Package 'fred'

November 1, 2026

Title Access 'Federal Reserve Economic Data'

Version 0.3.2

Description Provides clean, tidy access to economic data from the 'Federal Reserve Economic Data' ('FRED') API <https://fred.stlouisfed.org/docs/api/fred/>. 'FRED' is maintained by the 'Federal Reserve Bank of St. Louis' and contains over 800,000 time series from 118 sources covering GDP, employment, inflation, interest rates, trade, and more. Dedicated functions fetch series observations, search for series, browse categories, releases, and tags, and retrieve series metadata. Multiple series can be fetched in a single call, in long or wide format. Server-side unit transformations (percent change, log, etc.) and frequency aggregation are supported, with readable transform aliases such as 'yoy_pct' and 'log_diff'. Real-time and vintage helpers (built on 'ALFRED') return a series as it appeared on a given date, the first-release version, every revision, or a panel of selected vintages. An offline curated catalogue of around fifty popular series, NBER recession reference dates, and FOMC meeting dates support discoverability and event-study workflows. Default 'plot' method shades NBER recession periods. Reproducibility helpers produce BibTeX or plain-text citations and YAML manifests with per-object hashes. Data is cached locally for subsequent calls. This product uses the 'FRED' API but is not endorsed or certified by the 'Federal Reserve Bank of St. Louis'.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

URL https://charlescoverdale.github.io/fred/, https://github.com/charlescoverdale/fred

BugReports https://github.com/charlescoverdale/fred/issues

Depends R (>= 4.1.0)

Imports cli (>= 3.6.0), graphics, grDevices, httr2 (>= 1.0.0), stats, tools, utils

Suggests knitr, nowcast, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Charles Coverdale [aut, cre]

Maintainer Charles Coverdale <charlesfcoverdale@gmail.com>

Repository CRAN

Date/Publication 2026-11-01 15:20:14 UTC

Contents

clear_cache . . . 3
fred_aggregate . . . 3
fred_all_vintages . . . 5
fred_as_of . . . 6
fred_browse . . . 7
fred_cache_info . . . 8
fred_catalogue . . . 9
fred_category . . . 10
fred_category_children . . . 11
fred_category_series . . . 11
fred_cite_series . . . 12
fred_event_window . . . 13
fred_first_release . . . 14
fred_fomc_dates . . . 15
fred_get_key . . . 16
fred_info . . . 17
fred_interpolate . . . 18
fred_manifest . . . 19
fred_real_time_panel . . . 20
fred_recession_dates . . . 21
fred_related_tags . . . 22
fred_releases . . . 23
fred_release_dates . . . 23
fred_release_series . . . 24
fred_request . . . 25
fred_search . . . 25
fred_series . . . 27
fred_set_key . . . 29
fred_sources . . . 30
fred_source_releases . . . 30
fred_tags . . . 31
fred_updates . . . 32
fred_vintages . . . 33
fred_vintage_revisions . . . 33
plot.fred_tbl . . . 34
print.fred_manifest . . . 36

print.fred_tbl 36
summary.fred_tbl 37
[.fred_tbl 37

Index **38**

`clear_cache` *Clear the fred cache*

Description

Deletes all locally cached FRED data files. The next call to any data function will re-download from the FRED API.

Usage

```
clear_cache()
```

Value

Invisible `NULL`.

See Also

Other configuration: `fred_cache_info()`, `fred_get_key()`, `fred_request()`, `fred_set_key()`

Examples

```
op <- options(fred.cache_dir = tempdir())
clear_cache()
options(op)
```

`fred_aggregate` *Aggregate FRED observations to a coarser frequency*

Description

Aggregates a long-format `fred_tbl` (with `date`, `series_id`, `value`) or a wide-format `fred_tbl` (date plus one column per series) to a coarser calendar frequency. For long format, aggregation is performed per `series_id`; for wide format, per numeric column.

Usage

```
fred_aggregate(data, fun = "mean", by = "month")
```

Arguments

data	A fred_tbl or data.frame with a date column.
fun	Character. Aggregation function. One of "mean", "sum", "first", "last", "median", "min", "max". Default "mean".
by	Character. Target frequency. One of "week", "month", "quarter", "year". Default "month".

Details

Use this when you have, say, daily Treasury yields and need a monthly average, or weekly initial claims aggregated to monthly totals. For server-side aggregation that mirrors FRED's own interpolation conventions, pass frequency = "m" to fred_series() instead.

Value

A fred_tbl with the same columns as the input, with date collapsed to period start.

See Also

Other utilities: fred_event_window(), fred_interpolate()

Examples

```
# Synthetic example: aggregate daily synthetic data to monthly means
d <- seq(as.Date("2024-01-01"), as.Date("2024-06-30"), by = "day")
daily <- data.frame(date = d, series_id = "X", value = rnorm(length(d)))
fred_aggregate(daily, fun = "mean", by = "month")

# Wide-format input also works
wide <- data.frame(date = d, A = rnorm(length(d)), B = rnorm(length(d)))
fred_aggregate(wide, fun = "sum", by = "quarter")


op <- options(fred.cache_dir = tempdir())
try({
  ## Not run:
  daily_yields <- fred_series("DGS10", from = "2023-01-01")
  monthly_yields <- fred_aggregate(daily_yields, fun = "mean", by = "month")

## End(Not run)
})
options(op)
```

`fred_all_vintages` *Fetch every vintage of a series*

Description

Returns the full revision history: one row per (observation date, realtime range) combination. This is the FRED API's `output_type = 2` mode. The result can be reshaped into a vintage matrix or used to compute revision statistics.

