Package 'tidyBdE'

August 25, 2026

Title Retrieve Time Series Data from 'Banco de España'

Version 0.7.1

Description Tools for retrieving 'Banco de España' ('BdE') time series data as 'tibble' objects from bulk CSV files and the 'Statistics web service (API)'. Bulk CSV functions use stable 'BdE' sequential numbers, while API functions use API series codes. Catalog functions support discovery and local caching. Plotting helpers provide 'ggplot2' palettes, scales and themes. '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 cli, dplyr (>= 0.7.0), ggplot2 (>= 3.5.0), jsonlite, lifecycle, readr (>= 1.0.0), scales (>= 1.1.0), tidyr, utils

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

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 8.1.0

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>), Banco de España [cph] (ROR: <https://ror.org/02f26yq04>, Data copyright holder)

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

Repository CRAN

Date/Publication 2026-08-25 15:40:02 UTC

Contents

bde_catalogs 2
bde_indicators 4
bde_ind_db 6
bde_parse_dates 7
bde_series 8
bde_series_api 11
bde_tidy_palettes 13
scales_bde 15
theme_tidybde 16

Index **19**

`bde_catalogs` *Load, update and search BdE catalog metadata*

Description

These functions manage BdE time series catalog metadata from the bulk CSV files published by Banco de España.

`bde_catalog_load()` loads time series catalog metadata into a tibble, `bde_catalog_update()` refreshes the cached catalog files and `bde_catalog_search()` searches catalog metadata for keywords.

Usage

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

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

bde_catalog_search(pattern, ...)
```

Arguments

`catalog`	A catalog identifier or `"ALL"` to load or update every catalog. See **Details**.
`parse_dates`	Logical. If `TRUE`, date columns are parsed with `bde_parse_dates()`.
`cache_dir`	Path to a cache directory. The directory can also be set with `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 informative messages.
`pattern`	Regular expression to search for. See **Details** and **Examples**.
`...`	Additional arguments passed from `bde_catalog_search()` to `bde_catalog_load()`.

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 or updating all catalogs at once.

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

Note: BdE catalog metadata is currently available in Spanish only. Therefore, search terms passed to `bde_catalog_search()` must be in Spanish to retrieve results.

`bde_catalog_search()` uses `base::grep()` to find matches in the catalog metadata. You can pass regular expressions to broaden the search.

Value

`bde_catalog_load()` returns a tibble with the requested time series catalog metadata. See `vignette("csv_manual", package = "tidyBdE")` for details.

`bde_catalog_update()` returns an invisible list of download results.

`bde_catalog_search()` returns a tibble with matching catalog rows.

Source

Banco de España time series bulk data download.

See Also

Time series functions: `bde_series`, `bde_series_api`

Examples

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

# Simple search. Search terms must be in Spanish.
# PIB is the Spanish equivalent of GDP.
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("TI", verbose = TRUE)
```

---

bde_indicators *Selected Spanish macroeconomic indicators*

Description

These functions retrieve 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`	An optional character vector of labels to assign to the extracted series.
`...`	Arguments passed on to `bde_series`
	`out_format` The output format, either `"wide"` or `"long"`. See **Value** for details and the **Examples** section.
	`parse_numeric` Logical. If `TRUE`, parse observation columns as double vectors. See **Note**.
	`extract_metadata` Logical. If `TRUE`, return the metadata associated with 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` Path to a cache directory. The directory can also be set with `options(bde_cache_dir = "path/to/dir")`.
	`verbose` Logical. If `TRUE`, display informative messages.

Details

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

Value

A tibble with the requested indicator series.

Note

These functions attempt to parse columns as double values. For some time series, a warning may be displayed if parsing fails. Set `parse_numeric = FALSE` to disable numeric parsing.

Source

Banco de España time series bulk data download.

See Also

bde_series_load() for loading arbitrary bulk CSV series and bde_catalog_search() for finding series in catalog metadata.

Selected indicators and metadata: bde_ind_db

Examples

```
bde_ind_gdp_var()
```

---

bde_ind_db *Database of selected Spanish macroeconomic indicators*

Description

This dataset contains metadata for the selected Spanish macroeconomic indicators available through the convenience functions in this package. See indicator wrappers. Full catalog metadata is available with bde_catalog_load().

Format

A tibble of 9 rows and 8 columns with the following fields:

tidyBdE_fun Function name. See indicator wrappers.

Numero_secuencial Sequential number. See bde_series_load().

Nombre_de_la_serie API series code. See API series functions.

