ILR
Institute for
Food and
Resource Economics

UNIVERSITÄT BONN

SATELLITE DATA IN AGRICULTURAL AND ENVIRONMENTAL ECONOMICS

DAVID WUEPPER, HADI, WYCLIFE AGUMBA OLUOCH

Land Economics Group

UNIVERSITÄT BONN

ILR
Institute for
Food and
Resource Economics

Land Economics Group

- https://bit.ly/gee-bonn-2025

WORKFLOWS

Hadi

UNIVERSITÄT BONN
ILR Institute for Food and Resource Economics
Land Economics Group

INTRO TO EARTH ENGINE

UNIVERSITÄT BONN
ILR Institute for Food and Resource Economics
Land Economics Group

Google Earth Engine

+

+

Data

Compute

API

UNIVERSITÄT BONN

ILR Institute for Food and Resource Economics

Land Economics Group

EARTH ENGINE PUBLIC DATA CATALOG

Landsat & Sentinel
10-30m, weekly

MODIS
250m daily

Vector Data
WDPA, TIGER, WHC

Terrain & Land Cover

Weather & Climate
NOAA NCEP, OMI, ...

... and upload your own vectors and rasters

1000+ public datasets

100+ datasets added yearly

100+ petabytes (PB) of data

1+ PB of new data every month

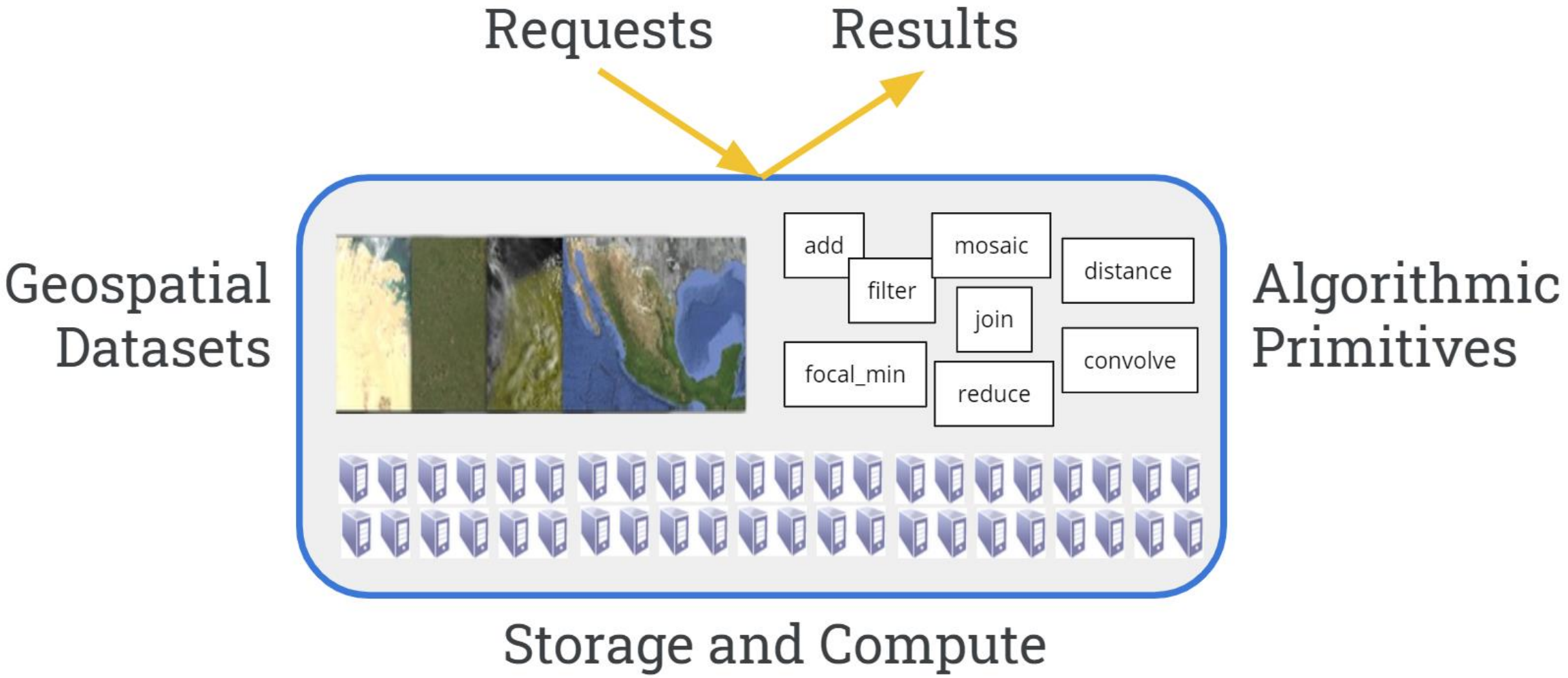
UNIVERSITÄT BONN

ILR Institute for Food and Resource Economics

Land Economics Group

https://developers.google.com/earth-engine/datasets/

Computation



API: Requests and Results

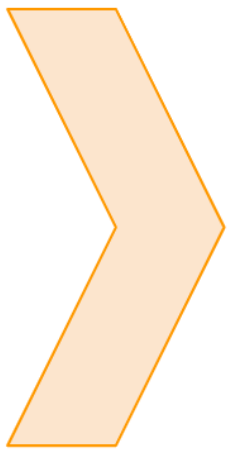
Requests

Compute Value

Generate Map

Fetch Map Tile

Perform Import/Export



Results

Numbers, Tables, other Data Structures

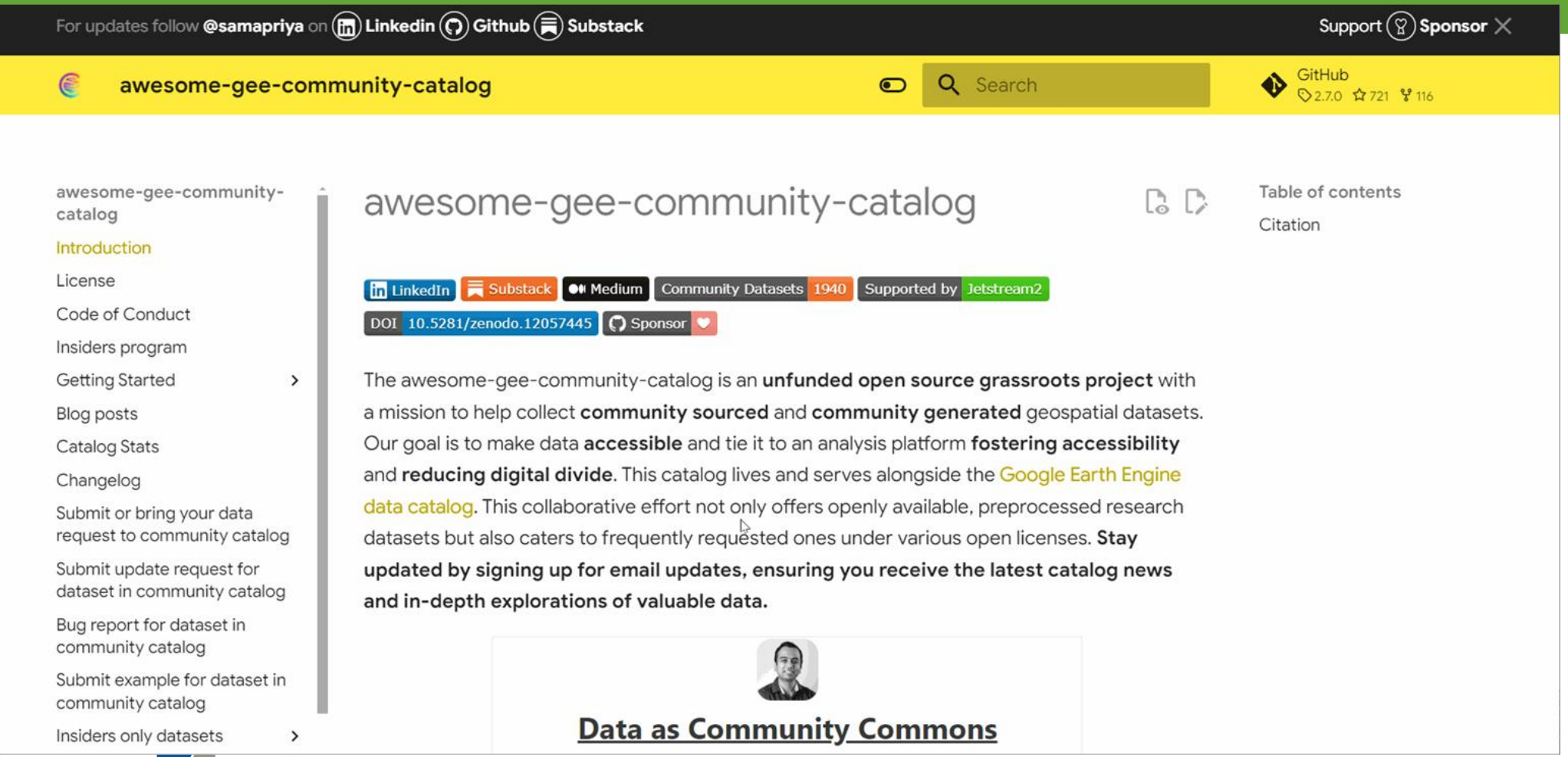
Map ID

Image

"OK"

"Error: ..."

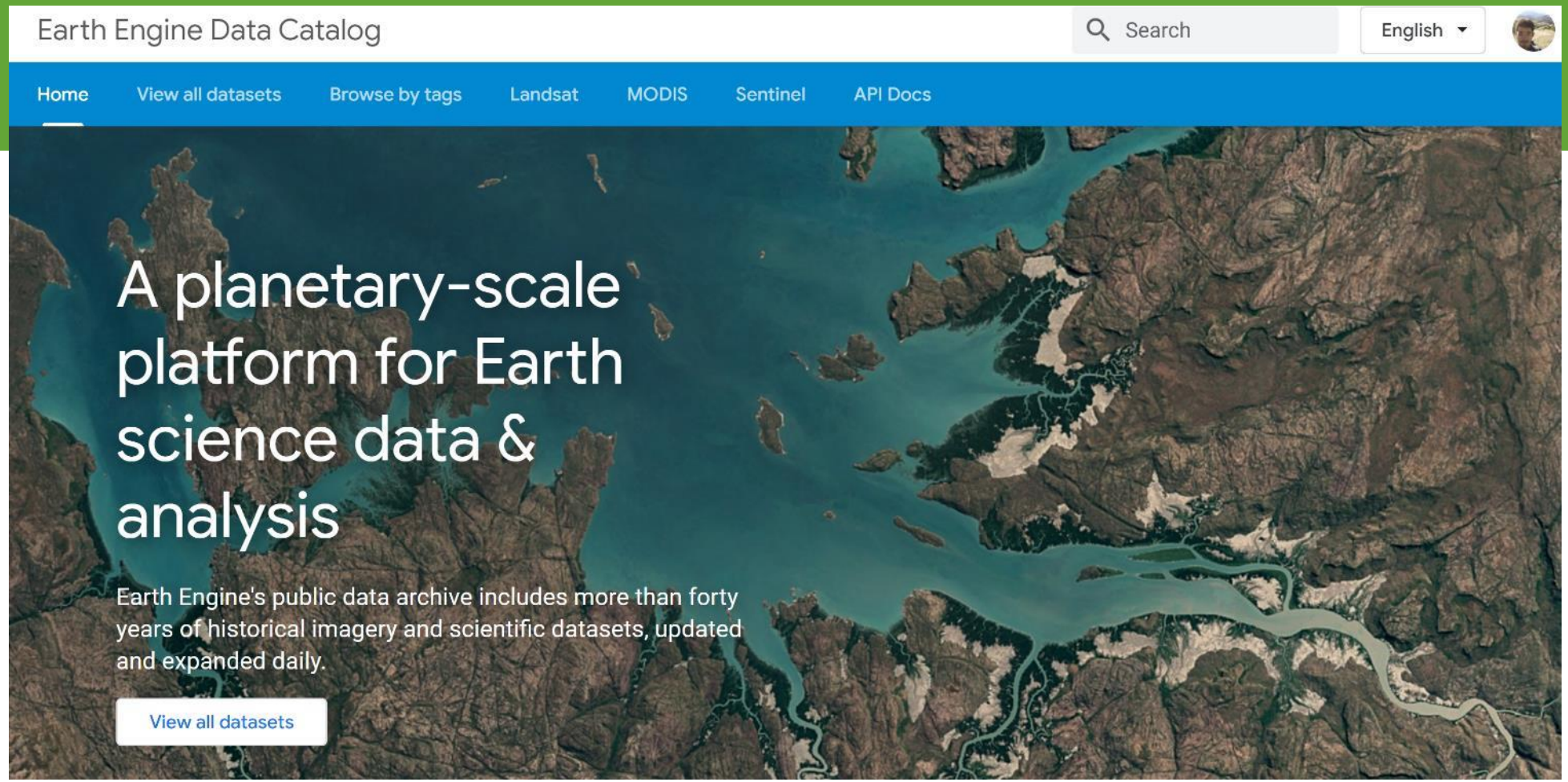
EARTH ENGINE COMMUNITY CATALOG



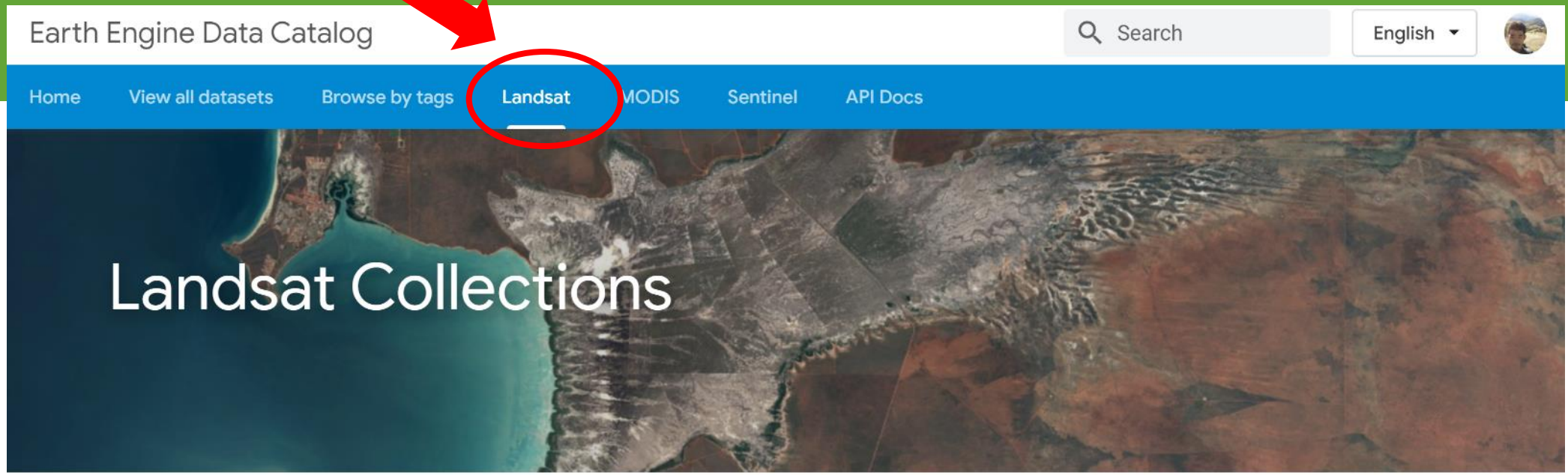
1,929 datasets

375+ TB of data

https://gee-community-catalog.org



developers.google.com/earth-engine/datasets



Landsat, a joint program of the USGS and NASA, has been observing the Earth continuously from 1972 through the present day. Today the Landsat satellites image the entire Earth's surface at a 30-meter resolution about once every two weeks, including multispectral and thermal data.

