GOOGLE EARTH ENGINE (GEE) TUTORIAL

Created by: Erika Higa

Adapted by: Megs Seeley & Dane Coats

GEE 101 – Basic Introduction

This tutorial requires no coding experience or familiarity with GEE. This tutorial is a general introduction to Google Earth Engine. Please see the complete GEE API for further learning: <u>https://developers.google.com/earth-engine/</u>

Section 0 – JAVASCRIPT OVERVIEW

Comments

// Line comments start with two forward slashes

/* multi line comments

start with a forward slash,

and end with a star and forward slash */

Remember your semicolons ;

Like javascript, all statements end with a semicolon

var example = 24;

Objects

Define objects using var

Numbers

var thisNumber = 24;

Strings

String objects have quotes. Can be single or double quotes.

var thisString = 'This is a tutorial';

Lists

Square brackets are used for items in a list. Use indices to call items in a list (starting at 0).

var myList = ['python', 'R', 'javascript'];

print (myList[0]);

Dictionaries

Curly brackets (or braces) can be used to define dictionaries (key : value pairs).

var my_dict = {
 'food':'bread',
 'color':'red',
 'number':87
};

Section 1 – CODE EDITOR

← → ⊂ ŵ	⑦ ▲ https://code.earthengine.google.com	… 🖸 🕁	⊻ III\ 🖸 🗎 ≡
Google Earth Engine	Search places and datasets 5		6 Help + ehigaearth +
Scripts Docs Assets Filter scripts > Owner (1) > Writer (2) > Reader > Examples > Archive No accessible repositories. 3	NEW Cet Link Save Run Reset Run	Inspector Console Tasks Use print() to write	to this console.
() () // () + -	WASHINGTON MONTANA DAROTA MINESOTA MINESOTA Ottawa Montral OREGON IDAHO WYOMING A MICHIGAN TO TOTO OREGON IDAHO WYOMING A MICHIGAN TO TOTO WYOMING A MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN A TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO MICHIGAN TO TOTO MICHIGAN TO TOTO MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TOTO NEW YORK MICHIGAN TO TOTO MICHIGAN TO TO TO TO TO TOTO MICHIGAN T		Map Satellite

- 1. Editor Panel -- writing and editing code
- 2. Right Panel
 - a. Console: viewing printed output
 - b. Inspector: Querying map
 - c. Tasks: Managing long-running tasks.
- 3. Left Panel
 - a. Scripts: Managing your programming scripts.
 - b. Docs: Accessing documentation of Earth Engine objects and methods
 - c. Assets: Managing assets that you uploaded
- 4. Interactive Map -- visualizing map layer outputs
- 5. Search Bar -- finding datasets and places of interest
- 6. Help Menu
 - a. User guide: reference documentation
 - b. Help forum: Google group for discussing Earth Engine
 - c. Shortcuts: Keyboard shortcuts for the Code Editor
 - d. Feature Tour: overview of the Code Editor
 - e. Feedback: submitting feedback on the Code Editor

f. Suggest a dataset: suggest which datasets GEE developers should ingest into the Earth Engine public archive.

Section 2 – IMPORT IMAGES

Vocab! Image: A single raster layer Image collection: A collection, or stack, of rasters Features: Points, polygons, or lines Feature collection: Multiple features

Import STRM 30m DEM

- 1. In the search bar, type "elevation"
- 2. Choose "SRTM Digital Elevation Data 30m"
- 3. On the right side of the window with the dataset metadata, there is an "import" button. Click import
- 4. The image is now in the Imports section of your code editor. By default, the variable name is "image" so rename it to "srtm".

Imports (1 entry) var srtm: Image "SRTM Digital Elevation Data 30m" (1 band)

Note: Alternate step to add an image collection

Instead of using the search bar, you can import the image using its Image ID and ee.Image()

In this SRTM example, the ID is "USGS/SRTMGL1_003"

var srtm = ee.Image('USGS/SRTMGL1_003');

Visualizing Image Collection

Explore the data in the right panel, console tab using:

print(srtm);

We will view the srtm image on the interactive map using the Map.addLayer(). When plotting maps, we can define the visualization parameters (such as color scale, min-max values, etc.).

Map.addLayer(layerName, {min:lowestPixelValue, max:greatestPixelValue, palette ['color1', 'color2', 'color3'}, 'Displayed Layer Name');

Here, we will stretch the color gradient from 0 to 3000. We will also name the layer "SRTM"

Map.addLayer(srtm, {min:0, max:3000}, 'SRTM');

Section 3 – Region of Interest

We can add a Region of Interest (ROI) to the map either as a point, line, or polygon.

ROIs allow us to process only the images within our study area, which speeds up computation time.

Note: We can import shapefiles from outside GEE in the "Assets" tab

Point

We will start with a point. Click on the \P icon and draw a point anywhere on the map. If we want to change the location of the point, we can simply click on it and drag it to a new location.

In the Imports section, rename the point object to "roi".

Polygon

If you want to create a larger roi, use the polygon \bowtie icon to draw a polygon.

Section 4 – FILTER IMAGE COLLECTIONS & COLOR COMPOSITES

Let's work with an image collection – Landsat 8.

Search for "Landsat 8 toa" and import the "USGS Landsat 8 Collection 1 Tier 1 TOA Reflectance" product. Rename the collection to "landsat8".

Note: For importing image collections use ee.ImageCollection() method

Filtering Image Collections

Filter by the ROI and date

var landsat8_filtered = landsat8
 .filterDate('2017-09-01', '2017-12-01')
 .filterBounds(roi);

The output will be grayscale, so let's apply better visuals. We can apply the parameters as a variable first so it can be used more often easily.

var trueColor = {min:0, max:0.3, bands:['B4', 'B3', 'B2']};

var falseColor = { min:0, max:0.3, bands:['B5', 'B4', 'B3']};

Now let's use these variables and map the landsat image in true and false color.

Map.addLayer(landsat8_filtered, trueColor, 'RBG color');

Map.addLayer(landsat8_filtered, falseColor, 'RBG false color');

Bonus: Filtering by path and row

Note: Individual images in remotely sensed imagery can be referred to as scenes or tiles.

We can filter for specific scenes using the filterMetadata() method. Let's filter for a scene over the Los Angeles, CA region (path 41, row 36).

var landat8_LA = landsat8

.filterMetadata('WRS_PATH', 'equals', 41)

.filterMetadata('WRS_ROW', 'equals', 36);

Section 5 – Functions & NDVI

Functions make our lives easier. Rather than performing the same task over and over again, we can create a function. We will write a function, called addNDVI, to add an NDVI band to every image in a collection.

First, we should mention that GEE has a build-in function to calculate the NDVI: normalizedDifference().

function addNDVI (i) { // Defines the function "addNDVI" and the input values "i"
var ndvi=i.normalizedDifference(['B5','B4']); // Create the NDVI band
return i.addBands(ndvi.rename('NDVI')); // Add NDVI band to image and rename
}

Now apply the function to the image collection using map()

var with_ndvi = landsat8_filtered.map(addNDVI);

map() is a GEE method that applies a function to every image in an image collection.

Now let's map the result!

First, make a palette for our NDVI bands from red to green

var palette = ['ff4d4d', '339900'];

Then map it!

Map.addLayer(with_ndvi,{bands:'NDVI',min:0, max:1, palette: palette}, 'NDVI - collection');