Usage

```
fred_all_vintages(
  series_id,
  from = NULL,
  to = NULL,
  units = "lin",
  frequency = NULL,
  aggregation = "avg",
  cache = TRUE
)
```

Arguments

`series_id`	Character. One or more FRED series IDs.
`from, to`	Optional observation date range.
`units`	Character. Raw FRED units code. Default `"lin"`.
`frequency, aggregation`	Optional frequency aggregation arguments (see `fred_series()`).
`cache`	Logical. Cache results locally for 24 hours (see the `cache` argument of `fred_series()`). Default `TRUE`.

Details

Be aware that some series have hundreds of thousands of vintage rows, so consider narrowing the date range with `from`/`to` for long-running indicators like GDP.

Value

A `fred_tbl` with columns `date`, `series_id`, `value`, `realtime_start`, `realtime_end`. The `realtime_start` and `realtime_end` columns identify the vintage window for each row.

See Also

Other vintages: `fred_as_of()`, `fred_first_release()`, `fred_real_time_panel()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # All vintages of recent GDP releases
  gdp_vint <- fred_all_vintages("GDP", from = "2020-01-01")
})
options(op)
```

`fred_as_of` *Fetch a series as it appeared on a given vintage date*

Description

Returns the values that were available in FRED on `date`, before any subsequent revisions. This is the standard real-time data access pattern: set `realtime_start` = `realtime_end` = `date`. Useful for backtesting forecasting models against the data that was actually available at the time, not the revised series we see today.

Usage

```
fred_as_of(
  series_id,
  date,
  from = NULL,
  to = NULL,
  units = "lin",
  frequency = NULL,
  aggregation = "avg",
  cache = TRUE
)
```

Arguments

`series_id`	Character. One or more FRED series IDs.
`date`	Character or Date. The vintage date (`"YYYY-MM-DD"`).
`from`, `to`	Optional observation date range.
`units`	Character. Raw FRED units code. Default `"lin"`.
`frequency`, `aggregation`	Optional frequency aggregation arguments (see `fred_series()`).
`cache`	Logical. Cache results locally for 24 hours (see the `cache` argument of `fred_series()`). Default `TRUE`.

Details

Underneath, this calls the `series/observations` endpoint with the realtime parameters set. Results are cached separately from the default (latest-vintage) cache, so calling `fred_series("GDP")` and `fred_as_of("GDP", "2020-01-15")` keep distinct cache entries.

Value

A `fred_tbl` with columns `date`, `series_id`, `value`, `realtime_start`, `realtime_end`.

See Also

Other vintages: `fred_all_vintages()`, `fred_first_release()`, `fred_real_time_panel()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # GDP as it looked on 1 March 2020
  gdp_2020 <- fred_as_of("GDP", "2020-03-01")
})
options(op)
```

`fred_browse` *Browse the FRED category tree*

Description

Pretty-prints the FRED category tree. With no arguments, shows the eight top-level FRED categories from a built-in static reference (no API call). Pass a `category_id` to drill into its children, which fetches from the API (and is cached). Use this to discover where series live before searching inside a category with `fred_category_series()`.

Usage

```
fred_browse(category_id = 0L, depth = 1L)
```

Arguments

`category_id`	Integer. Category to browse. Default `0` (root). The root prints from a static reference and does not call the API.
`depth`	Integer. How many levels deep to recurse. Default `1` (only immediate children). Higher values trigger one API call per parent category visited.

Value

A `fred_tbl` of categories at the requested level (invisibly).

See Also

Other catalogue: `fred_catalogue()`

Examples

```
# Top-level categories (no API call)
fred_browse()

op <- options(fred.cache_dir = tempdir())
try({
  # Drill into "National Accounts" (id 32992)
  fred_browse(32992)
})
options(op)
```

`fred_cache_info` *Inspect the local fred cache*

Description

Returns information about the local cache: where it lives, how many files it contains, and how much disk space they take. Useful when debugging stale results or deciding whether to call `clear_cache()`.

Usage

```
fred_cache_info()
```

Value

A list with elements `dir`, `n_files`, `size_bytes`, `size_human`, and `files` (a data frame with `name`, `size_bytes`, and `modified` columns). Returns the same shape with zero counts if the cache directory does not yet exist.

See Also

Other configuration: `clear_cache()`, `fred_get_key()`, `fred_request()`, `fred_set_key()`

Examples

```
op <- options(fred.cache_dir = tempdir())
fred_cache_info()
options(op)
```

`fred_catalogue` *Browse a curated catalogue of popular FRED series*

Description

Returns a data frame of around 50 of the most widely used FRED series organised by economic category. The catalogue is curated and embedded in the package: no API call is made. Use it as a starting point for common workflows or as a quick reference when you cannot remember a series ID.

Usage

```
fred_catalogue(category = NULL, query = NULL)
```

Arguments

`category`	Character. Optional category filter. One of `"GDP & Output"`, `"Employment"`, `"Inflation"`, `"Interest Rates"`, `"Housing"`, `"Financial"`, `"Money & Credit"`, `"Trade & FX"`, `"Consumer"`, `"Government & Fiscal"`. Multiple categories may be passed.
`query`	Character. Optional free-text filter (e.g. `"unemployment"`, `"mortgage"`).

Details

Filter by category, by free-text query, or both. Free-text matches against the series ID, title, and description (case-insensitive substring match).

