

Package ‘microdatasus’

June 18, 2024

Title Download and Process 'DataSUS' Files

Version 2.3.1

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

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

Description Downloads data files from 'DataSUS' health information systems from <ftp.datasus.gov.br> and process the data, including labeling categorical variables.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.1

Imports checkmate, cli, curl, data.table, dplyr, dtplyr, foreign, lubridate, magrittr, RCurl, read.dbc, rlang, stringi, tibble, utils, zip

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

Config/Needs/website rmarkdown

NeedsCompilation no

Author Raphael Saldanha [aut, cre] (<<https://orcid.org/0000-0003-0652-8466>>)

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

Repository CRAN

Date/Publication 2024-06-18 13:30:02 UTC

Contents

cnes_pf_sample	2
cnes_st_sample	3
equipe	3
fetch_datasus	4
fetch_sigtab	6
microdatasus	6
paisnet	7
process_cnes	7
process_sia	8
process_sih	9
process_sim	10
process_sinan_chikungunya	10
process_sinan_dengue	11
process_sinan_malaria	12
process_sinan_zika	13
process_sinasc	13
sia_pa_sample	14
sigtab	15
sih_rd_sample	15
sim_do_sample	15
sinan_chikungunya_sample	16
sinan_dengue_sample	16
sinan_malaria_sample	16
sinan_zika_sample	17
sinasc_sample	17
tabCBO	17
tabMun	18
tabNaturalidade	18
tabOcupacao	19
Index	20

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_datasus

Fetch and read microdata files from DataSUS

Description

fetch_datasus downloads microdata (DBC) files from DataSUS and reads them.

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
)
```

Arguments

year_start, year_end	numeric. Start and end year of files in the format yyyy.
month_start, month_end	numeric. Start and end month in the format mm. Those parameters are only used with the health information systems SIH, CNES and SIA. These parameters are ignored if the information health system is SIM or SINASC.
uf	an optional string or a vector of strings. By default all UFs ("Unidades Federativas") are downloaded. See <i>Details</i> .
information_system	string. The abbreviation of the health information system to be accessed. See <i>Details</i> .
vars	an optional string or a vector of strings. By default, all variables read and stored, unless a list of desired variables is informed by this parameter.
stop_on_error	logical. If TRUE, the download process will be stopped if an error occurs.
timeout	numeric (seconds). Sets a timeout tolerance for downloads, useful on large files and/or slow connections. Defaults to 240 seconds.
track_source	logical. If TRUE, adds a column called source with the downloaded file name.

Details

This function downloads DBC files from DataSUS following parameters about start date, end date, UF and health information system abbreviation. After the download process, the files are merged into a unique `data.frame` object.

A specific UF or a vector of UFs can be informed using the following abbreviations: "AC", "AL", "AP", "AM", "BA", "CE", "DF", "ES", "GO", "MA", "MT", "MS", "MG", "PA", "PB", "PR", "PE", "PI", "RJ", "RN", "RS", "RO", "RR", "SC", "SP", "SE", "TO".

The following systems are implemented: "SIH-RD", "SIH-RJ", "SIH-SP", "SIH-ER", "SIM-DO", "SIM-DOFET", "SIM-DOEXT", "SIM-DOINF", "SIM-DOMAT", "SINASC", "CNES-LT", "CNES-ST", "CNES-DC", "CNES-EQ", "CNES-SR", "CNES-HB", "CNES-PF", "CNES-EP", "CNES-RC", "CNES-IN", "CNES-EE", "CNES-EF", "CNES-GM", "SIA-AB", "SIA-ABO", "SIA-ACF", "SIA-AD", "SIA-AN", "SIA-AM", "SIA-AQ", "SIA-AR", "SIA-ATD", "SIA-PA", "SIA-PS", "SIA-SAD", "SINAN-DENGUE-FINAL", "SINAN-DENGUE-PRELIMINAR", "SINAN-CHIKUNGUNYA-FINAL", "SINAN-CHIKUNGUNYA-PRELIMINAR", "SINAN-ZIKA-FINAL", "SINAN-ZIKA-PRELIMINAR", "SINAN-MALARIA-FINAL", "SINAN-MALARIA-PRELIMINAR".

Value

a `data.frame` with the contents of the DBC files.

Warning

A Internet connection is needed to use this function.

Currently, DataSUS FTP server is restricting download calls from some countries, except Brazil.

The year and month used to download the files regards the processing month and year of the cases by DataSUS.

The UF regards where the cases were processed by DataSUS.

The files are downloaded to a temporary folder and deleted after the reading process.

Examples

