Galamsey

Boniface Kalong

7/2/2022

```
#if(!require("pacman")){install.packages("pacman")}
pacman::p_load(char = c('rgee','reticulate','raster','tidyverse','dplyr','sf','mapview','mapeddit','car
```

```
## Warning in pacman::p_load(char = c("rgee", "reticulate", "raster", "tidyverse", : Failed to install/
## tidyverse, mapeddit
```

```
#library(pacman)
#pacman::p_load(char = c('googleAnalyticR','googleAuthR','googleCloudRunner','googlePublicData','google
```

```
library(rgee)

library(reticulate)
#ee_install()
ee_check()
```

```
## (*)  Python version
## v [Ok] C:/Users/Guy/.conda/envs/rgee/python.exe v3.8
## (*)  Python packages:
## v [Ok] numpy
## v [Ok] earthengine-api
```

```
ee_Initialize("kalong",drive = TRUE) # initialize GEE, this will have you log in to Google Drive
```

```
## -- rgee 1.1.4 --------------------------------------- earthengine-api 0.1.317 --
##  v user: kalong
##  v Google Drive credentials: v Google Drive credentials:  FOUND
##  v Initializing Google Earth Engine: v Initializing Google Earth Engine:  DONE!
##  v Earth Engine account: users/Earth_Science
## --------------------------------------------------------------------------------
```

Time Series Analysis of Satellite Derived Vegetation Indices in R: A Case Study in Quantifying the Status of Galamsey In Ghana With Time Series Classification.

Adstract

This study examines whether MODIS (NDVI,EVI) satellite data time series can be used to detect land degradation and regeneration areas in Ghana. Time series analysis was applied to more than 20 years of satellite data record, based on the hypothesis that the resulting NDVI residual trend vectors would enable successful detection of changes in photosynthetically active vegetation. We performed regression analysis, derived regression slope values, and generated a map of significant trends. We also examined land cover development and meteorological data for the same period.

11-year time series of MODIS 16-day composite NDVI data proved sufficient for deriving statistically significant trend values for 50% of Mongolia's surface. MODIS land cover products proved suitable for identifying areas of vegetation cover change. Areas showing positive and negative NDVI trends mostly coincided with areas of land cover class change indicating an increase or a decrease in vegetation, respectively. Precipitation changes in the same time period seem to have had an influence on large NDVI trend areas. The NDVI time series trend analysis methodology applied successfully detected changes due to deforestation, forest fires, mining activities, urban expansion, and grassland regeneration. These findings demonstrate that NDVI time series trend analysis is suitable for detecting vegetation change areas and for identifying land degradation and regeneration.

Introduction:

All thing change, but how we respond to change is our responsibility, to fare it or embrasse it. Resisting change leads to one fiat. Our own extinction. Time is a smybole of freedom and peace

The purpose of this paper is to establish an understanding in time series analysis on remotely sensed data. Which will introduced us to the fundamentals of time series modeling, including decomposition, autocorrelation and modeling historical changes in Galamsey Operation in Ghana, the Cause,Dangers and it's Environmental impact. Galamsey("gather them and sell"),(OwusuNimo2018) is the term given by local Ghanaian for illegal small-scale gold mining in Ghana (DavidYawDanquah2019). The major cause of Galamsey is unemployment among the youth in Ghana(Gracia2018). Young university graduates rarely find work and when they do it hardly sustains them. The result is that these youth go the extra mile to earn a living for themselves and their family. Another factor is that lack of job security.

On November 13, 2009 a collapse occurred in an illegal, privately owned mine in Dompoase, in the Ashanti Region of Ghana. At least 18 workers were killed, including 13 women, who worked as porters for the miners. Officials described the disaster as the worst mine collapse in Ghanaian history(News2009).

Illegal mining causes damage to the land and water supply(Ansah2017). In March 2017, the Minister of Lands and Natural Resources, Mr. John Peter Amewu, gave the Galamsey operators/illegal miners a three-week ultimatum to stop their activities or be prepared to face the law(Allotey2017). The activities by Galamseyers have depleted Ghana's forest cover and they have caused water pollution, due to the crude and unregulated nature of the mining process(Gyekye2021).