Value

A `fred_tbl` (a `data.frame` subclass) with columns `id`, `title`, `frequency`, `units`, `category`, `description`.

See Also

Other catalogue: `fred_browse()`

Examples

```
# Full catalogue
fred_catalogue()

# Inflation series only
fred_catalogue(category = "Inflation")

# Free-text search
fred_catalogue(query = "mortgage")

# Combined
fred_catalogue(category = "Interest Rates", query = "treasury")
```

`fred_category` *Get a FRED category*

Description

Returns information about a single category. The FRED category tree starts at category 0 (the root) and branches into 8 top-level categories: Money, Banking & Finance; Population, Employment & Labor Markets; National Accounts; Production & Business Activity; Prices; International Data; U.S. Regional Data; and Academic Data.

Usage

```
fred_category(category_id = 0L)
```

Arguments

`category_id`	Integer. The category ID. Default `0` (root).

Value

A data frame with category metadata.

See Also

Other categories: `fred_category_children()`, `fred_category_series()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # Root category
  fred_category()

  # National Accounts (category 32992)
  fred_category(32992)
})
options(op)
```

fred_category_children *List child categories*

Description

Returns the child categories for a given parent category.

Usage

```
fred_category_children(category_id = 0L)
```

Arguments

category_id	Integer. The parent category ID. Default 0 (root).

Value

A data frame of child categories.

See Also

Other categories: fred_category(), fred_category_series()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # Top-level categories
  fred_category_children()
})
options(op)
```

fred_category_series *List series in a category*

Description

Returns all series belonging to a given category. Automatically paginates through all results.

Usage

```
fred_category_series(category_id, limit = 1000L)
```

Arguments

category_id	Integer. The category ID.
limit	Integer. Maximum number of results to return. Default 1000.

Value

A data frame of series metadata.

See Also

Other categories: fred_category(), fred_category_children()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_category_series(32992)
})
options(op)
```

---

fred_cite_series *Generate a citation for a FRED series*

Description

Produces a citation string for a single FRED series in BibTeX, plain text, or bibentry form. Works offline by default: the citation falls back to the series ID as the title. Pass fetch_metadata = TRUE to call fred_info() and use the official series title (requires an API key).

Usage

```
fred_cite_series(
  series_id,
  vintage_date = NULL,
  format = c("bibtex", "text", "bibentry"),
  fetch_metadata = FALSE
)
```

Arguments

series_id	Character. A single FRED series ID.
vintage_date	Optional character or Date. Vintage date the data was accessed as of. Defaults to today's date (the "accessed" date).
format	Character. One of "bibtex" (default), "text", or "bibentry" (an R utils::bibentry object).
fetch_metadata	Logical. If TRUE, call fred_info() to fetch the official series title. Requires an API key. Default FALSE.

Details

Cite a specific vintage by passing `vintage_date`. This is essential for reproducible research: a 2023 GDP figure published in October 2023 may differ materially from the same observation as it appears today.

Value

A character string (`"bibtex"`, `"text"`) or a `bibentry` object.

See Also

Other reproducibility: `fred_manifest()`, `fred_vintage_revisions()`

Examples

```
# BibTeX without an API call
fred_cite_series("GDPC1")

# Plain-text citation pinned to a vintage date
fred_cite_series("UNRATE", vintage_date = "2024-06-01", format = "text")

## Not run:
  # Use the official title (requires an API key)
  fred_cite_series("GDPC1", fetch_metadata = TRUE)

## End(Not run)
```

`fred_event_window` *Extract data windows around event dates*

Description

Given a `fred_tbl` (or any data frame with a `date` column) and a vector of event dates, returns one row per (event, observation) pair where the observation falls inside the requested window. Useful for event studies around FOMC decisions, recession peaks, or release dates.

Usage

```
fred_event_window(data, events, window = c(-30L, 90L))
```

Arguments

`data`	A `fred_tbl` or `data.frame` with a `date` column.
`events`	A character or Date vector of event dates.
`window`	Integer length-2. Days before (negative or zero) and after (positive or zero) each event date. Default `c(-30L, 90L)`.

Details

Long-format input with `series_id`/`value` is supported, as is wide-format input from `fred_series(..., format = "wide")`. The window is in calendar days; for monthly or quarterly data, choose a window large enough to capture at least one observation.

Value

A `fred_tbl` with the original columns plus `event_date` and `days_from_event`.

See Also

Other utilities: `fred_aggregate()`, `fred_interpolate()`

Examples

```
# Synthetic example — works offline
d <- seq(as.Date("2024-01-01"), as.Date("2024-12-31"), by = "month")
df <- data.frame(date = d, value = seq_along(d))
events <- as.Date(c("2024-03-15", "2024-09-15"))
fred_event_window(df, events = events, window = c(-30L, 60L))


op <- options(fred.cache_dir = tempdir())
try({
  # With live FRED data: UNRATE around 2024 SEP meetings (needs API key)
  ## Not run:
  ur <- fred_series("UNRATE", from = "2023-01-01")
  sep <- fred_fomc_dates(year = 2024, sep_only = TRUE)
  fred_event_window(ur, events = sep$date, window = c(-60L, 60L))

## End(Not run)
})
options(op)
```

`fred_first_release` *Fetch the first-release ("real-time") version of a series*

Description

Returns only the value that was published when each observation first appeared in FRED, with no subsequent revisions. Internally this fetches the full revision history and keeps the earliest `realtime_start` row for each observation date. Useful when you want a clean comparison between what policymakers saw at the time versus what the data look like after revisions.

Usage