```
# Fetch two years of data from SIM-DO
fetch_datasus(year_start = 2010, year_end = 2011,
              information_system = "SIM-DO")

# Fetch one year of data from SIM-DO and keep only three variables
fetch_datasus(year_start = 2014, year_end = 2014,
              information_system = "SIM-DO",
              vars = c("CODMUNRES", "DTOBITO", "CAUSABAS"))

# Fetch some months' data from SIH-RD for four states
fetch_datasus(year_start = 2014, month_start = 1,
              year_end = 2014, month_end = 2,
              uf = c("RJ", "MG", "SP", "ES"),
              information_system = "SIH-RD")
```

fetch_sigtab

Fetch SIGTAB table

Description

This function fetches an updated table of procedures (SIGTAB) from DatasUS

Usage

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

Arguments

timeout sets the download timeout reference. Defaults to 240

Value

a data.frame

microdatasus

microdatasus: Download and preprocess DataSUS files

Description

Downloads microdata files (DBC format) from DataSUS and imports the files for use.

Author(s)

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

See Also

Useful links:

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

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>Process CNES variables from DataSUS</i>
--------------	--

Description

process_cnes processes CNES variables retrieved by fetch_datasus().

Usage

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

Arguments

data	data.frame created by fetch_datasus().
information_system	string. CNES-ST or CNES-PF
nomes	optional logical. TRUE by default, add FANTASIA and RAZÃO SOCIAL names to the dataset.
municipality_data	optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes CNES-ST (Estabelecimentos) or CNES-PF (Pessoa Física) variables retrieved by `fetch_datasus()`, informing labels for categorical variables including NA values.

Value

a `data.frame` with the processed data.

Examples

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

process_sia

Process SIA variables from DataSUS

Description

`process_sia` processes SIA variables retrieved by `fetch_datasus()`.

Usage

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

Arguments

`data` `data.frame` created by `fetch_datasus()`.

`information_system` string. The abbreviation of the health information system. See *Details*.

`nome_proced` optional logical. TRUE by default, add PA_PROCED_NOME to the dataset. This setting will start to download a file from DataSUS to retrieve the updates list of procedures (SIGTAB).

`nome_ocupacao` optional logical. TRUE by default, add OCUPACAO name to the dataset.

`nome_equipe` optional logical. TRUE by default, add EQUIPE name to the dataset.

`municipality_data` optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SIA variables retrieved by `fetch_datasus()`, informing labels for categorical variables including NA values.

Currently, only "SIA-PA" is supported.

Value

a `data.frame` with the processed data.

Examples

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

process_sih

Process SIH variables from DataSUS

Description

`process_SIH` processes SIH variables retrieved by `fetch_datasus()`.

Usage

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

Arguments

`data` `data.frame` created by `fetch_datasus()`.

`information_system`

string. The abbreviation of the health information system. See *Details*.

`municipality_data`

optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SIH variables retrieved by `fetch_datasus()`, informing labels for categorical variables including NA values.

Currently, only "SIH-RD" is supported.

Value

a `data.frame` with the processed data.

Examples

```
process_sih(sih_rd_sample)
```

process_sim	<i>Process SIM variables from DataSUS</i>
-------------	---

Description

process_sim processes SIM variables retrieved by fetch_datasus().

Usage

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

Arguments

data data.frame created by fetch_datasus().
municipality_data optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SIM variables retrieved by fetch_datasus(), informing labels for categorical variables including NA values.

Value

a data.frame with the processed data.

Examples

