APRIL 6, 2022

TINKERING LAB PROJECT

GOOGLE EARTH ENGINE

AYUSH TEHARPURIA
2020EEB1164

What is NDVI ?

NDVI is Normal Difference Vegetation Index.

The NDVI is a dimensionless index that describes the difference between visible and near-infrared reflectance of vegetation cover.

The electromagnetic spectrum is central to us understanding plant health. It is fundamental to how NDVI works and allows us to determine how healthy, or unhealthy; a plant is based on how it reflects energy and light.

Satellite sensors in space measure wavelengths of light absorbed and reflected by green plants. They are an excellent source of spectral signature data for NDVI analysis.

The NDVI index detects and quantifies the presence of live green vegetation using this reflected light in the visible and near-infrared bands.

Formula and Calculations

$$\text{NDVI}=\frac{NIR-RED}{NIR+RED}$$

NIR – reflection in the near-infrared spectrum
RED – reflection in the red range of the spectrum

Facts about NDVI

The value of the NDVI will always fall between -1 and +1.

Values between -1 and 0 indicate dead plants, or inorganic objects such as stones, roads, and houses. 1 being the healthiest and 0 being the least healthy. Chlorophyll (a health indicator) strongly absorbs visible light, and the cellular structure of the leaves strongly reflect near-infrared light. When the plant becomes dehydrated, sick, afflicted with disease, etc., the spongy layer deteriorates, and the plant absorbs more of the near-infrared light, rather than reflecting it. Thus, observing how NIR changes compared to red light provides an accurate indication of the presence of chlorophyll, which correlates with plant health.

NDVI algorithms are trained to work with satellite imagery to calculate NDVI and produce an easy-to-work-with georeferenced output as a result.As we now know, NDVI algorithms calculate the NDVI value for each pixel of your AOI, assigning each one a value in the range of -1 to 1.