```
fred_first_release(
  series_id,
  from = NULL,
  to = NULL,
  units = "lin",
  frequency = NULL,
  aggregation = "avg",
  cache = TRUE
)
```

Arguments

series_id	Character. One or more FRED series IDs.
from, to	Optional observation date range.
units	Character. Raw FRED units code. Default "lin".
frequency, aggregation	Optional frequency aggregation arguments (see fred_series()).
cache	Logical. Cache results locally for 24 hours (see the cache argument of fred_series()). Default TRUE.

Value

A fred_tbl with columns date, series_id, value, realtime_start, realtime_end.

See Also

Other vintages: fred_all_vintages(), fred_as_of(), fred_real_time_panel()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # Initial-release GDP, never revised
  gdp_first <- fred_first_release("GDP", from = "2018-01-01")
})
options(op)
```

---

fred_fomc_dates *FOMC scheduled meeting dates and decisions*

Description

Returns scheduled regular FOMC meeting decision dates from 2017 to 2026, plus selected unscheduled meetings during stress periods (e.g. March 2020). Each row carries the decision date, meeting type, and a flag for whether a Summary of Economic Projections (SEP) was released. Useful as a left table for event-study windows around monetary-policy decisions.

Usage

```
fred_fomc_dates(from = NULL, to = NULL, year = NULL, sep_only = FALSE)
```

Arguments

from	Optional character or Date. Filter to meetings on or after this date.
to	Optional character or Date. Filter to meetings on or before this date.
year	Optional integer vector. Filter to specific calendar years.
sep_only	Logical. If TRUE, return only meetings that released an SEP (typically four per year). Default FALSE.

Details

Source: Federal Reserve Board press release calendars (https://www.federalreserve.gov/monetarypolicy/fomccalendars.htm). Curated and embedded; not auto-synced.

Value

A fred_tbl with columns date, type, sep.

See Also

Other dates: fred_recession_dates()

Examples

```
# All meetings since 2017
fred_fomc_dates()

# 2022 hiking cycle
fred_fomc_dates(year = 2022)

# SEP meetings only (Mar/Jun/Sep/Dec)
fred_fomc_dates(year = 2024, sep_only = TRUE)
```

---

fred_get_key *Get the current FRED API key*

Description

Returns the API key set via fred_set_key() or the FRED_API_KEY environment variable. Raises an error if no key is found.

Usage

```
fred_get_key()
```

Value

Character. The API key.

See Also

Other configuration: clear_cache(), fred_cache_info(), fred_request(), fred_set_key()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_get_key()
})
options(op)
```

---

fred_info *Get metadata for a FRED series*

Description

Returns descriptive information about a series, including its title, units, frequency, seasonal adjustment, and notes.

Usage

```
fred_info(series_id)
```

Arguments

series_id	Character. A single FRED series ID.

Value

A data frame with one row containing series metadata.

See Also

Other series: fred_search(), fred_series(), fred_updates(), fred_vintages()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_info("GDP")
})
options(op)
```

`fred_interpolate` *Fill missing values in a FRED series*

Description

Fills `NA` values in the `value` column (long format) or in numeric columns (wide format). Two methods are supported: last-observation-carry-forward (`"locf"`) and linear interpolation between adjacent observed values (`"linear"`). Use this for mixed-frequency analysis where a low-frequency series needs to be interpolated to a higher frequency.

Usage

```
fred_interpolate(data, method = c("locf", "linear"))
```

Arguments

`data`	A `fred_tbl` or `data.frame` with a `date` column.
`method`	Character. `"locf"` (default) carries the last observed value forward. `"linear"` does linear interpolation between adjacent non-`NA` values.

Details

Boundary behaviour: with `method = "locf"`, leading `NA`s remain `NA` because there is no prior observation to carry forward. With `method = "linear"`, neither leading nor trailing `NA`s are filled because `stats::approx()` is called with `rule = 1` (no extrapolation). If you need extrapolation, post-process the result.

Value

A `fred_tbl` with interior `NA`s filled (see boundary note above).

See Also

Other utilities: `fred_aggregate()`, `fred_event_window()`

Examples

```
# Synthetic example: fill interior NAs
d <- seq(as.Date("2024-01-01"), by = "month", length.out = 6L)
df <- data.frame(date = d, series_id = "X",
                 value = c(NA, 2, NA, NA, 5, NA))
fred_interpolate(df, method = "locf")
fred_interpolate(df, method = "linear")


op <- options(fred.cache_dir = tempdir())
try({
  ## Not run:
```

```
  gdp <- fred_series("GDPC1", from = "2020-01-01")
  gdp_monthly <- fred_interpolate(gdp, method = "linear")

## End(Not run)
})
options(op)
```

`fred_manifest` *Snapshot a session's FRED downloads as a YAML manifest*

Description

Produces a YAML manifest describing one or more `fred_tbl` objects, with query metadata, dimensions, date ranges, and an MD5 hash of each object. Intended to be saved alongside paper code for reproducibility checks: if the manifest still hashes to the same values, the data underlying the analysis has not changed.

Usage

```
fred_manifest(..., file = NULL)
```

Arguments

`...`	`fred_tbl` objects (positional or named) or a single list of `fred_tbl` objects.
`file`	Optional path to write the YAML manifest. If `NULL` (default), the manifest is returned as a `fred_manifest` object (printable).

Details

Pass `fred_tbl` objects positionally, named, or as a list. If passed a bare list, names from the list are used; otherwise objects are labelled `obj_1`, `obj_2`, ...

