Package 'totalcensus'

February 6, 2018

Type Package

Title Extract High Resolution Census Data

Version 0.3.0

Author Guanglai Li

Maintainer Guanglai Li <liguanglai@gmail.com>

Date 2018-01-20

Description Download summary files from Census Bureau <https://www2.census.gov/> and extract data, in particular high resolution data at block, block group, and tract level, from decennial census and American Community Survey 1-year and 5-year estimates.

URL https://github.com/GL-Li/totalcensus

BugReports https://github.com/GL-Li/totalcensus/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.2.1)

Imports stringr (>= 1.2.0), data.table (>= 1.10.1), magrittr (>= 1.5), purrr (>= 0.2.4), utils (>= 3.3.2)

Suggests knitr, rmarkdown, ggmap, ggplot2

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-02-06 15:16:39 UTC

R topics documented:

`convert_fips_to_names` *convert fips codes to names of a geographies*

Description

convert fips codes to names of a geographies

Usage

```
convert_fips_to_names(FIPs, states = NULL, geo_header = "STATE",
  in_states = NULL)
```

Arguments

FIPs	string vector of fips code such as c("021", "002")
states	string vector of state abbreviations having same length as FIPs
geo_header	string, taking values of "STATE", "COUNTY", "PLACE", "COUSUB" or "CBSA".
in_states	which states are these FIPs generated from. Use state abbrevations or "US" for national. Vector of unique states.

Value

vector of names corresponding to FIPs and states

Examples

```
aaa <- convert_fips_to_names(c("11", "44"))
# [1] "DC" "RI"

bbb <- convert_fips_to_names(c("001", "013"), states = c("RI", "MA"), geo_header = "COUNTY")
# [1] "Bristol County" "Hampden County"

## Not run:
# The only purpose of downloading data to temporary directory is to test the
# following examples. Do not do it when you have data in your local computer.
tmp <- tempdir()
url <- "https://s3.amazonaws.com/gl-shared-data/generated_census_data.zip"
download.file(url, paste0(tmp, "/tmp.zip"))
unzip(
    paste0(tmp, "/tmp.zip"),
    exdir = paste0(tmp, "/generated_data")
)
Sys.setenv(PATH_TO_CENSUS = tmp)

convert_fips_to_names(
    FIPs = c("14140", "76030"),
    states = c("RI", "MA"),
    geo_header = "PLACE",
    in_states = c("RI", "MA")
)
# [1] "Central Falls city" "Westfield city"

convert_fips_to_names(
    FIPs = c("39300", "46740"),
    states = c(NA, NA),
    geo_header = "CBSA",
    in_states = "US"
)
# [1] "Providence-New Bedford-Fall River, RI-MA Metro Area" "Valley, AL Micro Area"

## End(Not run)
```

dict_acs_geocomponent *List of geographic components used in American Community Survey*

Description

This dataset contains the geographic components and codes used in 1-year and 5-year surveys. Search geographic components with function `search_geoheaders`.

Usage

```
dict_acs_geocomponent
```

Format

A data.table with 19 rows and 6 variables:

code code for the geocomponent, such as "01" and "M3"

geo_component description of the geographic component

state_acs1 wheather a geocomponent available in 1-year survey state files

US_acs1 wheather a geocomponent available in 1-year survey national files

state_acs5 wheather a geocomponent available in 5-year survey state files

US_acs5 wheather a geocomponent available in 5-year survey national files

Source

generated from lookup files

dict_acs_geoheader *List of geographic headers used in American Community Survey*

Description

This dataset has the complete list of geographic header references and their discription used in ACS 1-year and 5-year summary file. Search the dataset with function `search_geoheaders`.

Usage

```
dict_acs_geoheader
```

Format

A data.table with 53 rows and 4 variables

reference reference of the geoheader

field description of the geoheader

start starting position of the geoheader in geography file

end ending position of the geoheader in geography file

Source

2016 ACS Summary File technical documentation, page 10 - 11.

---

dict_acs_summarylevel *Summary levels available in American Community Survey*

Description

This data contains summary levels and codes used in 1-year and 5-year survey. Search with function `search_summarylevels`.

