Package 'cpsvote'

June 25, 2026

Title A Toolbox for Using the CPS's Voting and Registration Supplement

Version 0.2.0

Description Provides automated methods for downloading, recoding, and merging selected years of the Current Population Survey's Voting and Registration Supplement <https://www.nber.org/research/data/current-population-survey-cps-supplements-voting-and-registration>, a large N national survey about registration, voting, and non-voting in United States federal elections. Provides documentation for appropriate use of sample weights to generate statistical estimates, drawing from Hur & Achen (2013) <doi:10.1093/poq/nft042> and McDonald (2018) <https://www.electproject.org/election-data/voter-turnout-data>.

URL https://github.com/Reed-EVIC/cpsvote, https://reed-evic.github.io/cpsvote/

BugReports https://github.com/Reed-EVIC/cpsvote/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

Suggests knitr, tidyverse, rmarkdown, survey, srvyr, here, scales, ggplot2, usmap, ggthemes, tweenr

VignetteBuilder knitr

RoxygenNote 7.3.3

Imports magrittr, readr, dplyr, stringr, forcats (>= 1.0.0), rlang

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Jay Lee [aut], Paul Gronke [aut, cre], Frank Adonteng [ctb], John Curiel [ctb]

Maintainer Paul Gronke <gronkep@reed.edu>

Repository CRAN

Date/Publication 2026-06-25 11:00:29 UTC

Contents

cps_2016_10k 2
cps_2020_10k 3
cps_2022_10k 4
cps_2024_10k 5
cps_allyears_100k 6
cps_cols 7
cps_data_dir 7
cps_docs_dir 8
cps_download_data 9
cps_download_docs 10
cps_factors 11
cps_label 11
cps_load_basic 12
cps_read 13
cps_read_year 14
cps_recode_vote 15
cps_refactor 16
cps_reweight 16
cps_reweight_turnout 17
na_ifin 18

Index **19**

`cps_2016_10k` *A sample of the raw 2016 CPS dataset*

Description

This is a 10,000 row sample of the data that comes out of `cps_read(years = 2016)`.

Usage

```
cps_2016_10k
```

Format

A tibble with 10,000 rows and 17 columns:

FILE Which default file the case came from

YEAR Year of interview

STATE State postal abbreviation

AGE Person's age as of the end of survey week; topcoded at 80 and 85

SEX Binary sex

EDUCATION Highest level of school completed or degree received

RACE Race

HISPANIC Hispanic status

WEIGHT Original CPS survey weight

VRS_VOTE Whether respondent voted in the election; self-reported

VRS_REG Whether respondent was registered to vote in the election; self-reported

VRS_REG_WHYNOT Reason for not being registered to vote

VRS_VOTE_WHYNOT Reason for not voting

VRS_VOTEMODE_2004toPRESENT Whether respondent voted by mail

VRS_VOTEWHEN_2004toPRESENT Whether respondent voted on election day or before

VRS_REG_METHOD Method of registration

VRS_RESIDENCE Duration of time living at current address

`cps_2020_10k` *A sample of the raw 2020 CPS dataset*

Description

This is a 10,000 row sample of the data that comes out of `cps_read(years = 2020)`.

Usage

```
cps_2020_10k
```

Format

A tibble with 10,000 rows and 17 columns:

FILE Which default file the case came from

YEAR Year of interview

STATE State postal abbreviation

AGE Person's age as of the end of survey week; topcoded at 80 and 85

SEX Binary sex

EDUCATION Highest level of school completed or degree received

RACE Race

HISPANIC Hispanic status

WEIGHT Original CPS survey weight

VRS_VOTE Whether respondent voted in the election; self-reported

VRS_REG Whether respondent was registered to vote in the election; self-reported

VRS_REG_WHYNOT Reason for not being registered to vote

VRS_VOTE_WHYNOT Reason for not voting

VRS_VOTEMODE_2004toPRESENT Whether respondent voted by mail

VRS_VOTEWHEN_2004toPRESENT Whether respondent voted on election day or before

VRS_REG_METHOD Method of registration

VRS_RESIDENCE Duration of time living at current address

`cps_2022_10k` *A sample of the raw 2022 CPS dataset*

Description

This is a 10,000 row sample of the data that comes out of `cps_read(years = 2022)`.

