

# Package 'RcensusPkg'

May 7, 2026

**Type** Package

**Title** Easily Access US Census Bureau Survey and Geographic Data

**Version** 0.1.5

**Description** The key function 'get\_vintage\_data()' returns a dataframe and is the window into the Census Bureau API requiring just a dataset name, vintage(year), and vector of variable names for survey estimates/percentages. Other functions assist in searching for available datasets, geographies, group/variable concepts of interest. Also provided are functions to access and layer (via standard piping) displayable geometries for the US, states, counties, blocks/tracts, roads, landmarks, places, and bodies of water. Joining survey data with many of the geometry functions is built-in to produce choropleth maps.

**License** MIT + file LICENSE

**URL** <https://github.com/deandev1/RcensusPkg>

**BugReports** <https://github.com/deandev1/RcensusPkg/issues>

**Depends** R (>= 4.4.0)

**Imports** ggplot2 (>= 3.5.1), data.table (>= 1.16.4), httr2 (>= 1.1.2), downloader (>= 0.4.1), jsonlite (>= 1.8.9), stringr (>= 1.5.1), purrr (>= 1.0.2), sf (>= 1.0-19), ggplotify (>= 0.1.2), gtable (>= 0.3.6), RplotterPkg (>= 0.1.3)

**Suggests** knitr (>= 1.4.9), testthat (>= 3.0.0), usethis (>= 3.1.0), usmap (>= 0.7.1), vdiffR (>= 1.0.8), classInt (>= 0.4-10), RColorBrewer (>= 1.1-3), here (>= 1.0.1), withr (>= 3.0.2)

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.3.2

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Rick Dean [aut, cre, cph] (ORCID:  
<<https://orcid.org/0009-0005-8086-8436>>)

**Maintainer** Rick Dean <deanr3@bardstown.com>

**Repository** CRAN

**Date/Publication** 2025-04-11 08:30:05 UTC

## Contents

get_category_strings . . . . .	2
get_dataset_names . . . . .	3
get_geography . . . . .	5
get_groups . . . . .	5
get_idb_data . . . . .	6
get_multi_vintage_data . . . . .	8
get_variable_names . . . . .	10
get_vintage_data . . . . .	12
join_it . . . . .	14
long_to_wide . . . . .	16
plot_us_data . . . . .	17
remove_area_water . . . . .	21
tiger_blocks_sf . . . . .	22
tiger_block_groups_sf . . . . .	24
tiger_cbsa_sf . . . . .	27
tiger_counties_sf . . . . .	29
tiger_county_subsection_sf . . . . .	32
tiger_landmarks_sf . . . . .	34
tiger_places_sf . . . . .	37
tiger_roads_sf . . . . .	39
tiger_states_sf . . . . .	42
tiger_tracts_sf . . . . .	45
tiger_urban_area_sf . . . . .	47
tiger_water_sf . . . . .	49
tiger_zctas_sf . . . . .	52
vote2020 . . . . .	54
wide_to_long . . . . .	55

**Index** **57**

---

get\_category\_strings    *get\_category\_strings*

---

## Description

For a Census Bureau categorical variable return a `data.table` with the variable's integer values and corresponding label strings.

The function will attempt to locate a Bureau's categorical variable by name and return a `data.table` with both the integer values and corresponding label strings. If the variable is not located then `NULL` is returned. Definitions for the variables were obtained from [Population Estimates Categorical Variables](#) and other Bureau sources.

**Usage**

```
get_category_strings(
  name = NULL,
  get_names = FALSE,
  start_idx = NULL,
  end_idx = NULL
)
```

**Arguments**

name	A required string that is the name of the categorical variable of interest.
get_names	An optional logical which if TRUE the function will return a vector of categorical variable names recognized by the function.
start_idx	An optional integer that sets the starting index of the variable's integer/label pairs. If the value is NULL(the default), then the variable's entire integer/label pairs are returned.
end_idx	An optional integer that sets the ending row index of the returned variable's integer/label pairs.

**Value**

A `data.table` with the variable's integer values and corresponding label strings.

**Examples**

```
library(data.table)
library(RcensusPkg)

# Get the names currently supported category variables
category_names_v <- RcensusPkg::get_category_strings(get_names = TRUE)

# Get the integer/string pairs for the category name "sex"
sex_category_dt <- RcensusPkg::get_category_strings(name = "sex")
```

---

get\_dataset\_names      *get\_dataset\_names*

---

**Description**

Get the acronym names and descriptions of the Census Bureau's datasets.

Function produces a `data.table` of the Census Bureau's dataset acronym names that can be used in other `RcensusPkg::` functions calling for a dataset acronym name. See Census Bureau's publicly available [datasets](#) for descriptions. Also see [All Surveys and Programs](#)

**Usage**

```
get_dataset_names(
  vintage = NULL,
  filter_name_str = NULL,
  filter_title_str = NULL,
  ignore_case = TRUE,
  brief = TRUE
)
```

**Arguments**

vintage	An optional numeric for the year to select the datasets. If NULL, then all the years are returned.
filter_name_str	An optional character string by which to filter the resultant data.table using the "name" column.
filter_title_str	An optional character string by which to filter the resultant data.table using the "title" column.
ignore_case	A logical which if FALSE will not ignore case in filtering the "title" column.
brief	A logical which if TRUE will return a resultant data.table with just columns "name", "vintage", "title". The default is TRUE.

**Value**

A list with a data.table of dataset names (named "data") and a vector of unique vintages (named "vintages").

**Examples**

```
## Not run:
# Requests for Census Bureau descriptions takes well over 10
# seconds in most cases.
library(data.table)
library(jsonlite)
library(httr2)
library(RcensusPkg)

# Get descriptions/vintages for 2020 datasets with "acs5" in their name.
acs5_datasets_ls <- RcensusPkg::get_dataset_names(
  vintage = 2020,
  filter_name_str = "acs5/"
)

## End(Not run)
```

---

get_geography	<i>get_geography</i>
---------------	----------------------

---

**Description**

Get the list of geography entities available (state, county, tract, etc) for a specific dataset.

Function produces a `data.table` of all the geography "name" and "geoLevelDisplay" variables available for a specific dataset and optionally a vintage.

**Usage**

```
get_geography(dataset, vintage = NULL)
```

**Arguments**

dataset	A required string that sets the acronym name of the data set of interest (e.g. "acs/acs5")
vintage	An optional numeric that sets the year of interest.

**Value**

A `data.table`

**Examples**

```
library(jsonlite)
library(data.table)
library(httr2)
library(RcensusPkg)
# Get the geographies available for dataset "acs/acs1/profile" with vintage 2019
acs1_profile_geo_dt <- RcensusPkg::get_geography(
  dataset = "acs/acs1/profile",
  vintage = 2019
)
```

---

get_groups	<i>get_groups</i>
------------	-------------------

---

**Description**

Get the names of Census Bureau variable groups and their descriptive parameters

Function produces a `data.table` of variable groups/tables and their descriptions.

**Usage**

```
get_groups(dataset = NULL, vintage = NULL)
```

**Arguments**

dataset	A required string that sets the acronym name of the data set of interest (e.g. "acs/acs5")
vintage	An required numeric that sets the year of interest.

**Value**

A data.table

**Examples**

```
library(jsonlite)
library(data.table)
library(httr2)
library(RcensusPkg)

acs5_groups_dt <- RcensusPkg::get_groups(
  dataset = "acs/acs5",
  vintage = 2019)
```

---

get_idb_data	<i>get_idb_data</i>
--------------	---------------------

---

**Description**

Get data from the Census Bureau's Time Series International Database for a specific dataset: *Single Year of Age and Sex*("1year") or *5-Year Age Groups and Sex*("5year"), year, variables, and country. See Census Bureau's publicly available [International Database](#) for dataset descriptions.