Usage

```
dict_acs_summarylevel
```

Format

A data.table with 87 rows and 6 variables

code code of summary level

summary_level description of summary level

state_acs1 wheather a summary level available in 1-year survey state files

US_acs1 wheather a summary level available in 1-year survey national files

state_acs5 wheather a summary level available in 5-year survey state files

US_acs5 wheather a summary level available in 5-year survey national files

Source

generated from lookup datasets

---

dict_acs_table *Complete list of ACS tables*

Description

This dataset contains all tables in 1-year and 5-year survey. Note that 5-year survey in 2014 and 2015 has the identical tables.

Usage

```
dict_acs_table
```

Format

A data.table with 1324 rows and 6 variables:

table_number table number such as "C27013"

table_name description of the table

acs5_2015 table available in 2015 ACS 5-year survey

acs1_2016 table available in 2016 ACS 1-year survey

acs1_2015 table available in 2015 ACS 1-year survey

acs1_2014 table available in 2014 ACS 1-year survey

Source

generated from lookup datasets.

`dict_all_geocomponent` *List of geographic components and codes*

Description

This dataset contains all available geographic components and codes.

Usage

```
dict_all_geocomponent
```

Format

A data.table with 96 rows and 4 variables:

code code for the geocomponent, such as "01" and "M3"

geo_component description of the geographic component

Source

2010 Census Summary File 1 technical documentation page 6-15

`dict_cbsa` *List CBSA code of Metropolitan Statistical Area/Micropolitan Statistical Area*

Description

This dataset contains Metropolitan Statistical Area/Micropolitan Statistical Area CBSA code and title, plus associated metrodivision, CSA, state, and county code. Search for CBSA with function `search_cbsa`.

Usage

```
data("dict_cbsa")
```

Format

A data.table with 1882 rows and 12 variables:

CBSA CBSA code

CBSA_title CBSA title

state_full full name of the state. A cbsa could include multiple states

county county or county equivalent

CSA code of the CSA to which the CBSA belongs

CSA_title CSA title

METDIV metro division code

METDIV_title metro division title

metro_micro is the CBSA a metropolitan or a micropolitan statistic area

STATE FIPS of the state

COUNTY FIPS of the county

central_outlying is the counry a central or outlying county in the CBSA

Source

List of CBSA

`dict_decennial_geocomponent`

List of geographic components and codes

Description

This dataset contains the geographic components and codes used in Census 2010 summary file 1 (with urban/rural update). Search geographic components with function `search_geocomponents`.

Usage

```
dict_decennial_geocomponent
```

Format

A data.table with 96 rows and 4 variables:

code code for the geocomponent, such as "01" and "M3"

geo_component description of the geographic component

in_state_file wheather the geocomponent available in state files

in_US_file wheather the geocomponent available in national files

Source

2010 Census Summary File 1 technical documentation page 6-15

`dict_decennial_geoheader`

List of geographic headers

Description

This dataset has the complete list of geographic header references and their discription used in Census 2010 summary file 1 (with urban/rural update). Search the dataset with function `search_geoheaders`.

Usage

```
dict_decennial_geoheader
```

Format

A data.table with 101 rows and 4 variables

reference reference of the geoheader record

field description of the geoheader record field

start starting position of the geoheader in the record

end ending position of the geoheader in the record

Source

2010 Census Summary File 1 technical documentation page 2-8

`dict_decennial_summarylevel`
Summary levels available in Census 2010

Description

This data contains summary levels and codes used in census 2010 summary file 1 (with urban/rural update). Search with function `search_summarylevels`.

Usage

```
dict_decennial_summarylevel
```

Format

A data.table with 165 rows and 4 variables

code code of summary level

summary_level description of summary level

in_state_file wheather the summary level available in state files

in_US_file wheather the summary level available in national files

Source

2010 Census Summary File 1 technical documentation page 4-16 state summary file with urban/rural update

---

`dict_decennial_table` *Complete list of census tables*

Description

This dataset contains all census tables in census 2010 summary file 1 (with urban/rural update).

