Title of template

**Skill level needed:**Advanced

**Sample designs supported:**

1, 2, 3, 4, 5, 6

Designing and Implementing Gridded Population Surveys

**gridpopsurvey.com**

**A6. PSU sample – R to select PSUs**

Last updated: Aug 2022

**PSU sample selection in R**

## Example: Uganda

**Motivation:** After a PSU sample frame has been constructed, use this tutorial to select a sample of PSUs if you are an advanced user with a reason to select your sample in R.

**Example:** In this example, survey implementers generated a sample frame of GridEZ units (see Tutorial A5), and then spatially joined the names of corresponding administrative units based on an external shapefile, and calculated the area and population density in each GridEZ unit. Next they used R to draw a systematic PPS sample of 40 PSUs that were implicitly stratified by administrative units and population density (a proxy for urbanicity). The names of administrative units and locations of PSUs have been masked in this example to ensure anonymity of survey respondents and vulnerable communities. The below code example assumes the following dataset variables:

* GridEZ\_id: Unique ID for each GridEZ unit (generated by GridEZ algorithm)
* GridEZ\_pop: Estimated population in GridEZ unit (generated by GridEZ algorithm)
* Admin 1: First level administrative unit (joined from external dataset)
* Admin 2: Second level administrative unit (joined from external dataset)
* GridEZ\_sqkm: area of GridEZ unit in square kilometers (calculated)
* popdens\_sqkm: calculated as GridEZ\_pop ÷ GridEZ\_sqkm

**Code example:**

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

# Author: Qixuan Chen #

# Edited by: Dana R. Thomson #

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

# read in the sampling frame

popu = read.csv("UGA\_GridEZ\_frame.csv")

# sort by administrative unit and population density, and create SortOrd

popu <- popu[with(popu, order(Admin1, Admin2, popdens\_sqkm)), ]

popu$SortOrd <- 1:nrow(popu)

# define sample size and sampling interval for the systematic PPS sampling

n = 40

popu$cumsum = cumsum(popu$GridEZ\_pop)

interval = sum(popu$GridEZ\_pop)/n

print(interval)

# define empty vectors for saving selection results

sampleindices = numeric()

Val = numeric()

# set the seed and generate the starting value

set.seed(192837)

startVal = sample(1:(interval\*100), 1)/100

# conduct the systematic PPS sampling

Val[1] = startVal

for (i in 1:n) {

 sampleindices[i] = min(which(Val[i] <= popu$cumsum))

 if (i < n) Val[i+1] = startVal + interval\*i

}

# output the selection indices and the corresponding cumulative sum of GridEZ\_pop

output = cbind(sampleindices, Val)

# the list of selected "SortOrd"

popu$SortOrd[sampleindices]

# output the selected GridEZs into a .CSV file

PSUsample = popu[sampleindices,]

write.csv(PSUsample, file="PSUsample.csv")

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