wopr: An R package to query the WorldPop Open Population Repository

Doug Leasure

WorldPop Research Group
University of Southampton

18 February 2020

Introduction

wopr is an R package that provides API access to the WorldPop Open Population Repository. This gives users the ability to:

1. Download WorldPop population data sets directly from the R console,
2. Submit spatial queries (points or polygons) to the WorldPop server to retrieve population estimates within user-defined geographic areas,
3. Get estimates of population sizes for specific demographic groups (i.e. age and sex), and
4. Get probabilistic estimates of uncertainty for all population estimates.

Code for the *wopr* package is openly available on GitHub: https://github.com/wpgp/wopr

Installation

Install the *wopr* R package from WorldPop on GitHub:

```r
devtools::install_github('wpgp/wopr')
library(wopr)
```

Usage

Demo code is provided in `demo/wopr_demo.R`.

See vignettes available with: `vignette(package='wopr')`

If you are intersted in developing your own front end applications that query the WOPR API, please read the vignette describing the API backend for developers: `vignette('wopr_api', package='wopr')`

woprVision

woprVision is a R shiny application that allows you to browse an interactive map to get population estimates for specific locations and demographic groups. To access woprVision, use:

```r
woprVision()
```

Data Download

One way to access data from WOPR is to simply download the files directly to your computer from the R console. This can be done with three easy steps:

```r
# Retrieve the WOPR data catalogue
catalogue <- getCatalogue()

# Select files from the catalogue by subsetting the data frame
selection <- subset(catalogue,
                    country == 'NGA' &
                      category == 'Population' &
                      version == 'v1.2')
```

```
# Download selected files
downloadData(selection)
```

Note: `'NGA'` refers to Nigeria. WOPR uses ISO country codes to abbreviate country names.

By default, `downloadData()` will not download files larger than 100 MB unless you change the `maxsize` argument (see `?downloadData`). Using the default settings, a folder named `./wopr` will be created in your R working directory for downloaded files. A spreadsheet listing all WOPR files currently saved to your hard drive can be found in `./wopr/wopr_catalogue.csv`. To list the files that have been downloaded to your working directory from within the R console, use `list.files('wopr', recursive=T)`. In multiple calls to downloadData(), files that you have previously downloaded will be overwritten if your local files do not match the server files (based on an md5sums check). This allows you to keep up-to-date local copies of every file.

You can download the entire WOPR data catalogue using: `downloadData(getCatalogue(), maxsize=Inf)`. Note: Some files in the WOPR data catalogue are very large (e.g. 140 GB), so please ensure that you have enough disk space. If disk space is limited, you can restrict the maximum file size that you woud like to download using the `maxsize` argument.

Spatial Query

Population estimates can also be obtained from WOPR using spatial queries (geographic points or polygons) for user-defined geographic area(s) and demographic group(s).

Spatial queries must be submitted using objects of class `sf`. You can explore this functionality using example data from Nigeria that are included with the `wopr` package. Plot the example data using:

```
plot(wopr_points, pch=16)
plot(wopr_polys)
```

Note: ESRI shapefiles (and other file types) can be read into R as `sf` objects using:

```
sf_feature <- sf::st_read('shapefile.shp')
```

To submit a spatial query, you must first identify which WOPR databases support spatial queries:

```
getCatalogue(spatialQuery=T)
```

This will return a `data.table`:

country	version
NGA	v1.2
NGA	v1.1
COD	v1.0

These results indicate that there are currently two WOPR databases for Nigeria (NGA) that support spatial queries and one database for Democratic Republic of Congo (COD).

Query total population at a single point

To get the total population for a single point location from the NGA v1.2 population estimates use:

```
N <- getPop(feature=wopr_points[1,],
            country='NGA',
            ver='1.2')
```

Notice that the population estimate is returned as a vector of samples from the Bayesian posterior distribution:

```
print(N)
hist(N)
```

This can be summarized using:

```
summaryPop(N, confidence=0.95, tails=2, abovethresh=1e5, belowthresh=5e4)
```

The `confidence` argument controls the width of the confidence intervals. The `tails` argument controls whether the confidence intervals are calculated as one-tailed or two-tailed probabilities. If `confidence=0.95` and `tails=2`, then there is a 95% probability that the true population falls within the confidence intervals, given the model structure and the data used to fit the model. If `confidence=0.95` and `tails=1`, then there is a 95% chance that the true population exceeds the lower confidence interval and a 95% chance that the true population is less than the upper confidence interval.

The `abovethresh` argument defines the threshold used to calculate the probability that the population will exceed this threshold. For example, if `abovethresh=1e5`, then the `abovethresh` result from `summaryPop()` is the probability that the population exceeds 100,000 people. The `belowthresh` argument is similar except it will return the probability that the population is less than this threshold.

Query total population within a single polygon

To query WOPR using a single polygon works exactly the same as a point-based query:

```
N <- getPop(feature=wopr_polygons[1,],
            country='NGA',
            ver='1.2')

summaryPop(N, confidence=0.95, tails=2, abovethresh=1e2, belowthresh=50)
```

Query population for specific demographic groups

To query population estimates for specific demographic groups, you can use the `agesex` argument (see `?getPop`). This argument accepts a character vector of age-sex groups. `'f0'` represents females less than one year old; `'f1'` represents females from age one to four; `'f5'` represents females from five to nine; `'f10'` represents females from 10 to 14; and so on. `'m0'` represents males less than one, etc.

Query the population of children under the age of five within a single polygon:

```
N <- getPop(feature=wopr_polygons[1,],
            country='NGA',
            ver='1.2',
            agesex=c('f0','f1','m0','m1'))

summaryPop(N, confidence=0.95, tails=2, abovethresh=10, belowthresh=1)
```

If the `agesex` argument is not included, the `getPop()` function will return estimates of the *total* population (as above).

Query multiple point or polygon features

We can query multiple point or polygon features using the `woprize()` function:

```
N_table <- woprize(features=wopr_polys,
                   country='NGA',
                   ver='1.2',
                   agesex=c('m0','m1','f0','f1'),
                   confidence=0.95,
                   tails=2,
```

```
                        abovethresh=2e4,
                        belowthresh=1e4
                        )
```

You can save these results in a number of ways:

```
# save results as shapefile
sf::st_write(N_table, 'example_shapefile.shp')

# save results as csv
write.csv(sf::st_drop_geometry(N_table), file='example_spreadsheet.csv', row.names=F)

# save image of mapped results
jpeg('example_map.jpg')
tmap::tm_shape(N_table) + tmap::tm_fill('mean', palette='Reds', legend.reverse=T)
dev.off()
```

Functions

cellids()
checkTask()
downloadData()
endpoint()
getCatalogue()
getPop()
getPopSql()
retrieveResults()
submitTasks()
summaryPop()
woprize()
writeCatalogue()

Contributing

The WorldPop Open Population Repository (WOPR) was developed by the WorldPop Research Group within the Department of Geography and Environmental Science at the University of Southampton. Dr. Maksym Bondarenko and Niko Ves from the WorldPop Spatial Data Infrastructure team developed the WOPR API server. Funding was provided by the Bill and Melinda Gates Foundation and the United Kingdom Department for International Development.

License

GNU General Public License v3.0 (GNU GPLv3)