Landsat data is available in Earth Engine in its raw form, as Surface Reflectance, TOA-corrected reflectance, and in various ready-to-use computed products such as NDVI and EVI vegetation indices.

USGS science for a changing world

NASA

Earth Engine Data Catalog

Search

English

Home | View all datasets | Browse by tags | Landsat | MODIS | Sentinel | API Docs

Landsat 8

2013 - Present

Landsat 7

1999 - Present

Landsat 5

1984 – 2012

Earth Engine Data Catalog

Search

English

more about the different tiers.

Surface Reflectance

Atmospherically corrected surface reflectance from the Landsat 8 OLI/TIRS sensors

Dataset Availability: April 2013 - Present

Tier 1 | Tier 2

Top of Atmosphere

Landsat 8 Collection 1 calibrated top-of-atmosphere (TOA) reflectance

Dataset Availability: April 2013 - Present

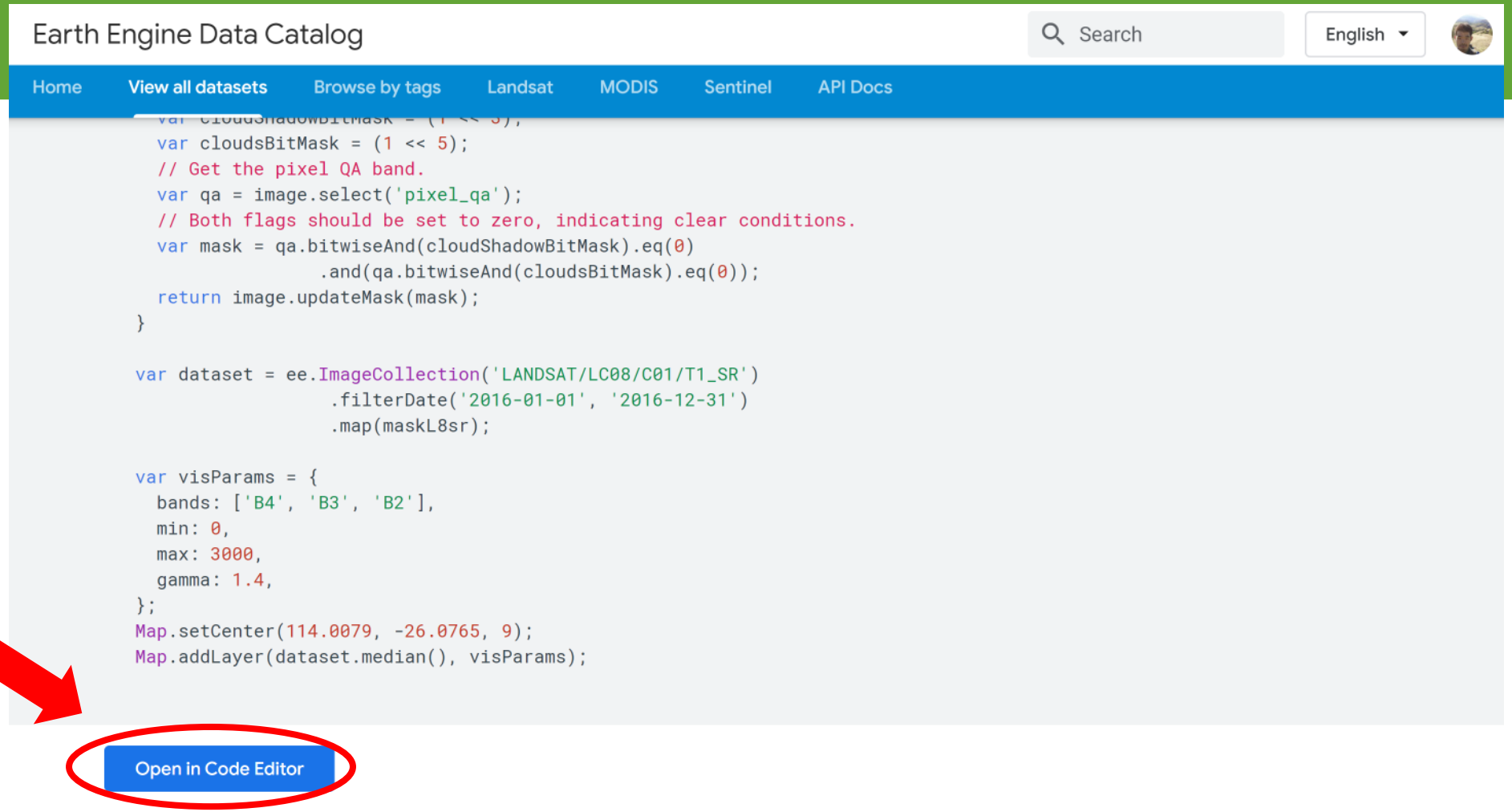
Tier 1 | T1 + Real-Time | Tier 2

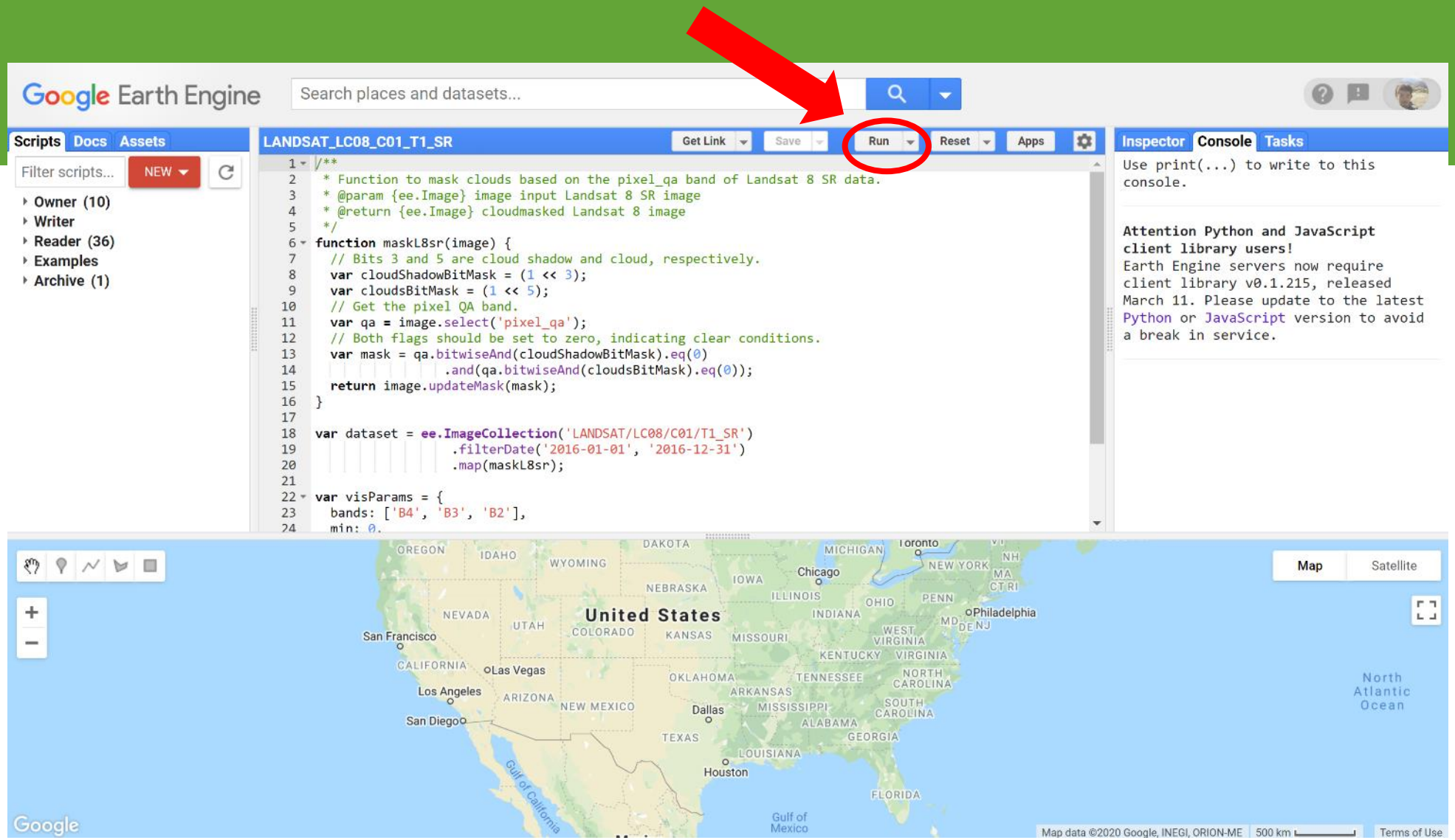
Raw Images

Landsat 8 Collection 1 DN values, representing scaled, calibrated at-sensor radiance.

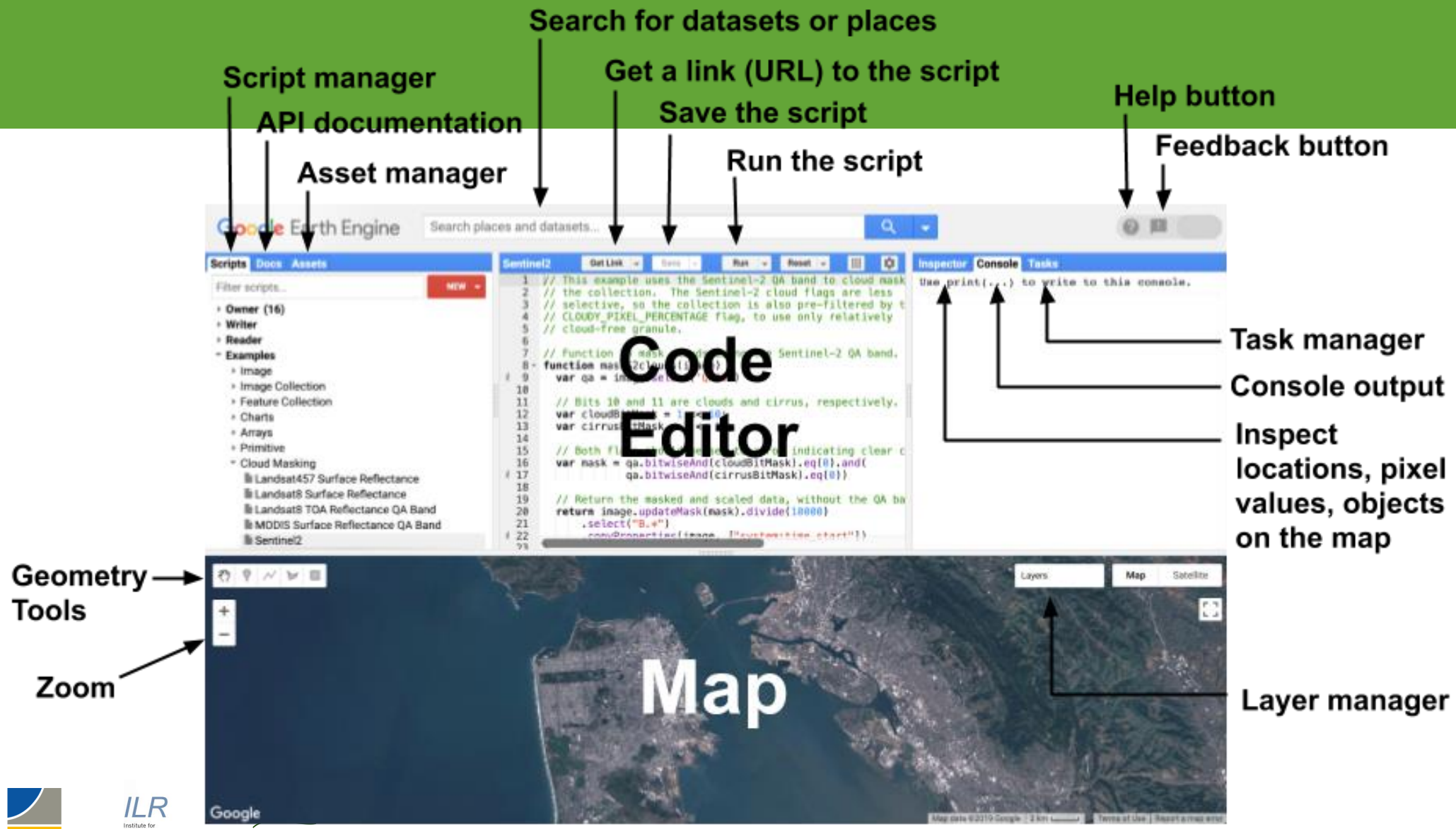
Dataset Availability: April 2013 - Present

