Geospatial_analysis.md

An Introduction to Geospatial Analysis Using the Google Earth Engine Platform

TC Chakraborty

&

Sabrina Szeto

Nov 13, 2018

https://github.com/TC25/Geospatial_workshop_EE

https://signup.earthengine.google.com

Agenda

Introduction to Geospatial Analysis
Data Types
The Google Earth Engine platform
Basic functions
Variable types
Geometry declaration
Functions and mapping
Operations on Geometries
Features and Feature Collections
Operations on Features
Filters
Operations on Images
Operations on Image Collections
Importing and exporting Data
Example Applications
Resources

Introduction

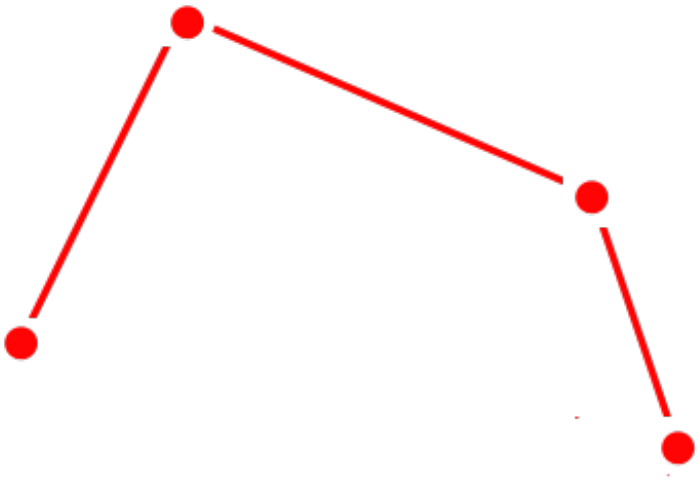
Collection, visualization, and analysis of geographical or spatial data.

Data types

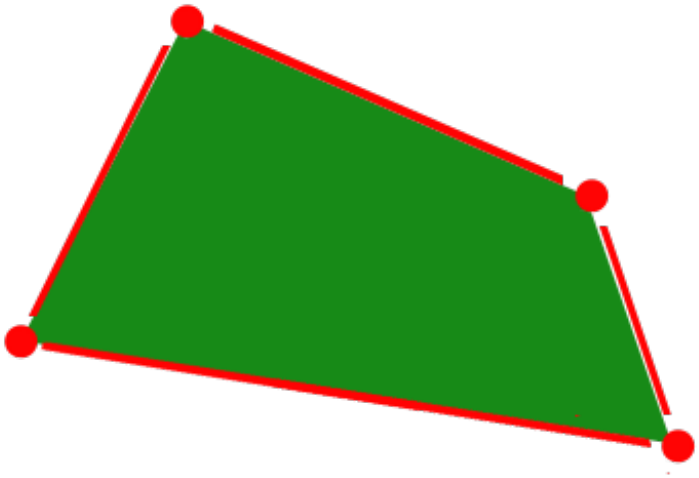
- Vector data represent lat-long coordinates
- Raster data comprises of pixels with associated values

