WORLD FOOD AND ECOSYSTEMS practical 1

BSC Future Planet Studies
Ac. Year 2021-2022

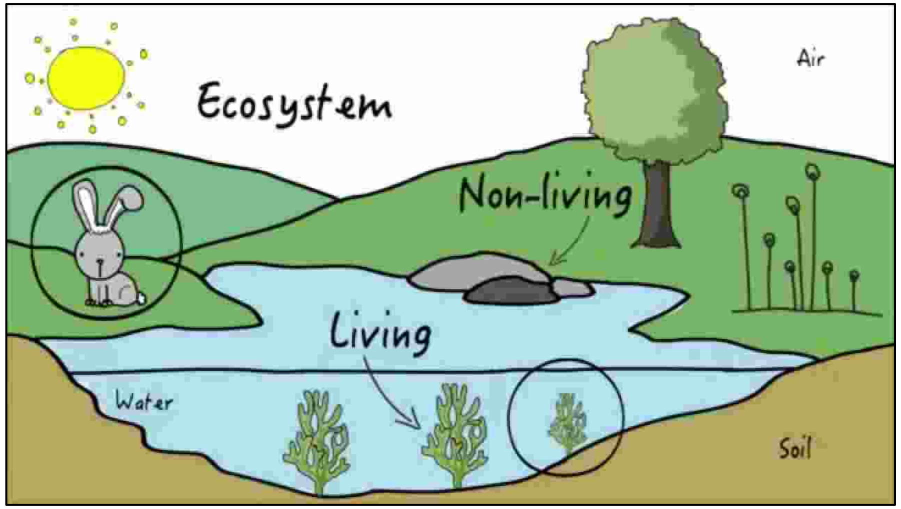
INTRODUCTION TO GEOPROCESSING

World food- and ecosystems are:

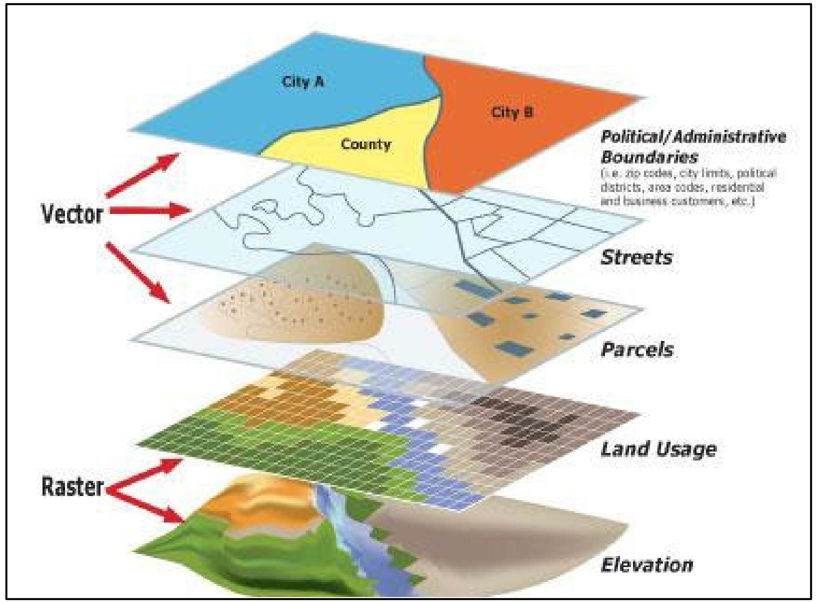
- Complex
- Multidimensional
- Context specific (space + time)
- Qualitative, generalized statements are difficult

GIS can help to:

- Combine/Visualize
- Quantify/Analyze
- Understand
- Predict



Len.com.ng


City A
City B
County
Political/Administrative Boundaries
(i.e. zip codes, city limits, political districts, area codes, residential and business customers, etc.)
Vector
Streets
Parcels
Land Usage
Raster
Elevation

newbergoregon.gov

INTRODUCTION TO GEOPROCESSING

World food- and ecosystems are:

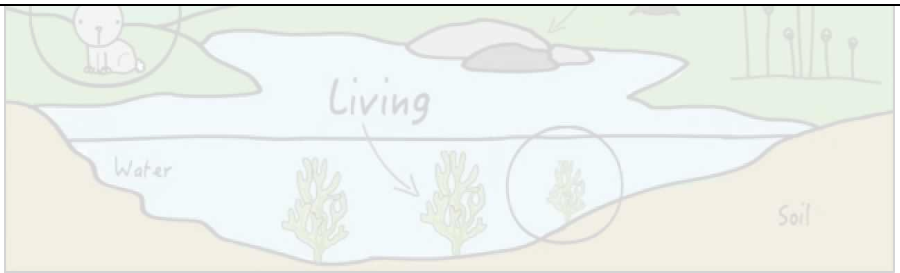
GIS can help to:

This course:

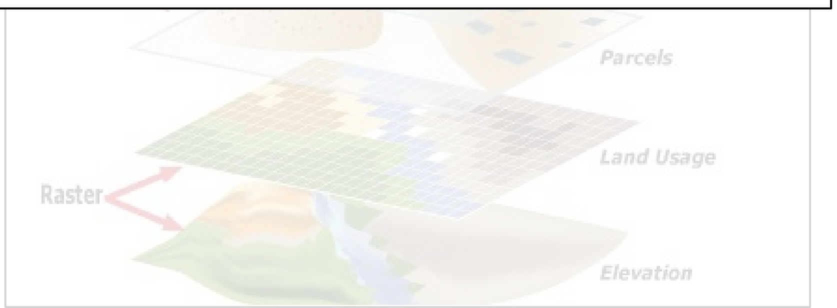
Not an introduction to spatial data/GIS/remote sensing

Intro into how to access/visualise/manipulate/…, i.e. to

use spatial data in understanding global food and ecosystems



Len.com.ng



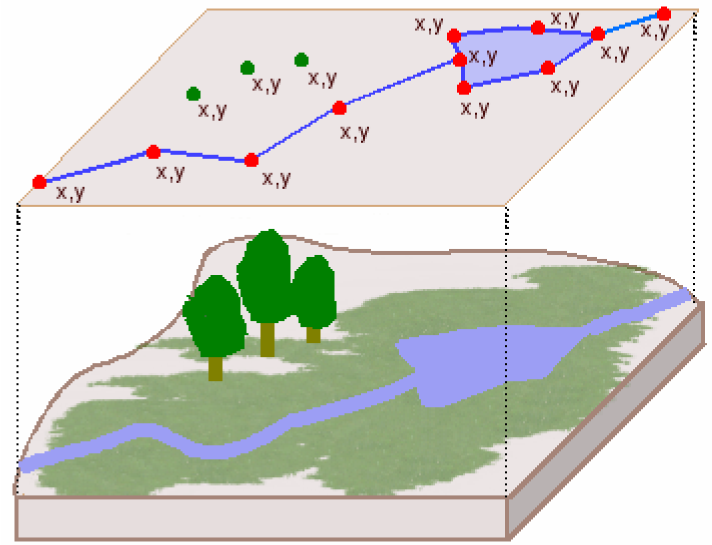
newbergoregon.gov

Type of data (vector)

