Community Contribution Tutorial

Annika Hsi (ah4287)

2024-11-05

0. Setting Up

```
# install packages
# install.packages("ncdf4")
# install.packages("em")
#install.packages("maps")
# install.packages("rnaturalearth")

# load packages
library(ncdf4)
library(raster)
```

```
## Loading required package: sp
```

```
library(em)
library(ggplot2)
library(maps)
library(rnaturalearth)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:raster':
##
##     intersect, select, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
```

1. Visualizing Chlorophyll-a Data

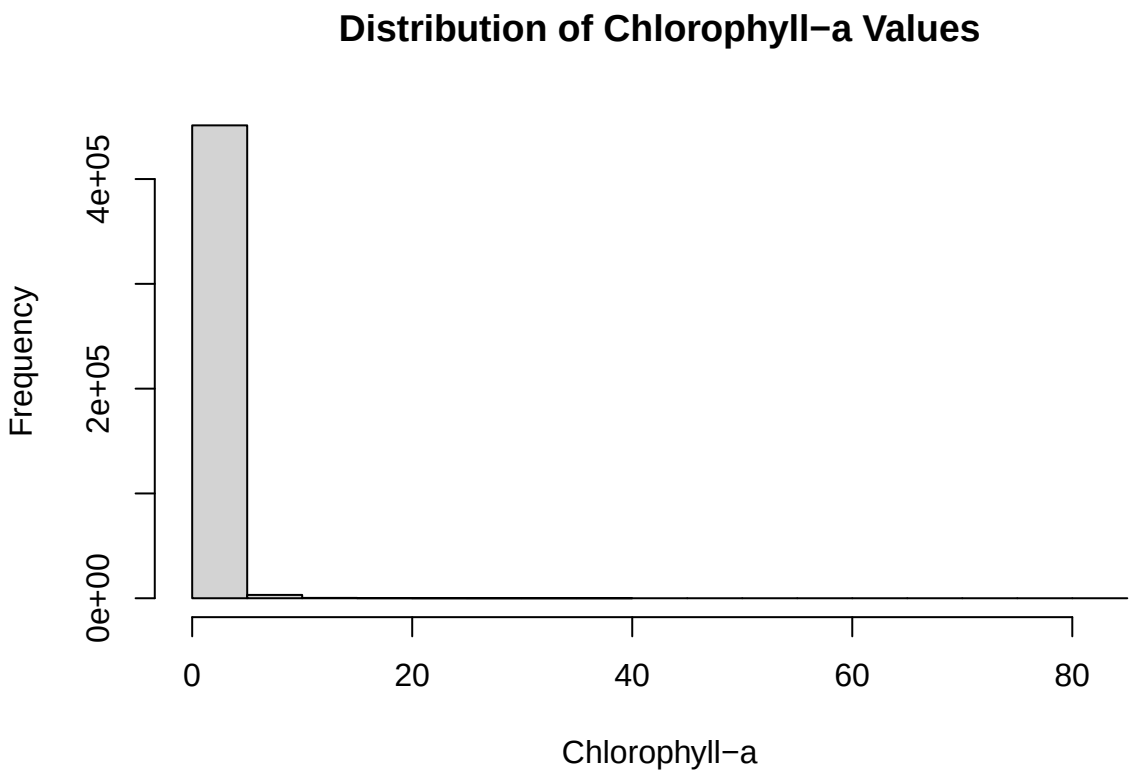
This section goes through extracting satellite data from a ncdf4 file and converting into a format that we are familiar with, and which is compatible with the `ggplot2` library. I acquired data from the NASA Earthdata Search site.

