PF"),
  nomes = FALSE,
  municipality_data = TRUE
)
```

Arguments

data	A data frame returned by <code>fetch_datusus()</code> or another data frame with a compatible CNES layout.
information_system	A single character string: "CNES-ST" for establishments or "CNES-PF" for professionals.
nomes	Logical scalar. For "CNES-ST" data, if TRUE, download the current CADGER table with <code>fetch_cadger()</code> and add establishment trade names. This requires network access. It has no effect for "CNES-PF".
municipality_data	Logical scalar. If TRUE, add municipality names and available territorial attributes for supported municipality-code columns.

Details

Columns not explicitly recoded are retained, but Unicode normalization is applied to every column and consequently the returned tibble contains character columns. Lookup joins can add establishment, occupation, and municipality information.

Value

A tibble with character columns. Supported codes are replaced with descriptions, and requested lookup fields are added where applicable.

References

Saldanha, R. F. (2026). **CNES – Cadastro Nacional de Estabelecimentos de Saúde**.

See Also

`fetch_datusus()`, `fetch_cadger()`

Examples

```
process_cnes(cnes_st_sample, information_system = "CNES-ST")
process_cnes(cnes_pf_sample, information_system = "CNES-PF")
```

process_sia	<i>Prepare SIA outpatient-production microdata</i>
-------------	--

Description

Recodes supported fields from SIA individual outpatient-production records ("SIA-PA") into descriptive values and normalizes escaped Unicode text. Lookup joins can add procedure, occupation, team, and municipality descriptions.

Usage

```
process_sia(
  data,
  information_system = "SIA-PA",
  nome_proced = TRUE,
  nome_ocupacao = TRUE,
  nome_equipe = TRUE,
  municipality_data = TRUE
)
```

Arguments

<code>data</code>	A data frame returned by fetch_datasus() with <code>information_system = "SIA-PA"</code> , or a compatible layout.
<code>information_system</code>	A single character string. Currently only "SIA-PA" is supported.
<code>nome_proced</code>	Logical scalar. If TRUE, download the current SIGTAB table with fetch_sigtab() and join procedure names. This requires network access.
<code>nome_ocupacao</code>	Logical scalar. If TRUE, join occupation descriptions for supported occupation-code columns.
<code>nome_equipe</code>	Logical scalar retained for API compatibility. Team descriptions are currently joined whenever PA_INE is present, regardless of this value.
<code>municipality_data</code>	Logical scalar. If TRUE, add municipality names and available territorial attributes for supported municipality-code columns.

Details

Columns not explicitly recoded are retained, but Unicode normalization is applied to every column and consequently the returned tibble contains character columns. Other SIA layouts downloadable with [fetch_datasus\(\)](#) are not currently supported by this processing function.

Value

A tibble with character columns. Supported codes are replaced with descriptions, and requested lookup fields are added where applicable.

References

Saldanha, R. F. (2026). *SIA – Sistema de Informações Ambulatoriais do SUS*.

See Also

[fetch_datasus\(\)](#), [fetch_sigtab\(\)](#)

Examples

```
process_sia(sia_pa_sample, nome_proced = FALSE)
```

process_sih

*Prepare SIH hospital-admission microdata***Description**

Recodes supported fields from reduced hospital-admission records ("SIH-RD") into descriptive values and normalizes escaped Unicode text.

Usage

```
process_sih(data, information_system = "SIH-RD", municipality_data = TRUE)
```

Arguments

data A data frame returned by [fetch_datasus\(\)](#) with `information_system = "SIH-RD"`, or a compatible layout.

information_system A single character string. Currently only "SIH-RD" is supported.

municipality_data Logical scalar. If TRUE, add municipality names and available territorial attributes for supported municipality-code columns.

Details

Columns not explicitly recoded are retained, but Unicode normalization is applied to every column and consequently the returned tibble contains character columns. Other SIH layouts downloadable with [fetch_datasus\(\)](#) are not currently supported by this processing function.

Value

A tibble with character columns. Supported codes are replaced with descriptions, and municipality fields are added when requested and available.

References

Saldanha, R. F. (2026). *SIH – Sistema de Informações Hospitalares do SUS*.

See Also

[fetch_datasus\(\)](#)

Examples