- Point data: e.g. species occurrence data

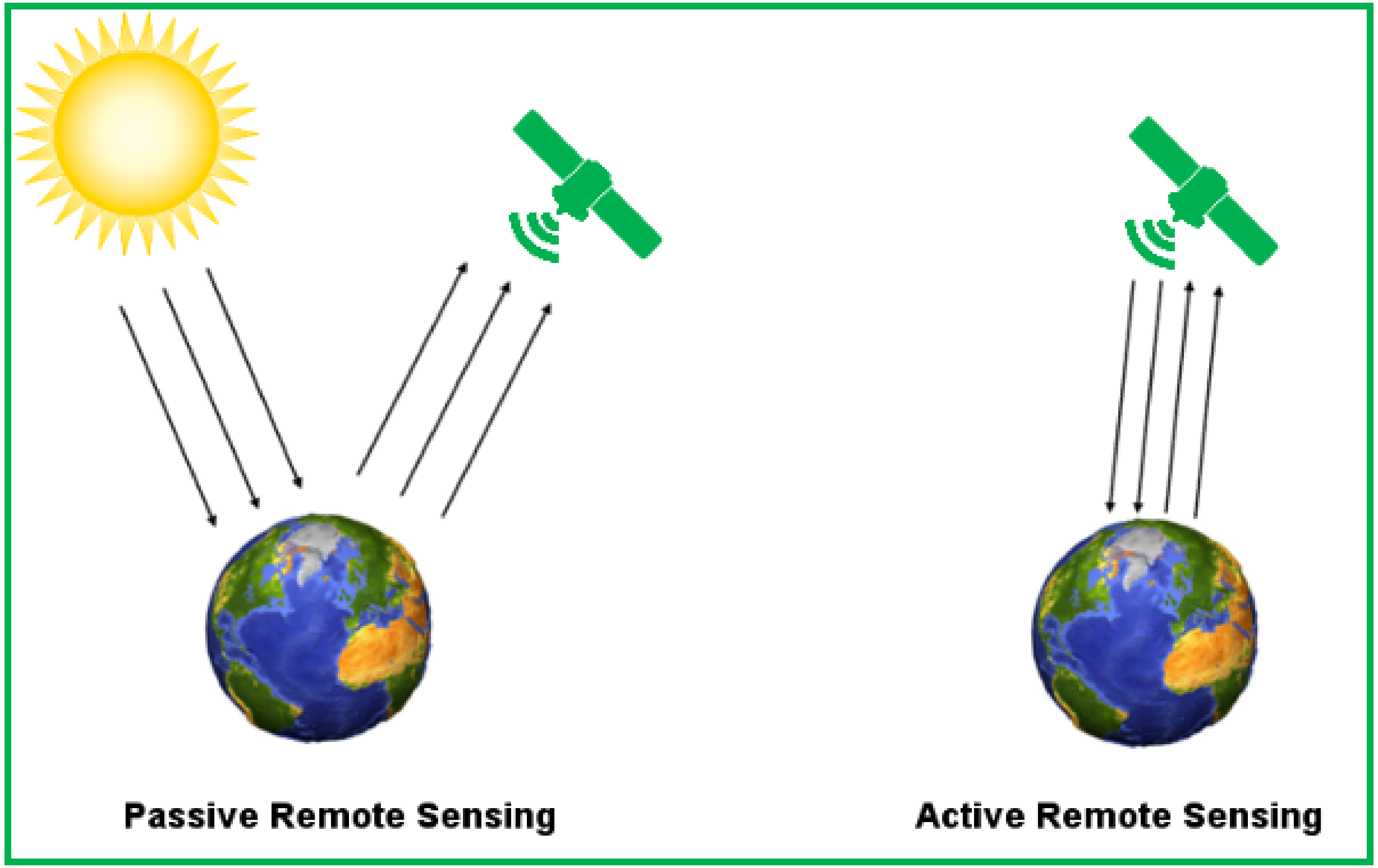
- Line data: e.g. transect, river, fault line…

- Polygon data: e.g. country boundaries, parcel boundary,



geography.hunter.cuny.edu

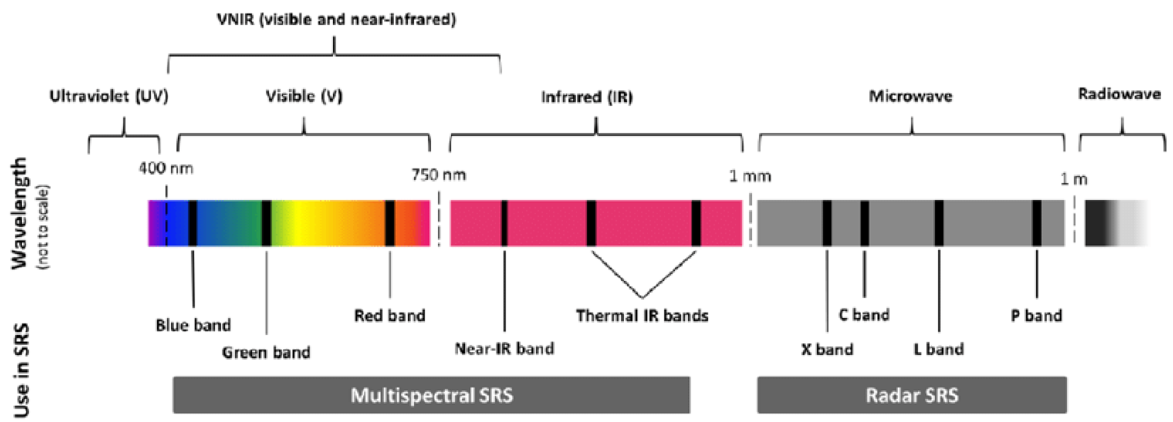
Type of data (raster)



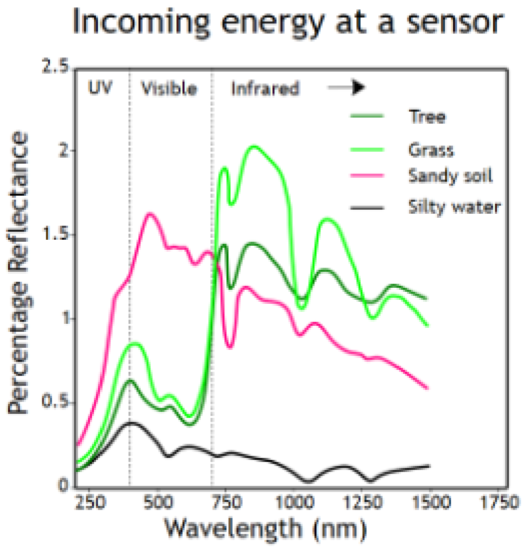
Grindgis.com

Type of data (raster)

- Derivatives of passive remote sensing:
    - Blue band: atmospheric corrections
    - Green band: helps visualize vegetation
    - Red band: absorbed by vegetation: monitoring vegetation and calculating vegetation indices (e.g. NDVI)
    - NIR: identification of water bodies (strong absorption)
    - Short-mid-IR: sensitive to water content
    - Thermal bands: sensitive to temperature, good to detect e.g. fires



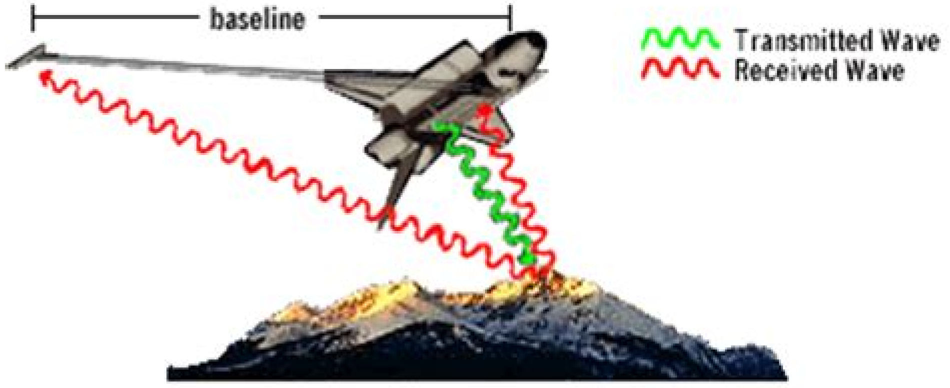
Pettorelli et al. 2018



$$\text{NDVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red})}$$

Type of data (raster)

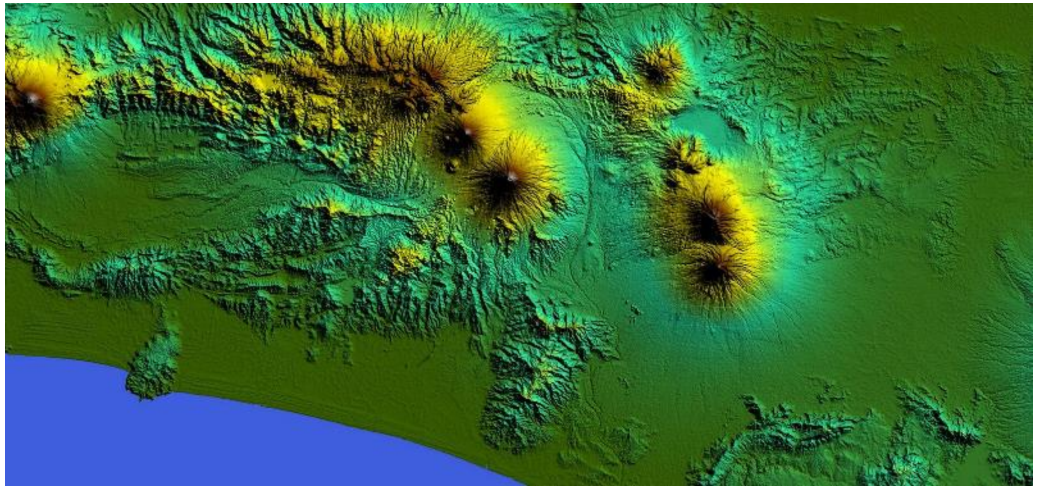
- Derivatives of active remote sensing:
    - Topographic information: e.g. SRTM Digital Elevation Model

