Google Earth Engine and machine learning for Earth monitoring

Alexander Uzhinskiy

Joint Institute for Nuclear Research, Joliot-Curie 6, 141980 Dubna, Russia, auzhinskiy@jinr.ru

auzhinskiy@jinr.ru; Tel.: +79057766865

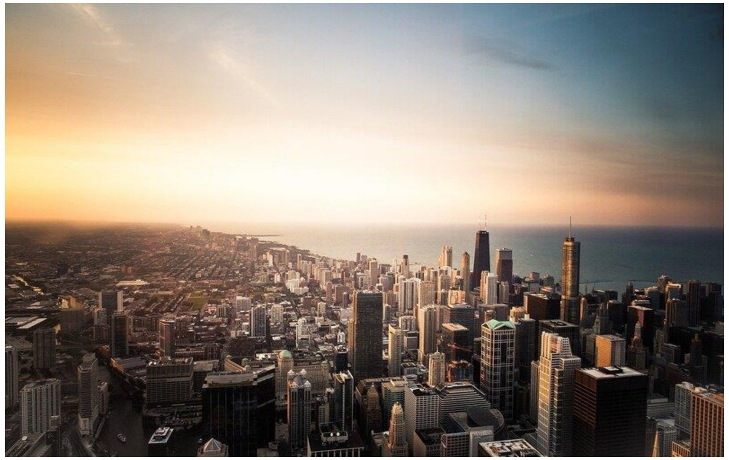
Environmental Concerns

The environmental problems like global warming, acid rain, air pollution, urban sprawl, waste disposal, ozone layer depletion, water pollution, climate change and many more affect every human, animal, and nation on this planet.

Over the last few decades, the exploitation of our planet and the degradation of our environment has gone up at an alarming rate. As our actions have been not in favor of protecting this planet, we have seen natural disasters striking us more often in the form of flash floods, earthquakes, blizzards, tsunamis, and cyclones.

