Package 'tidyBdE'

May 21, 2026

Title Retrieve Data from 'Banco de España'

Version 0.6.1

Description Tools for retrieving time series data from 'Banco de España' ('BdE') as 'tibble' objects. 'Banco de España' is the national central bank and, within the framework of the Single Supervisory Mechanism ('SSM'), the supervisor of the Spanish banking system alongside the European Central Bank. This package is not sponsored, endorsed or administered by 'Banco de España'.

License GPL (>= 3)

URL https://ropenspain.github.io/tidyBdE/, https://github.com/rOpenSpain/tidyBdE

BugReports https://github.com/rOpenSpain/tidyBdE/issues

Depends R (>= 4.1.0)

Imports dplyr (>= 0.7.0), ggplot2 (>= 3.5.0), readr (>= 1.0.0), scales (>= 1.1.0), tibble (>= 3.0.0), tidyr, utils

Suggests knitr, lifecycle, quarto, testthat (>= 3.0.0)

VignetteBuilder quarto

Config/Needs/coverage covr

Config/Needs/website cpp11, devtools, progress, reactable, remotes, styler, tidyverse, ropenspain/rostemplate

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/testthat/parallel true

Copyright See file inst/COPYRIGHTS

Encoding UTF-8

LazyData true

X-schema.org-applicationCategory Macroeconomics

X-schema.org-isPartOf https://ropenspain.es/

X-schema.org-keywords api, bde, cran, ggplot2, macroeconomics, r, r-package, ropenspain, rstats, series-data, spain

NeedsCompilation no

Author Diego H. Herrero [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-8457-4658>)

Maintainer Diego H. Herrero <dev.dieghernan@gmail.com>

Repository CRAN

Date/Publication 2026-05-21 12:50:02 UTC

Contents

`bde_catalog_load` *Load BdE catalog metadata*

Description

Load BdE time series catalog metadata.

Usage

```
bde_catalog_load(
  catalog = c("ALL", "BE", "SI", "TC", "TI", "PB"),
  parse_dates = TRUE,
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE
)
```

Arguments

`catalog`	A single catalog identifier to load, or `"ALL"` to load every catalog. See **Details**.
`parse_dates`	Logical. If `TRUE`, date columns are parsed with `bde_parse_dates()`.
`cache_dir`	A path to a cache directory. The directory can also be set with options using `options(bde_cache_dir = "path/to/dir")`.
`update_cache`	Logical. If `TRUE`, the requested file is refreshed in `cache_dir`.
`verbose`	Logical. If `TRUE`, display information useful for debugging.

Details

Accepted values for `catalog` are:

CODE	PUBLICATION	UPDATE FREQUENCY	FREQUENCY
`"BE"`	Statistical Bulletin	Daily	Monthly
`"SI"`	Summary Indicators	Daily	Daily
`"TC"`	Exchange Rates	Daily	Daily
`"TI"`	Interest Rates	Daily	Daily
`"PB"`	Bank Lending Survey	Quarterly	Quarterly

Use `"ALL"` as a shorthand for loading all catalogs at once.

If the requested catalog is not cached, this function calls `bde_catalog_update()`.

Value

A tibble with the requested catalog metadata.

Source

time series bulk data download.

See Also

Other catalog: `bde_catalog_search()`, `bde_catalog_update()`

Examples

```
bde_catalog_load("TI", verbose = TRUE)
```

bde_catalog_search *Search BdE catalogs*

Description

Search BdE time series catalog metadata for keywords.

Usage

```
bde_catalog_search(pattern, ...)
```

Arguments

pattern	regex pattern to search. See **Details** and **Examples**.
...	Arguments passed on to bde_catalog_load catalog A single catalog identifier to load, or "ALL" to load every catalog. See **Details**. parse_dates Logical. If TRUE, date columns are parsed with bde_parse_dates(). update_cache Logical. If TRUE, the requested file is refreshed in cache_dir. cache_dir A path to a cache directory. The directory can also be set with options using options(bde_cache_dir = "path/to/dir"). verbose Logical. If TRUE, display information useful for debugging.

Details

Note: BdE metadata is currently provided in Spanish only. Therefore, search terms must be provided in Spanish to retrieve results.

This function uses base::grep() to find matches in the catalogs. You can pass regular expressions to broaden the search.

Value

A tibble object with the results of the query.

