Emilie CREMIN

University of Glasgow

School of Interdisciplinary Studies

Dumfries campus

25/04/2022 – first version

07/03/2023 – second version

**R for Geoscience Workshop 1**

Logo

Description automatically generated

Table of Contents

[I. R: What is it? What for? for whom? 1](#_Toc129032287)

[II. Our R projects. 2](#_Toc129032288)

[III. Download and install R 3](#_Toc129032289)

[IV. R project development 4](#_Toc129032290)

[0/ Install packages 4](#_Toc129032291)

[1/ ggplot: 5](#_Toc129032292)

[2/ Leaflet: 5](#_Toc129032293)

[3/ Dplyr: 6](#_Toc129032294)

[4/ Map the population Nb/Zila > density/Zila 7](#_Toc129032295)

[5/ Map SDG2 > boro rice 9](#_Toc129032296)

[6/ Map SDG3 > Health 11](#_Toc129032297)

[7/ Now try to map other indicators! Good luck 12](#_Toc129032298)

[V. Make graphs. 12](#_Toc129032299)

## **R: What is it? What for? for whom?**

Quick poll:

How many of you have already used R?

How many of you uses statistics?

How many of you use programming languages?

What is R?

R is a programming language for statistical computing

The R software is Open-source, meaning that is free to use but also to modify and distribute

It has a wide community of contributors who have developed many tools that we call packages.

For more details read:

Geocomputation with R: <https://geocompr.robinlovelace.net/>

|  |  |
| --- | --- |
| [Diagram  Description automatically generated](https://geocompr.robinlovelace.net/) | [Welcome | Geocomputation with R](https://geocompr.robinlovelace.net/)  Welcome. This is the online home of Geocomputation with R, a book on geographic data analysis, visualization and modeling.. Note: The first edition of the book has been published by CRC Press in the R Series.You can buy the book from CRC Press, or Amazon, and see the archived First Edition hosted on bookdown.org.. Inspired by the Free and Open Source Software for Geospatial movement, the code ...  geocompr.robinlovelace.net |

USGS tutorial

<https://owi.usgs.gov/R/training-curriculum/intro-curriculum/index.html>

What for?

For data analysis and statistics

Some packages have been specifically created to address the needs for analysing and visualising geographical data.

Because it is based on code, an R project allows to:

* Process large amount of data
* Automate repetitive tasks
* Replicate and reuse code for data processing on different data sets
* Distribute the code – for instance as an appendix of a research paper allowing other researchers
  + to reproduce the processing and verify the results
  + to reuse and improve

## **Our R projects.**

How do you plan to use R? For what research?

Emilie: I’m using R to compute the data for the Global Delta Risk Index – The revised index (2022) is composed of 140 indicators to assess risk: hazard x exposure x vulnerability (social susceptibility/ Adaptation capacities/ Ecological robustness/ Ecological sensitivity or susceptibility)

The indicators are applied to 4 deltas region – coastal areas: The Red and the Mekong River Deltas in Vietnam, the GBM-B and the GBM-I (India and Bangladesh)

Therefore, we use the spatial boundaries at the level 2 (Zila/province) and level 3 (villages) as much as possible

In Bangladesh there are 64 Zilas > see the shapefile used in this tutorial

Social data are mainly informed by census: vectors

Ecological data are mainly informed by satellite images: rasters

You can explain here what you’d like to do with R

## **Download and install R**

Ahead of the workshop,

please: install R, R studio

A/ Install

A1/ # Install R

## MacOS

https://cloud.r-project.org/bin/macosx/base/R-4.1.3.pkg

## Windows

https://cloud.r-project.org/bin/windows/base/

## Linux

https://cloud.r-project.org/bin/linux/

A2/ # Install R Studio

## MacOS

https://download1.rstudio.org/desktop/macos/RStudio-2022.02.1-461.dmg

## Windows

https://download1.rstudio.org/desktop/windows/RStudio-2022.02.1-461.exe

## Linux

<https://www.rstudio.com/products/rstudio/download/#download>

B/ Data files to Download

B1/ Boundaries at the subnational level

On the webpage: <https://data.humdata.org/dataset/cod-ab-bgd>

download the following file:

