Package 'CodeCarbonR'

September 5, 2026

Title Track Energy Consumption and Carbon Emissions of R Code

Version 0.1.0

Description Wraps the Python 'codecarbon' package via 'reticulate' to measure the energy consumption and estimated carbon emissions of R code. Provides a self-contained setup routine that installs 'codecarbon' into a dedicated conda environment, and an R-facing tracker API for measuring a block of code or a longer-running session.

License MIT + file LICENSE

URL https://beabock.github.io/CodeCarbonR/, https://github.com/beabock/CodeCarbonR

BugReports https://github.com/beabock/CodeCarbonR/issues

Encoding UTF-8

RoxygenNote 7.3.3

Imports R6, reticulate

Suggests knitr, pkgdown, rmarkdown, testthat (>= 3.0.0)

Config/Needs/website rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

SystemRequirements Python (>= 3.9); codecarbon Python package (>= 2.2.2), installed via setup_carbon_tracker()

NeedsCompilation no

Author Beatrice Bock [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-2240-9360>), Rachel Paterson [aut], Dylan van Bramer [aut]

Maintainer Beatrice Bock <beabockm@gmail.com>

Repository CRAN

Date/Publication 2026-09-05 13:50:20 UTC

Contents

`carbon_tracker` *Create a carbon emissions tracker*

Description

Wraps a codecarbon `OfflineEmissionsTracker`. Call `$start()` before the code you want to measure and `$stop()` after, or use `with_emissions_tracked()` to measure a single block in one call.

Usage

```
carbon_tracker(country_iso_code = NULL, project_name = "CodeCarbonR", ...)
```

Arguments

`country_iso_code`	3-letter ISO code used to look up grid carbon intensity, e.g. `"USA"`. Call `list_carbon_tracker_countries()` for the supported codes.
`project_name`	Label attached to the tracked run.
`...`	Passed to `codecarbon.OfflineEmissionsTracker`, e.g. `measure_power_secs`, `output_dir`, `log_level`.

Details

Each tracker instance supports one `$start()`/`$stop()` cycle. Calling `$start()` again after `$stop()` on the same instance does not restart measurement, because codecarbon's underlying tracker never resets its internal clock – but exactly what the next `$stop()` returns depends on the installed codecarbon version: codecarbon 2.x returns the first cycle's emissions/energy figures again, frozen; codecarbon 3.x instead keeps accumulating energy from the original start, so the second reading comes back inflated rather than frozen. Either way it is not an isolated measurement of the second cycle's own work, and `duration` keeps climbing from the original start rather than resetting. To measure several phases separately, create a new `carbon_tracker()` per phase; each `$stop()` still appends its own row to the same `output_dir`'s `emissions.csv`.

Value

A `CarbonTracker` R6 object.

Examples

```
# Requires codecarbon to be installed (setup_carbon_tracker()); not run
# on CRAN's check machines, which don't have it.
if (carbon_tracker_ready()) {
  tracker <- carbon_tracker(country_iso_code = "USA", output_dir = tempdir())
  tracker$start()
  Sys.sleep(1)
  emissions <- tracker$stop()
  print(emissions)
}
```

---

carbon_tracker_ready *Check whether codecarbon is installed and importable*

Description

Check whether codecarbon is installed and importable

Usage

```
carbon_tracker_ready()
```

Value

`TRUE` if codecarbon can be imported, `FALSE` otherwise (including when no Python interpreter can be found at all).

Examples

```
# Not \dontrun{} -- this genuinely works, it's just slow: on a machine
# with no Python configured (e.g. a fresh CRAN check environment),
# reticulate's interpreter discovery alone can take >10s before this
# returns FALSE. \donttest{} keeps it out of the default check timing
# while still letting CRAN's extended checks and interactive users
# verify it actually runs.
carbon_tracker_ready()
```

list_carbon_tracker_countries *List supported country codes for carbon intensity lookup*

Description

Reads the country energy mix data bundled with the installed codecarbon package. The `country_iso_code` argument to `carbon_tracker()` and `with_emissions_tracked()` must be one of the codes returned here.

Usage

```
list_carbon_tracker_countries()
```

Value

A data frame with `iso_code` and `country_name` columns.

Examples

```
# Requires codecarbon to be installed (setup_carbon_tracker()); not run
# on CRAN's check machines, which don't have it.
if (carbon_tracker_ready()) {
  countries <- list_carbon_tracker_countries()
  head(countries)
}
```

setup_carbon_tracker *Install codecarbon into a dedicated conda environment*

Description

Installs Miniconda if it isn't already present, then creates the "r-codecarbon" conda environment and installs codecarbon into it. Run this once per machine before using `carbon_tracker()` or `with_emissions_tracked()`. Nothing is installed without confirmation, and the function refuses to run outside an interactive session.

Usage

```
setup_carbon_tracker(force = FALSE)
```

Arguments

`force`	Reinstall codecarbon even if it's already available.

Details

Installs codecarbon >= 2.2.2 – a floor, not an exact pin. That's the oldest version this package has actually been validated against (see comparison/coverage_matrix.md and NEWS.md for exactly which versions were validated, and where); newer codecarbon releases are expected and welcome, since they bring updated carbon-intensity data along with whatever bug fixes landed upstream. codecarbon's behavior has changed between versions before (see carbon_tracker()'s docs on tracker restart), so if something about CodeCarbonR's output looks different after a codecarbon upgrade, that's the first thing to check. Keep the floor here in sync with DESCRIPTION's SystemRequirements field if it changes.

Value

Invisibly, TRUE if codecarbon is ready to use after the call, FALSE if setup was cancelled.

Examples

```
## Not run:
# Installs software and prompts for confirmation, so this never runs
# under R CMD check (or any other non-interactive session) -- call it
# once, by hand, from an interactive R console.
setup_carbon_tracker()

## End(Not run)
```

---

with_emissions_tracked

Track emissions for a single block of code

Description

Track emissions for a single block of code

Usage

```
with_emissions_tracked(expr, country_iso_code = NULL, ...)
```

Arguments

expr	Code to run and measure.
country_iso_code	3-letter ISO code used to look up grid carbon intensity, e.g. "USA". Call list_carbon_tracker_countries() for the supported codes.
...	Passed to carbon_tracker().

Value

A carbon_emissions_result: $result holds the value of expr, $emissions holds the carbon_emissions object.

Examples

```
# Requires codecarbon to be installed (setup_carbon_tracker()); not run
# on CRAN's check machines, which don't have it.
if (carbon_tracker_ready()) {
  out <- with_emissions_tracked(
    Sys.sleep(1),
    country_iso_code = "USA",
    output_dir = tempdir()
  )
  print(out)
}
```

Index