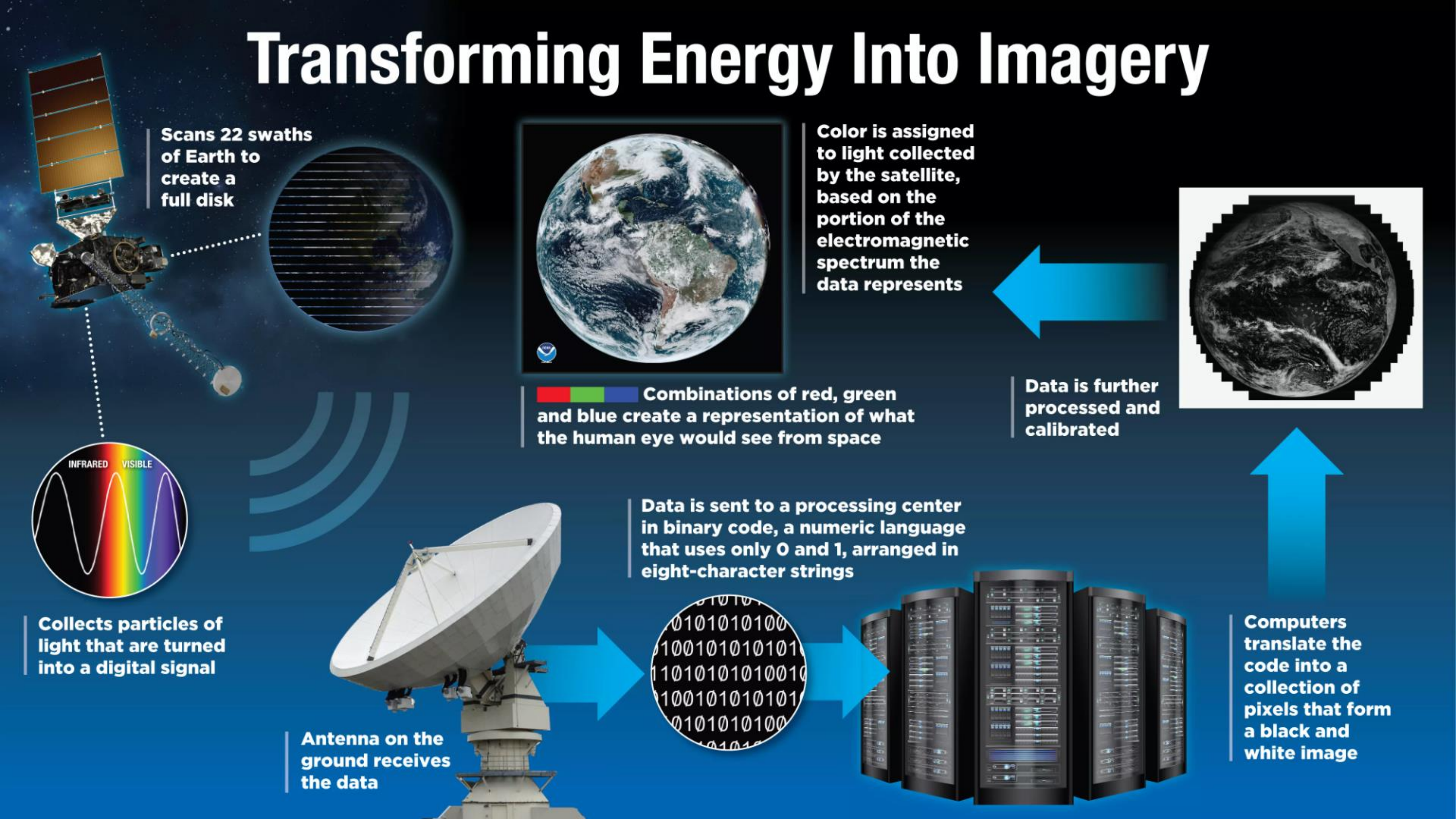




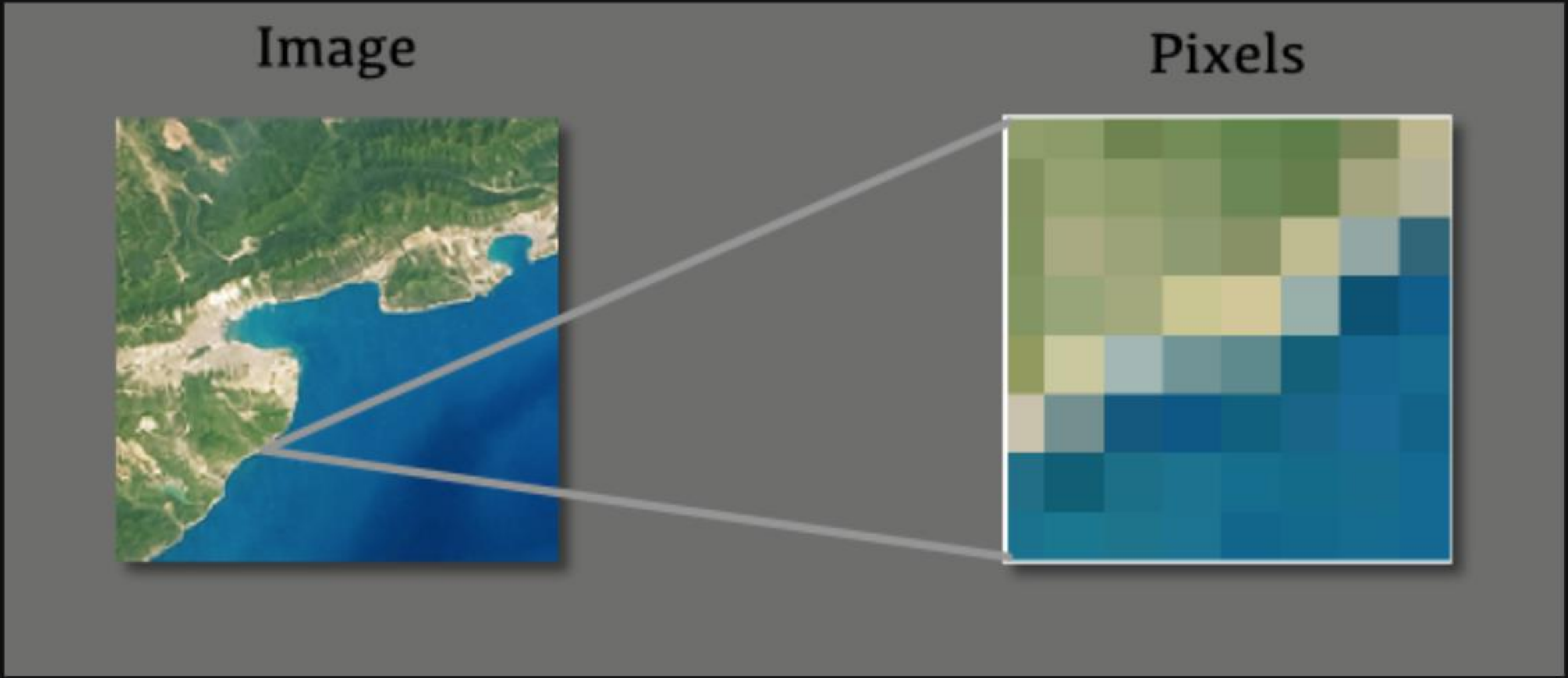



Satellite programs

Source: NASA

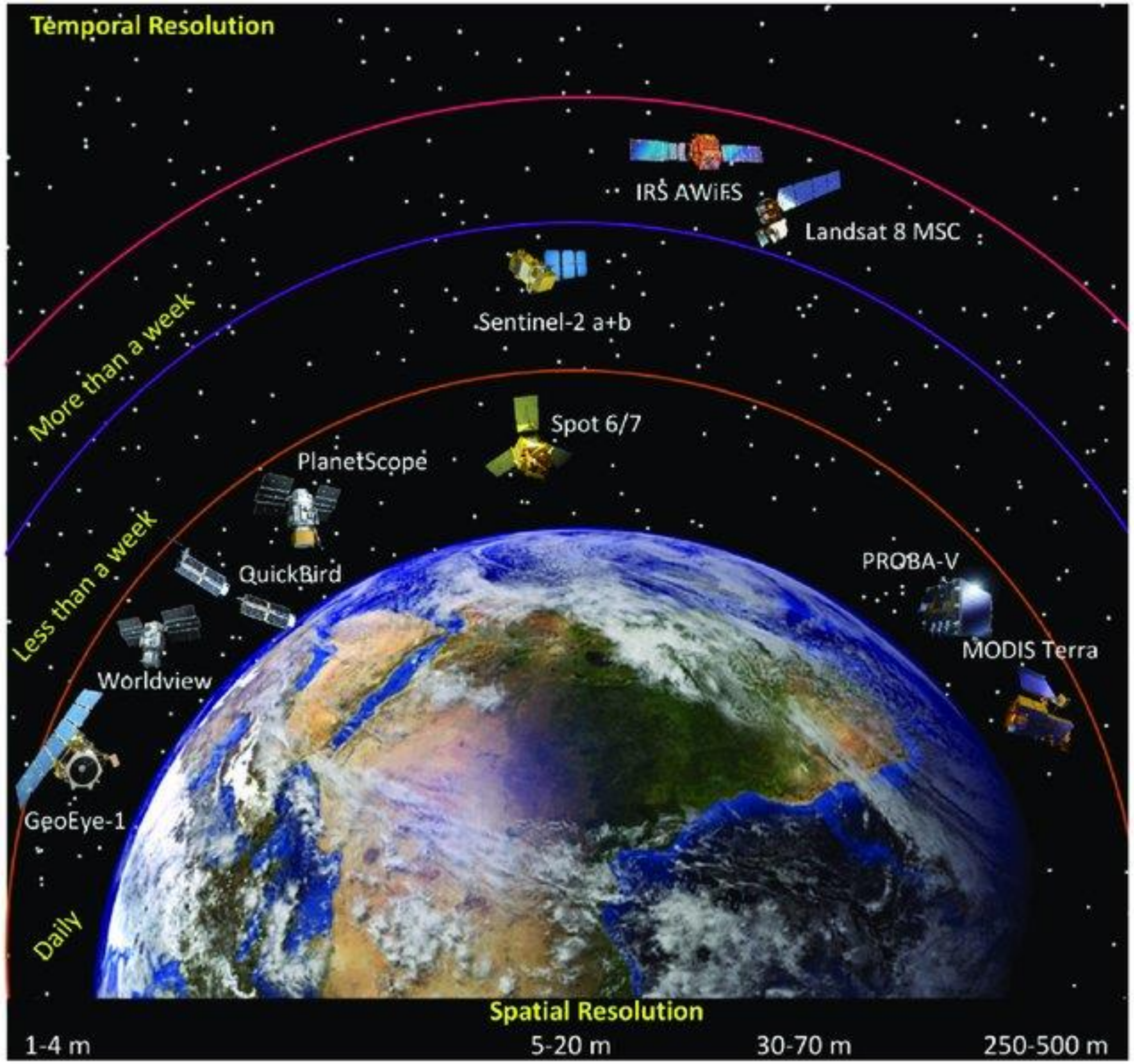


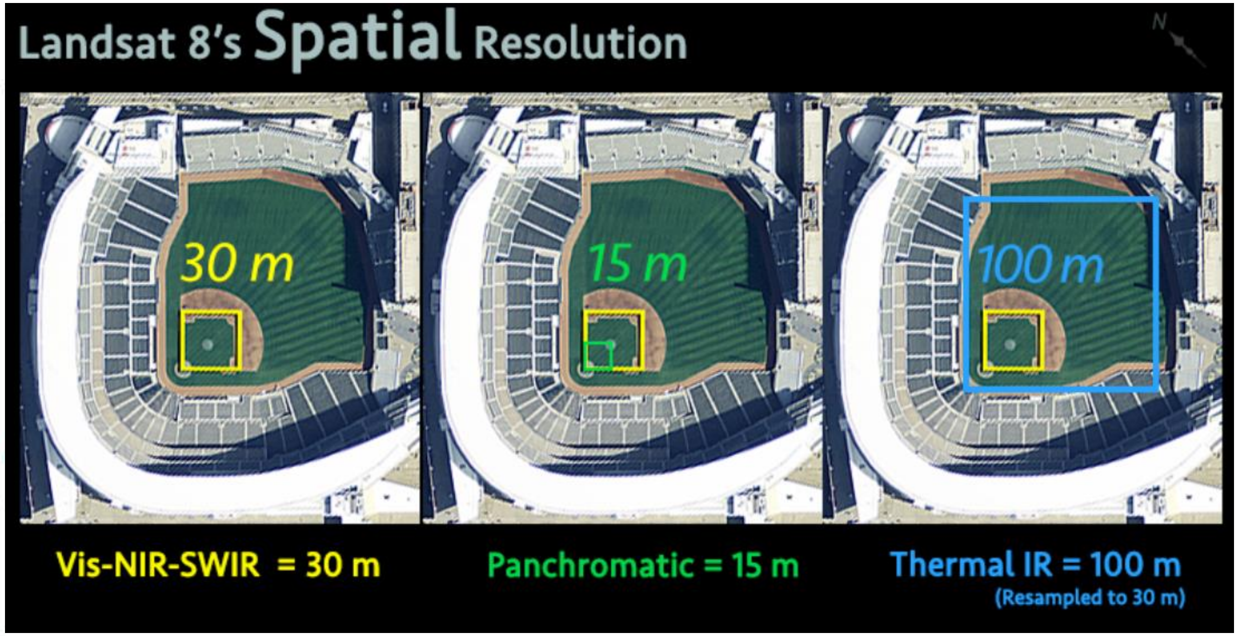
Transforming Energy Into Imagery
Scans 22 swaths of Earth to create a full disk
Color is assigned to light collected by the satellite, based on the portion of the electromagnetic spectrum the data represents
Combinations of red, green and blue create a representation of what the human eye would see from space
Data is further processed and calibrated
INFRARED
VISIBLE
Collects particles of light that are turned into a digital signal
Data is sent to a processing center in binary code, a numeric language that uses only 0 and 1, arranged in eight-character strings
Computers translate the code into a collection of pixels that form a black and white image
Antenna on the ground receives the data

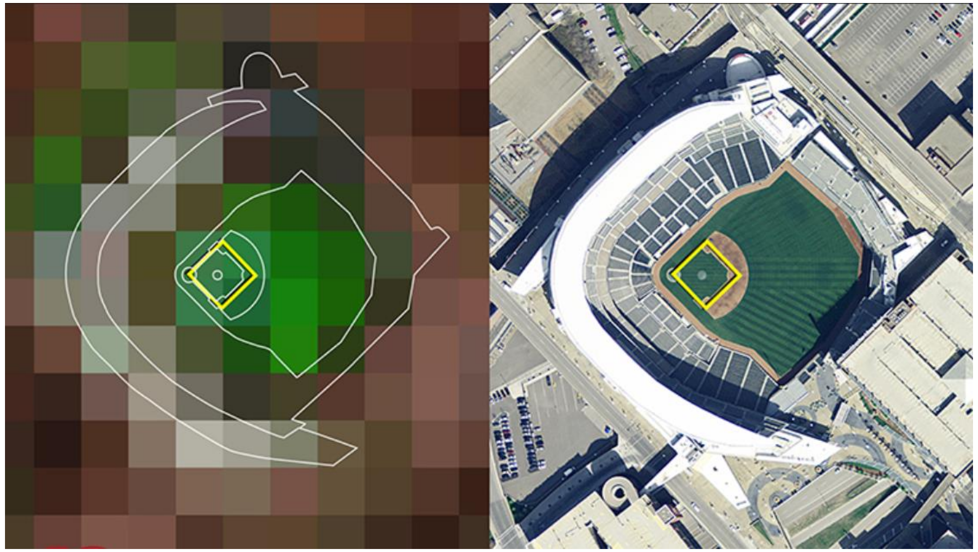


Digital images are made of pixels, and pixels are representations of numeric values

