Title of template

**Skill level needed:**Advanced

**Sample designs supported:**

1, 2, 3, 4, 5, 6

Designing and Implementing Gridded Population Surveys

**gridpopsurvey.com**

**A5. PSU sample – GridEZ frame**

Last updated: Aug 2022

**Generate sample frame with GridEZ R algorithm**

## Example: Uganda

**Motivation:** Use this tutorial to generate your own GridEZ sample frame when a gridded population dataset is not available in GridSample.org (e.g., HRSL or a bespoke gridded population dataset), or when you are an advanced user with a reason to select your sample in R. GridEZ is an R algorithm that groups contiguous gridded population cells into EA-like units of approximately similar total population. The algorithm requires (1) gridded population raster, (2) strata boundaries of the area(s) of interest, converted to raster, and (3) settlement type (e.g., urban/rural areas) raster. Recommended sources of strata and settlement type boundaries are provided if the user does not have their own.

**Resources:**

* GridEZ algorithm: [https://github.com/cadooley/GridEZ](https://github.com/cadooley/gridEZ)

**Example:** In this example, survey implementers generated a bespoke 100x100 m gridded population dataset of 12 refugee settlements in Uganda (see Tutorial A4), and used GridEZ to generate and sample “medium” frame units of approximately 500 people each across their areas of interest.

* All three input raster files should have the same extent and projection and be aligned.
* NA values (representing cells outside area of interest) should be identical across rasters.
* Spatially joining strata boundaries or other attributes in R ensures that grids are aligned.

**Code example:**

##############################################

# Adapted by: Dana R. Thomson #

##############################################

####### 0. Set-up

# Install and load R packages

library(sp) #version: sp\_1.2-7

library(raster) #version: raster\_2.6-7

library(rgdal) #version: rgdal\_1.2-18

library(parallel) #base package

# Specify number of cores to use and type of parallel processing

ncores <- detectCores() - 1

par\_type <- "PSOCK"

####### 1. Read in data

# Read population raster data, set NA=0

population\_raster <- raster("D:/project/data/population.tif")

population\_raster[is.na(population\_raster)] <- 0

# Create settlement type raster data (constant value means that GridEZs will not be separated by type – e.g. urban/rural – within boundaries)

type\_raster <- population\_raster

type\_raster[type\_raster >= 0] <- 1

# Read settlement boundaries data

unhcr\_polygons <- readOGR("D:/project/data/boundaries.shp")

unhcr\_polygons <- spTransform(unhcr\_polygons, CRS = crs(population\_raster))

unhcr\_raster <- rasterize(unhcr\_polygons, field="Settle\_ID", settlement\_raster)

# Crop and mask population + settlement data, by settlement boundaries

population\_raster <- crop(population\_raster, extent(unhcr\_raster))

population\_raster <- mask(population\_raster, unhcr\_raster)

type\_raster <- crop(type\_raster, extent(unhcr\_raster))

type\_raster <- mask(type\_raster, unhcr\_raster)

####### 2. Create GridEZ units

# Read the GridEZ function

source("D:/project/code/GridEZ-master/GridEZ\_fn\_public\_release\_v1.R")

# Run GridEZ code

GridEZ(population\_raster = population\_raster,

settlement\_raster = type\_raster,

strata\_raster = unhcr\_raster,

exclude\_unsettled = FALSE,

using\_ghs\_smod\_pop2015 = FALSE,

predefined\_EZ\_size = TRUE,

EZ\_target\_size = "medium",

output\_path = "D:/project/data",

run\_ID = "\_v1" )

####### 3. Check GridEZ outputs & create polygons

# Read GridEZ raster data

GridEZ\_raster\_id <- raster("D:/project/data/EZ\_raster\_master\_v1.tif")

names(GridEZ\_raster\_id) <- "GridEZ\_id"

GridEZ\_raster\_pop <- raster("D:/project/data/EZ\_pop\_raster\_master\_v1.tif")

names(GridEZ\_raster\_pop) <- "GridEZ\_pop"

# Stack GridEZ ID and pop datasets

GridEZ\_raster\_stack <- stack(GridEZ\_raster\_id, GridEZ\_raster\_pop)

# Vectorize GridEZ raster stack to polygons

GridEZ\_polygons <- rasterToPolygons(x = GridEZ\_raster\_stack, fun=NULL, n=4, na.rm=TRUE, digits=12, dissolve=TRUE)

####### 4. Attach data for sample stratification & save to shapefile + CSV

# Join admin unit names and calculate population density for each GridEZ

strata\_attributes <- over(GridEZ\_polygons, unhcr\_polygons)

GridEZ\_polygons <- spCbind(GridEZ\_polygons, strata\_attributes)

GridEZ\_polygons$area\_sqkm <- area(GridEZ\_polygons) / 1000000

GridEZ\_polygons$popdens\_sqkm <- GridEZ\_polygons$GridEZ\_pop / GridEZ\_polygons$area\_sqkm

# Write GridEZ sample frame to a shapefile

writeOGR(obj=GridEZ\_polygons, dsn="D:/project/output", layer="UGA\_GridEZ\_frame", driver="ESRI Shapefile", overwrite\_layer=TRUE)

# Write GridEZ sample frame to a csv

write.csv(GridEZ\_polygons, file="D:/project/output/UGA\_GridEZ\_frame.csv", row.names=FALSE )

**Example output:**

Chart, treemap chart

Description automatically generated Table

Description automatically generated

About CC Licenses - Creative Commons [This work](http://www.gridpopsurvey.com/)© 2022 is licensed under [CC BY 4.0](http://creativecommons.org/licenses/by/4.0/?ref=chooser-v1)