```
process_sim(sim_do_sample)
```

process_sinan_chikungunya	<i>Process SINAN Chikungunya variables from DataSUS</i>
---------------------------	---

Description

process_sinan_chikungunya processes SINAN Chikungunya variables retrieved by fetch_datasus().

Usage

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

Arguments

`data` data.frame created by `fetch_datasus()`.
`municipality_data` optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SINAN Chikungunya variables retrieved by `fetch_datasus()`, informing labels for categoric variables including NA values.

Value

a data.frame with the processed data.

Examples

```
process_sinan_chikungunya(sinan_chikungunya_sample)
```

`process_sinan_dengue` *Process SINAN Dengue variables from DataSUS*

Description

`process_sinan_dengue` processes SINAN Dengue variables retrieved by `fetch_datasus()`.

Usage

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

Arguments

`data` data.frame created by `fetch_datasus()`.
`municipality_data` optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SINAN Dengue variables retrieved by `fetch_datasus()`, informing labels for categoric variables including NA values.

Value

a data.frame with the processed data.

Examples

```
process_sinan_dengue(sinan_dengue_sample)
```

process_sinan_malaria *Process SINAN Malaria variables from DataSUS*

Description

process_sinan_malaria processes SINAN Malaria variables retrieved by fetch_datasus().

Usage

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

Arguments

data data.frame created by fetch_datasus().

municipality_data optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SINAN Malaria variables retrieved by fetch_datasus(), informing labels for categoric variables including NA values.

Value

a data.frame with the processed data.

Examples

```
process_sinan_malaria(sinan_malaria_sample)
```

process_sinan_zika *Process SINAN Zika variables from DataSUS*

Description

process_sinan_zika processes SINAN Zika variables retrieved by fetch_datasus().

Usage

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

Arguments

`data` data.frame created by fetch_datasus().
`municipality_data` optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SINAN Zika variables retrieved by fetch_datasus(), informing labels for categoric variables including NA values.

Value

a data.frame with the processed data.

Examples

```
process_sinan_zika(sinan_zika_sample)
```

process_sinasc *Process SINASC variables from DataSUS*

Description

process_sinasc processes SINASC variables retrieved by fetch_datasus().

Usage

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

Arguments

`data` data.frame created by `fetch_datusus()`.
`municipality_data` optional logical. TRUE by default, creates new variables in the dataset informing the full name and other details about the municipality of residence.

Details

This function processes SINASC variables retrieved by `fetch_datusus()`, informing labels for categoric variables including NA values.

Value

a data.frame with the processed data.

Examples

```
process_sinasc(sinasc_sample)
```

<code>sia_pa_sample</code>	<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_chikungunya_sample *SINAN Chikungunya sample*

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 *SINAN Dengue sample*

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_malaria_sample *SINAN Malaria sample*

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, 2
 - cnes_st_sample, 3
 - equipe, 3
 - paisnet, 7
 - sia_pa_sample, 14
 - sigtab, 15
 - sih_rd_sample, 15
 - sim_do_sample, 15
 - sinan_chikungunya_sample, 16
 - sinan_dengue_sample, 16
 - sinan_malaria_sample, 16
 - sinan_zika_sample, 17
 - sinasc_sample, 17
 - tabCBO, 17
 - tabMun, 18
 - tabNaturalidade, 18
 - tabOcupacao, 19
-
- cnes_pf_sample, 2
 - cnes_st_sample, 3
-
- equipe, 3
-
- fetch_datasus, 4
 - fetch_sigtab, 6
-
- microdatasus, 6
 - microdatasus-package (microdatasus), 6
-
- paisnet, 7
 - process_cnes, 7
 - process_sia, 8
 - process_sih, 9
 - process_sim, 10
 - process_sinan_chikungunya, 10
 - process_sinan_dengue, 11
 - process_sinan_malaria, 12
 - process_sinan_zika, 13
 - process_sinasc, 13
 - sia_pa_sample, 14
 - sigtab, 15
 - sih_rd_sample, 15
 - sim_do_sample, 15
 - sinan_chikungunya_sample, 16
 - sinan_dengue_sample, 16
 - sinan_malaria_sample, 16
 - sinan_zika_sample, 17
 - sinasc_sample, 17
 - tabCBO, 17
 - tabMun, 18
 - tabNaturalidade, 18
 - tabOcupacao, 19