<https://data.humdata.org/dataset/401d3fae-4262-48c9-891f-461fd776d49b/resource/08736818-ae72-44a9-abd6-a51915c24921/download/bgd_adm_bbs_20201113_shp.zip>

B2/ SDG Data

On the page: <https://www.worldbank.org/en/news/feature/2022/03/14/spatial-database-indicators-by-sdg>

download:

<https://thedocs.worldbank.org/en/doc/9c1d0b1657e3694b19dbd37c6f488acb-0310012022/original/Zila-SDG-v2.xlsx>

=> store all the files in data folder

C/ If you need Help

stack exchange

RDocumentation

D/ Open the project

RStudio > File > Open project…

Project name: Tutorials.Rproj

Note you can also create your own project

RStudio > File > New Project

## **R project development**

### 0/ Install packages

Open File> 0\_install.R

Install packages

install.packages("leaflet")

install.packages("sf")

install.packages("ggplot2")

install.packages("dplyr")

install.packages("stringr")

To be install later if needed for raster analysis:

Package: raster, stars, terra

### 1/ ggplot:

**ggplot2** is an [open-source](https://en.wikipedia.org/wiki/Open-source) [data visualization](https://en.wikipedia.org/wiki/Data_visualization) [package](https://en.wikipedia.org/wiki/R_package) for the [statistical programming](https://en.wikipedia.org/wiki/Computational_statistics) language [R](https://en.wikipedia.org/wiki/R_(programming_language)). (wikipedia : <https://en.wikipedia.org/wiki/Ggplot2> )

Open file > 1\_ggplot.R

Code as below

library(sf)

library(ggplot2)

bgd\_shp <- st\_read("data/bgd\_adm\_bbs\_20201113\_SHP/bgd\_admbnda\_adm2\_bbs\_20201113.shp")

head(bgd\_shp)

ggplot() +

geom\_sf(data=bgd\_shp) +

ggtitle("Bangladesh Boundary Plot") +

coord\_sf()

Expected output: Map 1 – Bangladesh boundary plot

Figure 1: Expected output 1

Chart, diagram

Description automatically generated with medium confidence

### 2/ Leaflet:

**Leaflet** is an [open source](https://en.wikipedia.org/wiki/Open-source_license) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) [library](https://en.wikipedia.org/wiki/Library_(computing)) used to build [web mapping](https://en.wikipedia.org/wiki/Web_mapping) [applications](https://en.wikipedia.org/wiki/Application_software). (wikipedia : <https://en.wikipedia.org/wiki/Leaflet_(software)> )

Open> 2\_leaflet.R

Code as below

library(leaflet)

library(sf)

bgd\_shp <- st\_read("data/bgd\_adm\_bbs\_20201113\_SHP/bgd\_admbnda\_adm2\_bbs\_20201113.shp")

head(bgd\_shp)

leaflet() %>%

addTiles() %>%

addPolygons(data = bgd\_shp,

label = bgd\_shp$ADMIN\_NAME, weight=1, col = 'green')

Figure 2: Expected output 2

Map

Description automatically generated

### 3/ Dplyr:

One of the core packages of the [tidyverse](https://en.wikipedia.org/wiki/Tidyverse" \o "Tidyverse) in the [R programming language](https://en.wikipedia.org/wiki/R_(programming_language)), **dplyr** is primarily a set of functions designed to enable dataframe manipulation in an intuitive, user-friendly way. Data analysts typically use dplyr in order to transform existing datasets into a format better suited for some particular type of analysis, or data visualization.[[1]](https://en.wikipedia.org/wiki/Dplyr#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Dplyr#cite_note-2)

(Wikipedia: <https://en.wikipedia.org/wiki/Dplyr>)

