

# Package ‘datazoom.amazonia’

February 10, 2023

**Title** Simplify Access to Data from the Amazon Region

**Version** 1.0.0

**Description** Functions to download and treat data regarding the Brazilian Amazon region from a variety of official sources.

**License** MIT + file LICENSE

**URL** <https://www.econ.puc-rio.br/datazoom/>

**Depends** R (>= 4.0)

**Imports** data.table, dplyr, Hmisc, janitor, lubridate, magrittr, purrr, readr, readxl, sf, sidrar, stringi, stringr, tibble, tidyr, utils, XML

**Suggests** foreign, googledrive, knitr, RCurl, rmarkdown, terra

**LinkingTo** Rcpp

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**NeedsCompilation** yes

**Author** Francisco de Lima Cavalcanti [aut, cre],  
DataZoom (PUC-Rio) [fnd],  
Guilherme Jardim [aut],  
Daniel AC Barbosa [aut],  
Bruno Alcantara Duarte [aut],  
Fredie Didier [aut],  
Tito Bruni [aut],  
Luiz Guilherme Lopes Moussatche [aut],  
Victor Aliende da Matta [aut],  
Anna Carolina Dutra Saraiva [aut],  
Arthur Carvalho Brito [aut],  
Igor Rigolon Veiga [aut],  
Maria Mittelbach [aut]

**Maintainer** Francisco de Lima Cavalcanti <francisco.lima.cavalcanti@gmail.com>

**Repository** CRAN

**Date/Publication** 2023-02-10 08:00:02 UTC

## R topics documented:

dbc2dbf . . . . .	2
load_aneel . . . . .	4
load_baci . . . . .	4
load_br_trade . . . . .	5
load_cempre . . . . .	6
load_cipo . . . . .	7
load_climate . . . . .	8
load_datasus . . . . .	9
load_degrad . . . . .	11
load_deter . . . . .	12
load_epe . . . . .	13
load_ibama . . . . .	13
load_iema . . . . .	14
load_imazon . . . . .	15
load_ips . . . . .	16
load_mapbiomas . . . . .	17
load_pam . . . . .	18
load_pevs . . . . .	19
load_pibmunic . . . . .	20
load_ppm . . . . .	21
load_prodes . . . . .	22
load_seeg . . . . .	23
load_sigmine . . . . .	24
municipalities . . . . .	25
read.dbc . . . . .	26

**Index** **28**

---

dbc2dbf *Decompress a DBC (compressed DBF) file*

---

### Description

This function allows you decompress a DBC file into its DBF counterpart. Please note that this is the file format used by the Brazilian Ministry of Health (DATASUS), and it is not related to the FoxPro or CANdb DBC file formats.

### Usage

```
dbc2dbf(input.file, output.file)
```

## Arguments

input.file      The name of the DBC file (including extension)  
output.file     The output file name (including extension)

## Details

DBC is the extension for compressed DBF files (from the 'XBASE' family of databases). This is a proprietary file format used by the brazilian government to make available public healthcare datasets (by it's agency called DATASUS).

It uses internally the PKWare's Data Compression Library (DCL) "implode" compression algorithm. When decompressed, it becomes a regular DBF file.

## Value

Return TRUE if succeeded, FALSE otherwise.

## Author(s)

Daniela Petruzalek, <daniela.petruzalek@gmail.com>

## Source

The internal C code for dbc2dbf is based on blast decompressor and blast-dbf (see *References*).

