Small area population estimates using random forest top-down disaggregation (Part 2): the **popRF** 'R' package

WorldPop, University of Southampton

2021-11-17

1 Introduction

The purpose of top-down disaggregation is to estimate population counts at a finer spatial resolution than the available population totals for administrative units. There are multiple ways of doing this. WorldPop implements a two-steps dasymetric mapping approach that uses (1) the random forest machine learning algorithm to estimate the relative spatial variation in population numbers (I.e., a weighting layer), and then (2) to disaggregate the projected census totals based on this weighting layer (Sorichetta et al. 2015, Stevens et al. 2015). The weighting layer is estimated statistically, based on relationships with high resolution geospatial covariates (e.g. building locations, roads, etc.).

In the previous tutorial (Leasure et al. 2020), the authors described the random forest methods in detail and demonstrated how to implement this method in the R statistical programming environment to create population count estimates for census enumeration areas (EAs). In this tutorial, we will apply the random forest method at 100m grid resolution using a new R package called **'popRF'** (Bondarenko et al 2021). Thus, the motivation of this tutorial is to (1) illustrate the process of how to create gridded outputs with the Random Forest disaggregation method, and (2) how to do this efficiently with a new **'popRF'** R package.

1.1 Pre-requisites

The following software may be needed to complete this tutorial:

- R environment (https://www.r-project.org/),
- RStudio (https://www.rstudio.com/) - optional, and
- QGIS (https://www.qgis.org/en/site/).

Before completing this tutorial, please, read and complete the Part 1 tutorial (Leasure et al 2020) to get more information on the concept, details of the intermediate steps, checking of results and limitations that are not shown/discussed here.

1.2 Data requirements

Please create a '*popRF_demo*' folder with an '*Inputs*' subfolder on your computer, for example: '*C:\Local\popRF_demo\Inputs*'. This tutorial uses a French Guiana dataset. Please download these into your '*Inputs'* subfolder:

- Rasterised administrative areas (https://www.worldpop.org/geodata/summary?id=24586)

- Population projections (https://data.worldpop.org/GIS/Population/Global_2000_2020/CensusTables/guf_population_2000_2020.csv). PLEASE make sure that you DELETE ALL columns except *'GID'* and the projections that you want to use, for example *'P_2020'*. When this is done, please, also delete row 1 that contains the header of the data.
- Geospatial covariates
    - Distance to edges of reclassified ESA-CCI-LC (landcover) classes 2015 (011, 040, 130, 140 and 150 layers) (https://www.worldpop.org/geodata/summary?id=22843)
    - Distance to open water coastline layer (https://www.worldpop.org/geodata/summary?id=23839)
    - STMR slope 2000 layer (https://www.worldpop.org/geodata/summary?id=23092)
- Surface water bodies (https://data.worldpop.org/GIS/Covariates/Global_2000_2020/GUF/ESA_CCI_Water/Binary/guf_esaccilc_water_100m_2000_2012.tif)
- Grid cell surface areas (https://www.worldpop.org/geodata/summary?id=23590)

2 Assessing the inputs

Before starting the random forest top-down disaggregation process, it is highly recommended that the input datasets are thoroughly assessed.

N.B. The data in this example application is already checked and prepared in the required format so you do not need to do this step. However, these checks and steps are essential when you are building your own application.

2.1 Administrative boundaries

It is advisable to use the lowest level of administrative units for which official population totals or projections are available. In most cases, having population totals for smaller inputs improves the spatial distribution of population in the output dataset through a larger training dataset and by considering finer scale variations in covariate values.

Necessary basic checks on the boundaries:

- It is critical that the administrative unit boundary accurately represents the geographic area within which the population was enumerated (i.e. that the reference year of the population data is the same as the administrative boundary data and the spatial extent matches the enumerated population)
- It is similarly important that the unique identifiers in the administrative boundary layer match the unique identifiers in the tabulated population data so that (i) every polygon in the administrative boundary layer has a corresponding record in the population data table and (ii) that the corresponding admin name or admin code match each other exactly.

Topological checks:

Administrative boundaries are generally digitised and stored as polygons (e.g. shapefiles) using a GIS software. During the digitisation process, small, not easily visible issues can be introduced that can result in errors during spatial analyses. Thus, these administrative boundary polygons require further checks to ensure that there are no gaps, overlaps, duplicates, self-intersection, etc. amongst the

polygons. To perform these checks, we can use ArcGIS or QGIS or similar software that can do spatial calculations. The simplest solution is to dissolve all the administrative boundaries in the country and look for the remaining gaps and their sizes (i.e. width of gaps), if any. However, there are also pre-coded functions to perform these checks and make corrections as needed manually or automatically.

