Package: carbondata (via r-universe)

November 1, 2026

Title Access Carbon Market Data from Emissions Trading Systems and Voluntary Registries

Version 0.2.1

Description Unified access to carbon market data from compliance emissions trading systems ('EU ETS', 'UK ETS', 'RGGI', California Cap-and-Trade) and voluntary carbon markets (Verra, Gold Standard, American Carbon Registry, Climate Action Reserve, via the Berkeley Voluntary Registry Offsets Database and the 'CarbonPlan' 'OffsetsDB' API). Includes cross-market price data from the 'International Carbon Action Partnership' ('ICAP') Allowance Price Explorer <https://icapcarbonaction.com/en/ets-prices>, global carbon pricing from the World Bank Carbon Pricing Dashboard <https://carbonpricingdashboard.worldbank.org/>, and the historical 'RFF' World Carbon Pricing Database following Dolphin, Pollitt and Newbery (2020) <doi:10.1038/s41597-022-01659-x>. Data is downloaded from public sources on first use and cached locally.

Depends R (>= 4.1.0)

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Imports cli (>= 3.6.0), httr2 (>= 1.0.0), jsonlite, readxl, tools, utils

Suggests arrow, nanoparquet, testthat (>= 3.0.0), writexl

Config/testthat/edition 3

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

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

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Charles Coverdale [aut, cre]

Maintainer Charles Coverdale <charlesfcoverdale@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-01 19:47:40 UTC

RemoteUrl https://github.com/cran/carbondata

RemoteRef HEAD

RemoteSha 9d6c5c86a0701411acc666d2e18ae42ac50ef829

Contents

co2_cache_info . . . 2
co2_cad_trust . . . 3
co2_california_caps . . . 4
co2_california_prices . . . 5
co2_clear_cache . . . 5
co2_ecp_prices . . . 6
co2_euets_allocations . . . 7
co2_euets_emissions . . . 8
co2_euets_files . . . 10
co2_euets_installations . . . 10
co2_euets_price . . . 11
co2_euets_surrendered . . . 12
co2_icap_prices . . . 13
co2_icap_systems . . . 14
co2_markets . . . 15
co2_offsets_db . . . 15
co2_rff_pricing . . . 16
co2_rggi_allowances . . . 17
co2_rggi_state_proceeds . . . 18
co2_ukets . . . 19
co2_ukets_allocations . . . 20
co2_vrod . . . 21
co2_world_bank . . . 22

Index **23**

`co2_cache_info` *Inspect the local carbondata cache*

Description

Returns information about the local cache: directory, file count, size, and list of cached files. Useful for debugging stale results or deciding whether to call `co2_clear_cache()`.

Usage

```
co2_cache_info()
```

Value

A list with elements `dir`, `n_files`, `size_bytes`, `size_human`, and `files` (a data frame with `name`, `size_bytes`, `modified`).

See Also

Other configuration: `co2_clear_cache()`

Examples

```
op <- options(carbondata.cache_dir = tempdir())
co2_cache_info()
options(op)
```

---

co2_cad_trust *Climate Action Data Trust (not supported)*

Description

CAD Trust is the Chia-Network "cadt" software designed to be self-hosted. There is no unauthenticated public API. To use CAD Trust data you must either:

- Self-host a cadt node, or
- Arrange a private partnership for API access via https://climateactiondata.org/how-to-connect/

Usage

```
co2_cad_trust(...)
```

Arguments

`...`	Ignored.

Details

This function is a placeholder that errors with guidance. It will gain an implementation once CAD Trust offers a stable public endpoint.

Value

Never returns (always errors).

See Also

Other voluntary markets: co2_offsets_db(), co2_vrod()

Examples

```
## Not run:
co2_cad_trust()

## End(Not run)
```

`co2_california_caps` *California overall emissions caps*

Description

Fetches the California Cap-and-Trade program overall cap and allocation channels, by vintage year.

Usage

```
co2_california_caps(refresh = FALSE)
```

Arguments

`refresh`	Re-download? Default `FALSE`.

Value

A data frame.

