Package: findSVI (via r-universe)

November 22, 2026

Title Calculate Social Vulnerability Index for Communities

Version 0.2.1

Description Developed by CDC/ATSDR (Centers for Disease Control and Prevention/ Agency for Toxic Substances and Disease Registry), Social Vulnerability Index (SVI) serves as a tool to assess the resilience of communities by taking into account socioeconomic and demographic factors. Provided with year(s), region(s) and a geographic level of interest, 'findSVI' retrieves required variables from US census data and calculates SVI for communities in the specified area based on CDC/ATSDR SVI documentation. Reference for the calculation methods: Flanagan BE, Gregory EW, Hallisey EJ, Heitgerd JL, Lewis B (2011) <doi:10.2202/1547-7355.1792>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

URL https://github.com/heli-xu/findSVI, https://heli-xu.github.io/findSVI/

BugReports https://github.com/heli-xu/findSVI/issues

Depends R (>= 3.5.0)

Imports cli, dplyr, magrittr, purrr, stringr, tidycensus, tidyr, tidyselect, rlang, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, leaflet, htmltools, glue, sf, ggplot2, reactable, cowplot

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Heli Xu [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-9792-2727>), Ran Li [aut] (ORCID: <https://orcid.org/0000-0002-4699-4755>), Usama Bilal [aut] (ORCID: <https://orcid.org/0000-0002-9868-7773>)

Maintainer Heli Xu <xuheli91@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-22 22:35:01 UTC

RemoteUrl https://github.com/cran/findSVI

RemoteRef HEAD

RemoteSha 08e1e5e5e91d6d5941f5f2c14078e7385e294015

Contents

census_variables 2
cty_cz_2020_xwalk 4
find_svi 5
find_svi_x 6
get_census_data 8
get_svi 9
get_svi_x 10
state_valid 11
variable_calculation 12
zcta_state_xwalk 14

Index **16**

`census_variables` *List of census variables for SVI calculation*

Description

Each of these datasets contains a list of census variable names for a year between 2012-2022.

Usage

```
census_variables_2012

census_variables_2013

census_variables_2014

census_variables_2015

census_variables_2016

census_variables_2017

census_variables_2018

census_variables_2019
```

census_variables_2020

census_variables_2021

census_variables_2022

census_variables_exp_2012

census_variables_exp_2013

census_variables_exp_2014

census_variables_exp_2015

census_variables_exp_2016

census_variables_exp_2017

census_variables_exp_2018

census_variables_exp_2019

census_variables_exp_2020

census_variables_exp_2021

census_variables_exp_2022

Format

a list of census variables, grouped into sublists named t0-t5 or:

t0-t4 (for 2012). t1-t4 represent the 4 themes the corresponding SVI variables are categorized into:

- Socioeconomic
- Household Composition/Disability
- Minority Status/Language
- Housing Type/Transportation

t0 represents 3 census variables of total counts, and their corresponding SVI variables are not categorized into any theme. t5 contains the census variables for SVI "adjunct variables", which are included for reference but not used in SVI calculation. For 2012, adjunct variables are not included, as the variable listed in 2014 documentation was not in 2012 Census data, and there's no adjunct variables in 2010 documentation.

Datasets starting with `census_variable_` contains variables corresponding to `variable_e_ep_calculation_` series of tables, where "EP_" (percent) variables are retrieved directly from Census when available; Datasets starting with `census_variables_exp_` contains variables corresponding to `variable_cal_exp_` tables, where denominators for "EP_" variables are explicitly defined using census variables (adjunct variables are not modified).

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 5.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

An object of class `list` of length 6.

Source

CDC/ATSDR SVI Documentation https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.html

---

`cty_cz_2020_xwalk` *Relationship file (crosswalk) between US counties and commuting zones*

Description

This table contains GEOIDs for US counties and the commuting zones they are nested in. Commuting zones can be used to study regional economy with considerations of urban-rural interconnections across state lines. For details refer to papers by Fowler Jensen and Rhubart (2016) doi:10.1007/s11113-016-9386-0 and Fowler (2024) doi:10.17605/OSF.IO/J256U.

Usage

```
cty_cz_2020_xwalk
```

Format

A data frame with 2 columns and 3222 rows::

GEOID US county FIPS code.

GEOID2 Commuting zone ID for the year 2020.