Usage

```
dict_decennial_table
```

Format

A data.table with 333 rows and 4 variables:

table_ref reference code such as "H0010", "PCT022G"

table_number table number such as "H1", "PCT22G"

table_name description of the table

universe universe of the data

Source

2010 Census Summary File 1 technical documentation all across chapter 5.

`dict_fips` *List of FIPS code as of 2016 in the US*

Description

This dataset contains a list of FIPS of states, counties, county subdivisions, places, consolidated cities, and their names and summary levels as well as full name and abbreviation of state. It does NOT contain FIPS of many small areas. Search for FIPS with function `search_fips`.

Usage

```
data("dict_fips")
```

Format

A data.table with 43934 rows and 9 variables:

state_full full name of a state such as "Alabama"

state_abbr abbreviation of a state such as "AL"

STATE FIPS code of the state

SUMLEV summary level of the entry in the row

COUNTY FIPS code of county

CUSUB FIPS of COUnty SUBdivision

PLACE FIPS code of place

CONCIT FIPS code of CONsolidated CITy

NAME name of the entry in the row

Source

List of FIPS as of 2016

`download_census` *download census data*

Description

Download census data from United States Census bureau. By default it downloads summary files and extract summary files of Census 2010, ACS 5-year survey of 2015, and ACS 1-year survey of 2016, 2015, and 2014. It also download generated data from Census 2010 if not exist.

Usage

```
download_census(survey = NULL, year = NULL, states = c(states_DC, "US",
  "PR"))
```

Arguments

`survey`	Which survey to download from, "decennial", "acs5year", or "acs1year"
`year`	year or ending year of the survey
`states`	vector of abbreviations of states such as c("MA", "RI")

`download_generated_data`
Download data generated from Census 2010

Description

This function downloads data generated from Census 2010 from Census 2010.

Usage

```
download_generated_data()
```

lookup_acs1year_2014 *ACS 1-year 2014 file segment and table lookup data*

Description

There is slightly difference in the lookup tables of each year.

Usage

```
lookup_acs1year_2014
```

Format

A data.table with 31711 rows and 6 variables

file_segment sequence number of segment data files, from "0001" to "0165"

table_content description of columns in a table

reference reference of the table content, such as "B01001_002". The reference is used to extract data of table content.

table_number table number such as "B01001"

table_name description of table. A table has multiple columns (table_content)

universe the universe of the data

Source

Check for each year of ACS 1-year and 5-year Sequence Number/Table Number Lookup File.

lookup_acs1year_2015 *ACS 1-year 2015 file segment and table lookup data*

Description

There is slightly difference in the lookup tables of each year.

Usage

```
lookup_acs1year_2015
```

Format

A data.table with 31751 rows and 6 variables

file_segment sequence number of segment data files, from "0001" to "0165"

table_content description of columns in a table

reference reference of the table content, such as "B01001_002". The reference is used to extract data of table content.

table_number table number such as "B01001"

table_name description of table. A table has multiple columns (table_content)

universe the universe of the data

Source

Check for each year of ACS 1-year and 5-year Sequence Number/Table Number Lookup File.

---

lookup_acs1year_2016 *ACS 1-year 2016 file segment and table lookup data*

Description

There is slightly difference in the lookup tables of each year.

Usage

```
lookup_acs1year_2016
```

Format

A data.table with 31835 rows and 6 variables

file_segment sequence number of segment data files, from "0001" to "0166"

table_content description of columns in a table

reference reference of the table content, such as "B01001_002". The reference is used to extract data of table content.

table_number table number such as "B01001"

table_name description of table. A table has multiple columns (table_content)

universe the universe of the data

Source

Check for each year of ACS 1-year and 5-year Sequence Number/Table Number Lookup File.

lookup_acs5year_2015 *ACS 5-year 2015 file segment and table lookup data*

Description

ACS 5-year 2015 file segment and table lookup data

Usage

```
lookup_acs5year_2015
```

Format

A data.table with 22910 rows and 6 variables

file_segment sequence number of segment data files, from "0001" to "0122"

table_content description of columns in a table

reference reference of the table content, such as "B01001_002". The reference is used to extract data of table content.

table_number table number such as "B01001"

table_name description of table. A table has multiple columns (table_content)

universe the universe of the data

Source

Check for each year of ACS 1-year and 5-year Sequence Number/Table Number Lookup File.

lookup_acs5year_2016 *ACS 5-year 2015 file segment and table lookup data*

Description

ACS 5-year 2015 file segment and table lookup data

Usage

```
lookup_acs5year_2016
```

Format

A data.table with 22910 rows and 6 variables

file_segment sequence number of segment data files, from "0001" to "0122"

table_content description of columns in a table

reference reference of the table content, such as "B01001_002". The reference is used to extract data of table content.

table_number table number such as "B01001"

table_name description of table. A table has multiple columns (table_content)

universe the universe of the data

Source

Check for each year of ACS 1-year and 5-year Sequence Number/Table Number Lookup File.

