

GSMaP Daily Data Historic Year Extraction

Yehezkiel

October 3, 2022

1 Introduction

To extract GSMap historical data from GSMap standard version 6 daily_grev 00Z data.

2 Script

First, clear the environment.

```
rm(list=ls(all=TRUE))
```

Then, import libraries.

```
library(sp)                #library for spatial data
library(raster)            #library for raster data
library(tidyverse)
library(R.utils)           #unzipping files
```

Import village to pixel mapping file.

```
#script directory
script_dir <- dirname(rstudioapi::getSourceEditorContext()$path)

#village to pixel mapping file location
reference_pixel_file <- paste0(
  substr(script_dir, 1, unlist(gregexpr("/2. script", script_dir))),
  "1. historical Indonesia\\village_to_pixel_final gsmap.csv"
)

#import village to pixel mapping file
reference_pixel_cite <- read.csv(reference_pixel_file)
```

Then, search for each unique pixel coordinate.

```
#unique pixel name to search for pixel coordinate
pixels <- unique(reference_pixel_cite$pixel_name)
longitude <- c()
latitude <- c()

#search for each longitude and latitidue pixel coordinate
for(i in pixels){
  pixel_loc <- reference_pixel_cite[which(
    reference_pixel_cite$pixel_name == i
  ),]

  longitude <- c(longitude, pixel_loc$pixel_long[1])
}
```

```
latitude <- c(latitude, pixel_loc$pixel_lat[1])
}
```

Create open and close connection function for the gsmmap data.

```
#open and close connection
read_gsmmap <- function(fname, nvals){
  ## gsmmap daily world data is 4 byte little endian float
  on.exit(close(flcon))
  flcon <- file(fname, 'rb')
  readBin(flcon, "double", n = nvals, endian = "little", size = 4)
}
```

Create a function to convert the data into raster data.

```
#convert the data into raster data
get.raster <- function(GSMaP.data){
  nrow <- 1200
  ncol <- 3600
  data.dat.file <- matrix(read_gsmmap(GSMaP.data, (nrow * ncol)),
                          nrow = nrow,
                          ncol = ncol,
                          byrow = TRUE)
  raster.dat.file <- raster(data.dat.file,
                           xmn = -180,
                           xmx = 180,
                           ymn = -60,
                           ymx = 60,
                           crs = CRS("+init=epsg:4326"))

  cell.index <- cellFromXY(raster.dat.file, coordinates.in.use)
  data.extracted <- t(raster :: extract(raster.dat.file, cell.index))

  return(data.extracted)
}
```

Create a function to unzip the data from gz zip file and also check whether the data is already extracted or not.

```
#unzip the data
file.check <- function(gs.file, month){
  if(substr(gs.file[1], nchar(gs.file[1]) - 5, nchar(gs.file[1])) == "
    dat.gz"){
    dat.file <- gunzip(paste0(wd.year,
                              "/", month,
                              "/", gs.file))
  }
}
```

```

    return(dat.file)
  } else {
    dat.file <- paste0(wd.year,
                      "/", month,
                      "/", gs.file)
    return(dat.file)
  }
}

```

Search for the list of folder inside the raw GSMaP dataset folder.

```

#raw GSMaP dataset directory
wd.year <- paste0(
  substr(script_dir, 1, unlist(gregexpr("/2. script", script_dir))),
  "3. V6 daily_grev 00Z data"
)

#list folder inside the directory
list.month <- list.files(wd.year)

```

Create an empty dataframe to store value.

```

#empty data frame to store value
final.data <- data.frame(unique(reference_pixel_cite$pixel_name))
final.data <- t(final.data)

```

Create a for loop to extract the data.

```
#for looping the folder
for(j in 1:length(list.month)){
  ##day directory
  wd.day <- paste0(wd.year,
                  "/",
                  list.month[j])
  ##list of day folder
  list.day <- list.files(wd.day)
  for(k in 1:length(list.day)){
    ##keep track of the file
    cat("month=",
        list.month[j],
        "files=",
        list.day[k],
        "\n")

    ##file's directory
    dat.file <- file.check(list.day[k],
                          list.month[j])          #unzipping the .dat
                                                    files
    ##access the function and append the data to final.data dataframe
    final.data <- rbind(final.data,
                       get.raster(dat.file)*24)
    ##change the row names
    file_date <- as.Date(paste0(list.month[j], " - ", k), "%Y%m - %d")
    file_date <- format(file_date, "%d-%b-%y")
    rownames(final.data)[dim(final.data)[1]] <- file_date
  }
}
```

Data cleaning.

```
#change the column names
colnames(final.data) <- final.data[1,]
#erase the first row of the data frame
final.data <- final.data[-1,]
```

Export the data.

```
save(final.data, file = "final_historical_indo_gsmmap_daily.RData")
```