Descripcion_de_la_serie Description of the series in Spanish.

Fecha_de_la_primera_observacion First observation date of the indicator.

Fecha_de_la_ultima_observacion Most recent observation date.

Fuente Data source.

Details

tidyBdE_fun	Numero_secuencial	Nombre_de_la_serie	Descripcion_de_la_s
bde_ind_cpi_var	1489713	DPU1NEAC920	Estadísticas Generales
bde_ind_euribor_12m_daily	905842	DPUDNBAF172	Tipo de interés. UEM
bde_ind_euribor_12m_monthly	587853	D_1NBAF472	Tipo de interés. UEM
bde_ind_gdp_quarterly	4663160	DSPC102020CB1QB00_SS1_TSC.T	Estadísticas Generales
bde_ind_gdp_var	4663788	DSPC102020VB1QB00_SS1_TSCTVA.T	Estadísticas Generales
bde_ind_ibex_daily	821340	DBLEESI100000000IBDI.B	Cotización y contratac
bde_ind_ibex_monthly	254433	D_1TGC000A	Cotización y contratac
bde_ind_population	4637737	DEPA202110.T	Estadísticas generales

bde_ind_unemployment_rate	4635980	DEPA202140_TSPA.T	Estadísticas Generales

See Also

`vignette("csv_manual", package = "tidyBdE")`.

Selected indicators and metadata: `bde_indicators`

Examples

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

`bde_parse_dates` *Parse date strings*

Description

Parses date strings with `as.Date()`. This function is tailored to the date formats used in this package and may not parse other datasets. See **Examples** for supported formats.

Usage

```
bde_parse_dates(dates_to_parse)
```

Arguments

`dates_to_parse` A character vector of date strings to parse.

Details

Date formats:

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

See `vignette("csv_manual", package = "tidyBdE")` for details.

Value

A vector of `Date` values.

See Also

- bde_catalog_load() and bde_series_load() use this parser.
- as.Date() provides base R date conversion.

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)

dplyr::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)

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

---

bde_series *Load BdE time series from bulk CSV files*

Description

These functions load BdE time series from the bulk CSV files published by Banco de España.

bde_series_load() extracts one or more series by their stable sequential number. bde_series_full_load() loads a complete bulk CSV file and returns all series included in that file.

See vignette("csv_manual", package = "tidyBdE") for details.

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
)

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_code	A numeric vector of stable BdE sequential numbers, or values coercible with base::as.double(), from the Numero_secuencial field of the corresponding series. This is not the API series code. See bde_catalog_load().
series_label	An optional character vector of labels to assign to the extracted series.
out_format	The output format, 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, parse observation columns as double vectors. See **Note**.
cache_dir	Path to a cache directory. The directory can also be set with 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 informative messages.
extract_metadata	Logical. If TRUE, return the metadata associated with the requested series.
series_csv	A bulk CSV file name for a series, as defined in the field Nombre_del_archivo_con_los_valores_de_la of the corresponding catalog. See bde_catalog_load() and the **About BdE file naming** section.

Value

bde_series_load() returns 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 contains the label of each series. serie_value contains the corresponding value.

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

`bde_series_full_load()` returns a tibble with a `Date` column and the aliases of the time series columns as described in catalog metadata. See `bde_catalog_load()` and `vignette("csv_manual", package = "tidyBdE")` for details.

About BdE file naming

A series alias is a positional code that identifies the location, column or row of a series in a table. An alias is unique within its context but is not stable because it may change when a series moves.

A single time series may appear in different tables, so it can have several aliases. Use `bde_series_full_load()` when you need to work with aliases or load a complete file.

The series alias is also used in full CSV files. 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.

Series identifiers

BdE identifies each series with a stable sequential number (`Numero_secuencial`) in bulk CSV files and an API series code (`Nombre_de_la_serie`) in the Statistics web service. `bde_series_load()` accepts stable sequential numbers in `series_code`. `bde_series_api_latest()` and `bde_series_api_load()` use the same argument for API series codes. Use `bde_catalog_load()` or `bde_catalog_search()` to find both identifiers.

Note

These functions attempt to parse columns as double values. For some time series, a warning may be displayed if parsing fails. Set `parse_numeric = FALSE` to disable numeric parsing.

Source

Banco de España time series bulk data download.

See Also

Indicator wrappers retrieve commonly used Spanish macroeconomic series.