Fig. 1 Various NDVI Numbers represent which vegetation

Plots and Observation of NDVI

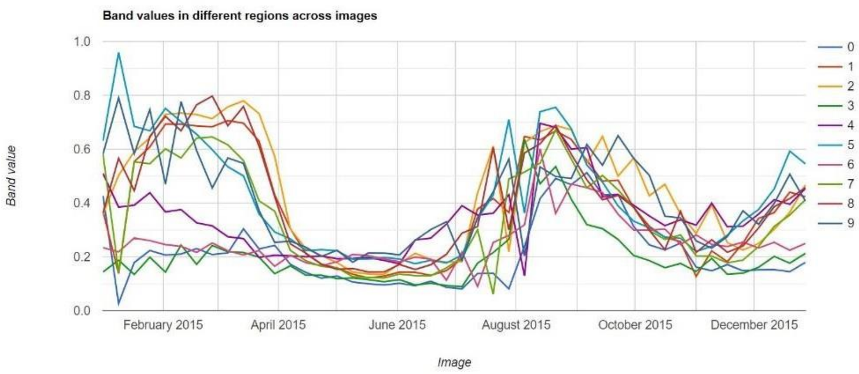

Fig 2. NDVI vs Time graph for Various Points

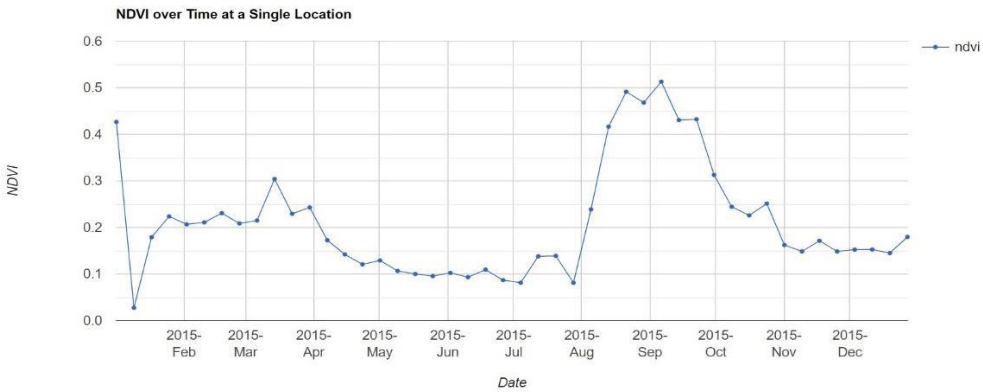

Fig 3. NDVI vs Time Graph for a single location

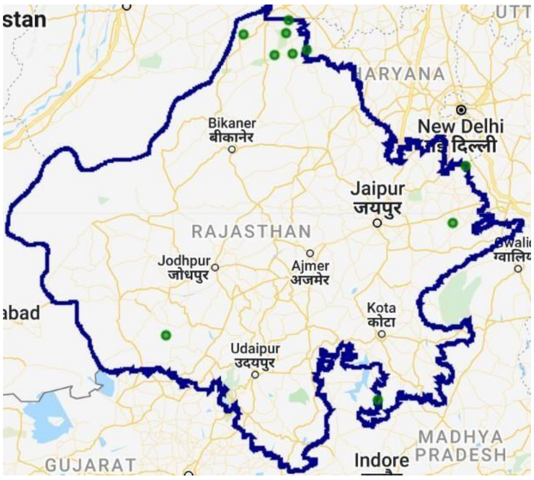

Fig. 4 Green Dots on the map

(Here Green dots are representing farms from which we are collecting the data from the Google Earth Engine.)

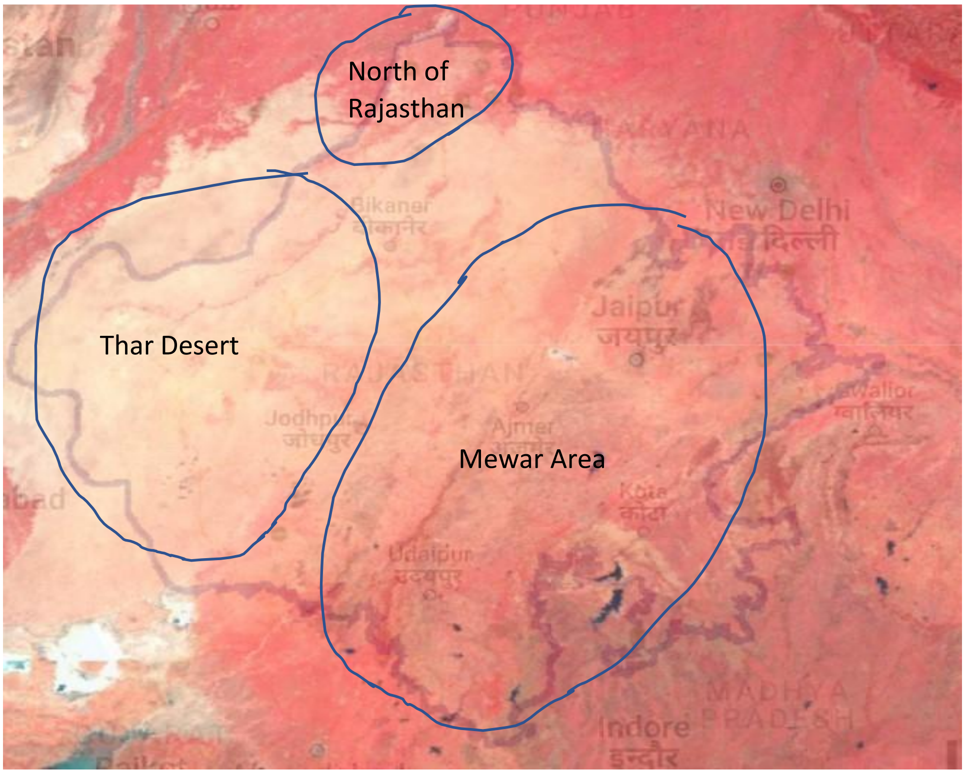

Fig 5. Various areas of Rajasthan

Analysis

Different Areas of Rajasthan

North Rajasthan: Comprises the area of Hanumangarh and Sriganganagar Region which is the crop hub of the state due to the Indira Gandhi Canal flowing through these districts. Rainfall is slightly better in the these regions as compared to the Thar regions but not good than the Mewar Region.