Example of processing functions in QGIS (version 3.16.8-Hannover):

- **Vector > Geometry tools > Check validity** (to check polygon geometry, including if a self-intersecting ring exists)
- **Vector > Topology checker** (to check if gaps, overlaps, duplicates, invalid geometries, multipart features exist)
- **Processing > Toolbox > Snap Geometry to layer** (this tool can automatically fix small gaps/overlaps. It is recommended that you do not fix gaps larger than 5m with this tool as this automated function will alter the boundaries without any supervision.)
- If manual adjustments are needed, please do not forget to turn on the snapping **(Project > Snapping Options...)**. This will ensure that the moved or newly created vertices are snapped to nearby vertices and thus eliminates the possibility to create new gaps or overlaps. For such manual adjustments you may want to find and use **'Vertex' tool** of 'Toggle editing'; **'Delete Part'** and '**Delete Ring'** tools of the 'Advance digitising' toolbar.

The final step of the input preparation is to rasterise this validated polygon shapefile with the same resolution as the other raster inputs (i.e. same cell size, same coordinate system, same spatial extent and the grid cells of all inputs are perfectly aligned). In our case it is 3 arc seconds (0.0008333333 decimal degrees) spatial resolution, which is approximately 100m at the Equator, and datasets should be in the geographic (unprojected) coordinate system of WGS84 (EPSG: 4326). Please, make sure that the pixel values are the unique IDs of the administrative boundary polygons that match the IDs in the population data table.

Example of processing functions in QGIS (version 3.16.8-Hannover):

- **Raster > Conversion > Rasterize (Vector to Raster)**

2.2 Creating and checking the covariates

The use of geospatial covariates enables the Random Forest method to pick up the small spatial differences during the disaggregation process. There are three basic criteria for covariates:

- They must be directly or indirectly related to the spatial distribution of population
- They must be available and consistently measured for the entire study area
- They must include geographical information on location, and they must match the output resolution. So, if the output will be gridded, the covariates must also be gridded.

There are many covariates with global coverage available to download on the WorldPop website (https://www.worldpop.org/project/categories?id=14) using the methodology of Lloyd et al (2019) or the aggregated Maxar/Ecopia building footprint raster data (https://wopr.worldpop.org/?/Buildings, Dooley et al 2021), but it is advisable to create your own covariates, based on the application needs and local context, and using best available data that ideally matches the time of the input population data. Generally, covariates with continuous values are better for statistical disaggregation than covariates with categorical values.

However, covariates should be checked that they are appropriate at the spatial scale you intend to use them. Please, revisit "*Chapter 5 Limitations - 2. Random Forest models are not good at extrapolating*" section of the Leasure et al (2020) tutorial for all the details and tips on how to perform the necessary checks.

2.3 Adjusting the spatial extent

In this tutorial we are using harmonised boundaries and covariates. But, if you are using pre-prepared covariates, such as those available on the WorldPop website, available to download for individual countries, together with your own administrative boundaries, the spatial extent of these layers may not match (Figure 1a). Thus, the spatial extent of the covariates must be adjusted to the be exactly the same as the spatial extent of the administrative boundaries, and thence, the population data. Instead of re-creating the covariates, the easiest solution is to (i) download all covariates for the country in question and all of its neighbours, (ii) mosaic them so all country layers are unified for a single covariate (Figure 1b), (iii) and finally cookie-cut the correct area using the official boundary (Figure 1c). Repeat this for all covariates. These steps can be done in any GIS software or programming environment capable of handling spatial data and raster files.

Figure 1: Example for a miss-match between a pre-prepared WorldPop covariate and the official national boundary (black line): a) raw national covariate overlaid with national boundary; b) merged national and neighbour country covariate; c) clipped merged covariate to match the national border

In case, a pre-prepared covariate is fairly old and newer data is available which better matches the time of the population data, you may want to re-create a covariate, specifying the spatial extent to match the administrative boundaries. If, however, newer data is unavailable and the pre-prepared covariate is not available for the neighbouring countries, you need to spatially extrapolate the values in the pre-prepared covariate raster and then cookie-cut the correct area using the administrative unit boundaries to ensure that the spatial extents match exactly.