Usage

```
cps_2022_10k
```

Format

A tibble with 10,000 rows and 17 columns:

FILE Which default file the case came from

YEAR Year of interview

STATE State postal abbreviation

AGE Person's age as of the end of survey week; topcoded at 80 and 85

SEX Binary sex

EDUCATION Highest level of school completed or degree received

RACE Race

HISPANIC Hispanic status

WEIGHT Original CPS survey weight

VRS_VOTE Whether respondent voted in the election; self-reported

VRS_REG Whether respondent was registered to vote in the election; self-reported

VRS_REG_WHYNOT Reason for not being registered to vote

VRS_VOTE_WHYNOT Reason for not voting

VRS_VOTEMODE_2004toPRESENT Whether respondent voted by mail

VRS_VOTEWHEN_2004toPRESENT Whether respondent voted on election day or before

VRS_REG_METHOD Method of registration

VRS_RESIDENCE Duration of time living at current address

`cps_2024_10k` *A sample of the raw 2024 CPS dataset*

Description

This is a 10,000 row sample of the data that comes out of `cps_read(years = 2024)`.

Usage

```
cps_2024_10k
```

Format

A tibble with 10,000 rows and 17 columns:

FILE Which default file the case came from

YEAR Year of interview

STATE State postal abbreviation

AGE Person's age as of the end of survey week; topcoded at 80 and 85

SEX Binary sex

EDUCATION Highest level of school completed or degree received

RACE Race

HISPANIC Hispanic status

WEIGHT Original CPS survey weight

VRS_VOTE Whether respondent voted in the election; self-reported

VRS_REG Whether respondent was registered to vote in the election; self-reported

VRS_REG_WHYNOT Reason for not being registered to vote

VRS_VOTE_WHYNOT Reason for not voting

VRS_VOTEMODE_2004toPRESENT Whether respondent voted by mail

VRS_VOTEWHEN_2004toPRESENT Whether respondent voted on election day or before

VRS_REG_METHOD Method of registration

VRS_RESIDENCE Duration of time living at current address

`cps_allyears_100k` *A sample of the full CPS dataset*

Description

This is a 100,000 row sample of the data that comes out of `cpsvote::cps_load_basic`.

Usage

```
cps_allyears_100k
```

Format

A tibble with 100,000 rows and 25 columns:

FILE Which default file the case came from

YEAR Year of interview

STATE State postal abbreviation

AGE Person's age as of the end of survey week; topcoded at 90 until 2002, 80 in 2004, and 80/85 after

SEX Binary sex

EDUCATION Highest level of school completed or degree received

RACE Race

HISPANIC Hispanic status

WEIGHT Original CPS survey weight

VRS_VOTE Whether respondent voted in the election; self-reported

VRS_REG Whether respondent was registered to vote in the election; self-reported

VRS_VOTE_TIME What time of day respondent voted

VRS_RESIDENCE Duration of time living at current address

VRS_VOTE_WHYNOT Reason for not voting

VRS_VOTEMETHOD_1996to2002 Method of voting, pre-2004

VRS_REG_SINCE95 Whether respondent had registered to vote since 1995

VRS_REG_DMV Whether respondent registered at the DMV

VRS_REG_METHOD Method of registration

VRS_REG_WHYNOT Reason for not being registered to vote

VRS_VOTEMODE_2004toPRESENT Whether respondent voted by mail, 2004 on

VRS_VOTEWHEN_2004toPRESENT Whether respondent voted on election day or before, 2004 on

VRS_VOTEMETHOD_CON A consolidation of VRS_VOTEMETHOD_1996to2002, VRS_VOTEMODE_2004toPRESE and VRS_VOTEWHEN_2004toPRESENT

cps_turnout Recode of VRS_VOTE for CPS turnout calculation

hurachen_turnout Recode of VRS_VOTE for adjusted Hur & Achen turnout calculation

turnout_weight Adjusted weight for calculating voter turnout (per Hur & Achen)

cps_cols *Sample column specifications for reading CPS data*

Description

Because the CPS is a fixed-width file that changes data locations (and variable names) across years, to correctly read the data you have to specify which start/end positions correspond to which column names in each year. This is one such specification. To add extra data or change column names, see the Vignette.

Usage

```
cps_cols
```

Format

A data frame with 253 rows and 8 columns:

year year

cps_name original column name as given by the CPS

new_name a new name, which tries to describe the variable and join sensibly across multiple years

start_pos which character of a line the variable starts with

end_pos which character of a line the variable ends with

col_type whether the column is character, numeric, or a factor

description the question text/description from the CPS

notes any notes for question administration or analysis

cps_data_dir *Get the cpsvote data directory*

Description

Returns the path where downloaded CPS data files are stored. The default is `~/cps_data`, which expands to your home directory on any platform.