## References

The PKWare ZIP file format documentation (contains the "implode" algorithm specification) available at <https://support.pkware.com/display/PKZIP/APPNOTE>, current version <https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT>.

blast source code in C: <https://github.com/madler/zlib/tree/master/contrib/blast>

blast-dbf, DBC to DBF command-line decompression tool: <https://github.com/eaglebh/blast-dbf>

## See Also

read.dbc

## Examples

```
## Not run:  
# Input file name  
in.f <- system.file("files/sids.dbc", package = "read.dbc")  
  
# Output file name  
out.f <- tempfile(fileext = ".dbc")  
  
# The call return logi = TRUE on success  
if (dbc2dbf(input.file = in.f, output.file = out.f)) {  
  print("File decompressed!")  
}
```

```

    file.remove(out.f)
  }

  ## End(Not run)

```

---

load_aneel	<i>ANEEL</i>
------------	--------------

---

### Description

National Electric Energy Agency - ANEEL

### Usage

```
load_aneel(dataset, raw_data = FALSE, language = "eng")
```

### Arguments

dataset	A dataset name ("energy_development_budget" or "energy_generation")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

### Examples

```

## Not run:
# download treated data about energy generation
clean_aneel <- load_aneel(
  dataset = "energy_generation",
  raw_data = FALSE
)

## End(Not run)

```

---

load_baci	<i>BACI - Global international trade</i>
-----------	--

---

### Description

Loads disaggregated data on bilateral trade flows for more than 5000 products and 200 countries.

### Usage

```
load_baci(dataset = "HS92", raw_data = FALSE, time_period, language = "eng")
```

**Arguments**

dataset	A dataset name ("HS92").
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:
# download treated data for 2016 (takes a long time to download)
clean_baci <- load_baci(
  raw_data = FALSE,
  time_period = 2016
)

## End(Not run)
```

---

load_br_trade	<i>Comex - Brazilian external trade</i>
---------------	---

---

**Description**

Loads data on all products imported to or exported from Brazil.

**Usage**

```
load_br_trade(dataset, raw_data = FALSE, time_period, language = "eng")
```

**Arguments**

dataset	A dataset name ("comex_export_mun", "comex_import_mun", "comex_export_prod" or "comex_import_prod").
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:
# download treated (raw_data = FALSE) exports data by municipality (dataset = "comex_export_mun")
# from 2020 to 2021 (time_period = 2020:2021)
data <- load_br_trade(
  dataset = "comex_export_mun",
  raw_data = FALSE,
  time_period = 2020:2021
)
# download treated (raw_data = FALSE) imports data by municipality (dataset = "comex_import_mun")
# from 2020 to 2021 (time_period = 2020:2021)
data <- load_br_trade(
  dataset = "comex_import_mun",
  raw_data = FALSE,
  time_period = 2020:2021
)

## End(Not run)
```

---

load\_cempre

*CEMPRE - Central Register of Companies*

---

**Description**

Loads information on companies and other organizations and their respective formally constituted local units, registered with the CNPJ - National Register of Legal Entities.

**Usage**

```
load_cempre(
  dataset = "cempre",
  raw_data = FALSE,
  geo_level,
  time_period,
  language = "eng",
  sectors = FALSE
)
```

**Arguments**

**dataset** A dataset name ("cempre").

**raw\_data** A boolean setting the return of raw (TRUE) or processed (FALSE) data.

geo_level	A string that defines the geographic level of the data. Can be one of "country", "state" or "municipality".
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.
sectors	A boolean that defines if the data will be return separated by sectors (TRUE) or not (FALSE).

**Value**

A tibble.

**Examples**

```
## Not run:
# Download raw data (raw_data = TRUE) at the country level
# from 2008 to 2010 (time_period = 2008:2010).
data <- load_cempre(
  raw_data = TRUE,
  geo_level = "country",
  time_period = 2008:2010
)

# Download trested data (raw_data = FALSE) by state (geo_level = "state")
# from 2008 to 2010 (time_period = 2008:2010) in portuguese (language = "pt").
# In this example, data is split by sector (sectors = TRUE)
data <- load_cempre(
  raw_data = FALSE,
  geo_level = "state",
  time_period = 2008:2010,
  language = "pt",
  sectors = TRUE
)

## End(Not run)
```

---

load\_cipo

*Plataforma CIPÓ - Mappings on environmental crimes*


---

**Description**

Plataforma CIPÓ - Mappings on environmental crimes

**Usage**

```
load_cipo(dataset = "brazilian_actors", raw_data = FALSE, search = "")
```

**Arguments**

dataset	A dataset name ("brazilian_actors", "international_cooperation", "forest_governance")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
search	A string that filters entries containing it.

