# Google Earth Engine code documentation

## Extraction of NDVI and EVI from MODIS imagery; Extraction-based method

```
// TO DO: Make sure to load in imports of data product, in this case MOD13Q1.006 Terra Vegetation Indices 16-Day Global 250m

// TO DO: Load coordinates in and center map to the area
var site = ee.Geometry.Point([145.7913456, -17.3530442]); // long, lat
Map.addLayer(site,{},"site");
Map.centerObject(site, 9); // 9 is the scale

/// Add ANNUAL PRE case study information ///
// Note: MODIS data product only available for post-2000 years
// TO DO: This is to narrow temporal resolution, so add in start and end dates
var sta = (VIs.filterBounds(site)
   .filterDate('2005-03-19', '2006-03-19') // mosaic helps convert collection to image
);
print(sta, "sta");  // print is useful incase error is present

/// Add POST case study information ///
// TO DO: add post start and end dates
var stb = (VIs.filterBounds(site)
   .filterDate('2006-03-21', '2006-04-06')
);
print(stb, "stb");

/// Add RECOVERY case study information ///
// TO DO: add recovery start and end dates
var stc = (VIs.filterBounds(site)
   .filterDate('2006-03-21', '2008-03-21')
);
print(stc, "stc");

/// Add SUB-ANNUAL PRE case study information ///
// TO DO: add sub-annual pre start and end dates
var std = (VIs.filterBounds(site)
   .filterDate('1998-03-19', '2006-03-19')
);
print(std, "std");
```

```
// annual pre EVI
// Use reduce to get the mean and SD of the image
var EVI1 = sta.select("EVI");
print(EVI1, "EVI1");
Map.addLayer(EVI1,{},"EVI1");  // add the correct bands to map layer to check area
var statm1e = EVI1.reduce(ee.Reducer.mean());
var statsd1e = EVI1.reduce(ee.Reducer.stdDev());

// post EVI
// Use reduce to get the mean and SD of the image
var EVI2 = stb.select("EVI");
print(EVI2, "EVI2");
Map.addLayer(EVI2,{},"EVI2");
var statm2e = EVI2.reduce(ee.Reducer.mean());
var statsd2e = EVI2.reduce(ee.Reducer.stdDev());

/// make recovery EVI timeseries
// ensure to select the correct band, in this case EVI, for the time series
var EVIreco = ui.Chart.image.series({
  imageCollection: stc.select('EVI'),
  region: site,
  reducer: ee.Reducer.first(),
  scale: 30
}).setOptions({title: 'EVI 2y recovery'});  // good chart title helps with extracting csv later

/// make sub-annual pre EVI timeseries
// ensure to select the correct band, in this case EVI, for the time series
var EVIsapre = ui.Chart.image.series({
  imageCollection: std.select('EVI'),
  region: site,
  reducer: ee.Reducer.first(),
  scale: 30
}).setOptions({title: 'sub-annual EVI'});  // good chart title helps with extracting csv later

// annual pre NDVI
// Use reduce to get the mean and SD of the image
var NDVI1 = sta.select("NDVI");
print(NDVI1, "NDVI1");
Map.addLayer(NDVI1,{},"NDVI1");
var statm1n = NDVI1.reduce(ee.Reducer.mean());
var statsd1n = NDVI1.reduce(ee.Reducer.stdDev());
```

```
// post NDVI
// Use reduce to get the mean and SD of the image
var NDVI2 = stb.select("NDVI");
print(NDVI2, "NDVI2");
Map.addLayer(NDVI2,{},"NDVI2");
var statm2n = NDVI2.reduce(ee.Reducer.mean());
var statsd2n = NDVI2.reduce(ee.Reducer.stdDev());

/// make recovery NDVI timeseries
// ensure to select the correct band, in this case NDVI, for the time series
var NDVIreco = ui.Chart.image.series({
  imageCollection: stc.select('NDVI'),