See Also

Other compliance markets: co2_california_prices(), co2_rggi_allowances(), co2_rggi_state_proceeds(), co2_ukets(), co2_ukets_allocations()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  caps <- co2_california_caps()
})
options(op)
```

co2_california_prices *California Cap-and-Trade auction settlement prices*

Description

Fetches the California Air Resources Board auction settlement price time series, updated quarterly after each joint California- Quebec auction.

Usage

```
co2_california_prices(refresh = FALSE)
```

Arguments

refresh	Re-download? Default FALSE.

Value

A data frame with joint_auction, quarter, settlement_price_usd, reserve_price_usd.

See Also

Other compliance markets: co2_california_caps(), co2_rggi_allowances(), co2_rggi_state_proceeds(), co2_ukets(), co2_ukets_allocations()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  prices <- co2_california_prices()
})
options(op)
```

co2_clear_cache *Clear the carbondata cache*

Description

Deletes all locally cached carbon market data files. The next call to any data function will re-download.

Usage

```
co2_clear_cache()
```

Value

Invisibly returns NULL.

See Also

Other configuration: co2_cache_info()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
co2_clear_cache()
options(op)
```

---

co2_ecp_prices *Emissions-weighted carbon prices (ECP)*

Description

Fetches the emissions-weighted carbon price dataset published alongside the World Carbon Pricing Database at https://worldcarbonpricing.org. Unlike the raw WCPD files, these endpoints remain open and are updated with each release, so this is the current route to World Carbon Pricing Database derived data.

Usage

```
co2_ecp_prices(
  level = c("jurisdiction", "sector", "coverage", "coverage_sector", "region"),
  gas = "CO2",
  country = NULL,
  refresh = FALSE
)
```

Arguments

level	Character. One of "jurisdiction" (default), "sector", "coverage", "coverage_sector", or "region".
gas	Character. Greenhouse gas. Default "CO2".
country	Optional character vector of jurisdiction names to filter (matched case-insensitively, e.g. "United Kingdom").
refresh	Re-download? Default FALSE. Cached copies refresh automatically after 30 days.

Details

`level = "jurisdiction"` returns average effective carbon prices by jurisdiction and year and `"sector"` disaggregates by IPCC sector. Coverage shares, the fraction of emissions subject to a price, come from `"coverage"` (by jurisdiction) and `"coverage_sector"`.

`"region"` is accepted because the publisher documents the endpoint, but their server currently has no data file behind it for any gas; the call errors with that explanation until they restore it.

Value

A data frame.

References

Dolphin, G., Xiahou, Q. (2022). "World carbon pricing database: sources and methods." *Scientific Data*, 9, 573. doi:10.1038/s41597-022-01659-x

See Also

Other aggregators: `co2_icap_prices()`, `co2_icap_systems()`, `co2_rff_pricing()`, `co2_world_bank()`

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  ecp <- co2_ecp_prices(country = "United Kingdom")
})
options(op)
```

---

co2_euets_allocations *EU ETS free allowance allocations*

Description

Returns free allowance allocations per installation per year. Same source file as `co2_euets_emissions()`.

Usage

```
co2_euets_allocations(
  country = NULL,
  year = NULL,
  file_year = NULL,
  scheme = c("ets1", "ets2", "all"),
  refresh = FALSE
)
```

Arguments

country	Optional character vector of two-letter registry codes to filter (e.g. c("DE", "FR", "PL")).
year	Optional integer vector of emissions years. When NULL, returns all years in the latest published file. Note that the file year (e.g. "2025") refers to publication year; emissions data covers calendar years up to publication year -1.
file_year	Publication year of the DG CLIMA file to use. Default is the latest year available in the package (see co2_euets_files()).
scheme	Which trading system to return. "ets1" (default) is the original EU ETS: power, industry, and aviation installations. "ets2" is the separate buildings, road transport, and small industry system that began reporting for compliance year 2024, which DG CLIMA publishes in the same file as one aggregate account per member state. "all" returns both, and is rarely what you want: the two are different systems on different accounting bases, so summing them together double-counts (ETS2 accounts added 981 Mt to the 2024 file, about 45% of the reported total).
refresh	Logical. Re-download? Default FALSE.