Thar Desert: Covers most of the Rajasthan is dry where there is not much rainfall and vegetation.

Mewar Region: Receives good Rainfall and is quite rich in vegetation because of the rainfall it receives in the Monsoon season.

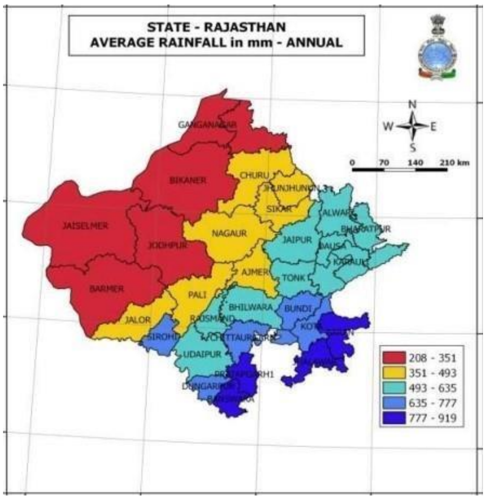

Fig 6. Avg Rainfall across Raj. Over the year

The above figures are showing the band values of different regions of the state of Rajasthan. We can see a general trend here since its Rajasthan is composed arid , semi arid and aravali mountains. Here it can be seen that the in arid and semi-arid regions which is in Thar region i.e (band 5,6 and 7) vegetation increases in the months of July and August i.e. Rainy Season.

There is sudden increase in NDVI in Monsoon and then it slowly decreases before increasing in winter. There are quite a few points in the north part of Rajasthan (Sriganganagar and Hanumangarh) where NDVI number comes from the color of farms and it decreases when the Harvesting Season of the crops arrive in the months of March-April and October-November .

We can also observe that at the places at which color concentration is more, Vegetation is abundant in that region.

Colorised Graph

Fig 7. Colorised Graph showing proportional vegetation in the state

Time Series Analysis of the of the year 2015 of the state of Rajasthan

Test Points	NDVI Index
1	-.27933
2	-.40865
3	-.44711
4	-.28487
5	-.39089
6	-.45778
7	-.34402
8	-.38817
9	-.42741
10	-.41707
Average	-0.38453

GEE Code Writing process:

- The necessary links are imported into code editor on the Google Earth Engine (GEE).
- Using Map.setCenter(), the coordinates of the selected location (in our case – the state of Maharashtra) were substituted as follows:
- Map.setCenter = (latitude, longitude, Zoom level)
- Landcover was selected since we want to find out NDVI (vegetation).
- Start date and End Date are also selected. (The year is 2015 since we have imported the 2015 Google Data of the states i.e. gaul ee.FeatureCollection("FAO/GAUL/2015/level1")
- Time Series Graphs were plotted for the selected no. of test points (regions).
- The spectral analysis Maps is also generated and printed using the suitable code.

Conclusion

- Hence, MODIS was used on the Google Search Engine (GEE) to compute and study one of the spectral indices on the state of Rajasthan.
- We analysed the State of Rajasthan with Time Series Analysis with the help of NDVI for the year of 2015 by plotting graphs successfully.
- Trends of Vegetation in the whole year of 2015 across the three different parts of Rajasthan studied successfully.
- The code used on the Google Earth Engine is attached below in the file and .CSV file is attached in the ZIP File.