See Also

bde_catalog_load(), base::regex

Other catalog: bde_catalog_load(), bde_catalog_update()

Examples

```
# Simple search. Search terms must be in Spanish.
# PIB [es] == GDP [en].

bde_catalog_search("PIB")
```

```
# Search with a single complex condition.
bde_catalog_search("Francia(.*)PIB")

# Search with multiple complex conditions.
bde_catalog_search("Francia(.*)PIB|Italia(.*)PIB|Alemania(.*)PIB")
```

`bde_catalog_update` *Update BdE catalog files*

Description

Update BdE time series catalog files.

Usage

```
bde_catalog_update(
  catalog = c("ALL", "BE", "SI", "TC", "TI", "PB"),
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

`catalog`	A single catalog identifier to update, or `"ALL"` to update every catalog. See **Details**.
`cache_dir`	A path to a cache directory. The directory can also be set with options using `options(bde_cache_dir = "path/to/dir")`.
`verbose`	Logical. If `TRUE`, display information useful for debugging.

Details

Accepted values for `catalog` are:

CODE	PUBLICATION	UPDATE FREQUENCY	FREQUENCY
`"BE"`	Statistical Bulletin	Daily	Monthly
`"SI"`	Summary Indicators	Daily	Daily
`"TC"`	Exchange Rates	Daily	Daily
`"TI"`	Interest Rates	Daily	Daily
`"PB"`	Bank Lending Survey	Quarterly	Quarterly

Use `"ALL"` as a shorthand for updating all catalogs at once.

Value

An invisible list of download results.

Source

time series bulk data download.

See Also

Other catalog: `bde_catalog_load()`, `bde_catalog_search()`

Examples

```
bde_catalog_update("TI", verbose = TRUE)
```

`bde_indicators` *Selected Spanish macroeconomic indicators*

Description

Convenience functions for downloading selected Spanish macroeconomic indicators. Metadata is available in bde_ind_db.

Usage

```
bde_ind_gdp_var(series_label = "GDP_YoY", ...)

bde_ind_unemployment_rate(series_label = "Unemployment_Rate", ...)

bde_ind_euribor_12m_monthly(series_label = "Euribor_12M_Monthly", ...)

bde_ind_euribor_12m_daily(series_label = "Euribor_12M_Daily", ...)

bde_ind_cpi_var(series_label = "Consumer_price_index_YoY", ...)

bde_ind_ibex_monthly(series_label = "IBEX_index_month", ...)

bde_ind_ibex_daily(series_label = "IBEX_index_day", ...)

bde_ind_gdp_quarterly(series_label = "GDP_quarterly_value", ...)

bde_ind_population(series_label = "Population_Spain", ...)
```

Arguments

series_label	Optional character string or vector of labels to assign to the extracted series.
...	Arguments passed on to bde_series_load
	out_format The format to return, either "wide" or "long". See **Value** for details and the **Examples** section.
	parse_numeric Logical. If TRUE, the columns are parsed to double (numeric) values. See **Note**.
	extract_metadata Logical. If TRUE, the output is the metadata of the requested series.
	parse_dates Logical. If TRUE, date columns are parsed with bde_parse_dates().
	update_cache Logical. If TRUE, the requested file is refreshed in cache_dir.
	cache_dir A path to a cache directory. The directory can also be set with options using options(bde_cache_dir = "path/to/dir").
	verbose Logical. If TRUE, display information useful for debugging.

Details

These functions are convenient wrappers around bde_series_load() for specific series. Use verbose = TRUE, extract_metadata = TRUE to inspect the metadata and source.

Value

A tibble with the required series.

See Also

bde_series_load(), bde_catalog_search()

Other indicators: bde_ind_db

Examples