`lookup_decennial_2010` *Lookup data files and table contents of Census 2010*

Description

This dataset includes all data fields of data files in census 2010 summary file 1 (with urban/rural update). Fucntion `search_tablecontents` searches content in this dataset.

Usage

```
lookup_decennial_2010
```

Format

A data.table with 9199 rows and 6 variables:

file_segment sequence number of segment data files, from 1 to 48

table_content description of columns in a decennial table

reference reference of table content, such as "PCT0240019"

table_number table number such as "H1", "PCT22G"

table_name description of table, which has many table_content

universe the universe of the decennial data

Source

2010 Census Summary File 1 technical documentation all across chapter 6.

read_acs1year *Read ACS 1-year estimates*

Description

This function retrieves data from summary file of ACS 1-year estimates. In addition to selected geographic headers and table contents, it also returns total population and coordinates of selected geographic areas, as well as summary levels and geographic components.

Usage

```
read_acs1year(year, states, table_contents = NULL, areas = NULL,
  geo_headers = NULL, summary_level = "*", geo_comp = "total",
  with_margin = FALSE, with_acsgeoheaders = FALSE, show_progress = TRUE)
```

Arguments

year	year of the estimate
states	vector of state abbreviations, for example "IN" or c("MA", "RI").
table_contents	selected references of contents in census tables. Users can choose a name for each reference, such as in c("abc = B01001_009", "fff = B00001_001"). Try to make names meaningful. To find the references of table contents of interest, search with function search_tablecontents.
areas	For metro area, in the format like "New York metro". For county, city, or town, must use the exact name as those in dict_fips in the format like "kent county, RI", "Boston city, MA", and "Lincoln town, RI". And special examples like "Salt Lake City city, UT" must keep the "city" after "City".
geo_headers	vector of references of selected geographci headers to be included in the return. Browse geoheaders in dict_acs_geoheader or search with search_geoheaders
summary_level	select which summary level to keep, "*" to keep all. It takes strings including "state", "county", "county subdivision", "place", "tract", "block group", and "block" for the most common levels. It also take code. Search all codes with search_summarylevels or browse dict_acs_summarylevel .
geo_comp	select which geographic component to keep, "*" to keep every geo-component, "total" for "00", "urban" for "01", "urbanized area" for "04", "urban cluster" for "28", "rural" for "43". Others should input code which can be found with search_geocomponents. Availability of geocomponent depends on summary level. State level contains all geographic component. County subdivision and higher level have "00", "01", and "43". Census tract and lower level have only "00".
with_margin	read also margin of error in addition to estimate
with_acsgeoheaders	whether to keep geographic headers from ACS data
show_progress	whether to show progress in fread()

Value

A data.table of selected data.

Examples

```
## Not run:
# read summary data using areas of selected cities
aaa <- read_acs1year(
    year = 2016,
    states = c("UT", "RI"),
    table_contents = c("male = B01001_002", "female = B01001_026"),
    areas = c("Salt Lake City city, UT",
              "Providence city, RI",
              "PLACE = RI19180"),
    summary_level = "place",
    with_margin = TRUE
)


# read data using geoheaders - all major counties
bbb <- read_acs1year(
    year = 2015,
    states = c("UT", "RI"),
    table_contents = c("male = B01001_002", "female = B01001_026"),
    geo_headers = c("COUNTY"),
    summary_level = "county",
    with_margin = TRUE
)

## End(Not run)
```

---

read_acs5year *Read ACS 5-year estimates*

Description

This function retrieves data from summary file of ACS 5-year estimates. In addition to selected geographic headers and table contents, it also returns total population and coordinates of selected geographic areas, as well as summary levels and geographic components.

