Package 'emissionTrackerR'

March 28, 2025

Title Lightweight Emission Tracking for Experiments

Version 0.0.1

Description Track carbon emissions for code experiments, with logging, metadata, and comparisons.

Depends R (>= 4.0.0)

License MIT

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

R topics documented:

`append_emissions_logs` *Append Emissions Metadata to Log Files (CSV and JSON)*

Description

Appends metadata from a single run to both a cumulative CSV and JSON file. Automatically flattens nested fields (like 'equivalents') and aligns columns with existing logs.

Usage

```
append_emissions_logs(
  metadata,
  csv_path = "emissions_log.csv",
  json_path = "emissions_log.json"
)
```

Arguments

metadata	A metadata list from ‘collect_metadata()‘.
csv_path	Path to the CSV log file.
json_path	Path to the JSON log file.

Value

None (writes files as side effect).

Examples

```
meta <- collect_metadata(duration = 3, emissions = 0.001)
append_emissions_logs(meta)
```

---

benchmark_emissions *Benchmark Emissions Across Multiple Methods*

Description

Runs each method and logs emissions, duration, and custom metadata.

Usage

```
benchmark_emissions(
  methods,
  setup_fn,
  csv_path = "emissions_log_comparative.csv",
  json_path = "emissions_log_comparative.json"
)
```

Arguments

methods	A named list of functions. Each function should return a performance metric (e.g. accuracy).
setup_fn	A function that returns a list of shared data to pass to each method.
csv_path	Path to the CSV file to append logs to.
json_path	Path to the JSON file to append logs to.

Value

A data.frame summarizing emissions and accuracy per method

collect_metadata *Collect Emissions Metadata*

Description

Aggregates metadata about an experiment run, including timestamp, duration, system and location info, energy usage, and equivalent emissions in real-world terms.

Usage

```
collect_metadata(duration, emissions, project_name = "codecarbon", ...)
```

Arguments

duration	Duration of the task (in seconds).
emissions	Emissions produced (in kilograms of CO2).
project_name	Optional project name (defaults to ‘"codecarbon"‘).
...	Additional parameters (not currently used).

Value

A named list of metadata fields.

Examples

```
collect_metadata(duration = 5, emissions = 0.001)
```

EmissionsTracker *EmissionsTracker R6 Class*

Description

The ‘EmissionsTracker‘ class allows you to estimate CO2 emissions based on the duration of a computation and a fixed energy factor.

Details

A lightweight tracker for estimating carbon emissions (kg CO2) of a code block by measuring the elapsed time and applying a fixed energy-to-emission factor.

Value

An R6 object of class ‘EmissionsTracker‘.

Public fields

`start_time` POSIX time when tracking began.

`end_time` POSIX time when tracking stopped.

`emissions` Estimated emissions in kg CO2.

`duration` Duration of the computation (in seconds).

`energy_factor` Emissions per second (default is 0.0002 kg/sec).

Methods

Public methods:

- EmissionsTracker$new()
- EmissionsTracker$start()
- EmissionsTracker$stop()
- EmissionsTracker$clone()

Method `new()`**:**

Usage:

```
EmissionsTracker$new(energy_factor = 2e-04)
```

Arguments:

`energy_factor` A numeric value representing emissions per second (kg CO2/sec).

Method `start()`**:**

Usage:

```
EmissionsTracker$start()
```

Method `stop()`**:**

Usage:

```
EmissionsTracker$stop()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
EmissionsTracker$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
tracker <- EmissionsTracker$new()
tracker$start()
Sys.sleep(1)
tracker$stop()
```

`log_emissions` *Log Emissions to JSON*

Description

Writes a metadata list of emissions results to a .json file.

Usage

```
log_emissions(data, filepath = "emissions_log.json")
```

Arguments

`data`	A named list of metadata fields (e.g., from ‘collect_metadata()‘).
`filepath`	Path to the JSON file (default is ‘"emissions_log.json"‘).

Details

Log Emissions to JSON

Value

None (writes file as side effect).

Examples

```
log_emissions(list(timestamp = Sys.time(), emissions = 0.001))
```

---

log_emissions_csv *Log Emissions to a CSV File (Single Row)*

Description

Flattens and writes a metadata list of emissions results to a '.csv' file. Designed for one-time runs with a single record.

Usage

```
log_emissions_csv(data, filepath = "emissions_log.csv")
```

Arguments

data	A named list of metadata fields (e.g., from 'collect_metadata()').
filepath	Path to the CSV file (default is '"emissions_log.csv"').

Value

None (writes file as side effect).

Examples

```
log_emissions_csv(list(project_name = "demo", emissions = 0.002))
```

---

track_emissions_for *Track Emissions for an Expression or Task*

Description

Wraps a block of code (expression) to track execution time and estimate

Usage

```
track_emissions_for(task_name, expr)
```

Arguments

task_name	A short string describing the task (e.g., '"train_model"'). Used as the 'project_name' in the metadata.
expr	An R expression to evaluate (e.g., a block of code to measure).

Details

Track Emissions for an Expression or Task

Value

The result of evaluating 'expr', invisibly.

Examples

```
track_emissions_for("example_sleep", {
  Sys.sleep(2)
})

track_emissions_for("iris_rf", {
  model <- randomForest(Species ~ ., data = iris)
})
```

Index