Under current Ghanaian constitution, it is illegal to operate as galamseyer.That is to dig on land granted to mining companies as concessions or licenses and any other land in search for gold. In some cases, Galamseyers are the first to discover and work extensive gold deposits before mining companies find out and take over. Galamseyers are the main indicator of the presence of gold in free metallic dust form or they process oxide or sulfide gold ore using liquid mercury. Between 20,000 to 50,000, including thousands from China are believed to be engaged in Galamsey in Ghana.But according to the Information Minister 200,000 and nearly 3 million people, recently are now into Galamsey operation and rely on it for their livelihoods(Burrows2017). Their operations are mostly in the southern part of Ghana where it is believe to have substantial reserves of gold deposits, usually within the area of large mining companies(Barenblitt2021). As a group, they are economically disad vantaged. Galamsey settlements are usually poorer than neighboring agricultural villages. They have high rates of accidents and are exposed to mercury poisoning from their crude processing methods. Many women are among the workers, acting mostly as porters for the miners.

Problem Statement

The Footprint of Galamsey is Spreading at a very faster rate, causing vegetation loss.Other factors accounting to vegetation loss may largely include climate change,urban and exurban development, bush fires. But not much works or research has been done to tell the extent to which Galamsey causes vegetation loss. This research attempts to segregate the variability climate is responsible for in vegetation loss so as to attribute the residual variability to Galamsey and other related activities such as bush-fires etc.

Objectives

Highlights

- MODIS NDVI time series were used to identify land degradation and regeneration areas.
- MODIS land cover class changes coincide with MODIS NDVI trend areas.
- Changes in precipitation might have an influence on large NDVI trend areas.
- Short-term changes in temperature have no influence on NDVI trend areas.

Overview

The purpose is to establish an understanding in time series analysis on remotely sensed data. We will be introduced to the fundamentals of time series modeling, including decomposition, autocorrelation and modeling historical changes.

- Perform time series analysis on satellite derived vegetation indices
- Estimate the extent to which Galamsey causes vegetation loss
- Dissociate or single out the variability climate is responsible for in vegetation loss
- Create a Statistical interactive dashboard ## Significance Of the Study ## Limitations Time series modeling aims to build an explanatory model of the data without over fitting the problem set, to use as simple a model as possible while accounting for as much of the data as possible. When breaking down time series data into component parts, remote sensing data has additional limitations that make this more challenging. It is almost inevitable that you will not get this same level of precision from remote sensing data. Additionally, atmospheric conditions can skew the visual results, where the hue of the vegetation changes drastically from image to image due to atmospheric conditions (fog,ground moisture, cloud cover). ## Data collection and Methodology # Data As Galamsey is considered an illegal activity, they operations are hidden to the eyes of the authorities.So locating them is quite tricky ,but with satellite imagery ,it now possible to locate their operating and put an end to it. One of the features of Google Earth Engine is the ability to access years of satellite imagery without needing to download, organize, store and process this information. For instance, within the Satellite image

collection, now it possible to access imagery back to the 90's, allowing us to look at areas of interest on the map to visualize and quantify how much things has changed over time. With Earth Engine, Google maintains the data and offers it's computing power for processing.Users can now access hundreds of time series images and analyze changes across decades using GIS and R or other programming language to analyze these datasets.

```r
library('sf')
# Load shape file

#setwd("C:/Users/Guy/Documents/GitHub/Artisanal-Mining-In-Ghana-Galamsey/New Regions")
aoi <- read_sf('Ghana shp file/GHA/gadm41_GHA_1.shp')
aoi <- st_transform(aoi, st_crs(4326))
aoi.ee <- st_bbox(aoi) %>%
st_as_sfc() %>%
sf_as_ee() #Converts it to an Earth Engine Object
```