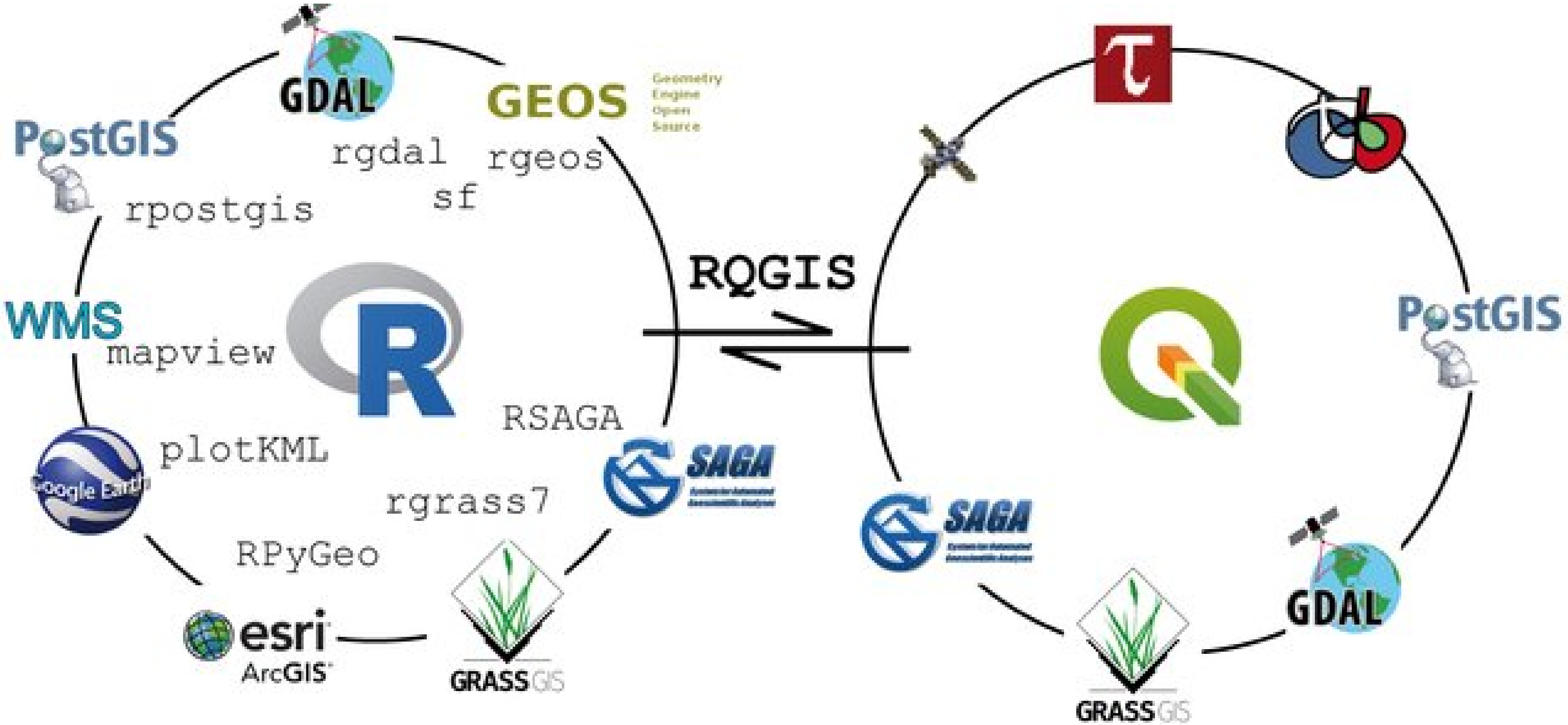


Radar signals being transmitted and recieved in the SRTM mission (image not to scale).

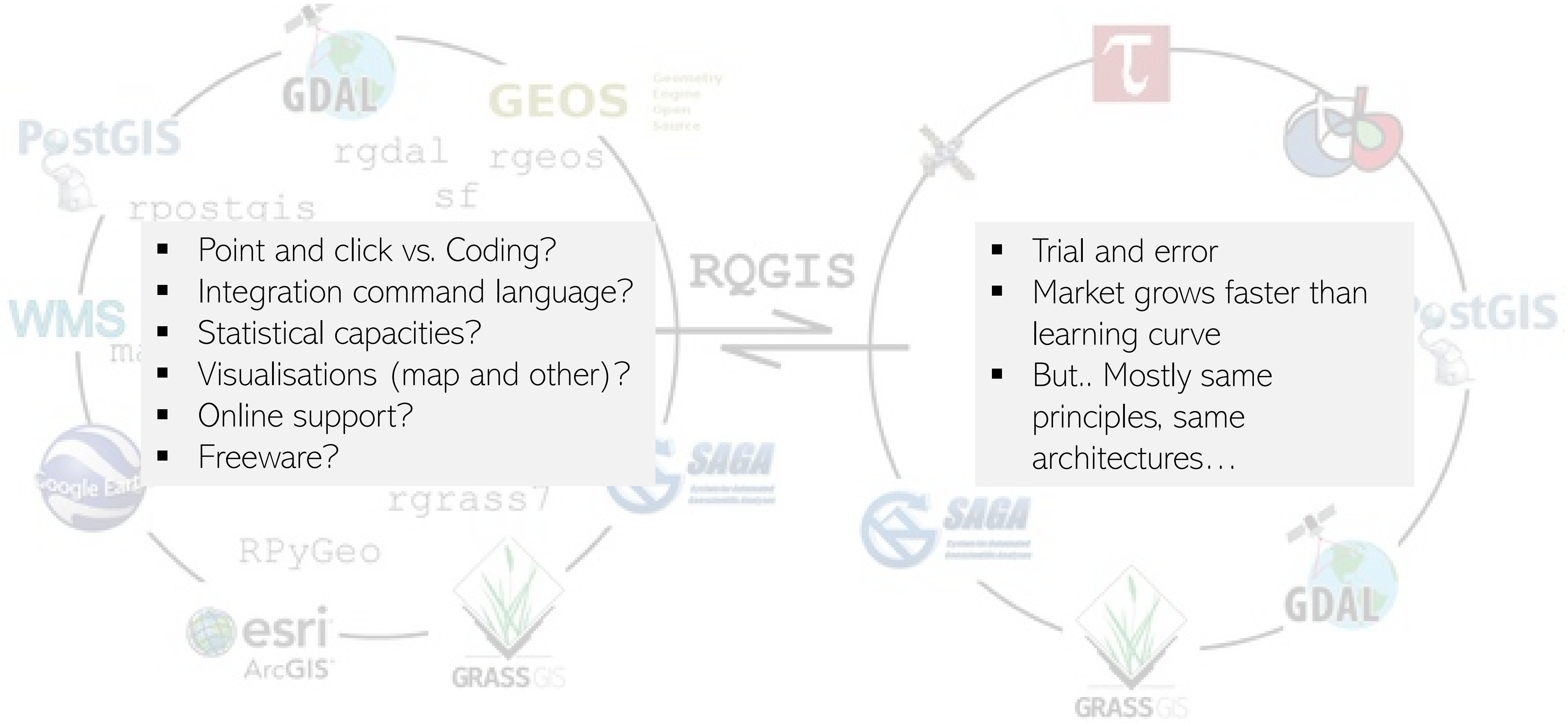
Jpl.nasa.gov

- Raster maps derived from other mapping efforts (or vectors), model outputs, weather sattelites….





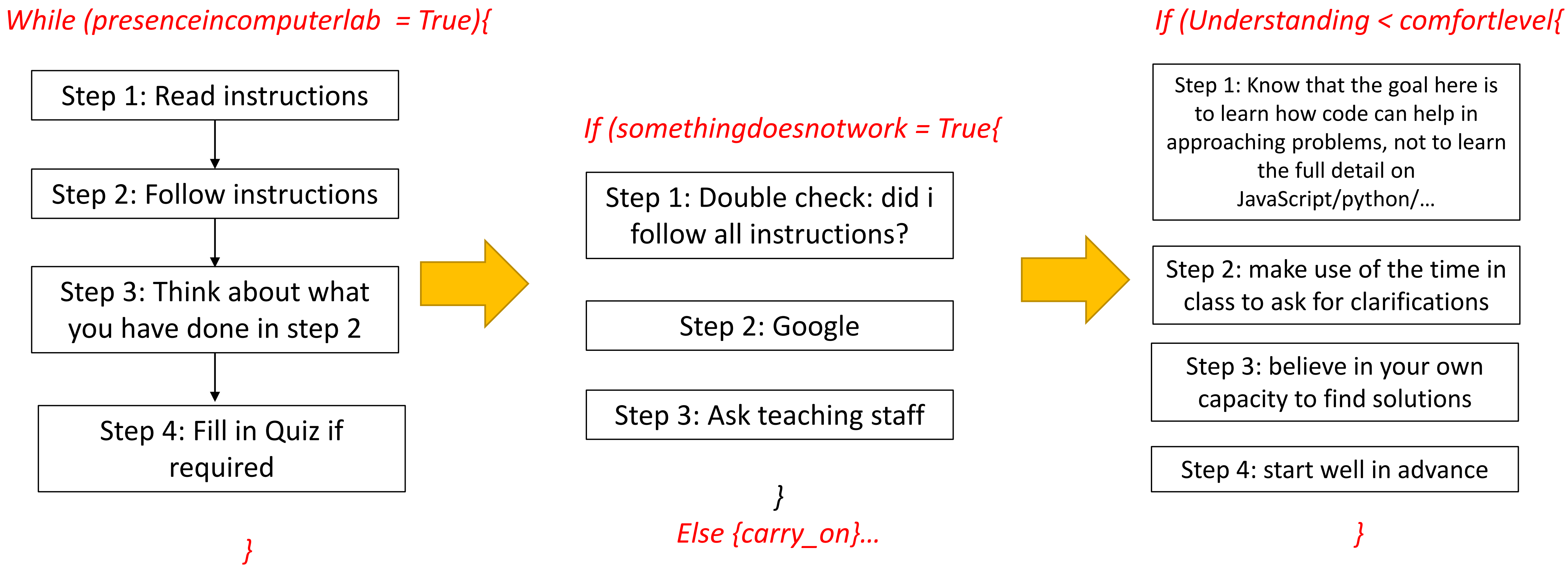
Muenchow et al. 2017, DOI: 10.32614/RJ-2017-067



- Point and click vs. Coding?
- Integration command language?
- Statistical capacities?
- Visualisations (map and other)?
- Online support?
- Freeware?

