

Package ‘USPopulationSampler’

September 12, 2026

Title Population-Weighted Sampling of Geographic Locations in the United States

Version 0.1.1

Description Generate geospatial locations within census block groups (BG; the smallest geographic unit for which population counts are available) within target counties, states, or across the entirety of the U.S. randomly selected according to population counts by using the Census Bureau reference data and optionally allow users to conduct temporal assignments on sampled locations using Covid-19 data.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports curl, parallel, sf, arrow, data.table, utils, stats

Suggests dplyr, usmap, ggplot2, tigris, patchwork, lubridate, lwgeom, testthat (>= 3.0.0), spelling

Config/testthat/edition 3

URL <https://github.com/Techavoan/USPopulationSampler>

BugReports <https://github.com/Techavoan/USPopulationSampler/issues>

Depends R (>= 3.5)

LazyData true

Language en-US

NeedsCompilation no

Author Rithy Techavoan Yean [aut, cre, cph],
Andrew Holbrook [aut, cph],
Jasen Zhang [ctb]

Maintainer Rithy Techavoan Yean <techavoan@gmail.com>

Repository CRAN

Date/Publication 2026-09-12 14:20:16 UTC

Contents

assign_dates_to_pts	2
covid_data_example	4
df_bg_example	4
load_bg_data	5
load_covid_data	6
load_synthetic_covid	7
open_synthetic_dataset	8
US_pop_sampler	9
Index	12

assign_dates_to_pts	<i>Assign dates to synthetic spatial points generated by US_pop_sampler() using observed COVID-19 temporal distributions.</i>
---------------------	---

Description

This function assigns dates to sampled spatial points using observed county-level COVID-19 date distributions. It supports both in-memory datasets and large streamed Parquet datasets.

Usage

```
assign_dates_to_pts(pts, covid_data, output_dir = NULL, seed = NULL)
```

Arguments

pts	One of: • a "synthetic_points" object returned by US_pop_sampler() • a data.frame or data.table containing sampled points • a character path to a directory containing Parquet files Sampled points must contain at minimum: fips 5-digit county FIPS code
covid_data	A data.frame containing at minimum: fips 5-digit county FIPS code date Observation date coercible to Date
output_dir	Optional directory where dated Parquet files will be written when pts is a streamed dataset. If NULL, a temporary directory is created.
seed	Optional integer seed for reproducible date assignment.

Details

Supported inputs include:

- in-memory "synthetic_points" objects returned by `US_pop_sampler()`
- streamed "synthetic_points" objects containing a path element
- `data.frames` or `data.tables` containing sampled points
- character paths to directories containing Parquet files

Dates are assigned hierarchically:

1. observed dates within the matching county FIPS code
2. observed dates within the matching state
3. observed dates from the full COVID-19 dataset

Dates are sampled from the corresponding observed distribution. Sampling is performed without replacement when enough observed dates are available. Otherwise sampling is performed with replacement.

County-level dates are used whenever available. If no matching county dates exist, dates are drawn from the corresponding state, and finally from the complete COVID-19 dataset.

When `pts` is stored as Parquet files, data are processed one file at a time and written directly to disk, allowing very large synthetic datasets to be dated without loading all points into memory simultaneously.

Value

A "synthetic_points" object may represent either an in-memory dataset or a streamed Parquet dataset.

If `pts` is supplied in memory, returns a `data.table` containing the original sampled points with an additional date column.

If `pts` refers to a Parquet directory, returns an object of class "synthetic_points" containing:

path Directory containing dated Parquet files

chunks Number of Parquet files written

fallback Character string describing the date assignment hierarchy.

Examples

```
data(df_bg_example)
data(covid_data_example)

# Generate synthetic points
pts <- US_pop_sampler(
  N = 100,
  df = df_bg_example,
  seed = 123
)

# Assign dates using the example COVID-19 dataset
```

```
pts <- assign_dates_to_pts(
  pts = pts,
  covid_data = covid_data_example,
  seed = 123
)

head(pts$date)
```

covid_data_example *Example COVID-19 county case data*

Description

A small example dataset containing county-level COVID-19 observations for Delaware, Rhode Island, and Vermont during January 2022. This dataset is small to avoid having to load the entire COVID-19 data frame.

Usage

```
covid_data_example
```

Format

A data frame containing:

fips Five-digit county FIPS code
date Date of observation.
county Name of County (stored as character)
state Name of State (stored as character)

Examples

```
data(covid_data_example)
head(covid_data_example)
```

df_bg_example *Example Census block group population data*

Description

A small subset of the 2020 Census block group population dataset used for examples and testing in **USPopulationSampler**.

Usage

```
df_bg_example
```

Format

A data frame with rows representing Census block groups and columns including:

GEO_ID 15-digit Census block group identifier.

NAMELSAD Block Group Name With Number Identification

STATEFP Two-digit state FIPS code.