Example of processing functions in QGIS (version 3.16.8-Hannover):

- **Raster > Miscellaneous > Merge**
- **Raster > Extraction > Clip Raster by Mask Layer**
- **Raster > Analyses > Fill noData...** (to fill raster regions with no data values by interpolation from edges)

2.4 Constraining the disaggregation

It is recommended to always constrain the outputs of these Random Forest models to non-water pixels, but further constraint(s) might be added, as needed for the application in question.

An important consideration must be made whether the output of the disaggregation should cover the entire country or should be constrained to only settled pixels (https://www.worldpop.org/methods/top_down_constrained_vs_unconstrained). If the aim is to predict only to settled pixels, the covariates must be adjusted by masking out the non-settled areas (assign a value of NA to non-settled pixels) and keeping the settled pixel values as is. Settled area boundaries can come from a variety of sources, for example the Maxar/Ecopia building footprints (Ecopia.AI and Maxar Technologies 2020), the World Settlement Footprint (Marconcini et al 2020), the Global Human Settlement layer (https://ghsl.jrc.ec.europa.eu/) or other sources.

Furthermore, if there is information available on whether a pixel is residential or non-residential, it is strongly suggested that you use that information to either mask out the non-residential pixels from the covariates OR create a covariate for non-residential areas, thus the disaggregation algorithm can consider them in the population estimation. In the former case, population will not be allocated to non-residential areas, whereas in the latter case, some population will be assigned to non-residential pixels that might be desirable, if there are also communal accommodations or there are mixed-used buildings on those sites.

3 The 'popRF' (Random Forest-Informed Population Disaggregation) package

The **popRF** package is running in 'R' and is published openly and described on the CRAN website (https://CRAN.R-project.org/package=popRF). This section illustrates a simple example to give the most important information, but the detailed description of all inputs and parametrization can be found in Nieves et al (2021).

The modelling process consists of four steps, but the **popRF** package does all these for you automatically:

1. Estimate the relationship between population density and geospatial covariates at the aggregate unit level,
2. Predict population densities at the level of output unit scale using the covariates,
3. Use predicted densities as a weighting layer to disaggregate the population totals at the output unit scale, and
4. Check the results that that the aggregated population estimates match the initial population totals

3.1 Installation

It is advisable to update all R packages before the installation begins.

The **popRF** package is installed by the following command in R or Rstudio:

```
install.packages("popRF")
```

3.2 Inputs

All input data (except the population totals) should be of a raster format and all inputs should have a common (unprojected) coordinate system (e.g. WGS-84) and pixel size. The mandatory inputs to use this packages are:

1. **Zonal Data** – raster file representing subnational areas, corresponding tabular population count data for those areas
2. **Population Counts** – tabular file (.csv) containing the IDs of the subnational areas and the corresponding total population
3. **Water Mask** - binary raster file indicating pixels of water (1) and no water (0). (If an accurate water mask does not exist, all the pixel values can be assigned to zero.)
4. **Pixel Area** - raster file containing the area of each pixel in square meters
5. **Ancillary Dataset(s)** – at least one ancillary raster covariate dataset must be provided. If multiple covariates are used, these need to be in a list format.

There are opportunities in this R package to run multi-country applications and to use a prior Random Forest model. This is recommended in countries where there are only a few admin boundaries. Random Forest is not able to properly disaggregate population if only a few (i.e., less then 15) admin boundaries exist. If this happens, an error message will be displayed. Such situation might happen if the country is very small or if subnational projections are only available at close to national scale. In such cases, neighbouring countries can be merged to have more administrative units and thus to enable the model to run.

For information on the optional input settings, please read Nieves et al (2021).

3.3 Parameters to set

The **popRF** package contains one primary function: *popRF()*, where all the parameters are set. The *popRF()* function has five required parameters and 12 optional parameters related to the overall process. The required parameters are the paths to the above input files:

- ***pop***: path to the 'Population Counts' tabular file
- ***mastergrid***: path to the 'Zonal Data' raster file
- ***cov***: path to the covariate rasters (i.e. 'Ancillary Datasets')
- ***watermask***: path to the 'Water Mask' raster file
- ***px_area***: path to the 'pixel Area' raster file

The input data can be accessed either be from local paths, from other network/internet locations, or a mix of all of these.