link to data: https://search.earthdata.nasa.gov/search

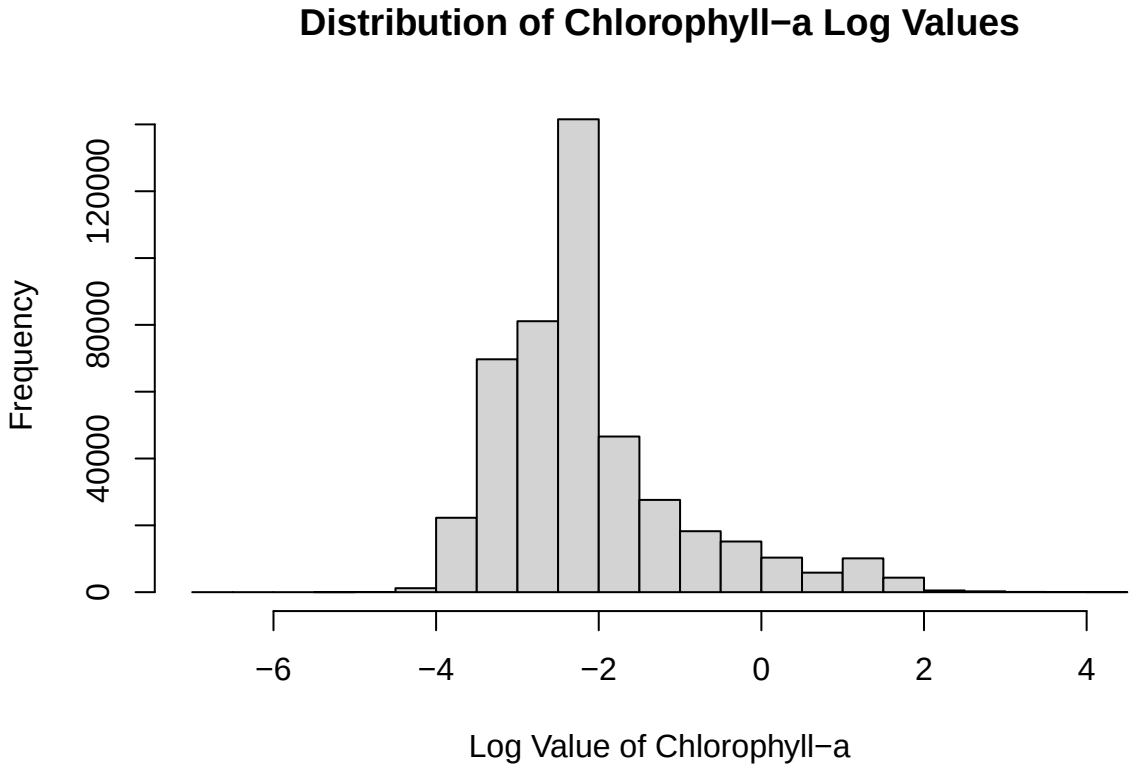
```r
# read in satellite data file
modis <- nc_open('AQUA_MODIS.20241102T181001.L2.OC.NRT.nc')

# extract variables of interest
chla <- ncvar_get(modis, 'geophysical_data/chlor_a')
lat <- ncvar_get(modis, 'navigation_data/latitude')
lon <- ncvar_get(modis, 'navigation_data/longitude')

# convert to df
chla_df <- as.data.frame(chla)
lat_df <- as.data.frame(lat)
lon_df <- as.data.frame(lon)

# close file
nc_close(modis)
```

```r
# flatten into 1 col
chla_flat <- data.frame(flatten(chla_df, by = 'col'), row.names = NULL)
lat_flat <- data.frame(flatten(lat_df, by = 'col'), row.names = NULL)
lon_flat <- data.frame(flatten(lon_df, by = 'col'), row.names = NULL)

# merge into a long data frame
df <- cbind(lat_flat, lon_flat, chla_flat)
colnames(df) <- c('latitude', 'longitude', 'chlor_a')

# get size of pixels
df_helper <- df |> mutate(lat_diff = latitude - lag(latitude),
                          lon_diff = longitude - lag(longitude))
pixel_height = mean(df_helper$lat_diff, na.rm = TRUE)
pixel_width = mean(df_helper$lon_diff, na.rm = TRUE)
```