**Usage**

```
get_idb_data(
  dataset = NULL,
  years = 2023,
  vars = NULL,
  group = FALSE,
  wide_to_long = FALSE,
  countries = NULL,
  ages = 1:100,
  sex = c(0, 1, 2),
  key = Sys.getenv("CENSUS_KEY")
)
```

**Arguments**

dataset	A required string that selects the dataset of interest. Acceptable values are "1year" or "5year".
years	An optional numeric vector that sets the years of interest. The default is 2023.
vars	An optional string vector of variable names to be acquired. The default is GEO_ID,NAME,GENC,POP for either "1year" or "5year".
group	A logical that if TRUE returns all the variables from the "5year" dataset. The default is FALSE.
wide_to_long	The returned data.table is normally in a wide format with all the variables as columns. If this logical parameter is TRUE then a long format is returned with variable names in one column (named "estimate") and their respective values in another column (named "value").
countries	An optional string vector of country/area abbreviations (GENC Standard Countries and Areas) of interest. GENC codes are available <a href="#">here</a> .
ages	An optional numeric that selects specific ages(0 to 100 range) from the 1year dataset. Default is all ages.
sex	An optional numeric vector to select the sex of interest from the 1year dataset. Acceptable values 0 - Both, 1 - Male, 2 - Female. Default is all three values for sex.
key	A string that sets the access key. All Census Bureau API requests require an access key. Sign-up for a key is free and can be obtained <a href="#">here</a> . The function will check for a global setting of the key via Sys.getenv("CENSUS_KEY"). Run <code>usethis::edit_r_environ()</code> and edit your .Renviron file with the line: CENSUS_KEY=your key to create the global association. This is a required parameter.

**Details**

For the "1year", a data.table is returned with the following columns:

<b>GEO_ID</b>	char - geo id string
<b>NAME</b>	char - name of the country
<b>GENC</b>	char - GENC country abbreviation
<b>POP</b>	char - the total population for a specific age and sex in the country
<b>YR</b>	char - the year of the estimate
<b>AGE</b>	char - the specific age are values from 1 to 100
<b>SEX</b>	char - the specific sex with values 0 - Both, 1 - Male, 2 - Female

More details are available [here](#).

For the "5year", see [5year](#) and the 119 variable names and descriptions.

**Value**

A data.table

## Examples

```
## Not run:
# Requires Census Bureau API key
library(data.table)
library(httr2)
library(jsonlite)
library(RcensusPkg)

# 1year wide form, default variables, 2 countries
# (Botswana, Norway)
one_year_dt <- RcensusPkg::get_idb_data(
  dataset = "1year",
  years = c(2023, 2024),
  countries = c("BW", "NO")
)

# 5year long form, all variables, 2023, country = US
five_year_US_long_group_dt <- RcensusPkg::get_idb_data(
  dataset = "5year",
  years = 2023,
  group = TRUE,
  countries = "US",
  wide_to_long = TRUE
)

## End(Not run)
```

---

get\_multi\_vintage\_data

*get\_multi\_vintage\_data*

---

## Description

Get Census Bureau data for a specific dataset, variables, and region in the form of a `data.table` for multiple vintages. The function requires a Census Bureau access key.

## Usage

```
get_multi_vintage_data(
  dataset = NULL,
  vintage_v = NULL,
  vars = NULL,
  region = NULL,
  regionin = NULL,
  key = Sys.getenv("CENSUS_KEY")
)
```

**Arguments**

dataset	A required string that sets the name of the dataset of interest (e.g. "acs/acs5").
vintage_v	A required numeric vector that sets the vintages of interest.
vars	A required string vector of variable acronym names to be acquired (e.g. "B15002_015"). See RcensusPkg::get_variable_names() for obtaining acronym names.
region	A string that specifies the geography of the request. See Rcensus::get_geography() for assistance in obtaining these values.
regionin	A string that sets a qualifier for region.
key	A string that sets the access key. All Census Bureau API requests require an access key. Sign-up for a key is free and can be obtained <a href="#">here</a> . The function will check for a global setting of the key via Sys.getenv("CENSUS_KEY"). This is a required parameter.

**Value**

A data.table

**Examples**

```
## Not run:
# Requires Census Bureau API key
# Obtain the median home value ("B25077_001E") for Deschutes County, Oregon
# back to 2005 through 2019.
library(jsonlite)
library(data.table)
library(httr2)
library(usmap)
library(RcensusPkg)

# Get the fips codes for state and county
deschutes_fips <- usmap::fips("OR", "Deschutes")
state <- substr(deschutes_fips, 1, 2)
county <- substr(deschutes_fips, 3, 5)

RcensusPkg::get_multi_vintage_data(
  dataset = "acs/acs1",
  vintage_v = 2005:2019,
  vars = c("B25077_001E", "B25077_001M"),
  region = paste0("county:", county),
  regionin = paste0("state:", state)
)

## End(Not run)
```

---

get\_variable\_names     *get\_variable\_names*

---

### Description

Get Census Bureau variable acronym names and their label descriptions.

Function produces a `data.table` of variable acronym names and their descriptions. The function returns 4 columns:

<b>name</b>	the name of the parameter
<b>label</b>	the Bureau's description of the parameter
<b>required</b>	a boolean indicating if the parameter is required
<b>predicateType</b>	a string indicating the variable primitive type

Note that a variable with a **required** character value "true" must be included in your data requests (i.e. `RcensusPkg::get_vintage_data()`) or the API will return an error.

### Usage

```
get_variable_names(
  dataset = NULL,
  category = NULL,
  vintage = 2020,
  vars = NULL,
  group = NULL,
  filter_group_est = FALSE,
  filter_name_str = NULL,
  filter_label_str = NULL,
  filter_concept_str = NULL,
  ignore_case = TRUE,
  fixed = FALSE
)
```

### Arguments

<code>dataset</code>	An optional string that sets the name of a dataset category of interest.
<code>category</code>	An optional string that sets the category of datasets. The available categories are one of the following:
<code>acs1</code>	American Community Survey 1-Year Data
<code>acs/acsse</code>	American Community Survey Supplemental Data
<code>acs5</code>	American Community Survey 5-Year Data
<code>dec</code>	Decennial Census
<code>ecnbasic</code>	Economy-Wide Key Statistics
<code>timeseries/idb</code>	Time Series International Database

	abs	Annual Business Survey
vintage	An required numeric that sets the year of interest. The default is 2020.	
vars	An optional vector of variable names whose descriptions are of interest. This parameter requires that either 'dataset' or 'category' had been defined.	
group	An optional string that sets the group name associated with a set of variables. This parameter requires that either 'dataset' or 'category' had been defined. See <code>Rcensus::get_groups()</code> for available group names under a specific dataset and vintage.	
filter_group_est	A logical which if TRUE will filter the variable names from 'group' and return only estimate related variable names. The default is FALSE.	
filter_name_str	A character string by which to filter the resultant data.table's "name" column.	
filter_label_str	A character string by which to filter the resultant data.table's "label" column.	
filter_concept_str	A character string by which to filter the resultant data.table's "concept" column.	
ignore_case	A logical which if FALSE will not ignore case in filtering the "name", "label", "concept" column.	
fixed	A logical which if TRUE, then the above filter strings are used 'as is' in matching.	

### Details

The function's search for variable names depends on either specifying the parameters 'dataset' name or a dataset 'category'. Entering an available 'vintage' also influences obtaining a full dataframe of variable names and descriptions. To assist in using the function the user should consult the Census Bureau's publicly available [datasets descriptions](#). Also of help is `Rcensus::get_dataset_names()` for available dataset acronym names and their available years.

### Value

A data.table

### Examples

```
# Get available variables that have the phrase "educational attainment"
# in "label" column of the resultant data.table.

library(data.table)
library(httr2)
library(RcensusPkg)

educational_attainment_2019_dt <- RcensusPkg::get_variable_names(
  dataset = "acs/acs1/profile",
  vintage = 2019,
```

```

    filter_label_str = "educational attainment"
  )

```

---

```

get_vintage_data      get_vintage_data

```

---

## Description

Get Census Bureau data for a specific dataset, variables, and region in the form of a `data.table`.

Function produces a `data.table` with selected Census Bureau variables as columns. The function requires an access key issued from the Bureau. Variables of interest can be specified individually or by group/table name. Predicate phrases can be specified for filtering the results.