rce: NASA

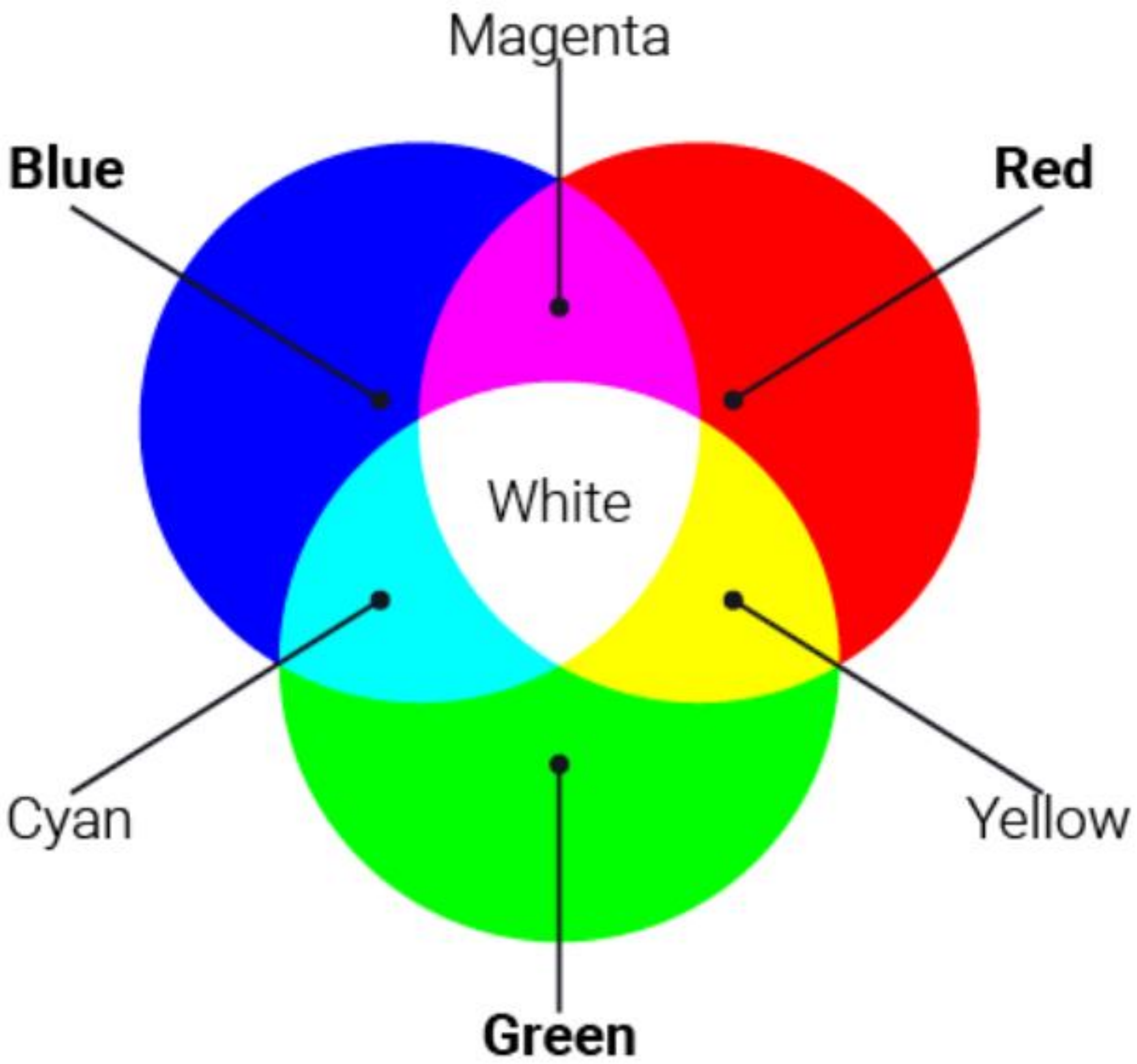

Temporal Resolution
More than a week
Less than a week
Daily
IRS AWiFS
Landsat 8 MSC
Sentinel-2 a+b
Spot 6/7
PlanetScope
QuickBird
Worldview
GeoEye-1
PROBA-V
MODIS Terra
Spatial Resolution
1-4 m
5-20 m
30-70 m
250-500 m


Landsat 8's Spatial Resolution
30 m
15 m
100 m
Vis-NIR-SWIR = 30 m
Panchromatic = 15 m
Thermal IR = 100 m
(Resampled to 30 m)



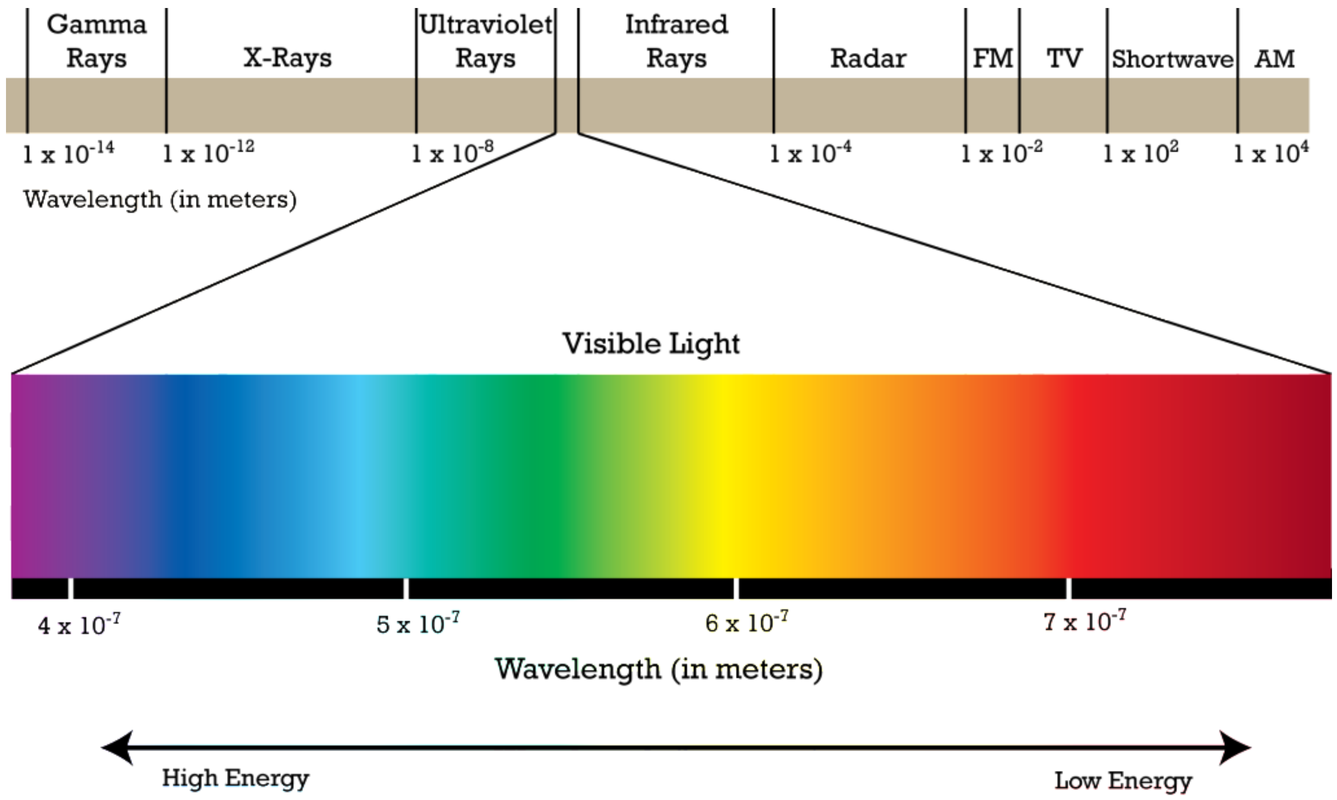
Source: NASA

RGB



Bands

- RS sensors can collect data in all portions of the EM spectrum
- Multispectral sensors have spectral sensitivity limitations (spectral resolution)
    - The wavelength ranges recorded by sensors are called bands or “channels”, varies with sensor