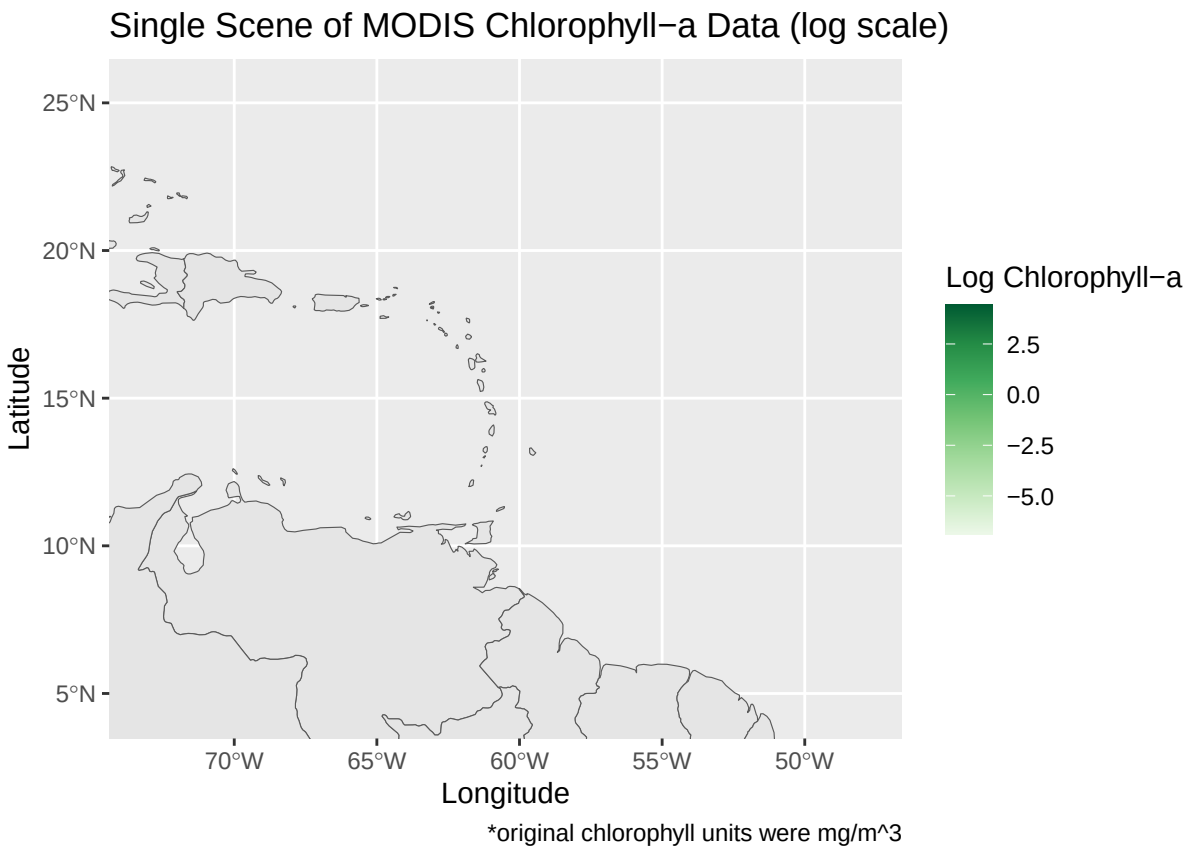
```r
# plot distribution of chlorophyll values
hist(df$chlor_a,
     xlab = 'Chlorophyll-a',
     main = 'Distribution of Chlorophyll-a Values' )
```




```
hist(log(df$chlor_a),
     xlab = 'Log Value of Chlorophyll-a',
     main = 'Distribution of Chlorophyll-a Log Values' )
```



```r
# get min and max coordinates to adjust zoom of base map
min_lat <- range(df$latitude, na.rm = TRUE)[1]
max_lat <- range(df$latitude, na.rm = TRUE)[2]
min_lon <- range(df$longitude, na.rm = TRUE)[1]
max_lon <- range(df$longitude, na.rm = TRUE)[2]

# get base map
world <- ne_countries(scale = "medium", returnclass = "sf")

# plot heat map
ggplot(data = world) +
  geom_sf() +
  coord_sf(xlim = c(min_lon, max_lon), ylim = c(min_lat, max_lat)) +
  geom_tile(data = df |> filter(!is.na(chlor_a)),
            aes(x = longitude, y = latitude, color = log(chlor_a)),
            width = pixel_width,
            height = pixel_height) +
  scale_color_distiller(palette = 'Greens', direction = 1, name = "Log Chlorophyll-a") +
  labs(title = 'Single Scene of MODIS Chlorophyll-a Data (log scale)',
       x = 'Longitude',
       y = 'Latitude',
       caption = '*original chlorophyll units were mg/m^3')
```



2. NDVI Timeseries

This section goes through working with a set of geoTIFFs and then creating a time series of their values. I accessed data from the APPEEARS site, which allows you to input a shapefile, dates, and the products of