```
bde_ind_gdp_var()
```

---

bde_ind_db *Database of selected Spanish macroeconomic indicators*

Description

Minimal metadata for the selected Spanish macroeconomic indicators included in the convenience functions of **tidyBdE** (see bde_indicators). Full metadata can be accessed with bde_catalog_load().

Format

A tibble of 9 rows and 7 columns with the following fields:

tidyBdE_fun Function name, see bde_indicators.

Numero_secuencial Series code, see `bde_series_load()`.

Descripcion_de_la_serie Description of the series in Spanish.

Fecha_de_la_primera_observacion Starting date of the indicator.

Fecha_de_la_ultima_observacion Most recent date available.

Fuente Data source.

Details

tidyBdE_fun	Numero_secuencial	Descripcion_de_la_serie
bde_ind_cpi_var	1489713	Estadísticas Generales. IPCA. Base 2015. Índice general. Tasa inter
bde_ind_euribor_12m_daily	905842	Tipo de interés. UEM. Mercado monetario. Euríbor. A 12 meses
bde_ind_euribor_12m_monthly	587853	Tipo de interés. UEM. Mercado monetario. Euríbor. A 12 meses
bde_ind_gdp_quarterly	4663160	Estadísticas Generales. Cuentas Nacionales. SEC2010. Año base 20
bde_ind_gdp_var	4663788	Estadísticas Generales. CNTR. Base 2020. PIB. Precios constantes.
bde_ind_ibex_daily	821340	Cotización y contratación. Acciones. Sociedad de Bolsas y Socieda
bde_ind_ibex_monthly	254433	Cotización y contratación. Acciones. Sociedad de Bolsas y Socieda
bde_ind_population	4637737	Estadísticas generales. INE. EPA. Base 2021. Total Nacional. Ambo
bde_ind_unemployment_rate	4635980	Estadísticas Generales. EPA. Base 2021. Total Nacional. Tasa de pa

See Also

Other indicators: `bde_indicators`

Examples

```
data("bde_ind_db")
bde_ind_db
```

`bde_parse_dates` *Parse dates from strings*

Description

This function is tailored to date formats used in this package and may fail with other datasets. See **Examples** for formats that are supported.

Date formats:

FREQUENCY	FORMAT
Daily / Business day	DD MMMMYYYY
Monthly	MMM YYYY
Quarterly	MMM YYYY, where MMM is the first or the last month of the quarter, depending on the value of its v
Half-yearly	MMM YYYY, where MMM is the first or the last month of the half-year period, depending on the val
Annual	YYYY

Usage

```
bde_parse_dates(dates_to_parse)
```

Arguments

`dates_to_parse`	Character vector of dates to parse.

Details

Parse strings representing dates with `as.Date()`.

Value

A vector of `Date` values.

See Also

`as.Date()`

Examples

```
# Supported formats.
would_parse <- c(
  "02 FEB2019", "15 ABR 1890", "MAR 2020", "ENE2020",
  "2020", "12-1993", "01-02-2014", "01/02/1990"
)

parsed_ok <- bde_parse_dates(would_parse)

class(parsed_ok)

tibble::tibble(raw = would_parse, parsed = parsed_ok)

# Unsupported formats.

wont_parse <- c("JAN2001", "2010-01-12", "01 APR 2017", "01/31/1990")
```

```
parsed_fail <- bde_parse_dates(wont_parse)

class(parsed_fail)

tibble::tibble(raw = wont_parse, parsed = parsed_fail)
```

`bde_series_full_load` *Load BdE full time series files*

Description

Load a full BdE time series file.

Usage

```
bde_series_full_load(
  series_csv,
  parse_dates = TRUE,
  parse_numeric = TRUE,
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE,
  extract_metadata = FALSE
)
```

Arguments

`series_csv`	CSV file of a series, as defined in the field `Nombre del archivo con los valores de la serie` of the corresponding catalog. See `bde_catalog_load()`.
`parse_dates`	Logical. If `TRUE`, date columns are parsed with `bde_parse_dates()`.
`parse_numeric`	Logical. If `TRUE`, the columns are parsed to double (numeric) values. See **Note**.
`cache_dir`	A path to a cache directory. The directory can also be set with options using `options(bde_cache_dir = "path/to/dir")`.
`update_cache`	Logical. If `TRUE`, the requested file is refreshed in `cache_dir`.
`verbose`	Logical. If `TRUE`, display information useful for debugging.
`extract_metadata`	Logical. If `TRUE`, the output is the metadata of the requested series.

Details

About BdE file naming:

The series name is a positional code showing the location of the table. For example, table **be_6_1** represents Table 1, Chapter 6 of the Statistical Bulletin ("BE"). Although it is unique, it is subject to change, for example when a new table is inserted before it.

For that reason, `bde_series_load()` is more suitable for extracting specific time series.

Value

A tibble with a Date field and the aliases of the series fields as described in the catalogs. See bde_catalog_load().

Note

This function tries to coerce the columns to numbers. For some series, a warning may be displayed if the parser fails. You can override the default behavior with parse_numeric = FALSE.

See Also

Other series: bde_series_load()

Examples