Bands

- Landsat collects data in 11 "channels" throughout the EM spectrum

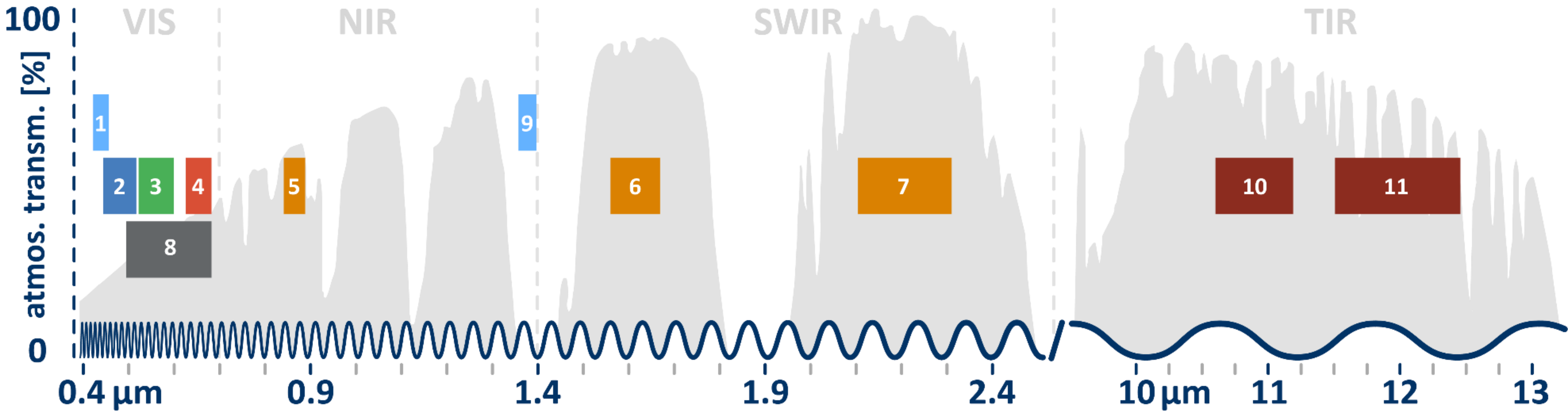
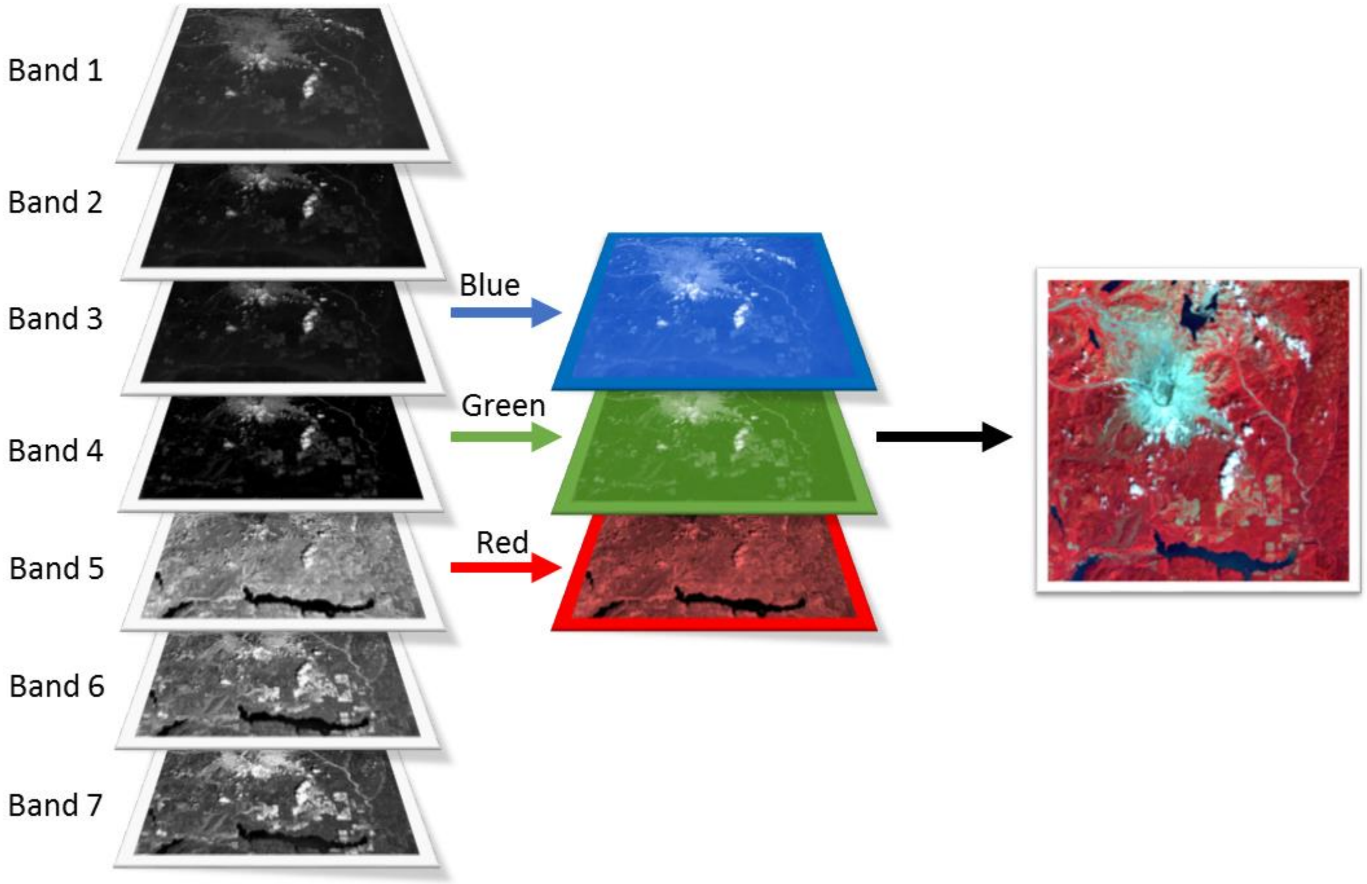
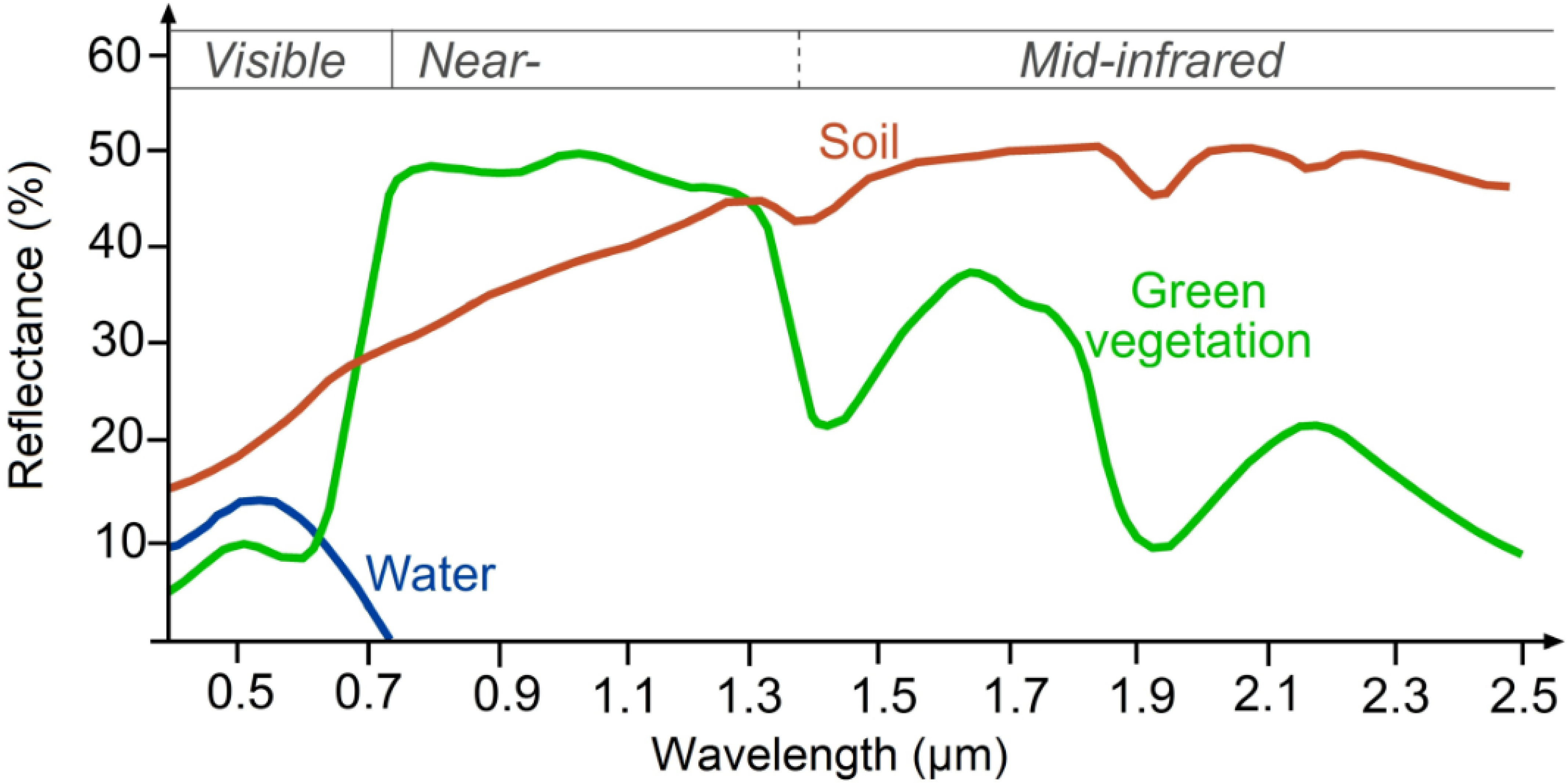