Time series functions: `bde_catalogs`, `bde_series_api`

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()

# Show metadata for a complete bulk CSV file.
bde_series_full_load("TI_1_1.csv", extract_metadata = TRUE)

# Load a complete bulk CSV file.
bde_series_full_load("TI_1_1.csv")
```

`bde_series_api` *Load BdE time series from the Statistics web service (API)*

Description

[Experimental]

These functions retrieve BdE time series using the **Statistics web service (API)**.

The API is a JSON web service that provides URL-based access to information available in the Statistics section of Banco de España and the BIEST application.

The API defines two request types. `bde_series_api_latest()` uses the Latest Data request to obtain the latest published observation for one or more series. `bde_series_api_load()` uses the Series List request to obtain the details of one or more complete series and their metadata.

Usage

```
bde_series_api_latest(series_code, language = c("en", "es"), verbose = FALSE)
```

```
bde_series_api_load(
  series_code,
  series_label = NULL,
  out_format = "wide",
  language = c("en", "es"),
  time_range = NULL,
  verbose = FALSE,
  extract_metadata = FALSE
)
```

Arguments

series_code	A character vector of API series codes from the Nombre_de_la_serie field of the corresponding catalog. These values are passed to the API series_list parameter. They are not the stable sequential numbers used by bde_series_load().
language	A string specifying the output language: "es" for Spanish or "en" for English.
verbose	Logical. If TRUE, display informative messages.
series_label	An optional character vector of labels to assign to the extracted series.
out_format	The output format, either "wide" or "long". See **Value** for details and the **Examples** section.
time_range	An optional string specifying a year or API range code. It can be a year, such as "2024", or one of "3M", "12M", "30M", "36M", "60M" or "MAX". If NULL, the API returns the smallest range for the series frequency. Range codes are validated against the frequency returned by bde_series_api_latest(). See **Details**.
extract_metadata	Logical. If TRUE, return the metadata associated with the requested series.

Details

Allowed time_range values depend on the series frequency:

- Daily frequency (D): "3M" (last 3 months), "12M" and "36M".
- Monthly frequency (M): "30M", "60M" and "MAX" (entire series).
- Quarterly frequency (Q): "30M", "60M" and "MAX".
- Annual frequency (A): "60M" and "MAX".

If time_range is not specified, the request returns the smallest range for the series frequency. For example, monthly series return "30M".

Value

bde_series_api_latest() returns a tibble with the latest published observation for each valid series. It includes fields returned by the Latest Data request such as serie, descripcionCorta, codFrecuencia, decimales, simbolo, tendencia, fechaValor and valor.

bde_series_api_load() returns a tibble. When extract_metadata = FALSE, API dates are parsed as Date values and observations are returned in wide or long format according to out_format. When extract_metadata = TRUE, it returns one row per valid series with fields returned by the Series List request, including fechaInicio, fechaFin and metadata fields derived from informacion.

Series identifiers

BdE identifies each series with a stable sequential number (`Numero_secuencial`) in bulk CSV files and an API series code (`Nombre_de_la_serie`) in the Statistics web service. `bde_series_load()` accepts stable sequential numbers in `series_code`. `bde_series_api_latest()` and `bde_series_api_load()` use the same argument for API series codes. Use `bde_catalog_load()` or `bde_catalog_search()` to find both identifiers.

See Also

Time series functions: `bde_catalogs`, `bde_series`

Examples

```
xr <- bde_catalog_load(catalog = "TC")

# Extract the latest value.
library(dplyr)
xr |>
  slice_head(n = 3) |>
  pull(Nombre_de_la_serie) |>
  bde_series_api_latest(language = "en") |>
  glimpse()

# Extract the last 12 months.
xr |>
  slice_head(n = 1) |>
  pull(Nombre_de_la_serie) |>
  bde_series_api_load(language = "en", time_range = "12M") |>
  glimpse()