Usage

```
cps_data_dir()
```

Details

To use a different location persistently, add this to your `.Rprofile`:

```
options(cpsvote.datadir = "~/my/path/to/cps_data")
```

Any path set this way will be used automatically by `cps_load_basic()`, `cps_read()`, and `cps_download_data()` without needing to pass `datadir` each time.

Value

A character string giving the expanded data directory path.

See Also

cps_load_basic(), cps_read(), cps_download_data()

---

cps_docs_dir *Get the cpsvote documentation directory*

Description

Returns the path where downloaded CPS technical documentation files are stored. The default is ~/cps_docs, which expands to your home directory on any platform.

Usage

```
cps_docs_dir()
```

Details

To use a different location persistently, add this to your .Rprofile:

```
options(cpsvote.docsdir = "~/my/path/to/cps_docs")
```

Any path set this way will be used automatically by cps_download_docs() without needing to pass path each time.

Value

A character string giving the expanded documentation directory path.

See Also

cps_download_docs()

`cps_download_data` *Download CPS microdata*

Description

Download CPS microdata

Usage

```
cps_download_data(
  path = cps_data_dir(),
  years = seq(1994, 2024, 2),
  overwrite = FALSE
)
```

Arguments

`path`	A file path (relative or absolute) where the downloads should go. Defaults to `cps_data_dir()`, which returns `~/cps_data` unless overridden via `options(cpsvote.datadir = "your/path")`.
`years`	Which years of data to download. Defaults to all even-numbered years from 1994 to 2024.
`overwrite`	Logical, whether to write over existing files or not. Defaults to FALSE.

Details

- File names will be written in the style "cps_nov2024.zip", with the appropriate years.
- The Voting and Registration Supplement is only conducted in even-numbered years (since 1964), so any entry in `years` outside of this will be skipped.
- Currently the package only supports downloads from 1994 onwards, so any entry in `years` before 1994 will be skipped.

Examples

```
## Not run:
cps_download_data(years = 2016, overwrite = TRUE)

## End(Not run)
```

`cps_download_docs` *Download CPS technical documentation*

Description

Download CPS technical documentation

Usage

```
cps_download_docs(
  path = cps_docs_dir(),
  years = seq(1994, 2024, 2),
  overwrite = FALSE
)
```

Arguments

`path`	A file path (relative or absolute) where the downloads should go. Defaults to `cps_docs_dir()`, which returns `~/cps_docs` unless overridden via `options(cpsvote.docsdir = "your/path")`.
`years`	Which years of documentation to download. Defaults to all even-numbered years from 1994 to 2024.
`overwrite`	Logical, whether to write over existing files or not. Defaults to FALSE.

Details

- File names will be written in the style "cps_nov2024.pdf", with the appropriate years.
- The Voting and Registration Supplement is only conducted in even-numbered years (since 1964), so any entry in `years` outside of this will be skipped.
- Currently the package only supports downloads from 1994 onwards, so any entry in `years` before 1994 will be skipped.

Examples

```
## Not run:
cps_download_docs(years = 2016, overwrite = TRUE)

## End(Not run)
```

cps_factors *Sample factor specifications for reading CPS data*

Description

Because the CPS changes factor levels across years, to correctly read the data you have to specify which numeric codes correspond to which character values in each year. This is one such specification. To add extra data, see the Vignette.

Usage

```
cps_factors
```

Format

A data frame with 2,475 rows and 5 columns:

year year

cps_name original column name as given by the CPS

new_name a new name, which tries to describe the variable and join sensibly across multiple years

code the numeric code contained in the raw CPS data

value the character value corresponding to each numeric code

Details

These match the exact specifications from the CPS, including NA codes and any typos that occur (e.g., "Hipsanic" is common in older years).

cps_label *Apply factor levels to raw CPS data*

Description

The CPS publishes their data in a numeric format, with a separate PDF codebook (not machine readable) describing factor values. This function labels the raw numeric CPS data according to a supplied factor key. Codes that appear in a given year and are not included in `factors` will be recoded as `NA`.