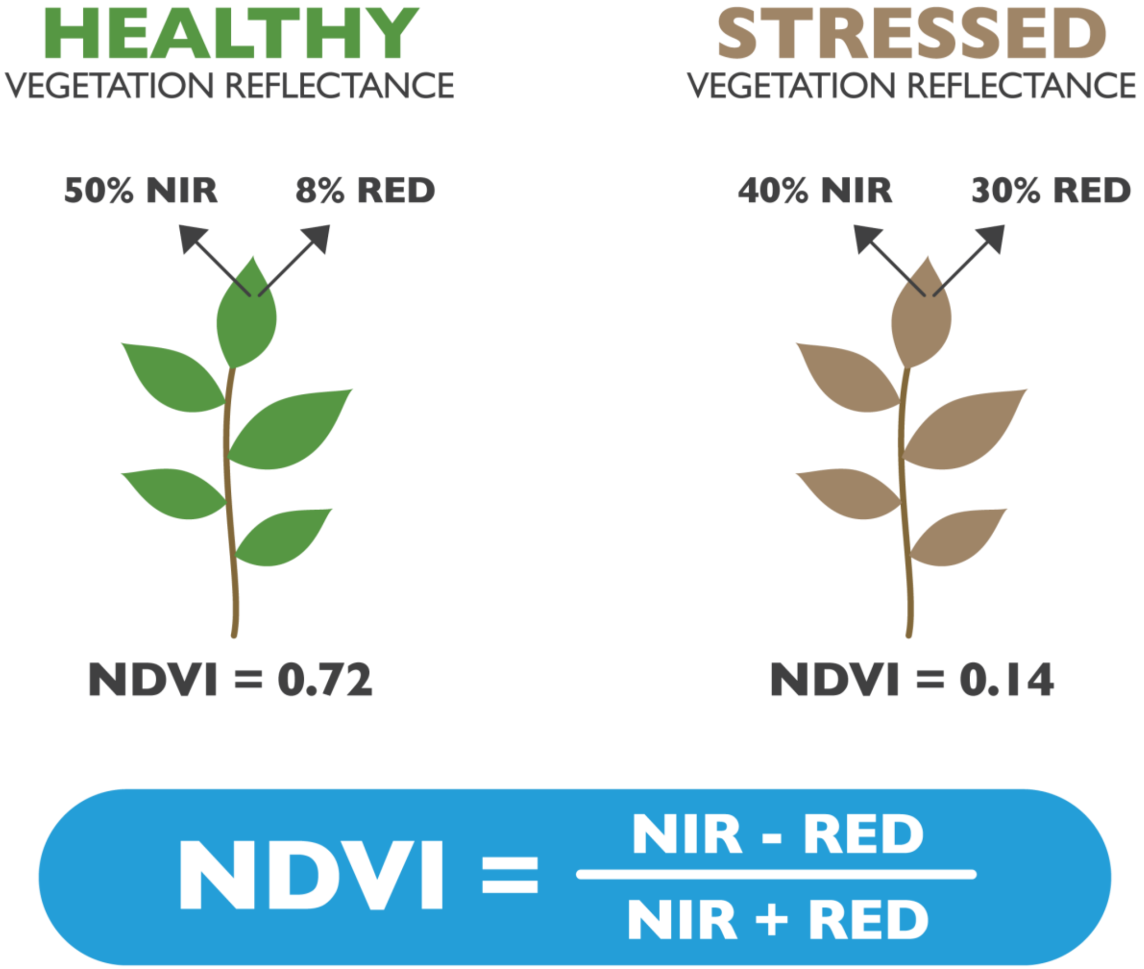
Figure reference

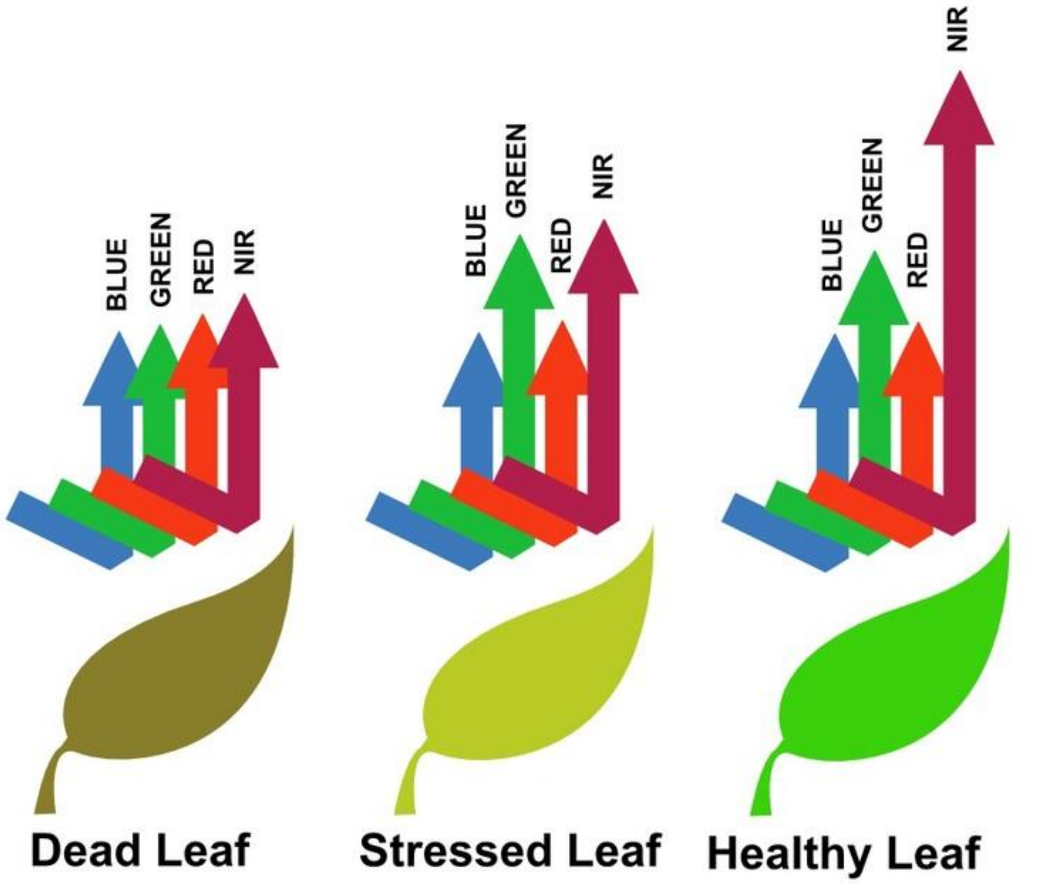
Band Combinations

- When you display a remote sensing image on a computer, you are limited to 3 bands
- This is what produces true color composites and false color composites




Visible
Near-
Mid-infrared
Soil
Green vegetation
Water
Reflectance (%)
60
50
40
30
20
10
0.5
0.7
0.9
1.1
1.3
1.5
1.7
1.9
2.1
2.3
2.5
Wavelength (μm)


HEALTHY
VEGETATION REFLECTANCE
50% NIR
8% RED
NDVI = 0.72
STRESSED
VEGETATION REFLECTANCE
40% NIR
30% RED
NDVI = 0.14
NDVI = NIR - RED / NIR + RED


BLUE
GREEN
RED
NIR
Dead Leaf
Stressed Leaf
Healthy Leaf

Old fashion approach

Task: Map all change between 2000 – 2010 over a specific forest

Extent of change

Year of occurrence

Solution:

