var countries = ee.FeatureCollection("FAO/GAUL/2015/level0");

var names = countries.aggregate\_array('ADM0\_NAME');

var keys = ee.Dictionary(names).keys();

print(keys);

var values = ee.Dictionary(names).values();

print(keys);

var values = ee.Dictionary(names).values();

print(values.sort());

Figure The output of the code above

preparing the data in sentinel 2

//here we have:

// B2: Blue (496.6 nm)

// B3: Green (560.0 nm)

//RED is: B4, 664.5 nm

//NIR is: B8, 835.1 nm

//Step 1: Access the boundary-defining geometry

var ext\_leb = ee.FeatureCollection("FAO/GAUL/2015/level0")

.filter(ee.Filter.eq('ADM0\_NAME', 'Lebanon'));

//filter for entry that equals 'Lebanon'

//Step 2: Access the Sentinel-2 data and filter it for all the images of the year 2022 and the month ‘May’ that lie within the geometries boundaries.

var sentinel = ee.ImageCollection('COPERNICUS/S2\_SR')

.filterBounds(ext\_leb)

.filterDate('2022-05-01', '2022-05-31')

.select('B1','B2','B3','B4','B5','B6','B7','B8','B8A','B9','B11','B12')

.filter(ee.Filter.lt('CLOUDY\_PIXEL\_PERCENTAGE', 10));

//keep only the suitable bands and filter for cloud coverage

Map.centerObject(sentinel,9)

//Step 3: reducing by Median to Create a single Image

var sentinel\_med = sentinel.median()

.clip(ext\_leb);

//Add your Image as a map layer

var visParams = {'min': 400,'max': [4000,3000,3000], 'bands':'B8,B4,B3'};

Map.addLayer(sentinel\_med, visParams, 'S2 Lebanon May 2022 Median');

//first way: Simple band operations

var nir = sentinel\_med.select('B8');

var red = sentinel\_med.select('B4');

var ndvi = nir.subtract(red).divide(nir.add(red)).rename('NDVI');

// Display the result.

var ndviParams = {min: -1, max: 1, palette: ['blue', 'white', 'green']};

Map.addLayer(ndvi, ndviParams, 'NDVI Lebanon May 2022 V1');

//second way: .expression function

var ndvi\_2 = sentinel\_med.expression(

'(NIR-RED)/(NIR+RED)', {

'NIR' : sentinel\_med.select('B8'),

'RED' : sentinel\_med.select('B4')

})

.rename('NDVI');

//Display the result

Map.addLayer(ndvi\_2, ndviParams , 'NDVI Lebanon May 2022 V2');

//third way: .normalizedDifference(NIR, RED) function

var ndvi\_3 = sentinel\_med.normalizedDifference(['B8', 'B4'])

.rename('NDVI');

//print(ndvi\_3, 'NDVI Lebanon May 2022 V3');

//this color combination is taken from the Examples script Image -> Normalized Difference:

var palette = ['FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718',

'74A901', '66A000', '529400', '3E8601', '207401', '056201',

'004C00', '023B01', '012E01', '011D01', '011301'];

// Display the input image and the NDVI derived from it.

Map.addLayer(ndvi\_3, {min: 0, max: 1, palette: palette}, 'NDVI Lebanon May 2022 V3')

**Modis**

//Access boundary data

var ext\_leb = ee.FeatureCollection("FAO/GAUL/2015/level0")

.filter(ee.Filter.eq('ADM0\_NAME', 'Lebanon'));

//filter for entry that equals the UN country name 'Lebanon'

// Center the map on the boundary

Map.centerObject(ext\_leb, 9);

//Access a MODIS ImageCollection.

var modis = ee.ImageCollection('MODIS/006/MOD09GA')

.filterBounds(ext\_leb)

.filterDate('2022-05-01', '2022-05-31')

.select('sur\_refl\_b01', 'sur\_refl\_b02', 'sur\_refl\_b03', 'sur\_refl\_b04', 'sur\_refl\_b05', 'sur\_refl\_b06', 'sur\_refl\_b07')

//reduce by median the MODIS ImageCollection to make it a single Image

var modis\_median = modis.median()

.clip(ext\_leb);

var ndvi = modis\_median.normalizedDifference(['sur\_refl\_b02', 'sur\_refl\_b01']);

var ndviParams = {min: -1, max: 1, palette: ['blue', 'white', 'green']};

// Display the input image and the NDVI derived from it.