Tier 1 | T1 + Real-Time | Tier 2


Earth Engine Data Catalog
Search
English
Home
View all datasets
Browse by tags
Landsat
MODIS
Sentinel
API Docs
var cloudsBitMask = (1 << 5);
// Get the pixel QA band.
var qa = image.select('pixel_qa');
// Both flags should be set to zero, indicating clear conditions.
var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
.and(qa.bitwiseAnd(cloudsBitMask).eq(0));
return image.updateMask(mask);
}
var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')
.filterDate('2016-01-01', '2016-12-31')
.map(maskL8sr);
var visParams = {
bands: ['B4', 'B3', 'B2'],
min: 0,
max: 3000,
gamma: 1.4,
};
Map.setCenter(114.0079, -26.0765, 9);
Map.addLayer(dataset.median(), visParams);
Open in Code Editor


Google Earth Engine
Search places and datasets...
Scripts
Docs
Assets
Filter scripts...
NEW
Owner (10)
Writer
Reader (36)
Examples
Archive (1)
LANDSAT_LC08_C01_T1_SR
Get Link
Save
Run
Reset
Apps
Inspector
Console
Tasks
Use print(...) to write to this console.
Attention Python and JavaScript client library users!
Earth Engine servers now require client library v0.1.215, released March 11. Please update to the latest Python or JavaScript version to avoid a break in service.
/**
* Function to mask clouds based on the pixel_qa band of Landsat 8 SR data.
* @param {ee.Image} image input Landsat 8 SR image
* @return {ee.Image} cloudmasked Landsat 8 image
*/
function maskL8sr(image) {
// Bits 3 and 5 are cloud shadow and cloud, respectively.
var cloudShadowBitMask = (1 << 3);
var cloudsBitMask = (1 << 5);
// Get the pixel QA band.
var qa = image.select('pixel_qa');
// Both flags should be set to zero, indicating clear conditions.
var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
.and(qa.bitwiseAnd(cloudsBitMask).eq(0));
return image.updateMask(mask);
}
var dataset = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')
.filterDate('2016-01-01', '2016-12-31')
.map(maskL8sr);
var visParams = {
bands: ['B4', 'B3', 'B2'],
min: 0,
Map
Satellite
United States
San Francisco
Los Angeles
San Diego
Las Vegas
Chicago
Dallas
Houston
Philadelphia
Toronto
North Atlantic Ocean
Gulf of Mexico
Google
Map data ©2020 Google, INEGI, ORION-ME
500 km
Terms of Use

CODE EDITOR (JAVASCRIPT API)



Introduction to JavaScript

Definitions and rendering:

```
var variable = expression;
print(expression);
```

Data Structures:

List: `[expression, expression, ...]`

Dictionary: `{key1: value1, key2: value2, ...}`

Syntax highlighting and documentation.

Introduction to JavaScript: Functions

Pieces of computation to execute "later".

REUSE At multiple points in the script

DEFER Until we know the object(s) of interest

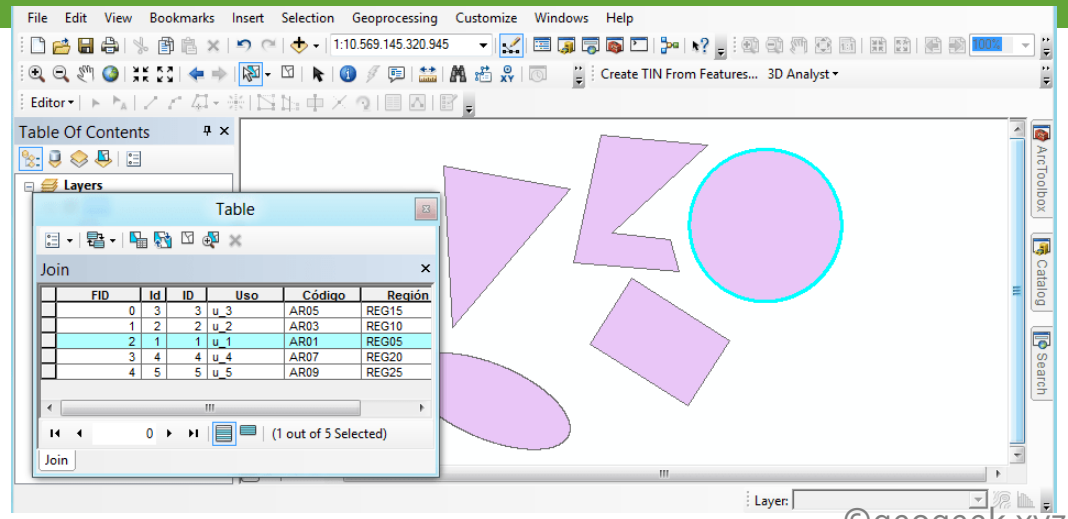
CALLBACK When something happens, eg click on map

```
function name(a, b) {
    // do stuff
    return expression;
}

// later ...

var result = name(x, y);
```

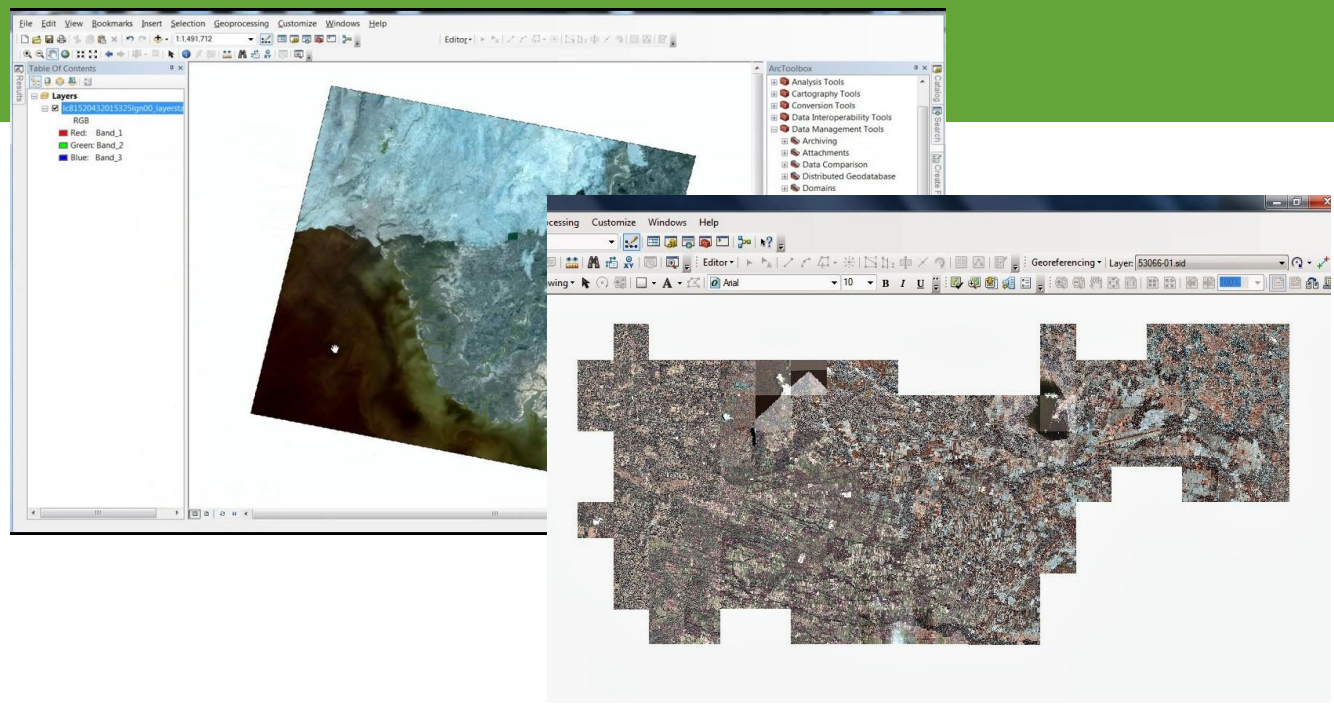
Examples:

- Predefined: Earth Engine Library
- Calculation: Normalised Difference
- Transform Image: Add Bands, Remove Clouds, Gather Statistics

DATA MODEL/STRUCTURE IN EARTH ENGINE



©geogeek.xyz



Arcmap: Shapefile

Arcmap: Raster (GeoTIFF)

Earth Engine:

- Feature (a point / a line / a polygon)
- Feature Collection (multiple features)

Earth Engine:

- Image (a 1-band raster / a multiband raster)
- Image Collection (multiple images)

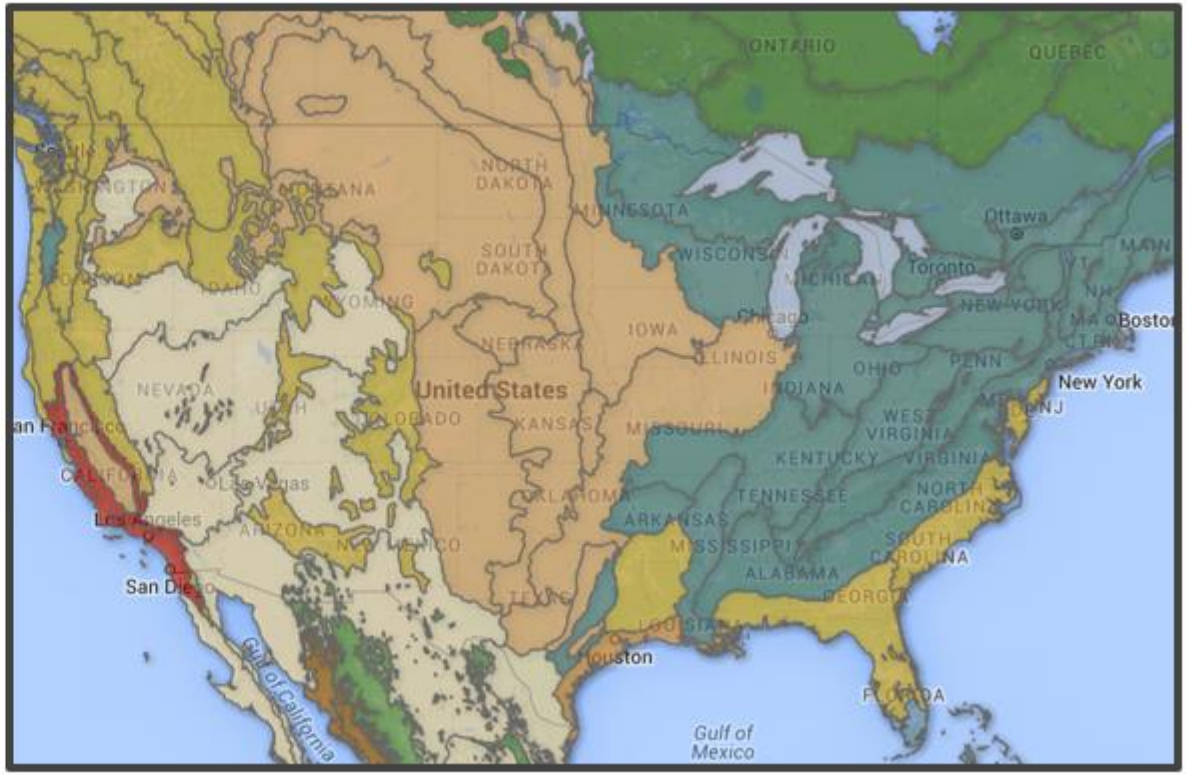
UNIVERSITÄT BONN ILR Institute for Food and Resource Economics Land Economics Group

DATA MODEL/STRUCTURE IN EARTH ENGINE

Feature

- A **Geometry-** Line / Point / Polygon
- List of **Properties-** Just like images

Basically a **row** in a **table** with a geometry (.geo) column



TNC Ecoregions

id	.geo	city	country	population	year
"my_feature"	Point(1,2)	Reno	USA	300,000	2018

DATA MODEL/STRUCTURE IN EARTH ENGINE

Feature

Image

Each image has

Bands

2D grid of pixels with a:

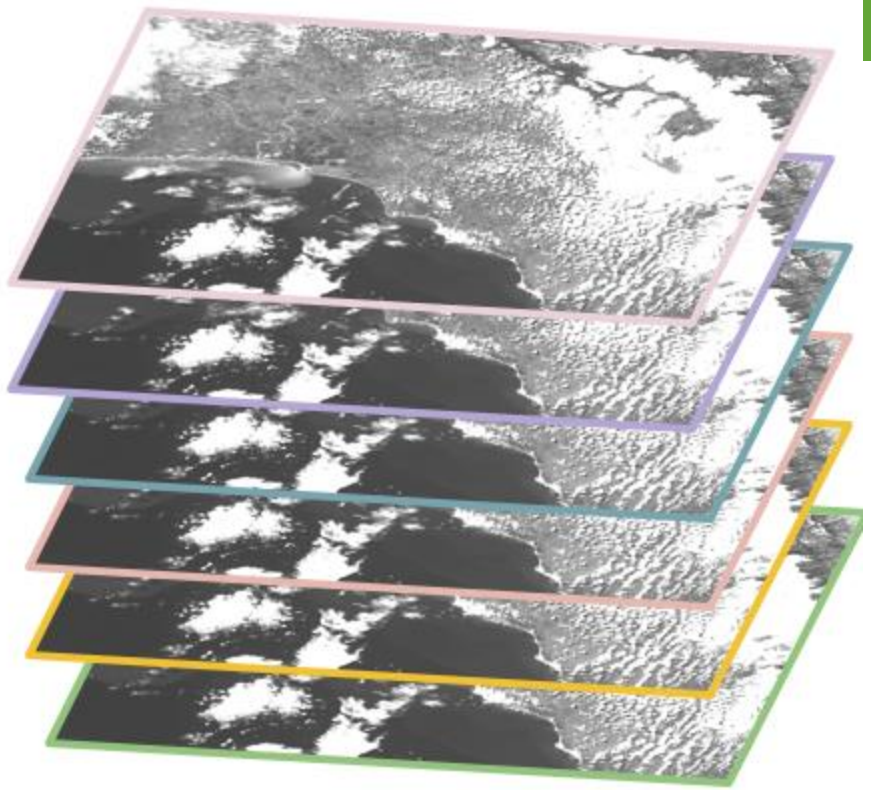
Name

CRS/Projection

Pixel Scale/Resolution

Properties, including:

Date, Bounding-box, unique ID



An Image containing 6 bands

DATA MODEL/STRUCTURE IN EARTH ENGINE

Feature

Image

Collection

Bag of Elements

Table of Features

Directory of Images

An image collection containing 10 images

DATA MODEL/STRUCTURE IN EARTH ENGINE

ImageCollection ID, metadata

Image ID, metadata ...