Optional parameters include the location for writing outputs (*output_dir*), options for parallel processing (*cores*), processing messaging options (*verbose* and *log*) and a few others. Please read Nieves et al (2021) for details of these.

3.4 Execution of the Random Forest package

The **popRF** package is designed to be simple to run. An example model code is shown in Appendix A, but the main steps are listed here:

- start R or Rstudio
- load the popRF library
- set the directory location of the inputs
- create a list from the names and locations of the needed covariates
- state the names and locations of the mandatory inputs
- write the command line that executes the Random Forest application.

And as simple as that! 😊

N.B. R is very specific about folder paths. You cannot simply copy the folder path from for example Windows Explorer. R requires that slashes of the folder path are forward (*C:/Local/popRF_demo*) OR that you use double backward slashes (*C:\\Local\\popRF_demo*).

3.5 Output files

The **popRF** package saves the outputs into one output folder with several subfolders. The results of each modelled country are stored in a separate folder within the main output folder. The most important outputs are the following (XXX refers to the country code set by the user – see Appendix A example 'GUF'):

- *output\XXX**ppp_XXX.tif***: This is the final disaggregated population map.
- *output\XXX**check_result_prj_XXX.csv***: This shows the results of the automated check that the disaggregated population totals match the population totals of the 'Population Counts' input file.
- *output\XXX\tmp**IncMSE_IncNodePurity_XXX.csv***: Different covariates' contribution to the final population estimates to different degrees. This file shows each covariates importance in the prediction. There are two quantified metrics: (i) the increase of the mean squared error (%IncMSE) and (ii) the increased in node purity. Higher values indicate higher importance. More details can be found in Leasure et al (2020) and Nieves et al (2021).

3.6 Pro-tip for using the popRF package

- The **popRF** package has a demo function "*popRFdemo*" (see Appendix B). The function will download all the currently available covariates automatically from WorldPop website and use them to disaggregate 2020 WorldPop population projection units into grid cells. This is a good way to test the **popRF** package and also to download all public WorldPop covariates for a specific country. Please, note that depending on the internet speed and due to the multiple covariates, it might take a long time (30+ minutes) to download all necessary files and run the **popRF** application.
- If a user PC does not have enough RAM memory, one can use the additional arguments ("*binc*" and "*proximity*") in **popRF** function to process a large dataset. See Appendix C for more details.

- **PopRF** output requires thorough visual assessments. You may need to create new covariates or remove existing covariates to address specific concerns. Every application and setting are different, but you should look out for these in all cases:
    - Check that the aggregated admin totals match the values in the used population input file
    - Check that the explained variance (calculated automatically by **popRF**) is sufficiently high (>80%). Remember that smaller-scale inputs improve the results greatly!
    - Are there any weird shapes/patterns? Maybe a ring of high values around the settlements? Or unusual high pixel values in a low-value area? Or a sharp change in pixel magnitude in a specific area?
    - Check the results against a satellite image base layer. Zoom in on an area and check that the population estimates are realistic for that neighbourhood. E.g. count the buildings of a pixel and multiply it with an average household size to get a feel for the 'expected' magnitude.

4 Concluding remarks

Disaggregation of population totals can be done in multiple ways. The Random Forest-Informed Population Disaggregation uses geographical information to quantify the statistical relationships between these ancillary data and the population counts (or densities), which informs the dasymetric disaggregation process. The **popRF** package is an efficient one-stop-shop solution to create mass-conservative high-resolution outputs. However, there are a few very important assumptions when doing this: (i) the population data is complete, accurate and trusted; (ii) the administrative boundaries are correct and match the population totals; (iii) the geospatial covariates have a complete coverage for the study area; (iv) population density is correlated with the selected environmental and physical factors (i.e. covariates).

There are limitations to this disaggregation method. These are detailed in the Leasure et al (2020) tutorial but listed here:

- Random Forest can be time-intensive for large datasets.
- Random Forest models are not good at extrapolating.
- Random Forest models are not ideal for inferring covariate effects.

Contributions

This tutorial was written by Attila N Lazar and Maksym Bondarenko with advice from W Chris Jochem, Edith Darin, Sarchil Qader, Heather Chamberlain and Thomas Abbott with oversight from Andy Tatem.