## Usage

```

get_vintage_data(
  dataset = NULL,
  vintage = NULL,
  vars = NULL,
  NAME_GEOID = TRUE,
  predicates = NULL,
  group = NULL,
  wide_to_long = FALSE,
  region = NULL,
  regionin = NULL,
  na_cols = NULL,
  key = Sys.getenv("CENSUS_KEY")
)

```

## Arguments

dataset	A required string that sets the name of the data set of interest (e.g. "acs/acs5"). Descriptions/vintages for datasets can be found by running <code>RcensusPkg::get_dataset_names()</code> .
vintage	An optional numeric that sets the vintage of interest. Available vintages for a specific dataset can be found by running <code>RcensusPkg::get_dataset_names()</code> .
vars	A required string vector (if the parameter 'group' is NULL) of variable names to be acquired. Available variable names can be determined by running <code>RcensusPkg::get_variable_names()</code> .
NAME_GEOID	A logical which if TRUE will add "NAME" and "GEO_ID" variables to the 'vars' string vector. The default is TRUE.
predicates	An optional vector of strings that adds data filtering. See <a href="#">Census Data API User Guide</a> for forming predicates and filtering or limiting variables. As noted in the guide each predicate must start with an ampersand sign.
group	An optional string that names an entire group of similar variables to be retrieved. For example the group value "B01001" would return values of all variables related to "SEX BY AGE". To find available groups submit a dataset and vintage to <code>RcensusPkg::get_groups</code> .

wide_to_long	If 'group' is defined then the returned data.table is normally in a wide format with all the group variables as columns. If this logical parameter is TRUE then a long format is returned with group variable names in one column (named "estimate") and their respective values in another column (named "value").
region	An optional string that specifies the geography of the request. See <a href="#">Federal Information Processing Series (FIPS)</a> for a listing of codes for this and the 'regionin' parameter. Not all regions such as counties, blocks, or tracts are available for a specific dataset and vintage. Use <code>RcensusPkg::get_geography()</code> to check on both 'region' and 'regionin'.
regionin	An optional string that sets a qualifier for 'region'.
na_cols	If TRUE will remove all rows with missing values. If a vector of column names/integers will check only those columns for missing values.
key	A required string that sets the access key. All Census Bureau API requests require an access key. Sign-up for a key is free and can be obtained <a href="#">here</a> . The function will check for a global setting of the key via <code>Sys.getenv("CENSUS_KEY")</code> . Run <code>usethis::edit_r_environ()</code> and edit your <code>.Renviron</code> file with the line: <code>CENSUS_KEY=your key</code> to create the global association.

## Details

See Census Bureau's publicly available [datasets](#) for dataset descriptions.

Some of the more common Census Bureau datasets include:

- [American Community Survey 1-year Data](#)
- [American Community Survey 1-year Supplemental Data](#)
- [American Community Survey 5-Year Data](#)
- [Decennial Census](#)
- [Economic Census](#)
- [Population Estimates and Projections](#)
- [Annual Business Survey \(ABS\)](#)

## Value

A data.table

## Examples

```
## Not run:
# Requires Census Bureau API key
# -----American Community Survey 5-Year Data (2009-2021)-----
# Description: https://www.census.gov/data/developers/data-sets/acs-5year.html

library(data.table)
library(httr2)
library(jsonlite)
library(stringr)
library(RcensusPkg)
```

```

# Get the data for the total number of white females ("B01001A_017E") and
# total number of white individuals ("B01001A_001E") across the US

white_females_dt <- RcensusPkg::get_vintage_data(
  dataset = "acs/acs5",
  vars = c("B01001A_017E", "B01001A_001E"),
  vintage = 2021,
  region = "state:*"
)

## End(Not run)

```

---

join\_it

*join\_it*


---

## Description

Outer join two dataframes that have a common column variable.

Function uses fast data.table techniques to join two data.tables by their common key values. Examples might include using the "GEOID" variable as a key to join data from RcensusPkg::get\_vintage\_data() with a simple feature with its geometries for counties, states, countries for example from RcensusPkg::tiger\_\*\_sf(). The resulting dataframe could then display the geometries (with RplotterPkg::create\_sf\_plot()) with an aesthetic mapping (e.g. fill/color/size) with a joined data column. Joining could also take place between two simple features (created by RcensusPkg::tiger\_\*\_sf()) or between two dataframes (created by RcensusPkg::get\_vintage\_data()).

The important thing to remember is that all the rows in 'df\_2' will be present in the resultant data.table.

## Usage

```

join_it(
  df_1 = NULL,
  df_2 = NULL,
  key_1 = NULL,
  key_2 = NULL,
  negate = FALSE,
  match = FALSE,
  return_sf = FALSE,
  na_rm = FALSE
)

```

## Arguments

df\_1            The first dataframe to be joined.

df\_2            The second dataframe to be joined with 'df\_1'. All rows in this dataframe will be present in the resultant dataframe.

key_1	A string that names the column from 'df_1' that is common to 'df_2'.
key_2	A string that names the column from 'df_2' that is common to 'df_1'.
negate	An optional logical which if TRUE will return a dataframe that has rows in 'df_1' but not in 'df_2'.
match	An optional logical which if TRUE will return a dataframe that has rows where only both 'df_1' and 'df_2' have matches.
return_sf	An optional logical which if TRUE will convert the resultant data.table to a simple feature if it has a geometries column.
na_rm	An optional logical which if TRUE then remove rows with NA values. The default is FALSE.

### Value

A data.table or simple feature object if 'return\_sf' is TRUE.

### Examples

```
## Not run:
# Requires Census Bureau API key
# Get the median household income by tract for Washington DC and join
# this data with DC tract boundaries.

library(data.table)
library(httr2)
library(jsonlite)
library(sf)
library(usmap)
library(withr)
library(ggplot2)
library(RcensusPkg)

# Get the 2020 median household income data by tract for DC
dc_fips <- usmap::fips(state = "dc")
dc_B19013_dt <- RcensusPkg::get_vintage_data(
  dataset = "acs/acs5",
  vintage = 2020,
  vars = "B19013_001E",
  region = "tract",
  regionin = paste0("state:", dc_fips)
)
# Get the simple feature DC tract geometries and join the data dataframe "dc_B19013_dt"
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}
dc_tracts_sf <- RcensusPkg::tiger_tracts_sf(
  state = dc_fips,
  output_dir = output_dir,
  general = TRUE,
  delete_files = FALSE
```

```

)
# Join the data with simple feature object
dc_joined_sf <- RcensusPkg::join_it(
  df_1 = dc_B19013_dt,
  df_2 = dc_tracts_sf,
  key_1 = "GEOID",
  key_2 = "GEOID",
  return_sf = TRUE
)

## End(Not run)

```

---

long\_to\_wide

*long\_to\_wide*


---

### Description

Reshape a `data.table` from a "long" format to a "wide" format.

Function calls `data.table::dcast()` to reshape a long single column and its values to multiple columns.

### Usage

```

long_to_wide(
  dt = NULL,
  id_v = c("NAME", "GEOID"),
  parameter_col = NULL,
  value_col = NULL
)

```

### Arguments

<code>dt</code>	The required <code>data.table</code> with a long column format.
<code>id_v</code>	A required vector of column names from <code>'dt'</code> that act as identifiers and are not part of the widened column.
<code>parameter_col</code>	A column name from <code>'dt'</code> whose unique values will become column names for the new expanded <code>data.table</code> .
<code>value_col</code>	A required column name or vector of column names from <code>'dt'</code> whose values will fall under the new expanded <code>data.table</code> .

### Value

A reshaped `data.table` in the "wide" format.

## Examples

```
## Not run:
# Requires Census Bureau API key
library(data.table)
library(httr2)
library(jsonlite)
library(stringr)
library(RcensusPkg)

# Request for data from Census Bureau in "long" form
B19001_1yr_long_dt <- RcensusPkg::get_vintage_data(
  dataset = "acs/acs1",
  vintage = 2016,
  group = "B19001",
  region = "state",
  wide_to_long = TRUE
)

# Resulting data.table is in the "long" form. Convert it back to
# to the wide form.
B19001_1yr_wide_dt <- RcensusPkg::long_to_wide(
  dt = B19001_1yr_long_dt,
  parameter_col = "variable",
  value_col = c("estimate", "moe")
)

## End(Not run)
```

---

plot\_us\_data

*plot\_us\_data*

---

## Description

This function produces a ggplot2 based choropleth map of a discrete/continuous variable across all/selected US states including placement of Alaska, Hawaii, and Puerto Rico. The function accepts a data frame with a column of state names and a column with their respective values. If the data frame (parameter 'df') is submitted as NULL then only the simple feature state geometries are returned for mapping. The function offers several options for control/selection of state geographies and variable scaling.

