

# Package ‘cpsvote’

November 5, 2020

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

**Version** 0.1.0

**Description** Provides automated methods for downloading, recoding, and merging selected years of the Current Population Survey's Voting and Registration Supplement, 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) <<http://www.electproject.org/home/voter-turnout/voter-turnout-data>>.

**URL** <https://github.com/Reed-EVIC/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, rmarkdown, survey, srvyr, here, scales, ggplot2, usmap

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**Imports** magrittr, readr, dplyr, stringr, forcats, rlang

**NeedsCompilation** no

**Author** Jay Lee [aut, cre],  
Paul Gronke [aut],  
Canyon Foot [ctb]

**Maintainer** Jay Lee <jaylee@reed.edu>

**Repository** CRAN

**Date/Publication** 2020-11-05 16:00:02 UTC

## R topics documented:

cps_2016_10k . . . . .	2
cps_allyears_10k . . . . .	3
cps_cols . . . . .	4
cps_download_data . . . . .	5
cps_download_docs . . . . .	6
cps_factors . . . . .	7
cps_label . . . . .	7
cps_load_basic . . . . .	8
cps_read . . . . .	9
cps_read_year . . . . .	10
cps_recode_vote . . . . .	11
cps_refactor . . . . .	12
cps_reweight . . . . .	12
cps_reweight_turnout . . . . .	13
na_ifin . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

cps_2016_10k	<i>A sample of the raw 2016 CPS dataset</i>
--------------	---

---

### 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\_allyears\_10k      *A sample of the full CPS dataset*

---

### Description

This is a 10,000 row sample of the data that comes out of cpsvote::cps\_load\_basic.

### Usage

cps\_allyears\_10k

### Format

A tibble with 10,000 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\_2004toPRESENT 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 204 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_download_data	<i>Download CPS microdata</i>
-------------------	-------------------------------

---

## Description

Download CPS microdata

## Usage

```
cps_download_data(  
  path = "cps_data",  
  years = seq(1994, 2018, 2),  
  overwrite = FALSE  
)
```

## Arguments

path	A file path (relative or absolute) where the downloads should go.
years	Which years of data to download. Defaults to all even-numbered years from 1994 to 2018.
overwrite	Logical, whether to write over existing files or not. Defaults to FALSE.

## Details

- File names will be written in the style "cps\_nov2018.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(path = "cps_docs", years = 2016, overwrite = TRUE)  
  
## End(Not run)
```

---

cps_download_docs	<i>Download CPS technical documentation</i>
-------------------	---

---

## Description

Download CPS technical documentation

## Usage

```
cps_download_docs(  
  path = "cps_docs",  
  years = seq(1994, 2018, 2),  
  overwrite = FALSE  
)
```

## Arguments

path	A file path (relative or absolute) where the downloads should go.
years	Which years of documentation to download. Defaults to all even-numbered years from 1994 to 2018.
overwrite	Logical, whether to write over existing files or not. Defaults to FALSE.

## Details

- File names will be written in the style "cps\_nov2018.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(path = "cps_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 204 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

**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	<i>load some basic/default CPS data into the environment</i>
----------------	--

---

**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_10k`.



**Usage**

```
cps_load_basic(years = seq(1994, 2018, 2), datadir = "cps_data", outdir = NULL)
```

**Arguments**

years	Which years should be read
datadir	The location where the CPS zip files live (or should be downloaded to)
outdir	The location where the final data file should be saved to

**Examples**

```
## Not run: cps_load-basic(years = 2016, outdir = "data")
```

---

cps_read	<i>Read in CPS data</i>
----------	-------------------------

---

**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, 2018, 2),
  dir = "cps_data",
  cols = cpsvote::cps_cols,
  names_col = "new_name",
  join_dfs = TRUE
)
```

**Arguments**

years	Which years to read in. This 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.
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	<i>Load a single CPS file</i>
---------------	-------------------------------

---

**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 <code>start_pos</code> and <code>end_pos</code> . The default value is <code>cps_cols</code> , which reads from the list <code>cpsvote::cps_cols</code> . See <code>vignette("add-variables")</code> for details about how to specify a different set of cols.
names_col	The column in <code>cols</code> that contains column names for the specified columns. If none exists, use <code>names_col = NULL</code>
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	<i>recode the voting variable for turnout calculations</i>
-----------------	--

---

## 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	<i>combine factor levels across years</i>
--------------	---

---

### 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-HISPANIC" 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	<i>Calculations to reweight properly for voter turnout</i>
--------------	--

---

### 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 1,326 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 <http://www.electproject.org/home/voter-turnout/voter-turnout-data>

---

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	<i>vectorized</i> 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\\_allyears\\_10k](#), 3
- [cps\\_cols](#), 4
- [cps\\_factors](#), 7
- [cps\\_reweight](#), 12

- [cps\\_2016\\_10k](#), 2
- [cps\\_allyears\\_10k](#), 3
- [cps\\_cols](#), 4
- [cps\\_download\\_data](#), 5
- [cps\\_download\\_docs](#), 6
- [cps\\_factors](#), 7
- [cps\\_label](#), 7
- [cps\\_load\\_basic](#), 8
- [cps\\_read](#), 9
- [cps\\_read\\_year](#), 10
- [cps\\_recode\\_vote](#), 11
- [cps\\_refactor](#), 12
- [cps\\_reweight](#), 12
- [cps\\_reweight\\_turnout](#), 13

- [na\\_ifin](#), 14