  region: site,
  reducer: ee.Reducer.first(),
  scale: 30
}).setOptions({title: 'NDVI 2y recovery'});  // good chart title helps with extracting
csv later

/// make sub-annual pre NDVI timeseries
// ensure to select the correct band, in this case NDVI, for the time series
var NDVIsapre = ui.Chart.image.series({
  imageCollection: std.select('NDVI'),
  region: site,
  reducer: ee.Reducer.first(),
  scale: 30
}).setOptions({title: 'sub-annual NDVI'});  // good chart title helps with extracting
csv later

// annual pre EVI
// outputs the annual pre mean and SD for EVI
var statsm1e = statm1e.reduceRegion(ee.Reducer.mean(),site,30);
print(statsm1e, "EVI annual pre m");  // name for ease of extraction after running
var statssd1e = statsd1e.reduceRegion(ee.Reducer.mean(),site,30);
print(statssd1e, "EVI annual pre sd");  // name for ease of extraction after running

//post EVI
var statsm2e = statm2e.reduceRegion(ee.Reducer.mean(),site,30);
print(statsm2e, "EVI post");  // name for ease of extraction after running
var statssd2e = statsd2e.reduceRegion(ee.Reducer.mean(),site,30);
print(statssd2e, "EVI post sd");  // name for ease of extraction after running

//recovery EVI
```

```
// outputs the recovery time series for EVI
print(EVIreco, "EVI recovery");  // name for ease of extraction after running

// sub-annual pre EVI
print(EVIsapre, "EVI sub-annual pre");  // name for ease of extraction after running

// annual pre NDVI
var statsm1n = statm1n.reduceRegion(ee.Reducer.mean(),site,30);
print(statsm1n, "NDVI annual pre m");  // name for ease of extraction after running
var statssd1n = statsd1n.reduceRegion(ee.Reducer.mean(),site,30);
print(statssd1n, "NDVI annual pre sd");  // name for ease of extraction after running

//post NDVI
var statsm2n = statm2n.reduceRegion(ee.Reducer.mean(),site,30);
print(statsm2n, "NDVI post");  // name for ease of extraction after running
var statssd2n = statsd2n.reduceRegion(ee.Reducer.mean(),site,30);
print(statssd2n, "NDVI post sd");  // name for ease of extraction after running

//recovery NDVI
print(NDVIreco, "NDVI recovery");  // name for ease of extraction after running

// sub-annual pre NDVI
print(NDVIsapre, "NDVI sub-annual pre");  // name for ease of extraction after running
```

## Extraction of EVI from Landsat 4, 5, and 8 imagery; Calculation-based method

```
// TO DO: Make sure to load in imports of data product, in this case USGS Landsat 5 TM Collection 1 Tier 1 TOA Reflectance, USGS Landsat 8 Collection 1 Tier 1 TOA Reflectance, and USGS Landsat 4 TM Collection 1 Tier 1 TOA Reflectance

/// Cloud Masking; created by Yanlei Feng ///
// The Landsat needs cloud masking, while the MODIS does not
// Get BQA band bits
var getQABits = function(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);
};
var cloudMaskL4578 = function(image) {
  var qa = image.select("BQA");
  // If the cloud bit (5) is set and the cloud confidence (7) is high
  // or the cloud shadow bit is set (3), then it's a bad pixel.
  var internalCloud = getQABits(qa, 4,4, 'L5TOA_Cloud');
  var internalCloudshade = getQABits(qa, 5,6, 'L5TOA_cloudshade');
  var internalshade = getQABits(qa, 7,8, 'L5TOA_shade');
  var m =
(internalCloud.eq(1).add(internalCloudshade.gte(2)).add(internalshade.gte(2))).no
t();
  // Return an image masking out cloudy areas.