Source

https://sites.psu.edu/psucz/data/

`find_svi` *Retrieve census data and calculate SVI for one or multiple year(s)/state(s)*

Description

`find_svi()` is like a wrapper for `get_census_data()` and `get_svi()` that retrieves census data and produces SVI for one or multiple years(s) and state(s). For multiple year-state entries, SVI is obtained from percentile rankings for each entry and summarised into one table. Note that a Census API key is required for this function to work, which can be obtained at https://api.census.gov/data/key_signup.html and set up using `tidycensus::census_api_key()`.

Usage

```
find_svi(year, state = NULL, geography, key = NULL, full.table = FALSE)
```

Arguments

`year`	A vector containing years of interest (available 2012-2022). Length >=1. Acting as pairs with `state`, `year` should be of the same length as `state`. The exception is when it's a single year entry (length 1), other than providing one state of interest, supply `state = NULL` as default or `state = 'US'` retrieves and processes nation level data to obtain SVI.
`state`	A vector containing states of interest. Length >=0. Length 0 (`state = NULL`), or `state = 'US'` must be used with single year argument, when SVI is calculated from nation-level census data. In other cases, `state` must have the same elements as `year` (same length).
`geography`	One geography level of interest for all year-state combination (e.g."county", "zcta", "tract").
`key`	Your Census API key. Obtain one at https://api.census.gov/data/key_signup.html. To set up, use `tidycensus::census_api_key("YOUR KEY GOES HERE")`, or include it as an argument.
`full.table`	Default as `FALSE`, returning SVI table with only "GEOID", and SVI for each theme and all themes. If set as `TRUE`, a full SVI table with individual SVI variables and intermediate ranking calculations are also included in addition to the theme-related SVIs (similar style to tables from CDC/ATSDR database(https://www.atsdr.cdc.gov/placear

Value

A tibble of summarised SVI for one or multiple year-state combination(s) of interest. Rows represent the geographic units, and columns represent its SVI for each theme and all themes. Additional two columns at the end indicate the corresponding state and year information. For `full.table = TRUE`, estimated count and percentage values for individual SVI variables are also included. For description of variable names (column names), please refer to CDC/ATSDR documentation (https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.html).

Examples

```
# Census API key required
# For one year-state entry
find_svi(
      year = 2019,
      state = "AZ",
      geography = "county"
   )


# For multiple year-state pairs
## All ZCTAs for 2017-AZ; 2017-DE; and 2018-DC
year <- c(2017, 2017, 2018)
state <- c("AZ", "DE", "DC")
info <- data.frame(year, state)

find_svi(
      year = info$year,
      state = info$state,
      geography = "zcta"
   )
```

`find_svi_x` *Retrieve census data and calculate SVI for a customized geographic level*

Description

`find_svi_x()` is like a wrapper for `get_census_data()` and `get_svi_x()` that retrieves census data and produces SVI for a customized geographic level consisted of a Census geography. The census data is retrieved at the Census geographic level, and estimate counts are summed across the customized geographic level to calculate SVI. Note that a Census API key is required for this function to work, which can be obtained at https://api.census.gov/data/key_signup.html and set up using `tidycensus::census_api_key()`.

Usage

```
find_svi_x(
  year,
  geography,
  state = NULL,
  county = NULL,
  key = NULL,
  geometry = FALSE,
  xwalk
)
```

Arguments

year	A year of interest (available 2012-2022).
geography	The Census geography level of interest (e.g."county", "zcta", "tract").
state	(Optional) Specify the state of interest. Default `state = NULL` or `state = 'US'` return nation-level data.
county	(Optional) Specify the county of interest, must be combined with a value supplied to "state".
key	Your Census API key. Obtain one at https://api.census.gov/data/key_signup.html. To set up, use `tidycensus::census_api_key("YOUR KEY GOES HERE")`, or include it as an argument.
geometry	Default as `FALSE` for retrieving census data without spatial information. Set as `TRUE` for retrieving census data with an additional `geometry` column containing simple feature geometry.
xwalk	A crosswalk (relationship table) between the Census geographic level and the customized geographic level of interest. A crosswalk between US counties and commuting zones `cty_cz_2020_xwalk` is included as an example, and please set the column names of the crosswalk as follows: **GEOID** Identifiers for the Census geographic level. Must contain values from `GEOID` column in `data`, and be in a compatible data type (character). **GEOID2** Identifiers (characters or numeric values) for the customized geographic level that is larger geographic than the Census geographic level. The Census geographic level should be nested in the customized geographic level. **NAME** An optional column of the names or description of the customized geographic level.