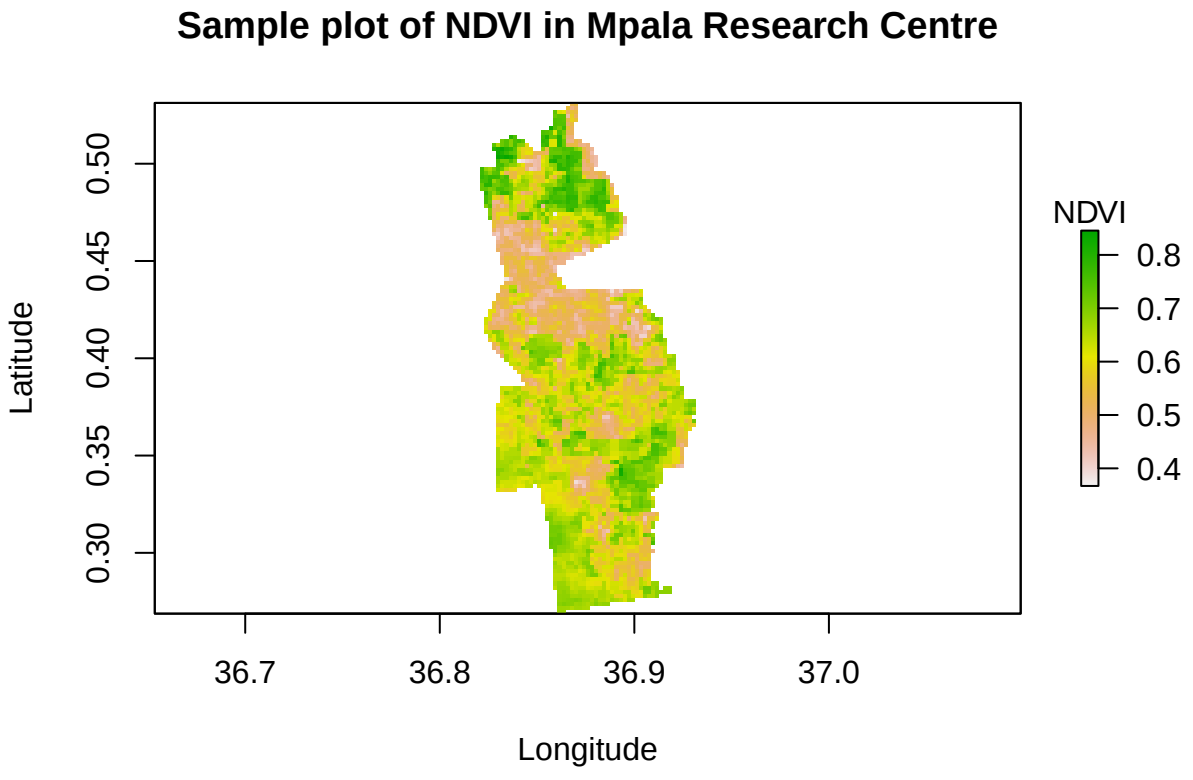
your choosing. I chose NDVI (normalized difference vegetation index), a measure of vegetation quality, from the satellite Terra MODIS as my product. The shapefile is for Mpala Research Centre in Kenya.