Band ...

ID *B1*

CRS *EPSG:32601*

CRS Transform *[a,b,c,d,e,f]*

Data Type *int16*

Dimensions *1830 x 1830*

DATA MODEL/STRUCTURE IN EARTH ENGINE

Vector / Tables

- Geometries
  - Point
  - LineString
  - LinearRing
  - Polygon
  - MultiPoint
  - MultiLineString
  - MultiPolygon
- Geometry Collections
  - Collection of geometries
- Features
  - Geometry + **attributes**
- Feature Collections
  - **Collection** of features

Raster / Image

Image

- Arbitrary number of bands + band designations
- Metadata properties

Image Collection

- **Collection** of images
- Can have collection-level metadata attributes

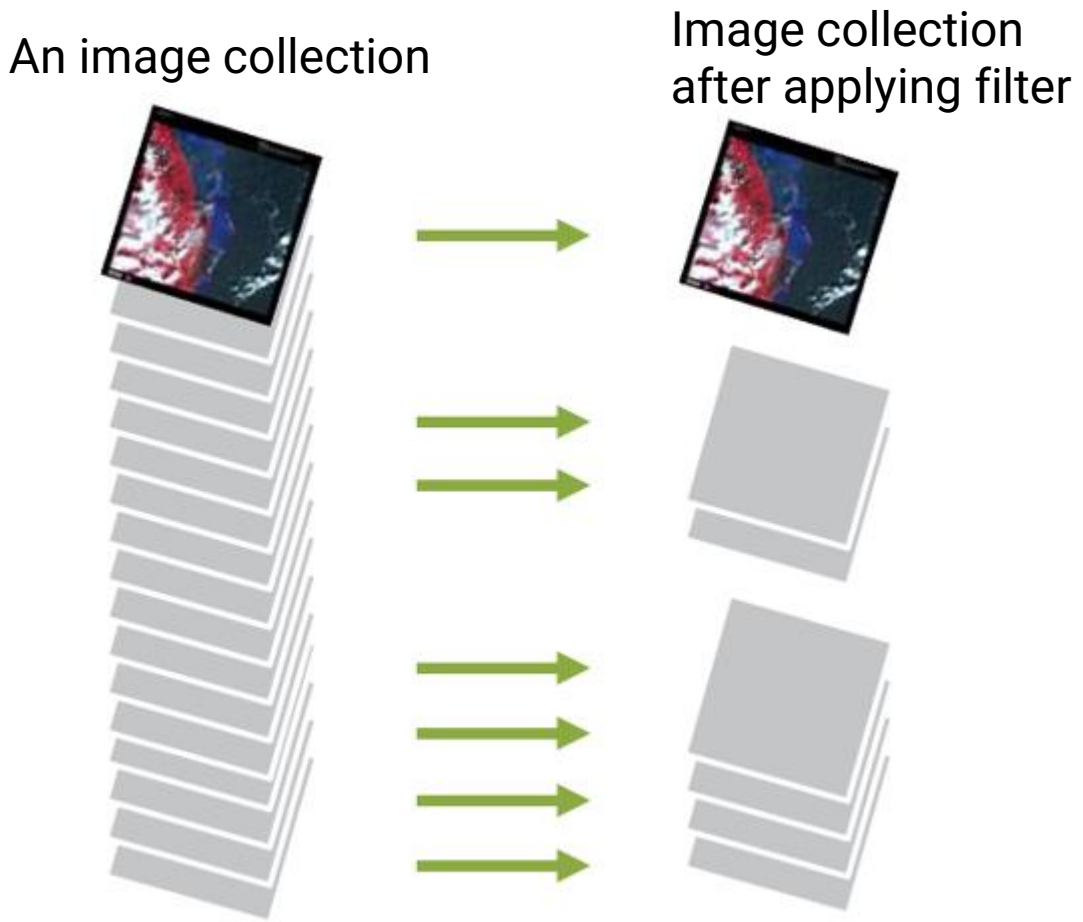
Source: Chris Herwig's slide

FILTER, MAP, REDUCE

ImageCollection.filterDates()
ImageCollection.filterBounds()
ImageCollection.filterMetadata()

- **Filter** a collection
- Time, Space, and Metadata Search

•**Time** (example: all Landsat scenes (images) from January to June 2020)
•**Space** (example: all Sentinel-1 scenes that *intersect* Germany country area)
•**Metadata** (example: all Sentinel-2 scenes with CLOUD_PERCENT < 10)

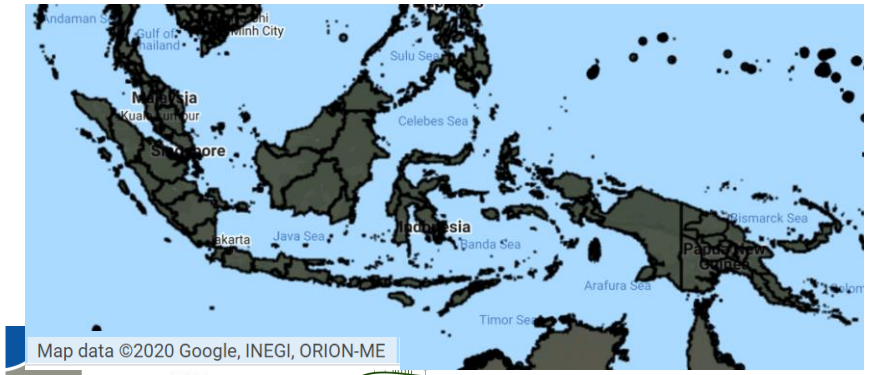


©David Gibson; Google

FILTER, MAP, REDUCE

FeatureCollection.filter()
FeatureCollection.filterBounds()

Filter bounds



Filter by "geometry area" property

Filter, **Map**, Reduce

Apply a function to *each* element of a collection

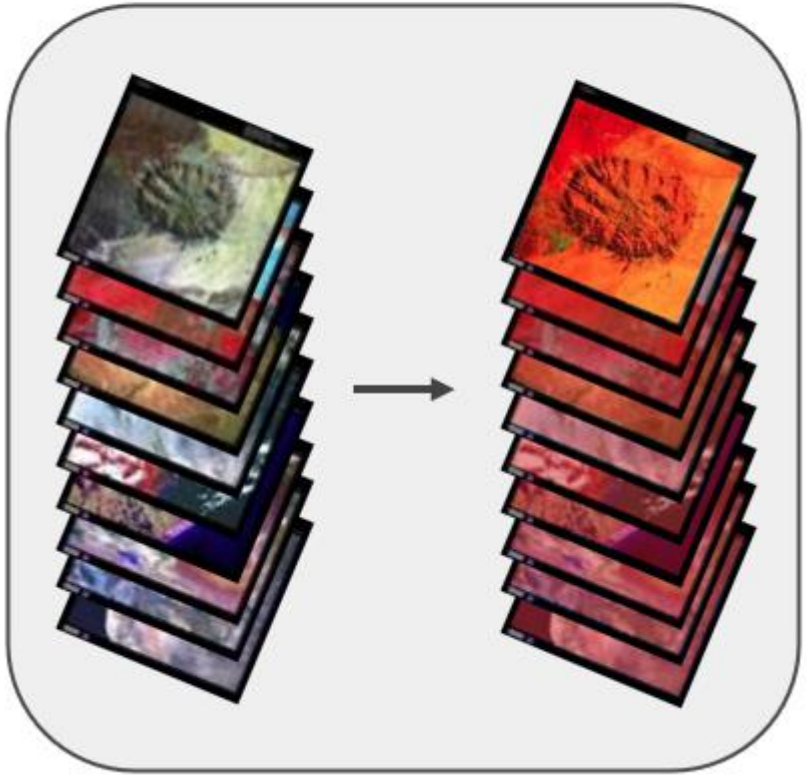
A "For Each" operation

Examples

Calculate the area (function to calculate area) of every province (feature) in country X (feature collection)

Mask the cloud (function to mask cloud) in each Landsat scene (image) in all Landsat scenes in 2023 that cover the city of Bonn (image collection)

Make an NDVI band [function to perform band math NDVI = (NIR-RED/NIR+RED)] for every Landsat scenes in 2023 that cover the city of Bonn (image collection)



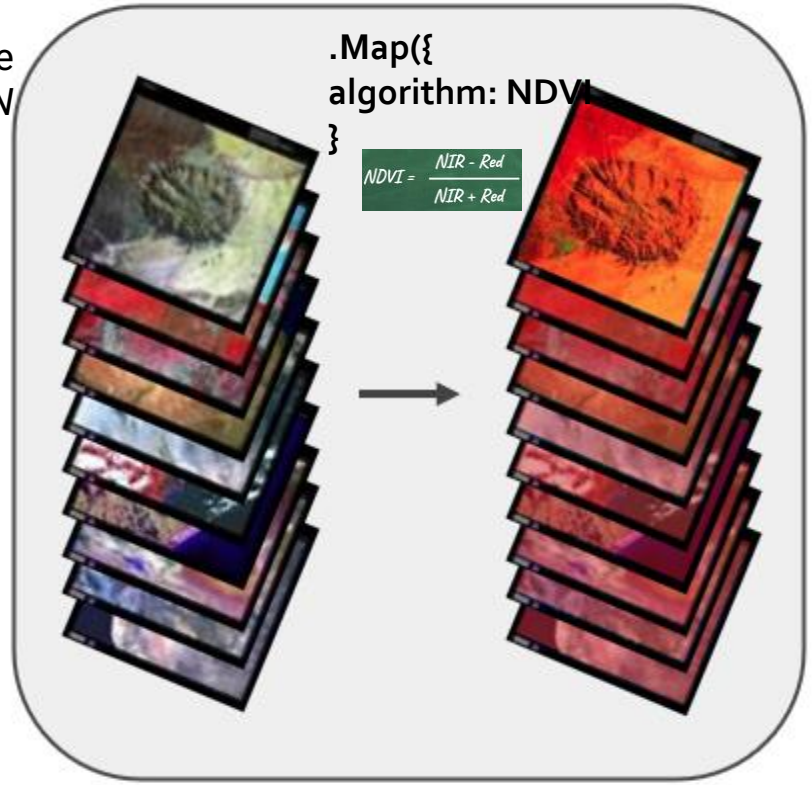
$N \rightarrow N$

Filter, **Map**, Reduce: : ImageCollection.map()

Example

Make an NDVI band [function to perform band math NDVI = (NIR-RED/NIR+RED)] for every Landsat scene in 2023 that covers the city of Bonn (image collection)

a Landsat image collection with *N* images (e.g. *N* dates). Each image has multiple bands.



a NDVI image collection with *N* images (e.g. *N* dates)

$N \rightarrow N$

Filter, Map, Reduce: FeatureCollection.map()


```
.map({
  algorithm: mybuffer
})

function mybuffer(feature){
  return feature.buffer(250)
}
```

Feature	Plot
1	25
2	34
3	99

.map(addSurveyorProperty)

function addSurveyorProperty(feature){…}

Feature	Plot	Surveyor
1	25	Rick
2	34	Morty
3	99	Justin

Filter, Map, **Reduce**

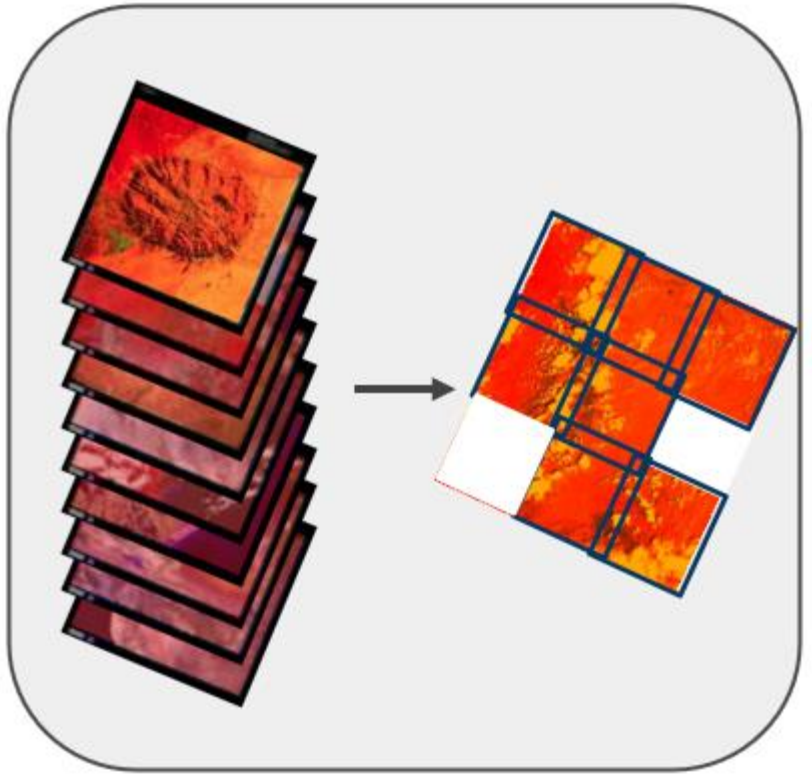
Apply a function to *everything* in a collection

"Aggregation"

"Combine a bunch of values"

Example

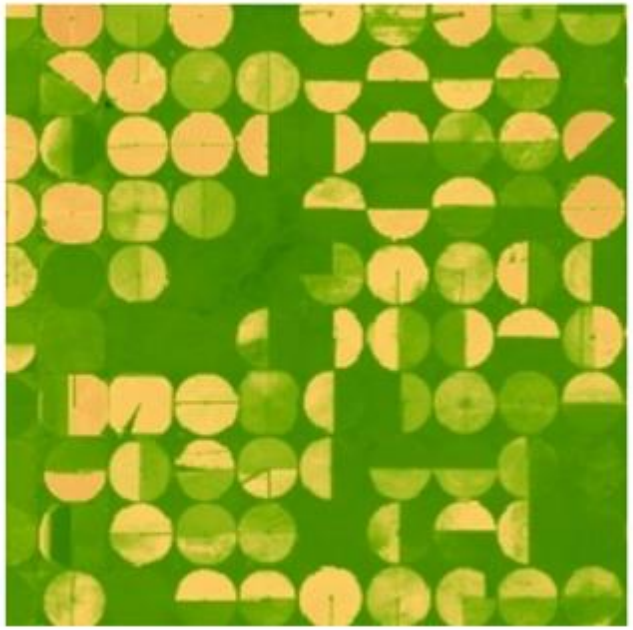
> Aggregate all Landsat scenes in 2019 (an image collection) to create a cloud-free composite (an image) by taking the median for every pixels.