This function depends extensively on `RcensusPkg::tiger_states_sf()` for obtaining state geometries, so many of that function's parameters are repeated in this function. Also `RplotterPkg::create_sf_plot()` is called upon for displaying the shapefile geometries, so this package should be installed.

## Usage

```
plot_us_data(
  df = NULL,
  states_col = NULL,
```

```

value_col = NULL,
title = NULL,
title_fontsz = 18,
text_col = NULL,
text_size = 3,
text_color = "black",
text_fontface = "plain",
output_dir = tempdir(),
delete_files = TRUE,
vintage = 2020,
general = FALSE,
resol = "500k",
na_rm = FALSE,
scale_breaks = NULL,
scale_values = NULL,
scale_limits = NULL,
scale_labels = waiver(),
scale_colors = grDevices::heat.colors(8),
scale_na_value = "gray50",
own_scale = FALSE,
sf_color = "black",
sf_fill = "gray",
sf_linewidth = 0.1,
sf_alpha = 1,
display_plot = TRUE,
show_legend = TRUE,
legend_pos = "right",
legend_key_width = 0.5,
legend_key_height = 0.7,
legend_key_backgrd = "white"
)

```

### Arguments

<code>df</code>	The data frame with a column of full state names and a second variable column of their respective values. The column name for the states must be "NAME". If 'df' is NULL, then only a sf object with the state geometries are returned.
<code>states_col</code>	A required string (if 'df' is not NULL) that sets the column name from 'df' containing the state names of interest. These are full state names, either capitalized or lower case.
<code>value_col</code>	A required string (if 'df' is not NULL) that sets the column name from 'df' where values(discrete or continuous) are defined. If the column has discrete values then it must be a factor.
<code>title</code>	A string that sets the plot title.
<code>title_fontsz</code>	A numeric that sets the title's font size. The default is 18.
<code>text_col</code>	An optional string that sets the column name from 'df' for labeling each state polygon.

text_size	A numeric value that sets the size of labeled state text.
text_color	A string that sets the color of labeled state text.
text_fontface	A string that sets the fontface of labeled state text. Acceptable values: "plain", "bold", "italic", "bold.italic". The default is "plain".
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by tempdir().
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the state geometries.
resol	If 'general' is TRUE, then the resolution to return. Acceptable values are strings "500k", "5m", "20m".
na_rm	A logical which if TRUE, missing observations are removed. If FALSE, the default, missing observations are removed with a warning.
scale_breaks	A required string/numeric vector that defines the scale breaks.
scale_values	A string/numeric vector that defines the possible values. For factor values, this is required and is a vector string of colors.
scale_limits	A required string/numeric vector that defines the scale limits.
scale_labels	An optional string vector that defines the scale labels. Vector must be the same length as scale_breaks.
scale_colors	Vector of colors to use for n-color gradient.
scale_na_value	A string that sets the color for missing values.
own_scale	A logical which if TRUE, then your own scaling may be appended to the plot without using the above scale_* parameters.
sf_color	A string that sets the polygon border line color.
sf_fill	A string that sets the polygon area fill color.
sf_linewidth	A numeric that sets the border line thickness.
sf_alpha	A numeric that sets the alpha level attribute of the polygon fill.
display_plot	A logical that if TRUE will display the plot. The default is TRUE. If FALSE, then a ggplot2 object is returned
show_legend	A logical that controls the appearance of the legend.
legend_pos	A string that sets the legend position. Acceptable values are "right", "top", "bottom".
legend_key_width	A numeric that sets the legend width in cm.
legend_key_height	A numeric that sets the legend height in cm.
legend_key_backgrd	A string that sets the legend's background color.

**Value**

A list of ggplot2 objects if 'display\_plot' is FALSE. Included in the list is the plot of all the states ("us\_states") along with the original ggplot2 `geom_sf` plots of the lower 48 ("lower\_48"), Alaska ("alaska"), Hawaii ("hawaii") and Puerto Rico ("puerto\_rico").

**Examples**

```
## Not run:
library(sf)
library(grid)
library(ggplot2)
library(ggplotify)
library(data.table)
library(gtable)
library(httr2)
library(withr)
library(RplotterPkg)

# Plot just the states without joining data (the default case)
# Define a temporary output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}
us_without_data_plot <- RcensusPkg::plot_us_data(
  title = "A Default Mapping of US States",
  output_dir = output_dir,
  delete_files = FALSE
)

# Requires US Census Bureau API key
# Plot of US map with discrete 2020 presidential results
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}
a_plot <- RcensusPkg::plot_us_data(
  df = RcensusPkg::vote2020,
  title = "US Presidential Vote 2020",
  states_col = "State",
  value_col = "Party",
  output_dir = output_dir,
  delete_files = FALSE,
  scale_breaks = c("R", "D"),
  scale_limits = c("R", "D"),
  scale_values = c("red", "blue"),
  scale_labels = c("Republican", "Democrat"),
  sf_color = "white"
)

## End(Not run)
```

---

remove\_area\_water      *remove\_area\_water*

---

### Description

Function removes water area geometries from downloaded Census Bureau shapefiles

### Usage

```
remove_area_water(  
  x,  
  vintage = 2020,  
  output_dir = tempdir(),  
  delete_files = TRUE  
)
```

### Arguments

x	An sf shapefile object with possible areas of water downloaded from the Census Bureau
vintage	An integer that specifies the year of the shapefile. The default is 2020.
output_dir	A full directory path where supportive shapefiles and their associated files will be downloaded. The default is the directory defined by the value returned by tempdir().
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.

### Value

An sf object with the water area geometries removed.

### Examples

```
## Not run:  
# Request for Census Bureau data plus CPU processing can exceed well over 10  
# seconds in some cases.  
library(downloader)  
library(sf)  
library(usmap)  
library(withr)  
library(data.table)  
library(RcensusPkg)  
  
# Remove water areas from New York, New York sf object  
ny_state_county_fips <- usmap::fips(state = "New York", county = "New York")  
ny_state_fips <- substr(ny_state_county_fips, 1,2)  
ny_county_fips <- substr(ny_state_county_fips, 3, 5)
```

```

# Define a temporary output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

# Define an expression for filtering just the New York county
express <- parse(text = paste0("COUNTYFP == ", "'", ny_county_fips, "'"))

# Get the New York,New York county tract sf geometries object
ny_tracts_sf <- RcensusPkg::tiger_tracts_sf(
  state = ny_state_fips,
  vintage = 2020,
  general = FALSE,
  transform_crs = 6538,
  express = express,
  output_dir = output_dir,
  delete_files = FALSE
)

# Remove the area waters from New York, New York sf object
ny_without_water_sf <- RcensusPkg::remove_area_water(
  ny_tracts_sf,
  output_dir = output_dir,
  delete_files = FALSE
)

## End(Not run)

```

---

tiger_blocks_sf	<i>tiger_blocks_sf</i>
-----------------	------------------------

---

## Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

## Usage

```

tiger_blocks_sf(
  state = NULL,
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  set_crs = NULL,
  transform_crs = NULL,

```

```

    sf_info = FALSE,
    do_progress = FALSE,
    shapefile = NULL,
    express = NULL,
    check_na = FALSE
  )

```

## Arguments

state	A required two-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "COUNTYFP". If you wanted to return just the geometries for Los Alamos, New Mexico (which has a fips code of "028"), then you assign 'express' equal to: <code>expression(COUNTYFP == "028")</code> . The expression will be evaluated and only the tract geometries for Los Alamos will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature (sf) of US Census block boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional block related variables are provided. See [Appendix G-1. Record Layout: Block State-based Shapefile](#) for a description of block related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.2 Blocks](#) – "Census blocks are statistical areas bounded on all sides by visible features (e.g., streets, roads, streams, and railroad tracks), and by non-visible boundaries (e.g., city, town, township, county limits, and short line-of-sight extensions of streets and roads)."

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data via the GEOID value. Be aware that shapefiles for blocks can be very large and time consuming to download for some states.

