# Package: hypsoLoop (via r-universe)

September 16, 2024

**Type** Package

**Title** A Tool Used to Conduct Hypsometric Analysis of a Watershed

**Version** 0.2.0

**Author** Faustin GASHAKAMBA

**Maintainer** Faustin GASHAKAMBA <gashakamba@gmail.com>

**Description** Functions for generating tables required for drawing and calculating hypsometric curves and hypsometric integrals. These functions accept as input the DEM of the region of interest (your watershed) and a spatial data frame file specifying delineation of sub-catchments within the watershed. They then generate output in the form of PNG images and HTML files contained in a folder named ``HYPSO_OUTPUT'' created in the current directory. S. K. Sharma, S. Gajbhiye, et al. (2018) <doi:10.1007/978-981-10-5801-1_19>. Omvir Singh, A. Sarangi, and Milap C. Sharma (2006) <doi:10.1007/s11269-008-9242-z>. James A. Vanderwaal and Herbert Ssegane (2013) <doi:10.1111/jawr.12089>.

**Depends** R (>= 4.1.0)

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Imports** PolynomF, dplyr, ggplot2, terra, stars, sf, sjPlot

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**Config/testthat/edition** 3

**NeedsCompilation** no

**Date/Publication** 2022-02-08 09:00:02 UTC

**Repository** https://fgashakamba.r-universe.dev

**RemoteUrl** https://github.com/cran/hypsoLoop

**RemoteRef** HEAD

**RemoteSha** 90d9cd740c6339670c3149b9090bb7530114012d

# Contents



---

`calc_areas` *Function to calculate areas of classes within a categorical raster*

---

## Description

This function takes a categorical raster object and calculates the areas covered by each class within the raster.

## Usage

```
calc_areas(x)
```

## Arguments

| | |
|---|---|
| x | An object of class RasterLayer. |

## Details

The raster package's "area()" function normally returns the same raster where cell values have been replaced by their areas in Square Km. Using this function, the values of all cells of same class are aggregated and the raster is transformed into a data frame. If the CRS of the input raster is projected, then the area is calculated by multiplying the resolution of the raster by the count of cells for each class.

## Value

A data frame with one column representing the values of the classes of the raster and the other variable representing corresponding areas in Ha.

## Author(s)

Faustin Gashakamba

## See Also

area

aggregate

values

## Examples

```
calc_areas(lulcYanze)
```

---

`check_arguments` *Test if user-supplied inputs match the expected arguments type and form*

---

## Description

This function tests whether the input supplied by the user are of the expected type (class), and are in the right form (overlap of extents and same projection).

## Usage

```
check_arguments(x, y)
```

## Arguments

| | |
|---|---|
| x | The input provided by the user for the x argument. |
| y | The input provided by the user for the y argument. |

## Value

There is no return value. If any error is found, the execution is just halted.

---

`drawHypsoCurves` *Draw Hypsometric curves and calculate hypsometric integrals*

---

## Description

This function takes as input the DEM and sub-catchments boundaries and calls the hypsoTables function to produce the hypsometric tables. It then draws and prints out the hypsometric curves for each sub-catchment. Then, it fits a function to the table of each sub-cacthment and uses it to calculate the hypsometric integral. Finally, it summarizes the results in a well-formatted table and prints it out as CSV. All these results are stored in a folder called "HYPSO_OUTPUT" created in the current working directory.

## Usage

```
drawHypsoCurves(x, y, print_result = FALSE)
```

## Arguments

| | |
|---|---|
| x | An object of class SpatialPolygonsDataFrame or simple features (sf). For instance, you can use RGDAL package's "readOGR()" function to create such an object from Shapefiles saved on disk. Alternatively, you can use sf's st_read() function to get a simple features object from the shapefile. NB: at least one column (attribute) named 'Name' should be present in the shapefile or input data frame. |
| y | An object of class RasterLayer. You can use the raster package to read GeoTIFF and other raster formats from disk. |
| print_result | A logical variable to decide whether the output folder will be created to contain the figures and summary table (CSV) or not. |

## Details

The elevation range of each sub-cacthment is divided into 30 contour intervals and the area covered by each contour interval is calculated. The result is put into tables (one table for each sub-catchment). This data is then used to construct the hypsometric curve through ggplot2. A 3rd polynomial function is then fitted to the normalized table and PolynomF package is used to calculate the area under the hypsometric curve (its integral). Finally, the integral values for each sub-catchment are compiled into a data frame that is exported as CSV.

## Value

A data frame containing the hypsometric integral for each sub-catchment along with other data such as maximum & minimum elevation.