Usage

```
read_acs5year(year, states, table_contents = NULL, areas = NULL,
  geo_headers = NULL, summary_level = "*", geo_comp = "total",
  with_margin = FALSE, with_acsgeoheaders = FALSE, show_progress = TRUE)
```

Arguments

year	end year of the 5-year estimate
states	vector of state abbreviations, for example "IN" or c("MA", "RI").
table_contents	selected references of contents in census tables. Users can choose a name for each reference, such as in c("abc = B01001_009", "fff = B00001_001"). Try to make names meaningful. To find the references of table contents of interest, search with function search_tablecontents.
areas	For metro area, in the format like "New York metro". For county, city, or town, must use the exact name as those in dict_fips in the format like "kent county, RI", "Boston city, MA", and "Lincoln town, RI". And special examples like "Salt Lake City city, UT" must keep the "city" after "City".
geo_headers	vector of references of selected geographci headers to be included in the return, like "COUNTY" or c("PLACE", "CBSA"). Browse geoheaders in dict_acs_geoheader or search with search_geoheaders
summary_level	select which summary level to keep, "*" to keep all. It takes string including "state", "county", "county subdivision", "place", "tract", "block group", and "block" for the most common levels. It also take code. Search all codes with search_summarylevels or browse dict_acs_summarylevel .
geo_comp	select which geographic component to keep, "*" to keep every component, "total" for "00", "urban" for "01", "urbanized area" for "04", "urban cluster" for "28", "rural" for "43". Others should be input as code which can be found with search_geocomponents. Availability of geocomponent depends on summary level. State level contains all geographic component. County subdivision and higher level have "00", "01", and "43". Census tract and lower level have only "00".
with_margin	read also margin of error in addition to estimate if TRUE
with_acsgeoheaders	whether to keep geographic headers from ACS data
show_progress	whether to show progress in fread()

Value

A data.table of selected data.

Examples

```
## Not run:
# read data using areas
aaa <- read_acs5year(
    year = 2015,
    states = c("UT", "RI"),
    table_contents = c(
        "white = B02001_002",
        "black = B02001_003",
        "asian = B02001_005"
    ),
    areas = c(
```

```
        "Lincoln town, RI",
        "Salt Lake City city, UT",
        "Salt Lake City metro",
        "Kent county, RI",
        "COUNTY = UT001",
        "PLACE = UT62360"
    ),
    summary_level = "block group",
    with_margin = TRUE
)


# read data using geoheaders
bbb <- read_acs5year(
    year = 2015,
    states = c("UT", "RI"),
    table_contents = c("male = B01001_002", "female = B01001_026"),
    geo_headers = "PLACE",
    summary_level = "block group"
)

## End(Not run)
```

`read_decennial` *Read decennial census data*

Description

This function retrieves data from summary file 1 (with urban/rural update) of decennial censuses. In addition to selected geographic headers and table contents, it also returns total population and coordinates of selected geographic areas, as well as summary levels and geographic components.

Usage

```
read_decennial(year, states, table_contents = NULL, areas = NULL,
  geo_headers = NULL, summary_level = "*", geo_comp = "total",
  show_progress = TRUE)
```

Arguments

`year`	year of the decennial census
`states`	vector of state abbreviations, for example "IN" or c("MA", "RI").
`table_contents`	selected references of contents in census tables. Users can choose a name for each reference, such as in c("abc = PCT012F139", "fff = P0030008", "rural_p = P0020005"). Try to make names meaningful. To find the references of table contents of interest, search with function `search_tablecontents`.

areas — For metro area, in the format like "New York metro". For county, city, or town, must use the exact name as those in dict_fips in the format like "kent county, RI", "Boston city, MA", and "Lincoln town, RI". And special examples like "Salt Lake City city, UT" must keep the "city" after "City".

geo_headers — vector of references of selected geographci headers to be included in the return. Browse geoheaders in dict_decennial_geoheader or search with search_geoheaders

summary_level — select which summary level to keep, "*" to keep all. It takes strings including "state", "county", "county subdivision", "place", "tract", "block group", and "block" for the most common levels. It also take code for level. Search all codes with search_summarylevels or browse dict_decennial_summarylevel .

geo_comp — select which geographic component to keep, "*" to keep every geo-component, "total" for "00", "urban" for "01", "urbanized area" for "04", "urban cluster" for "28", "rural" for "43". Others should input code which can be found with search_geocomponents. Availability of geocomponent depends on summary level. State level contains all geographic component. County subdivision and higher level have "00", "01", and "43". Census tract and lower level have only "00".

show_progress — show progress of file reading if TRUE. Turn off if FALSE, which is useful in RMarkdown output.