Value

A data frame with country, installation_id, installation_name, activity, scheme, year, allocation_eua.

See Also

Other EU ETS: co2_euets_emissions(), co2_euets_files(), co2_euets_installations(), co2_euets_price(), co2_euets_surrendered()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
alloc <- co2_euets_allocations(country = "DE")
options(op)
```

---

co2_euets_emissions *EU ETS verified emissions*

Description

Fetches annual verified greenhouse-gas emissions for installations covered by the EU Emissions Trading System, from the DG CLIMA Union Registry bulk download. Coverage: ~10,000 stationary installations plus ~1,500 aircraft operators across the EU, EEA, and linked Swiss ETS. Emissions data from 2005 onwards; annual files published each April for the prior compliance year.

Usage

```
co2_euets_emissions(
  country = NULL,
  year = NULL,
  file_year = NULL,
  scheme = c("ets1", "ets2", "all"),
  refresh = FALSE
)
```

Arguments

country	Optional character vector of two-letter registry codes to filter (e.g. c("DE", "FR", "PL")).
year	Optional integer vector of emissions years. When NULL, returns all years in the latest published file. Note that the file year (e.g. "2025") refers to publication year; emissions data covers calendar years up to publication year -1.
file_year	Publication year of the DG CLIMA file to use. Default is the latest year available in the package (see co2_euets_files()).
scheme	Which trading system to return. "ets1" (default) is the original EU ETS: power, industry, and aviation installations. "ets2" is the separate buildings, road transport, and small industry system that began reporting for compliance year 2024, which DG CLIMA publishes in the same file as one aggregate account per member state. "all" returns both, and is rarely what you want: the two are different systems on different accounting bases, so summing them together double-counts (ETS2 accounts added 981 Mt to the 2024 file, about 45% of the reported total).
refresh	Logical. Re-download? Default FALSE.

Value

A data frame with columns country, installation_id, installation_name, activity, scheme, year, verified_emissions_tco2e, allocation_eua.

See Also

Other EU ETS: co2_euets_allocations(), co2_euets_files(), co2_euets_installations(), co2_euets_price(), co2_euets_surrendered()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
de <- co2_euets_emissions(country = "DE")
options(op)
```

co2_euets_files *List the EU ETS data file vintages available*

Description

Returns a data frame of the DG CLIMA file vintages available. The package scrapes the Union Registry page at runtime so new annual releases appear automatically; when the page is unreachable, the vintages bundled with the package are listed instead.

Usage

```
co2_euets_files()
```

Value

A data frame.

See Also

Other EU ETS: `co2_euets_allocations()`, `co2_euets_emissions()`, `co2_euets_installations()`, `co2_euets_price()`, `co2_euets_surrendered()`

Examples

```
co2_euets_files()
```

co2_euets_installations *EU ETS installation registry (historical 2012 snapshot)*

Description

Returns the registry of stationary EU ETS installations with operator name, activity type, country, and permit details.

Usage

```
co2_euets_installations(country = NULL, refresh = FALSE)
```

Arguments

`country`	Optional two-letter registry codes.
`refresh`	Re-download? Default `FALSE`.

Details

Vintage warning: the only bulk installation registry DG CLIMA publishes is a Union Registry snapshot dated 28 August 2012 (Phase 2). It predates every installation added since, and permit statuses have not been updated. For current installation-level detail, use the identifier columns returned by `co2_euets_emissions()`, which come from the latest annual verified-emissions file.

Value

A data frame.