```
process_sih(sih_rd_sample)
```

process_sim	<i>Prepare SIM mortality microdata</i>
-------------	--

Description

Recodes supported SIM mortality fields into descriptive values and normalizes escaped Unicode text. Codes without a documented conversion are retained.

Usage

```
process_sim(data, municipality_data = TRUE)
```

Arguments

data	A data frame returned by fetch_datasus() for a SIM mortality system, or another data frame with a compatible layout.
municipality_data	Logical scalar. If TRUE, add municipality names and available territorial attributes for supported municipality-code columns.

Details

Columns not explicitly recoded are retained, but Unicode normalization is applied to every column and consequently the returned tibble contains character columns.

Value

A tibble with character columns. Supported codes are replaced with descriptions, and municipality fields are added when requested and available.

References

Saldanha, R. F. (2026). [SIM – Sistema de Informação sobre Mortalidade](#).

See Also

[fetch_datasus\(\)](#)

Examples

```
process_sim(sim_do_sample)
```

process_sinan_chagas	<i>Prepare SINAN Chagas disease microdata</i>
----------------------	---

Description

Recodes supported fields from SINAN Chagas disease notifications into descriptive values and normalizes escaped Unicode text. Columns not explicitly recoded are retained, but the returned tibble contains character columns.

Usage

```
process_sinan_chagas(data, municipality_data = TRUE)
```

Arguments

data	A data frame returned by <code>fetch_datasus()</code> with <code>information_system = "SINAN-CHAGAS"</code> , or a compatible layout.
municipality_data	Logical scalar retained for API compatibility. It is not currently used by this processing function.

Value

A tibble with character columns. Supported codes are replaced with descriptions.

References

Saldanha, R. F. (2026). *SINAN – Sistema de Informação de Agravos de Notificação*.

See Also

`fetch_datasus()`

Examples

```
process_sinan_chagas(sinan_chagas_sample)
```

`process_sinan_chikungunya`*Prepare SINAN chikungunya microdata*

Description

Recodes supported fields from SINAN chikungunya notifications into descriptive values and normalizes escaped Unicode text. Columns not explicitly recoded are retained, but the returned tibble contains character columns.

Usage

```
process_sinan_chikungunya(data, municipality_data = TRUE)
```

Arguments

`data` A data frame returned by `fetch_datasus()` with `information_system = "SINAN-CHIKUNGUNYA"`, or a compatible layout.

`municipality_data` Logical scalar. If TRUE, add municipality names and available territorial attributes for supported municipality-code columns.

Value

A tibble with character columns. Supported codes are replaced with descriptions, and municipality fields are added when requested and available.

References

Saldanha, R. F. (2026). *SINAN – Sistema de Informação de Agravos de Notificação*.

See Also

`fetch_datasus()`

Examples

```
process_sinan_chikungunya(sinan_chikungunya_sample)
```

process_sinan_dengue	<i>Prepare SINAN dengue microdata</i>
----------------------	---------------------------------------

Description

Recodes supported fields from SINAN dengue notifications into descriptive values and normalizes escaped Unicode text. Columns not explicitly recoded are retained, but the returned tibble contains character columns.

Usage

```
process_sinan_dengue(data, municipality_data = TRUE)
```

Arguments

data	A data frame returned by <code>fetch_datasus()</code> with <code>information_system = "SINAN-DENGUE"</code> , or a compatible layout.
municipality_data	Logical scalar. If TRUE, add municipality names and available territorial attributes for supported municipality-code columns.

Value

A tibble with character columns. Supported codes are replaced with descriptions, and municipality fields are added when requested and available.

References

Saldanha, R. F. (2026). *SINAN – Sistema de Informação de Agravos de Notificação*.

See Also

`fetch_datasus()`

Examples

```
process_sinan_dengue(sinan_dengue_sample)
```

`process_sinan_leishmaniose_tegumentar`*Prepare SINAN tegumentary leishmaniasis microdata*

Description

Recodes supported fields from SINAN tegumentary leishmaniasis notifications into descriptive values and normalizes escaped Unicode text. Columns not explicitly recoded are retained, but the returned tibble contains character columns.

Usage