- Trial and error
- Market grows faster than learning curve
- But.. Mostly same principles, same architectures…

- Google Earth Engine (JavaScript)
- Rstudio (R)
- QGIS (python)



While (presenceincomputerlab = True){

Step 1: Read instructions

Step 2: Follow instructions

Step 3: Think about what you have done in step 2

Step 4: Fill in Quiz if required

}

If (somethingdoesnotwork = True{

Step 1: Double check: did i follow all instructions?

Step 2: Google

Step 3: Ask teaching staff

}

Else {carry_on}…

If (Understanding < comfortlevel{

Step 1: Know that the goal here is to learn how code can help in approaching problems, not to learn the full detail on JavaScript/python/…

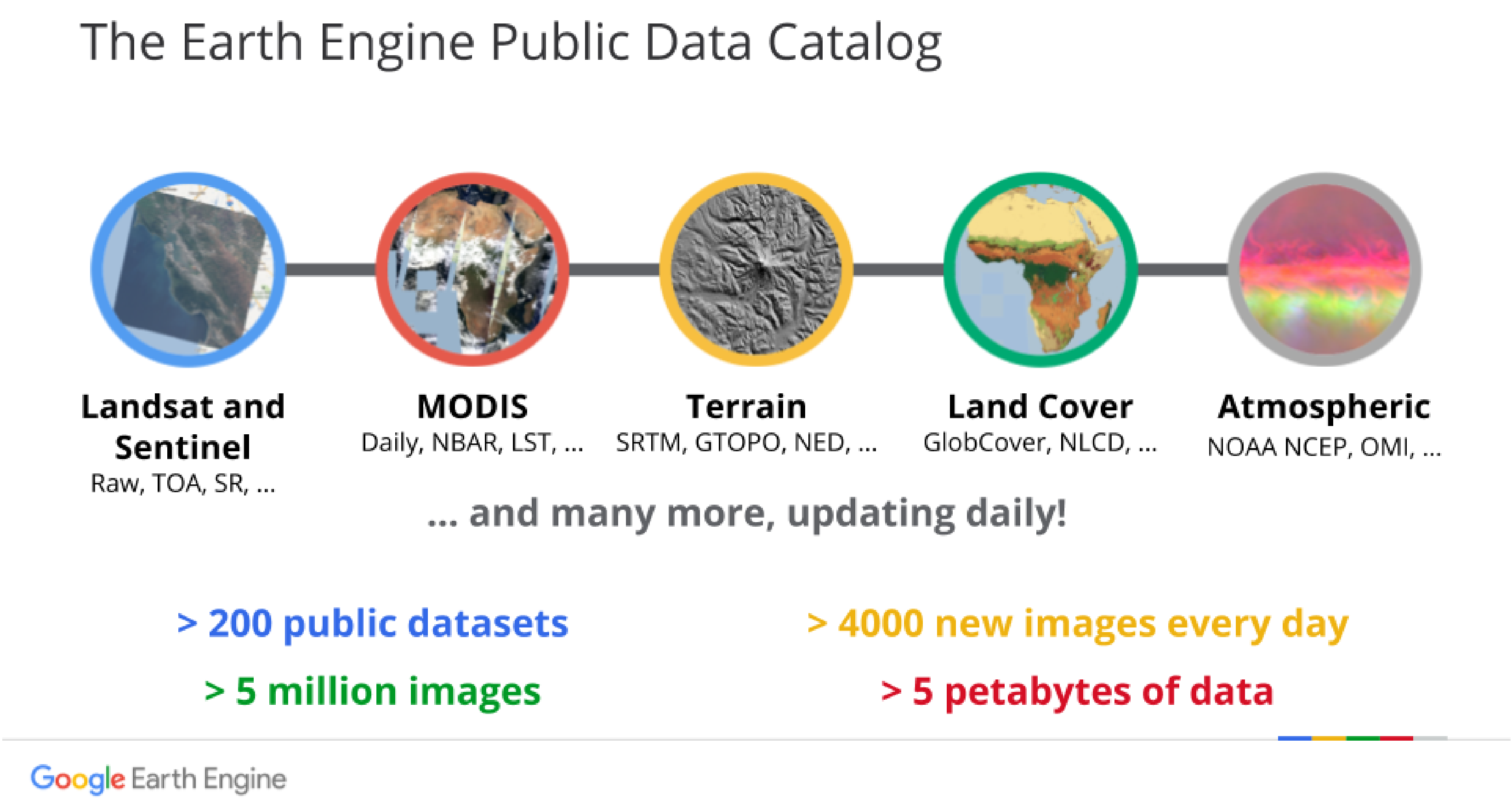
Step 2: make use of the time in class to ask for clarifications

Step 3: believe in your own capacity to find solutions

Step 4: start well in advance

}

Session 1: understanding gradients in GEE



The Earth Engine Public Data Catalog

Landsat and Sentinel
Raw, TOA, SR, ...

MODIS
Daily, NBAR, LST, ...

Terrain
SRTM, GTOPO, NED, ...

Land Cover
GlobCover, NLCD, ...

Atmospheric
NOAA NCEP, OMI, ...

... and many more, updating daily!