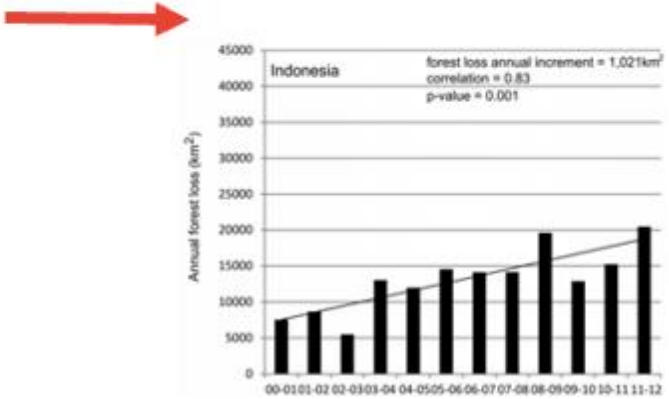
$N \rightarrow 1$ or $N \rightarrow M$

©Noel Gorelick; Google

Filter, Map, **Reduce**



Gabon	1891	391	11898
Lithuania	1845	1226	40296
Cuba	1725	2271	68008
Mali	1694	0	1247103
Costa Rica	1653	382	11327
Czech Republic	1646	1331	46934
South Sudan	1635	38	460581
North Korea	1605	137	67695
Italy	1603	898	201331



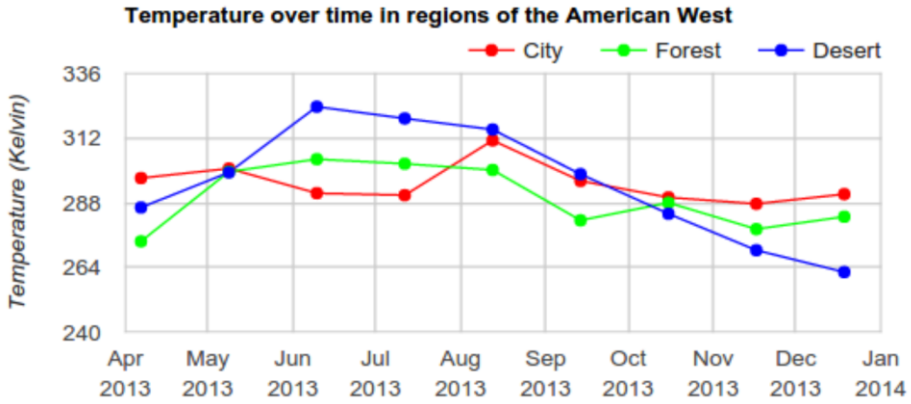


Figure 1. Time series of mean land surface temperature in three regions.

Filter, Map, **Reduce**

Where do the values come from?

Aggregate all Landsat scenes in 2019 (an image collection) to create a cloud-free composite (an image) by taking the median for every pixels.

Where do the results go?

How are the values combined?

Filter, Map, **Reduce**

How are the values combined?

Extract time series of the mean of NDVI
from a temporal stack (an image collection)
in several biomass measurement plots (a feature collection)

Where do the values come from?

Where do the results go?

Reducers galore

Where do the values come from?

Where do the results go?

How are the values combined?

8 ways to reduce

- Image.reduce
- Image.reduceNeighborhood
- Image.reduceRegion
- Image.reduceRegions
- Image.reduceToVectors
- ImageCollection.reduce
- FeatureCollection.reduceColumns
- FeatureCollection.ReduceToImage

40+ reducers

- Reducer.allNonZero
- Reducer.and
- Reducer.anyNonZero
- Reducer.count
- Reducer.countEvery
- Reducer.histogram
- Reducer.intervalMean
- Reducer.linearFit
- Reducer.linearRegression
- Reducer.max
- Reducer.mean
- Reducer.median
- Reducer.min
- Reducer.minMax
- Reducer.mode
- Reducer.or
- Reducer.percentile
- Reducer.product
- Reducer.sampleStdDev
- Reducer.sampleVariance
- Reducer.stdDev
- Reducer.sum
- Reducer.toCollection
- Reducer.toList
- Reducer.variance

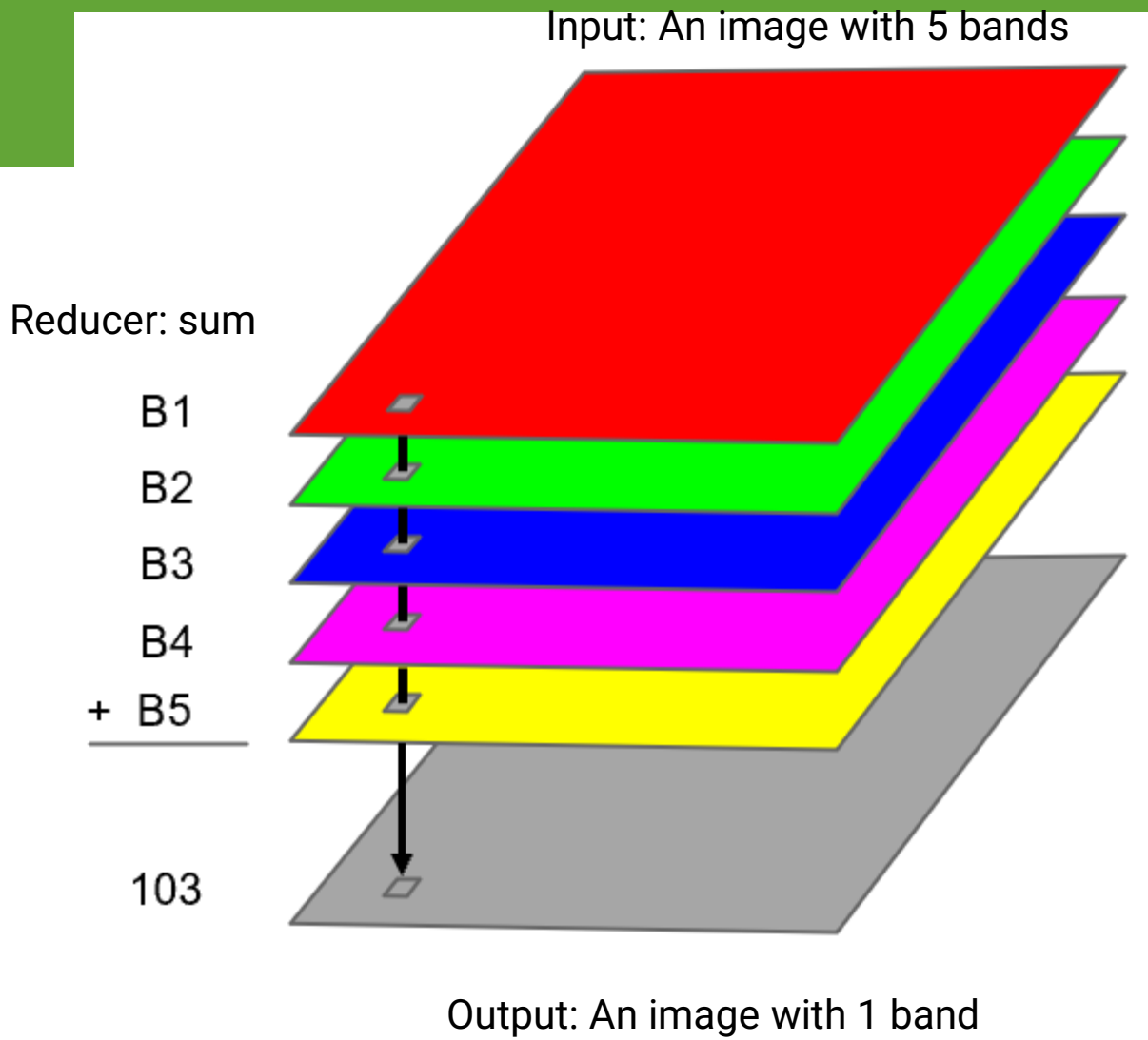
Note: shortcut function sometimes available to accomplish a reducer operation, such as ImageCollection.median(), Image.focal_mode(), FeatureCollection.aggregate_histogram(), etc.

Reduce Bands: Image.reduce(reducer, ...)

Reduction over **bands**

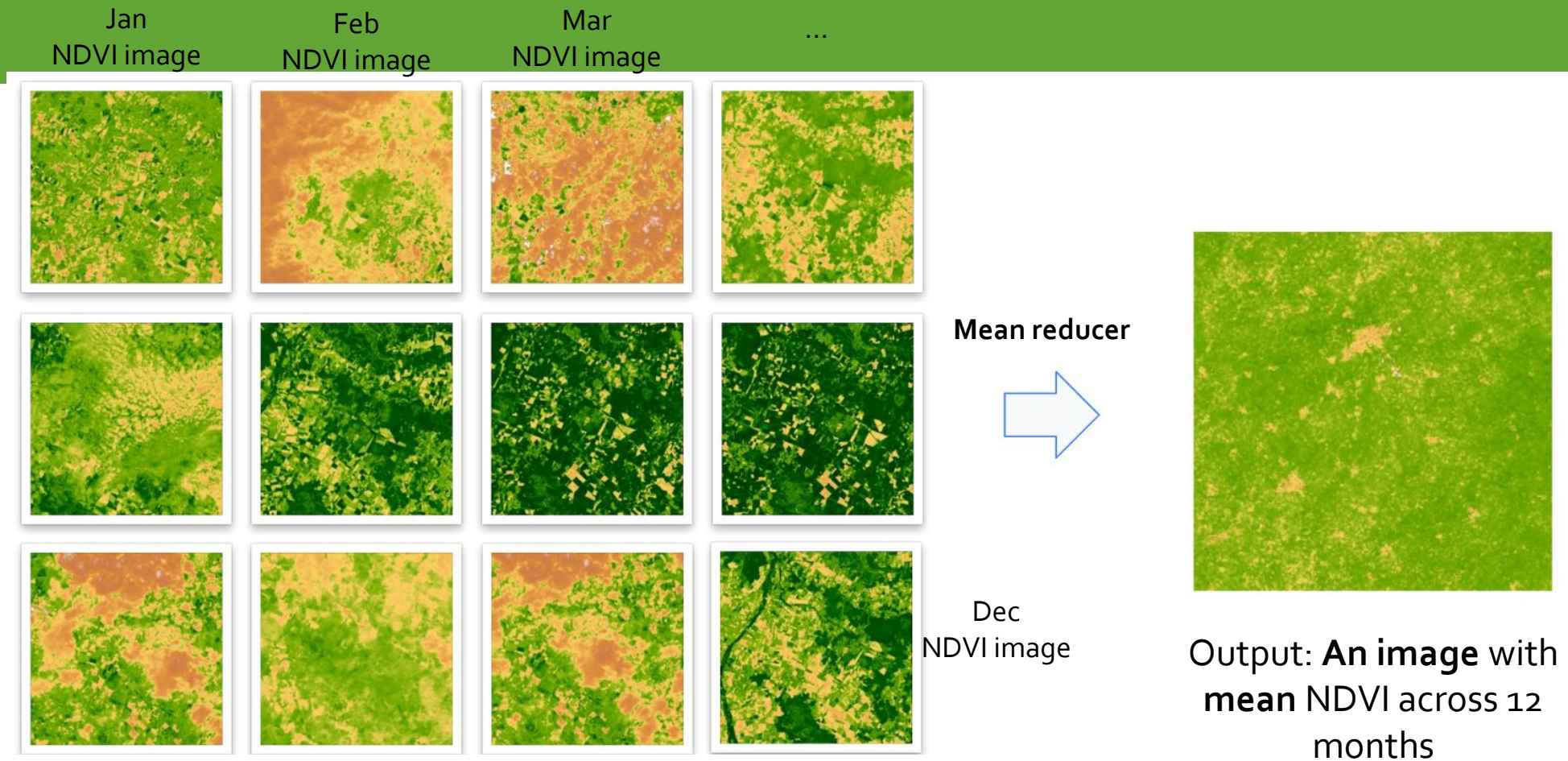
Example:

Calculate the sum of reflectance values *per pixel across bands* in a Landsat image.



Reduce Image Collection: ImageCollection.reduce(reducer, ...)