Code

```
var modis = ee.ImageCollection("MODIS/MOD09A1"),

    gaul = ee.FeatureCollection("FAO/GAUL/2015/level1"),

    gfsad = ee.Image("USGS/GFSAD1000_V0");



Map.setCenter = (25.0238,74.2179,5);

//Latitude Longitude Zoom

// Select 'landcover' band with pixel values 1 which represent

//Rice and Wheat Rainfed crops

var wheatrice = gfsad.select('landcover').eq(1)

// Rajasthan

// We use the Global Administrative Unit Layers (GAUL) dataset to get the state boundary

var Rajasthan = gaul.filter(ee.Filter.eq('ADM1_NAME', 'Rajasthan'))

// wheatrice image contains 1 and 0 pixels. We want to generate points

// only in the pixels that are 1 (representing crop areas)

// selfMask() masks the pixels with 0 value.

var points = wheatrice.selfMask().stratifiedSample({numPoints:10, region:Rajasthan,
geometries: true} )

// We need a unique id for each point. We take the feature id and set it as

// a property so we can refer to each point easily

var points = points.map(function(feature) {

  return ee.Feature(feature.geometry(), {'id': feature.id()})
```

```
})
// Show the state polygon with a blue outline
var outline = ee.Image().byte().paint({
featureCollection: Rajasthan,
  color: 1,
  width: 3
});
Map.addLayer(outline, {palette: ['darkblue']}, 'AOI')
// Show the farm locations in green
Map.addLayer(points, {color: 'green'}, 'Farm Locations')

//define the time period
var startDate = '2015-01-01'
var endDate = '2015-12-31'

// bands
var modisBands =
['sur_refl_b03','sur_refl_b04','sur_refl_b01','sur_refl_b02','sur_refl_b06','sur_refl_b07'];
var lsBands = ['blue','green','red','nir','swir1','swir2'];

// helper function to extract the QA bits
function getQABits(image, start, end, newName) {
    // Compute the bits we need to extract.
    var pattern = 0;
    for (var i = start; i <= end; i++) {
       pattern += Math.pow(2, i);
    }
    // Return a single band image of the extracted QA bits, giving the band a new name.
    return image.select([0], [newName])
                  .bitwiseAnd(pattern)
                  .rightShift(start);
```

```
}

// A function to mask out cloudy pixels.
function maskQuality(image) {
  // Select the QA band.
  var QA = image.select('StateQA');
  // Get the internal_cloud_algorithm_flag bit.
  var internalQuality = getQABits(QA,8, 13, 'internal_quality_flag');
  // Return an image masking out cloudy areas.
  return image.updateMask(internalQuality.eq(0));
}

// create cloud free composite
var noCloud = modis.filterDate(startDate,endDate)
                             .map(maskQuality)
                             .select(modisBands,lsBands)
                             .filter(ee.Filter.bounds(points))




// vis parameters
var visParams = {bands:['nir','red','green'],min:0,max:3000,gamma:1.3};

// add the cloud free composite
Map.addLayer(noCloud.median(),visParams,'MODIS Composite');

// Adding a NDVI band
function addNDVI(noCloud) {
  var ndvi = noCloud.normalizedDifference(['sur_refl_b02', 'sur_refl_b01']).rename('ndvi')
  return noCloud.addBands([ndvi])
```

```
 }

var collection = modis.filterDate(startDate, endDate)
    .map(addNDVI)
    //.filter(ee.Filter.bounds(points))



// View the median composite
var vizParams = {bands: ['ndvi'], min: -1, max: 1}
Map.addLayer(collection.median(), vizParams, 'collection')

var testPoint = ee.Feature(points.first())
Map.centerObject(testPoint, 4)
var chart = ui.Chart.image.series({
    imageCollection: collection.select('ndvi'),
    region: testPoint.geometry()
    }).setOptions({
      interpolateNulls: true,
      lineWidth: 1,
      pointSize: 3,
      title: 'NDVI over Time at a Single Location',
      vAxis: {title: 'NDVI'},
      hAxis: {title: 'Date', format: 'YYYY-MMM', gridlines: {count: 12}}
    })
print(chart)

var chart = ui.Chart.image.seriesByRegion({
    imageCollection:  collection.select('ndvi'),
    regions: points,
    reducer: ee.Reducer.mean()
```

```
})
print(chart)
```