**Value**

A tibble of the chosen CIPÓ spreadsheet.

**Examples**

```
## Not run:
# download the spreadsheet on Brazilian actors involved in fighting environmental crimes
brazilian_actors <- load_cipo(dataset = "brazilian_actors")

# searching only for entries containing IBAMA
actors_ibama <- load_cipo(
  dataset = "brazilian_actors",
  search = "ibama"
)

# entries containing IBAMA or FUNAI
actors_ibama <- load_cipo(
  dataset = "brazilian_actors",
  search = "ibama|funai"
)

## End(Not run)
```

---

load\_climate

*TerraClimate - Climate monitoring*

---

**Description**

Spatial data on climate variables, extracted from Climatology Lab's TerraClimate.

**Usage**

```
load_climate(
  dataset,
  raw_data = FALSE,
  time_period,
  language = "eng",
  legal_amazon_only = FALSE
)
```



**Arguments**

dataset	A dataset name, choosing which variable will be loaded. One of ("max_temperature", "min_temperature", "wind_speed", "vapor_pressure_deficit", "vapor_pressure", "snow_water_equivalent", "shortwave_radiation_flux", "soil_moisture", "runoff", "precipitation", "potential_evaporation", "climatic_water_deficit", "water_evaporation", "palmer_drought_severity_index"). For extra details, try vignette("TERRACLIMATE").
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.
legal_amazon_only	A boolean setting the return of Legal Amazon Data (TRUE) or Country's Data (FALSE). Defaults to FALSE

**Value**

A tibble.

**Examples**

```
## Not run:
# Downloading maximum temperature data from 2000 to 2001
max_temp <- load_climate(dataset = "max_temperature", time_period = 2000:2001)

# Downloading precipitation data only for the legal Amazon in 2010
amz_precipitation <- load_climate(
  dataset = "precipitation",
  time_period = 2010,
  legal_amazon_only = TRUE
)

## End(Not run)
```

---

load\_datusus

*DATASUS - Mortality, hospitalizations and hospital beds*


---

**Description**

Loads DATASUS data on health establishments, mortality, access to health services and several health indicators.

**Usage**

```
load_datasus(
  dataset,
  time_period,
  states = "all",
  raw_data = FALSE,
  keep_all = FALSE,
  language = "eng"
)
```

**Arguments**

dataset	A dataset name, can be one of ("datasus_sim_do", "datasus_sih", "datasus_cnes_lt"), or more. For more details, try vignette("DATASUS").
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
states	A string specifying for which states to download the data. It is "all" by default, but can be a single state such as "AC" or any vector such as c("AC", "AM").
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
keep_all	A boolean choosing whether to aggregate the data by municipality, in turn losing individual-level variables (FALSE) or to keep all the original variables. Only applies when raw_data is TRUE.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:
# download raw data for the year 2010 in the state of AM.
data <- load_datasus(
  dataset = "datasus_sim_do",
  time_period = 2010,
  states = "AM",
  raw_data = TRUE
)

# download treated data with the number of deaths by cause in AM and PA.
data <- load_datasus(
  dataset = "datasus_sim_do",
  time_period = 2010,
  states = c("AM", "PA"),
  raw_data = FALSE
)

# download treated data with the number of deaths by cause in AM and PA
```

```
# keeping all individual variables.
data <- load_datasus(
  dataset = "datasus_sim_do",
  time_period = 2010,
  states = c("AM", "PA"),
  raw_data = FALSE,
  keep_all = TRUE
)

## End(Not run)
```

---

load\_degrad

*Degrad - Forest Degradation in the Brazilian Amazon*

---

### Description

Loads information on forest degradation in the Brazilian Amazon, replaced by DETER-B in December 2016.

### Usage