link to data: https://appeears.earthdatacloud.nasa.gov

```
# read in file that has information on satellite data (dates, file names, etc.)
info_df <- read_csv('MOD13Q1-061-Statistics.csv')
```

```
## Rows: 12 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): File Name, Dataset, aid, Range
## dbl  (11): Count, Minimum, Maximum, Mean, Standard Deviation, Variance, Uppe...
## date  (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

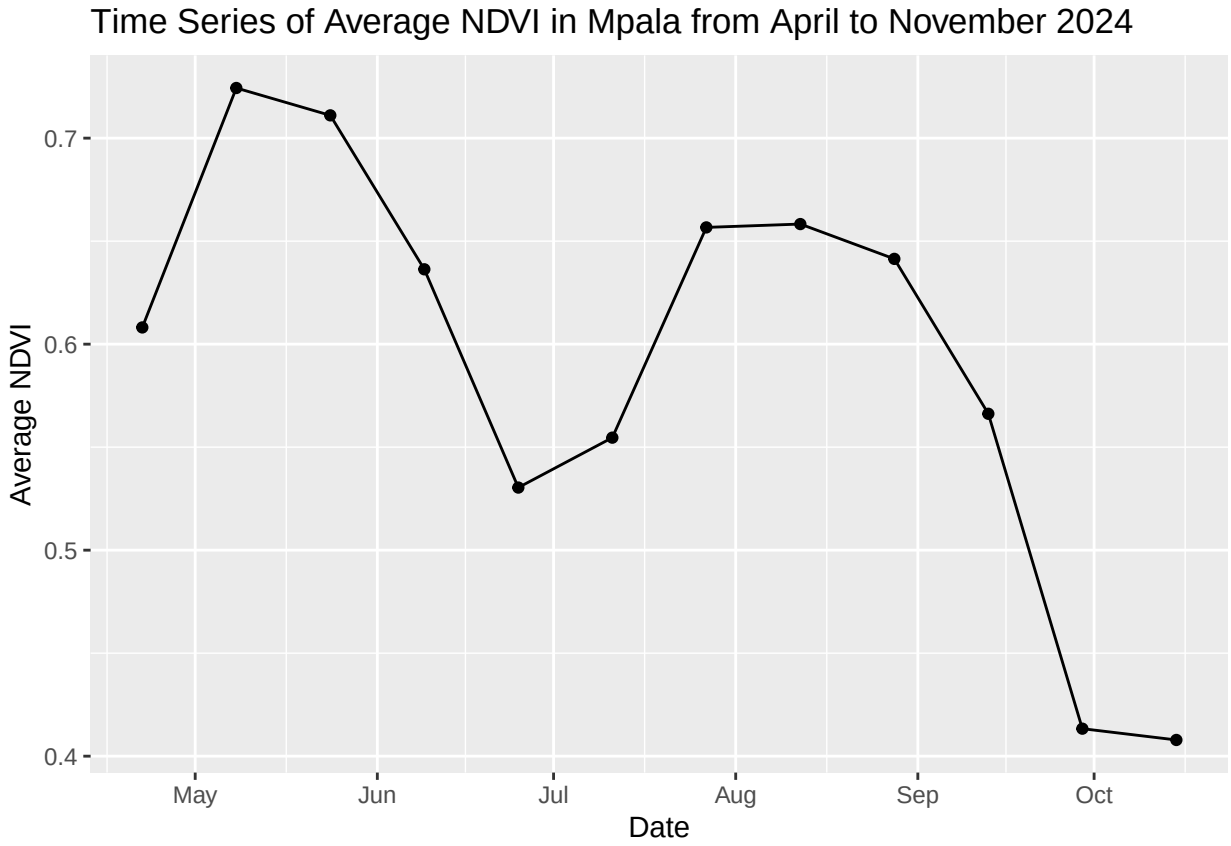
```
# sample raster of NDVI
plot(raster('NDVI/MOD13Q1.061__250m_16_days_NDVI_doy2024113_aid0001.tif'),
     xlab = 'Longitude',
     ylab = 'Latitude',
     main = 'Sample plot of NDVI in Mpala Research Centre',
     legend.args = list(text = 'NDVI'))
```




```
# get satellite files as data frames
filenames <- list.files('NDVI')
dates <- info_df$Date
dfnames <- c()
for (i in 1:length(filenames)) {
  assign(paste0("ndvi_", dates[i], '_df'),
         as.data.frame(raster(paste0('NDVI/', filenames[i])), xy = TRUE))
```

```
  dfnames[i] <- paste0("ndvi_", dates[i], '_df')
}
```

```
# append all data frames
ndvi_all <- data.frame(matrix(ncol = 4))
colnames(ndvi_all) <- c('Longitude', 'Latitude', 'NDVI', 'Date')
for (i in 1:length(dfnames)) {
  curr_d <- get(dfnames[i])

  # correct column names
  colnames(curr_d) <- c('Longitude', 'Latitude', 'NDVI')

  # add column for date
  curr_d <- curr_d |> mutate(Date = dates[i])

  # append to final df
  ndvi_all <- rbind(ndvi_all, curr_d)
}

# convert dates
ndvi_all$Date <- as.Date(ndvi_all$Date, origin = '1970-01-01')

# remove extra column at top
ndvi_all <- ndvi_all[2:nrow(ndvi_all), ]

# group by date and calculate average NDVI for the region
ndvi_summarized <- ndvi_all |>
  group_by(Date) |>
  summarise('Average NDVI' = mean(NDVI, na.rm = TRUE))

# plot time series
ggplot(ndvi_summarized, aes(x = Date, y = `Average NDVI`)) +
  geom_point() +
  geom_line() +
  labs(title = 'Time Series of Average NDVI in Mpala from April to November 2024')
```


Time Series of Average NDVI in Mpala from April to November 2024
Average NDVI
0.7
0.6
0.5
0.4
May
Jun
Jul
Aug
Sep
Oct
Date