> 200 public datasets

> 4000 new images every day

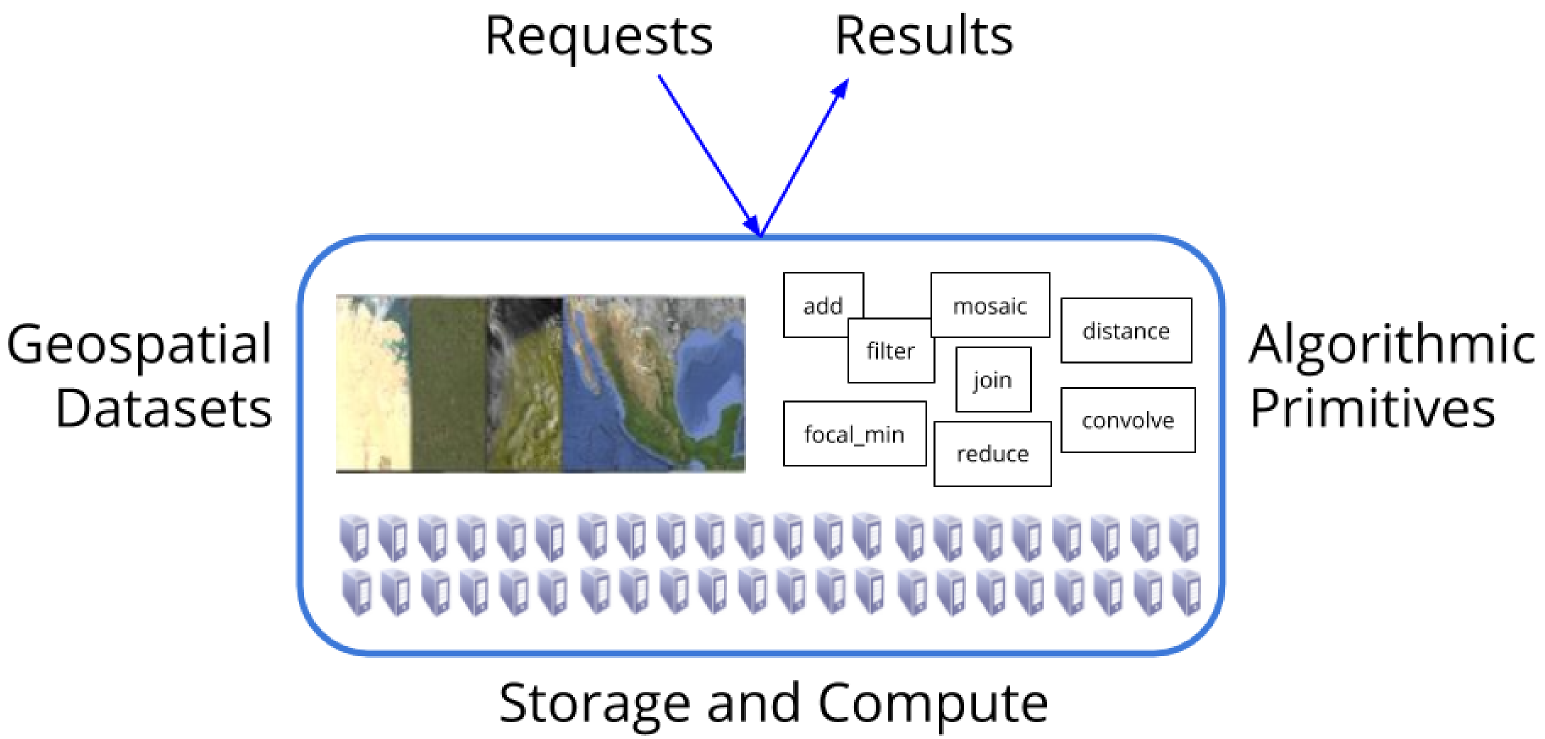
> 5 million images

> 5 petabytes of data

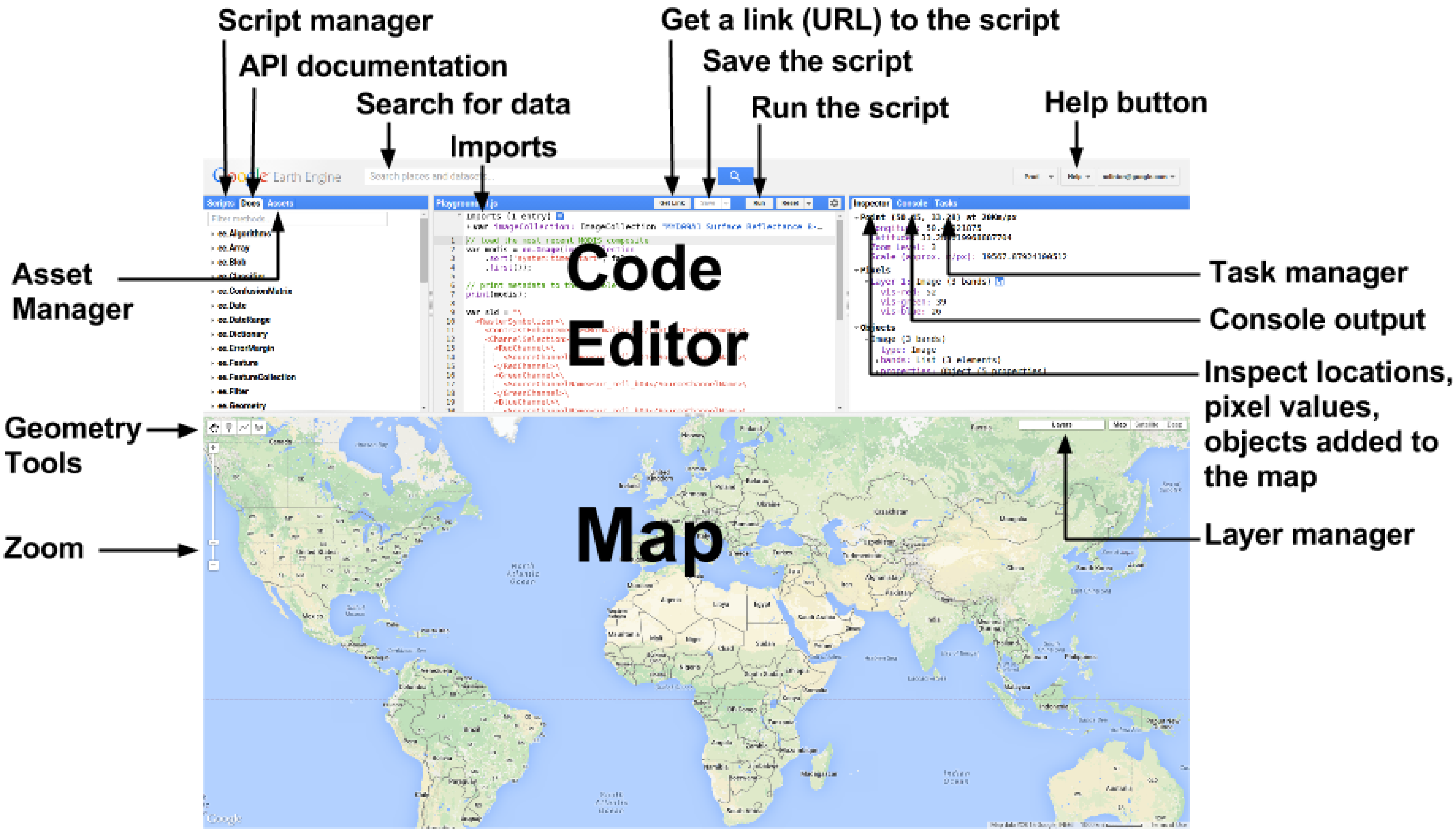
Google Earth Engine

https://scienceparkstudygroup.github.io/GoogleEarthEngine/

Session 1: understanding gradients in GEE

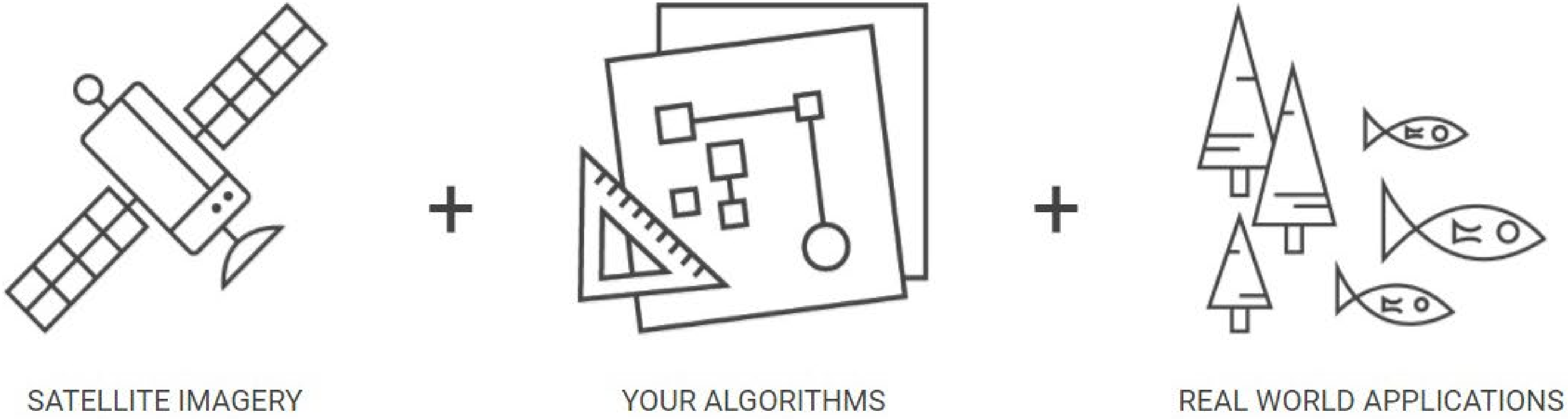


Session 1: understanding gradients in GEE

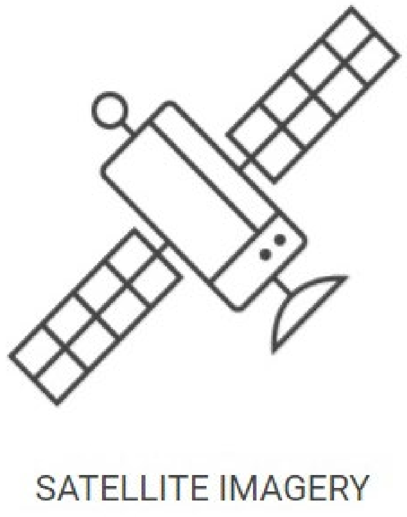


- •The Code Editor the place for accessing GEE data catalog and conducting geospatial analysis.
- •You can use the code editor to develop, share and manage scripts.
- •You can use the code editor to import, export, share and manage your own personal raster and vector datasets
- •Use the Docs tab, Example scripts and the HELP button to access the User Guides and Help Forum

Exercise 1: exploring GEE functionalities

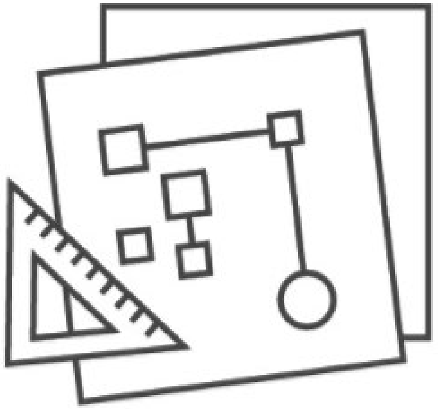


Exercise 1: exploring GEE functionalities



- Digital Elevation Models
- (multi)spectral Images (e.g. SENTINEL-2, LANDSAT)
- Radar images (e.g. SENTINEL-1)
- Thematic maps :
  - Land cover
  - Population maps
  - Watershed delineations
  - …
- Other raster products:
  - Derivatives of spectral information (e.g. vegetation indices)
  - Air pollution indices
  - Water extent layers
  - Fire extent products
  - Climate variables
  - ….