Value

A `fred_manifest` object (a character string with an attached print method). If `file` is supplied, written to disk and returned invisibly.

See Also

Other reproducibility: `fred_cite_series()`, `fred_vintage_revisions()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  ## Not run:
  gdp <- fred_series("GDPC1", from = "2020-01-01")
  un  <- fred_series("UNRATE", from = "2020-01-01")
  m <- fred_manifest(gdp = gdp, unrate = un)
  print(m)
  fred_manifest(gdp = gdp, file = file.path(tempdir(), "manifest.yml"))

## End(Not run)
})
options(op)
```

`fred_real_time_panel` *Fetch a real-time panel of a series across selected vintages*

Description

Returns the values that were available on each of a chosen set of vintage dates. This is the FRED API's `vintage_dates` parameter: instead of asking for every revision (potentially huge), you ask for only the snapshots you care about, e.g. quarterly vintages aligned to GDP release dates.

Usage

```
fred_real_time_panel(
  series_id,
  vintages,
  from = NULL,
  to = NULL,
  units = "lin",
  frequency = NULL,
  aggregation = "avg",
  cache = TRUE
)
```

Arguments

`series_id`	Character. One or more FRED series IDs.
`vintages`	Character or Date vector. Vintage dates to fetch.
`from, to`	Optional observation date range.
`units`	Character. Raw FRED units code. Default `"lin"`.
`frequency, aggregation`	Optional frequency aggregation arguments (see `fred_series()`).
`cache`	Logical. Cache results locally for 24 hours (see the `cache` argument of `fred_series()`). Default `TRUE`.

Value

A fred_tbl with columns date, series_id, value, realtime_start, realtime_end.

See Also

Other vintages: fred_all_vintages(), fred_as_of(), fred_first_release()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # GDP as published at three quarterly snapshots
  gdp_panel <- fred_real_time_panel(
  "GDP",
  vintages = c("2023-04-30", "2023-07-31", "2023-10-31")
  )
})
options(op)
```

---

fred_recession_dates *NBER US business-cycle reference dates*

Description

Returns the NBER Business Cycle Dating Committee's reference dates for US business-cycle peaks and troughs since 1857. With flag = NULL (default), returns one row per recession (peak, trough, duration). Pass a vector of observation dates as flag to get back a data frame indicating whether each observation falls within a recession.

Usage

```
fred_recession_dates(from = NULL, to = NULL, flag = NULL)
```

Arguments

from	Optional character or Date. Filter to recessions whose peak is on or after this date.
to	Optional character or Date. Filter to recessions whose trough is on or before this date.
flag	Optional Date vector. If supplied, returns a data frame data.frame(date, in_recession) indicating whether each date falls within an NBER recession. Useful as a regression covariate or for plotting.

Details

Source: NBER Business Cycle Dating Committee (https://www.nber.org/research/data/us-business-cycle-expan
Dates are ISO month-start (peak = first month of recession, trough = last month of recession).

Value

A `fred_tbl`. Default columns: `peak`, `trough`, `duration_months`. With `flag`: `date`, `in_recession`.

See Also

Other dates: `fred_fomc_dates()`

Examples

```
# All recessions since 1857
fred_recession_dates()

# Modern era only
fred_recession_dates(from = "1948-01-01")

# Flag a vector of dates
fred_recession_dates(flag = seq(as.Date("2007-01-01"), as.Date("2010-12-01"),
                                by = "month"))
```

---

fred_related_tags *Find tags related to a given tag*

Description

Returns tags that are frequently used together with the specified tag.

Usage

```
fred_related_tags(tag_names)
```

Arguments

`tag_names`	Character. One or more tag names, separated by semicolons (e.g. `"gdp"`, `"usa;quarterly"`).

Value

A data frame of related tags.

See Also

Other tags: `fred_tags()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_related_tags("gdp")
})
options(op)
```

fred_releases *List all FRED releases*

Description

Returns all data releases available on FRED. A release is a collection of related series published together (e.g. "Employment Situation", "GDP").

Usage

```
fred_releases()
```

Value

A data frame of releases with columns including `id`, `name`, `press_release`, and `link`.

See Also

Other releases: `fred_release_dates()`, `fred_release_series()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_releases()
})
options(op)
```

fred_release_dates *Get release dates*

Description

Returns the dates on which data for a release were published. Useful for understanding the data calendar and when revisions occurred.

Usage

```
fred_release_dates(release_id)
```

Arguments

`release_id`	Integer. The release ID.

Value

A data frame with columns `release_id` and `date`.

See Also

Other releases: fred_release_series(), fred_releases()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_release_dates(53)
})
options(op)
```

---

fred_release_series *List series in a release*

Description

Returns all series belonging to a given release.

Usage

```
fred_release_series(release_id)
```

Arguments

release_id	Integer. The release ID.

Value

A data frame of series metadata.

See Also

Other releases: fred_release_dates(), fred_releases()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # G.19 Consumer Credit release
  fred_release_series(14)
})
options(op)
```

fred_request *Make a raw request to the FRED API*

Description

Low-level function that sends a request to any FRED API endpoint and returns the parsed JSON as a list. Most users should use the higher-level functions such as `fred_series()` or `fred_search()`.

Usage

```
fred_request(endpoint, ...)
```

Arguments

`endpoint`	Character. The API endpoint path (e.g. `"series/observations"`).
`...`	Named parameters passed as query string arguments to the API.

Value

A list parsed from the JSON response.