Suggested Citation

Lazar AN, Bondarenko M, Chamberlain H, Jochem WC, Darin E, Qader S, Tatem AJ. 2021. Small area population estimates using random forest top-down disaggregation (Part 2): the popRF 'R' package. WorldPop, University of Southampton. doi: 10.5258/SOTON/WP00727.

License

References

Bondarenko M., Nieves J.J., Forrest R.S., Andrea E.G., Jochem C., Kerr D., and Sorichetta A. (2021): popRF: Random Forest-informed Population Disaggregation R package, _Comprehensive R Archive Network (CRAN)_, https://cran.r-project.org/package=popRF

Dooley, C. A., Leasure, D.R., Boo, G. and Tatem, A.J. 2021. Gridded maps of building patterns throughout sub-Saharan Africa, version 2.0. University of Southampton: Southampton, UK. Source of building footprints: Ecopia Vector Maps Powered by Maxar Satellite Imagery (C) 2020/2021. doi:10.5258/SOTON/WP00712

Nieves J.J., Bondarenko M., Stevens F.R., Gaughan A.E., Jochem W.C., Kerr D., Tatem A.J., Sorichetta A. (2021). popRF: Random Forest-informed Disaggregative Population Modelling and Mapping. 10.13140/RG.2.2.24822.93763.

Leasure DR, Darin E, Tatem AJ. 2020. Small area population estimates using random forest top-down disaggregation: An R tutorial. WorldPop, University of Southampton. doi:10.5258/SOTON/WP00697.

Lloyd CT, Chamberlain H, Kerr D, Yetman G, Pistolesi L, Stevens FR, Gaughan AE, Nieves JJ, Hornby G, MacManus K, Sinha P, Bondarenko M, Sorichetta A, Tatem AJ. 2019. Global spatio-temporally harmonised datasets for producing high-resolution gridded population distribution datasets. Big Earth Data, 3(2), 108-139. https://dx.doi.org/10.1080/20964471.2019.1625151

Marconcini, M., Metz-Marconcini, A., Üreyen, S. et al. Outlining where humans live, the World Settlement Footprint 2015. Sci Data 7, 242 (2020). https://doi.org/10.1038/s41597-020-00580-5

Ecopia.AI and Maxar Technologies. 2020. Digitize Africa data. Ecopia.AI and Maxar Technologies

Sorichetta A, Hornby G, Stevens F, Gaughan A, Linard C, Tatem A. 2015. High-resolution gridded population datasets for Latin America and the Caribbean in 2010, 2015, and 2020. Scientific Data 2:1–12. doi:10.1038/sdata.2015.45.

Stevens F, Gaughan A, Linard C, Tatem A. 2015. Disaggregating census data for population mapping using random forests with remotely-sensed and ancillary data. PLOS ONE 10:e0107042. doi:10.1371/journal.pone.0107042.

Appendix A: Example R code

```
# Load the popRF library
library(popRF)

# Set the location of the root directory, the covariates and outputs
root.dir       <- "C:/Local/popRF_demo"
covariates.dir <- file.path(root.dir, "Inputs")
output.dir     <- file.path(root.dir, "output")

# Create the output folder if it does not exist
if(!file.exists(output.dir)){
  dir.create(output.dir, recursive = TRUE, showWarnings = FALSE)
}

# Set the country code and the required covariates as a list
iso <- "GUF"
input_covariates <- list(
  "GUF" = list(
    "ccilc_dst011_2010"      = paste0(covariates.dir, "/guf_esaccilc_dst011_100m_2015.tif"),
    "ccilc_dst040_2010"      = paste0(covariates.dir, "/guf_esaccilc_dst040_100m_2015.tif"),
    "ccilc_dst130_2010"      = paste0(covariates.dir, "/guf_esaccilc_dst130_100m_2015.tif"),
    "ccilc_dst140_2010"      = paste0(covariates.dir, "/guf_esaccilc_dst140_100m_2015.tif"),
    "ccilc_dst150_2010"      = paste0(covariates.dir, "/guf_esaccilc_dst150_100m_2015.tif"),
    "cciwat_dst"             = paste0(covariates.dir, "/guf_dst_coastline_100m_2000_2020.tif"),
    "bsgm_wpgp_2005_dst" = paste0(covariates.dir, "/guf_srtm_slope_100m.tif")
  )
)

# Set the file names of the other mandatory inputs
input_mastergrid <- list("GUF" =paste0(covariates.dir, "/guf_subnational_admin_2000_2020.tif"))
input_watermask <- list("GUF" = paste0(covariates.dir, "/guf_esaccilc_water_100m_2000_2012.tif"))
input_px_area     <- list("GUF" = paste0(covariates.dir, "/guf_px_area_100m.tif"))
input_poptables   <- list("GUF" = paste0(covariates.dir, "/guf_population_2000_2020.csv"))

# Run the Random Forest application
out <- popRF(pop  = input_poptables,
             cov        = input_covariates,
             mastergrid = input_mastergrid,
             watermask = input_watermask,
             px_area     = input_px_area,
             output_dir = output.dir,
             cores      = 2,
             verbose    = TRUE,
             log        = TRUE)
```

