
Google Earth Engine: a powerful tool to study heatwaves & droughts

Sofia Ermida

GEE Platform

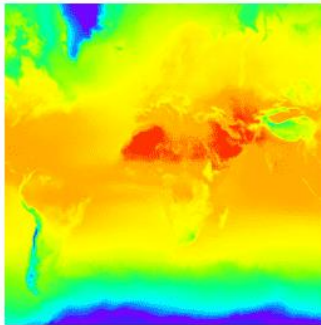
- Multi-petabyte catalogue of satellite imagery and geospatial datasets
- Data is available instantly and updated daily
- Google's cloud infrastructure
- User developed datasets and tools
- Python and JavaScript API

<https://earthengine.google.com/>

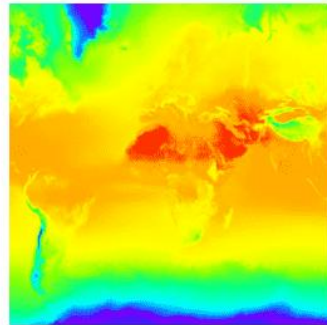


GEE Datasets

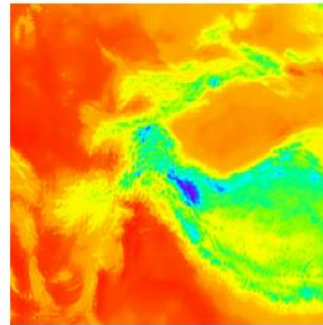
ERA5 Daily Aggregates - Latest Climate Reanalysis Produced by ECMWF / Copernicus Climate



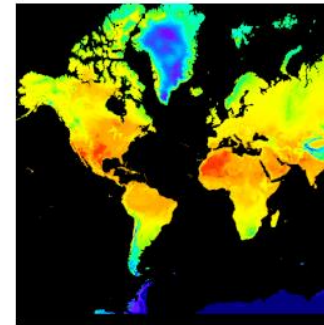
ERA5 Monthly Aggregates - Latest Climate Reanalysis Produced by ECMWF / Copernicus Climate



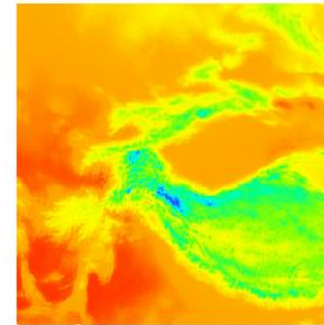
ERA5-Land Daily Aggregated - ECMWF Climate Reanalysis



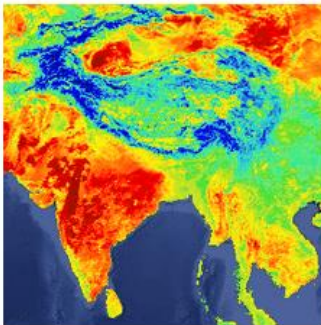
ERA5-Land Hourly - ECMWF Climate Reanalysis



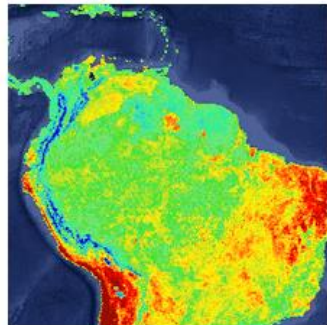
ERA5-Land Monthly Aggregated - ECMWF Climate Reanalysis



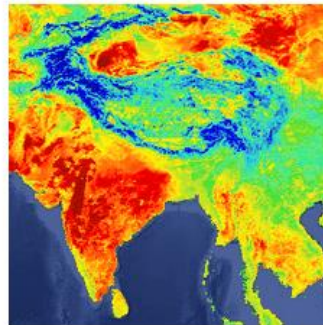
MOD11A1.061 Terra Land Surface Temperature and Emissivity Daily Global 1km



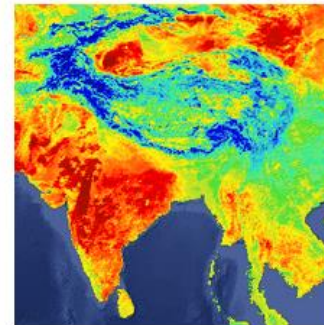
MOD11A2.061 Terra Land Surface Temperature and Emissivity 8-Day Global 1km



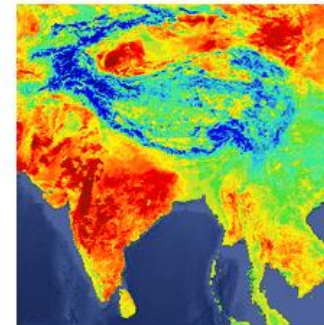
MOD21A1D.061 Terra Land Surface Temperature and 3-Band Emissivity Daily Global 1km