Some earlier vintages may have NA for the crs so you may need to specify the `crs_transform` to 3426. Also you may be interested in using a state level crs. See [epsg.io](http://epsg.io) to search worldwide for crs.

## Value

A data frame object of class sf

## Examples

```
## Not run:
# Beware that downloading and processing state blocks can be time consuming
library(sf)
library(downloader)
library(usmap)
library(withr)
library(data.table)
library(RcensusPkg)

vt_fips <- usmap::fips(state="vermont")
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}
vt_blocks_sf <- RcensusPkg::tiger_blocks_sf(
  state = vt_fips,
  output_dir = output_dir,
  delete_files = FALSE
)

## End(Not run)
```

---

tiger\_block\_groups\_sf *tiger\_block\_groups\_sf*

---

## Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

**Usage**

```

tiger_block_groups_sf(
  state = NULL,
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  sf_info = FALSE,
  do_progress = FALSE,
  set_crs = NULL,
  transform_crs = NULL,
  shapefile = NULL,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = "GEOID",
  express = NULL,
  check_na = FALSE
)

```

**Arguments**

state	A required two-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the block group geometries.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.

express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "COUNTYFP". If you wanted to return just the geometries for Los Alamos, New Mexico (which has a fips code of "028"), then you assign 'express' equal to: <code>expression(COUNTYFP == "028")</code> . The expression will be evaluated and only the tract geometries for Los Alamos will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

### Details

Returns simple feature (sf) of US Census block group boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional block group related variables are provided. See [Appendix G-2. Record Layout: Block Group State-based Shapefile](#) for a description of block group related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.3 Block Groups](#) – "Standard block groups are clusters of blocks within the same census tract that have the same first digit of their 4-character census block number (e.g., Blocks 3001, 3002, 3003 to 3999 in census tract 1210.02 belong to block group 3). Current block groups do not always maintain these same block number to block group relationships due to boundary and feature changes that occur throughout the decade. A block group usually covers a contiguous area. Each census tract contains one or more block groups and block groups have unique numbers within census tract."

A more generalized, recognizable version of the block group geometries that has less download size is also available. For more information on cartographic boundary files see [Cartographic Boundary File Description](#). These files are available for vintages greater than 2013 with resolution 1:500k meters.

Some earlier vintages may have NA for the crs so you may need to specify the `crs_transform` to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data via the GEOID value.

### Value

A data frame object of class sf

### Examples

```
library(sf)
library(downloader)
library(data.table)
library(usmap)
library(withr)
library(RcensusPkg)

# Get block groups for Washington, DC
# Get the DC fips code
dc_fips <- usmap::fips(state = "dc")
```

```
# Define a temporary output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

dc_block_groups_sf <- RcensusPkg::tiger_block_groups_sf(
  state = dc_fips,
  output_dir = output_dir,
  delete_files = FALSE
)
```

---

tiger\_cbsa\_sf

*tiger\_cbsa\_sf*

---

## Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

## Usage

```
tiger_cbsa_sf(
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  resol = "500k",
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = "GEOID",
  state_filter = NULL,
  city_filter = NULL,
  check_na = FALSE
)
```

## Arguments

output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the state geometries.
resol	If 'general' is TRUE, then the resolution to return. Acceptable values are strings "500k", "5m", "20m".
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.
state_filter	A string that filters the resultant sf by a state name (e.g. "TX").
city_filter	A string that filters the resultant sf by a city name (e.g. "Kansas City").
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature (sf) of core based statistical area (CBSA) boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features.

A more generalized, recognizable version of the CBSA geometries that has less download size is also available. For more information on cartographic boundary files see [Cartographic Boundary File Description](#). These files are available for vintages greater than 2009 with resolution 1:500k, 1:5m, 1:20m meters. For descriptive information on CBSA see [About](#).

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data. To help incorporate data files, this function has a 'datafile' parameter which will be joined with the resultant simple feature object. The only requirement is that a common "key" for joining exist between the data dataframe and the simple feature dataframe.

**Value**

A data frame object of class sf

**Examples**

```
library(downloader)
library(sf)
library(data.table)
library(withr)
library(RcensusPkg)

# Define a temporary, self deleting output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

# Get shapefile geometries for the Core-based statistical areas
# of Texas with vintage 2020.
# Note that there is also a city filter parameter (city_filter).
cbsa_tx_sf <- RcensusPkg::tiger_cbsa_sf(
  vintage = 2020,
  resol = "20m",
  # city_filter = "Kansas City",
  state_filter = "TX",
  general = TRUE,
  output_dir = output_dir,
  delete_files = FALSE
)
```

---

tiger\_counties\_sf      *tiger\_counties\_sf*

---

**Description**

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

**Usage**

```
tiger_counties_sf(
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
```

```

general = FALSE,
resol = "500k",
set_crs = NULL,
transform_crs = NULL,
sf_info = FALSE,
do_progress = FALSE,
shapefile = NULL,
datafile = NULL,
datafile_key = NULL,
sf_key = "COUNTYFP",
express = NULL,
check_na = FALSE
)

```

### Arguments

output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the county geometries.
resol	If 'general' is TRUE, then the resolution to return. Acceptable values are strings "500k", "5m", "20m".
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "STATEFP". If you wanted to return just the geometries for Florida (which has a fips code of "12"), then you assign 'express' equal to: <code>expression(STATEFP == "12")</code> . The expression will be evaluated and only the geometries for Florida will be returned.

check\_na        A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature (sf) of the entire US county boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional county related variables are provided. See [Appendix I-3. Record Layout: County Subdivision State-based Shapefile](#)) for a description of county related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.7 Counties and Equivalent Entities](#) – "Counties and equivalent entities are primary legal divisions of states. In most states, these entities are termed 'counties.'"

A more generalized, recognizable version of the county geometries that has less download size is also available. For more information on cartographic boundary files see [Cartographic Boundary File Description](#). These files are available for vintages greater than 2013 with resolution 1:500k, 1:5m, 1:20m meters.

Some earlier vintages may have NA for the crs so you may need to specify the 'crs\_transform' to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

The function returns the simple feature object which can easily be mapped (see [RplotterPkg::create\\_sf\\_plot\(\)](#)) or joined with US Census Bureau demographic data via the GEOID value.

## Value

An object of class sf

## Examples

```
library(downloader)
library(sf)
library(data.table)
library(usmap)
library(withr)
library(RcensusPkg)

# Get the geometries for Ohio counties
# Determine the fips code for Ohio (returns "39")
ohio_fips <- usmap::fips(state = "ohio")
# Create an expression to filter out just Ohio counties
# from the simple feature dataframe
express <- parse(text = paste0("STATEFP == ", "'", ohio_fips, "'"))
# Define a temporary output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}
# Get the Ohio county's generalized geometries
ohio_counties_sf <- RcensusPkg::tiger_counties_sf(
  general = TRUE,
  express = express,
```

```

    output_dir = output_dir,
    delete_files = FALSE
  )

```

---

```

tiger_county_subsection_sf
      tiger_county_subsection_sf

```

---

## Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

## Usage

```

tiger_county_subsection_sf(
  state = NULL,
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = "GEOID",
  express = NULL,
  check_na = FALSE
)

```

## Arguments

state	A required two-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.

general	A logical which if TRUE will download a less detailed, more generalized version of the tract geometries.
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "COUNTYFP". If you wanted to return just the geometries for Los Alamos, New Mexico (which has a fips code of "028"), then you assign 'express' equal to: <code>expression(COUNTYFP == "028")</code> . The expression will be evaluated and only the tract geometries for Los Alamos will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature (sf) of US Census county subsection boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional county subsection related variables are provided. See [Appendix I-3. Record Layout: County Subdivision State-based Shapefile](#) for a description of county subsection related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). See [Chapter 4.8 County Subdivisions](#) for a description of the term.

A more generalized, recognizable version of the tract geometries that has less download size is also available. For more information on cartographic boundary files see [Cartographic Boundary File Description](#). These files are available for vintages greater than 2013 with resolution 1:500k meters.

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data. To help incorporate data files, this function has a 'datafile' parameter which will be joined with the resultant simple feature object. The only requirement is that a common "key" for joining exist between the data dataframe and the simple feature dataframe.