- Select Area of Interest
- Find WRS path/row(s) – assume an area of 4 scenes

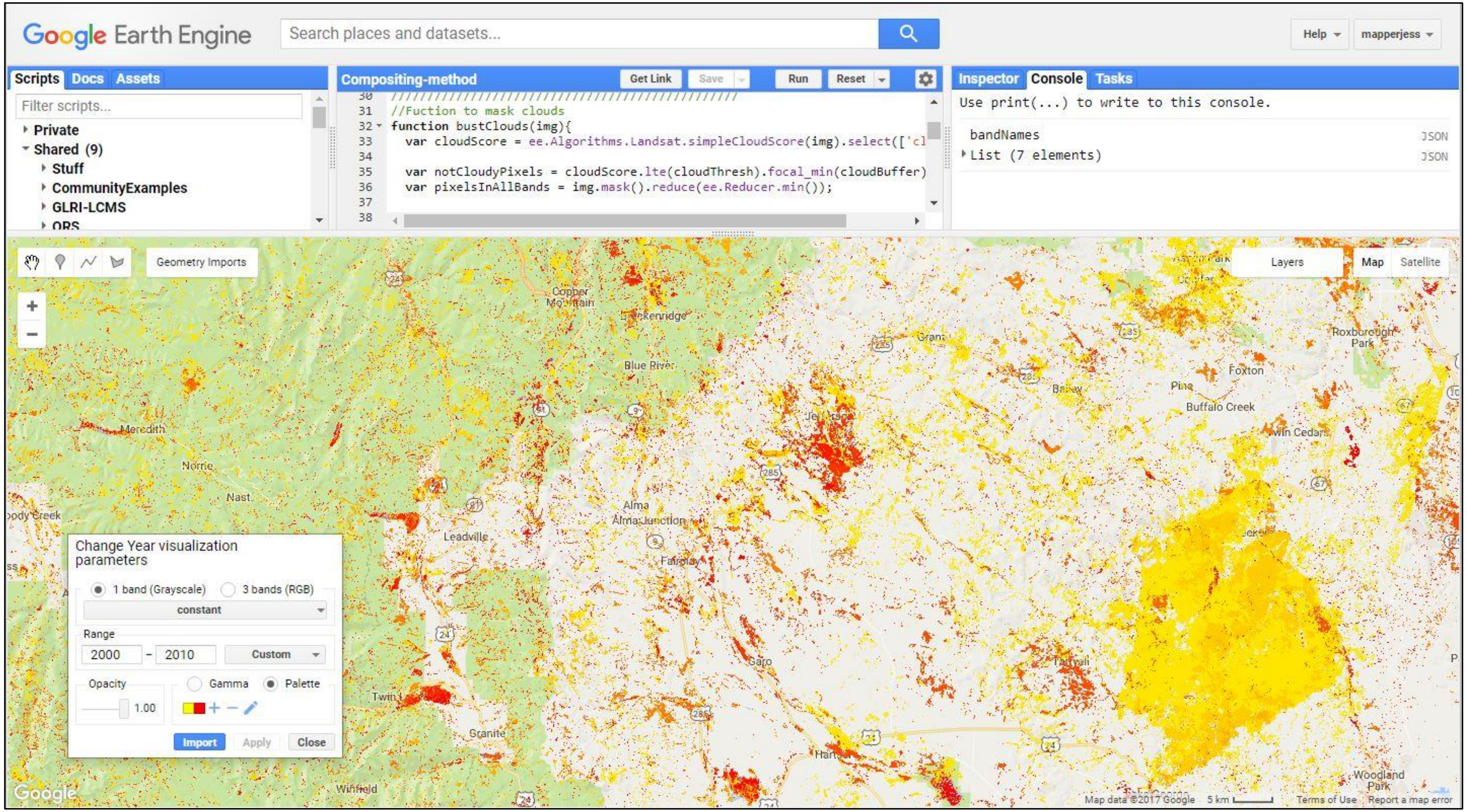
Data Prep:

- Download and store all Landsat during growing season (1 GB / zipped scene)
- Extract and layer stack all Landsat (1.75 GB / scene)
- ~48 scenes per year * 11 years = ~528 scenes, or 924 GB
- Apply atmospheric correction (924 more GB)
- Normalize scenes
- Apply FMASK or similar to remove clouds and shadows
- Create composite and mosaic by year
- Generate vegetation index per year (NDVI and/or NBR)

Analysis:

- Build spatial model in ERDAS to compare year pairs
- Generate change layer
- Classify pixels > certain value as "change"
- Build spatial model to apply year attribute to each pixel
- Repeat for each year pair
- Build spatial model to stack all attributed change layers into a single raster image, with the most recently changed pixel on top
- Apply color ramp visually demonstrating change

GEE approach



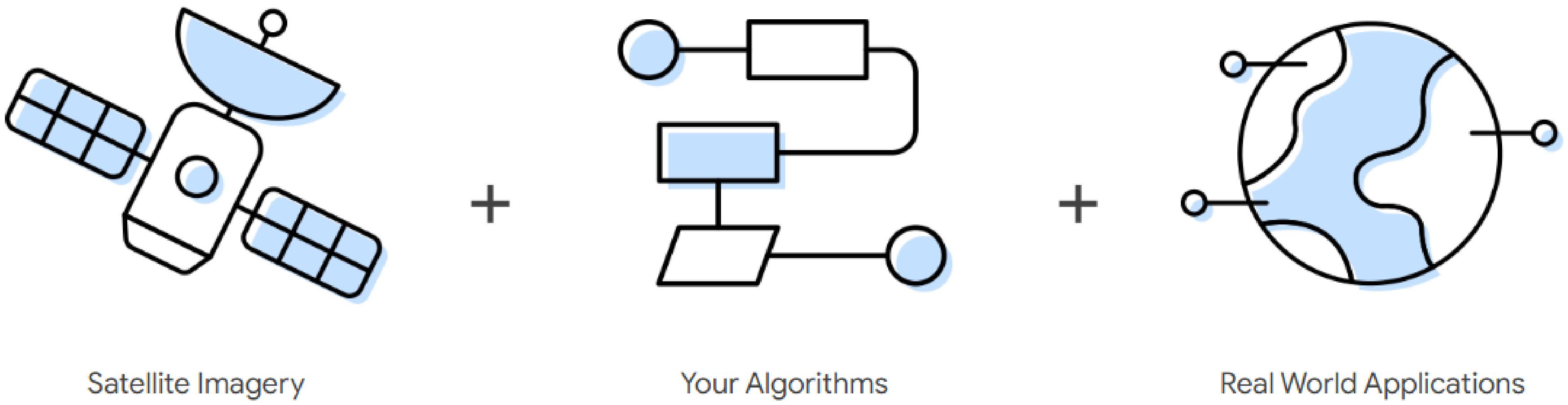
The new method took an experienced geospatial programmer about an hour and ~100 lines of code to generate a raster layer showing the extent of landscape change thematically colored by year of change (in this case, yellow changed closer to the year 2000 and red colors changed closer to 2010.

Users can change the AOI very simply and run this same process anywhere in the world, then export the results to a raster TIFF image.

This represents an extremely dramatic improvement in efficiency. In fact, it allows us to ask new questions.

Google Earth Engine

Google Earth Engine combines a multi-petabyte catalog of satellite imagery and geospatial datasets with planetary-scale analysis capabilities. Scientists, researchers, and developers use Earth Engine to detect changes, map trends, and quantify differences on the Earth's surface. Earth Engine is now available for commercial use, and remains free for academic and research use.



Earth Engine provides easy, web-based access to an extensive catalog of satellite imagery and other geospatial data in an analysis-ready format. The data catalog is paired with scalable compute power backed by Google data centers and flexible APIs that let you seamlessly implement your existing geospatial workflows. This enables cutting-edge, global scale analysis and visualization.

The Earth Engine Data Catalog

Landsat & Sentinel 1, 2
10-30m, weekly

MODIS
250m daily

Vector Data
WDPA, Tiger

Terrain & Land Cover

Weather & Climate
NOAA NCEP, OMI, ...

... and upload your own vectors and rasters

> 200 public datasets

> 4000 new images every day

> 5 million images

> 60 petabytes of data

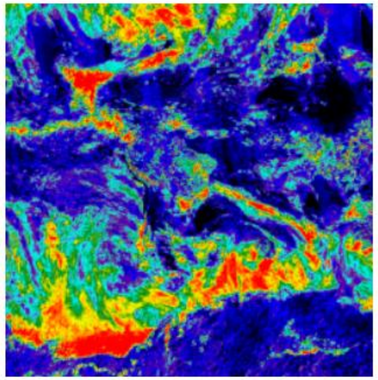
The Earth Engine Data Catalog

Earth Engine Data Catalog   Search   Русский   Войти

Datasets tagged climate in Earth Engine

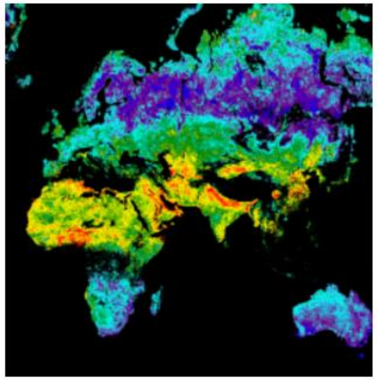
Filter list of datasets

Sentinel-5P NRTI CLOUD: Near Real-Time Cloud



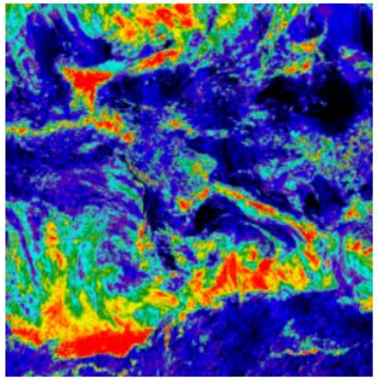
NRTI/L3_CLOUD This dataset provides near real-time high-resolution imagery of cloud parameters. The TROPOMI/S5P cloud properties retrieval is based on the OCRA and ROCINN algorithms currently being used in the operational GOME and GOME-2 products. OCRA retrieves the cloud fraction using measurements in the UV/VIS spectral ...

