Package 'gridsample'

August 7, 2018

Title Tools for Grid-Based Survey Sampling Design

Version 0.2.1

Description Multi-stage cluster surveys of households are commonly performed by governments and programmes to monitor population-level demographic, social, economic, and health outcomes. Generally, communities are sampled from subpopulations (strata) in a first stage, and then households are listed and sampled in a second stage. In this typical two-stage design, sampled communities are the Primary Sampling Units (PSUs) and households are the Secondary Sampling Units (SSUs). Census data typically serve as the sample frame from which PSUs are selected. However, if census data are outdated inaccurate, or too geographically course, gridded population data (such as <http://www.worldpop.org.uk>) can be used as a sample frame instead. GridSample (<doi:10.1186/s12942-017-0098-4>) generates PSUs from gridded population data according to user-specified complex survey design characteristics and household sample size. In gridded population sampling, like census sampling, PSUs are selected within each stratum using a serpentine sampling method, and can be oversampled in urban or rural areas to ensure a minimum sample size in each of these important sub-domains. Furthermore, because grid cells are uniform in size and shape, gridded population sampling allows for samples to be representative of both the population and of space, which is not possible with a census sample frame.

Depends R (>= 3.2.3)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports rgdal (>= 1.2-4), raster (>= 2.5-8), data.table (>= 1.10.4), rgeos (>= 0.3-21), geosphere (>= 1.5-5), sp (>= 1.2-4), spatstat (>= 1.49-0), methods, maptools (>= 0.8-41), spatstat.utils

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-07 13:50:03 UTC

Author Dana R. Thomson [aut] (University of Southampton),
Nick Ruktanonchai [cre] (University of Southampton),
Forrest R. Stevens [aut] (University of Louisville),
Marcia Castro [aut] (Harvard University),
Andrew J. Tatem [aut] (University of Southampton),
Guilherme A. Zagatti [ctb] (Flowminder)

Maintainer Nick Ruktanonchai <nrukt00@gmail.com>

R topics documented:

gs_mode 2
gs_rasterize 3
gs_sample 4
gs_zonal_raster 8
RWAshp 9

Index **10**

`gs_mode` *Most common stratum calculator*

Description

For each cell in the user-defined "coarse grid" used to select spatially-representative samples (i.e. when `cfg_sample_spatial == TRUE`), this function calculates the stratum that occurs most often within each course grid cell.

Usage

```
gs_mode(rast)
```

Arguments

`rast`	`data.table` object. This `data.table` where each cell that lies within a larger grid cell is represented as a row. For each row, the variable `grid_id` is the ID of the cell from the coarser grid, `sampled` denotes whether a cell has been sampled, `stratum` defines the stratum each cell lies within, and `raster_index` is a unique value for each cell in the raster.

Value

Vector of values representing the stratum that occurs most often within a given subset of the raster.

Author(s)

Forrest R. Stevens, <forrest.stevens@louisville.edu>

gs_rasterize *Rasterize polygon layer*

Description

This function creates a raster layer that adopts values from a defined field in a polygon layer, using `rasterize` from the raster package. This function also converts values to binary if desired, where all zero values are recorded as zero, and all non-zero values are recorded as one. This function also saves the output raster in the working directory.

Usage

```
gs_rasterize(input_features, output_raster, template_raster,
  binary = FALSE, field = "ID", overwrite = FALSE,
  format = "GTiff")
```

Arguments

`input_features`	SpatialPolygons* object. Name of input shapefile layer. Should be a SpatialPolygons object.
`output_raster`	Character. Desired name of output raster layer.
`template_raster`	Raster* object. Raster with desired characteristics (resolution, extent) of output raster.
`binary`	logical. If `TRUE`, any non-zero values will be converted to one.
`field`	character. Name of variable that output raster should inherit.
`overwrite`	logical. Defines whether to overwrite if `output_raster` already exists.
`format`	character. Desired format of output raster file.

Value

Vector of values representing the stratum that occurs most often within a given subset of the raster.

Author(s)

Forrest R. Stevens, <forrest.stevens@louisville.edu>

gs_sample *GridSample sampling algorithm*

Description

The gs_sample algorithm creates primary sampling units (PSUs) for multi-stage cluster household surveys based on gridded population data. Typical complex survey design is supported with input of a raster of population counts, a raster of urbanized areas, and a raster of study strata. Each of these rasters need to be in an identical projection and have an identical grid resolution. The algorithm first selects PSU seed cells with a probability proportionate to population size according to strata, rural-urban, and spatial parameters specified, then it optionally grows PSUs around the seed cells until a minimum population threshold is met in each PSU.