Exercise 1: exploring GEE functionalities



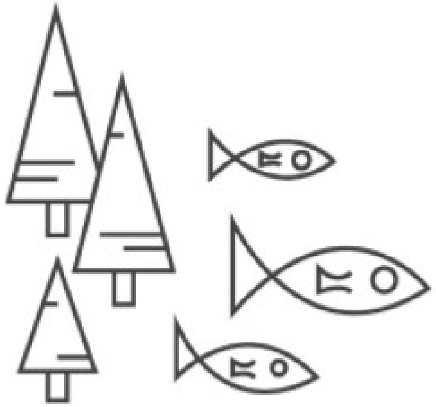
YOUR ALGORITHMS

- Data types:
  - Strings var woordje = ‘Hello world’;
  - Numbers var cijfertje = 42;
  - Lists var lijstcijfers = [0,1,2,99]; → lijstcijfer(3) ?
  - Objects var object = {

    cursus : ‘world food and ecosystems’,
    aantal: 72,
    motivatie: ‘zeer hoog’
    }; → object.aantal ?
- Functions
  - User-defined var zegwatikzeg = function(watikzeg) { return ‘ik heb’+ watikzeg + ‘gezegd’} → zegwatikzeg(woordje) ?
  - Built-in (out of the box) var tweeenveertig = ee.Algorithms.String(42);

Exercise 1: exploring GEE functionalities

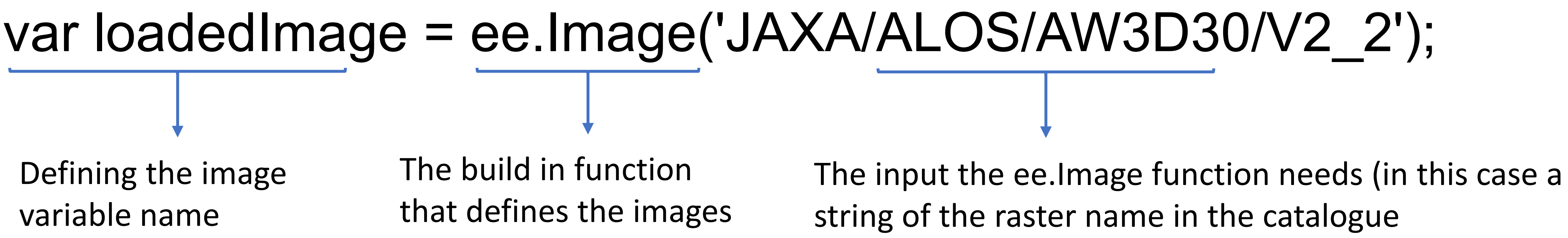
- Visualisation:
  - E.g. exploring forest fire extents visually in near real time
- Export of data
  - E.g. downloading GEE product to upload in your own GIS database
- Manipulation of data:
  - Make simple statistics (e.g. per country) on base layer (e.g. hectares of forest in the Netherlands)
  - Building land cover classifications
  - Building regression models to understand relationship soil moisture – fire susceptibility
  - ...



REAL WORLD APPLICATIONS

Exercise 2: plotting a raster file

Images: the name google earth engine gives to simple raster files.

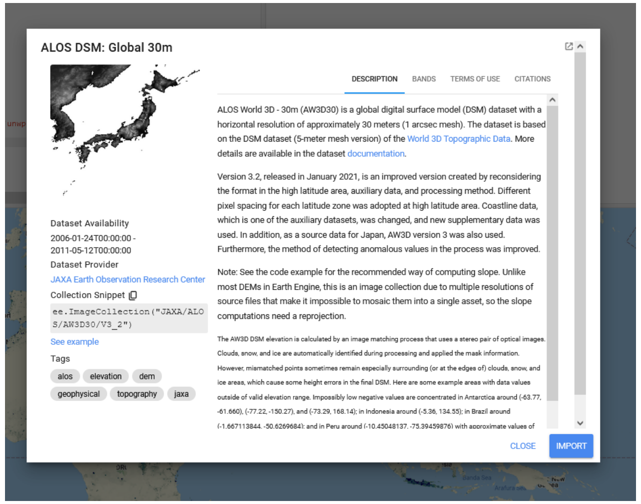


```
var loadedImage = ee.Image('JAXA/ALOS/AW3D30/V2_2');
```

Defining the image variable name

The build in function that defines the images

The input the ee.Image function needs (in this case a string of the raster name in the catalogue



ALOS DSM: Global 30m

Dataset Availability

2006-01-24T00:00:00 -
2011-05-12T00:00:00

Dataset Provider

JAXA Earth Observation Research Center

Collection Snippet

```
ee.ImageCollection("JAXA/ALOS/AW3D30/V3_2")
```

See example

Tags

alos elevation dem geophysical topography jaxa

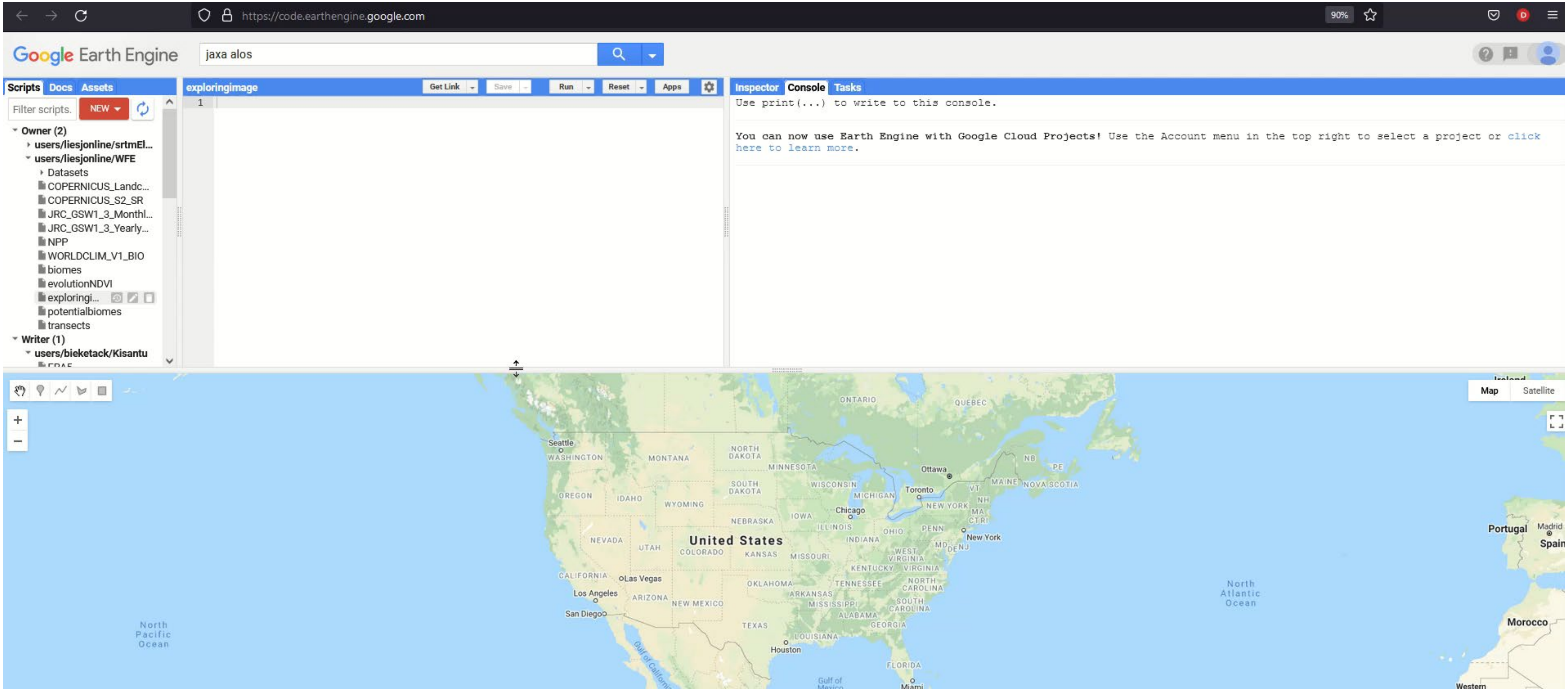
ALOS World 3D - 30m (AW3D30) is a global digital surface model (DSM) dataset with a horizontal resolution of approximately 30 meters (1 arcsec mesh). The dataset is based on the DSM dataset (5-meter mesh version) of the World 3D Topographic Data. More details are available in the dataset documentation.

