Package 'UScensus2010blocks'

December 4, 2015

Type Package

Title USA Census Bureau 2010 block-level data (pop, area, urban, lat/lon)

Version 0.2.0

Date 2015-05-29

Author ejanalyst

Maintainer info@ejanalysis.com <info@ejanalysis.com>

Description The entire set of over 11 million Census blocks in the United States of America as a single data.frame, with just population count, FIPS code, latitude and longitude, area (size), and whether the block is urban.

License None

URL http://ejanalysis.github.io

http://www.ejanalysis.com/

LazyData TRUE

R topics documented:

`UScensus2010blocks-package`

Census Bureau 2010 block-level data (pop, area, urban, lat/lon) easily compiled into a single data.frame

---

Description

These datasets provide population count, size of block (area), latitude and longitude of internal point, whether the block is urban, for each US block, based on Census Bureau Census 2010 data. All States/DC are compiled into a single vector for each field (e.g. pop).

References

http://ejanalysis.github.io
http://www.ejanalysis.com/

Examples

```
## Not run:
   blocks <- get.blocks(charfips=FALSE)
   by(1e6 * blocks$pop / blocks$area, INDICES=blocks$urban, FUN=mean)

## End(Not run)
```

---

blocks.area *area: Over 11 million Census Bureau 2010 block-level values in a single data.frame*

Description

These data sets provide population count, size of block (area), latitude and longitude of internal point, whether the block is urban, for each US block, based on Census Bureau Census 2010 data, each of these fields as a single data file (RData), all sorted in the same order, enabling quick combination into a data.frame. All States/DC are compiled into a single data.frame.

Usage

```
blocks <- get.blocks()
  # or to load into memory just this one vector:
  data(blocks.area)
```

Format

A vector with 11078297 elements (Census 2010 blocks). If all the related datasets are compiled as a blocks data.frame, they provide the following:

1. "fips" (numeric - can be converted to character with leading zeroes via lead.zeroes(blocks$fips, 15)
2. "pop" (integer) - Population count in Census 2010
3. "urban" (logical)
4. "lat" (numeric) - decimal degrees
5. "lon" (numeric) - decimal degrees
6. "area" (numeric) - units? Need to check. ****

Source

2010 Census from Census Bureau http://www.census.gov obtained 2014/2015 compiled from multiple Census files of State-level population, area, internal point, or urban code. Slightly modified to store FIPS as numeric field, pop as integer, and urban as logical, to save RAM.

See Also

See get.blocks in **UScensus2010blocks** to assemble this and other fields into a blocks data.frame. See the UScensus2010 package and related datasets, some of which are on CRAN and others only here: http://lakshmi.calit2.uci.edu/census2000/ but note that package provides spatial data in a single file per State, while this package provides non-spatial data (just lat/lon) that can quickly be assembled into a single large data.frame.

---

blocks.fips *fips: Over 11 million Census Bureau 2010 block-level values in a single data.frame*

Description

These data sets provide population count, size of block (area), latitude and longitude of internal point, whether the block is urban, for each US block, based on Census Bureau Census 2010 data, each of these fields as a single data file (RData), all sorted in the same order, enabling quick combination into a data.frame. All States/DC are compiled into a single data.frame.

Usage

```
blocks <- get.blocks()
  # or to load into memory just this one vector:
  data(blocks.fips)
```

Format

A vector with 11078297 elements (Census 2010 blocks). If all the related datasets are compiled as a blocks data.frame, they provide the following:

1. "fips" (numeric - can be converted to character with leading zeroes via lead.zeroes(blocks$fips, 15)
2. "pop" (integer) - Population count in Census 2010
3. "urban" (logical)
4. "lat" (numeric) - decimal degrees
5. "lon" (numeric) - decimal degrees
6. "area" (numeric) - units? Need to check. ****

Source

2010 Census from Census Bureau http://www.census.gov obtained 2014/2015 compiled from multiple Census files of State-level population, area, internal point, or urban code. Slightly modified to store FIPS as numeric field, pop as integer, and urban as logical, to save RAM.

See Also

See `get.blocks` in **UScensus2010blocks** to assemble this and other fields into a blocks data.frame. See the `UScensus2010` package and related datasets, some of which are on CRAN and others only here: `http://lakshmi.calit2.uci.edu/census2000/` but note that package provides spatial data in a single file per State, while this package provides non-spatial data (just lat/lon) that can quickly be assembled into a single large data.frame.