See Also

Other configuration: `clear_cache()`, `fred_cache_info()`, `fred_get_key()`, `fred_set_key()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_request("series", series_id = "GDP")
})
options(op)
```

fred_search *Search for FRED series*

Description

Searches the FRED database by keywords or series ID substring. Returns matching series with their metadata, ordered by relevance.

Usage

```
fred_search(
  query,
  type = "full_text",
  limit = 100L,
  order_by = "search_rank",
  filter_variable = NULL,
  filter_value = NULL,
  tag_names = NULL
)
```

Arguments

query	Character. Search terms (e.g. "GDP", "unemployment rate").
type	Character. Either "full_text" (default) for keyword search or "series_id" for series ID substring matching (supports * wildcards).
limit	Integer. Maximum number of results to return. Default 100, maximum 1000.
order_by	Character. How to order results. One of "search_rank" (default), "series_id", "title", "units", "frequency", "seasonal_adjustment", "last_updated", "popularity", "group_popularity".
filter_variable	Character. Optional variable to filter by. One of "frequency", "units", or "seasonal_adjustment".
filter_value	Character. The value to filter on (e.g. "Monthly", "Quarterly"). Required if filter_variable is specified.
tag_names	Character. Optional comma-separated tag names to filter results (e.g. "gdp", "usa;quarterly").

Value

A data frame of matching series with columns including id, title, frequency, units, seasonal_adjustment, last_updated, popularity, and notes.

See Also

Other series: fred_info(), fred_series(), fred_updates(), fred_vintages()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # Keyword search
  fred_search("unemployment rate")

  # Filter to monthly series only
  fred_search("consumer price index", filter_variable = "frequency",
  filter_value = "Monthly")
```

```
  # Search by series ID pattern
  fred_search("GDP*", type = "series_id")
})
options(op)
```

`fred_series` *Fetch observations for one or more FRED series*

Description

The main function in the package. Downloads time series observations from FRED and returns a tidy data frame. Multiple series can be fetched in a single call, in either long or wide format.

Usage

```
fred_series(
  series_id,
  from = NULL,
  to = NULL,
  units = "lin",
  transform = NULL,
  frequency = NULL,
  aggregation = "avg",
  format = c("long", "wide"),
  cache = TRUE
)
```

Arguments

`series_id`	Character. One or more FRED series IDs (e.g. `"GDP"`, `c("GDP", "UNRATE", "CPIAUCSL")`).
`from`	Optional start date. Character (`"YYYY-MM-DD"`) or Date.
`to`	Optional end date. Character (`"YYYY-MM-DD"`) or Date.
`units`	Character. Raw FRED units code. Default `"lin"` (levels). Mutually exclusive with `transform`.
`transform`	Character. Readable transformation name. See Details.
`frequency`	Character. Frequency aggregation. One of `"d"` (daily), `"w"` (weekly), `"bw"` (biweekly), `"m"` (monthly), `"q"` (quarterly), `"sa"` (semiannual), `"a"` (annual), or `NULL` (native frequency, the default).
`aggregation`	Character. Aggregation method when `frequency` is specified. One of `"avg"` (default), `"sum"`, or `"eop"` (end of period).
`format`	Character. `"long"` (default) returns one row per `(series_id, date)`. `"wide"` returns one row per date with one column per series.

cache — Logical. If TRUE (the default), results are cached locally and returned from the cache on subsequent calls for 24 hours, after which they are downloaded again (set options(fred.cache_ttl =) to a number of hours to change this). Set to FALSE to force a fresh download from the API.

Details

FRED supports server-side unit transformations via the units argument. This avoids the need to compute growth rates or log transforms locally. For readability you can pass transform instead of units:

- "level", "raw" -levels (the default)
- "diff", "change" -change from previous period
- "yoy_diff" -change from one year ago
- "qoq_pct", "mom_pct", "pop_pct" -percent change from previous period
- "yoy_pct" -percent change from one year ago
- "annualised", "qoq_annualised" -compounded annual rate of change
- "log" -natural log
- "log_diff" -continuously compounded rate of change
- "log_diff_annualised" -continuously compounded annual rate

Raw FRED units codes ("lin", "chg", "ch1", "pch", "pc1", "pca", "cch", "cca", "log") are also accepted.

Value

A fred_tbl (a data.frame subclass that prints with a one-line provenance header). In long format, columns are date, series_id, value. In wide format, columns are date plus one numeric column per series.

See Also

Other series: fred_info(), fred_search(), fred_updates(), fred_vintages()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # Single series
  gdp <- fred_series("GDP")

  # Multiple series, long format
  macro <- fred_series(c("GDP", "UNRATE", "CPIAUCSL"))

  # Multiple series, wide format
  macro_w <- fred_series(c("GDP", "UNRATE"), format = "wide")

  # Readable transformation: year-on-year percent change
  gdp_growth <- fred_series("GDP", transform = "yoy_pct")
```

```
  # Aggregate daily to monthly
  rates <- fred_series("DGS10", frequency = "m")
})
options(op)
```

`fred_set_key` *Set the FRED API key*

Description

Sets the API key used to authenticate requests to the FRED API. The key persists for the current R session. Alternatively, set the `FRED_API_KEY` environment variable in your `.Renviron` file.

Usage

```
fred_set_key(key)
```

Arguments

`key`	Character. A 32-character FRED API key.

Details

Register for a free API key at https://fredaccount.stlouisfed.org/apikeys.

Value

Invisible `NULL`.

See Also

Other configuration: `clear_cache()`, `fred_cache_info()`, `fred_get_key()`, `fred_request()`

Examples

```
## Not run:
fred_set_key("your_api_key_here")

## End(Not run)
```

fred_sources *List all FRED data sources*

Description

Returns all data sources that contribute series to FRED. Sources include the Bureau of Labor Statistics, Bureau of Economic Analysis, Federal Reserve Board, U.S. Census Bureau, and over 100 others.

Usage

```
fred_sources()
```

Value

A data frame of sources with columns including `id`, `name`, and `link`.

See Also

Other sources: `fred_source_releases()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_sources()
})
options(op)
```

fred_source_releases *List releases from a source*

Description

Returns all releases published by a given data source.