These functions return the QA value from MODIS imagery and apply a quality Mask, returning quality masked EVI values, this technique was adapted from one presented by Cesar Aybar (one of the rgee authors) here.

```r
getQABits <- function(image, qa) {
  # Convert binary (character) to decimal (little endian)
  qa <- sum(2^(which(rev(unlist(strsplit(as.character(qa), "")) == 1))-1))
  # Return a mask band image, giving the qa value.
  image$bitwiseAnd(qa)$lt(1)
}

mod.clean <- function(img) {
  # Extract the NDVI band
  ndvi_values <- img$select("EVI")
  # Extract the quality band
  ndvi_qa <- img$select("SummaryQA")
  # Select pixels to mask
  quality_mask <- getQABits(ndvi_qa, "11")
  # Mask pixels with value zero.
  ndvi_values$updateMask(quality_mask)$divide(ee$Image$constant(10000)) #0.0001 is the MODIS Scale Fact
}

modis.evi <- ee$ImageCollection("MODIS/006/MOD13Q1")$filter(ee$Filter$date('2000-01-01','2022-01-01'))$m
```

Now we will create a hexagonal grid over the study area

```
library(tibble)
aoi.proj <- st_transform(aoi, st_crs(2392))
hex <- st_make_grid(x = aoi.proj, cellsize = 17280, square = FALSE) %>%
st_sf() %>%
rowid_to_column('hex_id')
hex <- hex[aoi.proj,]
plot(hex)
```

Galamsey_files/figure-latex/unnamed-chunk-7-1.pdf

Now we will use the grid created above to extract the mean EVI values within each cell for the years 2000-2020. The finer the spatial resolution, the longer it will take, this next chunk should take 30 minutes.

```
#This will take about 30 minutes
if(readline(prompt = "Hit enter to proceed or type 'no' to download the data from G-Drive. ") == "no"){
googledrive::drive_download(file =
                              googledrive::as_id("https://drive.google.com/drive/folders/1ZnCpYz38ezSU1
  #https://drive.google.com/drive/folders/1ZnCpYz38ezSU1XG7ixJ2sPg_DX7bO07J?usp=sharing
evi.df <- read.csv("rgee_file_2d44527a3b0e_2022_07_28_15_40_52.csv")
evi.df <- evi.df[,3:ncol(evi.df)]
colnames(evi.df) <- c('hex_id', stringr::str_replace_all(substr(colnames(evi.df[, 2:ncol(evi.df)]), 2,
} else {
#This will take about 30 minutes
paste0(system.time(expr = aoi.evi <- ee_extract(x = modis.evi, y = hex["hex_id"], sf = FALSE, scale = 25
evi.df <- as.data.frame(aoi.evi)
colnames(evi.df) <- c('hex_id', stringr::str_replace_all(substr(colnames(evi.df[, 2:ncol(evi.df)]), 2,
write.csv(x = evi.df, file = "~/rgee_file_2d44527a3b0e_2022_07_28_15_40_52.csv")}
```

```
## Hit enter to proceed or type 'no' to download the data from G-Drive.
```

```
## File downloaded:
```

```
## * 'rgee_file_304c670c7f13_2022_08_09_18_51_32.csv'
##   <id: 13zbxGf_1v50FP6pGLTAHGpviP4fpAZ1M>
```

```
## Saved locally as:
```

```
## * 'C:/Users/Guy/AppData/Local/Temp/RtmpCURYhK/rgee_file_304c670c7f13.csv'
```

Now we are going to perform a time series analysis on the data within each grid cell. But first, we will work through the procedure one step at a time.

```
evi.hw.lst <- list() #Create an empty list, this will be used to house the time series projections for
evi.dcmp.lst <- list() #Create an empty list, this will be used to house the time series decomposition
evi.trend <- data.frame(hex_id = evi.df$hex_id, na.cnt = NA, na.cnt.2 = NA, trend = NA, p.val = NA, r2
Dates <- data.frame(date = seq(as.Date('2001-01-01'), as.Date('2019-11-01'), "month"))
Dates$month <- month(Dates$date)
Dates$year <- year(Dates$date)
i <- 1
tsv <- data.frame(evi = t(evi.df[i, 2:ncol(evi.df)])) #converting the data to a transposed data frame
colnames(tsv) <- c("evi")
#write.csv(tsv,"Data/tsv.csv")
head(tsv) #let's take a look
```

```
##                  evi
## 2001-01-17 0.3103816
## 2001-03-22 0.6017811
## 2001-04-23 0.5585050
## 2002-01-17 0.3728227
## 2002-02-02 0.4369971
## 2002-04-07 0.5701539
```