See Also

Other EU ETS: `co2_euets_allocations()`, `co2_euets_emissions()`, `co2_euets_files()`, `co2_euets_price()`, `co2_euets_surrendered()`

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  inst <- co2_euets_installations(country = "PL")
})
options(op)
```

---

co2_euets_price *EU ETS allowance (EUA) auction settlement prices*

Description

Fetches EU Allowance (EUA) primary-auction settlement prices from the EEX public auction report. Auctions are held several times per week; each year's file accumulates all auctions for that year.

Usage

```
co2_euets_price(from = NULL, to = NULL, year = NULL, refresh = FALSE)
```

Arguments

`from`, `to`	Optional character or Date. Date range.
`year`	Optional integer year. If supplied, only the file for that year is fetched (faster than `from`/`to`).
`refresh`	Re-download? Default `FALSE`.

Value

A data frame with `date`, `auction_name`, `contract`, `status`, `price_eur`, `min_bid_eur`, `max_bid_eur`, `volume_t`.

See Also

Other EU ETS: co2_euets_allocations(), co2_euets_emissions(), co2_euets_files(), co2_euets_installations
co2_euets_surrendered()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  prices <- co2_euets_price(year = 2025)
})
options(op)
```

---

co2_euets_surrendered *EU ETS compliance status and cumulative surrendered units*

Description

Returns cumulative total surrendered allowances per installation, compliance code, and account closure date. Per-year surrender events require the EUTL transaction log ZIP, which is deferred to v0.2.0 because it is ~1.5 GB uncompressed and requires LZMA extraction.

Usage

```
co2_euets_surrendered(country = NULL, file_year = NULL, refresh = FALSE)
```

Arguments

country	Optional two-letter registry codes.
file_year	Publication year of the compliance file. Default is the latest in the package.
refresh	Re-download? Default FALSE.

Value

A data frame with country, installation_id, installation_name, compliance_code, compliance_status, total_verified_emissions, total_surrendered_allowances, year_of_first_emissions, year_of_last_emissions account_closure.

See Also

Other EU ETS: co2_euets_allocations(), co2_euets_emissions(), co2_euets_files(), co2_euets_installations
co2_euets_price()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  comp <- co2_euets_surrendered(country = "FR")
})
options(op)
```

`co2_icap_prices` *ICAP Allowance Price Explorer*

Description

Fetches allowance prices across 20+ Emissions Trading Systems from the International Carbon Action Partnership (ICAP) Allowance Price Explorer. Covers EU ETS, UK ETS, California, RGGI, New Zealand, Korea, and other jurisdictions with auction and secondary-market prices where available.

Usage

```
co2_icap_prices(jurisdiction = NULL, include_download = FALSE, refresh = FALSE)
```

Arguments

`jurisdiction`	Optional character vector. Filter by jurisdiction name. Matching is case-insensitive and partial, so `"EU ETS"` matches `"European Union Emissions Trading System (from 2019)"`. When `NULL`, returns all. Use `co2_icap_systems()` to list the exact names.
`include_download`	Logical. ICAP publishes a second "(download)" copy of several systems carrying the same prices as the main series. These are excluded by default, because including them duplicates a system in any chart or average. Set `TRUE` to keep them.
`refresh`	Re-download? Default `FALSE`. The cached copy is refreshed automatically when it is more than a day old.

Value

A data frame with `date`, `jurisdiction`, `market_type` (`"primary"` auction or `"secondary"`), `price`, and `currency`.

See Also

Other aggregators: `co2_ecp_prices()`, `co2_icap_systems()`, `co2_rff_pricing()`, `co2_world_bank()`

Examples

```
op <- options(carbondata.cache_dir = tempdir())
prices <- co2_icap_prices(jurisdiction = "EU ETS")
options(op)
```

`co2_icap_systems` *ICAP systems available*

Description

Lists the emissions trading systems carried by the ICAP Allowance Price Explorer, with the date range and number of observations for each. Use this to find the exact `jurisdiction` names accepted by `co2_icap_prices()`.

Usage

```
co2_icap_systems(include_download = FALSE, refresh = FALSE)
```

Arguments

`include_download`	Logical. ICAP publishes a second "(download)" copy of several systems carrying the same prices as the main series. These are excluded by default, because including them duplicates a system in any chart or average. Set `TRUE` to keep them.
`refresh`	Re-download? Default `FALSE`.

Value

A data frame with `jurisdiction`, `currency`, `market_type`, `first_date`, `last_date`, and `n_obs`.

See Also

Other aggregators: `co2_ecp_prices()`, `co2_icap_prices()`, `co2_rff_pricing()`, `co2_world_bank()`

Examples