Some earlier vintages may have NA for the crs so you may need to specify the 'crs\_transform' to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

**Value**

A data frame object of class sf

**Examples**

```
library(downloader)
library(sf)
library(data.table)
library(withr)
library(usmap)
library(RcensusPkg)

# Find the subsections for a county in Ohio
# Define a temporary, self deleting output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

# Define the fips values for state and county
ohio_hc_fips <- usmap::fips(state = "ohio", county = "holmes")
ohio_fips <- substr(ohio_hc_fips,1,2)
hc_fips <- substr(ohio_hc_fips,3,5)

# Define a filtering expression for the county
express <- parse(text = paste0("COUNTYFP == ", "'", hc_fips, "'"))

# Get the subsection sf object for the Ohio county
hc_ctysub_sf <- RcensusPkg::tiger_county_subsection_sf(
  state = ohio_fips,
  vintage = 2020,
  general = TRUE,
  express = express,
  output_dir = output_dir,
  delete_files = FALSE
)
```

---

tiger\_landmarks\_sf      *tiger\_landmarks\_sf*

---

**Description**

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

**Usage**

```
tiger_landmarks_sf(
  state = NULL,
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  entity = "point",
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  express = NULL,
  check_na = FALSE
)
```

**Arguments**

state	The required two-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020. The value should be greater than 2010.
entity	A character string that sets the specific landmark entity of interest. The acceptable values are "area" or "point". The default is "point".
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "STATEFP". If you wanted to return just the geometries for Nelson County, Kentucky (which has a fips code of "179"), then you assign 'express' equal to: <code>expression(COUNTYFP == "179")</code> . The expression will be evaluated and only the landmark geometries for Nelson County will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

**Note: Vintage must be greater than 2010.**

## Details

Returns simple feature (sf) landmark related geometric polygons provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional landmark related variables are provided. See [Appendix K-1.K-2, Record Layout: Area/Point Landmark Shapefile](#) for a description of landmark related area and point variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.11 Landmark \(Area and Point – "The Census Bureau includes landmarks in the MAF/TIGER System to locate special features and help enumerators during field operations. Some of the more common landmark types include area landmarks \(e.g., airports, cemeteries, parks, and educational facilities\) and point landmarks \(e.g., schools and churches\)"](#)

The function returns the simple feature object which can easily be mapped (see [RplotterPkg::create\\_sf\\_plot\(\)](#)) or joined with US Census Bureau demographic data via the GEOID value.

Some earlier vintages may have NA for the crs so you may need to specify the 'transform\_crs' to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

## Value

A data frame object of class sf

## Examples

```
library(downloader)
library(sf)
library(data.table)
library(withr)
library(usmap)
library(RcensusPkg)

# Get the sf object for landmarks in the state of Kentucky
# Define a temporary, self deleting output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

# Get the fips code for the state
kentucky_fips <- usmap::fips(state = "kentucky")

# Get the "point" landmarks for Kentucky
kentucky_landmarks_sf <- RcensusPkg::tiger_landmarks_sf(
  state = kentucky_fips,
  entity = "point",
  check_na = TRUE,
  output_dir = output_dir,
  delete_files = FALSE
)
```

---

tiger_places_sf	<i>tiger_places_sf</i>
-----------------	------------------------

---

## Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

## Usage

```
tiger_places_sf(
  state = NULL,
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  express = NULL,
  check_na = FALSE
)
```

## Arguments

state	The required two-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020. The value should be greater than 2010.
general	A logical which if TRUE will download a less detailed, more generalized version of the places geometries.
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.

sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "NAME" for major places. Say we wanted to return just the geometries for major places in Kentucky such as "Bowling Green" and "Louisville". We would assign 'express' equal to: <code>expression(NAME %in% c("Bowling Green", "Louisville"))</code> . The expression will be evaluated and only the geometries for the two places will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature (sf) places related geometric polygons provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional places related variables are provided. See [Appendix I-5. Record Layout: Place State-based Shapefile](#) for a description of place related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [4.14 Places – Incorporated Places and Census Designated Places](#) –

"Incorporated Places: An incorporated place provides governmental functions for a concentration of people. Incorporated places may extend across county and county subdivision boundaries, but never across state boundaries. An incorporated place usually is a city, town, village, or borough, but can have other legal descriptions."

"Census Designated Places (CDPs): CDPs are the statistical counterparts of incorporated places. CDPs are settled concentrations of population that are identifiable by name but not legally incorporated under the laws of the state in which the CDPs are located."

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data via the GEOID value.

Some earlier vintages may have NA for the crs so you may need to specify the 'crs\_transform' to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

## Value

A data frame object of class sf

## Examples

```
library(downloader)
library(sf)
library(data.table)
library(withr)
library(usmap)
library(RcensusPkg)
```

```

# Get the major places locations in the state of Kentucky
# Define a temporary, self deleting output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

# Get the fips code for the state
kentucky_fips <- usmap::fips(state = "kentucky")

# Create a filter expression to get select places
major_places_express <- expression(NAME %in% c(
  "Bardstown",
  "Bowling Green",
  "Louisville",
  "Versailles",
  "Owensboro",
  "Frankfort",
  "Elizabethtown",
  "Danville"
))

# Get the sf object and the geometric locations of the selected places
kentucky_places_sf <- RcensusPkg::tiger_places_sf(
  state = kentucky_fips,
  express = major_places_express,
  general = TRUE,
  output_dir = output_dir,
  delete_files = FALSE
)

```

---

tiger\_roads\_sf

*tiger\_roads\_sf*

---

### Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

### Usage

```

tiger_roads_sf(
  state = NULL,
  county = NULL,
  output_dir = tempdir(),

```

```

delete_files = TRUE,
vintage = 2020,
entity = "us_roads",
set_crs = NULL,
transform_crs = NULL,
sf_info = FALSE,
do_progress = FALSE,
shapefile = NULL,
datafile = NULL,
datafile_key = NULL,
sf_key = "GEOID",
express = NULL,
check_na = FALSE
)

```

### Arguments

state	A required 2-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
county	The <i>three-digit</i> FIPS code for the county of interest.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020. The value should be greater than 2010.
entity	A string that defines the category of road geometries to return. Acceptable value are "us_roads" (state & county arguments not required), "state_roads" (state argument required), "county_roads" (state & county arguments required).
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.

express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "STATEFP". If you wanted to return just the geometries for Florida (which has a fips code of "12"), then you assign 'express' equal to: <code>expression(STATEFP == "12")</code> . The expression will be evaluated and only the geometries for Florida will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values. <b>Note: vintage must be greater than 2010</b>

## Details

Returns simple feature (sf) of US Census roads boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional roads related variables are provided. See [Appendix L-3. Record Layouts: Roads Shapefile](#) for a description of road related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.12.3 Roads – Primary, Secondary and All Roads](#) – "The primary and secondary roads shapefile contains all linear street features with MTFCCs of primary roads (S1100) or secondary roads (S1200) in the MAF/TIGER System. Secondary roads are main arteries, usually in the U.S. highway, state highway, or county highway system. These roads have one or more lanes of traffic in each direction, may or may not be divided, and usually have at-grade intersections with many other roads and driveways. These roads often have both a local name and a route number."

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data via the GEOID value.

Some earlier vintages may have NA for the crs so you may need to specify the 'crs\_transform' to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

## Value

A data frame object of class sf

## Examples

```
library(downloader)
library(sf)
library(data.table)
library(withr)
library(usmap)
library(RcensusPkg)

# Get the sf geometries of roads in Ohio
# Get the fips for Ohio
oh_fips <- usmap::fips(state = "ohio")

# Define a temporary, self deleting output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
```

```

    dir.create(output_dir)
  }

# Get the sf object
ohio_roads_sf <- RcensusPkg::tiger_roads_sf(
  state = oh_fips,
  entity = "state_roads",
  output_dir = output_dir,
  delete_files = FALSE
)

```

---

tiger_states_sf	<i>tiger_states_sf</i>
-----------------	------------------------

---

### Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

### Usage

```

tiger_states_sf(
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  resol = "500k",
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = "GEOID",
  express = NULL,
  check_na = FALSE
)

```

### Arguments

`output_dir` A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by `tempdir()`.

delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the state geometries.
resol	If general is TRUE, then the resolution to return. Acceptable values are strings "500k", "5m", "20m".
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant sf dataframe used to key with the 'datafile' dataframe.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "STATEFP". If you wanted to return just the geometries for Florida (which has a fips code of "12"), then you assign 'express' equal to: <code>expression(STATEFP == "12")</code> . The expression will be evaluated and only the geometries for Florida will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature (sf) of state boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional state related variables are provided. See [Appendix P-5. Record Layout: State and Equivalent Entity National Shapefile](#) for a description of state related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.17 States and State Equivalent Entities](#) – "States and equivalent entities are the primary governmental divisions of the United States. In addition to the fifty states, the Census Bureau treats the District of Columbia, Puerto Rico, and the Island Areas (American Samoa, the Commonwealth of the Northern Mariana Islands, Guam, and the U.S. Virgin Islands) as statistical equivalents of states for the purpose of data presentation."