MOD21A1N.061 Terra Land Surface Temperature and 3-Band Emissivity Daily Global 1km

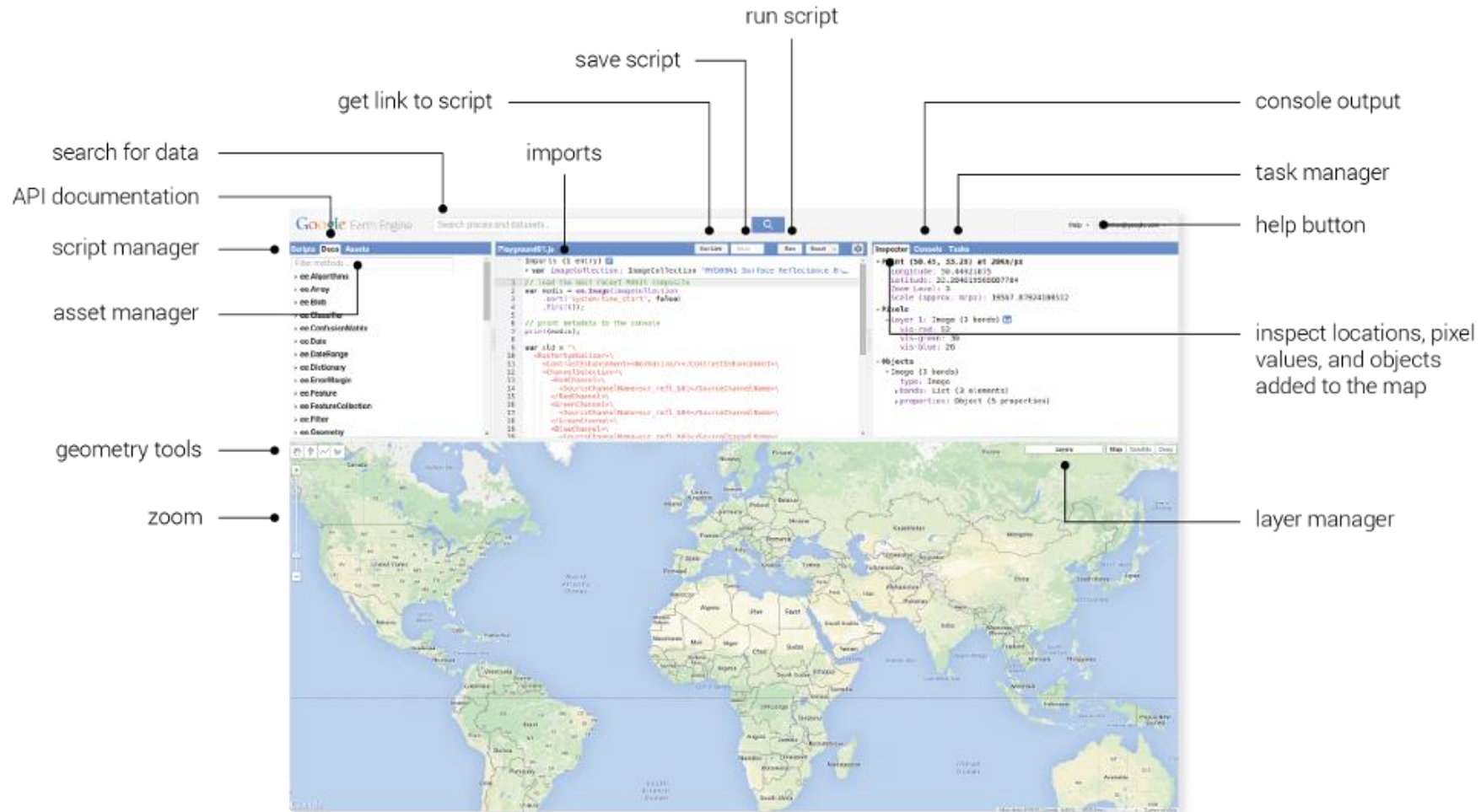


MOD21C1.061 Terra Land Surface Temperature and 3-Band Emissivity Daily L3 Global 0.05



GEE Code Editor

<https://code.earthengine.google.com/>



GEE Objects



Image

The fundamental raster data type in Earth Engine.



ImageCollection

A set of images.



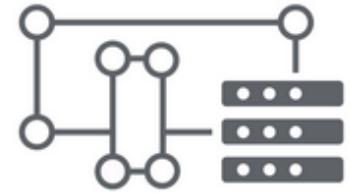
Geometry

The fundamental vector data type in Earth Engine.



Feature

A geometry with attributes.



FeatureCollection

A set of features.