```
op <- options(carbondata.cache_dir = tempdir())
co2_icap_systems()
options(op)
```

co2_markets *List supported carbon markets*

Description

Returns a data frame of the carbon markets (compliance and voluntary) supported by this package, with coverage metadata.

Usage

```
co2_markets(type = c("all", "compliance", "voluntary"))
```

Arguments

type	Character. One of "all" (default), "compliance", or "voluntary".

Value

A data frame with columns market, name, type, coverage_start, jurisdiction, function_name, and notes.

Examples

```
co2_markets()
co2_markets(type = "compliance")
```

co2_offsets_db *CarbonPlan OffsetsDB (via S3 parquet)*

Description

Fetches the CarbonPlan OffsetsDB daily snapshots of voluntary carbon market projects and credits, stored as Parquet in a public S3 bucket. Covers Verra, ART TREES, Gold Standard, American Carbon Registry, and Climate Action Reserve.

Usage

```
co2_offsets_db(
  kind = c("projects", "credits"),
  date = NULL,
  max_lookback = 180L,
  refresh = FALSE
)
```

Arguments

kind	Character. "projects" (default) or "credits".
date	Optional character. ISO date of the snapshot to fetch. The function walks backwards from this date to find a published snapshot. Default Sys.Date().
max_lookback	Integer. How many days to walk back before giving up. Default 180, because CarbonPlan's publication cadence became irregular during 2026 and the last snapshot seen was 1 June 2026.
refresh	Re-download? Default FALSE.

Details

The CarbonPlan REST API at offsets-db-api.carbonplan.org was deprecated; this function uses the S3 bucket directly. Parquet reading requires an installed Parquet reader; the arrow package is the recommended suggest.

Value

A file path to the downloaded Parquet file. The caller must have arrow or nanoparquet installed to read it.

See Also

Other voluntary markets: co2_cad_trust(), co2_vrod()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  path <- co2_offsets_db("projects")
  if (requireNamespace("arrow", quietly = TRUE)) {
  df <- arrow::read_parquet(path)
  }
})
options(op)
```

---

co2_rff_pricing *RFF World Carbon Pricing Database*

Description

Fetches the Dolphin-Pollitt-Newbery World Carbon Pricing Database for one country: national-level carbon pricing with CO2 tax and ETS instruments harmonised by IPCC sector across 200+ jurisdictions.

Usage

```
co2_rff_pricing(country, version = "v2025.0.0", refresh = FALSE)
```

Arguments

country	Character. Country name using underscores (e.g. "United_Kingdom", "Germany", "Antigua_and_Barbuda").
version	Character. Git tag of the dataset release to read. Default "v2025.0.0", the last openly published snapshot.
refresh	Re-download? Default FALSE.

Details

On 30 May 2026 the maintainer stopped distributing WCPD through GitHub and stripped the data files from the default branch; new releases moved to https://worldcarbonpricing.org behind an emailed download token. This function therefore reads the last openly published snapshot, the v2025.0.0 tag, which remains available and covers 1990 to 2023. For newer vintages, request access on the WCPD site. For the companion emissions-weighted carbon price, which is still served openly, see co2_ecp_prices().

Value

A data frame of annual carbon prices by IPCC sector code.

References

Dolphin, G. G., Pollitt, M. G. and Newbery, D. M. (2020). "The political economy of carbon pricing: a panel analysis." *Oxford Economic Papers*, 72(2), 472–500. doi:10.1093/oep/gpz042

See Also

Other aggregators: co2_ecp_prices(), co2_icap_prices(), co2_icap_systems(), co2_world_bank()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  uk <- co2_rff_pricing("United_Kingdom")
})
options(op)
```

---

co2_rggi_allowances *RGGI allowance distribution by state and year*

Description

Fetches the Regional Greenhouse Gas Initiative (RGGI) annual CO2 allowance distribution table from the public RGGI website. Covers 11 participating US states from 2009 onwards.

Usage