```
load_degrad(
  dataset = "degrad",
  raw_data = FALSE,
  time_period,
  language = "eng"
)
```

### Arguments

dataset	A dataset name ("degrad").
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

### Value

A list of tibbles (if raw\_data = TRUE) or a tibble (if raw\_data = FALSE).

**Examples**

```
## Not run:
# download treated data (raw_data = TRUE) related to forest degradation
# from 2010 to 2012 (time_period = 2010:2012).
data <- load_degrad(
  dataset = "degrad",
  raw_data = FALSE,
  time_period = 2010:2012
)

## End(Not run)
```

---

load\_deter

*DETER - Forest Degradation in the Brazilian Amazon*


---

**Description**

Loads information on changes in forest cover in the Amazon.

**Usage**

```
load_deter(dataset, raw_data = FALSE, language = "eng")
```

**Arguments**

dataset	A dataset name ("deter_amz", "deter_cerrado") with information about both Amazon and Cerrado
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble (if raw\_data = TRUE) or a sf object (if raw\_data = FALSE).

**Examples**

```
## Not run:
# Download treated data (raw_data = FALSE) from Amazonia (dataset = "deter_amz")
deter_amz <- load_deter(
  dataset = "deter_amz",
  raw_data = FALSE
)

## End(Not run)
```

---

load_epe	<i>EPE</i>
----------	------------

---

**Description**

Electrical Energy Monthly Consumption per Class

**Usage**

```
load_epe(dataset, raw_data = FALSE, geo_level = "state", language = "eng")
```

**Arguments**

dataset	A dataset name, ("energy_consumption_per_class") or ("national_energy_balance")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
geo_level	A geographical level, ("state") or ("subsystem"), only available for "energy_consumption_per_class"
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Examples**

```
## Not run:
# download treated data about energy consumption at the state level
clean_epe <- load_epe(
  dataset = "energy_consumption_per_class",
  geo_level = "state",
  raw_data = FALSE
)

## End(Not run)
```

---

load_ibama	<i>IBAMA - Brazilian Institute for the Environment and Renewable Natural Resources</i>
------------	--

---

**Description**

Loads information on environmental fines in the Amazon region

**Usage**

```
load_ibama(dataset, raw_data = FALSE, states = "all", language = "eng")
```

**Arguments**

dataset	A dataset name ("areas_embargadas", "distributed_fines", or "collected_fines")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
states	A string specifying for which states to download the data. It is "all" by default, but can be a single state such as "AC" or any vector such as c("AC", "AM"). Does not apply to the "areas_embargadas" dataset.
language	A string that indicates in which language the data will be returned. Currently, only Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:
# Download treated embargoes data (raw_data = FALSE) in english (language = "eng")
data <- load_ibama(
  dataset = "areas_embargadas", raw_data = FALSE,
  language = "eng"
)

# Download treated collected fines data from "BA"
data <- load_ibama(
  dataset = "collected_fines", raw_data = FALSE,
  states = "BA", language = "pt"
)

## End(Not run)
```

---

load\_iema

*IEMA - Institute of Environment and Water Resources*


---

**Description**

Loads information on electric energy access at the municipality level in the Amazon region

**Usage**

```
load_iema(dataset = "iema", raw_data = FALSE, language = "eng")
```

**Arguments**

dataset	A dataset name ("iema")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:  
# Download treated data  
data <- load_iema(raw_data = FALSE)  
  
## End(Not run)
```

---

load_imazon	<i>IMAZON - Deforestation pressure by municipality</i>
-------------	--

---

**Description**

Loads data categorizing each municipality by the level of deforestation pressure it faces

**Usage**

```
load_imazon(dataset = "imazon_shp", raw_data = FALSE, language = "eng")
```

**Arguments**

dataset	There is one dataset available ("imazon_shp")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:  
# Download treated data  
data <- load_imazon(raw_data = FALSE)  
  
## End(Not run)
```

---

`load_ips`*IPS - Amazon Social Progress Index*

---

**Description**

Loads information on the social and environmental performance of the Legal Amazon.