Value

A data.table whose columns include the selected geoheaders and table contents plus SUMLEV, GEOCOMP, and state.

Examples

```
## Not run:
# read one table and one area from one state
aaa = read_decennial(
    year = 2010,
    states = "UT",
    table_contents = c("urban = P0020002", "rural = P0020005"),
    geo_headers = "CBSA",
    summary_level = "tract"
)


# read multiple table contents and areas from multiple states
bbb = read_decennial(
    year = 2010,
    states = c("UT", "RI"),
    table_contents = c("urban = P0020002", "rural = P0020005"),
    areas = c(
        "place = ut62360",
        "Providence city, RI",
        "cousub = ri41500",
        "cbsa = 39300"
    ),
    summary_level = "block"
```

```
)


# read table contents of all county subdivisions in Providence metro
library(data.table)
library(magrittr)
ccc <- read_decennial(
    year = 2010,
    states = "US",
    table_contents = c("urban = P0020002", "rural = P0020005"),
    geo_headers = c("NAME", "CBSA"),
    summary_level = "county subdivision",
    geo_comp = "*"
) %>%
    .[CBSA == "39300"]

## End(Not run)
```

`search_cbsa` *Search CBSA code and title*

Description

Search CBSA code of Core Based Statistical Area in dataset `dict_cbsa`. The search also returns which CSA (Combined Statistical Area) that contains the CBSA. If the CBSA contains multiple counties, each county is returned as a row.

Usage

```
search_cbsa(keyword = "*", view = TRUE)
```

Arguments

`keyword`	keyword to be searched in CBSA or CBSA title.
`view`	display the search result with View if TRUE.

Details

Quite often, multiple rows are returned. It is necessary to hand pick the right one you are really looking for.

Value

A data.table

Examples

```
# Change view = TRUE (default) to View the returned data.
aaa <- search_cbsa("providence", view = FALSE)

bbb <- search_cbsa("new york", view = FALSE)

## Not run:
  # view all CBSA code
  search_cbsa()

## End(Not run)
```

---

search_fips *Search FIPS codes*

Description

Search FIPS code of a states, counties, county subdivisions, places, or consolidated cities in dataset dict_fips. The search also returns summary levels.

Usage

```
search_fips(keyword = "*", state = NULL, view = TRUE)
```

Arguments

keyword	keyword to be searched in NAMES or FIPS.
state	abbreviation of a state.
view	display the search result with View if TRUE.

Details

Quite often, multiple rows are returned. It is necessary to hand pick the right one you are really looking for.

The function search_fips has changed summary level 061 to 060, and 162 to 160 in search results. The summary levels in dict_fips are 010, 040, 050, 061, 162, and 170. The level 061 is for Minor Civil Division (MCD)/Census County Division (CCD) (10,000+). It does not appear in dict_decennial_summarylevel and dict_acs_summarylevel, which instead have 060 for County Subdivision. Level 061 is part of 060 and is replaced with 060 in order to use the census data. Similarly, both level 162 in dict_fips and l60 in dict_decennial_summarylevel and dict_decennial_summarylevel are for State-Place. Always use 160 in census data.