```
co2_rggi_allowances(year = NULL, refresh = FALSE)
```

Arguments

year	Integer year (2009-present). Default is the current year.
refresh	Re-download? Default FALSE.

Details

Auction clearing prices are NOT returned here because RGGI publishes them only as per-auction PDFs (no CSV aggregation exists). For RGGI prices, use co2_icap_prices() with jurisdiction = "Regional Greenhouse Gas Initiative".

Value

A data frame with one row per participating state plus a "Total" row: state and one numeric column per measure in RGGI's table (allowance budget, sold at auction, cost containment reserve, and so on; the set varies a little by year). Cells RGGI marks N/A are NA.

See Also

Other compliance markets: co2_california_caps(), co2_california_prices(), co2_rggi_state_proceeds(), co2_ukets(), co2_ukets_allocations()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  a <- co2_rggi_allowances(year = 2026)
})
options(op)
```

---

co2_rggi_state_proceeds

RGGI cumulative state proceeds by auction

Description

Fetches cumulative auction proceeds time series for one RGGI state, indexed by auction number. Derived from the per-state XLSX files on rggi.org. Combining all states gives the full RGGI auction time series.

Usage

```
co2_rggi_state_proceeds(state, refresh = FALSE)
```

Arguments

state	Two-letter state code (one of: CT, DE, MA, MD, ME, NH, NJ, NY, RI, VA, VT).
refresh	Re-download? Default FALSE.

Value

A data frame.

See Also

Other compliance markets: co2_california_caps(), co2_california_prices(), co2_rggi_allowances(), co2_ukets(), co2_ukets_allocations()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  ny <- co2_rggi_state_proceeds("NY")
})
options(op)
```

---

co2_ukets *UK ETS verified emissions and surrenders*

Description

Fetches the UK Emissions Trading Scheme Section 4 compliance report (verified emissions and allowance surrenders per account per scheme year), published annually by the UK Emissions Trading Registry after the 30 April reconciliation deadline.

Usage

```
co2_ukets(refresh = FALSE)
```

Arguments

refresh	Re-download? Default FALSE.

Value

A data frame with one row per account-year.

See Also

Other compliance markets: co2_california_caps(), co2_california_prices(), co2_rggi_allowances(), co2_rggi_state_proceeds(), co2_ukets_allocations()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  uk <- co2_ukets()
})
options(op)
```

`co2_ukets_allocations` *UK ETS free allocations*

Description

Fetches the UK ETS free-allocation table for stationary installations (OHA) published by DESNZ on GOV.UK. Filename includes a media GUID that changes when DESNZ republishes, so the package scrapes the stable publications landing page.

Usage

```
co2_ukets_allocations(sector = c("installations", "aviation"), refresh = FALSE)
```

Arguments

`sector`	Character. `"installations"` (default) for stationary installations, `"aviation"` for aviation allocations.
`refresh`	Re-download? Default `FALSE`.

Value

A data frame.

See Also

Other compliance markets: `co2_california_caps()`, `co2_california_prices()`, `co2_rggi_allowances()`, `co2_rggi_state_proceeds()`, `co2_ukets()`

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  alloc <- co2_ukets_allocations()
})
options(op)
```

co2_vrod *Berkeley Voluntary Registry Offsets Database*

Description

Fetches the Berkeley GSPP Voluntary Registry Offsets Database, an aggregator of Verra, Gold Standard, ACR, CAR, and ART TREES project registrations and issuances. Released bimonthly under CC BY 4.0.

Usage

```
co2_vrod(sheet = "PROJECTS", refresh = FALSE)
```

Arguments

sheet	Character. Which workbook sheet to read. "PROJECTS" (default) is the project register and the only sheet that is a tidy table: it is parsed properly, with the header row located and tally rows dropped. The workbook's other sheets are charts, documentation, and drop-down lookups rather than datasets; naming one reads it verbatim and you get whatever layout it happens to have.
refresh	Re-download? Default FALSE. Each release has a date-stamped filename; the package scrapes the landing page to find the latest release on each call unless a cached copy exists.

Value

A data frame of project-level data.

See Also

Other voluntary markets: co2_cad_trust(), co2_offsets_db()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
try({
  vrod <- co2_vrod()
})
options(op)
```

co2_world_bank *World Bank Carbon Pricing Dashboard*

Description

Reads the World Bank Carbon Pricing Dashboard Excel file, which covers 70+ carbon pricing initiatives worldwide (carbon taxes and emissions trading systems) with price, coverage, and revenue data.