`blocks.lat` *lat: Over 11 million Census Bureau 2010 block-level values in a single data.frame*

Description

These data sets provide population count, size of block (area), latitude and longitude of internal point, whether the block is urban, for each US block, based on Census Bureau Census 2010 data, each of these fields as a single data file (RData), all sorted in the same order, enabling quick combination into a data.frame. All States/DC are compiled into a single data.frame.

Usage

```
blocks <- get.blocks()
  # or to load into memory just this one vector:
  data(blocks.lat)
```

Format

A vector with 11078297 elements (Census 2010 blocks). If all the related datasets are compiled as a blocks data.frame, they provide the following:

1. "fips" (numeric - can be converted to character with leading zeroes via lead.zeroes(blocks$fips, 15)
2. "pop" (integer) - Population count in Census 2010
3. "urban" (logical)
4. "lat" (numeric) - decimal degrees
5. "lon" (numeric) - decimal degrees
6. "area" (numeric) - units? Need to check. ****

Source

2010 Census from Census Bureau `http://www.census.gov` obtained 2014/2015 compiled from multiple Census files of State-level population, area, internal point, or urban code. Slightly modified to store FIPS as numeric field, pop as integer, and urban as logical, to save RAM.

See Also

See `get.blocks` in **UScensus2010blocks** to assemble this and other fields into a blocks data.frame. See the `UScensus2010` package and related datasets, some of which are on CRAN and others only here: `http://lakshmi.calit2.uci.edu/census2000/` but note that package provides spatial data in a single file per State, while this package provides non-spatial data (just lat/lon) that can quickly be assembled into a single large data.frame.

`blocks.lon` *lon: Over 11 million Census Bureau 2010 block-level values in a single data.frame*

Description

These data sets provide population count, size of block (area), latitude and longitude of internal point, whether the block is urban, for each US block, based on Census Bureau Census 2010 data, each of these fields as a single data file (RData), all sorted in the same order, enabling quick combination into a data.frame. All States/DC are compiled into a single data.frame.

Usage

```
blocks <- get.blocks()
  # or to load into memory just this one vector:
  data(blocks.lon)
```

Format

A vector with 11078297 elements (Census 2010 blocks). If all the related datasets are compiled as a blocks data.frame, they provide the following:

1. "fips" (numeric - can be converted to character with leading zeroes via lead.zeroes(blocks$fips, 15)
2. "pop" (integer) - Population count in Census 2010
3. "urban" (logical)
4. "lat" (numeric) - decimal degrees
5. "lon" (numeric) - decimal degrees
6. "area" (numeric) - units? Need to check. ****

Source

2010 Census from Census Bureau http://www.census.gov obtained 2014/2015 compiled from multiple Census files of State-level population, area, internal point, or urban code. Slightly modified to store FIPS as numeric field, pop as integer, and urban as logical, to save RAM.

See Also

See `get.blocks` in **UScensus2010blocks** to assemble this and other fields into a blocks data.frame. See the `UScensus2010` package and related datasets, some of which are on CRAN and others only here: http://lakshmi.calit2.uci.edu/census2000/ but note that package provides spatial data in a single file per State, while this package provides non-spatial data (just lat/lon) that can quickly be assembled into a single large data.frame.

`blocks.pop` *pop: Over 11 million Census Bureau 2010 block-level values in a single data.frame*

Description

These data sets provide population count, size of block (area), latitude and longitude of internal point, whether the block is urban, for each US block, based on Census Bureau Census 2010 data, each of these fields as a single data file (RData), all sorted in the same order, enabling quick combination into a data.frame. All States/DC are compiled into a single data.frame.

Usage

```
blocks <- get.blocks()
  # or to load into memory just this one vector:
  data(blocks.pop)
```

Format

A vector with 11078297 elements (Census 2010 blocks). If all the related datasets are compiled as a blocks data.frame, they provide the following:

1. "fips" (numeric - can be converted to character with leading zeroes via lead.zeroes(blocks$fips, 15)
2. "pop" (integer) - Population count in Census 2010
3. "urban" (logical)
4. "lat" (numeric) - decimal degrees
5. "lon" (numeric) - decimal degrees
6. "area" (numeric) - units? Need to check. ****

Source

2010 Census from Census Bureau http://www.census.gov obtained 2014/2015 compiled from multiple Census files of State-level population, area, internal point, or urban code.