Important aspects of coding

- Always use GEE objects: client objects may yield unexpected behavior

For operations with numbers or strings
use the GEE objects:

```
var clientNumber = 29;  
var serverNumber = ee.Number(29);
```

```
var clientString = 'I am a String';  
var serverString =  
  ee.String('I am a String');
```

Use the **IF** algorithm instead of the
client *if*:

```
ee.Algorithms.If(  
  ee.String('Tree').compareTo('Tree'),  
  'a tree', 'not a tree');  
ee.Algorithms.If(  
  ee.String('NotTree').compareTo('Tree'),  
  'a tree', 'not a tree');
```

Important aspects of coding

Mathematical operations:

```
// Load Landsat 5 image
var image = ee.Image('LANDSAT/LT05/C02/T1_TOA/LT05_044034_19900604');

// Compute Normalized Difference Vegetation Index
// NDVI = (NIR - RED) / (NIR + RED)
var ndvi = image.select('B5').subtract(image.select('B4'))
    .divide(image.select('B5').add(image.select('B4')));

// Compute the NDVI using an expression.
var ndvi_exp = image.expression(
    '(NIR - RED) / (NIR + RED)', {
        'NIR': image.select('B5'),
        'RED': image.select('B4')
    });
```

Important aspects of coding

Mapping (instead of a for-loop):

```
// This function sums one to the value
var addOne = function(n) {
  return ee.Number(n).add(1);
};

var years = ee.List.sequence(2000,2020);

var next_year = years.map(addOne);
```

```
// This function gets NDVI from Landsat 8 imagery.
var addNDVI = function(image) {
  return image.addBands(
    image.normalizedDifference(['SR_B5', 'SR_B4']));
};

// Load the Landsat 8 surface data, filter by location
and date.
var collection =
  ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
    .filterBounds(ee.Geometry.Point(-122.262, 37.8719))
    .filterDate('2014-06-01', '2014-10-01');

// Map the function over the collection.
var ndviCollection = collection.map(addNDVI);
```


GEE Algorithms

ee.Algorithms

ee.Algorithms.FMask

ee.Algorithms.GeometryConstructors

ee.Algorithms.Image

ee.Algorithms.Landsat

ee.Algorithms.Sentinel2

ee.Algorithms.TemporalSegmentation

ee.Algorithms.CannyEdgeDetector(image, threshold, *sigma*)

ee.Algorithms.Collection(features)

ee.Algorithms.CrossCorrelation(imageA, imageB, maxGap, windowSize, *maxMaskedFrac*)

ee.Algorithms.Date(value, *timeZone*)

ee.Algorithms.Describe(input)

ee.Algorithms.Dictionary(*input*)

ee.Algorithms.Feature(*geometry, metadata, geometryKey*)

ee.Algorithms.HillShadow(image, azimuth, zenith, *neighborhoodSize, hysteresis*)

ee.Algorithms.HoughTransform(image, *gridSize, inputThreshold, lineThreshold, smooth*)

ee.Algorithms.If(*condition, trueCase, falseCase*)

ee.Algorithms.IsEqual(*left, right*)

ee.Algorithms.ObjectType(*value*)

ee.Algorithms.Proj(*crs, transform, transformWkt*)

ee.Algorithms.ProjectionTransform(*feature, proj, maxError*)

ee.Algorithms.String(input)

ee.Algorithms.Terrain(input)

ee.Classifier

ee.Classifier.amnhMaxent(*categoricalNames, outputFormat, autoFeature, linear, quadratic, product, threshold, hinge,*

ee.Classifier.decisionTree(*treeString*)

ee.Classifier.decisionTreeEnsemble(*treeStrings*)

ee.Classifier.libsvm(*decisionProcedure, svmType, kernelType, shrinking, degree, gamma, coef0, cost, nu, terminationE*

ee.Classifier.minimumDistance(*metric, kNearest*)

ee.Classifier.smileCart(*maxNodes, minLeafPopulation*)

ee.Classifier.smileGradientTreeBoost(*numberOfTrees, shrinkage, samplingRate, maxNodes, loss, seed*)

ee.Classifier.smileNaiveBayes(*lambda*)

ee.Classifier.smileRandomForest(*numberOfTrees, variablesPerSplit, minLeafPopulation, bagFraction, maxNodes, see*

ee.Classifier.spectralRegion(*coordinates, schema*)

confusionMatrix()

explain()

mode()

schema()

setOutputMode(mode)

train(features, classProperty, *inputProperties, subsampling, subsamplingSeed*)

ee.Clusterer

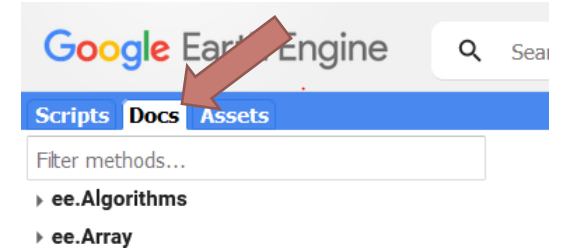
ee.Clusterer.wekaCascadeKMeans(*minClusters, maxClusters, restarts, manual, init, distanceFunction, maxIterations*)

ee.Clusterer.wekaCobweb(*acuity, cutoff, seed*)