Map.addLayer(modis\_median.select(['sur\_refl\_b01', 'sur\_refl\_b04', 'sur\_refl\_b03']),

{gain: [0.1, 0.1, 0.1]}, 'MODIS May 2022 Lebanon Bands 1/4/3');

Map.addLayer(ndvi, ndviParams, 'NDVI MODIS May 2022 Lebanon');

// Directly access the NDVI through specific MODIS collections

var modis\_2 = ee.ImageCollection('MODIS/006/MOD13Q1')

.filterBounds(ext\_leb)

.filterDate('2022-05-01', '2022-05-31');

var scale\_factor = function(image) {

return image.expression('float(b("NDVI")\*0.0001)')

//the value 0.0001 will be multiplied to all pixels of our ImageCollection

};

// Select the NDVI-Bands

var ndvi\_2 = modis\_2.select('NDVI');

// Apply the scaling factor

var ndvi\_2\_scaled = ndvi\_2.map(scale\_factor);

//Reduce the ImageCollection to an Image using the mean values

var ndvi\_2\_scaled\_clipped = ndvi\_2\_scaled.mean().clip(ext\_leb);

//visualize the data

var ndviParams = {min: -1, max: 1, palette: ['blue', 'white', 'green']};

Map.addLayer(ndvi\_2\_scaled\_clipped, ndviParams, 'NDVI MODIS May 2022 Lebanon Alternative');

**Explanation**

**Sentinel 2**

The code starts by accessing the boundary-defining geometry. It then filters for all the entries that have a name equal to Lebanon, then it filters for all the images of the year 2022 and the month ‘May’ that lie within the geometry’s boundaries. Next, it filters for only those bands which are suitable to be used in a map layer (B8, B4, B3) to analyse cloud coverage and to filter it by the function “. filter(ee.Filter.lt('CLOUDY\_PIXEL\_PERCENTAGE', 10))”. Then it creates a single image from these filtered Sentinel-2 data using the median() function on the sentinel variable and put the output in the sentinel\_med variable. Finally, we add it as a map layer called “S2 Lebanon May 2022 Median”.

So after we explained the “preparing the data” part of the code, now we will explain the 3 ways of calculating the NDVI index, that is first by using simple band operations, so at the beginning, we make two variables nir(B8) and red(B4), and we will calculate the NDVI manually by subtracting B4 from B8 and dividing that result by the sum of B8 and B4 because as we know the formula of NDVI is NDVI = (NIR – Red) / (NIR + Red), and we will rename the result of this equation as NDVI, and finally to display the results, we add it as a map layer called “NDVI Lebanon May 2022 V1”. The second way to calculate the NDVI index is by using the .expression function, so by using the same formula we can compute the result and rename it NDVI, and to display the results we add it as a map layer called “NDVI Lebanon May 2022 V2”. The third way to calculate the NDVI index is by using the normalizedDifference function, so by using this function on B8 and B4 it means that we are calculating (B8-B4)/(B8+B4), and then we rename it as NDVI, and now we create a palette of colours, then the code goes through each colour in the palette that was created earlier and calculates their values using hex strings (e.g., FFFFFF = 255). It then adds these values together to create a new array with all of the colours in order from lightest to darkest (e.g., ['FCD163', '99B718']). So, the code is a function that takes an array of colours and returns the median value for each colour and if there are no values in the input array, it will return a number between 0 and 1, and finally, display the results we add it as a map layer called “NDVI Lebanon May 2022 V3”.

Now that we have calculated and displayed the NDVI in three different ways, we can use the Inspector and click on various places in the map panel to see that the pixel values are always identical. In many cases, the most suitable way to calculate the NDVI index is by using the normalizedDifference function as it’s easy and quick to execute.