COUNTYFP Three-digit county FIPS code.

pop 2020 Census population count for the block group.

TRACTCE 2020 Census Tract Information

BLKGRPCE Block Group Number Identification

geometry Simple feature geometry for the block group.

FIPS 5 Digit FIPS Code

Details

The dataset contains Census block groups from three states: Delaware, Rhode Island, and Vermont. Population counts are derived from the 2020 Decennial Census P1 table.

This dataset is intentionally small and is provided to demonstrate population-weighted sampling workflows without requiring download of the full national Census block group dataset.

Source

U.S. Census Bureau 2020 Decennial Census.

Examples

```
data(df_bg_example)
```

```
head(df_bg_example)
```

load_bg_data	<i>Downloads the preprocessed df_bg dataset containing 2020 Census block-group information, geometries, and population counts from Zenodo.</i>
--------------	--

Description

The file is cached locally so repeated calls do not trigger additional downloads unless `overwrite = TRUE`.

Usage

```
load_bg_data(
  destdir = tempdir(),
  filename = "US_Census_Blockgroup_Data_2020.rda",
  overwrite = FALSE
)
```

Arguments

destdir	Directory used to store downloaded files. Defaults to tempdir(). Users can provide a persistent directory to retain downloaded files across R sessions.
filename	Name of the locally cached .rda file.
overwrite	Logical; if TRUE, force re-download even when the file already exists.

Details

The downloaded file is validated before loading:

- File existence check
- Detection of accidental HTML downloads
- Verification that the object df_bg exists in the archive

Downloaded files are cached locally and reused in subsequent calls unless `overwrite = TRUE`.

Value

An sf object containing 2020 Census block-group geometries, population counts, and identifying geographic variables.

Examples

```
destdir <- tempfile()
dir.create(destdir)

bg_data <- load_bg_data(destdir = destdir)

unlink(destdir, recursive = TRUE)
```

load_covid_data	<i>Downloads the COVID-19 event dataset from Zenodo used by assign_dates_to_pts().</i>
-----------------	--

Description

The file is cached locally so repeated calls do not trigger additional downloads unless `overwrite = TRUE`.

Usage

```
load_covid_data(
  destdir = tempdir(),
  filename = "Covid_19_County_Case_Data.rda",
  overwrite = FALSE
)
```

Arguments

destdir	Directory used to store downloaded files. Defaults to tempdir(). Users can provide a persistent directory to retain downloaded files across R sessions.
filename	Name of the locally cached .rda file.
overwrite	Logical; if TRUE, forces a fresh download even if a cached copy already exists.

Details

Downloaded files are validated before loading by checking file existence, detection of accidental HTML downloads, and verification that the object covid_cases exists in the archive.

Value

A data.frame containing county-level COVID-19 observations used for temporal assignment.

See Also

[load_bg_data\(\)](#), [assign_dates_to_pts\(\)](#), [load_synthetic_covid\(\)](#)

Examples

```
destdir <- tempfile()
dir.create(destdir)

covid_data <- load_covid_data(destdir = destdir)

unlink(destdir, recursive = TRUE)
```

load_synthetic_covid	<i>Downloads one of the synthetic COVID-19 datasets stored as a Parquet file on Zenodo and loads it into memory.</i>
----------------------	--

Description

Files are cached locally so repeated calls do not trigger additional downloads unless overwrite = TRUE.

Usage

```
load_synthetic_covid(dataset, destdir = tempdir(), overwrite = FALSE)
```

Arguments

dataset	Integer between 1 and 28 indicating which synthetic dataset to download.
destdir	Directory used to store downloaded files. Defaults to tempdir(). Users can provide a persistent directory to retain downloaded files across R sessions.
overwrite	Logical; if TRUE, force re-download even when the file already exists.

Details

Synthetic datasets are hosted on Zenodo as compressed Parquet files. Downloaded files are cached locally and reused in subsequent calls unless `overwrite = TRUE`.

By default, downloaded files are stored in the temporary directory returned by `tempdir()`. To retain downloaded files across R sessions, users can provide a persistent directory through `destdir`.

Value

A `data.frame` containing the requested synthetic COVID-19 observations.

Examples

```
destdir <- tempfile()
dir.create(destdir)

sampled_dataset <- load_synthetic_covid(
  dataset = 1,
  destdir = destdir
)

head(sampled_dataset)

unlink(destdir, recursive = TRUE)
```

open_synthetic_dataset

Open a synthetic dataset stored in Parquet format

Description

Returns an Arrow Dataset object for lazy querying over partitioned Parquet files or returns an already materialized in-memory dataset.

Usage

```
open_synthetic_dataset(x)
```

Arguments

- | | |
|---|--|
| x | One of: <ul style="list-style-type: none"> • a character path to a directory containing Parquet files • a "synthetic_points" object returned by <code>US_pop_sampler(streaming = TRUE)</code> • a list-like object containing a path element pointing to a Parquet dataset • an existing Arrow Dataset |
|---|--|

Details

When supplied with a directory or Arrow dataset reference, the function returns an `arrow::Dataset` without loading the data into memory. If an in-memory "synthetic_points" object is supplied, it is returned unchanged.