Version 3.2, released in January 2021, is an improved version created by reconsidering the format in the high latitude area, auxiliary data, and processing method. Different pixel spacing for each latitude zone was adopted at high latitude area. Coastline data, which is one of the auxiliary datasets, was changed, and new supplementary data was used. In addition, as a source data for Japan, AW3D version 3 was also used. Furthermore, the method of detecting anomalous values in the process was improved.

Note: See the code example for the recommended way of computing slope. Unlike most DEMs in Earth Engine, this is an image collection due to multiple resolutions of source files that make it impossible to mosaic them into a single asset, so the slope computations need a reprojection.

The AW3D DSM elevation is calculated by an image matching process that uses a stereo pair of optical images. Clouds, snow, and ice are automatically identified during processing and applied the mask information. However, mismatched points sometimes remain especially surrounding (or at the edges of) clouds, snow, and ice areas, which cause some height errors in the final DSM. Here are some example areas with data values outside of valid elevation range. Impossibly low negative values are concentrated in Antarctica around (-63.77, -61.660), (-77.22, -150.27), and (-73.29, 168.14); in Indonesia around (-5.36, 134.55); in Brazil around (-1.667113844, -50.6269684); and in Peru around (-10.45048137, -75.39459876) with approximate values of

Exercise 2: plotting a raster file



```
Use print(...) to write to this console.

You can now use Earth Engine with Google Cloud Projects! Use the Account menu in the top right to select a project or click
here to learn more.
```

Exercise 2: plotting a raster file

ImageCollection→ the name GEE gives to a collection of images (e.g. timeseries).

➔ how to get image out of collection? FILTERS

```
var first = ee.ImageCollection('COPERNICUS/S2_SR')
                .filterBounds(ee.Geometry.Point(-70.48, 43.3631))
                .filterDate('2019-01-01', '2019-12-31')
                .sort('CLOUDY_PIXEL_PERCENTAGE')
                .first();
```

Exercise 2: plotting a raster file

ImageCollection→ the name GEE gives to a collection of images (e.g. timeseries).

➔ how to get image out of collection? FILTERS

```
var first = ee.ImageCollection('COPERNICUS/S2_SR')
                .filterBounds(ee.Geometry.Point(-70.48, 43.3631))
                .filterDate('2019-01-01', '2019-12-31')
                .sort('CLOUDY_PIXEL_PERCENTAGE')
                .first();
```

Exercise 2: plotting a raster file

ImageCollection→ the name GEE gives to a collection of images (e.g. timeseries).

➔ how to get image out of collection? FILTERS

```
var first = ee.ImageCollection('COPERNICUS/S2_SR')
                .filterBounds(ee.Geometry.Point(-70.48, 43.3631))
                .filterDate('2019-01-01', '2019-12-31')
                .sort('CLOUDY_PIXEL_PERCENTAGE')
                .first();
```

Exercise 2: plotting a raster file

ImageCollection→ the name GEE gives to a collection of images (e.g. timeseries).

➔ how to get image out of collection? FILTERS

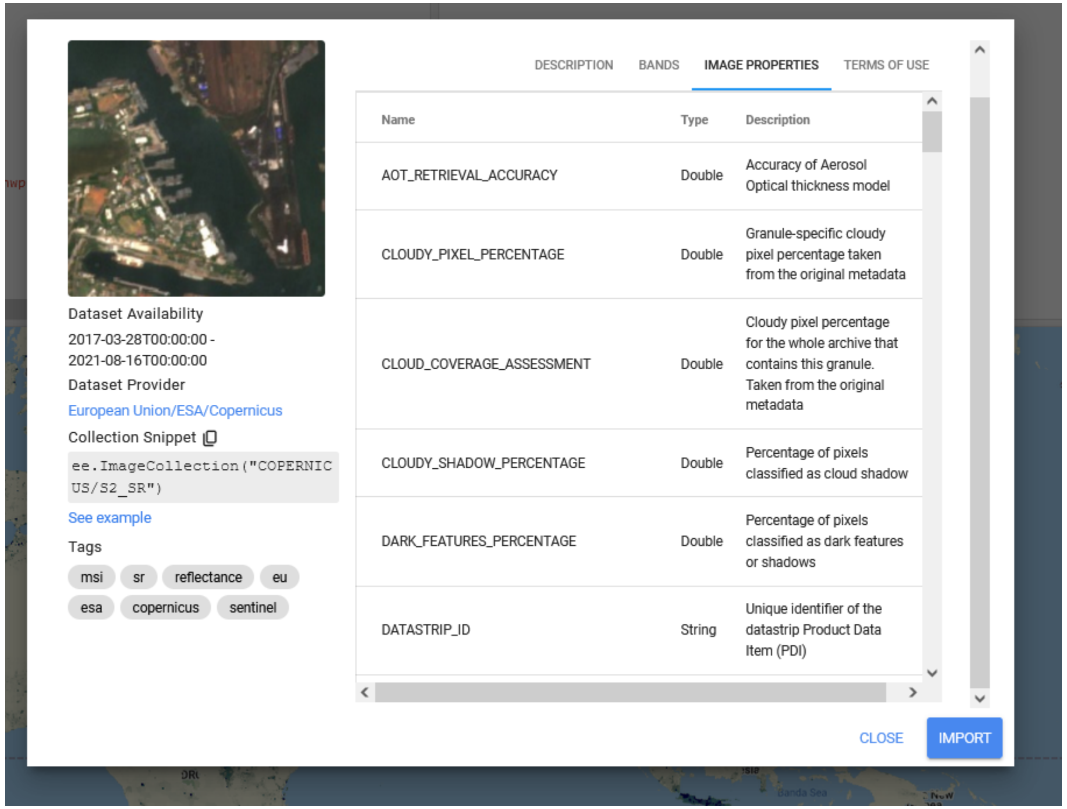
```
var first = ee.ImageCollection('COPERNICUS/S2_SR')
            .filterBounds(ee.Geometry.Point(-70.48, 43.3631))
            .filterDate('2019-01-01', '2019-12-31')
            .sort('CLOUDY_PIXEL_PERCENTAGE')
            .first();
```

Exercise 2: plotting a raster file

ImageCollection→ the name GEE gives to a collection of images (e.g. timeseries).

➔ how to get image out of collection? FILTERS

```
var first = ee.ImageCollection('COPERNICUS/S2_SR')
               .filterBounds(ee.Geometry.Point(-70.48, 43.3631))
               .filterDate('2019-01-01', '2019-12-31')
               .sort('CLOUDY_PIXEL_PERCENTAGE')
               .first();
```



Dataset Availability

2017-03-28T00:00:00 - 2021-08-16T00:00:00

Dataset Provider

European Union/ESA/Copernicus

Collection Snippet

```
ee.ImageCollection("COPERNICUS/S2_SR")
```

See example

Tags

msi sr reflectance eu esa copernicus sentinel

Name	Type	Description
AOT_RETRIEVAL_ACCURACY	Double	Accuracy of Aerosol Optical thickness model
CLOUDY_PIXEL_PERCENTAGE	Double	Granule-specific cloudy pixel percentage taken from the original metadata
CLOUD_COVERAGE_ASSESSMENT	Double	Cloudy pixel percentage for the whole archive that contains this granule. Taken from the original metadata
CLOUDY_SHADOW_PERCENTAGE	Double	Percentage of pixels classified as cloud shadow
DARK_FEATURES_PERCENTAGE	Double	Percentage of pixels classified as dark features or shadows
DATASTRIP_ID	String	Unique identifier of the datastrip Product Data Item (PDI)