  return image.mask(m);
};

/// Add ANNUAL PRE case study information ///
// Note: use Landsat 8 if between 2013 and 2021 and 5 or 4 if between 1984 and
2012
// TO DO: Narrow spatial resolution with coordinates
var site = ee.Geometry.Point([-65.744506, 18.315856]); // long, lat
Map.addLayer(site,{},"site");

// TO DO: create site1, a region of 3 X 3 tiles with the area from coordinates in
center; will end up with 9 tiles selected; to do this utilize the geometry feature and
rename the selection site1

// TO DO: Chose Landsat # and narrow temporal resolution with annual pre start
and end dates
var sta1 = (l8.filterBounds(site)
    .filterDate('2016-09-19', '2017-09-19') // mosaic helps convert collection to
image
);

// this code applies the cloud masking to selected location and dates
var sta2 = sta1.map(cloudMaskL4578);
Map.addLayer(sta2, {}, "sta2");
print(sta2,"sta2");
Map.centerObject(site, 9); // 9 is the scale
```

```
// Calculate EVI //
// Function to compute the EVI using an expression
var addEVI = function(image) {
  var evi = image.expression(
      '2.5 * ((NIR - RED) / (NIR + 6 * RED - 7.5 * BLUE + 1))', {
        'NIR': image.select('B5'), // L5: 4  L8: 5
        'RED': image.select('B4'), // L5: 3  L8: 4
        'BLUE': image.select('B2') // L5: 1  L8: 2
  }).rename('evi');
  return image.addBands(evi)};  // name of band created is evi

// applying the function, addEVI, written above to images
var withEVI1 = sta2.map(addEVI);
var EVI1 = withEVI1.select("evi"); // every image now only has one band which is
EVI
print(EVI1, "EVI1");

/// Add POST case study information ///
// TO DO: Narrow temporal resolution with post start and end dates
var stb1 = (l8.filterBounds(site)
    .filterDate('2017-09-21', '2017-10-07')
);

// this code applies the cloud masking to selected location and dates
var stb2 = stb1.map(cloudMaskL4578);
Map.addLayer(stb2, {}, "stb2");
print(stb2,"stb2");
Map.centerObject(site, 9); // 9 is the scale

/// Calculates EVI ///
var withEVI2 = stb2.map(addEVI);
var EVI2 = withEVI2.select("evi");
print(EVI2, "EVI2");

/// Add RECOVERY case study information ///
// TO DO: Narrow temporal resolution with recovery start and end dates
var stc1 = (l8.filterBounds(site)
    .filterDate('2017-09-21', '2019-09-21')
);

// this code applies the cloud masking to selected location and dates
var stc2 = stc1.map(cloudMaskL4578);
```

```
Map.addLayer(stc2, {}, "stc2");
print(stc2,"stc2");
Map.centerObject(site, 9); // 9 is the scale

/// Calculates EVI ///
var withEVI3 = stc2.map(addEVI);
var EVI3 = withEVI3.select("evi");
print(EVI3, "EVI3");

/// Add SUB-ANNUAL PRE case study information ///
// TO DO: Narrow temporal resolution with sub-annual pre start and end dates
var std1 = (l8.filterBounds(site)
    .filterDate('2016-09-19', '2017-09-19')
);

// this code applies the cloud masking to selected location and dates
var std2 = std1.map(cloudMaskL4578);
Map.addLayer(std2, {}, "std2");
print(std2,"std2");
Map.centerObject(site, 9);

/// Calculates EVI ///
var withEVI4 = std2.map(addEVI);
var EVI4 = withEVI4.select("evi");
print(EVI4, "EVI4");

/// print annual pre EVI mean
// selects evi band and calculated mean
var EVI1 = withEVI1.select("evi");
var EVI1m = EVI1.mean(); // average EVI over entire year
print(EVI1m, "EVI1m");

/// print post EVI mean
var EVI2 = withEVI2.select("evi");
var EVI2m = EVI2.mean();
print(EVI2m, "EVI2m");

/// make recovery EVI time series
// creates time series with the selected band over site 1
var chart1 = ui.Chart.image.series({
  imageCollection: withEVI3.select('evi'),  // select the evi band
  region: site1,
  reducer: ee.Reducer.first(),
  scale: 30
}).setOptions({title: 'EVI over time'});  // title helps when extracting csv in future

Map.addLayer(EVI1,{},"evi");

/// make sub-annual pre EVI timeseries
var chart2 = ui.Chart.image.series({
  imageCollection: withEVI4.select('evi'),
  region: site1,
  reducer: ee.Reducer.first(),
  scale: 30
}).setOptions({title: 'EVI over time'});  // title helps when extracting csv in future

// annual pre
// calculates mean using the reducer function, within site 1
var stats1 = EVI1m.reduceRegion(ee.Reducer.mean(),site1,30);
print(stats1, "annual pre m");  // name for ease of extraction after running

//post
var stats2 = EVI2m.reduceRegion(ee.Reducer.mean(),site1,30);
print(stats2, "post m");  // name for ease of extraction after running

//recovery
// printing time series chart
print(chart1, "recovery");  // name for ease of extraction after running

// sub-annual pre
print(chart2, "sub-annual pre");  // name for ease of extraction after running
```