

Package: epiextractr (via r-universe)

October 11, 2026

Type Package

Title Tools to use Economic Policy Institute Microdata Extracts

Version 0.12.0

Description Tools to download and load the EPI microdata extracts from
microdata.epi.org.

License MIT + file LICENSE

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

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

Encoding UTF-8

LazyData true

Imports cli, purrr, tibble, arrow (>= 2.0), magrittr, rlang,
data.table, utils, haven, dplyr

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

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Config/testthat/edition 3

Config/Needs/website economic/epitemplate

Config/pak/sysreqs cmake make libssl-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-08-12 17:38:13 UTC

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

RemoteRef HEAD

RemoteSha 2f29f5abc8e5139a9ead268dc829a7df9e46e4f7

Contents

cps_files	2
cps_metadata	3
download_cps	4
load_cps	4

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

cps_files	<i>Get file paths for EPI CPS extracts</i>
-----------	--

Description

Returns file paths for the specified CPS sample and years. Useful for targets-based workflows where file dependencies need to be tracked. The result can be passed directly to `load_cps()` or the `load_org()`, `load_basic()`, etc. wrappers in place of `.years`.

Usage

```
cps_files(
  sample,
  years,
  extracts_dir = NULL,
  .quiet = getOption("epiextractr.quiet", FALSE)
)
```

Arguments

sample	CPS sample ("org", "basic", "march", "may")
years	years of CPS data (integers)
extracts_dir	directory where EPI extracts are
.quiet	Logical. Suppress informational messages? Defaults to <code>getOption("epiextractr.quiet", FALSE)</code> .

Value

A character vector of file paths

Examples

```
cps_files("org_sample", 2023:2025)

# Pass directly to load functions:
load_org_sample(cps_files("org_sample", 2023:2025), year, month, wage)

## Not run:
# Use with targets
```

```
library(tarchetypes)
tar_assign({
  org_files = cps_files("org", 2020:2025) |> tar_file()
  org_data = load_org(org_files, year, month, wage) |> tar_target()
})

## End(Not run)
```

cps_metadata	<i>Retrieve metadata from CPS extract</i>
--------------	---

Description

Retrieve metadata from CPS extract

Usage

```
cps_version(x)

cps_citation(x)

assert_cps_version(x, version)
```

Arguments

x	EPI CPS extract generated from load_cps() functions
version	String version number

Value

cps_version and cps_citation return version or citation strings.

assert_cps_version returns an error when the provided version is incorrect.

Examples

```
cps_org <- load_org_sample(2023:2025)
cps_citation(cps_org)
cps_version(cps_org)
```

download_cps	<i>Download EPI CPS extracts data</i>
--------------	---------------------------------------

Description

Download the EPI CPS extracts to your local machine

Usage

```
download_cps(sample, extracts_dir = NULL, overwrite = FALSE)
```

Arguments

sample	CPS sample ("org", "basic", "may")
extracts_dir	directory where EPI extracts should be placed
overwrite	when TRUE, overwrite data

Value

downloaded files

Examples

```
## Not run:
download_cps(sample = "march", extracts_dir = "/data/cps")

## End(Not run)
```

load_cps	<i>Load a selection of EPI CPS extracts</i>
----------	---

Description

Select years and variables from the EPI CPS microdata extracts. These data must first be downloaded using `download_cps()` or from <https://microdata.epi.org>.

Usage

```
load_cps(
  .sample,
  .years,
  ...,
  .extracts_dir = NULL,
  .version_check = TRUE,
  .quiet = getOption("epiextractr.quiet", FALSE)
)
```

```
load_basic(
  .years,
  ...,
  .extracts_dir = NULL,
  .version_check = TRUE,
  .quiet = getOption("epiextractr.quiet", FALSE)
)
```

```
load_may(
  .years,
  ...,
  .extracts_dir = NULL,
  .version_check = TRUE,
  .quiet = getOption("epiextractr.quiet", FALSE)
)
```

```
load_org(
  .years,
  ...,
  .extracts_dir = NULL,
  .version_check = TRUE,
  .quiet = getOption("epiextractr.quiet", FALSE)
)
```

```
load_org_sample(
  .years,
  ...,
  .extracts_dir = NULL,
  .version_check = TRUE,
  .quiet = getOption("epiextractr.quiet", FALSE)
)
```

Arguments

<code>.sample</code>	CPS sample ("org", "basic", "march", "may")
<code>.years</code>	years of CPS data (integers), or file paths from <code>cps_files()</code>
<code>...</code>	tidy selection of variables to keep
<code>.extracts_dir</code>	directory where EPI extracts are
<code>.version_check</code>	when TRUE, confirm data are same version
<code>.quiet</code>	Logical. Suppress informational messages? Defaults to <code>getOption("epiextractr.quiet", FALSE)</code> .

Details

All columns are selected if `...` is missing.

`.years` accepts either integer years or file paths from `cps_files()`. When file paths are passed, the files are read directly and `.extracts_dir` is ignored.

.extracts_dir is required, but if NULL it will look for the environment variables

```
EPIEXTRACTS_CPSBASIC_DIR
EPIEXTRACTS_CPSORG_DIR
EPIEXTRACTS_CPSMAY_DIR
```

which could be set in your .Renviron, for example.

Value

A tibble of CPS microdata

Functions

- load_cps(): base function group
- load_basic(): Load CPS Basic Monthly files
- load_may(): Load CPS May files
- load_org(): Load CPS ORG files
- load_org_sample(): Load a demonstration sample of CPS ORG files; only useful for examples

Examples

```
# Load all columns from the demonstration sample
load_org_sample(2023:2024)

# Load a selection of columns
load_org_sample(2023:2025, year, month, female, wage)

# Use cps_files() for targets workflows:
org_files = cps_files("org_sample", 2023:2025)
load_org_sample(org_files, year, month, wage)

## Not run:
# Load real CPS ORG data (requires downloaded extracts):
load_org(2010:2019, year, month, orgwgt, female, wage)
# This is equivalent to
load_cps("org", 2010:2019, year, month, orgwgt, female, wage)

## End(Not run)
```

Index

`assert_cps_version (cps_metadata)`, [3](#)

`cps_citation (cps_metadata)`, [3](#)

`cps_files`, [2](#)

`cps_metadata`, [3](#)

`cps_version (cps_metadata)`, [3](#)

`download_cps`, [4](#)

`load_basic (load_cps)`, [4](#)

`load_cps`, [4](#)

`load_may (load_cps)`, [4](#)

`load_org (load_cps)`, [4](#)

`load_org_sample (load_cps)`, [4](#)