**Usage**

```
load_ips(  
  dataset = "all",  
  raw_data = FALSE,  
  time_period = c(2014, 2018, 2021),  
  language = "eng"  
)
```

**Arguments**

<code>dataset</code>	A dataset name ("all", "life_quality", "sanit_habit", "violence", "educ", "communic", "mortality", or "deforest")
<code>raw_data</code>	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
<code>time_period</code>	Year to download. Can be 2014, 2018, 2021, or a vector with some combination thereof
<code>language</code>	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:  
# Download raw data from 2014  
data <- load_ips(dataset = "all", raw_data = TRUE, time_period = 2014)  
  
# Download treated deforest data from 2018 in portuguese  
data <- load_ips(  
  dataset = "deforest", raw_data = FALSE,  
  time_period = 2018, language = "pt"  
)  
  
## End(Not run)
```



---

load_mapbiomas	<i>MAPBIOMAS - The Annual Land Cover and Use Mapping Project in Brazil</i>
----------------	--

---

## Description

Loads information about land cover and use

## Usage

```
load_mapbiomas(
  dataset,
  raw_data = FALSE,
  geo_level = "municipality",
  language = "eng",
  cover_level = 1
)
```

## Arguments

dataset	A dataset name ("mapbiomas_cover", "mapbiomas_transition", "mapbiomas_irrigation", "mapbiomas_deforestation_regeneration", "mapbiomas_grazing_quality", or "mapbiomas_mining")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
geo_level	A string that defines the geographic level of the data. <ul style="list-style-type: none"> <li>• For datasets "mapbiomas_cover" and "mapbiomas_transition", can be "municipality" or "state" (faster download).</li> <li>• For dataset "mapbiomas_mining", can be "biome" or "indigenous_land".</li> </ul>
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.
cover_level	A numeric or string that indicates the cover aggregation level. Can be "0", "1", "2", "3", "4", or "none", which means no aggregation. Aggregation only supported for "mapbiomas_cover" and "mapbiomas_grazing_quality" datasets.

## Value

A tibble.

## Examples

```
## Not run:
# download treated Mapbiomas Cover data in english at the highest aggregation level
data <- load_mapbiomas(
  dataset = "mapbiomas_cover",
  raw_data = FALSE,
  geo_level = "municipality",
```

```

  language = "eng",
  cover_level = 0
)

# download treated Mapbiomas Transition data in portuguese
data <- load_mapbiomas(
  dataset = "mapbiomas_transition", raw_data = FALSE,
  geo_level = "state", language = "pt"
)

# download treated data on mining on indigenous lands
data <- load_mapbiomas("mapbiomas_mining",
  raw_data = FALSE,
  geo_level = "indigenous_land"
)

## End(Not run)

```

---

load\_pam

*PAM - Municipal Agricultural Production*


---

## Description

Loads information on the quantity, value and area of temporary and permanent crops cultivated.

## Usage

```
load_pam(dataset, raw_data = FALSE, geo_level, time_period, language = "eng")
```

## Arguments

dataset	A dataset name ("all_crops", "permanent_crops", "temporary_crops" or many individual crop possibilities (see vignette(load_pam))). You can also use SIDRA codes (see <a href="https://sidra.ibge.gov.br/pesquisa/pam/tabelas">https://sidra.ibge.gov.br/pesquisa/pam/tabelas</a> )
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
geo_level	A string that defines the geographic level of the data. Can be one of "country", "state" or "municipality".
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

## Value

A tibble consisting of geographic units that present positive values for any of the variables in the dataset.

**Examples**

