

Package: epitargets (via r-universe)

May 24, 2026

Title Economic Policy Institute tools for working with targets

Version 0.3.0

Description Additional utilities to use the targets R package.

License MIT + file LICENSE

URL <https://economic.github.io/epitargets/>,
<https://github.com/Economic/epitargets>

BugReports <https://github.com/economic/epitargets/issues>

Imports arrow, readr, targets, tarchetypes, tibble

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website economic/epitemplate

Config/pak/sysreqs cmake libglpk-dev make libuv1-dev libxml2-dev libssl-dev libx11-dev

Repository <https://economic.r-universe.dev>

Date/Publication 2026-04-18 19:09:10 UTC

RemoteUrl <https://github.com/economic/epitargets>

RemoteRef HEAD

RemoteSha 626064e53c75723ece9946ccdb9c137c76741fd2

Contents

collect_target_date	2
create_csv	2
tar_age_date	3
tar_csv_read	4
tar_parquet_read	5
tar_target_date	6

Index	7
--------------	----------

`collect_target_date` *Collect date targets into a tibble*

Description

Gathers the values of companion `_date` targets (created by `tar_target_date()` or `tar_age_date()`) into a `tibble::tibble` with columns name and time.

Usage

```
collect_target_date(...)
```

Arguments

... Date values from `_date` targets. Pass each `_date` target by name (e.g., `x_date`, `y_date`).

Value

A `tibble::tibble` with columns name (character) and time.

Examples

```
collect_target_date(x_date = as.Date("2024-01-01"), y_date = as.Date("2024-06-15"))
```

`create_csv` *Write data to a CSV file and return the file path*

Description

A thin wrapper around `readr::write_csv()` that writes data to file and returns the file path. This is useful as a targets format function where the target value should be the path to the written file.

Usage

```
create_csv(data, file, ...)
```

Arguments

`data` A data frame to write.
`file` A string giving the file path to write to.
... Additional arguments passed to `readr::write_csv()`.

Value

The file path `file`, invisibly.

Examples

```
targets::tar_dir({
  create_csv(data.frame(x = 1:3), "test.csv")
})
```

tar_age_date	<i>Create an age-cued target with a companion date target</i>
--------------	---

Description

Like `tar_target_date()` but adds an age-based cue via `tarchetypes::tar_cue_age_raw()` so the target automatically re-runs after a configurable time period. Useful for targets that fetch data from APIs.

Usage

```
tar_age_date(name, command, age = as.difftime(1, units = "days"), ...)
```

Arguments

name	Symbol, name of the target.
command	Expression, R command to run the target.
age	A <code>difftime</code> object specifying the maximum age before the target re-runs. Defaults to 1 day.
...	Additional arguments passed to <code>targets::tar_target_raw()</code> .

Value

A list of two target objects: the primary target (with an age cue) and the companion `_date` target.

Examples

```
if (identical(Sys.getenv("TAR_EXAMPLES"), "true")) {
  targets::tar_dir({
    targets::tar_script({
      library(targets)
      list(
        epitargets::tar_age_date(x, 1 + 1, age = as.difftime(1, units = "days"))
      )
    })
    targets::tar_make()
    targets::tar_read(x)
    targets::tar_read(x_date)
  })
}
```

tar_csv_read	<i>Read a CSV file as a target</i>
--------------	------------------------------------

Description

A convenience wrapper around `tarchetypes::tar_file_read()` for CSV files. Creates a pair of targets: one to track the file with `format = "file"`, and another to read the file with `readr::read_csv()`.

Usage

```
tar_csv_read(name, command, .read_csv_args = list(show_col_types = FALSE), ...)
```

Arguments

<code>name</code>	Symbol, name of the target.
<code>command</code>	Expression, R code that returns the file path to the CSV.
<code>.read_csv_args</code>	A named list of additional arguments passed to <code>readr::read_csv()</code> . Defaults to <code>list(show_col_types = FALSE)</code> . Supplying this argument replaces the defaults entirely.
<code>...</code>	Additional arguments passed to <code>targets::tar_target_raw()</code> for the read target.

Details

The storage format of the read target is inherited from `targets::tar_option_get("format")` (by default `"rds"`). Callers can override it by passing e.g. `format = "qs"` via `...`

Value

A list of two target objects: a file-tracking target (`name_file`) and a CSV-reading target (`name`).

Examples

```
if (identical(Sys.getenv("TAR_EXAMPLES"), "true")) {
  targets::tar_dir({
    targets::tar_script({
      library(targets)
      list(
        epitargets::tar_csv_read(my_data, "data.csv")
      )
    })
    targets::tar_manifest()
  })
}
```

tar_parquet_read	<i>Read a Parquet file as a target</i>
------------------	--

Description

A convenience wrapper around `tarchetypes::tar_file_read()` for Parquet files. Creates a pair of targets: one to track the file with `format = "file"`, and another to read the file with `arrow::read_parquet()`.

Usage

```
tar_parquet_read(name, command, .read_parquet_args, ...)
```

Arguments

name	Symbol, name of the target.
command	Expression, R code that returns the file path to the Parquet file.
.read_parquet_args	A named list of additional arguments passed to <code>arrow::read_parquet()</code> .
...	Additional arguments passed to <code>targets::tar_target_raw()</code> for the read target.

Details

The read target's storage format defaults to "parquet" (overriding the usual `targets::tar_option_get("format")` inheritance), so the cached object is written as a Parquet file in `_targets/objects/`. Callers can override this by passing e.g. `format = "rds"` via `...`

Value

A list of two target objects: a file-tracking target (`name_file`) and a Parquet-reading target (`name`).

Examples

```
if (identical(Sys.getenv("TAR_EXAMPLES"), "true")) {
  targets::tar_dir({
    targets::tar_script({
      library(targets)
      list(
        epitargets::tar_parquet_read(my_data, "data.parquet")
      )
    })
    targets::tar_manifest()
  })
}
```

tar_target_date	<i>Create a target with a companion date target</i>
-----------------	---

Description

Wraps `targets::tar_target_raw()` to create a primary target and a companion `_date` target that records `Sys.Date()` whenever the primary target runs.

Usage

```
tar_target_date(name, command, ...)
```

Arguments

name	Symbol, name of the target.
command	Expression, R command to run the target.
...	Additional arguments passed to <code>targets::tar_target_raw()</code> .

Value

A list of two target objects: the primary target and the companion `_date` target.

Examples

```
if (identical(Sys.getenv("TAR_EXAMPLES"), "true")) {
  targets::tar_dir({
    targets::tar_script({
      library(targets)
      list(
        epitargets::tar_target_date(x, 1 + 1)
      )
    })
    targets::tar_make()
    targets::tar_read(x)
    targets::tar_read(x_date)
  })
}
```

Index

`arrow::read_parquet()`, 5

`collect_target_date`, 2

`create_csv`, 2

`difftime`, 3

`readr::read_csv()`, 4

`readr::write_csv()`, 2

`Sys.Date()`, 6

`tar_age_date`, 3

`tar_age_date()`, 2

`tar_csv_read`, 4

`tar_parquet_read`, 5

`tar_target_date`, 6

`tar_target_date()`, 2, 3

`tarchetypes::tar_cue_age_raw()`, 3

`tarchetypes::tar_file_read()`, 4, 5

`targets::tar_target_raw()`, 3–6

`tibble::tibble`, 2