Value

A tibble of SVI with rows representing the customized geographic units (with a column name of `GEOID`), and columns indicating variable names (first two columns containing geographic information). For detailed description of the variable names (column names), please refer to CDC/ATSDR documentation (https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.html).

See Also

find_svi() for retrieving census data and calculating SVI for multiple year-state pairs at a Census geographic level. get_census_data() (with exp = TRUE) and get_svi_x() for separate functions for data retrieval and SVI calculation at a customized geographic level.

Examples

```
# Census API key required

  find_svi_x(
    year = 2020,
    geography = "county",
    xwalk = cty_cz_2020_xwalk
    )
```

---

get_census_data *Retrieve American Community Survey data for SVI variables*

Description

This function uses tidycensus::get_acs() with a pre-defined list of variables to retrieves ACS data for SVI calculation. Note that a Census API key is required for this function to work, which can be obtained at https://api.census.gov/data/key_signup.html and set up using tidycensus::census_api_key().

Usage

```
get_census_data(
  year,
  geography,
  state = NULL,
  county = NULL,
  key = NULL,
  geometry = FALSE,
  exp = FALSE,
  ...
)
```

Arguments

year	The year of interest (available 2012-2022).
geography	The geography of interest (eg. state, county, zcta, tract)
state	(Optional) Specify the state of interest. If data for multiple states are retrieved together, ranking for SVI calculation will be performed among all states. state = NULL as default, or state = 'US' return nation-level data.
county	(Optional) Specify the county(s) of interest, must be combined with a value supplied to "state".

key	Your Census API key. Obtain one at https://api.census.gov/data/key_signup.html. Include it in this argument or set up your key using tidycensus::census_api_key("YOUR KEY GOES HERE").
geometry	Default as FALSE for a regular tibble of census data. If set as TRUE, returns a tibble with an additional geometry column containing simple feature geometry.
exp	Default as FALSE for retrieving EP_(percent estimate) variables directly from ACS when available (as described in CDC/ADSTR SVI dictionary (https://www.atsdr.cdc.gov/placeandhe If set as TRUE, uses explicitly defined denominators and retrieves the ACS data to calculate the EP variables for easier aggregation in downstream analysis (e.g. get_svi_x()).
...	Other arguments; more details please see tidycensus::get_acs()

Value

A tibble of ACS data with each row represents an enumeration (geographic) unit and each column represents a census variable ("wide" form).

Examples

```
# Census API key required
 get_census_data(
    year = 2018,
    geography = "county",
    state = "PA"
  )
```

---

get_svi *Calculate SVI for communities in a region from census data*

Description

get_svi() calculates and constructs an SVI table for a geographic level of interest based on CDC/ATSDR SVI documentation (https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.html). Briefly, by taking into account 4 themes of census variables that represent challenges in socioeconomic status, household characteristics, racial and ethnic minority status and housing/transportation, SVI uses percentile ranking within a region to indicate the relative social vulnerability of the geographic units (communities) in that region.

Usage

```
get_svi(year, data)
```

Arguments

year	The year of interest (available 2012-2021), must match the year specified in retrieving census data.
data	The census data retrieved by get_census_data().

Value

A tibble of SVI with rows representing geographic units, and columns indicating variable names (first two columns containing geographic information). For detailed description of the variable names (column names), please refer to CDC/ATSDR documentation.

Examples

```
# Census API key required
 pa2018 <- get_census_data(
    year = 2018,
    geography = "county",
    state = "PA")

 get_svi(2018, pa2018)
```

---

get_svi_x *Calculate SVI for communities from census data using customized boundaries*

Description

`get_svi_x()` calculates and constructs an SVI table for a *customized* geographic level of interest based on CDC/ATSDR SVI documentation (https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.ht
By supplying a crosswalk (relationship table) between a Census geographic level and a customized geographic level, census data are summed across the customized geographic units, and SVI is calculated accordingly to indicate the relative social vulnerability of the geographic units (communities).

Usage

```
get_svi_x(year, data, xwalk)
```

Arguments

`year`	The year of interest (available 2012-2021), must match the year specified in retrieving census data.
`data`	The census data retrieved by `get_census_data()`.
`xwalk`	A crosswalk (relationship table) between the Census geographic level and the customized geographic level of interest. A crosswalk between US counties and commuting zones `cty_cz_2020_xwalk` is included as an example, and please set the column names of the crosswalk as follows:

GEOID Identifiers for the Census geographic level. Must contain values from `GEOID` column in `data`, and be in a compatible data type (character).