A more generalized, recognizable version of the state geometries that has less download size is also available. For more information on cartographic boundary files see [Cartographic Boundary File](#)

**Description.** These files are available for vintages greater than 2013 with resolution 1:500k, 1:5m, 1:20m meters.

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data. To help incorporate data files, this function has a 'datafile' parameter which will be joined with the resultant simple feature object. The only requirement is that a common "key" for joining exist between the data dataframe and the simple feature dataframe.

Some earlier vintages may have NA for the crs so you may need to specify the 'crs\_transform' to 3426. Also you may be interested in using a state level crs. See [epsg.io](http://epsg.io) to search worldwide for crs.

### Value

A data frame object of class sf (simple feature)

### Examples

```
library(usmap)
library(sf)
library(data.table)
library(downloader)
library(withr)
library(RcensusPkg)

# Get the fips code for Florida
florida_fips <- usmap::fips(state = "florida")
#
# Define an expression that will filter out Florida from
# the simple feature obtained from the downloaded/converted
# tiger states shapefile.
express <- parse(text = paste0("STATEFP == ", "'", florida_fips, "'"))
#
# Define a temporary output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

# Download, convert, and filter the states shapefile for Florida
# using the expression.
florida_general_sf <- RcensusPkg::tiger_states_sf(
  general = TRUE,
  express = express,
  output_dir = output_dir,
  delete_files = FALSE
)
```

---

tiger_tracts_sf	<i>tiger_tracts_sf</i>
-----------------	------------------------

---

## Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

## Usage

```
tiger_tracts_sf(
  state = NULL,
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = "GEOID",
  express = NULL,
  check_na = FALSE
)
```

## Arguments

state	A required two-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A required numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the tract geometries.
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.

transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "COUNTYFP". If you wanted to return just the geometries for Los Alamos, New Mexico (which has a fips code of "028"), then you assign 'express' equal to: <code>expression(COUNTYFP == "028")</code> . The expression will be evaluated and only the tract geometries for Los Alamos will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature (sf) of US Census tract boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional tract related variables are provided. See [Appendix G-3. Record Layout: Census Tract State-based Shapefile](#) for a description of tract related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.4 Census Tracts](#) – "Census tracts are small and relatively permanent statistical subdivisions of a county or equivalent entity. Local participants review and update census tracts prior to each decennial census as part of the Census Bureau's PSAP."

A more generalized, recognizable version of the tract geometries that has less download size is also available. For more information on cartographic boundary files see [Cartographic Boundary File Description](#). These files are available for vintages greater than 2013 with resolution 1:500k meters.

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data. To help incorporate data files, this function has a `datafile` parameter which will be joined with the resultant simple feature object. The only requirement is that a common "key" for joining exist between the data dataframe and the simple feature dataframe.

Some earlier vintages may have NA for the crs so you may need to specify the 'crs\_transform' to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

## Value

A data frame object of class sf (simple feature)

**Examples**

```

library(downloader)
library(sf)
library(data.table)
library(usmap)
library(withr)
library(RcensusPkg)

# Get the generalized tract geometries for Los Alamos, New Mexico
# Determine the fips codes for New Mexico and Los Alamos
nm_los_alamos_fips <- usmap::fips(state = "new mexico", county = "los alamos")
nm_fips <- substr(nm_los_alamos_fips, 1, 2)
los_alamos_fips <- substr(nm_los_alamos_fips, 3, 5)

# Create an expression to filter just the Los Alamos geometries
express <- parse(text = paste0("COUNTYFP == ", "'", los_alamos_fips, "'"))

# Define a temporary output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

# Get the tract geometries for just Los Alamos
losalamos_tracts_sf <- RcensusPkg::tiger_tracts_sf(
  state = nm_fips,
  general = TRUE,
  express = express,
  output_dir = output_dir,
  delete_files = FALSE
)

```

---

tiger\_urban\_area\_sf    *tiger\_urban\_area\_sf*

---

**Description**

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

The function returns geometries for the Census Bureau's defined "urban areas" across the entire US

**Usage**

```
tiger_urban_area_sf(
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  resol = "500k",
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = "GEOID",
  check_na = FALSE
)
```

**Arguments**

output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the state geometries.
resol	If 'general' is TRUE, then the resolution to return. Acceptable values are strings "500k", "5m", "20m".
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.

## Details

Returns simple feature(sf) of the Census Bureau's defined "urban area" boundary related geometric polygons, provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features.

A more generalized, recognizable version of the CBSA geometries that has less download size is also available. For more information on cartographic boundary files see [Cartographic Boundary File Description](#). These files are available for vintages greater than 2009 with resolution 1:500k, 1:5m, 1:20m meters. For descriptive information on CBSA see [About](#).

The function returns the simple feature object which can easily be mapped (see `RplotterPkg::create_sf_plot()`) or joined with US Census Bureau demographic data. To help incorporate data files, this function has a 'datafile' parameter which will be joined with the resultant simple feature object. The only requirement is that a common "key" for joining exist between the data dataframe and the simple feature dataframe.

## Value

A data frame object of class sf

## Examples

```
library(downloader)
library(sf)
library(data.table)
library(withr)
library(RcensusPkg)

# Get the defined urban areas for 2019
# Define a temporary, self deleting output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}

us_urban_areas_sf <- RcensusPkg::tiger_urban_area_sf(
  output_dir = output_dir,
  vintage = 2019,
  general = TRUE,
  delete_files = FALSE
)
```

## Description

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

## Usage

```
tiger_water_sf(
  state = NULL,
  county = NULL,
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  entity = "area",
  set_crs = NULL,
  transform_crs = NULL,
  sf_info = FALSE,
  do_progress = FALSE,
  shapefile = NULL,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = "GEOID",
  express = NULL,
  check_na = FALSE
)
```

## Arguments

state	The two-digit FIPS code for the state of interest. See <a href="#">usmap::fips function</a> for finding FIPS codes.
county	The <i>three-digit</i> FIPS code for the county of interest.
output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by <code>tempdir()</code> .
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020. The value should be greater than 2010.
entity	A character string that sets the specific water entity of interest. The acceptable values are "area", "linear", or "coastline". The default is "area".
set_crs	A numeric or character string which if non-NULL calls <code>sf::st_crs()</code> to set the crs of the geometries and transforms them.
transform_crs	A numeric or character string which if non-NULL calls <code>sf::st_transform()</code> to perform a crs transform of the geometries. Note that the crs of the shapefile must not be NA.

sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
shapefile	A full file path to a shapefile folder with its unzipped files to be processed instead of downloading.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "STATEFP". If you wanted to return just the geometries for Florida (which has a fips code of "12"), then you assign 'express' equal to: <code>expression(STATEFP == "12")</code> . The expression will be evaluated and only the geometries for Florida will be returned.
check_na	A logical which if TRUE will remove rows that have missing values for any of the columns. The default is to not check the columns for NA values.  <b>Note: If entity equals "coastline" then the state and county arguments are not required.</b>

## Details

Returns simple feature (sf) water related geometric polygons provided by the US Census Bureau's TIGER/Line Shapefiles database. See [Simple Features for R](#) for more information on simple features. Along with the geometries, additional water related variables are provided. See [Appendix J. Record Layouts: Hydrography \(Area and Line\)](#) for a description of water related area and line variables of the sf file. See [Appendix L-2. Record Layout: Coastlines Shapefile](#) for a description of coastline related variables of the sf file. For further information on the Census Bureau's shape files see [About the 2021 TIGER/Line Shapefiles](#). From [Chapter 4.10 Hydrography \(Area and Linear\)](#) and [4.12.2 Coastline](#) – "The area hydrography shapefile contains the geometry and attributes of both perennial and intermittent area hydrography features (e.g., ponds, lakes, oceans, swamps, glaciers, and the area covered by large streams represented as double-line drainage). The linear hydrography shapefile includes streams/rivers, braided streams, canals, ditches, artificial paths, and aqueducts. A linear hydrography feature may include edges with both perennial and intermittent persistence"

The function returns the simple feature object which can easily be mapped (see [RplotterPkg::create\\_sf\\_plot\(\)](#)) or joined with US Census Bureau demographic data via the GEOID value.