Method

Time series data is the collection of observations made sequentially at different points in time.Because data points in time series are collected at adjacent time periods there is potential for correlation between observations. we propose some new tools to allow machine learning classifiers to cope with time series data. We first argue that, time-series classification problems can be solved by detecting and combining local properties or patterns in time series. Then, a technique is proposed to find patterns which are useful for classification. These patterns are combined to build interpretable classification rules. First, we will pull Sentinel 2 to select NDVI and EVI data from Google Earth Engine,applying a quality filter to mask poor quality pixels.Instead of performing our analysis on the imagery itself, we will be summarizing the mean NDVI and EVI value , this will allow the analysis to take less time while producing a visually appealing and informative map.Some cells may not contain NDVI and EVI for a given month, to correct this, we will apply smoothing method using an ARIMA function. Once NA values are remove, we will decompose the time series to remove seasonality and fit a linear model to the normalized data. Once we have extracted the linear trend, we will then make a move to classifier our dataon the map and map it.

```
na.cnt <- length(tsv[is.na(tsv)]) #We want to get an idea of the number of entries with no EVI value
evi.trend$na.cnt[i] <- na.cnt
td <- tsv %>%
  mutate(month = month(as.Date(rownames(tsv))), year = year(as.Date(rownames(tsv)))) %>%
  group_by(year, month) %>%
  summarise(mean_evi = mean(evi, na.rm = T), .groups = "keep") %>%
  as.data.frame()
head(td)
```

```
##   year month  mean_evi
## 1 2001     1 0.3103816
## 2 2001     3 0.6017811
## 3 2001     4 0.5585050
## 4 2002     1 0.3728227
## 5 2002     2 0.4369971
## 6 2002     4 0.6160278
```