Usage

```
cps_label(
  data,
  factors = cpsvote::cps_factors,
  names_col = "new_name",
  na_vals = c("-1", "BLANK", "NOT IN UNIVERSE"),
  expand_year = TRUE,
  rescale_weight = TRUE,
  toupper = TRUE
)
```

Arguments

data	The raw CPS data that factors should be applied to
factors	A data frame containing the label codes to be applied
names_col	Which column of factors contains the column names of data
na_vals	Which character values should be considered "missing" across the dataset and be set to NA after labelling
expand_year	Whether to change the two-digit year listed in earlier surveys (94, 96) into a four-digit year (1994, 1996)
rescale_weight	Whether to rescale the weight, dividing by 10,000. The CPS describes the given weight as having "four implied decimals", so this rescaling adjusts the weight to produce sensible population totals.
toupper	Whether to convert all factor levels to uppercase

Value

CPS data with factor labels in place of the raw numeric data

Examples

```
cps_label(cps_2016_10k)
```

---

cps_load_basic *load some basic/default CPS data into the environment*

Description

This function is a quick starter to working with the CPS, using all of the defaults that are baked into this package. Because the data is so large, it made more sense to ship a "basic" CPS data set as a function rather than as a package data object (which would have been over 10 MB). This function will take you from nothing to having some basic CPS data in your environment, with the option to save this data locally for future ease. A sample of the data that comes out of this function is provided as cpsvote::cps_allyears_100k.

Usage

```
cps_load_basic(
  years = seq(1994, 2024, 2),
  datadir = cps_data_dir(),
  outdir = NULL
)
```

Arguments

years	Which years should be read
datadir	The location where the CPS zip files live (or should be downloaded to). Defaults to cps_data_dir(), which returns ~/cps_data unless overridden via options(cpsvote.datadir = "your/path").
outdir	The location where the final data file should be saved to

Examples

```
## Not run: cps_load_basic(years = 2016, outdir = "data")
```

---

cps_read *Read in CPS data*

Description

Load multiple years of data from the Current Population Survey. This function will also download the data for you, if it is not present in the given dir.

Usage

```
cps_read(
  years = seq(1994, 2024, 2),
  dir = cps_data_dir(),
  cols = cpsvote::cps_cols,
  names_col = "new_name",
  join_dfs = TRUE
)
```

Arguments

years	Which years to read in. Thie function will read data from files in dir whose names contain these 4-digit years.
dir	The folder where the CPS data files live. These files should follow a naming scheme that contains the 4-digit year of the results in question, and have a ".zip" or ".gz" extension. Defaults to cps_data_dir(), which returns ~/cps_data unless overridden via options(cpsvote.datadir = "your/path").

cols	Which columns to read. This must be a data frame, with required columns start_pos,end_pos, and year. The default value is cps_cols, which reads from the list cpsvote::cps_cols. See vignette("add-variables") for details about how to specify a different set of cols.
names_col	The column in cols that contains column names for the specified columns. If none exists, use names_col = NULL
join_dfs	Whether to combine all of the years into a single data frame, or leave them as a list of data frames. Defaults to TRUE with a warning.

Value

a data frame, or list of data frames

Examples

```
## Not run: cps_read(years = 2016, names_col = "new_name")
```

---

cps_read_year *Load a single CPS file*

Description

Read one year of data from the Current Population Survey

Usage

```
cps_read_year(
  file,
  cols = cpsvote::cps_cols,
  names_col = "new_name",
  year = as.numeric(stringr::str_extract(file, "\\d{4}"))
)
```

Arguments

file	Where the fixed-width or zip/gz file for this year's data lives
cols	Which columns to read. This must be a data frame, with required columns start_pos and end_pos. The default value is cps_cols, which reads from the list cpsvote::cps_cols. See vignette("add-variables") for details about how to specify a different set of cols.
names_col	The column in cols that contains column names for the specified columns. If none exists, use names_col = NULL
year	Which year is being read; defaults to 4-digit year in file name

Value

a data frame, with dimensions depending on the year and columns specified

cps_recode_vote *recode the voting variable for turnout calculations*

Description

When the CPS calculates voter turnout, they consider the values "Don't know", "Refused", and "No response" to be non-voters, that is they lump these in with "No". With increased levels of survey non-response in recent years, this has caused turnout estimates to artificially deflate when compared to measures of voter turnout from state election offices. This function adds two recodes of the original voting variable, one which applies the CPS recoding where multiple categories map to "No", and one which follows the guidelines from Hur & Achen (2013) of setting these categories to NA. See the Vignette for more information on this process.