```
## Not run:
# download treated data at the state level from 2010 to 2011 for all crops
data <- load_pam(
  dataset = "all_crops",
  raw_data = FALSE,
  geo_level = "state",
  time_period = 2010:2011,
  language = "eng"
)

## End(Not run)
```

load\_pevs

*PEVS - Forestry Activities***Description**

Lloads information on the amount and value of the production of the exploitation of native plant resources and planted forest massifs, as well as existing total and harvested areas of forest crops.

**Usage**

```
load_pevs(dataset, raw_data = FALSE, geo_level, time_period, language = "eng")
```

**Arguments**

dataset	A dataset name ("pevs_forest_crops", "pevs_silviculture" or "pevs_silviculture_area"). You can also use SIDRA codes (see <a href="https://sidra.ibge.gov.br/pesquisa/pevs/quadros/brasil/2019">https://sidra.ibge.gov.br/pesquisa/pevs/quadros/brasil/2019</a> )
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
geo_level	A string that defines the geographic level of the data. Can be one of "country", "region", "state", or "municipality".
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble consisting of geographic units that present positive values for any of the variables in the dataset.

## Examples

```
## Not run:
# Download treated (raw_data = FALSE) silviculture data (dataset = 'pevs_silviculture')
# by state (geo_level = 'state') from 2012 (time_period = 2012)
# in portuguese (language = "pt")
data <- load_pevs(
  dataset = "pevs_silviculture",
  raw_data = FALSE,
  geo_level = "state",
  time_period = 2012,
  language = "pt"
)

# Download raw (raw_data = TRUE) forest crops data by region from 2012 to 2013 in english
data <- load_pevs(
  dataset = "pevs_forest_crops",
  raw_data = TRUE,
  geo_level = "region",
  time_period = 2012:2013
)

## End(Not run)
```

---

load\_pibmunic

*PIB MUNICIPAL - Municipal GDP*

---

## Description

Loads information on gross domestic product at current prices, taxes, net of subsidies, on products at current prices and gross value added at current prices, total and by economic activity, and respective shares.

## Usage

```
load_pibmunic(
  dataset = "pibmunic",
  raw_data = FALSE,
  geo_level,
  time_period,
  language = "eng"
)
```

## Arguments

**dataset** A dataset name ("pibmunic") with Municipal GDP information. You can also use SIDRA codes (See <https://sidra.ibge.gov.br/pesquisa/pib-munic/tabelas>)

raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
geo_level	A string that defines the geographic level of the data. Can be one of "country", "state" or "municipality".
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:
# download treated municipal GDP data at the state level for 2010 to 2012
data <- load_pibmunic(
  raw_data = FALSE,
  geo_level = "state",
  time_period = 2010:2012
)

## End(Not run)
```

---

load\_ppm

*PPM - Municipal Livestock Production*


---

**Description**

Loads information on animal farming inventories and livestock products (IBGE).

**Usage**

```
load_ppm(dataset, raw_data = FALSE, geo_level, time_period, language = "eng")
```

**Arguments**

dataset	A dataset name ("ppm_livestock_inventory", "ppm_sheep_farming", "ppm_animal_orig_production", "ppm_cow_farming" or "ppm_aquaculture". You can also use SIDRA codes (see <a href="https://sidra.ibge.gov.br/pesquisa/ppm/tabelas/brasil/2019">https://sidra.ibge.gov.br/pesquisa/ppm/tabelas/brasil/2019</a> )
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
geo_level	A string that defines the geographic level of the data. Can be one of "country", "state" or "municipality".
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble consisting of geographic units that present positive values for any of the variables in the dataset.