Sentinel-5P OFFL CH4: Offline Methane



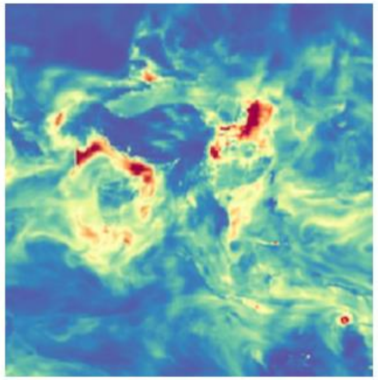
OFFL/L3_CH4 This dataset provides offline high-resolution imagery of methane concentrations. Methane (CH4) is, after carbon dioxide (CO2), the most important contributor to the anthropogenically enhanced greenhouse effect. Roughly three-quarters of methane emissions are anthropogenic and as such it is important to continue

Sentinel-5P OFFL CLOUD: Near Real-Time Cloud



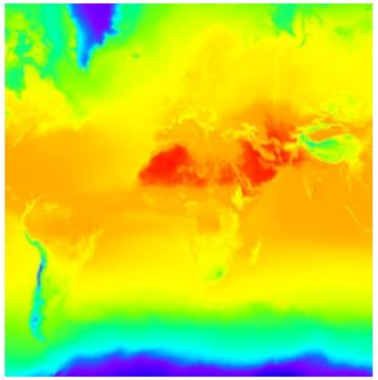
OFFL/L3_CLOUD This dataset provides offline high-resolution imagery of cloud parameters. The TROPOMI/S5P cloud properties retrieval is based on the OCRA and ROCINN algorithms currently being used in the operational GOME and GOME-2 products. OCRA retrieves the cloud fraction using measurements in the UV/VIS spectral regions ...

Copernicus Atmosphere Monitoring Service (CAMS) Global Near-Real-Time



The Copernicus Atmosphere Monitoring Service provides the capacity to continuously monitor the composition of the Earth's atmosphere at global and regional scales. The main global near-real-time production system is a data assimilation and forecasting suite providing two 5-day forecasts per day for aerosols and chemical ...

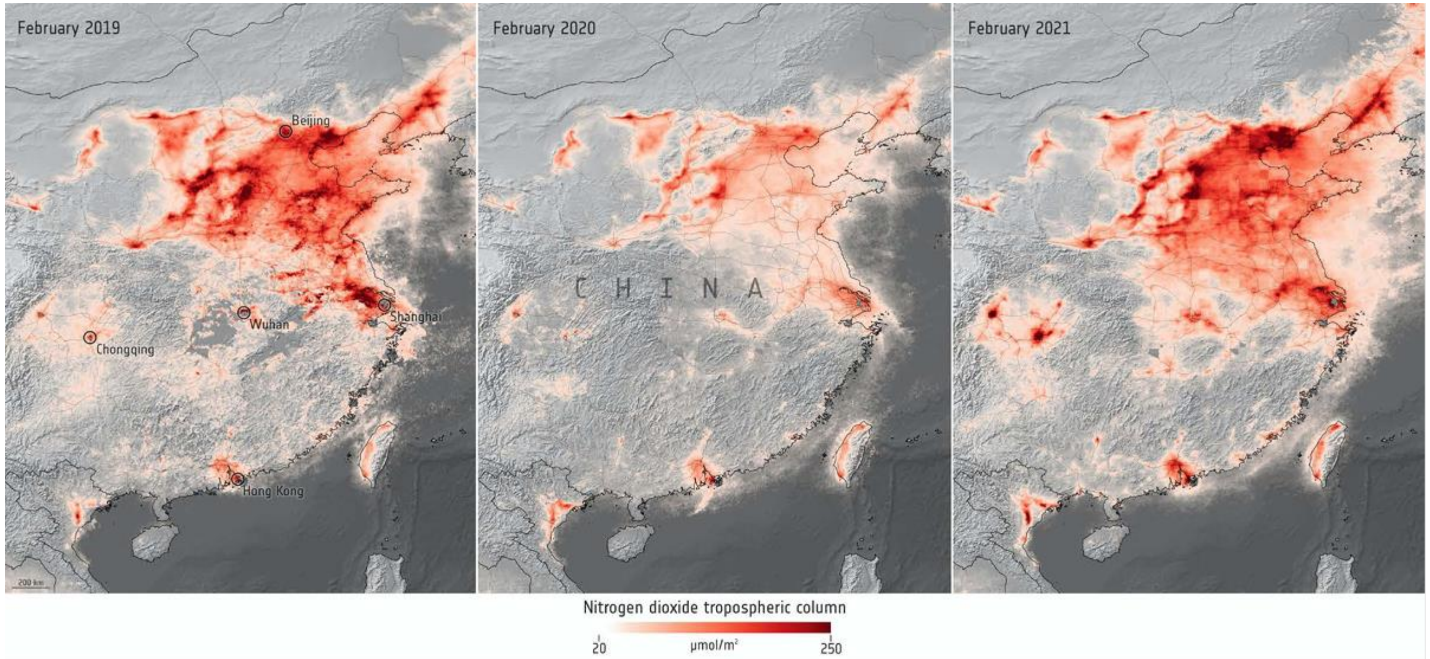
ERA5 Daily Aggregates - Latest Climate Reanalysis Produced by ECMWF / Copernicus Climate



ERA5 is the fifth generation ECMWF atmospheric reanalysis of the global climate. Reanalysis combines model data with observations from across the world into a globally complete and consistent dataset. ERA5 replaces its predecessor, the ERA-Interim reanalysis. ERA5 DAILY provides aggregated values for each day for ...

Sentinel-5

The Sentinel-5 mission is part of the European Earth Observation Programme "Copernicus" which is a coordinated and managed by the European Commission (EC). The space component of the Copernicus observation infrastructure is developed under the aegis of the European Space Agency (ESA).

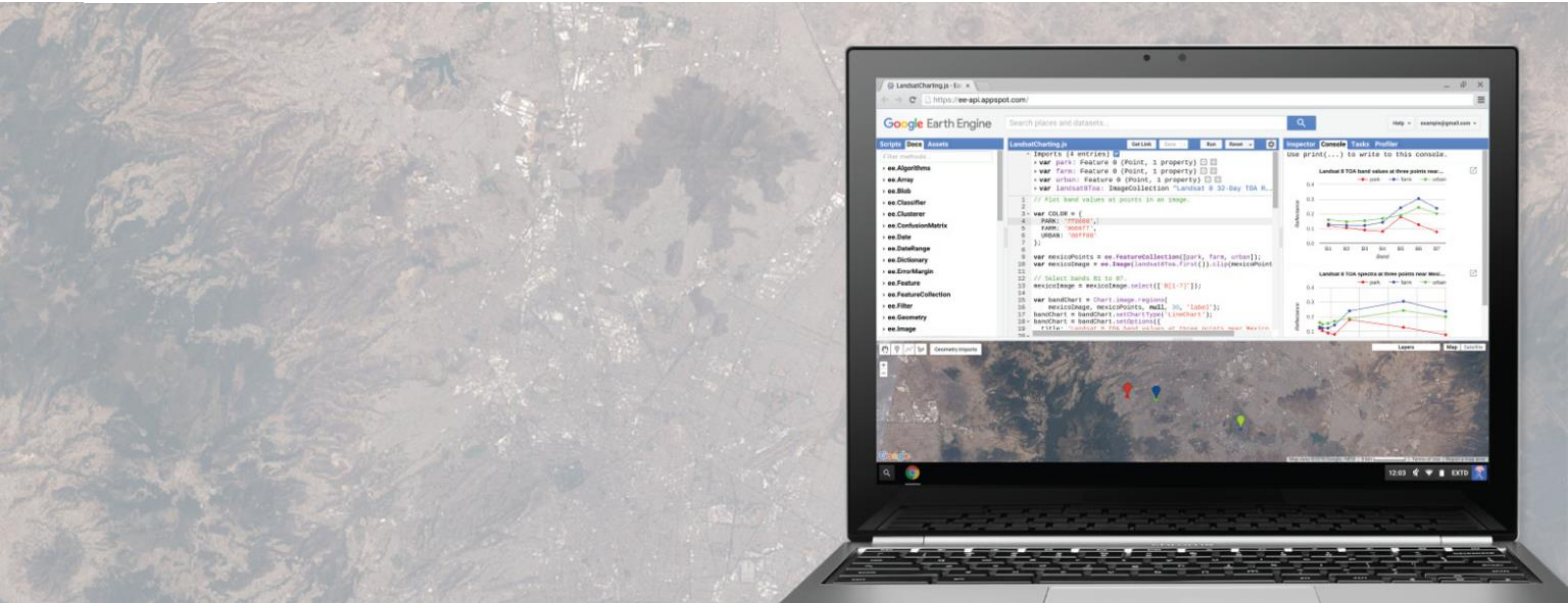


contains modified Copernicus Sentinel data (2019-21), processed by ESA

Sentinel-5 is focused on air quality and composition-climate interaction with the main data products being O_3, NO_2, SO_2, HCHO, CHOCHO and aerosols. Additionally Sentinel-5 will also deliver quality parameters for CO, CH_4, and stratospheric O_3 with daily global coverage for climate, air quality, and ozone/surface UV applications.

