# Package 'PL94171'

February 19, 2025

**Type** Package

**Title** Tabulate P.L. 94-171 Redistricting Data Summary Files

**Version** 1.1.3

**Maintainer** Cory McCartan <mccartan@psu.edu>

**Description** Tools to process legacy format summary redistricting data files produced by the United States Census Bureau pursuant to P.L. 94-171. These files are generally available earlier but are difficult to work with as-is.

**Depends** R (>= 4.0.0)

**Imports** cli, stringr, readr, dplyr (>= 1.0.0), tinytiger, sf, withr, httr

**Suggests** testthat (>= 3.0.0), lifecycle, knitr, rmarkdown, ggplot2

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**URL** https://corymccartan.com/PL94171/, https://github.com/CoryMcCartan/PL94171/

**BugReports** https://github.com/CoryMcCartan/PL94171/issues

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Language** en-US

**NeedsCompilation** no

**Author** Cory McCartan [aut, cre], Christopher T. Kenny [aut]

**Repository** CRAN

**Date/Publication** 2025-02-19 05:30:06 UTC

# Contents



---

`pl_crosswalk` *Download Block Crosswalk Files*

---

## Description

Downloads crosswalks from https://www.census.gov/geographies/reference-files/time-series/geo/relationship-files.html. Adjusts land overlap area to ensure weights sum to 1.

## Usage

```
pl_crosswalk(abbr, from_year = 2010L, to_year = from_year + 10L)
```

## Arguments

| | |
|---|---|
| `abbr` | the state to download the crosswalk for. |
| `from_year` | the year with the blocks that the data is currently tabulated with respect to. |
| `to_year` | the year with the blocks that the data should be tabulated into. |

## Value

A tibble, with two sets of GEOIDs and overlap information.

## Examples

```
## Not run:
# Takes a bit of time to run
pl_crosswalk("RI", 2010, 2020)

## End(Not run)
```

## pl_ex *PL Example File*

### Description

This data contains a subset of the 2020 prototype PL data

### Usage

```
data("pl_ex")
```

### Format

list of tibbles containing the four PL files.

`00001` Tables P1 and P2

`00002` Tables P3, P4, and H1

`00003` Table P5

`geo` geographic header file

### Examples

```
data(pl_ex)
```

## pl_geog_levels *List of Summary Levels and Official Descriptions*

### Description

This dataset is tibble version of the descriptions of (potentially) available summary levels within the P.L. 94-171 data, as described in the 2018 Redistricting Data Prototype (Public Law 94-171) Summary File documentation.

### Usage

```
pl_geog_levels
```

### Format

a tibble with two columns:

**SUMLEV** The three character summary level code

**SUMLEV_description** The summary level description

## pl_get_baf *Download 2020 Block Assignment Files for a State*

### Description

**[Experimental]** From the Census: "The Block Assignment Files (BAFs) are among the geographic products that the Census Bureau provides to states and other data users containing the small area census data necessary for legislative redistricting. The BAFs contain Census tabulation block codes and geographic area codes for a specific geographic entity type."

### Usage

```
pl_get_baf(abbr, geographies = NULL, cache_to = NULL, refresh = FALSE)
```

### Arguments

| | |
|---|---|
| abbr | the state abbreviation to get the BAF for |
| geographies | the geographies to get. Defaults to all available. |
| cache_to | the file name, if any, to cache the results to (as an RDS). If a file exists and refresh=FALSE, will read BAF from this file. |
| refresh | if TRUE, force a re-download of the BAF data. |

### Value

A list of data frames, one for each available BAF geography.

### Examples

```
pl_get_baf("RI")
pl_get_baf("RI", "VTD")
```

## pl_get_prototype *Download TIGER Prototype shapefiles*

### Description

**[Experimental]** These prototype shapefiles correspond to the Rhode Island end-to-end Census test and the accompanying prototype P.L. 94-171 data. This function is unlikely to be useful for working with any actual decennial Census data. The corresponding tinytiger or tigris functions should be used instead.

## Usage

```
pl_get_prototype(
  geog,
  year = 2020,
  full_state = TRUE,
  cache_to = NULL,
  clean_names = TRUE,
  refresh = FALSE
)
```

## Arguments

| | |
|---|---|
| geog | Geography to download data for. See details for full list. |
| year | year, either 2010 or 2020 |
| full_state | whether to return the full state (TRUE) or the single county subset (FALSE) |
| cache_to | the file name, if any, to cache the results to (as an RDS). If a file exists and refresh=FALSE, will read from this file. |
| clean_names | whether to clean and rename columns |
| refresh | if TRUE, force a re-download of the data. |

## Details

Current acceptable arguments to geog include:

- block: block
- block_group: block group
- tract: tract
- county: county
- state: state
- sld_low: state legislative district lower house
- sld_up: state legislative district upper house
- congressional_district: federal congressional district
- place: Census place
- voting_district: voting tabulation district

## Value

An sf object containing the blocks.

## Examples

```
shp <- pl_get_prototype("block")
```

## pl_get_vtd *Download 2020 Voting District Shapefiles*

### Description

**[Experimental]** A (likely temporary) function to download TIGER shapefiles for 2020 voting tabulation districts (VTDs).

### Usage

```
pl_get_vtd(abbr, cache_to = NULL, refresh = FALSE)
```

### Arguments

| | |
|---|---|
| abbr | Geography to download data for. See details for full list. |
| cache_to | the file name, if any, to cache the results to (as an RDS). If a file exists and refresh=FALSE, will read from this file. |
| refresh | if TRUE, force a re-download of the data. |

### Value

An sf object containing the VTDs.