Usage

```
cps_recode_vote(
  data,
  vote_col = "VRS_VOTE",
  items = c("DON'T KNOW", "REFUSED", "NO RESPONSE")
)
```

Arguments

data	the input data set
vote_col	which column contains the voting variable
items	which items should be "No" in the CPS coding and NA in the Hur & Achen coding

Value

data with two columns attached, cps_turnout and hurachen_turnout, voting variables recoded according to the process above

Examples

```
cps_recode_vote(cps_refactor(cps_label(cps_2016_10k)))
```

cps_refactor *combine factor levels across years*

Description

The response sets in certain CPS questions change between years. This function consolidates several of these response sets across years (and fixes typos from the CPS documentation), specifically race, Hispanic status, duration of residency, reason for not voting, and method of registration. Additionally, this creates a new column `VRS_VOTEMETHOD_CON` which consolidates multiple expressions of vote method across years (By Mail, Early, and Election Day) into one variable.

Usage

```
cps_refactor(data, move_levels = TRUE)
```

Arguments

`data`	A dataset containing already-labelled CPS data
`move_levels`	Whether to move the levels "OTHER", "DON'T KNOW", and "REFUSED" to the end of each factor's level set

Details

While consolidating response sets across multiple surveys can be fraught with peril, this function attempts to combine disparate levels for race and other CPS variable across multiple years. Some of these are relatively straightforward typos fixes ("NON-HIPSANIC" should clearly match "NON-HISPANIC"), but others have differing degrees of subjectivity applied. Take this function with a grain of salt, as it depends on some exact variable names you may or may not be using, and recode variables as needed for your own uses. To explore exactly how these variables were recoded, you can run `table(data$RACE, cps_refactor(data)$RACE)` in the console, substituting your column of interest in for `RACE`.

Examples

```
cps_refactor(cps_label(cps_2016_10k))
```

cps_reweight *Calculations to reweight properly for voter turnout*

Description

While the U.S. Census Bureau provides one weight with the CPS, a modified weight is needed to properly calculate voter turnout. This data set provides those calculations, according to Hur and Achen (2013). The comparison data comes from Dr. Michael McDonald's estimates of voter turnout among the voting-eligible population (VEP). It can be joined with CPS data to calculate the new weights needed for analysis, using the function `cps_reweight_turnout`.

Usage

```
cps_reweight
```

Format

A tibble with 1,428 rows and 6 columns:

YEAR year

STATE state

response indicator of turnout in recent election

vep_turnout proportion of turnout indicator, calculated by McDonald

cps_turnout proportion of turnout indicator, calculated by CPS

reweight the factor by which to scale original CPS weights

Source

Turnout data from https://www.electproject.org

---

cps_reweight_turnout *apply weight correction for voter turnout*

Description

This function applies the turnout correction recommended by Hur & Achen (2013). The data set containing the scaling factor is `cpsvote::cps_reweight`.

Usage

```
cps_reweight_turnout(data)
```

Arguments

`data`	the input data set, containing columns `YEAR`, `STATE`, and `hurachen_turnout`

Examples

```
cps_reweight_turnout(cps_recode_vote(cps_refactor(cps_label(cps_2016_10k))))
```

na_ifin *vectorized* na_if

Description

vectorized na_if

Usage

```
na_ifin(x, y)
```

Arguments

x	the vector to be checked
y	the values which should be replaced with NA

Index

∗ **datasets**
    cps_2016_10k, 2
    cps_2020_10k, 3
    cps_2022_10k, 4
    cps_2024_10k, 5
    cps_allyears_100k, 6
    cps_cols, 7
    cps_factors, 11
    cps_reweight, 16

cps_2016_10k, 2
cps_2020_10k, 3
cps_2022_10k, 4
cps_2024_10k, 5
cps_allyears_100k, 6
cps_cols, 7
cps_data_dir, 7
cps_data_dir(), *9*, *13*
cps_docs_dir, 8
cps_docs_dir(), *10*
cps_download_data, 9
cps_download_data(), *7, 8*
cps_download_docs, 10
cps_download_docs(), *8*
cps_factors, 11
cps_label, 11
cps_load_basic, 12
cps_load_basic(), *7, 8*
cps_read, 13
cps_read(), *7, 8*
cps_read_year, 14
cps_recode_vote, 15
cps_refactor, 16
cps_reweight, 16
cps_reweight_turnout, 17

na_ifin, 18