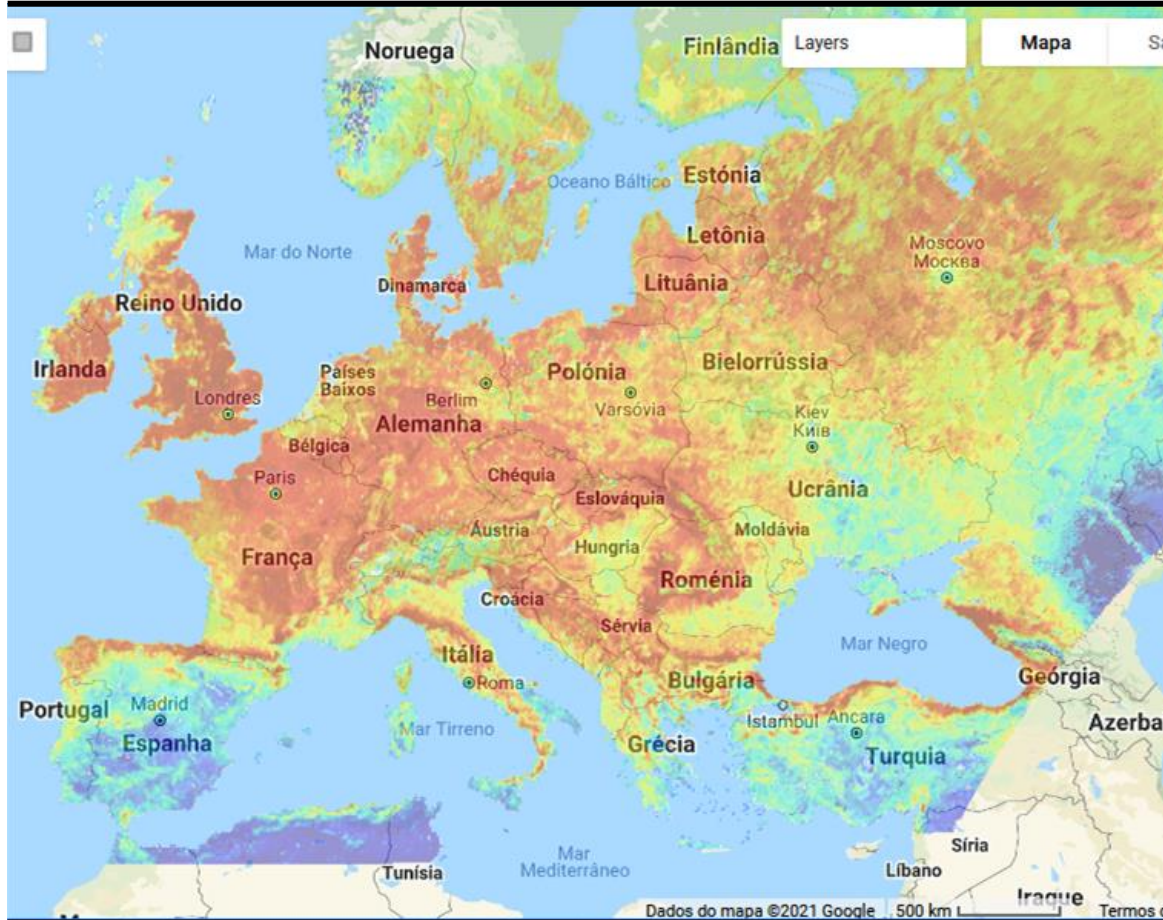
ee.Clusterer.wekaKMeans(*nClusters, init, canopies, maxCandidates, periodicPruning, minDensity, t1, t2, distanceFunc*

ee.Clusterer.wekaLVQ(*numClusters, learningRate, epochs, normalizeInput*)

ee.Clusterer.wekaXMeans(*minClusters, maxClusters, maxIterations, maxKMeans, maxForChildren, useKD, cutoffFact*



GEE Examples



Repository with some examples:

https://code.earthengine.google.com/?accept_repo=users/sofiaermida/EUMETrain

Sample of LSA-SAF data:

<https://code.earthengine.google.com/?asset=user/s/sofiaermida/LSASAF>

Landsat LST repository:

https://code.earthengine.google.com/?accept_repo=users/sofiaermida/landsat_smw_lst