Population was obtained from files such as tabblock2010_01_pophu.dbf from within
http://www2.census.gov/geo/tiger/TIGER2010BLKPOPHU/tabblock2010_01_pophu.zip
as linked from here: http://www.census.gov/geo/maps-data/data/tiger-line.html

See http://www.census.gov/geo/maps-data/data/tiger.html for various related data products. See http://tigerweb.geo.census.gov/tigerwebmain/TIGERweb_county_based_files.html for html formatted versions. The data in this package is based on those TIGER files slightly modified to store FIPS as numeric field, pop as integer, and urban as logical, to save RAM.

See Also

See `get.blocks` in **UScensus2010blocks** to assemble this and other fields into a blocks data.frame. See the `UScensus2010` package and related datasets, some of which are on CRAN and others only here: http://lakshmi.calit2.uci.edu/census2000/ but note that package provides spatial data in a single file per State, while this package provides non-spatial data (just lat/lon) that can quickly be assembled into a single large data.frame.

`blocks.urban` *urban: Over 11 million Census Bureau 2010 block-level values in a single data.frame*

Description

These data sets provide population count, size of block (area), latitude and longitude of internal point, whether the block is urban, for each US block, based on Census Bureau Census 2010 data, each of these fields as a single data file (RData), all sorted in the same order, enabling quick combination into a data.frame. All States/DC are compiled into a single data.frame.

Usage

```
blocks <- get.blocks()
  # or to load into memory just this one vector:
  data(blocks.urban)
```

Format

A vector with 11078297 elements (Census 2010 blocks). If all the related datasets are compiled as a blocks data.frame, they provide the following:

1. "fips" (numeric - can be converted to character with leading zeroes via lead.zeroes(blocks$fips, 15)
2. "pop" (integer) - Population count in Census 2010
3. "urban" (logical)
4. "lat" (numeric) - decimal degrees
5. "lon" (numeric) - decimal degrees
6. "area" (numeric) - units? Need to check. ****

Source

2010 Census from Census Bureau http://www.census.gov obtained 2014/2015 compiled from multiple Census files of State-level population, area, internal point, or urban code. Slightly modified to store FIPS as numeric field, pop as integer, and urban as logical, to save RAM.

See Also

See `get.blocks` in **UScensus2010blocks** to assemble this and other fields into a blocks data.frame. See the `UScensus2010` package and related datasets, some of which are on CRAN and others only here: http://lakshmi.calit2.uci.edu/census2000/ but note that package provides spatial data in a single file per State, while this package provides non-spatial data (just lat/lon) that can quickly be assembled into a single large data.frame.

`get.blocks` *Get data.frame with data on all US Census 2010 blocks (pop, lat/lon, etc.)*

Description

Returns a large dataframe with one row per block. This helps assemble the desired fields for all 11m+ blocks, into a single data.frame.

Usage

```
get.blocks(fields = c("fips", "pop", "lat", "lon", "area", "urban"),
  charfips = TRUE)
```

Arguments

`fields`	Optional vector of character elements specifying which fields to return.
`charfips`	Optional TRUE by default, specifies if FIPS should be converted to character class with any necessary leading zeroes, which uses more RAM and takes much longer – It can take 1-2 minutes for this function to return results unless charfips=FALSE.

Details

Warning: It can take 1-2 minutes for this function to return results with default settings (i.e., unless charfips=FALSE is specified). The full blocks data.frame created by default uses approximately 1 GB of RAM. The blocks data.frame with just numeric fips and pop uses only about 133 MB and is

Value

Returns a (large, >11 million rows) dataframe that has specified fields or by default these 6 columns: fips, pop, lat, lon, area, urban

See Also

`blocks.fips` and **UScensus2010**

Examples

```
## Not run:
# To assemble blocks data.frame:
# 1) Much faster if charfips=FALSE, but
#    then cannot treat fips as character with leading zeroes where needed:
blocks <- get.blocks( charfips=FALSE )
# To convert numeric to character fips later:
blocks$fips <- lead.zeroes(blocks$fips, 15)
# 2) Slower way, but can get fips as character to begin with:
blocks <- get.blocks()
# To get just certain fields:
blocks <- get.blocks(c('fips','pop'))
# This function using defaults is the equivalent of the following:
#   require(UScensus2010blocks)
#   blocks <- data.frame(
```

```
#     fips=lead.zeroes(blocks.fips,15),
#     pop=blocks.pop,
#     lat=blocks.lat,
#     lon=blocks.lon,
#     area=blocks.area,
#     urban=blocks.urban
#   )

## End(Not run)
```

Index