Value

A data.table

Examples

```
# Change view = TRUE (default) to View the returned data.table.

# Search fips of Lincoln in Rhode Island.
aaa <- search_fips("lincoln", "RI", view = FALSE)

# search FIPS number in all states
bbb <- search_fips("08375", view = FALSE)

## Not run:
  # view all fips code
  search_fips()

## End(Not run)
```

`search_geocomponents` *Search geographic components*

Description

Search the code or content of geographic components

Usage

```
search_geocomponents(survey, keyword = "*", view = TRUE)
```

Arguments

`survey`	"decennial" or "acs"
`keyword`	keyword to search in code or description, "*" for any words.
`view`	display the search result with View if TRUE

Details

The most frequently used geographic components are:

00 : all geographic component

01 : urban

43 : rural

Value

A data.table

Examples

```
# Change view = TRUE (default) to View the returned data.
aaa <- search_geocomponents("decennial", "urban", view = FALSE)
bbb <- search_geocomponents("acs", "43", view = FALSE)

## Not run:
  # view all geocomponents
  search_geocomponents("decennial")
  search_geocomponents("acs")

## End(Not run)
```

`search_geoheaders` *Search geographic headers*

Description

Search in field reference or description of geographic header with keyword in dataset `dict_decennial_geoheader` or `dict_acs_geoheader`.

Usage

```
search_geoheaders(survey, keyword = "*", view = TRUE)
```

Arguments

`survey`	type of survey, either "decennial" or "acs".
`keyword`	keyword in description or reference. The default "*" includes all geoheaders.
`view`	display the search result with View() if TRUE

Value

data.table matching the search criteria

Examples

```
# Change view = TRUE (default) to View the returned data.
# search geoheader that contains keyword "india" in decennial 2010
aaa <- search_geoheaders("decennial", "india", view = FALSE)

# search for lattitude
bbb <- search_geoheaders("decennial", "latitu", view = FALSE)


## Not run:
  # browse all geoheaders in ACS in View()
  search_geoheaders("acs")
```

```
## End(Not run)
```

`search_summarylevels` *Search summary levels*

Description

Search code or description of summary levels

Usage

```
search_summarylevels(survey, keyword = "*", view = TRUE)
```

Arguments

`survey`	"decennial" or "acs"
`keyword`	keyword to search in code or description
`view`	display the search result with View if TRUE

Value

A data.table of searched results.

Examples

```
# Change view = TRUE (default) to View the returned data.
aaa = search_summarylevels("decennial", "block", view = FALSE)
bbb <- search_summarylevels("acs", "40", view = FALSE)

## Not run:
  # view all summary levels
  search_summarylevels("decennial")
  search_summarylevels("acs")

## End(Not run)
```

search_tablecontents *Search table contents in data files*

Description

Search in lookup datasets of each survey to find references of for table_contents argument in function read_decennial, read_acs1year, and read_acs5year.

Usage

```
search_tablecontents(survey, keyword = "*", year = NULL, view = TRUE)
```

Arguments

survey	either "decennial" for decenial or "acs" or American Community Survey.
keyword	keyword to be searched
year	ending year of the survey
view	display the search result with View if TRUE

Value

A data.table

Examples

```
# Change view = TRUE (default) to View the returned data.
# search by what you want
aaa <- search_tablecontents("decennial", "federal prison", view = FALSE)

# search by table reference
bbb <- search_tablecontents("acs", "B02003", view = FALSE)

## Not run:
  # view all decennial census table contents
  search_tablecontents("decennial")

  # view all ACS table contents
  search_tablecontents("acs")

## End(Not run)
```

search_tables *search decennial and acs tables by keyword in table numbers or table descriptions*

Description

search decennial and acs tables by keyword in table numbers or table descriptions

Usage

```
search_tables(survey, keyword = "*", view = TRUE)
```

Arguments

survey	"decennial" or "acs"
keyword	keyword to search in code or description.
view	display the search result with View if TRUE

Value

A data.table

Examples

```
# Change view = TRUE (default) to View the returned data.
aaa <- search_tables("decennial", "occupancy", view = FALSE)
bbb <- search_tables("acs", "detailed race", view = FALSE)

## Not run:
  # view all tables
  search_tables("decennial")
  search_tables("acs")

## End(Not run)
```

set_path_to_census *Set file path to directory storing downloaded census data*

Description

Set file path to directory storing downloaded census data

Usage

```
set_path_to_census(path)
```

Arguments

path	path to directory holding all downloaded census data, such as "E:/my_census_data" and "~/my_census_data/".

---

states_DC *Vector of the abbreviations of 50 states and DC*

Description

Abbrivation only

Usage

```
data("states_DC")
```

Format

A vector of 51 element

Index