Reduction over (typically) **time**



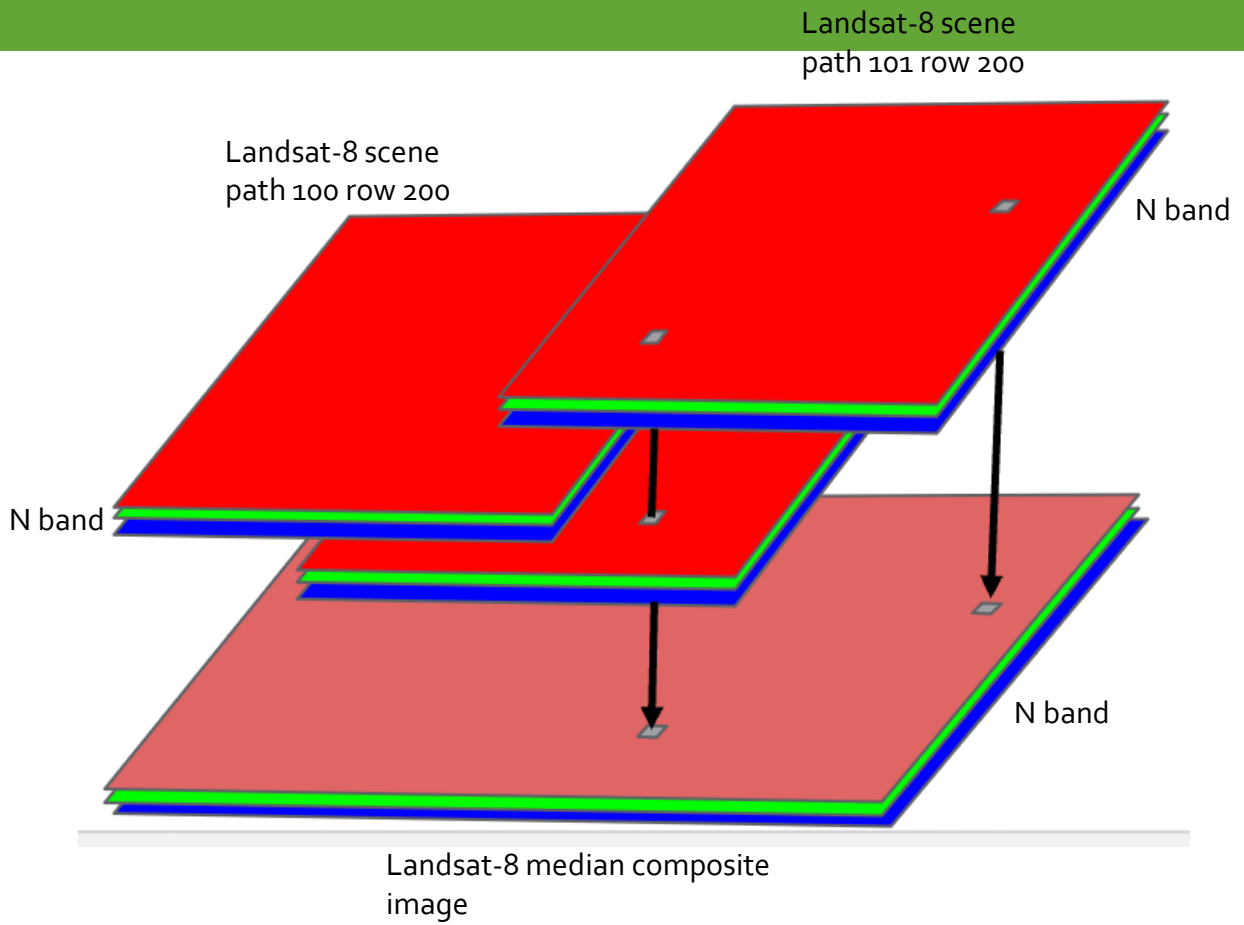
Output: **An image** with **mean** NDVI across 12 months

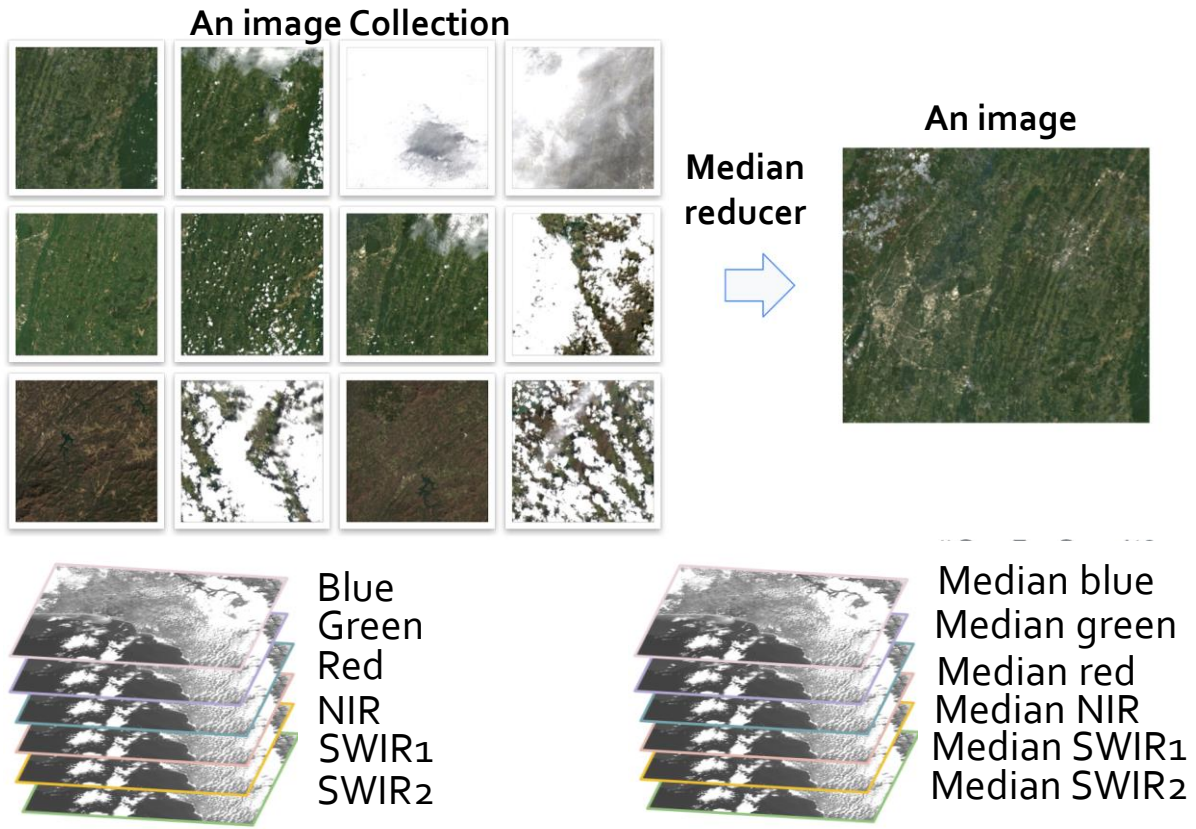
Input: an NDVI **image collection** containing twelve (12) **1-band image** of monthly NDVI composite

Reduce Image Collection: ImageCollection.reduce(reducer, ...)

Reduction over (typically) **time**

Example: cerate a median composite of Landsat images. Input Landsat image collection. Median reducer applied *across images* within the *image collection,* **for *each band* separately.**

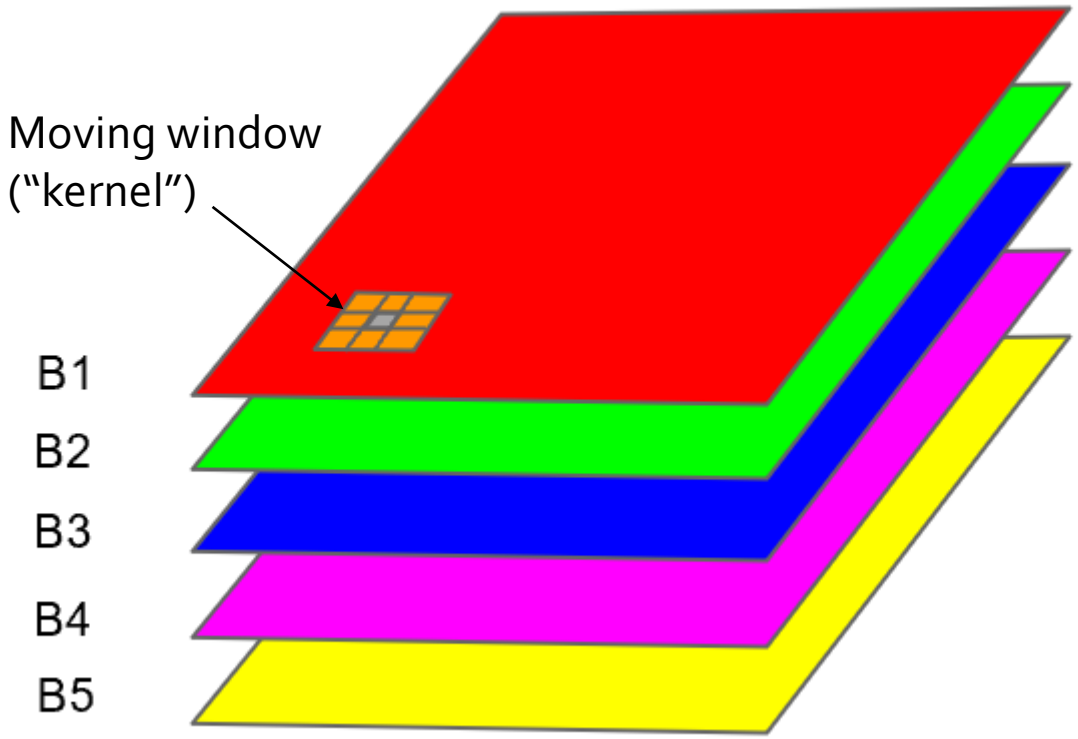




Converts stack of n-band images to a single n-band image

Reduce Neighborhood: Image.reduce(reducer, ...)

Reduction over **space**

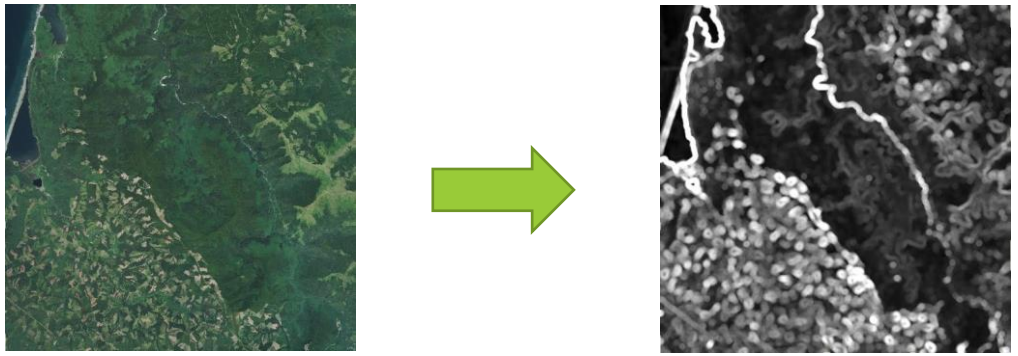


Example:

cerate texture image from an NDVI image with a square window (kernel of radius 1.5 pixels (3 by 3 pixels neighbourhood)

aggregate connected pixels into a cluster (object)

spatial filter (smoothing) to a classified image by taking majority class in every 3 by 3 pixels neighbourhood

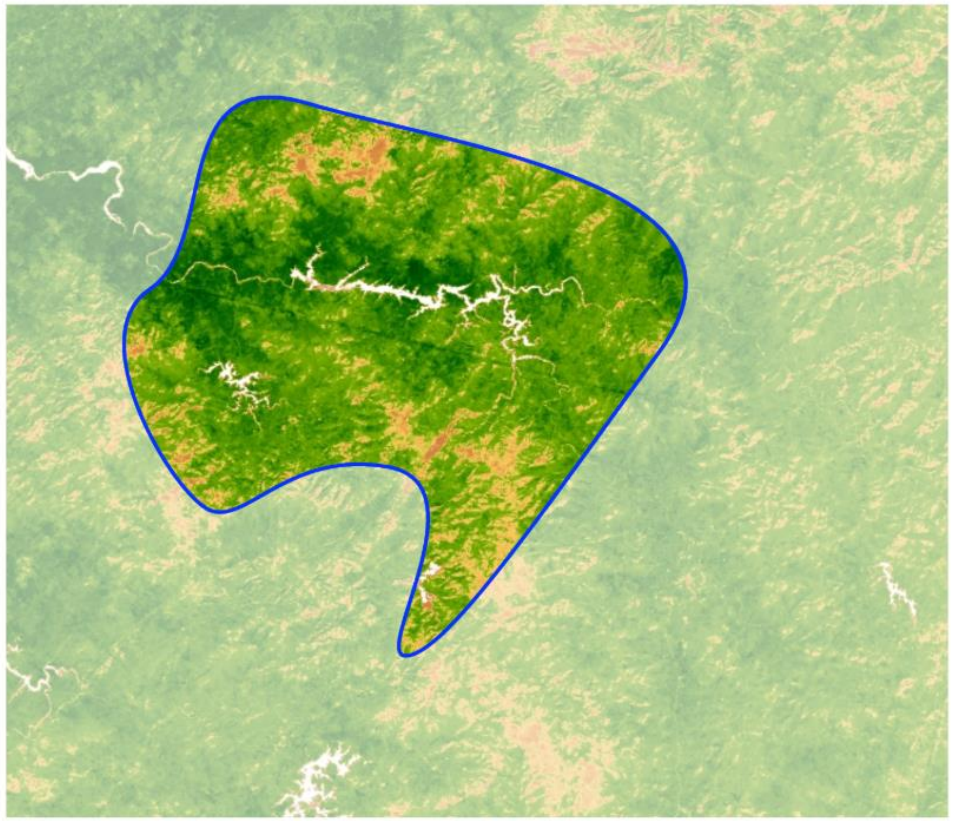


Standard deviation ("texture")

Reduce Region: Image.reduceRegion(reducer, geometry, ...)

Reduction over space

To get statistics of pixel values in *a* region of an image.