Usage

```
fred_source_releases(source_id)
```

Arguments

`source_id`	Integer. The source ID.

Value

A data frame of releases.

See Also

Other sources: `fred_sources()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  # Bureau of Labor Statistics
  fred_source_releases(22)
})
options(op)
```

`fred_tags` *List or search FRED tags*

Description

Returns FRED tags, optionally filtered by a search query. Tags are keywords used to categorise series (e.g. "gdp", "monthly", "usa", "seasonally adjusted").

Usage

```
fred_tags(query = NULL, limit = 1000L)
```

Arguments

`query`	Character. Optional search string to filter tags.
`limit`	Integer. Maximum number of results. Default 1000.

Value

A data frame of tags with columns including `name`, `group_id`, `notes`, `popularity`, and `series_count`.

See Also

Other tags: `fred_related_tags()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_tags()
  fred_tags("inflation")
})
options(op)
```

`fred_updates` *List recently updated FRED series*

Description

Returns series that have been recently updated or revised.

Usage

```
fred_updates(limit = 100L)
```

Arguments

`limit`	Integer. Maximum number of results. Default 100, maximum 100.

Value

A data frame of recently updated series.

See Also

Other series: `fred_info()`, `fred_search()`, `fred_series()`, `fred_vintages()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_updates()
})
options(op)
```

fred_vintages *Get vintage dates for a FRED series*

Description

Returns the dates on which data for a series were revised. This is useful for real-time analysis and understanding data revisions.

Usage

```
fred_vintages(series_id)
```

Arguments

series_id	Character. A single FRED series ID.

Value

A data frame with columns `series_id` and `vintage_date`.

See Also

Other series: `fred_info()`, `fred_search()`, `fred_series()`, `fred_updates()`

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  fred_vintages("GDP")
})
options(op)
```

fred_vintage_revisions *Summarise revision behaviour for a FRED series*

Description

For each observation date in a series' vintage history, computes summary statistics on how the value has been revised: number of vintages, first and final value, total revision (final minus first), mean and SD of inter-vintage changes, and elapsed days from first publication to final. Useful for choosing series for real-time analysis (low-revision series are more reliable for nowcasting).

Usage

```
fred_vintage_revisions(series_id, from = NULL, to = NULL, cache = TRUE)
```

Arguments

series_id	Character. A single FRED series ID.
from, to	Optional observation date range.
cache	Logical. Cache the underlying vintage download for 24 hours (see the cache argument of fred_series()). Default TRUE.

Details

Internally fetches fred_all_vintages(series_id, ...) and reduces. For long-running indicators, narrow the window with from/to to keep the API call manageable.

Value

A fred_tbl with columns series_id, date, n_vintages, first_value, final_value, revision_total, revision_total_pct, revision_mean, revision_sd, days_to_final.

See Also

Other reproducibility: fred_cite_series(), fred_manifest()

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  ## Not run:
  rev <- fred_vintage_revisions("GDPC1", from = "2018-01-01")
  summary(rev$revision_total_pct)

## End(Not run)
})
options(op)
```

---

plot.fred_tbl *Plot a fred_tbl*

Description

Time-series plot for a fred_tbl. Detects long or wide format, draws one line per series, and optionally shades NBER recession periods. Uses base graphics so no extra dependencies are pulled in.

Usage

```
## S3 method for class 'fred_tbl'
plot(
  x,
  recessions = TRUE,
  legend = TRUE,
  col = NULL,
  type = "l",
  main = NULL,
  xlab = "",
  ylab = "value",
  ...
)
```

Arguments

x	A fred_tbl.
recessions	Logical. If TRUE (default), shade NBER recession periods that overlap the plotted date range.
legend	Logical. If TRUE (default) and there are multiple series, add a top-left legend.
col	Character. Optional vector of colours, one per series. Default uses a fixed six-colour qualitative palette, or HCL colours for >6 series.
type	Character. Plot type, passed to graphics::lines(). Default "l" (line).
main, xlab, ylab	Plot labels. Sensible defaults are inferred.
...	Other arguments passed to the initial graphics::plot() call.

Details

For long-format input (date, series_id, value), one line per series_id. For wide-format input (date plus one numeric column per series), one line per numeric column.

Value

x, invisibly.

Examples

```
op <- options(fred.cache_dir = tempdir())
try({
  ## Not run:
  gdp <- fred_series("GDPC1", from = "2000-01-01")
  plot(gdp)

  panel <- fred_series(c("UNRATE", "CIVPART"), from = "2000-01-01",
  format = "wide")
  plot(panel)

## End(Not run)
```

```
})
options(op)
```

`print.fred_manifest` *Print method for fred_manifest*

Description

Print method for fred_manifest

Usage