That looks better! Unfortunately though, there are a number of dates which don't have any evi value at all, let's figure out which ones these are.

```
td$date <- as.Date(paste0(td$year, "-", td$month, "-01"))
dx <- Dates[!(Dates$date %in% td$date),]
dx
```

```
##            date month year
## 2   2001-02-01     2 2001
## 5   2001-05-01     5 2001
## 6   2001-06-01     6 2001
## 7   2001-07-01     7 2001
## 8   2001-08-01     8 2001
## 9   2001-09-01     9 2001
## 10  2001-10-01    10 2001
## 11  2001-11-01    11 2001
## 12  2001-12-01    12 2001
## 15  2002-03-01     3 2002
## 17  2002-05-01     5 2002
## 18  2002-06-01     6 2002
## 19  2002-07-01     7 2002
## 20  2002-08-01     8 2002
## 21  2002-09-01     9 2002
## 22  2002-10-01    10 2002
## 23  2002-11-01    11 2002
## 29  2003-05-01     5 2003
## 31  2003-07-01     7 2003
## 32  2003-08-01     8 2003
## 33  2003-09-01     9 2003
## 34  2003-10-01    10 2003
## 35  2003-11-01    11 2003
## 37  2004-01-01     1 2004
## 39  2004-03-01     3 2004
## 41  2004-05-01     5 2004
## 42  2004-06-01     6 2004
## 44  2004-08-01     8 2004
## 45  2004-09-01     9 2004
## 47  2004-11-01    11 2004
## 48  2004-12-01    12 2004
## 50  2005-02-01     2 2005
## 53  2005-05-01     5 2005
## 54  2005-06-01     6 2005
## 55  2005-07-01     7 2005
## 56  2005-08-01     8 2005
## 57  2005-09-01     9 2005
## 58  2005-10-01    10 2005
## 59  2005-11-01    11 2005
## 63  2006-03-01     3 2006
## 65  2006-05-01     5 2006
## 66  2006-06-01     6 2006
## 67  2006-07-01     7 2006
## 68  2006-08-01     8 2006
## 69  2006-09-01     9 2006
## 70  2006-10-01    10 2006
## 71  2006-11-01    11 2006
## 72  2006-12-01    12 2006
## 77  2007-05-01     5 2007
## 78  2007-06-01     6 2007
## 79  2007-07-01     7 2007
## 80  2007-08-01     8 2007
## 81  2007-09-01     9 2007
## 82  2007-10-01    10 2007
## 87  2008-03-01     3 2008
## 91  2008-07-01     7 2008
## 92  2008-08-01     8 2008
## 93  2008-09-01     9 2008
## 94  2008-10-01    10 2008
## 95  2008-11-01    11 2008
## 99  2009-03-01     3 2009
## 102 2009-06-01     6 2009
## 103 2009-07-01     7 2009
## 104 2009-08-01     8 2009
## 105 2009-09-01     9 2009
## 106 2009-10-01    10 2009
## 107 2009-11-01    11 2009
## 111 2010-03-01     3 2010
## 113 2010-05-01     5 2010
## 114 2010-06-01     6 2010
## 115 2010-07-01     7 2010
## 116 2010-08-01     8 2010
## 117 2010-09-01     9 2010
## 118 2010-10-01    10 2010
## 119 2010-11-01    11 2010
## 120 2010-12-01    12 2010
## 121 2011-01-01     1 2011
## 124 2011-04-01     4 2011
## 125 2011-05-01     5 2011
## 126 2011-06-01     6 2011
## 127 2011-07-01     7 2011
## 128 2011-08-01     8 2011
## 129 2011-09-01     9 2011
## 130 2011-10-01    10 2011
## 131 2011-11-01    11 2011
## 137 2012-05-01     5 2012
## 138 2012-06-01     6 2012
## 139 2012-07-01     7 2012
## 140 2012-08-01     8 2012
## 141 2012-09-01     9 2012
## 142 2012-10-01    10 2012
## 143 2012-11-01    11 2012
## 149 2013-05-01     5 2013
## 150 2013-06-01     6 2013
## 151 2013-07-01     7 2013
## 152 2013-08-01     8 2013
## 153 2013-09-01     9 2013
## 154 2013-10-01    10 2013
## 155 2013-11-01    11 2013
## 157 2014-01-01     1 2014
## 158 2014-02-01     2 2014
## 161 2014-05-01     5 2014
## 162 2014-06-01     6 2014
## 163 2014-07-01     7 2014
## 164 2014-08-01     8 2014
## 165 2014-09-01     9 2014
## 166 2014-10-01    10 2014
## 169 2015-01-01     1 2015
## 170 2015-02-01     2 2015
## 172 2015-04-01     4 2015
## 173 2015-05-01     5 2015
## 174 2015-06-01     6 2015
## 175 2015-07-01     7 2015
## 176 2015-08-01     8 2015
## 177 2015-09-01     9 2015
## 178 2015-10-01    10 2015
## 179 2015-11-01    11 2015
## 180 2015-12-01    12 2015
## 181 2016-01-01     1 2016
## 182 2016-02-01     2 2016
## 184 2016-04-01     4 2016
## 185 2016-05-01     5 2016
## 186 2016-06-01     6 2016
## 187 2016-07-01     7 2016
## 188 2016-08-01     8 2016
## 189 2016-09-01     9 2016
## 190 2016-10-01    10 2016
## 191 2016-11-01    11 2016
## 192 2016-12-01    12 2016
## 194 2017-02-01     2 2017
## 195 2017-03-01     3 2017
## 196 2017-04-01     4 2017
## 198 2017-06-01     6 2017
## 199 2017-07-01     7 2017
## 200 2017-08-01     8 2017
## 201 2017-09-01     9 2017
## 202 2017-10-01    10 2017
## 204 2017-12-01    12 2017
## 207 2018-03-01     3 2018
## 208 2018-04-01     4 2018
## 209 2018-05-01     5 2018
## 210 2018-06-01     6 2018
## 211 2018-07-01     7 2018
## 212 2018-08-01     8 2018
## 213 2018-09-01     9 2018
## 214 2018-10-01    10 2018
## 215 2018-11-01    11 2018
## 221 2019-05-01     5 2019
## 222 2019-06-01     6 2019
## 223 2019-07-01     7 2019
## 224 2019-08-01     8 2019
## 225 2019-09-01     9 2019
```

```
dx$mean_evi <- NA
tdx <- rbind(td, dx) %>%
  arrange(date)
write.csv(tdx,"Data/tdx.csv")
tdx <- read.csv("Data/tdx.csv")
head(tdx)
```

```
##     X year month  mean_evi       date
## 1   1 2001     1 0.3103816 2001-01-01
## 2 216 2001     2        NA 2001-02-01
## 3   2 2001     3 0.6017811 2001-03-01
## 4   3 2001     4 0.5585050 2001-04-01
## 5 510 2001     5        NA 2001-05-01
## 6 610 2001     6        NA 2001-06-01
```

```
na.cnt <- length(tdx[is.na(tdx)])
evi.trend$na.cnt.2[i] <- na.cnt #add count of na values to dataframe
rm(td, dx) #remove data we're no longer using, this is a good rule of thumb, especially when working wi
tdx <- ts(data = tdx$mean_evi, start = c(2001, 1), end = c(2019, 11), frequency = 12) #convert data to
plot(tdx)
```