```
process_sinan_leishmaniose_tegumentar(data, municipality_data = TRUE)
```

Arguments

<code>data</code>	A data frame returned by <code>fetch_datasus()</code> with <code>information_system = "SINAN-LEISHMANIOSE-TEGUMENTAR"</code> or a compatible layout.
<code>municipality_data</code>	Logical scalar retained for API compatibility. It is not currently used by this processing function.

Value

A tibble with character columns. Supported codes are replaced with descriptions.

References

Saldanha, R. F. (2026). *SINAN – Sistema de Informação de Agravos de Notificação*.

See Also

`fetch_datasus()`

Examples

```
process_sinan_leishmaniose_tegumentar(sinan_leishmaniose_tegumentar_sample)
```

`process_sinan_leishmaniose_visceral`*Prepare SINAN visceral leishmaniasis microdata*

Description

Recodes supported fields from SINAN visceral leishmaniasis notifications into descriptive values and normalizes escaped Unicode text. Columns not explicitly recoded are retained, but the returned tibble contains character columns.

Usage

```
process_sinan_leishmaniose_visceral(data, municipality_data = TRUE)
```

Arguments

<code>data</code>	A data frame returned by <code>fetch_datasus()</code> with <code>information_system = "SINAN-LEISHMANIOSE-VISCERAL"</code> or a compatible layout.
<code>municipality_data</code>	Logical scalar retained for API compatibility. It is not currently used by this processing function.

Value

A tibble with character columns. Supported codes are replaced with descriptions.

References

Saldanha, R. F. (2026). *SINAN – Sistema de Informação de Agravos de Notificação*.

See Also

`fetch_datasus()`

Examples

```
process_sinan_leishmaniose_visceral(sinan_leishmaniose_visceral_sample)
```

`process_sinan_malaria` *Prepare SINAN malaria microdata*

Description

Recodes supported fields from SINAN malaria notifications into descriptive values and normalizes escaped Unicode text. Columns not explicitly recoded are retained, but the returned tibble contains character columns.

Usage

```
process_sinan_malaria(data, municipality_data = TRUE)
```

Arguments

<code>data</code>	A data frame returned by <code>fetch_datasus()</code> with <code>information_system = "SINAN-MALARIA"</code> , or a compatible layout.
<code>municipality_data</code>	Logical scalar retained for API compatibility. It is not currently used by this processing function.

Value

A tibble with character columns. Supported codes are replaced with descriptions.

References

Saldanha, R. F. (2026). *SINAN – Sistema de Informação de Agravos de Notificação*.

See Also

`fetch_datasus()`

Examples

```
process_sinan_malaria(sinan_malaria_sample)
```

process_sinan_zika	<i>Prepare SINAN Zika virus disease microdata</i>
--------------------	---

Description

Recodes supported fields from SINAN Zika virus disease notifications into descriptive values and normalizes escaped Unicode text. Columns not explicitly recoded are retained, but the returned tibble contains character columns.

Usage

```
process_sinan_zika(data, municipality_data = TRUE)
```

Arguments

data	A data frame returned by <code>fetch_datasus()</code> with <code>information_system = "SINAN-ZIKA"</code> , or a compatible layout.
municipality_data	Logical scalar retained for API compatibility. It is not currently used by this processing function.

Value

A tibble with character columns. Supported codes are replaced with descriptions.

References

Saldanha, R. F. (2026). *SINAN – Sistema de Informação de Agravos de Notificação*.

See Also

`fetch_datasus()`

Examples

```
process_sinan_zika(sinan_zika_sample)
```

process_sinasc	<i>Prepare SINASC live-birth microdata</i>
----------------	--

Description

Recodes supported SINASC live-birth fields into descriptive values and normalizes escaped Unicode text. Codes without a documented conversion are retained.

Usage

```
process_sinasc(data, municipality_data = TRUE)
```

Arguments

data	A data frame returned by fetch_datasus() with <code>information_system = "SINASC"</code> , or a compatible layout.
municipality_data	Logical scalar. If TRUE, add municipality names and available territorial attributes for supported municipality-code columns.

Details

Columns not explicitly recoded are retained, but Unicode normalization is applied to every column and consequently the returned tibble contains character columns.

Value

A tibble with character columns. Supported codes are replaced with descriptions, and municipality fields are added when requested and available.

References

Saldanha, R. F. (2026). [SINASC – Sistema de Informação sobre Nascidos Vivos](#).

See Also

[fetch_datasus\(\)](#)

Examples