Usage

```
gs_sample(population_raster, strata_raster, urban_raster,
  cfg_hh_per_stratum, cfg_hh_per_urban, cfg_hh_per_rural, cfg_pop_per_psu,
  cfg_sample_rururb = FALSE, cfg_sample_spatial = FALSE,
  cfg_sample_spatial_scale = NA, cfg_desired_cell_size = NA,
  cfg_max_psu_size = Inf, cfg_min_pop_per_cell = 0,
  cfg_psu_growth = TRUE, cfg_random_number = NA, output_path,
  sample_name)
```

Arguments

population_raster
: Raster* layer. Input gridded population dataset to use as sample frame. Values should be number of people in each pixel as a whole number or decimal value.

strata_raster
: Raster* layer. Raster that defines the stratum numeric ID of each pixel. Generally created by rasterizing a shapefile of polygons that define strata.

urban_raster
: Raster* layer. Raster of urbanized areas where a cell value of 1 indicates urban cells and 0 indicates rural cells.

cfg_hh_per_stratum
: numeric. Target household sample size per stratum. In a non-stratified sample, this is the total sample size of households. In a stratified sample, this is the household sample size per stratum.

cfg_hh_per_urban
: numeric. Number of households expected to be selected per urban PSU during survey fieldwork.

cfg_hh_per_rural
: numeric. Number of households expected to be selected per rural PSU during survey fieldwork.

cfg_pop_per_psu
: numeric. Minimum population per PSU (e.g. 500 persons).

`cfg_sample_rururb`
: logical. A flag to oversample rural/urban areas if one domain does not meet the target sample size per stratum. Default is `FALSE`.

`cfg_sample_spatial`
: logical. A flag to oversample in space ensuring that at least one PSU is selected within each "coarse grid" cell with cell size defined by the user. Default is `FALSE`.

`cfg_sample_spatial_scale`
: If `cfg_sample_spatial == TRUE`, this defines the length in kilometres of the side of the cell (e.g. 20 for 20km X 20km) of each coarse grid cell where the algorithm will ensure at least one PSU is located in each coarse grid cell.

`cfg_desired_cell_size`
: numeric. Desired length of the side of the cell in 100m (e.g. 4 for 400m X 400m) for output raster of PSUs. Defaults to NA, which yields an output raster at the same resolution as population_raster.

`cfg_max_psu_size`
: numeric. Maximum allowed geographic size of a given PSU in kilometres squared (e.g. 5 for PSUs smaller than 5km X 5km). Defaults to infinity.

`cfg_min_pop_per_cell`
: numeric. Minimum population in a raster cell required for it to be considered for sampling. Cells with less than this value will be excluded from the sample. Defaults to 0, therefore including all cells.

`cfg_psu_growth`
: logical. Determines whether to grow PSUs until either there are no available cells or each PSU covers a population defined by `cfg_pop_per_psu`.

`cfg_random_number`
: numeric. The random number seed to reproduce a previous gridded population sample.

`output_path`
: character. Output path and folder name.

`sample_name`
: character. Name of output PSU shapefile.

Details

A number of sampling features are optional. Oversampling in urban/rural areas, oversampling to be spatially representative, and stratification are not required. At a minimum, the user generates a simple random sample of PSUs in a study area by inputting a `population_raster`, defining the study area boundary as one stratum with `strata_raster`, defining the output shapefile parameters `output_path` and `sample_name`, and configuring the parameters `cfg_hh_per_stratum`, `cfg_hh_per_urban`, `cfg_hh_per_rural`, and `cfg_pop_per_psu`. See the "Stratification", "Urban/rural domains", "Spatial sampling", and "PSU size and framework" sections for additional information. Note that all datasets are re-projected into WGS84 before the sampling process begins. A real-world example can be seen using the code `vignette("Rwanda")`, a vignette that replicates the sample design of the 2010 Rwanda DHS survey.

Value

Shapefile of household survey primary sampling unit (PSU) boundaries

Stratification

To stratify the sample, define geographic strata boundaries with `strata_raster`, and specify the sample size per strata with `cfg_hh_per_stratum`. For example, if a national survey will sample 10,000 households from 5 provinces, then `cfg_hh_per_stratum = 2000`. The parameter `cfg_hh_per_stratum` is the minimum sample size to generate representative population statistics. In some surveys, strata follow urban/rural boundaries within administrative units. If this is the case, then `strata_raster` should include the boundaries of urban and rural sampling areas within each administrative area, and `cfg_hh_per_stratum` should reflect the correct sample size per stratum - for example, a national sample of 10,000 households from each urban and rural areas in 5 provinces would have `cfg_hh_per_stratum = 1000`.

Urban/rural domains

If urban/rural populations are not part of the stratification scheme, then they are often treated as a sub-domain. Sub-domains represent important sub-populations for which representative statistics are generated from the survey data, and thus each sub-domain (at the national-level) should meet the minimum sample size specified for each stratum. If either the urban/rural sub-domain does not include enough households to generate population statistics with the desired precision, then extra PSUs are oversampled in the smaller sub-domain. To implement this step with `gs_sample`, set `cfg_sample_rururb = 1`. In practice, rural areas are often more difficult and expensive to visit, and thus a greater number of households might be sampled from rural PSUs than urban PSUs. This is why the user may specify different numbers of households to be sampled from each urban PSUs (`cfg_hh_per_urban`) and rural PSUs (`cfg_hh_per_rural`); if the same number of households will be sampled from all PSUs, then configure both of these parameters with the same value. Note, the number of PSUs that will be generated in each stratum is `cfg_hh_per_stratum` divided by some number between `cfg_hh_per_urban` and `cfg_hh_per_rural`.