Galamsey_files/figure-latex/unnamed-chunk-13-1.pdf

```
library(imputeTS)
tdx <- if(na.cnt > 0){imputeTS::na_kalman(tdx, model = "auto.arima", smooth = T)} else {
    tdx
}
plot(tdx)
```

Galamsey_files/figure-latex/unnamed-chunk-14-1.pdf

```
tdx.dcp <- stl(tdx, s.window = 'periodic')
plot(tdx.dcp)
```

Galamsey_files/figure-latex/unnamed-chunk-15-1.pdf

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.1.3
```

```
Tt <- trendcycle(tdx.dcp)
St <- seasonal(tdx.dcp)
Rt <- remainder(tdx.dcp)
plot(Tt)
```

Galamsey_files/figure-latex/unnamed-chunk-16-1.pdf

```
plot(St)
```

Galamsey_files/figure-latex/unnamed-chunk-16-2.pdf

```
plot(Rt)
```

Galamsey_files/figure-latex/unnamed-chunk-16-3.pdf

```
library(tseries)
```

```
##
## Attaching package: 'tseries'
```

```
## The following object is masked from 'package:imputeTS':
##
##     na.remove
```

```
adf.test(Rt)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Rt
## Dickey-Fuller = -8.639, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(Tt)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Tt
## Dickey-Fuller = -3.4545, Lag order = 6, p-value = 0.04798
## alternative hypothesis: stationary
```

```
tdx <- data.frame(time = c(1:length(tdx)), trend = tdx - tdx.dcp$time.series[,1])
trend.summ <- summary(lm(formula = trend ~ time, data = tdx))
trend.summ
```

```
##
## Call:
## lm(formula = trend ~ time, data = tdx)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.159773 -0.020549  0.000499  0.016519  0.135949
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.798e-01  5.293e-03  90.635  < 2e-16 ***
## time        -1.118e-04  4.026e-05  -2.778  0.00593 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03974 on 225 degrees of freedom
## Multiple R-squared:  0.03317,    Adjusted R-squared:  0.02887
## F-statistic: 7.718 on 1 and 225 DF,  p-value: 0.005928
```

```
plot(tdx)
abline(a = trend.summ$coefficients[1,1], b = trend.summ$coefficients[2,1], col = 'blue')
```

Galamsey_files/figure-latex/unnamed-chunk-20-1.pdf

Finding