```
process_sinasc(sinasc_sample)
```

read_dbc*Read a DBC file*

Description

Decompresses a DataSUS DBC file to a temporary DBF file and reads it into a tibble. Use this function for a DBC file already available locally; use [fetch_datasus\(\)](#) to discover and download files from DataSUS.

Usage

```
read_dbc(file, as_character = TRUE)
```

Arguments

file	A single character string with the path to a readable, non-empty DBC file.
as_character	If TRUE (the default), converts every column to character. If FALSE, preserves the types inferred from the DBF metadata.

Details

Decompression is performed through the package's bundled DBC implementation. The intermediate DBF file is created in the R temporary directory and removed before the function returns or aborts. The implementation was adapted from the `heal thbR` package.

Invalid input files, decompression failures, and DBF reading failures abort with errors in the `microdatasus_dbc_error` family.

Value

A tibble with one column per DBF field. By default, all columns are character vectors; with `as_character = FALSE`, DBF-inferred types are retained.

References

Saldanha, R. F. (2026). *Sistemas de Informação em Saúde no Brasil*.

See Also

[fetch_datasus\(\)](#)

sia_pa_sample	<i>SIA-PA sample</i>
---------------	----------------------

Description

SIA-PA sample with the first 100 rows from AC 2016-6 file.

Usage

sia_pa_sample

Format

An object of class `data.frame` with 100 rows and 60 columns.

sigtab	<i>Procediments table</i>
--------	---------------------------

Description

Procediments table, based on TB_SIGTAP.dbf file from SIA TabWIN definitions file.

Usage

sigtab

Format

A data frame with 5325 rows and 2 variables:

COD Code

nome_proced Procediment name

sih_rd_sample	<i>SIH-RD sample</i>
---------------	----------------------

Description

SIH-RD sample with the first 100 rows from AC 2016-6 file.

Usage

sih_rd_sample

Format

An object of class `data.frame` with 100 rows and 113 columns.

sim_do_sample	<i>SIM-DO sample</i>
---------------	----------------------

Description

SIM-DO sample with the first 100 rows from AC 2016 file.

Usage

sim_do_sample

Format

An object of class `data.frame` with 100 rows and 88 columns.

sinan_chagas_sample	<i>SINAN Chagas sample</i>
---------------------	----------------------------

Description

SINAN Chagas sample with the first 100 rows from the 2023 file.

Usage

sinan_chagas_sample

Format

An object of class `data.frame` with 100 rows and 108 columns.

sinan_chikungunya_sample	<i>SINAN Chikungunya sample</i>
--------------------------	---------------------------------

Description

SINAN Chikungunya sample with the first 100 rows from the 2022 file.

Usage

sinan_chikungunya_sample

Format

An object of class `data.frame` with 100 rows and 122 columns.

sinan_dengue_sample	<i>SINAN Dengue sample</i>
---------------------	----------------------------

Description

SINAN Dengue sample with the first 100 rows from the 2010 file.

Usage

```
sinan_dengue_sample
```

Format

An object of class `data.frame` with 100 rows and 66 columns.

sinan_leishmaniose_tegumentar_sample	<i>SINAN Leishmaniose Tegumentar sample</i>
--------------------------------------	---

Description

SINAN Leishmaniose Tegumentar sample with the first 100 rows from the 2023 file.

Usage

```
sinan_leishmaniose_tegumentar_sample
```

Format

An object of class `data.frame` with 100 rows and 74 columns.

sinan_leishmaniose_visceral_sample	<i>SINAN Leishmaniose Visceral sample</i>
------------------------------------	---

Description

SINAN Leishmaniose Visceral sample with the first 100 rows from the 2023 file.

Usage

```
sinan_leishmaniose_visceral_sample
```

Format

An object of class `data.frame` with 100 rows and 76 columns.

sinan_malaria_sample	<i>SINAN Malaria sample</i>
----------------------	-----------------------------

Description

SINAN Malaria sample with the first 100 rows from the 2016 file.

Usage