# Extract metadata.
xr |>
  slice_head(n = 1) |>
  pull(Nombre_de_la_serie) |>
  bde_series_api_load(
    language = "en", time_range = "12M",
    extract_metadata = TRUE
  ) |>
  glimpse()
```

---

bde_tidy_palettes *BdE color palettes*

Description

These palettes are manually defined 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 to return. Must be at least 1.
palette	A palette name: "bde_vivid_pal", "bde_rose_pal" or "bde_qual_pal".
alpha	Alpha transparency level in the range [0, 1], where 0 is transparent and 1 is opaque. If alpha = NULL, the function does not append opacity codes ("FF") to individual color hex codes. See ggplot2::alpha().
rev	Logical. If TRUE, reverse the color order.

Value

A character vector of hex color codes.

See Also

Plotting functions: 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

These functions provide color scales for **ggplot2**. Discrete scales are named `scale_*_bde_d`. Continuous scales 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`	A BdE palette to apply. See `bde_tidy_palettes()` for details.
`alpha`	Alpha transparency level in the range `[0, 1]`, where `0` is transparent and `1` is opaque. If `alpha = NULL`, the function does not append opacity codes (`"FF"`) to individual color hex codes. See `ggplot2::alpha()`.

rev	Logical. If TRUE, 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() and ggplot2::continuous_scale() for the underlying scale constructors.

Plotting functions: 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 theme for* R*hrefhttps://CRAN.R-project.org/package=ggplot2***ggplot2**

Description

This **ggplot2** theme is 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 extends ggplot2::theme_classic().

Value

A **ggplot2** theme object.

See Also

ggplot2::theme_classic() for the base theme extended here.

Plotting functions: 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 = "Banco de España"
  ) +
  theme_tidybde()
}
```

Index

∗ **bde_plot**
  bde_tidy_palettes, 13
  scales_bde, 15
  theme_tidybde, 16
∗ **catalog**
  bde_catalogs, 2
∗ **datasets**
  bde_ind_db, 6
∗ **indicators**
  bde_ind_db, 6
  bde_indicators, 4
∗ **series**
  bde_catalogs, 2
  bde_series, 8
  bde_series_api, 11
∗ **utils**
  bde_parse_dates, 7

API series functions, *6*
as.Date(), *7, 8*

base::as.double(), *9*
base::grep(), *3*
bde_catalog_load *(*bde_catalogs*)*, 2
bde_catalog_load(), *3*, *6*, *8–10*, *13*
bde_catalog_search *(*bde_catalogs*)*, 2
bde_catalog_search(), *6*, *10*, *13*
bde_catalog_update *(*bde_catalogs*)*, 2
bde_catalog_update(), *3*
bde_catalogs, 2, *10*, *13*
bde_ind_cpi_var *(*bde_indicators*)*, 4
bde_ind_db, *4*, *6*, 6
bde_ind_euribor_12m_daily *(*bde_indicators*)*, 4
bde_ind_euribor_12m_monthly *(*bde_indicators*)*, 4
bde_ind_gdp_quarterly *(*bde_indicators*)*, 4
bde_ind_gdp_var *(*bde_indicators*)*, 4
bde_ind_ibex *(*bde_indicators*)*, 4
bde_ind_ibex_daily *(*bde_indicators*)*, 4
bde_ind_ibex_monthly *(*bde_indicators*)*, 4
bde_ind_population *(*bde_indicators*)*, 4
bde_ind_unemployment_rate *(*bde_indicators*)*, 4
bde_indicators, 4, *7*
bde_parse_dates, 7
bde_parse_dates(), *3*, *5*, *9*
bde_series, *4*, *5*, 8, *13*
bde_series_api, *4*, *10*, 11
bde_series_api_latest *(*bde_series_api*)*, 11
bde_series_api_latest(), *10–13*
bde_series_api_load *(*bde_series_api*)*, 11
bde_series_api_load(), *10*, *11*, *13*
bde_series_full_load *(*bde_series*)*, 8
bde_series_load *(*bde_series*)*, 8
bde_series_load(), *5*, *6*, *8*, *10*, *12*, *13*
bde_tidy_palettes, 13
bde_tidy_palettes(), *15–17*

Date, *7*, *12*

ggplot2::alpha(), *14*, *15*
ggplot2::continuous_scale(), *16*
ggplot2::discrete_scale(), *16*
ggplot2::ggplot(), *9*
ggplot2::theme_classic, *17*
ggplot2::theme_classic(), *17*
guides(), *16*

Indicator wrappers, *10*
indicator wrappers, *6*

regular expressions, *3*

scale_color_bde_c *(*scales_bde*)*, 15
scale_color_bde_d *(*scales_bde*)*, 15
scale_colour_bde_c *(*scales_bde*)*, 15
scale_colour_bde_d *(*scales_bde*)*, 15
scale_fill_bde_c *(*scales_bde*)*, 15

scale_fill_bde_d *(*scales_bde*)*, 15
scales_bde, *14*, 15, *17*

theme_tidybde, 16
theme_tidybde(), *14*, *16*
tibble, *4–6*, *9*, *10*, *12*
tidyr::pivot_longer(), *9*
tidyr::pivot_wider(), *9*