While dplyr actually includes several dozen functions that enable various forms of data manipulation, the package features five primary verbs:[[5]](https://en.wikipedia.org/wiki/Dplyr#cite_note-5)

**filter()**, which is used to extract rows from a dataframe, based on conditions specified by a user;

**select()**, which is used to subset a dataframe by its columns;

**arrange()**, which is used to sort rows in a dataframe based on attributes held by particular columns;

**mutate()**, which is used to create new variables, by altering and/or combining values from existing columns; and

**summarize()**, also spelled **summarise()**, which is used to collapse values from a dataframe into a single summary.

Here we use **filter**

Open > 3\_dplyr.R

library(dplyr)

library(ggplot2)

library(sf)

bgd\_shp <- st\_read("data/bgd\_adm\_bbs\_20201113\_SHP/bgd\_admbnda\_adm2\_bbs\_20201113.shp")

head(bgd\_shp)

aoi <- bgd\_shp %>% dplyr::filter(ADM2\_EN=="Khulna")

head(aoi)

ggplot() +

geom\_sf(data=aoi) +

ggtitle("Khulna Boundary Plot") +

coord\_sf()

Figure 3: Expected output 3

Diagram

Description automatically generated

### 4/ Map the population Nb/Zila > density/Zila

Open > 4\_population.R

library(dplyr)

library(ggplot2)

library(leaflet)

library(sf)

library(stringr)

bgd\_pop <- read.csv(file = "data/SDGs/population.csv")

head(bgd\_pop)

bgd\_shp <- read\_sf(dsn = "data/bgd\_adm\_bbs\_20201113\_SHP/bgd\_admbnda\_adm2\_bbs\_20201113.shp")

head(bgd\_shp)

bgd\_pop\_2011 <- bgd\_pop %>% dplyr::filter(Year==2011 & Definition=="Total Population in 2011")

head(bgd\_pop\_2011)

bgd\_pop\_2011$ADM2\_PCODE <- paste("BD", str\_pad(bgd\_pop\_2011$DivisionCode, 2, pad = "0"), str\_pad(bgd\_pop\_2011$ZilaCode, 2, pad = "0"), sep="")

head(bgd\_pop\_2011)

m <- bgd\_pop\_2011 %>% full\_join(y = bgd\_shp, by = "ADM2\_PCODE") %>%

st\_as\_sf()

head(m)

# For each Zila

# 1. Calculate Geometry area in km^2

# 2. Calculate pop density

ggplot() +

geom\_sf(data = m, aes(fill = Estimate)) +

scale\_fill\_gradient(name="Population", low = "yellow", high = "red", na.value = NA, breaks = scales::breaks\_extended(), labels = scales::comma) +

ggtitle("Bangladesh Population") +

coord\_sf()

# example

# https://rstudio.github.io/leaflet/choropleths.html

labels <- sprintf(

"<strong>%s</strong><br/>%g Millions",

m$Zila, round(m$Estimate / 10^6, digits = 2)

) %>% lapply(htmltools::HTML)

leaflet() %>%

addTiles() %>%

addPolygons(data = m,

label = labels, weight=1, col = 'green')

Figure 4: Expected output 4

Chart

Description automatically generated

### 5/ Map SDG2 > boro rice

Open > 5\_agriculture.R

bgd\_agriculture <- read.csv(file = "data/SDGs/agriculture.csv")

head(bgd\_agriculture)

bgd\_shp <- read\_sf(dsn = "data/bgd\_adm\_bbs\_20201113\_SHP/bgd\_admbnda\_adm2\_bbs\_20201113.shp")

bgd\_boro\_rice <- bgd\_agriculture %>% dplyr::filter(Indicator=="Boro Rice (Local) Production (Metric ton)")

head(bgd\_boro\_rice)

addPostcode <- function(df) {

df$ADM2\_PCODE <- paste("BD", str\_pad(df$DivisionCode, 2, pad = "0"), str\_pad(df$ZilaCode, 2, pad = "0"), sep="")

return(df)