Some earlier vintages may have NA for the crs so you may need to specify the 'crs\_transform' to 3426. Also you may be interested in using a state level crs. See [epsg.io](#) to search worldwide for crs.

## Value

A data frame object of class sf

**Examples**

```

library(sf)
library(data.table)
library(downloader)
library(usmap)
library(withr)
library(RcensusPkg)

# Get the water areas for a county in Ohio
state_county_fips <- usmap::fips(state = "Ohio", county = "Geauga")
state_fips <- substr(state_county_fips,1,2)
county_fips <- substr(state_county_fips,3,5)
# Define a temporary output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
}
geauga_area_water_sf <- RcensusPkg::tiger_water_sf(
  state = state_fips,
  county = county_fips,
  output_dir = output_dir,
  delete_files = FALSE
)

```

---

*tiger\_zctas\_sf*
*tiger\_zctas\_sf*


---

**Description**

This function performs three tasks:

1. Download to an output directory a zip file from the TIGER/Line Shapefiles database.
2. Unzip the zip file and locate the shape file of interest.
3. Read and convert the shape file to a simple feature object.

**Usage**

```

tiger_zctas_sf(
  output_dir = tempdir(),
  delete_files = TRUE,
  vintage = 2020,
  general = FALSE,
  sf_info = FALSE,
  do_progress = FALSE,
  datafile = NULL,
  datafile_key = NULL,
  sf_key = NULL,
  express = NULL
)

```

**Arguments**

output_dir	A full directory path where the shapefile and its associated files will be downloaded. The default is the directory defined by the value returned by tempdir().
delete_files	A logical which if TRUE will delete the shapefile and associated files in 'output_dir'. The default is TRUE.
vintage	A numeric that sets the vintage of interest. The default is 2020.
general	A logical which if TRUE will download a less detailed, more generalized version of the county geometries.
sf_info	A logical which if TRUE displays info on the resulting simple feature object.
do_progress	A logical which if TRUE displays a progress bar during the download.
datafile	A dataframe containing data that should be joined with this function's resultant simple feature object.
datafile_key	The column name from 'datafile' dataframe used to key with the 'sf_key' column of the resultant simple feature dataframe.
sf_key	The column from the resultant dataframe used to key with the 'datafile' dataframe.
express	A logical expression object used to filter the resultant simple feature dataframe. For example, one of the columns of the resultant simple feature dataframe is "STATEFP". If you wanted to return just the geometries for Florida (which has a fips code of "12"), then you assign 'express' equal to: expression(STATEFP == "12"). The expression will be evaluated and only the geometries for Florida will be returned.

**Details**

Returns simple feature (sf) shapefile Zip Code Tabulation Areas(ZCTA) shapefile of the entire US.

**Value**

A data frame object of class sf

**Examples**

```
## Not run:
# Beware that downloading and processing Zip Code Tabulation Areas(ZCTA)
# shapefiles of the entire US can be time consuming
library(downloader)
library(sf)
library(data.table)
library(withr)
library(RcensusPkg)

# Get the sf geometries for the Zip Code Tabulation Area in the county of
# Middlesex, MA near Boston
# Define a temporary, self deleting output folder for the downloaded shapefiles
output_dir <- withr::local_tempdir()
if(!dir.exists(output_dir)){
  dir.create(output_dir)
```

```

}

# Define the codes of interest in a dataframe
mun_zcta_df <- data.frame(
  name = c(
    "Carlisle", "Concord", "Bedford",
    "Lexington", "Lexington", "Lincoln",
    "Lincoln", "Sudbury", "Wayland",
    "Weston", "Waltham"
  ),
  zcta = c(
    "01741", "01742", "01730",
    "02420", "02421", "01773",
    "01731", "01776", "01778",
    "02493", "02451"
  )
)

# Define an expression to filter the downloaded shapefile
express <- expression(ZCTA5CE20 %in% mun_zcta_df$zcta)

# Get the sf geometries
mun_zcta_sf <- RcensusPkg::tiger_zctas_sf(
  vintage = 2020,
  general = TRUE,
  do_progress = TRUE,
  express = express,
  output_dir = output_dir,
  delete_files = FALSE
)

## End(Not run)

```

---

vote2020

*vote2020 data*

---

### Description

Dataframe shows national presidential voting results for 2020 by state. Letters "D" and "R" under the Party column indicate respectively Democrat or Republican win for a state.

### Usage

```
vote2020
```

### Format

An object of class `data.table` (inherits from `data.frame`) with 61 rows and 22 columns.

**Details**

**State** character string full state name  
**Party** character string with values "D" or "R"

---

wide_to_long	<i>wide_to_long</i>
--------------	---------------------

---

**Description**

Reshape a data frame from a "wide" format to a "long" format.

Function is a helper in calling data.table's melt() function to reshape a wide data frame to a long form.

**Usage**

```
wide_to_long(
  dt = NULL,
  id_v = c("NAME", "GEOID"),
  measure_v = NULL,
  variable_name = "variable",
  value_name = "estimate",
  na_rm = FALSE
)
```

**Arguments**

dt	The data frame with a wide collection of column variables. This parameter is required.
id_v	A character vector of column from dt that are not to be consolidated and act as identifier columns for the new long form. This parameter is required.
measure_v	An optional character vector that sets the column measures from dt that are to be consolidated. If not specified then all the columns not in id_v are considered measures and will be consolidated.
variable_name	An optional string that sets the column name for the consolidated column names.
value_name	An optional string that sets the column name for the consolidated values.
na_rm	An optional logical which if TRUE will remove rows with NA values.

**Value**

A reshaped data frame in the "long" format.

**Examples**

```
## Not run:
# Requires Census Bureau API key
library(data.table)
library(downloader)
library(jsonlite)
library(RcensusPkg)
# Request for data from Census Bureau which comes in the "wide" form
B19001_1yr_wide_dt <- RcensusPkg::get_vintage_data(
  dataset = "acs/acs1",
  vintage = 2016,
  group = "B19001",
  region = "state"
)
# Convert the returned data.table into "long" form
B19001_1yr_long_dt <- RcensusPkg::wide_to_long(
  dt = B19001_1yr_wide_dt,
  id_v = c("NAME", "GEOID")
)
## End(Not run)
```

# Index

## \* datasets

vote2020, [54](#)

get\_category\_strings, [2](#)

get\_dataset\_names, [3](#)

get\_geography, [5](#)

get\_groups, [5](#)

get\_idb\_data, [6](#)

get\_multi\_vintage\_data, [8](#)

get\_variable\_names, [10](#)

get\_vintage\_data, [12](#)

join\_it, [14](#)

long\_to\_wide, [16](#)

plot\_us\_data, [17](#)

remove\_area\_water, [21](#)

tiger\_block\_groups\_sf, [24](#)

tiger\_blocks\_sf, [22](#)

tiger\_cbsa\_sf, [27](#)

tiger\_counties\_sf, [29](#)

tiger\_county\_subsection\_sf, [32](#)

tiger\_landmarks\_sf, [34](#)

tiger\_places\_sf, [37](#)

tiger\_roads\_sf, [39](#)

tiger\_states\_sf, [42](#)

tiger\_tracts\_sf, [45](#)

tiger\_urban\_area\_sf, [47](#)

tiger\_water\_sf, [49](#)

tiger\_zctas\_sf, [52](#)

vote2020, [54](#)

wide\_to\_long, [55](#)