**Examples**

```
## Not run:
# Download treated data (raw_data = FALSE) about aquaculture (dataset = "ppm_aquaculture")
# from 2013 to 2015 (time_period = 2013:2015) in english
# with the level of aggregation being the country (geo_level = "country").
data <- load_ppm(
  dataset = "ppm_aquaculture",
  raw_data = FALSE,
  geo_level = "country",
  time_period = 2013:2015
)

# Download raw data about sheep farming by state from 1980 to 1995 in portuguese (language = "pt")
data <- load_ppm(
  dataset = "ppm_sheep_farming",
  raw_data = TRUE,
  geo_level = "state",
  time_period = 1980:1995,
  language = "pt"
)

## End(Not run)
```

---

load\_prodes

*PRODES - Deforestation Monitoring Project in the Legal Amazon by Satellite*

---

**Description**

Loads information on clearcut deforestation in the Legal Amazon and annual deforestation rates in the region.

**Usage**

```
load_prodes(
  dataset = "prodes",
  raw_data = FALSE,
  time_period,
  language = "eng"
)
```

**Arguments**

dataset	A dataset name ("prodes").
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
time_period	A numeric indicating for which years the data will be loaded, in the format YYYY. Can be any vector of numbers, such as 2010:2012.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble with the selected data.

**Examples**

```
## Not run:
# Download treated data (raw_data = FALSE) from 2010 (time_period = 2010)
# in portuguese (language = 'pt').
data <- load_prodes(
  raw_data = FALSE,
  time_period = 2010,
  language = "pt"
)

## End(Not run)
```

---

load_seeg	<i>Greenhouse gas emission estimates (SEEG)</i>
-----------	---

---

**Description**

Loads data of estimates of emission of greenhouse gases

**Usage**

```
load_seeg(dataset, raw_data = FALSE, geo_level, language = "eng")
```

**Arguments**

dataset	A dataset name ("seeg", "seeg_farming", "seeg_industry", "seeg_energy", "seeg_land", "seeg_residuals"). On which "seeg" contains all five sectors (only works with raw_data = TRUE) and the others are filtered specifically by a main source of emission.
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
geo_level	A string that defines the geographic level of the data. Can be one of "country", "state" or "municipality".
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.

**Examples**

```
## Not run:
# Download raw data (raw_data = TRUE) of greenhouse gases (dataset = "seeg")
# by state (geo_level = "state")
data <- load_seeg(
  dataset = "seeg",
  raw_data = TRUE,
  geo_level = "state"
)

# Download treated data (raw_data = FALSE) of industry greenhouse gases (dataset = "seeg_industry")
data <- load_seeg(
  dataset = "seeg_industry",
  raw_data = FALSE,
  geo_level = "state"
)

## End(Not run)
```

---

load\_sigmine

*SIGMINE - Mining Geographic Information System*

---

**Description**

Loads information the mines being explored legally in Brazil, including their location, status, product being mined and area in square meters.

**Usage**

```
load_sigmine(dataset = "sigmine_active", raw_data = FALSE, language = "eng")
```

**Arguments**

dataset	A dataset name ("sigmine_active")
raw_data	A boolean setting the return of raw (TRUE) or processed (FALSE) data.
language	A string that indicates in which language the data will be returned. Portuguese ("pt") and English ("eng") are supported.

**Value**

A tibble.



**Examples**

```
## Not run:
# Download treated data (raw_data = FALSE) in portuguese (language = "pt").
data <- load_sigmine(
  dataset = "sigmine_active",
  raw_data = FALSE,
  language = "pt"
)

## End(Not run)
```

---

municipalities	<i>IBGE codes and Legal Amazon identification of Brazilian municipalities</i>
----------------	---

---

**Description**

A dataset containing each municipality's IBGE code, state, mesoregion, microregion, as well as a binary variable for whether it is part of the Legal Amazon. Mostly for our functions' internal use.