### Examples

```
shp <- pl_get_vtd("RI")
```

## pl_read *Read a set of PL Files*

### Description

PL files come in one of four types and are pipe-delimited with no header row. This function speedily reads in the files and assigns the appropriate column names and types.

### Usage

```
pl_read(path, ...)

read_pl(path, ...)
```

### Arguments

| | |
|---|---|
| path | a path to a folder containing PL files. Can also be path or a URL for a ZIP file, which will be downloaded and unzipped. |
| ... | passed on to readr::read_delim() |

### Value

A list of data frames containing the four PL files.

### Examples

```
pl_ex_path <- system.file('extdata/ri2018_2020Style.pl', package = 'PL94171')
pl <- pl_read(pl_ex_path)

# or try `pl_read(pl_url("RI", 2010))`
```

---

## pl_retally *Approximately re-tally Census data under new block boundaries*

---

### Description

Applies a block crosswalk to a table of block data using areal interpolation. That is, the fraction of land area in the overlapping region between old and new blocks is used to divide the population of the old blocks into the new.

### Usage

```
pl_retally(d_from, crosswalk)
```

### Arguments

| | |
|---|---|
| d_from | The data frame to process. All numeric columns will be re-tallied. Integer columns will be re-tallied with rounding. Character columns will be preserved if constant across new block geometries. |
| crosswalk | The crosswalk data frame, from pl_crosswalk() |

### Details

All numeric columns will be re-tallied. Integer columns will be re-tallied with rounding. Character columns will be preserved if constant across new block geometries.

Blocks from other states will be ignored.

### Value

A new data frame, like d_from, except with the geometry column dropped, if one exists. New geometry should be loaded, perhaps with tinytiger::tt_blocks().

## Examples

```
crosswalk = pl_crosswalk("RI", 2010, 2020)
RI_2010 = pl_tidy_shp("RI", pl_url("RI", 2010), 2010)
pl_retally(RI_2010, crosswalk)
```

---

`pl_select_standard` *Select the Standard Redistricting Columns*

---

## Description

Selects the standard set of basic population groups and VAP groups. Optionally renames them from the PXXXYYYY naming convention (where XXX is the table and YYYY is the variable) to more human readable names. pop_* is the total population, from tables 1 and 2, while vap_* is the 18+ population (voting age population).

## Usage

```
pl_select_standard(pl, clean_names = TRUE)
```

## Arguments

| | |
|---|---|
| `pl` | A list of PL tables, as read in by `pl_read()` |
| `clean_names` | whether to clean names |

## Details

If `clean names=TRUE`, then the variables extracted are as follows:

- `\*_hisp`: Hispanic or Latino (of any race)
- `\*_white`: White alone, not Hispanic or Latino
- `\*_black`: Black or African American alone, not Hispanic or Latino
- `\*_aian`: American Indian and Alaska Native alone, not Hispanic or Latino
- `\*_asian`: Asian alone, not Hispanic or Latino
- `\*_nhpi`: Native Hawaiian and Other Pacific Islander alone, not Hispanic or Latino
- `\*_other`: Some Other Race alone, not Hispanic or Latino
- `\*_two`: Population of two or more races, not Hispanic or Latino

where * is pop or vap.

## Value

A data frame with the selected and optionally renamed columns

## Examples

```
pl_ex_path <- system.file('extdata/ri2018_2020Style.pl', package = 'PL94171')
pl <- pl_read(pl_ex_path)
pl <- pl_select_standard(pl)
```

---

`pl_subset` *Subset to a Summary Level*

---

## Description

This subsets a pl table to a desired summary level. Typical choices include:

- '750': block
- '150': block group
- '630': voting district
- '050': county

## Usage

```
pl_subset(pl, sumlev = "750")
```

## Arguments

| | |
|---|---|
| `pl` | A list of PL tables, as read in by `pl_read()` |
| `sumlev` | the summary level to filter to. A 3 character SUMLEV code. Default is '750' for blocks. |

## Details

All summary levels are listed in pl_geog_levels.

## Value

A data frame

## Examples

```
pl_ex_path <- system.file('extdata/ri2018_2020Style.pl', package = 'PL94171')
pl <- pl_read(pl_ex_path)
pl <- pl_subset(pl)
```

## pl_tidy_shp *All-in-one Shapefile Function*

### Description

Downloads block geography and merges with the cleaned PL 94-171 file.

### Usage

```
pl_tidy_shp(abbr, path, year = 2020, type = c("blocks", "vtds"), ...)
```

### Arguments

| | |
|---|---|
| abbr | The state to make the shapefile for |
| path | The path to the PL files, as in pl_read() |
| year | The year to download the block geography for. Should match the year of the PL files. |
| type | If "blocks", make a Census block shapefile; if "vtds" make a VTD shapefile. |
| ... | passed on to dplyr::filter(); use to subset to a certain county, for example. |

### Value

an sf object with demographic and shapefile information for the state.

### Examples

```
pl_ex_path <- system.file("extdata/ri2018_2020Style.pl", package = "PL94171")
pl_tidy_shp("RI", pl_ex_path)
```

## pl_url *Get the URL for PL files for a particular state and year*

### Description

Get the URL for PL files for a particular state and year

### Usage

```
pl_url(abbr, year = 2010)
```

## Arguments

| | |
|---|---|
| abbr | The state to download the PL files for |
| year | The year of PL file to download. Supported years: 2000, 2010, 2020 (after release). 2000 files are in a different format. Earlier years available on tape or CD-ROM only. |

## Value

a character vector containing the URL to a ZIP containing the PL files.

## Examples

```
pl_url("RI", 2010)
```

# Index