Spatial sampling

To select a sample that is both representative of the population and of space, set `cfg_sample_spatial = 1` and specify `cfg_sample_spatial_scale`, the spatial scale at which the sample should be representative. The spatial scale should be meaningful; for example, it will facilitate small area estimates with limited statistical error for administrative units that are smaller than the stratification units. Determining an appropriate spatial scale might take trial and error. If the study area has large regions of sparse population, a typical non-spatially representative sample will follow the population distribution and have large areas without a PSU. In this case, the user might need to increase the spatial resolution `cfg_sample_spatial_scale` of the sample, or force the algorithm to generate more PSUs in each stratum by increasing `cfg_hh_per_stratum` and/or reducing `cfg_hh_per_urban` and `cfg_hh_per_rural`.

PSU size and fieldwork

Four additional parameters can be configured to deal with idiosyncrasies of gridded population data and improve feasibility of fieldwork. The user can set a maximum geographic size of PSU in kilometres squared, `cfg_max_psu_size`. We recommend choosing a size that can feasibly be visited by a field team on foot during one day. The user might also specify which cells are included in the sample frame with `cfg_min_pop_per_cell`. Selection of a sensible value is highly dependent on the gridded population dataset being used, and the scale of the input data (e.g. 200m X 200m grid cells). The cell size of the output raster can be specified with `cfg_desired_cell_size`. Gridded

population datasets generated from old population figures or old covariates may be inaccurate at a very local scale (e.g. 100m X 100m cells), but will generally increase in accuracy as cells are aggregated (e.g. 300m X 300m cells). Finally, the PSU growth portion of the algorithm can be switched off by setting `cfg_psu_growth = FALSE` resulting in a sample of single grid cells (and their centroids).

Examples

```
require(raster)

poprast <- raster(ncols = 100, nrows = 100, xmx = 10, xmn = 9, ymn = 9, ymx = 10,
   crs = CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"),
   vals = runif(10000, 0, 100))
stratarast <- raster(ncols = 100, nrows = 100, xmx = 10, xmn = 9, ymn = 9, ymx = 10,
   crs = CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"),
   vals = c(rep(1, times = 5000), rep(2, times = 5000)))
urbanrast <- poprast > 25

example_1 <- gs_sample(
    population_raster = poprast,
    strata_raster = stratarast,
    urban_raster = urbanrast,
    cfg_hh_per_stratum = 800,
    cfg_hh_per_urban = 20,
    cfg_hh_per_rural = 20,
    cfg_pop_per_psu = 500,
    cfg_sample_rururb = TRUE,
    cfg_sample_spatial = FALSE,
    cfg_sample_spatial_scale = 100,
    cfg_desired_cell_size = NA,
    cfg_max_psu_size = 5,
    cfg_min_pop_per_cell = 0.01,
    output_path = tempdir(),
    sample_name="Example"
)
plot(example_1)

#### Example two is the identical, except PSUs aren't grown,
#### so the shapefile returned includes a single grid cell for each PSU.

example_2 <- gs_sample(
   population_raster = poprast,
   strata_raster = stratarast,
   urban_raster = urbanrast,
   cfg_hh_per_stratum = 800,
   cfg_hh_per_urban = 20,
   cfg_hh_per_rural = 20,
   cfg_pop_per_psu = 500,
   cfg_sample_rururb = TRUE,
   cfg_sample_spatial = FALSE,
   cfg_sample_spatial_scale = 100,
   cfg_desired_cell_size = NA,
```

```
   cfg_max_psu_size = 5,
   cfg_min_pop_per_cell = 0.01,
   cfg_psu_growth = FALSE,
   output_path = tempdir(),
   sample_name="Example_without_growth"
)
plot(example_2)
```

`gs_zonal_raster` *Zonal statistics calculator*

Description

This function calculates zonal statstics across a raster layer, for each polygon in a rasterized polygon layer.

Usage

```
gs_zonal_raster(x, z, stat = "mean", digits = 1, na.rm = TRUE, ...)
```

Arguments

`x`	Raster* layer. The layer that zonal statistics should be calculated from.
`z`	Raster* layer. A rasterized version of the zonal layer.
`stat`	character. Name of statistic used to calculate a value across each polygon in `z`. ex. "mean", "sum".
`digits`	numeric. Number of significant digits in zonal statistic output.
`na.rm`	logical. Defines whether to remove `NA`
`...`	Other variables

Value

Vector of values representing the calculated statistic for each polygon, sorted by the order of polygons in the polygon layer.

Author(s)

Forrest R. Stevens, <forrest.stevens@louisville.edu>

RWAshp *First-level administrative units for Rwanda*

Description

First-level administrative units for Rwanda

Usage

```
RWAshp
```

Format

A shapefile of first-level administrative units for Rwanda

Source

http://gadm.org/

Index

∗Topic **datasets**
RWAshp, 9

gs_mode, 2
gs_rasterize, 3
gs_sample, 4
gs_zonal_raster, 8

RWAshp, 9