Points

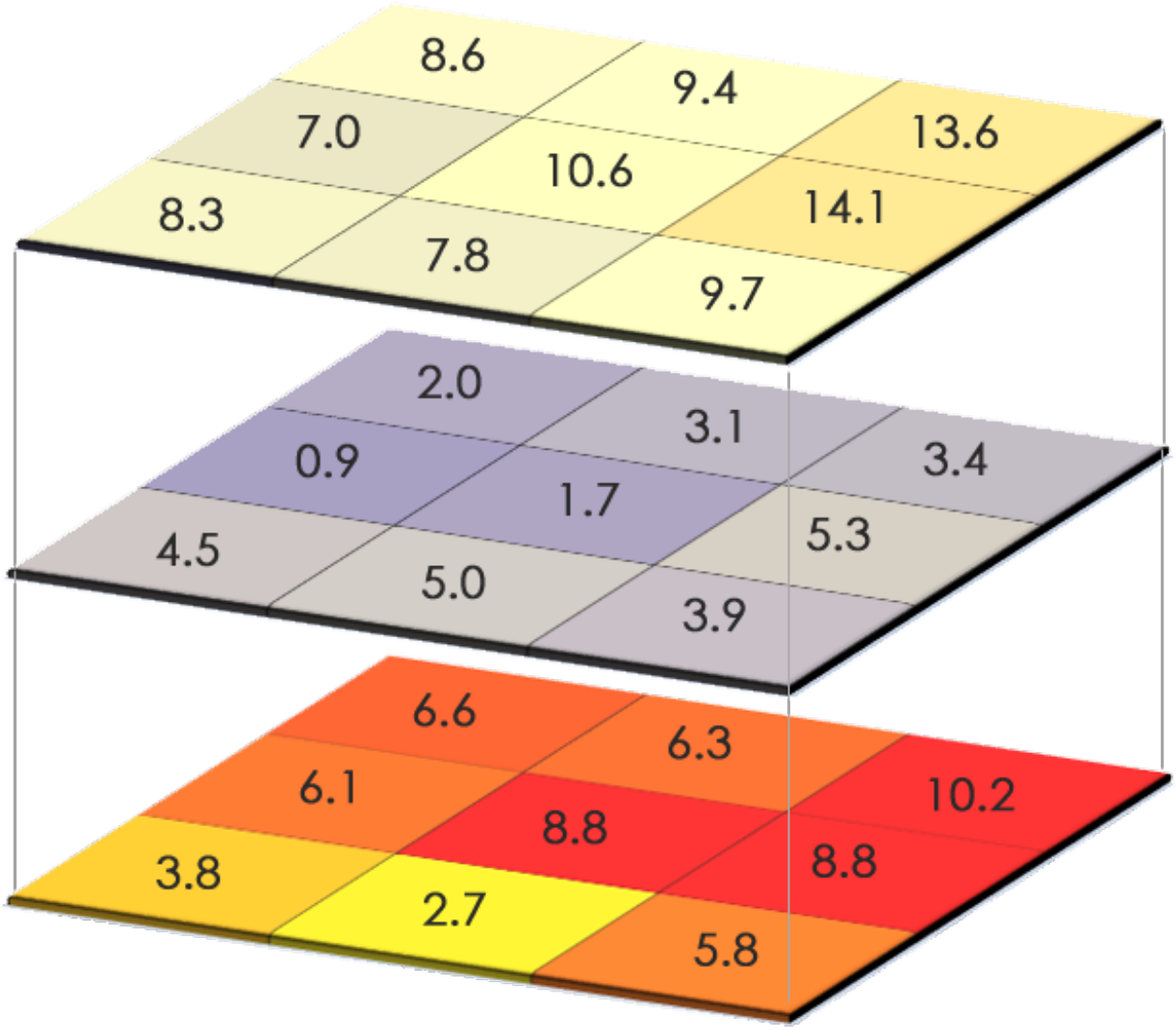
Lines



Polygons



Raster layers/bands



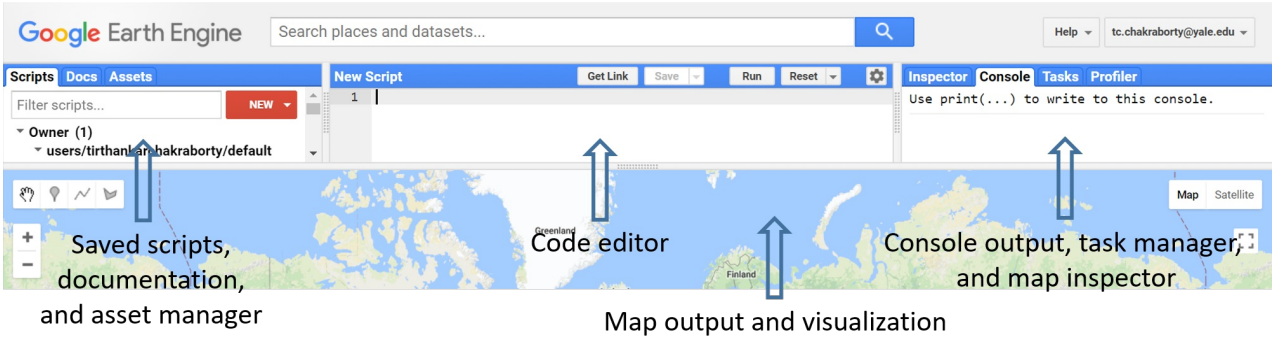
Google Earth Engine platform

Introductory video

Code Editor

- Cloud-based platform for planetary scale geospatial analysis
- Uses Google's computational resources to reduce processing time
- Massive archive of remote sensing data
- 200 public datasets
- greater than 4000 new images every day
- greater than 5 million images
- greater than 5 petabytes of data

Source: Google Earth Engine User summit



Basic Functions

Declaring variables

```
var varname = Containerforvariabletype(variable name);
```

Centering map

```
Map.setCenter(long, lat, zoom level);
```

> Zoom level varies from 0 (no zoom) to 20 (highest zoom level)

Displaying metadata

```
print(variable name)
```

Adding a layer to the map

```
Map.addLayer(VARIABLENAME);
```

Variable types in Earth Engine

Strings

```
var var_String = ee.String("This is a string. Or is it? It is.");
```

Numbers

```
var var_Numbers = ee.Number(5);
```

Arrays

```
var var_Array = ee.Array([[5, 2, 3],  [-2, 7, 10],  [6, 6, 9]]);
```

Lists

```
var var_List = ee.List([5, "five" , 6, "six"]);
```

Dictionaries

```
var var_Dictionary = ee.Dictionary({five: 5 , six: 6});
```

And the fun stuff

- Geometries
- Features
- Feature Collections
- Images
- Image Collections

Geometries – declaration and types

Points

```
var var_Point = ee.Geometry.Point(0, 45);
```

Multi Points

```
var var_MultiPoint = ee.Geometry.MultiPoint(0, 45, 5,6, 70,-56);
```

Line String

```
var var_LineString = ee.Geometry.LineString([[0, 45], [5,6], [70,-56]]);
```

Multi Line String

```
var var_MultiLineString = ee.Geometry.MultiLineString([[[0, 45], [5,6], [70,-56]], [[0, -45], [-5,-6], [-70,56]]]);
```

Linear Ring

```
var var_LinearRing = ee.Geometry.LinearRing(0, 45, 5,6, 70,-56, 0,45);
```