GEOID2 Identifiers (characters or numeric values) for the customized geographic level that is larger geographic than the Census geographic level. The Census geographic level should be nested in the customized geographic level.

NAME An optional column of the names or description of the customized geographic level.

Value

A tibble of SVI with rows representing the customized geographic units (with a column name of `GEOID`), and columns indicating variable names (first two columns containing geographic information). For detailed description of the variable names (column names), please refer to CDC/ATSDR documentation.

See Also

`get_svi()` for SVI calculation from census data at a Census geographic level, and `find_svi()` for retrieving census data and calculating SVI for multiple year-state pairs.

Examples

```
# Census API key required
cty2020 <- get_census_data(
    year = 2020,
    geography = "county",
    exp = TRUE
   )

get_svi_x(year = 2020, data = cty2020, xwalk = cty_cz_2020_xwalk)
```

---

state_valid *Table of valid full names/abbreviations/FIPS codes of 52 states*

Description

A reference table for valid input for `state` in `get_census_data()` and `find_svi()`. In addition, `state = "US"` or `state = NULL` is also accepted for nation-level data.

Usage

```
state_valid
```

Format

A data frame with 3 columns and 52 rows::

st_abbr 2-letter abbreviation for states.

state State full name.

fips_code Federal Information Processing System (FIPS) Codes for states.

Source

County-state reference file 2020 https://www.census.gov/programs-surveys/popest/geographies/reference-files.html

variable_calculation *Table of census variables and formula for SVI calculation*

Description

Each of these datasets contains a table of SVI variable names, related census variable names and their corresponding calculation formula for a year between 2012-2022. This is used to construct SVI results for the variables starting with "E_"(estimate) and "EP_"(percentage) after obtaining census data. Sometimes SVI variables are directly linked to census variables, and other times one or more census variable(s) are included to derive an SVI variable. Two series of calculation tables are included with different approaches to construct EP_ variables. variable_cal_exp_ series of dataset uses census variables explicitly as denominators, whereas variable_e_ep_calculation_ series of dataset retrieves percent from ACS when available (as described by CDC SVI documentation). Variables in theme 5 (adjunct variables) follow CDC SVI documentation and remain the same across both series of tables.

Usage

```
variable_e_ep_calculation_2012

variable_e_ep_calculation_2013

variable_e_ep_calculation_2014

variable_e_ep_calculation_2015

variable_e_ep_calculation_2016

variable_e_ep_calculation_2017

variable_e_ep_calculation_2018

variable_e_ep_calculation_2019

variable_e_ep_calculation_2020

variable_e_ep_calculation_2021

variable_e_ep_calculation_2022

variable_cal_exp_2012

variable_cal_exp_2013

variable_cal_exp_2014

variable_cal_exp_2015
```

variable_cal_exp_2016

variable_cal_exp_2017

variable_cal_exp_2018

variable_cal_exp_2019

variable_cal_exp_2020

variable_cal_exp_2021

variable_cal_exp_2022

Format

A data frame with 3 columns and various number of rows, depending:

on the year:

_variable_name With a prefix "x" followed by the year, eg. x2018_variable_name, this column is the SVI variable name

theme SVI variables are categorized into four themes/domains: socioeconomic, household composition/disability, minority status/language and housing type/transportation. Theme 0 is used for 3 variables representing total counts, while theme 5 is used for adjunct variables (not included in calculation). Adjunct variables are not included in 2012 due to unavailable data/documentation.

_table_field_calculation With a prefix "x" followed by the year, eg. x2018_table_field_calculation, this column contains the corresponding census variable names, and/or the calculation using SVI/census variables.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 51 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 51 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 51 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 51 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 33 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 35 rows and 3 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 35 rows and 3 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 35 rows and 3 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 35 rows and 3 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 51 rows and 3 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 51 rows and 3 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 51 rows and 3 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 51 rows and 3 columns.

Source

CDC/ATSDR SVI Documentation https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.html

---

zcta_state_xwalk *Relationship file (crosswalk) for ZCTAs by state*

Description

Each of these tables contains ZIP Code Tabulation Areas (ZCTAs), their intersecting counties and the states (state name, abbreviation, state FIPS code) they are nested in. It's used in get_census_data() for retrieving ZCTA-level census data by state, as tidycensus::get_acs() (CRAN version) currently does not support obtaining state-specific ZCTA-level data.