Appendix B: The popRFdemo function

The **popRF** package has a demo function "popRFdemo". The function will download the following covariates automatically from WorldPop website and use them to disaggregate population (2020 year) from census units into grid cells.

- *subnational_admin_2000_2020.tif* - Sub-national units
- *esaccilc_dst011_2015.tif* - Distance to ESA-CCI-LC cultivated area edges 2015.
- *esaccilc_dst040_2015.tif* - Distance to ESA-CCI-LC woody-tree area edges 2015.
- *esaccilc_dst130_2015.tif* - Distance to ESA-CCI-LC shrub area edges 2015.
- *esaccilc_dst140_2015.tif* - Distance to ESA-CCI-LC herbaceous area edges 2015.
- *esaccilc_dst150_2015.tif* - Distance to ESA-CCI-LC sparse vegetation area edges 2015.
- *esaccilc_dst160_2015.tif* - Distance to ESA-CCI-LC aquatic vegetation area edges 2015.
- *esaccilc_dst190_2015.tif* - Distance to ESA-CCI-LC artificial surface edges 2015.
- *esaccilc_dst200_2015.tif* - Distance to ESA-CCI-LC bare area edges 2015.
- *esaccilc_dst_water_100m_2000_2012.tif* - ESA-CCI-LC inland waterbodies 2000-2012.
- *coastline_100m_2000_2020.tif* - Distance to coastline 2000-2020.
- *dst_roadintersec_100m_2016.tif* - Distance to OSM major road intersections.
- *dst_waterway_100m_2016.tif* - Distance to OSM major waterways.
- *dst_road_100m_2016.tif* - Distance to OSM major roads.
- *px_area.tif* - Grid-cell surface areas.
- *srtm_slope_100m.tif* - SRTM-based slope 2000 (SRTM is Shuttle Radar Topography Mission).
- *srtm_topo_100m.tif* - SRTM elevation 2000.
- *viirs_100m_2016.tif* - VIIRS night-time lights 2015 (VIIRS is Visible Infrared Imaging Radiometer Suite).
- *wdpa_dst_cat1_100m_2017.tif* - Distance to IUCN strict nature reserve and wilderness area edges 2017.
- *dst_bsgme_100m_2020.tif* - Distance to predicted built-settlement extents in 2020.

The following script will produce a population layer for Senegal (three letter ISO 3166-1 country code: SEN) using 2020 census projection data. Please, note that downloaded all data, depending on the internet connection, can take a long time.

```
library("popRF")

popRFdemo(project_dir = "C:\\Local\\popRF_demo2",
          Country     = "SEN",
          Cores       = 4)
```

Appendix C: Troubleshooting if the RAM is insufficient

The **popRF** package is doing a geoprocessing operation across multiple processes to speed up performance by taking advantage of more than one core. If a user PC does not have enough RAM memory, one can use the additional arguments in popRF() function to process a large dataset.

- "*binc*" – is the parameter to increase number of operational blocks for processing raster files (default is 1).
- "*proximity*" – is the parameter indicating whether proximity measures among the rows should be computed (default is TRUE).

If **popRF** can not process the task, please increase the parameter value of "*binc*" to 2 or 3, etc. as well as turn the parameter "*proximity*" to FALSE.

An example of running the Random Forest application with the additional parameters are below:

```
out <- popRF(
                pop          = input_poptables,
                cov          = input_covariates,
                mastergrid = input_mastergrid,
                watermask = input_watermask,
                px_area     = input_px_area,
                output_dir = output.dir,
                cores        = 2,
                verbose     = TRUE,
                log          = TRUE,
                binc         = 4,
                proximity   = FALSE
              )
```