Use `dplyr::collect()` to materialize an `arrow::Dataset` into memory.

Value

Either:

- an `arrow::Dataset` object for lazy querying, or
- the supplied in-memory `data.table` if `x` is already materialized.

Examples

```
if (requireNamespace("arrow", quietly = TRUE)) {
  tmp <- tempfile()

  arrow::write_dataset(
    data.frame(
      longitude = c(-118.2, -118.3),
      latitude = c(34.0, 34.1)
    ),
    path = tmp,
    format = "parquet"
  )

  ds <- open_synthetic_dataset(tmp)
  ds
}
```

US_pop_sampler	<i>Samples synthetic points from US census geographies (tracts or counties) with probabilities proportional to population.</i>
----------------	--

Description

Locations are generated uniformly within census block-group polygons. Sample counts are first allocated to block groups with probability proportional to population (optionally within user-defined state or county filters), and points are then drawn independently within each polygon. Input rows with missing (NA), non-finite, or non-positive population values are removed before sampling. Remaining rows are used as the sampling frame.

Usage

```
US_pop_sampler(
  N,
  df = NULL,
  state = NULL,
  fip_codes = NULL,
  parallel = FALSE,
  n_cores = NULL,
  seed = NULL,
  streaming = TRUE,
  stream_threshold = 1e+06,
  target_batch_points = NULL,
  output_dir = NULL
)
```

Arguments

N	Integer scalar or named vector of target specifying number of synthetic points to generate. If a scalar, points are allocated across all eligible polygons proportional to population. If a named vector: • with <code>state =</code> , names must correspond to state FIPS codes or state identifiers after cleaning • with <code>fip_codes =</code> , names must correspond to county FIPS codes
df	An sf object containing census polygons. Must include at minimum: • pop: numeric population weights • FIPS: 5-digit county FIPS code • STATEFP: 2-digit state FIPS code If GEO_ID is missing, FIPS will be used instead. If NULL, the function searches for an object named <code>df_bg</code> in the calling environment.
state	Optional character vector of state names or abbreviations. If provided, restricts sampling to block groups within the specified states; allocation is then performed within those subsets
fip_codes	Optional vector of county FIPS codes used (County Fip Codes 5 Digits) to subset and allocate sampling. County filtering restricts the pool of eligible block groups rather than sampling counties directly. Cannot be used simultaneously with <code>state</code> .
parallel	Logical; if TRUE, polygon sampling is distributed across multiple workers using the parallel package when beneficial.
n_cores	Integer number of cores to use when <code>parallel = TRUE</code> . If NULL, defaults to available cores minus one.
seed	Optional integer seed for reproducible random sampling.

<code>streaming</code>	Logical; if TRUE and the total sample size exceeds <code>stream_threshold</code> , sampled points are written to partitioned Parquet files instead of being returned in memory.
<code>stream_threshold</code>	Integer threshold above which streaming mode is automatically enabled. Default is 1e6.
<code>target_batch_points</code>	Approximate maximum number of points processed in each sampling and write batch. If NULL, an appropriate value is selected automatically.
<code>output_dir</code>	Directory where streamed Parquet files are written. If NULL, a temporary directory is created automatically.

Details

Large sampling jobs can be streamed directly to Parquet files to avoid excessive memory use.

Population values are cleaned before sampling. Rows with missing, infinite, zero, or negative population values are excluded from the sampling frame. This allows users to supply Census-derived datasets containing uninhabited or incomplete geographies.

Streamed datasets can be reopened lazily using `open_synthetic_dataset()` and materialized with `dplyr::collect()` when needed.

Value

If streaming is not triggered, returns a `data.table` with class "synthetic_points" containing:

lon Longitude in WGS84 coordinates

lat Latitude in WGS84 coordinates

fips County FIPS code

GEO_ID Source polygon identifier

If streaming is triggered, returns a list with class "synthetic_points" containing:

path Directory containing Parquet files

batches Number of sampling batches written

total_points Total number of generated points

Examples

```
data(df_bg_example)

# Sample 100 synthetic points across Rhode Island
pts <- US_pop_sampler(
  N = 100,
  df = df_bg_example,
  seed = 123
)

head(pts)
```

Index

* datasets

covid_data_example, [4](#)

df_bg_example, [4](#)

assign_dates_to_pts, [2](#)

assign_dates_to_pts(), [7](#)

covid_data_example, [4](#)

df_bg_example, [4](#)

load_bg_data, [5](#)

load_bg_data(), [7](#)

load_covid_data, [6](#)

load_synthetic_covid, [7](#)

load_synthetic_covid(), [7](#)

open_synthetic_dataset, [8](#)

US_pop_sampler, [9](#)