Exercise 2: plotting a raster file

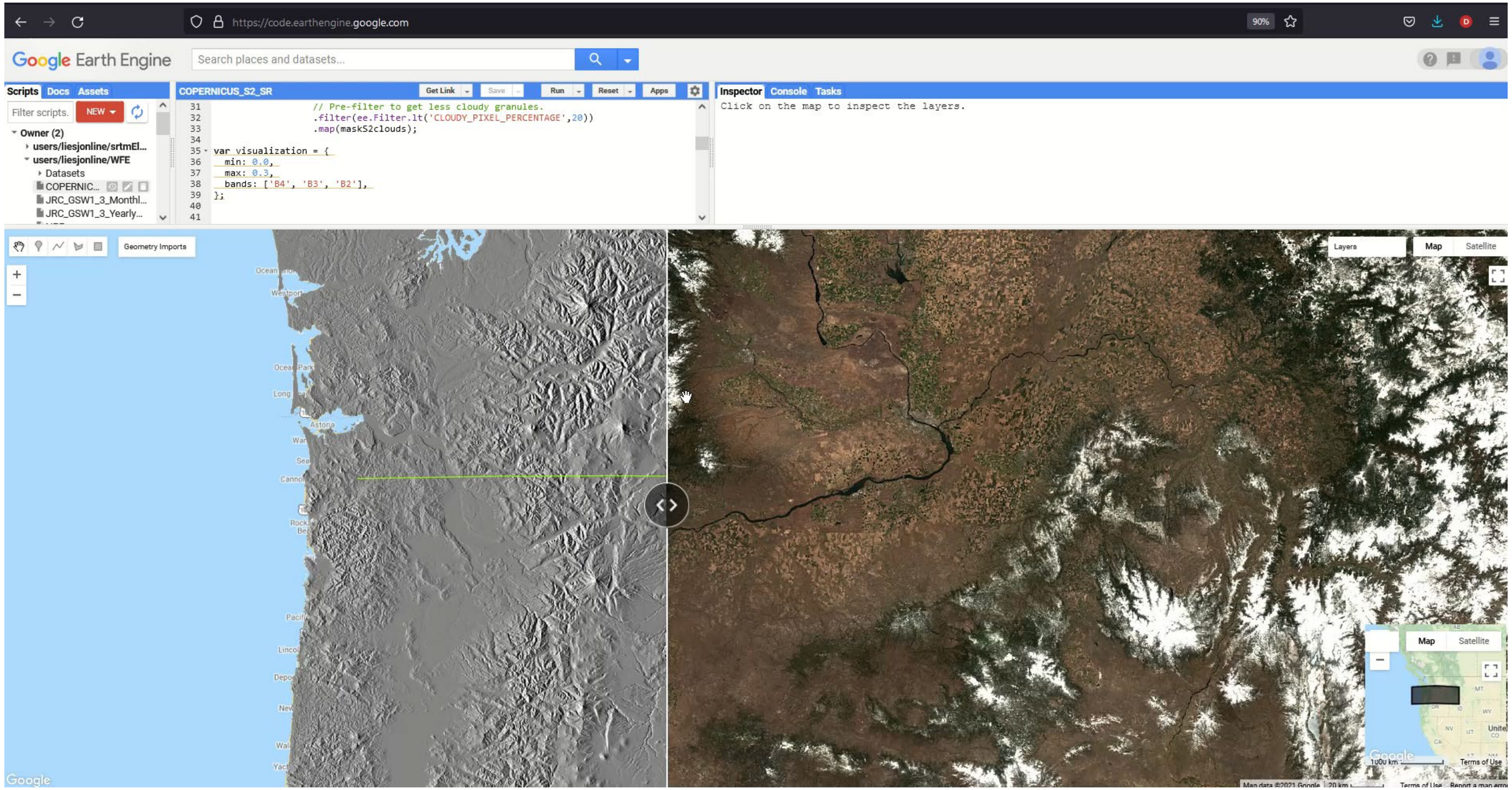
ImageCollection→ the name GEE gives to a collection of images (e.g. timeseries).

➔ how to get image out of collection? FILTERS

```
var first = ee.ImageCollection('COPERNICUS/S2_SR')
            .filterBounds(ee.Geometry.Point(-70.48, 43.3631))
            .filterDate('2019-01-01', '2019-12-31')
            .sort('CLOUDY_PIXEL_PERCENTAGE')
            .first();
```

How to visualize?
See GITHUB instructions

Exercise 3: understanding a gradient

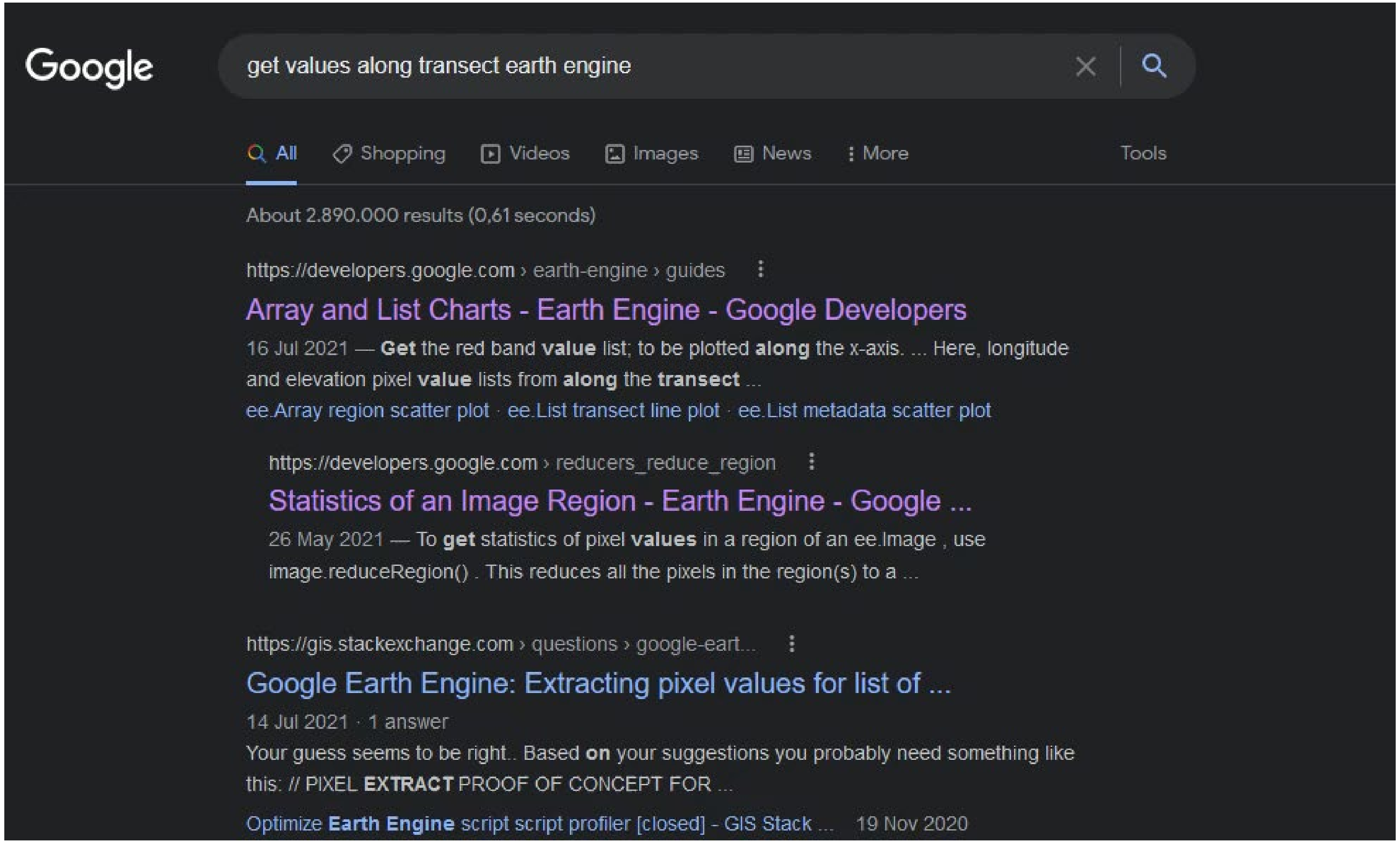


```
                  // Pre-filter to get less cloudy granules.
                  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',20))
                  .map(maskS2clouds);

var visualization = {
  min: 0.0,
  max: 0.3,
  bands: ['B4', 'B3', 'B2'],
};
```

```
Click on the map to inspect the layers.
```

Exercise 3: understanding a gradient



get values along transect earth engine

About 2.890.000 results (0,61 seconds)

https://developers.google.com › earth-engine › guides

Array and List Charts - Earth Engine - Google Developers

16 Jul 2021 — **Get** the red band **value** list; to be plotted **along** the x-axis. ... Here, longitude and elevation pixel **value** lists from **along** the **transect** ...

ee.Array region scatter plot · ee.List transect line plot · ee.List metadata scatter plot

https://developers.google.com › reducers_reduce_region

Statistics of an Image Region - Earth Engine - Google ...

26 May 2021 — To **get** statistics of pixel **values** in a region of an ee.Image , use image.reduceRegion() . This reduces all the pixels in the region(s) to a ...

https://gis.stackexchange.com › questions › google-eart...

Google Earth Engine: Extracting pixel values for list of ...

14 Jul 2021 · 1 answer

Your guess seems to be right.. Based **on** your suggestions you probably need something like this: // PIXEL **EXTRACT** PROOF OF CONCEPT FOR ...

Optimize **Earth Engine** script script profiler [closed] - GIS Stack ... 19 Nov 2020

END GOAL OF THIS SESSION

Understanding and recognizing the basic structure of

- Variable definitions
- Function calling

Knowing where to find information on

- Datasets
- Functionalities

Understanding the capacity of GEE to

- Visualize data
- Extract information
- Plot multiple dimensions

NOT AN END GOAL:

- Reproducing all the code
- Knowing by heart all the functions and their input
- Knowing by heart the available datasets
- …