1-band (single band) image

Example:

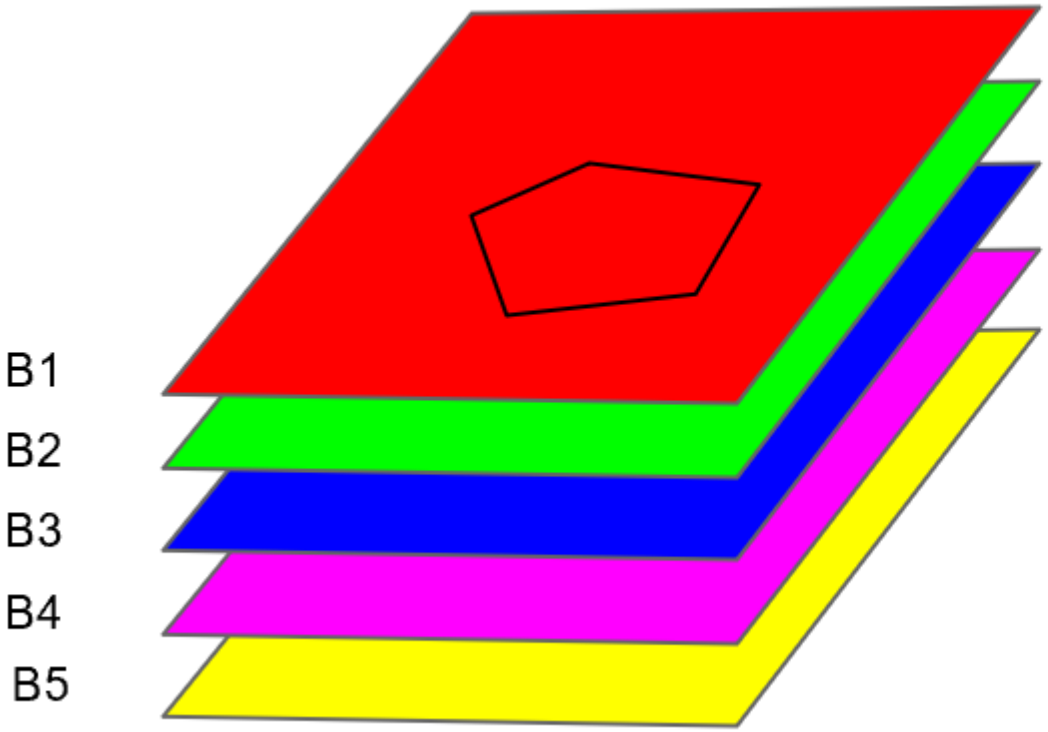
Calculate the minimum and maximum NDVI in a protected area boundary (geomery).

{ min: -0.013,
max: 0.879 }

Reduce Region: Image.reduceRegion(reducer, geometry, …)

Reduction over space

To get statistics of pixel values in *a* region of an image.



Multiple band image

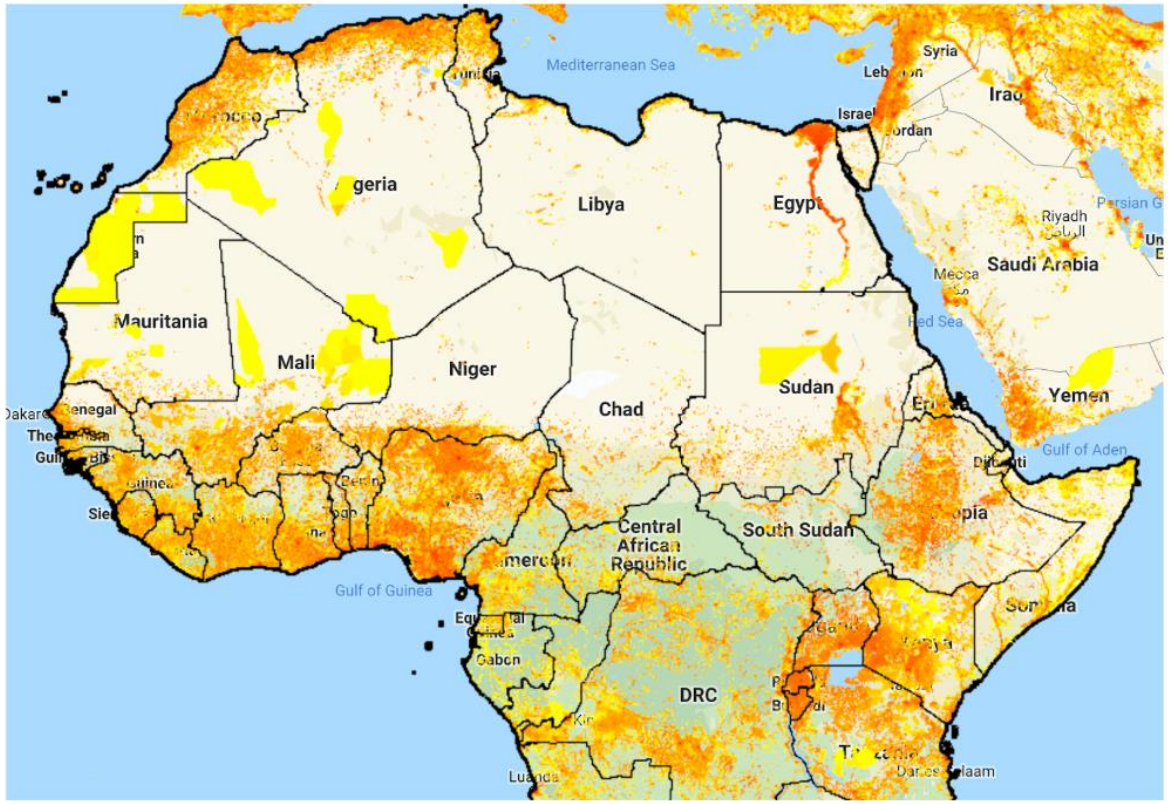
Dictionary
{
B1: 8.3,
B2: 14,
B3: 176,
B4: 1.6,
B5: 7
}

The output of **`reduce region`** is a **dictionary** of the results for **each band.**

Reduce Regions: Image.reduceRegions(collection, reducer, ...)

Reduction over space

To get statistics of pixel values in *several* regions of an image.



- **1-band (single band) image**
- **region: a feature collection**

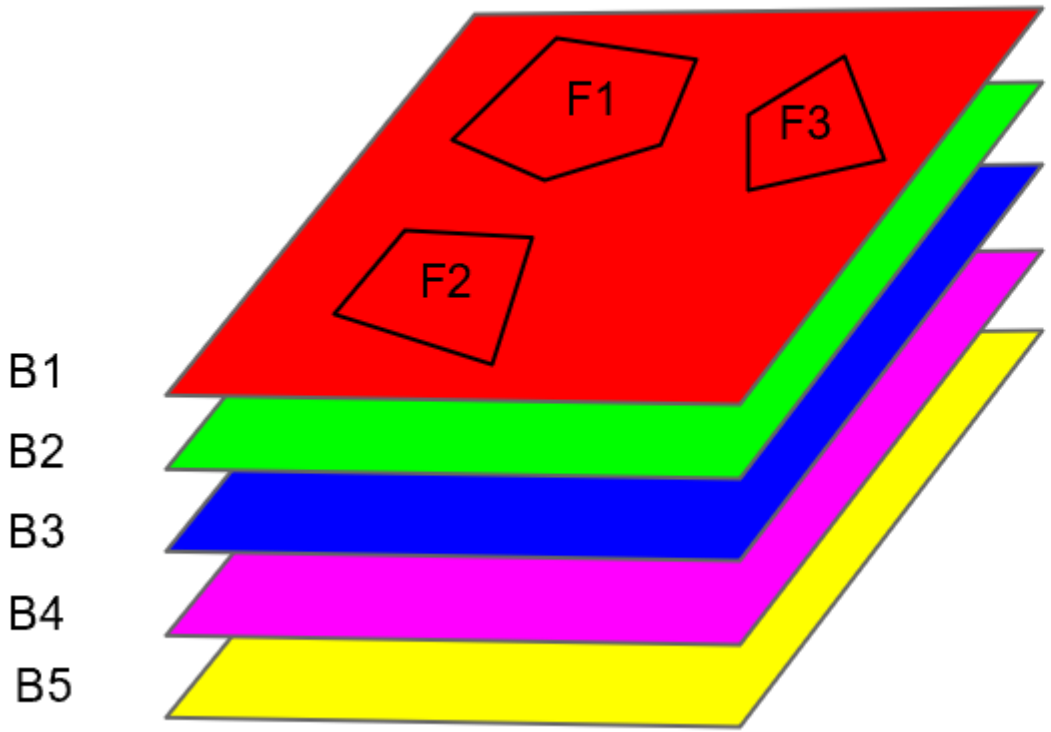
	Median population density
Namibia	2,435
Botswana	2,687
Liberia	3,141
Malawi	3,198
Madagascar	25,485
Ethiopia	31,876
Cabo Verde	34,582
Uganda	35,468
Chad	35,966

In the output feature collection, each input feature is annotated with the results of reducing the image over the pixels enclosed by that feature.

Reduce Regions: Image.reduceRegions(collection, reducer, ...)

Reduction over space

To get statistics of pixel values in *several* regions of an image.



Multiple band image

FeatureCollection

	B1	B2	B3	B4	B5
F1					
F2					
F3					

In the output feature collection, each input feature is annotated with the results of reducing the image over the pixels enclosed by that feature.

Reduce Columns: FeatureCollection.reduceColumns(reducer, selectors, ...)

Reduction over attribute of a feature collection

Example:
Calculate the average number of cases for disease A and disease B across four districts.

District	Number_of_cases_for_disease_A	Number_of_cases_for_disease_B
1	100	1000
2	100	1000
3	200	2000
4	200	2000

Dictionary
{
Number_of_cases_for_disease_A: 150,
Number_of_cases_for_disease_B: 1500
}

ILR
Institute for
Food and
Resource Economics

UNIVERSITÄT BONN

THANK YOU

Land Economics Group

UNIVERSITÄT BONN

ILR
Institute for
Food and
Resource Economics

Land Economics Group

EXTRA: OTHER EXPORT MODES

Life of a Batch request

Instructions

```
def earth_engine_do_a_thing(image):
  mask = image.select("band").eq(1)
  return image.updateMask(mask)

earth_engine_do_a_thing(image)
```

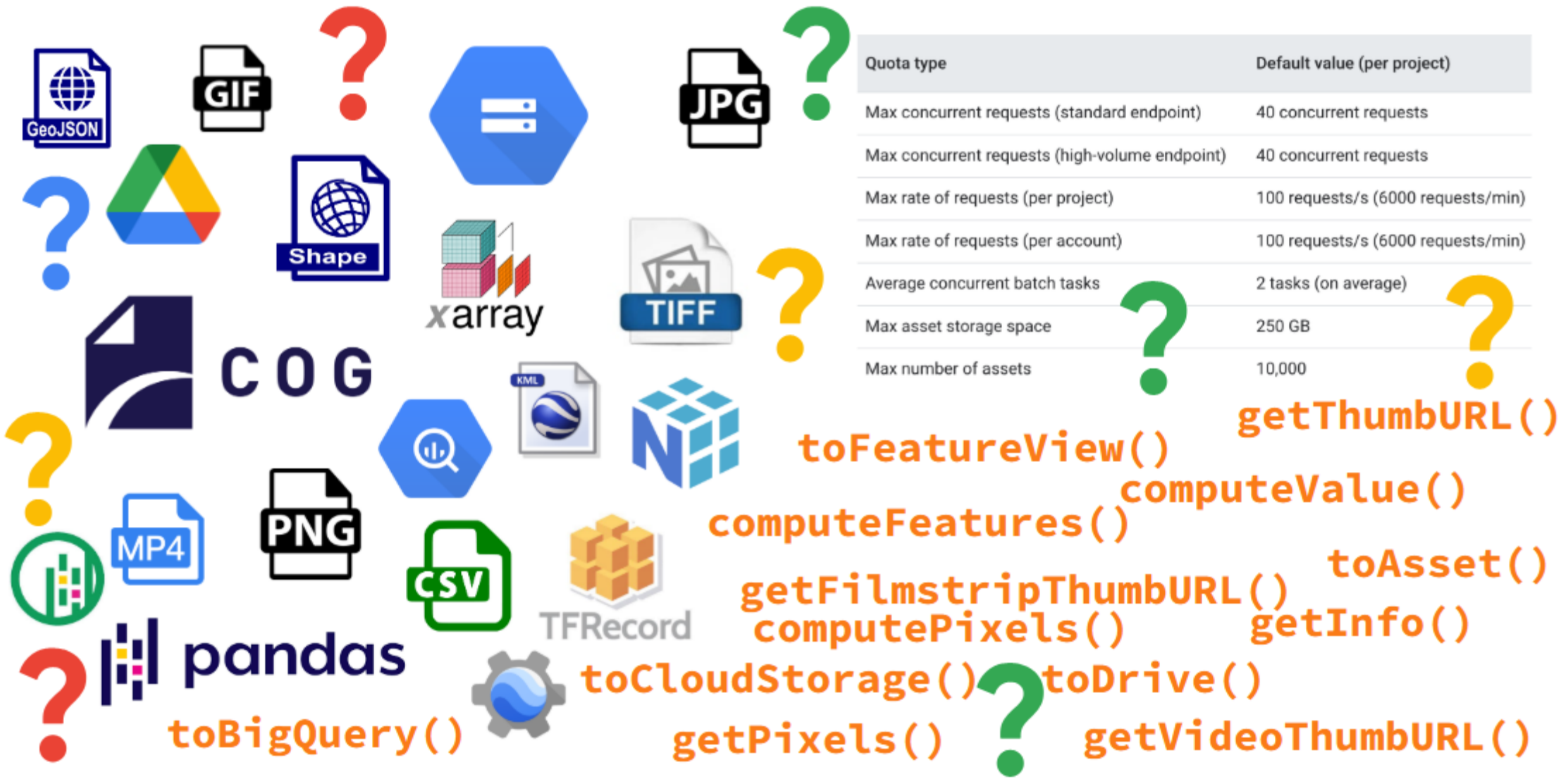
Messenger

Web (REST) API

Workers

Export Destination

Given technology constraints and developer needs, complexity exists



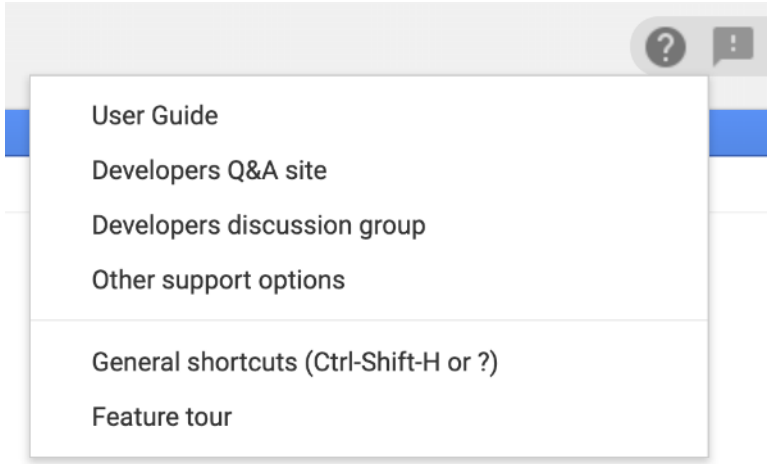
Quota type	Default value (per project)
Max concurrent requests (standard endpoint)	40 concurrent requests
Max concurrent requests (high-volume endpoint)	40 concurrent requests
Max rate of requests (per project)	100 requests/s (6000 requests/min)
Max rate of requests (per account)	100 requests/s (6000 requests/min)
Average concurrent batch tasks	2 tasks (on average)
Max asset storage space	250 GB
Max number of assets	10,000