```
## S3 method for class 'fred_manifest'
print(x, ...)
```

Arguments

`x`	A `fred_manifest` object.
`...`	Ignored.

Value

`x`, invisibly.

`print.fred_tbl` *Print method for fred_tbl*

Description

Adds a one-line provenance header above the data frame body. The header summarises the query: number of series, observation count, transformation in effect, vintage information, or for non-observation queries, the endpoint and result count.

Usage

```
## S3 method for class 'fred_tbl'
print(x, ...)
```

Arguments

`x`	A `fred_tbl`.
`...`	Passed to the underlying `print.data.frame` method.

Value

`x`, invisibly.

summary.fred_tbl *Summary method for fred_tbl*

Description

Prints query metadata, dimensions, date range (when present), and value range (when present), then dispatches to the standard `summary.data.frame`.

Usage

```
## S3 method for class 'fred_tbl'
summary(object, ...)
```

Arguments

`object`	A `fred_tbl`.
`...`	Passed to the underlying `summary.data.frame` method.

Value

Invisibly returns the standard data frame summary.

[.fred_tbl *Subset method for fred_tbl*

Description

Preserves the `fred_tbl` class and `fred_query` attribute when subsetting rows or columns. Falls back to a plain vector when `drop = TRUE` reduces the result to a single column (matching `data.frame` behaviour).

Usage

```
## S3 method for class 'fred_tbl'
x[i, j, ..., drop = TRUE]
```

Arguments

`x`	A `fred_tbl`.
`i`	Row selector.
`j`	Column selector.
`...`	Other arguments passed to `[.data.frame`.
`drop`	Logical. As in `[.data.frame`.

Value

A `fred_tbl` (or a vector if `drop` collapses the result).

Index

∗ **catalogue**
  fred_browse, 7
  fred_catalogue, 9
∗ **categories**
  fred_category, 10
  fred_category_children, 11
  fred_category_series, 11
∗ **configuration**
  clear_cache, 3
  fred_cache_info, 8
  fred_get_key, 16
  fred_request, 25
  fred_set_key, 29
∗ **dates**
  fred_fomc_dates, 15
  fred_recession_dates, 21
∗ **fred_tbl methods**
  plot.fred_tbl, 34
∗ **releases**
  fred_release_dates, 23
  fred_release_series, 24
  fred_releases, 23
∗ **reproducibility**
  fred_cite_series, 12
  fred_manifest, 19
  fred_vintage_revisions, 33
∗ **series**
  fred_info, 17
  fred_search, 25
  fred_series, 27
  fred_updates, 32
  fred_vintages, 33
∗ **sources**
  fred_source_releases, 30
  fred_sources, 30
∗ **tags**
  fred_related_tags, 22
  fred_tags, 31
∗ **utilities**
  fred_aggregate, 3
  fred_event_window, 13
  fred_interpolate, 18
∗ **vintages**
  fred_all_vintages, 5
  fred_as_of, 6
  fred_first_release, 14
  fred_real_time_panel, 20
[.fred_tbl, 37

clear_cache, 3
clear_cache(), *8*, *17*, *25*, *29*

fred_aggregate, 3
fred_aggregate(), *14*, *18*
fred_all_vintages, 5
fred_all_vintages(), *7*, *15*, *21*
fred_as_of, 6
fred_as_of(), *5*, *15*, *21*
fred_browse, 7
fred_browse(), *9*
fred_cache_info, 8
fred_cache_info(), *3*, *17*, *25*, *29*
fred_catalogue, 9
fred_catalogue(), *7*
fred_category, 10
fred_category(), *11, 12*
fred_category_children, 11
fred_category_children(), *10*, *12*
fred_category_series, 11
fred_category_series(), *7*, *10, 11*
fred_cite_series, 12
fred_cite_series(), *19*, *34*
fred_event_window, 13
fred_event_window(), *4*, *18*
fred_first_release, 14
fred_first_release(), *5*, *7*, *21*
fred_fomc_dates, 15
fred_fomc_dates(), *22*
fred_get_key, 16

fred_get_key(), *3*, *8*, *25*, *29*
fred_info, 17
fred_info(), *12*, *26*, *28*, *32, 33*
fred_interpolate, 18
fred_interpolate(), *4*, *14*
fred_manifest, 19
fred_manifest(), *13*, *34*
fred_real_time_panel, 20
fred_real_time_panel(), *5*, *7*, *15*
fred_recession_dates, 21
fred_recession_dates(), *16*
fred_related_tags, 22
fred_related_tags(), *31*
fred_release_dates, 23
fred_release_dates(), *23, 24*
fred_release_series, 24
fred_release_series(), *23, 24*
fred_releases, 23
fred_releases(), *24*
fred_request, 25
fred_request(), *3*, *8*, *17*, *29*
fred_search, 25
fred_search(), *17*, *25*, *28*, *32, 33*
fred_series, 27
fred_series(), *4–6*, *15*, *17*, *20*, *25, 26*, *32–34*
fred_set_key, 29
fred_set_key(), *3*, *8*, *16, 17*, *25*
fred_source_releases, 30
fred_source_releases(), *30*
fred_sources, 30
fred_sources(), *31*
fred_tags, 31
fred_tags(), *22*
fred_updates, 32
fred_updates(), *17*, *26*, *28*, *33*
fred_vintage_revisions, 33
fred_vintage_revisions(), *13*, *19*
fred_vintages, 33
fred_vintages(), *17*, *26*, *28*, *32*

graphics::lines(), *35*
graphics::plot(), *35*

plot.fred_tbl, 34
print.fred_manifest, 36
print.fred_tbl, 36

summary.fred_tbl, 37