Rectangle

```
var var_Rectangle = ee.Geometry.Rectangle(0, 0, 60,30);
```

Polygon

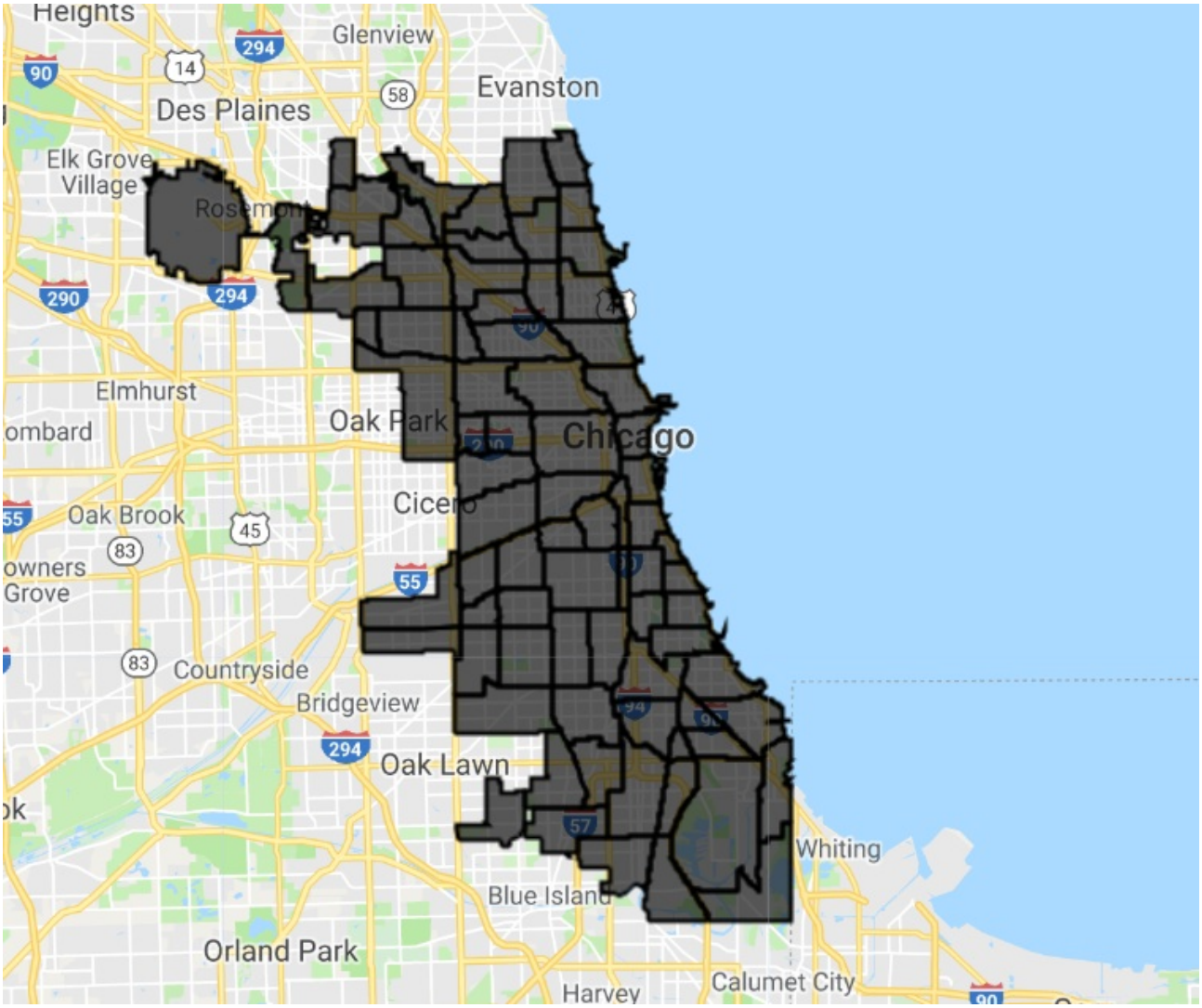
```
var var_Polygon = ee.Geometry.Polygon([[[0, 0], [6,3], [5, 5], [-30,2], [0,0]]]);
```

Multi Polygon

```
var var_MultiPolygon = ee.Geometry.MultiPolygon([ee.Geometry.Polygon([[0, 0], [6, 3], [5, 5], [-30, 2], [0,0]]), ee.Geometry.Polygon([[0, 0], [-6, -3], [-5, -5], [30, -2], [0, 0]])]);
```

Features and Feature Collections

- Features are geometries associated with specific properties
- Feature Collections are groups of features



Functions and mapping

A set of instructions to perform a specific task

```
function  function_Name(Arguments) {statements};
```

Call function

```
var result = function_Name(Input);
```

Map function over Feature or Image Collection

```
var result = Input.map(function_Name);
```

Operations on Geometries

Geometry operations

Find area of geometry

```
var Geometry_area = var_Geometry.area();
```

Find length of line

```
var Line_length = var_LineString.length();
```

Find perimeter of geometry

```
var Geometry_perimeter = var_Geometry.perimeter();
```

Reduce number of vertices in geometry

```
var SimplifiedGeometry = var_Geometry.simplify(100);
```

Find centroid of geometry

```
var Centroid = var_Geometry.centroid();
```

Create buffer around geometry

```
var Buffer = var_Geometry.buffer(100);
```

Find bounded rectangle of the Geometry

```
var BoundedGeometry = var_Geometry.bounds();
```

Find the smallest envelope that can envelop the Geometry

```
var Convexhull_Geometry = var_Geometry.convexHull();
```