**Usage**

```
municipalities
```

**Format**

A data frame with 5570 rows and 12 variables:

**code\_muni** IBGE 7-digit municipality code  
**name\_muni** municipality name  
**code\_state** 2-digit state code  
**abbrev\_state** state abbreviations (e.g. "AM")  
**name\_state** full name of the states  
**code\_region** 1-digit regional code  
**name\_region** name of the region  
**legal\_amazon** takes value 1 for municipalities in the legal amazon, 0 otherwise  
**code\_micro** 5-digit microregion code  
**name\_micro** name of the microregion  
**code\_meso** 4-digit mesoregion code  
**name\_meso** name of the mesoregion

**Source**

Package geobr and <https://www.ibge.gov.br/geociencias/cartas-e-mapas/mapas-regionais/15819-amazonia-legal.html?=&t=acesso-ao-produto>

---

read.dbc	<i>Read Data Stored in DBC (Compressed DBF) Files</i>
----------	---

---

### Description

This function allows you to read a DBC (compressed DBF) file into a data frame. Please note that this is the file format used by the Brazilian Ministry of Health (DATASUS), and it is not related to the FoxPro or CANdb DBC file formats.

### Usage

```
read.dbc(file, ...)
```

### Arguments

file	The name of the DBC file (including extension)
...	Further arguments to be passed to read.dbf

### Details

DBC is the extension for compressed DBF files (from the 'XBASE' family of databases). This is a proprietary file format used by the Brazilian government to make available public healthcare datasets (by its agency called DATASUS). read.dbc relies on the dbc2dbf function to decompress the DBC into a temporary DBF file.

After decompressing, it reads the temporary DBF file into a data.frame using read.dbf from the foreign package.

### Value

A data.frame of the data from the DBC file.

### Note

DATASUS is the name of the Department of Informatics of the Brazilian Health System and is responsible for publishing public healthcare data. Besides the DATASUS, the Brazilian National Agency for Supplementary Health (ANS) also uses this file format for its public data.

This function was tested using files from both DATASUS and ANS to ensure compliance with the format, and hence ensure its usability by researchers.

As a final note, neither this project, nor its author, has any association with the Brazilian government.

### Author(s)

Daniela Petruzalek, <daniela.petruzalek@gmail.com>

### See Also

dbc2dbf

**Examples**

```
## Not run:
# The 'sids.dbc' file is the compressed version of 'sids.dbf' from the "foreign" package.
x <- read.dbc(system.file("files/sids.dbc", package = "read.dbc"))
str(x)
summary(x)

# This is a small subset of U.S. NOAA storm database.
storm <- read.dbc(system.file("files/storm.dbc", package = "read.dbc"))
head(x)
str(x)

## Don't run!
## The following code will download data from the "Declarations of Death" database for
## the Brazilian state of Parana, year 2013. Source: DATASUS / Brazilian Ministry of Health
url <- "ftp://ftp.datasus.gov.br/dissemin/publicos/SIM/CID10/DORES/DOPR2013.dbc"
download.file(url, destfile = "DOPR2013.dbc")
dopr <- read.dbc("DOPR2013.dbc")
head(dopr)
str(dopr)

## End(Not run)
```

# Index

- \* **datasets**
  - municipalities, [25](#)
- \* **datasus**
  - read.dbc, [26](#)
- \* **dbc**
  - dbc2dbf, [2](#)
  - read.dbc, [26](#)
- \* **dbf**
  - dbc2dbf, [2](#)

dbc2dbf, [2](#)

load\_aneel, [4](#)  
load\_baci, [4](#)  
load\_br\_trade, [5](#)  
load\_cempre, [6](#)  
load\_cipo, [7](#)  
load\_climate, [8](#)  
load\_datasus, [9](#)  
load\_degrad, [11](#)  
load\_deter, [12](#)  
load\_epe, [13](#)  
load\_ibama, [13](#)  
load\_iema, [14](#)  
load\_imazon, [15](#)  
load\_ips, [16](#)  
load\_mapbiomas, [17](#)  
load\_pam, [18](#)  
load\_pevs, [19](#)  
load\_pibmunic, [20](#)  
load\_ppm, [21](#)  
load\_prodes, [22](#)  
load\_seeg, [23](#)  
load\_sigmine, [24](#)

municipalities, [25](#)

read.dbc, [26](#)