```
sinan_malaria_sample
```

Format

An object of class `data.frame` with 100 rows and 50 columns.

sinan_zika_sample	<i>SINAN Zika sample</i>
-------------------	--------------------------

Description

SINAN Zika sample with the first 100 rows from the 2016 file.

Usage

```
sinan_zika_sample
```

Format

An object of class `data.frame` with 100 rows and 38 columns.

sinasc_sample	<i>SINASC sample</i>
---------------	----------------------

Description

SINASC sample with the first 100 rows from AC 2016 file.

Usage

```
sinasc_sample
```

Format

An object of class `data.frame` with 100 rows and 61 columns.

tabCBO	<i>CBO table</i>
--------	------------------

Description

CBO table, based on CBO2002.CNV file from SIM TabWIN definitions file.

Usage

tabCBO

Format

A data frame with 2428 rows and 2 variables:

cod Code

nome Label

tabMun	<i>Municipalities table</i>
--------	-----------------------------

Description

Municipalities table

Usage

tabMun

Format

A data frame with 5659 rows and 2 variables:

munResCod Municipality IBGE code with 6 numbers

munResStatus Status

munResTipo Type

munResNome Name

munResUf UF (state)

munResLat Latitude

munResLon Longitude

munResAlt Altitude

munResArea Area

tabNaturalidade	<i>Naturalidade table</i>
-----------------	---------------------------

Description

Naturalidade table, based on NATUR.CNV file from SIM TabWIN definitions file.

Usage

tabNaturalidade

Format

A data frame with 292 rows and 2 variables:

cod Code

nome Label

tab0cupacao	<i>Ocupations table</i>
-------------	-------------------------

Description

Ocupations table, based on OCUPACAO.CNV file from SIM TabWIN definitions file.

Usage

tab0cupacao

Format

A data frame with 350 rows and 2 variables:

cod Code

nome Label

Index

* datasets

cnes_pf_sample, 4
cnes_st_sample, 4
equipe, 5
paisnet, 9
sia_pa_sample, 23
sigtab, 23
sih_rd_sample, 23
sim_do_sample, 24
sinan_chagas_sample, 24
sinan_chikungunya_sample, 24
sinan_dengue_sample, 25
sinan_leishmaniose_tegumentar_sample, 25
sinan_leishmaniose_visceral_sample, 25
sinan_malaria_sample, 26
sinan_zika_sample, 26
sinasc_sample, 26
tabCBO, 27
tabMun, 27
tabNaturalidade, 28
tabOcupacao, 28

cnes_pf_sample, 4
cnes_st_sample, 4
curl::curl_download(), 7

equipe, 5

fetch_cadger, 5
fetch_cadger(), 3, 10
fetch_datasus, 6
fetch_datasus(), 3, 6, 9–22
fetch_sigtab, 8
fetch_sigtab(), 3, 11

microdatasus (microdatasus-package), 3
microdatasus-package, 3

paisnet, 9

process_cnes, 9
process_cnes(), 3, 5, 6, 8
process_sia, 10
process_sia(), 3, 8, 9
process_sih, 12
process_sih(), 3, 8
process_sim, 13
process_sim(), 3, 8
process_sinan_chagas, 14
process_sinan_chikungunya, 15
process_sinan_dengue, 16
process_sinan_leishmaniose_tegumentar, 17
process_sinan_leishmaniose_visceral, 18
process_sinan_malaria, 19
process_sinan_zika, 20
process_sinasc, 21
process_sinasc(), 3, 8

read_dbc, 22
read_dbc(), 3, 6, 8

sia_pa_sample, 23
sigtab, 23
sih_rd_sample, 23
sim_do_sample, 24
sinan_chagas_sample, 24
sinan_chikungunya_sample, 24
sinan_dengue_sample, 25
sinan_leishmaniose_tegumentar_sample, 25
sinan_leishmaniose_visceral_sample, 25
sinan_malaria_sample, 26
sinan_zika_sample, 26
sinasc_sample, 26

tabCBO, 27
tabMun, 27
tabNaturalidade, 28
tabOcupacao, 28