Find common area between two or more geometries

```
var Inter_geometry = var_Geometry1.intersection(var_Geometry2);
```

Find area that includes two or more geometries

```
var Union_geometry = var_Geometry1.union(var_Geometry2);
```

Operations on Features

Feature operations

Set property name and value of geometry to create a feature

```
var var_Feature = ee.Feature(var_Geometry, {Name: "Feature name", Size: 500};
```

Create a new feature from existing feature while renaming a property

```
var var_Featurenew = var_Feature.select(["Name"], ["Descriptor"]);
```

Extract values of a property from a Feature

```
var values = var_Feature.get(''Size'');
```

Filters

Creator a filter for values of a property

```
var BFilter = ee.Filter.eq(Property_name, Value);
```

> or .neq , .gt , .gte , .lt , and .lte

Create a filter based on maximum difference from a threshold

```
var DiffFilter = ee.Filter.maxDifference(threshold, Property_name, Value);
```

Create a text filter

```
var TxtFilter = ee.Filter.stringContains( Property_name, StringValue);
```

> or .stringStartsWith, and .stringEndsWith

Create a range filter

```
var RangeFilter = ee.Filter.rangeContains( Property_name, StringValue, MinValue,
MaxValue);
```

Create a list filter to check for certain values

```
var ListFilter = ee.Filter.listContains(Property_name, Value1, Property_name2, Value2);
```

> .inList to test against list of values

Create a filter of dates

```
var DateFilter = ee.Filter.calendarRange(StartDate, StopDate);
```

Create a filter for particular days of the year

```
var DayFilter = ee.Filter.dayOfYear(startDay, StopDay);
```

Create a filter to subset geospatial data

```
var BoundsFilter= ee.Filter.bounds(GeometryorFeature);
```

Combining and inversing filters

```
var NewFilter=ee.Filter.and(Listoffilters);
var NewFilter=ee.Filter.or(Listoffilters);
var inverseFilter = ee.Filter.not(filter);
```

Operations on Images

Image operations

Selecting the bands of an image

```
var band = var_Image.select(band name);
```

Creating masks

```
var mask =var_Image.eq(value);
```

> or .neq or .gt or .gte or .lt or .lte

Applying masks

```
var masked =var_Image.mask(mask);
```

Pixelwise calculation

```
var results =var_Image.sum(value);
```

> or .subtract , .multiply , .divide , .max , .min , .abs , .round , .floor , .ceil , .sqrt , .exp, .log, .log10, .sin , .cos , .tan , .sinh , .cosh , .tanh , .acos, .asin

Shift pixels of an image

```
newImage = oldImage.leftShift(valueofshift);
```

> or .rightShift

Create a single value from an image by applying a reducer based on regions of interest

```
var outputDictionary = var_Image.reduceRegion(Reducer, var_Geometry, scale);
```

Operations on Image Collections

Select the first n numbers of images in a collection (based on property)

```
var SelectedImages =var_ImCollection.limit (n, Property_name, Order);
```

Select images in collection based on particular properties

```
var SelectedImages = var_ImCollection.filterMetadata (Property_name, Relation , Value);
```

> Relations could be "equals", "less_than", "greater_than", "starts_with", "ends_with", and "contains"

Select images within date range

```
var SelectedImages = var_ImCollection.filterDate (StartDate, StopDate);
```

Select images within Geometry

```
var SelectedImages = var_ImCollection.filterBounds (var_Geometry);
```

Perform pixelwise calculations for all images in collection

```
var sumofimages = var_ImCollection.sum();
```