Usage

```
zcta_state_xwalk2021

zcta_state_xwalk2020

zcta_state_xwalk2019

zcta_state_xwalk2022
```

Format

A data frame with 5 columns and various number of rows, depending on the year::

ZCTA 5 digit ZCTA code.

st_code Federal Information Processing System (FIPS) Codes for States.

county County name within the state that the ZCTA intersects/corresponds to.

state State full name corresponding to the FIPS code.

st_abb Two-letter state abbreviation.

An object of class tbl_df (inherits from tbl, data.frame) with 538426 rows and 5 columns.

An object of class data.frame with 519726 rows and 5 columns.

An object of class tbl_df (inherits from tbl, data.frame) with 538152 rows and 5 columns.

Source

Census ZCTA-county relationship file (2010) https://www.census.gov/geographies/reference-files/time-series/geo/relationship-files.2010.html#list-tab-1709067297 Geocorr ZCTA-county relationship file (2020) https://mcdc.missouri.edu/applications/geocorr2022.html County-state reference file (2019, 2020, 2021, 2022) https://www.census.gov/programs-surveys/popest/geographies/reference-files.html

Index

∗ **datasets**
census_variables, 2
cty_cz_2020_xwalk, 4
state_valid, 11
variable_calculation, 12
zcta_state_xwalk, 14

census_variables, 2
census_variables_2012 (census_variables), 2
census_variables_2013 (census_variables), 2
census_variables_2014 (census_variables), 2
census_variables_2015 (census_variables), 2
census_variables_2016 (census_variables), 2
census_variables_2017 (census_variables), 2
census_variables_2018 (census_variables), 2
census_variables_2019 (census_variables), 2
census_variables_2020 (census_variables), 2
census_variables_2021 (census_variables), 2
census_variables_2022 (census_variables), 2
census_variables_exp_2012 (census_variables), 2
census_variables_exp_2013 (census_variables), 2
census_variables_exp_2014 (census_variables), 2
census_variables_exp_2015 (census_variables), 2
census_variables_exp_2016 (census_variables), 2
census_variables_exp_2017 (census_variables), 2
census_variables_exp_2018 (census_variables), 2
census_variables_exp_2019 (census_variables), 2
census_variables_exp_2020 (census_variables), 2
census_variables_exp_2021 (census_variables), 2
census_variables_exp_2022 (census_variables), 2
cty_cz_2020_xwalk, 4

find_svi, 5
find_svi(), *8*, *11*
find_svi_x, 6

get_census_data, 8
get_census_data(), *5*, *6*, *8*
get_svi, 9
get_svi(), *5*, *11*
get_svi_x, 10
get_svi_x(), *6*, *8*

state_valid, 11

tidycensus::census_api_key(), *5*, *6*, *8*
tidycensus::get_acs(), *8*, *9*, *14*

variable_cal_exp_2012 (variable_calculation), 12
variable_cal_exp_2013 (variable_calculation), 12
variable_cal_exp_2014 (variable_calculation), 12
variable_cal_exp_2015 (variable_calculation), 12
variable_cal_exp_2016 (variable_calculation), 12

variable_cal_exp_2017
(variable_calculation), 12
variable_cal_exp_2018
(variable_calculation), 12
variable_cal_exp_2019
(variable_calculation), 12
variable_cal_exp_2020
(variable_calculation), 12
variable_cal_exp_2021
(variable_calculation), 12
variable_cal_exp_2022
(variable_calculation), 12
variable_calculation, 12
variable_e_ep_calculation_2012
(variable_calculation), 12
variable_e_ep_calculation_2013
(variable_calculation), 12
variable_e_ep_calculation_2014
(variable_calculation), 12
variable_e_ep_calculation_2015
(variable_calculation), 12
variable_e_ep_calculation_2016
(variable_calculation), 12
variable_e_ep_calculation_2017
(variable_calculation), 12
variable_e_ep_calculation_2018
(variable_calculation), 12
variable_e_ep_calculation_2019
(variable_calculation), 12
variable_e_ep_calculation_2020
(variable_calculation), 12
variable_e_ep_calculation_2021
(variable_calculation), 12
variable_e_ep_calculation_2022
(variable_calculation), 12

zcta_state_xwalk, 14
zcta_state_xwalk2019
(zcta_state_xwalk), 14
zcta_state_xwalk2020
(zcta_state_xwalk), 14
zcta_state_xwalk2021
(zcta_state_xwalk), 14
zcta_state_xwalk2022
(zcta_state_xwalk), 14