Usage

```
co2_world_bank(path = NULL, sheet = 1L, refresh = FALSE)
```

Arguments

path	Optional path to a locally downloaded dashboard .xlsx file. When supplied, no network request is made.
sheet	Sheet to read. Default 1. The workbook carries several sheets; pass a name or number to read another.
refresh	Re-download? Default FALSE.

Details

Manual download may be required. Since 2026 the dashboard host rejects non-browser HTTP clients (every request returns HTTP 403, regardless of user agent), so this function often cannot fetch the file for you. When the automatic download fails it reports the direct file URL: download that in a browser and pass the local path via path. The file is republished roughly twice a year, so a manual copy stays current for months.

Value

A data frame.

See Also

Other aggregators: co2_ecp_prices(), co2_icap_prices(), co2_icap_systems(), co2_rff_pricing()

Examples

```
op <- options(carbondata.cache_dir = tempdir())
wb <- try(co2_world_bank(), silent = TRUE)
options(op)
```

Index

∗ **EU ETS**
co2_euets_allocations, 7
co2_euets_emissions, 8
co2_euets_files, 10
co2_euets_installations, 10
co2_euets_price, 11
co2_euets_surrendered, 12
∗ **aggregators**
co2_ecp_prices, 6
co2_icap_prices, 13
co2_icap_systems, 14
co2_rff_pricing, 16
co2_world_bank, 22
∗ **compliance markets**
co2_california_caps, 4
co2_california_prices, 5
co2_rggi_allowances, 17
co2_rggi_state_proceeds, 18
co2_ukets, 19
co2_ukets_allocations, 20
∗ **configuration**
co2_cache_info, 2
co2_clear_cache, 5
∗ **helpers**
co2_markets, 15
∗ **voluntary markets**
co2_cad_trust, 3
co2_offsets_db, 15
co2_vrod, 21

co2_cache_info, 2
co2_cache_info(), *6*
co2_cad_trust, 3
co2_cad_trust(), *16*, *21*
co2_california_caps, 4
co2_california_caps(), *5*, *18–20*
co2_california_prices, 5
co2_california_prices(), *4*, *18–20*
co2_clear_cache, 5
co2_clear_cache(), *2*, *3*
co2_ecp_prices, 6
co2_ecp_prices(), *13*, *14*, *17*, *22*
co2_euets_allocations, 7
co2_euets_allocations(), *9–12*
co2_euets_emissions, 8
co2_euets_emissions(), *7*, *8*, *10–12*
co2_euets_files, 10
co2_euets_files(), *8*, *9*, *11*, *12*
co2_euets_installations, 10
co2_euets_installations(), *8–10*, *12*
co2_euets_price, 11
co2_euets_price(), *8–12*
co2_euets_surrendered, 12
co2_euets_surrendered(), *8–12*
co2_icap_prices, 13
co2_icap_prices(), *7*, *14*, *17*, *18*, *22*
co2_icap_systems, 14
co2_icap_systems(), *7*, *13*, *17*, *22*
co2_markets, 15
co2_offsets_db, 15
co2_offsets_db(), *4*, *21*
co2_rff_pricing, 16
co2_rff_pricing(), *7*, *13*, *14*, *22*
co2_rggi_allowances, 17
co2_rggi_allowances(), *4*, *5*, *19*, *20*
co2_rggi_state_proceeds, 18
co2_rggi_state_proceeds(), *4*, *5*, *18–20*
co2_ukets, 19
co2_ukets(), *4*, *5*, *18–20*
co2_ukets_allocations, 20
co2_ukets_allocations(), *4*, *5*, *18*, *19*
co2_vrod, 21
co2_vrod(), *4*, *16*
co2_world_bank, 22
co2_world_bank(), *7*, *13*, *14*, *17*