## Author(s)

Faustin Gashakamba <gashakamba@gmail.com>

## See Also

lm

## Examples

```
drawHypsoCurves(watersheds, DEM) #Draw hypsographic curves of the sub-catchments in 'watersheds'.
```

`generateHypsoTables` *Generating hypsometric tables*

## Description

This function accepts as input the DEM and sub-catchments boundaries and pre-process this data to extract the tables that are used to calculate the hypsographic curves and integral of these sub-catchments.

## Usage

```
generateHypsoTables(x, y)
```

## Arguments

| | |
|---|---|
| x | An object of class SpatialPolygonsDataFrame or simple features (sf). For instance, you can use RGDAL package's "readOGR()" function to create such an object from Shapefiles saved on disk. Alternatively, you can use sf's st_read() function to get a simple features object from the shapefile. NB: at least one column (attribute) named 'Name' should be present in the shapefile or input data frame. |
| y | An object of class RasterLayer. You can use the raster package to read GeoTIFF and other raster formats from disk. |

## Details

The raster package's "crop()" and "mask()" functions are used to iteratively clip the DEM to each sub-catchment. For each sub-cacthment, the elevation range is divided into 30 equidistant contours. Then, the "area()" function is used to calculate the areas between each contour.

## Value

A list of length 2. The 1st element is a list of data frames, each representing the hypsometric tables for each sub-catchment. The 2nd element is a data frame with min & max elevation values for each sub-catchment.

## Author(s)

Faustin Gashakamba <gashakamba@gmail.com>

## See Also

`crop`

`mask`

`classify`

## raster_conversion *Convert the input raster object into terra's SpatRaster object*

### Description

This function tests whether the supplied vector input is a simple features (sf) object, if not, the object is converted using the st_as-sf() function.

### Usage

```
raster_conversion(y)
```

### Arguments

| | |
|---|---|
| y | The input provided by the user for the y argument. |

### Value

the converted SpatRaster object

## Vector_conversion *Convert the input vector object into simple features if it's provided as a Spatial object*

### Description

This function tests whether the supplied vector input is a simple features (sf) object, if not, the object is converted using the st_as-sf() function.

### Usage

```
Vector_conversion(x)
```

### Arguments

| | |
|---|---|
| x | The input provided by the user for the x argument. |

### Value

the converted simple features object

## Yanze_DEM — *Digital Elevation Model (DEM) of Yanze Watershed*

### Description

This data consists of a 30X30m-resolution digital elevation model (DEM) of Yanze, a small watershed located in north-west of Kigali City, Rwanda. The spatial reference system used by the coordinates of DEM is GCS_WGS_1984 with a D_WGS_1984 datum. The cell values are expressed in Meters with a minimum 1370m of and a maximum of 2241.

### Usage

```
DEM
```

### Format

An S4 object of class stars

### Details

extent: 29.91865, 30.03865, -1.941896, -1.755507 (xmin, xmax, ymin, ymax) dimensions : 671, 432, 289872 (nrow, ncol, ncell)

## Yanze_LULC — *Land Use - Land Cover map of Yanze*

### Description

This data consists of a raster representing the landuse in Yanze watershed, a small watershed located in north-west of Kigali City, Rwanda. he spatial reference system used by the coordinates of DEM is the WGS 84 UTM zone 35S - EPSG:32735. Each land use/land cover class is represented by its code as follows: 1 = Forest 2 = Open Areas or Grass 3 = Agriculture (Seasonal) 5 = Settlements and Buildings 6 = Water 10 = Sparse Forest 11 = Agriculture (Perennial)

### Usage

```
lulcYanze
```

### Format

An S4 object of class stars

### Details

extent: 824728, 838128, -214942.7, -194282.7 (xmin, xmax, ymin, ymax)

## Yanze_sub-catchments *Yanze sub-catchments boundaries*

### Description

This data consists of the boundaries of sub-catchments of Yanze watershed, a small watershed located in north-west of Kigali City, Rwanda. The spatial reference system used by the coordinates of this SpatialPolygonsDataFrame is GCS_WGS_1984 with a D_WGS_1984 datum.

### Usage

```
watersheds
```

### Format

An S4 object of class simple features

### Details

extent: 29.91865, 30.03858, -1.755426, -1.941896 (xmin, xmax, ymin, ymax)

**gridcode** The id of each polygon

**gridcode** The name of each polygon

## Yanze_sub-catchments-table *Yanze sub-catchments table*

### Description

This data consists of a table containing the attributes of sub-catchment of Yanze watersehd.

### Usage

```
watersheds_df
```

### Format

a data frame.

### Details

Each rows represents a sub-catchment in yanze watershed, namely: Mulindi, cyonyonyo, and Yanze downstream.

**gridcode** The name of each sub-catchment

# Index