```
# Show metadata.
bde_series_full_load("TI_1_1.csv", extract_metadata = TRUE)

# Load data.
bde_series_full_load("TI_1_1.csv")
```

---

bde_series_load *Load a single BdE time series*

Description

The series alias is a positional code showing the location (column and/or row) of the series in the table. Although it is unique, it is not stable enough to use as the series ID because it may change when the series moves.

To ensure series can still be identified after these changes, they are assigned a sequential number, referred to as series_code in this function.

Note that a single series may appear in different tables, so it can have several aliases. If you need to search by alias, use bde_series_full_load().

Usage

```
bde_series_load(
  series_code,
  series_label = NULL,
  out_format = "wide",
  parse_dates = TRUE,
  parse_numeric = TRUE,
  cache_dir = NULL,
  update_cache = FALSE,
```

```
  verbose = FALSE,
  extract_metadata = FALSE
)
```

Arguments

series_code	A numeric value, or one coercible with base::as.double(), or a vector of time series codes, as defined in the field Número secuencial of the corresponding series. See bde_catalog_load().
series_label	Optional character string or vector of labels to assign to the extracted series.
out_format	The format to return, either "wide" or "long". See **Value** for details and the **Examples** section.
parse_dates	Logical. If TRUE, date columns are parsed with bde_parse_dates().
parse_numeric	Logical. If TRUE, the columns are parsed to double (numeric) values. See **Note**.
cache_dir	A path to a cache directory. The directory can also be set with options using options(bde_cache_dir = "path/to/dir").
update_cache	Logical. If TRUE, the requested file is refreshed in cache_dir.
verbose	Logical. If TRUE, display information useful for debugging.
extract_metadata	Logical. If TRUE, the output is the metadata of the requested series.

Details

Load a single BdE time series.

Value

A tibble with a Date column:

- With out_format = "wide", each series is presented in a separate column with the name defined by series_label.
- With out_format = "long", the tibble has two additional columns: serie_name, with the label of each series, and serie_value, with the corresponding value.

"wide" format is more suitable for exporting to a .csv file, while "long" format is more suitable for creating plots using ggplot2::ggplot(). See also tidyr::pivot_longer() and tidyr::pivot_wider().

Note

This function attempts to coerce the columns to numbers. For some series, a warning may be displayed if the parsing fails.

See Also

bde_catalog_load(), bde_catalog_search(), bde_indicators()

Other series: bde_series_full_load()

Examples

```
# Show metadata.
bde_series_load(573234, verbose = TRUE, extract_metadata = TRUE)

# Load data.
bde_series_load(573234, extract_metadata = FALSE)

# Load multiple series.
bde_series_load(c(573234, 573214),
  series_label = c("US/EUR", "GBP/EUR"),
  extract_metadata = TRUE
)

wide <- bde_series_load(c(573234, 573214),
  series_label = c("US/EUR", "GBP/EUR")
)

# Show wide output.
wide

# Show long output.
long <- bde_series_load(c(573234, 573214),
  series_label = c("US/EUR", "GBP/EUR"),
  out_format = "long"
)

long

# Use with `ggplot2`.
library(ggplot2)

ggplot(long, aes(Date, serie_value)) +
  geom_line(aes(group = serie_name, color = serie_name)) +
  scale_color_bde_d() +
  theme_tidybde()
```

`bde_tidy_palettes` *BdE color palettes*

Description

Manually defined palettes based on BdE publications. Each palette contains at most six colors.

Usage

```
bde_tidy_palettes(
```

```
  n = 6,
  palette = c("bde_vivid_pal", "bde_rose_pal", "bde_qual_pal"),
  alpha = NULL,
  rev = FALSE
)
```

Arguments

n	The number of colors (>= 1) to return.
palette	A valid palette name.
alpha	An alpha transparency level in the range [0, 1] (0 means transparent and 1 means opaque). If missing (i.e., alpha = NULL), the function does not append opacity codes ("FF") to the individual color hex codes. See ggplot2::alpha().
rev	Logical indicating whether to reverse the color order.

Value

A character vector of hex color codes.

See Also

Other bde_plot: scales_bde, theme_tidybde()

Examples