}

bgd\_boro\_rice <- addPostcode(bgd\_boro\_rice)

head(bgd\_boro\_rice)

joinOnPostcode <- function(df, shp) {

return(

df %>%

full\_join(y = shp, by = "ADM2\_PCODE") %>%

st\_as\_sf()

)

}

m <- joinOnPostcode(bgd\_boro\_rice, bgd\_shp)

head(m)

mapPlot <- function(shp, columnName, unit, title) {

ggplot() +

geom\_sf(data = shp, aes(fill = eval(as.name(columnName)))) +

scale\_fill\_gradient(name=unit, low = "yellow", high = "red", na.value = NA, breaks = scales::breaks\_extended(), labels = scales::comma) +

ggtitle(title) +

coord\_sf()

}

mapPlot(m, "Estimate", "Metric tons", "Bangladesh Local Boro Rice production")

Figure 5: Expected output 5

Chart

Description automatically generated

### 6/ Map SDG3 > Health

Open 6\_health

library(dplyr)

library(ggplot2)

library(sf)

library(stringr)

source("helpers.R")

bgd\_health <- read.csv(file = "data/SDGs/Health.csv")

bgd\_shp <- read\_sf(dsn = "./data/bgd\_adm\_bbs\_20201113\_SHP/bgd\_admbnda\_adm2\_bbs\_20201113.shp")

bgd\_fertility\_rate <- bgd\_health %>% dplyr::filter(Indicator=="General fertility rate (Number per 1000 women)")

bgd\_fertility\_rate <- addPostcode(bgd\_fertility\_rate)

m <- joinOnPostcode(bgd\_fertility\_rate, bgd\_shp)

mapPlot(m, "Estimate", "Nb per 1000 women", "Fertility rate")

### 7/ Now try to map other indicators! Good luck

## **Make graphs.**

<https://r-graph-gallery.com/>

Basic histogram with geom\_histogram

It is relatively straightforward to build a histogram with ggplot2 thanks to the geom\_histogram() function. Only one numeric variable is needed in the input. Note that a warning message is triggered with this code: we need to take care of the bin width as explained in the next section.

*# library*

**library**(ggplot2)

*# dataset:*

data=**data.frame**(value=**rnorm**(100))

*# basic histogram*

p <- **ggplot**(data, **aes**(x=value)) +

**geom\_histogram**()

*#p*

# Libraries

library(tidyverse)

library(hrbrthemes)

# Load dataset from github

data <- read.table("https://raw.githubusercontent.com/holtzy/data\_to\_viz/master/Example\_dataset/1\_OneNum.csv", header=TRUE)

# plot

p <- data %>%

filter( price<300 ) %>%

ggplot( aes(x=price)) +

geom\_histogram( binwidth=3, fill="#69b3a2", color="#e9ecef", alpha=0.9) +

ggtitle("Bin size = 3") +

theme\_ipsum() +

theme(

plot.title = element\_text(size=15)

)

#p

1. **Basics of coding**
   1. data structures and operations
      1. numbers: integers, doubles, floats (+ - / \*)
      2. characters: strings (concatenation “hello”+” ”+”world”, contains, strip, replace)
      3. Lists: array, map (loop iterations, key => value attributes)

vehicules = { “cars”: [“Peugeot 207”, “Jaguar XR”], “aeroplanes”: [“Airbus 320”, “Boeign 747”] }

* 1. Functions

function add(a,b) { return a + b }

result = add(1,1)

function greeting(name) { return “Hello ”+name }

greeting(“Emilie”)

1. R dataframes
2. **R used for other spatial analysis**

Raster support satellite images, DEM and other types of data

It is an image composed of pixel – each pixel has a value

Data visualisation – mapping

Final cartography

References:

Geocomputation with R : <https://geocompr.robinlovelace.net/index.html>

R Programming tutorial: <https://www.youtube.com/watch?v=_V8eKsto3Ug>