**MODIS**

The code starts by accessing the boundary of the entry equals the value “Lebanon”, we centred the map on the boundary, then it filters for all the images of the year 2022 and the month ‘May’ that lie within the geometry’s boundaries. Next, it filters for only those bands which are suitable to be used in a map layer (e.g.: 'sur\_refl\_b01'), then we reduce the image collections by median to make it a single image, we used the function “.median()”. And as we do it before in the third way of calculating the NDVI index by using sentinel-2, so we use the “.normalizedDifference” function on these two bands ('sur\_refl\_b02', 'sur\_refl\_b01'), and we make a palette of the three colors blue, white and green, and finally to display the results we add the two as a map layer called: for the median is 'MODIS May 2022 Lebanon Bands 1/4/3' and for NDVI is 'NDVI MODIS May 2022 Lebanon'.

For the second way to calculate the NDVI index by using MODIS, we will use the MOD13Q1.006 Terra Vegetation Indices, so now we can directly access the NDVI through specific MODIS collections, then Access the imagecollections of MODIS for the specific time we want “May 2022” by the “.filterDate” function. Now we want to make a scale to the NDVI, so by the function named scale we apply a scaling factor of 0.0001 for the NDVI-bands to be displayed correctly, and this value will be multiplied by all pixels of our ImageCollections. Then we select the NDVI bands and apply the scaling factor (the scale method) and after that, by using the mean function we reduce the imagecollections to an image. finally, to display the results we add it as a map layer called 'NDVI MODIS May 2022 Lebanon 2').

Step one: is to determine the objective of the project, and here is to calculate the NDVI index for the country “Lebanon” at a specific time “May 2022” by using Sentinel-2 in the Google Earth Engine.

Step two: is gathering the necessary data we need to use, here in our case we gather all the data for the country “Lebanon” from the FAO dataset and this dataset provides free access to food and agriculture data for over 245 countries and Lebanon is one of them like we see in the above code.

Step three: is data preparation, we prepare the data to calculate NDVI by accessing the Sentinel-2 data and filtering it for all the images of the year 2022 and the month ‘May’ that lie within the geometry’s boundaries. Also, we keep only the suitable bands and filters for cloud coverage.

Step four: data cleaning, we clean the data by using the median filter so we can then reduce the noise in an image so it preserves useful detail in the image.

Step five: is analysing the data, so now we calculate the NDVI index in three ways, are simple band operations, expression function and normalized difference function. So, by these three ways we calculate the formula of the NDVI index, which is NDVI = (NIR-RED) / (NIR+RED).

Step six: is sharing our results, so we visualize all the calculations we made in the previous steps in Google Earth Engine as a map layer, and you can see the results of the above code by clicking the below link, and this link will lead you to my Google Earth Engine APP:

<https://antonysakr.users.earthengine.app/view/calculating-ndvi-by-using-sentinel-2>

Step one: is to determine the objective of this part, and here is to calculate the NDVI index for the country “Lebanon” at a specific time “May 2022” by using MODIS in the Google Earth Engine.

Step two: is gathering the necessary data, as we did in Sentinel-2, in our case we gather all the data for the country “Lebanon” from the FAO dataset.

Step three: is data preparation, we prepare the data to calculate NDVI by accessing the MODIS data and filtering it for all the images of the year 2022 and the month ‘May’ that lie within the geometry’s boundaries (same country and time as we did before). Also, we keep only the suitable bands.

Step four: data cleaning, we clean the data by using the median filter so we can then reduce the noise in an image, so it preserves useful detail in the image.

Step five: is analysing the data, so now we calculate the NDVI index in two ways, are: using band maths (identical to Sentinel-2) and using the MODIS dataset that already includes NDVI bands. So, by these two ways we calculate the formula of the NDVI index, which is NDVI = (NIR-RED) / (NIR+RED).

Step six: it is the final step, and it is about sharing our results, so we visualize all the calculations we made in the previous steps in Google Earth Engine as a map layer, and you can see the results of the above code by clicking the below link, and this link will lead you to my Google Earth Engine APP:

<https://antonysakr.users.earthengine.app/view/calculating-ndvi-by-using-modis>