```
# Show the BdE vivid palette.
scales::show_col(bde_tidy_palettes(palette = "bde_vivid_pal"),
  labels = FALSE
)

# Show the BdE rose palette.
scales::show_col(bde_tidy_palettes(palette = "bde_rose_pal"),
  labels = FALSE
)

# Show the BdE qualitative palette.
scales::show_col(bde_tidy_palettes(palette = "bde_qual_pal"),
  labels = FALSE
)
```

---

scales_bde *BdE color scales*

Description

Color scales for the **ggplot2** package. Discrete scales are named scale_*_bde_d, while continuous palettes are named scale_*_bde_c.

Usage

```
scale_color_bde_d(
  palette = c("bde_vivid_pal", "bde_rose_pal", "bde_qual_pal"),
  alpha = NULL,
  rev = FALSE,
  ...
)

scale_fill_bde_d(
  palette = c("bde_vivid_pal", "bde_rose_pal", "bde_qual_pal"),
  alpha = NULL,
  rev = FALSE,
  ...
)

scale_color_bde_c(
  palette = c("bde_rose_pal", "bde_vivid_pal", "bde_qual_pal"),
  alpha = NULL,
  rev = FALSE,
  guide = "colorbar",
  ...
)

scale_fill_bde_c(
  palette = c("bde_rose_pal", "bde_vivid_pal", "bde_qual_pal"),
  alpha = NULL,
  rev = FALSE,
  guide = "colorbar",
  ...
)
```

Arguments

palette	BdE palette to apply. See bde_tidy_palettes() for details.
alpha	An alpha transparency level in the range [0, 1] (0 means transparent and 1 means opaque). If missing (i.e., alpha = NULL), the function does not append opacity codes ("FF") to the individual color hex codes. See ggplot2::alpha().
rev	Logical indicating whether to reverse the color order.
...	Additional arguments passed to ggplot2::discrete_scale() or ggplot2::continuous_scale().
guide	A function used to create a guide or its name. See guides() for more information.

Value

A **ggplot2** scale object.

See Also

ggplot2::discrete_scale(), ggplot2::continuous_scale()

Other bde_plot: bde_tidy_palettes(), theme_tidybde()

Examples

```
library(ggplot2)

set.seed(596)
txsamp <- subset(
  txhousing,
  city %in% c(
    "Houston", "Fort Worth",
    "San Antonio", "Dallas", "Austin"
  )
)

ggplot(txsamp, aes(x = sales, y = median)) +
  geom_point(aes(colour = city)) +
  scale_color_bde_d() +
  theme_minimal()

ggplot(txsamp, aes(x = sales, y = median)) +
  geom_point(aes(colour = city)) +
  scale_color_bde_d("bde_qual_pal") +
  theme_minimal()
```

---

theme_tidybde *BdE ggplot2 theme*

Description

Custom **ggplot2** theme based on BdE publications.

Usage

```
theme_tidybde(...)
```

Arguments

...	Arguments passed on to ggplot2::theme_classic
	base_size base font size, given in pts.
	base_family base font family
	header_family font family for titles and headers. The default, NULL, uses theme inheritance to set the font. This setting affects axis titles, legend titles, the plot title and tag text.

`base_line_size` base size for line elements

`base_rect_size` base size for rect elements

`ink, paper, accent` colour for foreground, background, and accented elements respectively.

Details

This theme is based on `ggplot2::theme_classic()`.

Value

A **ggplot2** theme object.

See Also

`ggplot2::theme_classic()`

Other bde_plot: `bde_tidy_palettes()`, `scales_bde`

Examples

```
library(ggplot2)
library(dplyr)
library(tidyr)

series_TC <- bde_series_full_load("TC_1_1.csv")

# Plot if the download succeeds.
if (nrow(series_TC) > 0) {
  series_TC <- series_TC[c(1, 2)]

  series_TC_pivot <- series_TC |>
    filter(
      Date >= "2020-01-01" & Date <= "2020-12-31",
      !is.na(series_TC[[2]])
    )

  names(series_TC_pivot) <- c("x", "y")

  ggplot(series_TC_pivot, aes(x = x, y = y)) +
    geom_line(linewidth = 0.8, color = bde_tidy_palettes(n = 1)) +
    labs(
      title = "Title",
      subtitle = "Some metric",
      caption = "Bank of Spain"
    ) +
    theme_tidybde()
}
```

Index