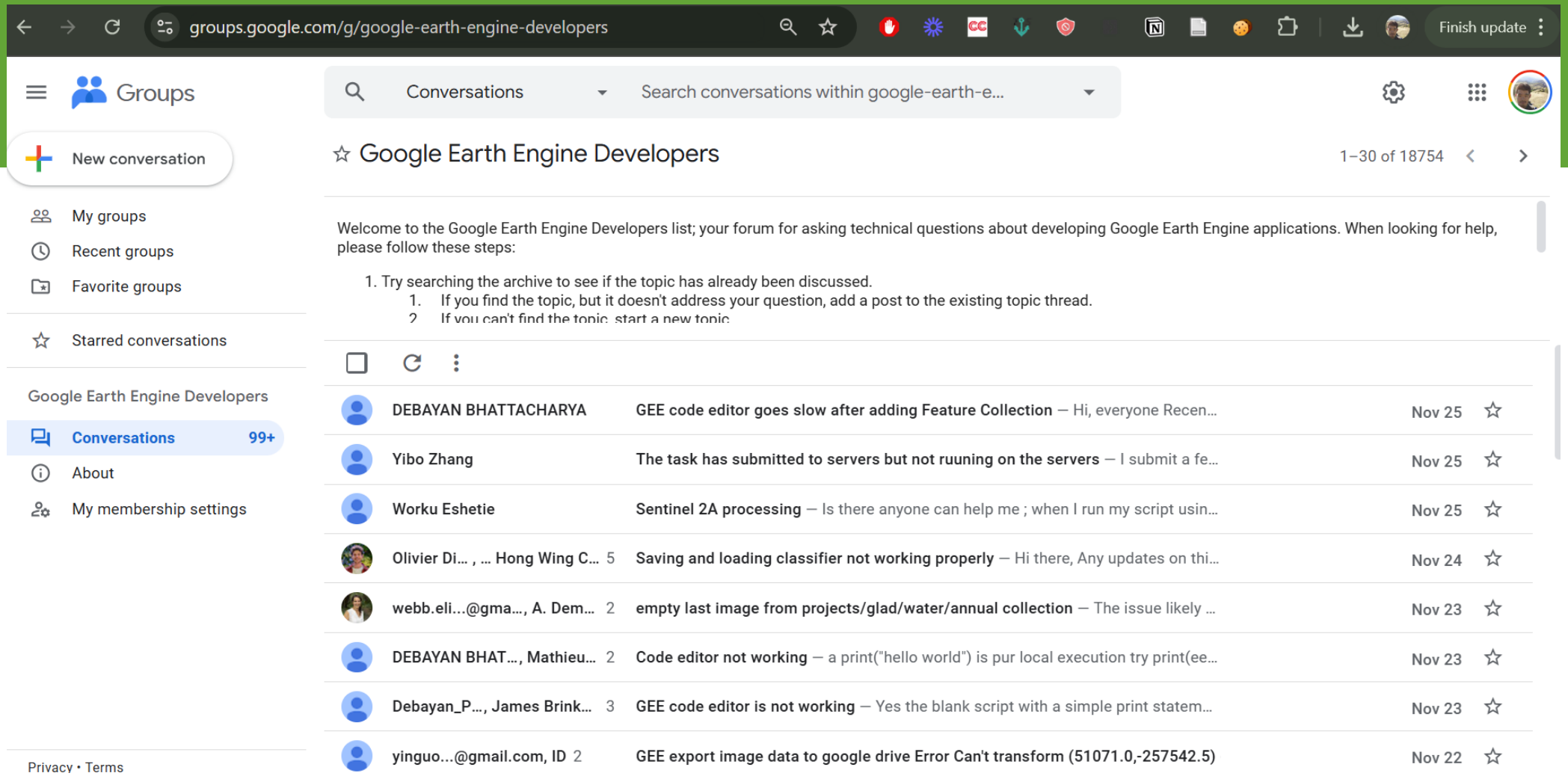
EXTRA: PITFALLS IN USING EARTH ENGINE (AT SCALE)

FINDING HELP

- Primary Help Menu:
  - User Guide
    - Tutorials
  - GIS StackExchange
  - Developer Forum

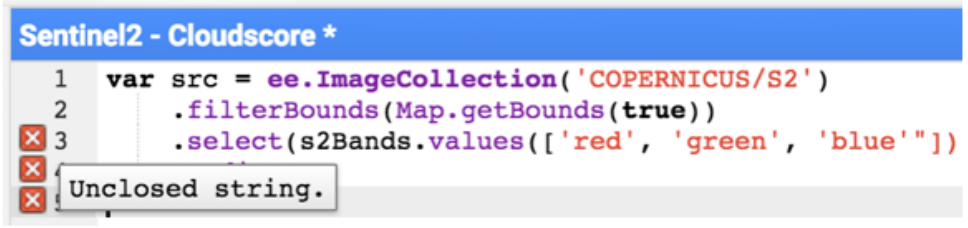
- Learn (and steal) from other code:
  - Examples
  - Google Search




groups.google.com/g/google-earth-engine-developers
Finish update
Groups
Conversations
Search conversations within google-earth-e...
New conversation
My groups
Recent groups
Favorite groups
Starred conversations
Google Earth Engine Developers
Conversations 99+
About
My membership settings
Privacy • Terms
☆ Google Earth Engine Developers
1–30 of 18754
Welcome to the Google Earth Engine Developers list; your forum for asking technical questions about developing Google Earth Engine applications. When looking for help, please follow these steps:
1. Try searching the archive to see if the topic has already been discussed.
1. If you find the topic, but it doesn't address your question, add a post to the existing topic thread.
2. If you can't find the topic, start a new topic
DEBAYAN BHATTACHARYA GEE code editor goes slow after adding Feature Collection — Hi, everyone Recen... Nov 25
Yibo Zhang The task has submitted to servers but not ruuning on the servers — I submit a fe... Nov 25
Worku Eshetie Sentinel 2A processing — Is there anyone can help me ; when I run my script usin... Nov 25
Olivier Di... , ... Hong Wing C... 5 Saving and loading classifier not working properly — Hi there, Any updates on thi... Nov 24
webb.eli...@gma..., A. Dem... 2 empty last image from projects/glad/water/annual collection — The issue likely ... Nov 23
DEBAYAN BHAT..., Mathieu... 2 Code editor not working — a print("hello world") is pur local execution try print(ee... Nov 23
Debayan_P..., James Brink... 3 GEE code editor is not working — Yes the blank script with a simple print statem... Nov 23
yinguo...@gmail.com, ID 2 GEE export image data to google drive Error Can't transform (51071.0,-257542.5) Nov 22

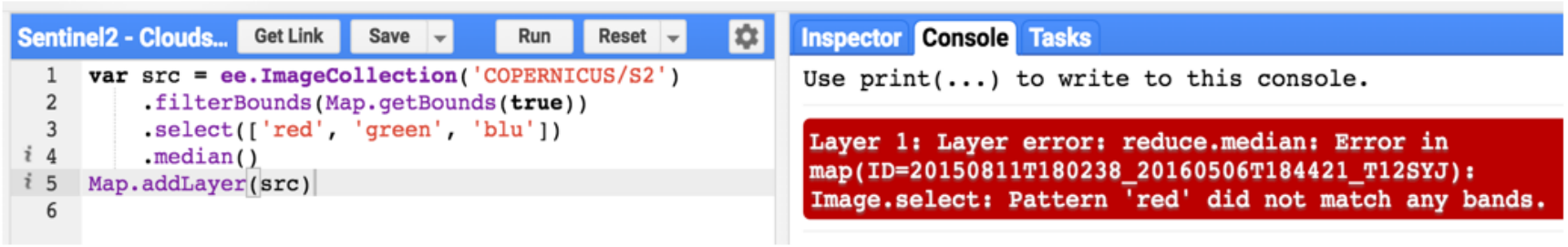
THREE TYPE OF ERRORS

Syntax Errors



Client (symantic error) Errors

Server Errors



https://developers.google.com/earth-engine/debugging

COMMON GOTCHA'S

- Casting

```
collection.first().date()
```

collection.first(...).date is not a function

```
ee.Image(collection.first()).date()
```

- Mapped functions

Error – this code doesn't work!

```
var collection = ee.ImageCollection('MODIS/051/MOD44B');

var badMap2 = collection.map(function(image) {
  return image.date();
});

// Error: Collection.map: A mapped algorithm must return a Feature or Image.
print(badMap2);
```

Solution – set a property!

```
var collection = ee.ImageCollection('MODIS/051/MOD44B');

var okMap2 = collection.map(function(image) {
  return image.set('date', image.date());
});
print(okMap2);
```

SCALING ISSUES

Scaling issues
”Error: Computation timed out”
”Error: User memory limit exceeded”
“Error: Too many aggregations”
“Error: Internal server error”

- Export the result to Asset / Drive / Cloud Storage

Bad – don't do this!

```
var ridiculousComputation = ee.Image(1).reduceRegion({
  reducer: 'count',
  geometry: ee.Geometry.Rectangle([-180, -90, 180, 90],
  scale: 100,
  maxPixels: 1e11
});

// Error: Computation timed out.
print(ridiculousComputation);
```

Good – use `Export`!

```
Export.table.toDrive({
  collection: ee.FeatureCollection([
    ee.Feature(null, ridiculousComputation)
  ]),
  description: 'ridiculousComputation',
  fileFormat: 'CSV'
});
```

UNIVERSITÄT BONN ILR Institute for Food and Resource Economics Land Economics Group

For more best practices see https://developers.google.com/earth-engine/best_practices

ebugging

How I Debug Your Code - my checklist

Low Hanging Fruit
- getInfo()
- for loops
- iterate()
- toList()
- Complex geometries
- Don't clip
- image.reduceToVectors()
- image.reproject()
- image.resample()
- image.reduceResolution()
- Joins

Collections
- filterDate / filterBounds
- calendarRange() without filterDate()
- collection.geometry()
- toBands() / toArray()

Aggregations
- Tilescale in reduce*
- Combine reducers
- image.reduceNeighborhood()
  - Neighborhood size
  - Use optimizations

How I Debug Your Code - my checklist

Distances

- use fastDistanceTransform() / ee.Image.pixelArea().sqrt()

Geometries

- Specify an error margin
- Simplify if possible
- bounds() vs. drawing a box
- Discard them completely

Pre-caching before sampling

Classifier size (trees, training)

Never use Math.random()

Some quotas are adjustable

Adjustable quota limits

The following limits **may be adjusted** on a per-user or per-project basis. See the help page for how to request additional quota.

Quota type	Default value
Max concurrent requests (standard endpoint)	40 concurrent requests
Max concurrent requests (high-volume endpoint)	40 concurrent requests
Max rate of requests (per project)	100 requests/s (6000 requests/min)
Max rate of requests (per user)	100 requests/s (6000 requests/min)
Average Concurrent batch tasks	2 tasks (on average)
Max asset storage space	250 GB
Max number of assets	10,000

COMMON EE ERRORS IN ML WORKFLOW

- Out of memory error
  - Usually happens when tile is too large. When you sample from an image you're sampling from a tile. In Earth Engine each tile 1000x1000 so with a lot of bands that can be too much memory for us to handle.
- Computation timed out
  - Can happens if classifying the table/image takes too long
- Request too large
  - Too large of a training dataset or the model created is too large.

What Could Possibly Go Wrong?

Computation Timed Out

300 second timeout.

Easy to hit with large regions & sparse training data, large classifiers or complex pre-processing

Out Of Memory

2 gigabyte 'stack' depth

256x256 pixel tile *
10 bands * 100 images *
8 bytes / sample = 512MB

Computed Object Is Too Large

100MB cache limit.

Applies to both sampled training data and the final computed classifier.

©Google

Punchlines

- Training table too large
  Break up the region into multiple tables
- Training timeout (5 minutes)
  Break up the region for more parallelism & caching (or export)
- Combined training table too large
  Call train() multiple times
- Classifier too large
  Use fewer trees, or increase minLeafPopulation
- OOM when extracting training data
  Use tileScale in aggregation.

ILR
Institute for
Food and
Resource Economics

UNIVERSITÄT BONN

THANK YOU

Land Economics Group

UNIVERSITÄT BONN

ILR
Institute for
Food and
Resource Economics

Land Economics Group