> or .product, .max, .min, .mean, .mode, .median, and .count

Create composite of images in collection with the last image on top

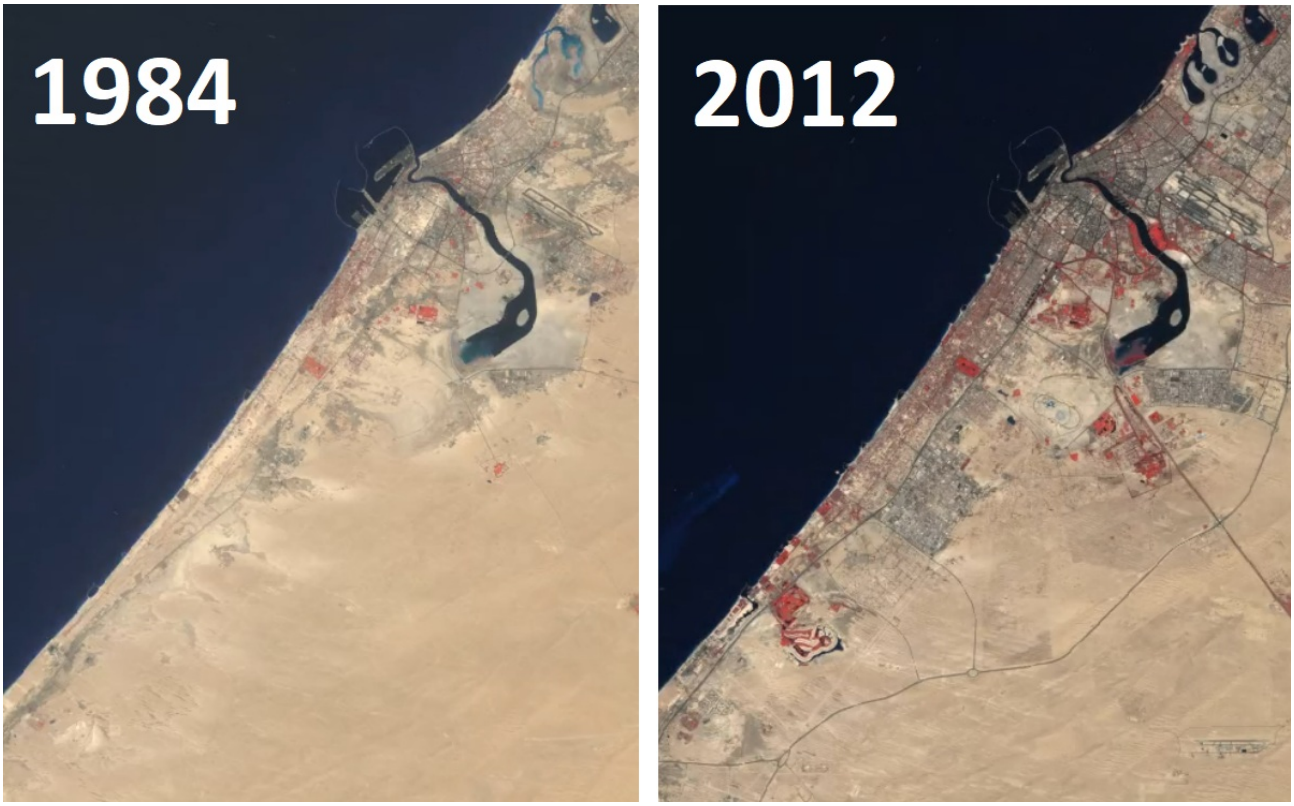
```
var mosaicofimages = var_ImCollection.mosaic();
```

Importing and exporting data

Image to table example

Timelapse example

Dubai timelapse



Export image, video or table to Google Drive, Asset, or Google Cloud

```
Export.image.toDrive({
  collection: var_Image, description: 'FileName', region: var_Geometry, scale: 1000}
});
```

> or image.toCloudStorage, image.toAsset, table.toDrive, table.toCloudStorage, video.toCloudStorage, and video.toDrive

Example Applications

What can you do with Google Earth Engine?

EE Population Explorer

EE Ocean Time Series Investigator

Global Surface UHI Explorer

Stratifi - cloud-based stratification

Resources

Geospatial Software Design

Google Earth Engine API documentation

Google Earth Engine Developers forum

Example scripts from Prof. Dana Tomlin's handouts for his course on Geospatial Software Design

- tc.chakraborty@yale.edu
- sabrina.szeto@yale.edu