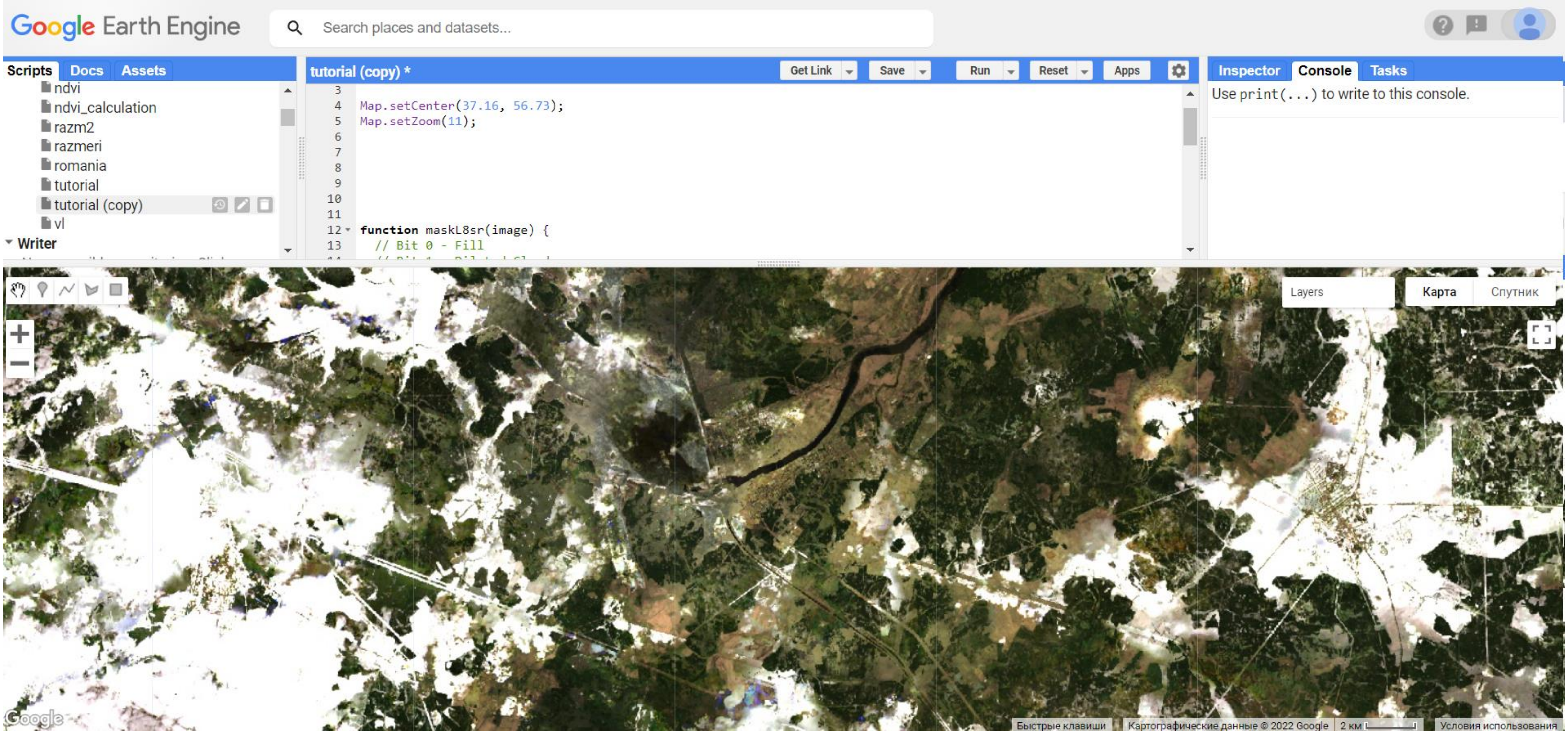
The Sentinel-5 mission consists of high resolution spectrometer system operating in the ultraviolet to shortwave infrared range with 7 different spectral bands: UV-1 (270-300nm), UV-2 (300-370nm), VIS (370-500nm), NIR-1 (685-710nm), NIR-2 (745-773nm), SWIR-1 (1590-1675nm) and SWIR-3 (2305-2385nm).

JavaScript code editor https://code.earthengine.google.com/

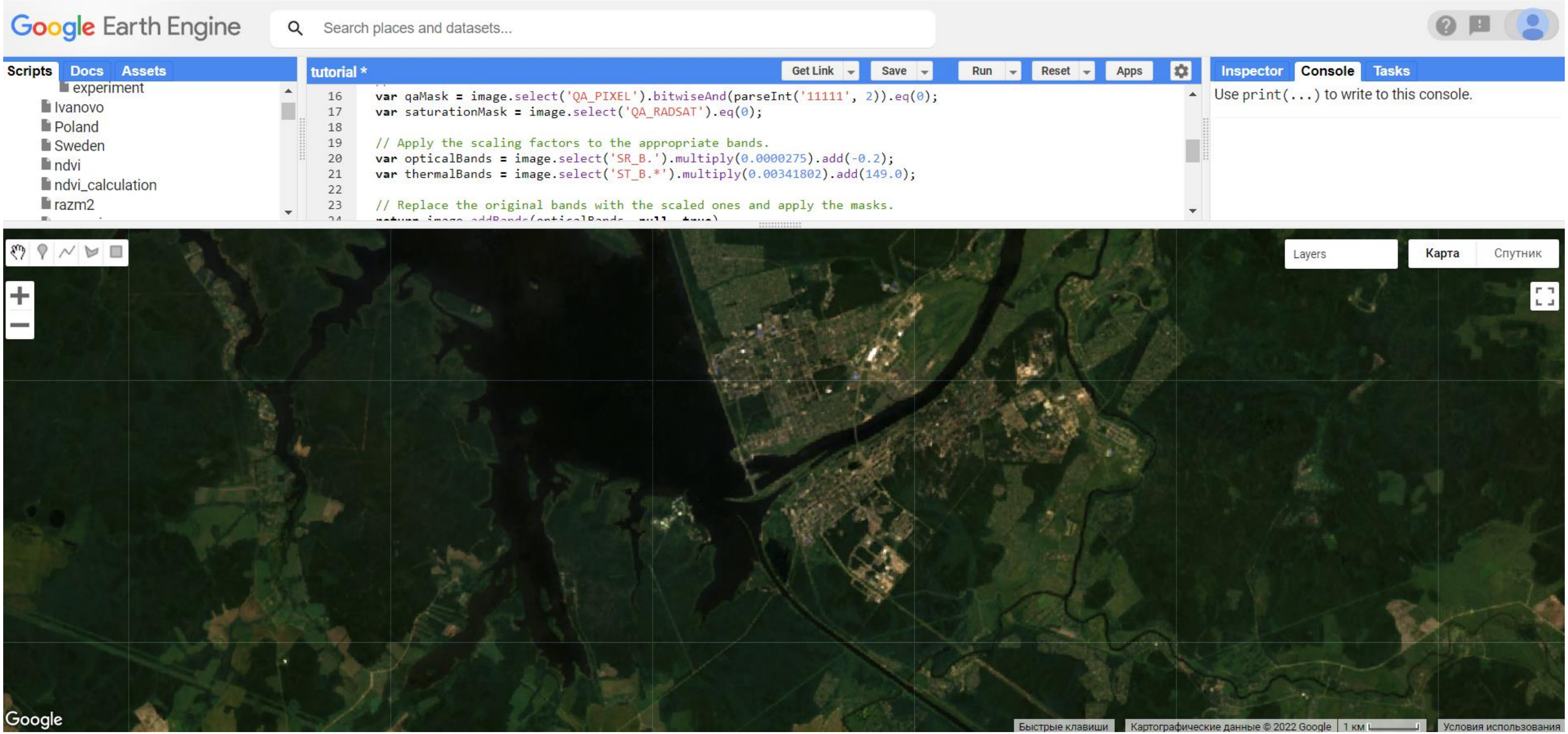


* python too!

Example of map



Cloud free map



Data Types and Geospatial Processing Functions

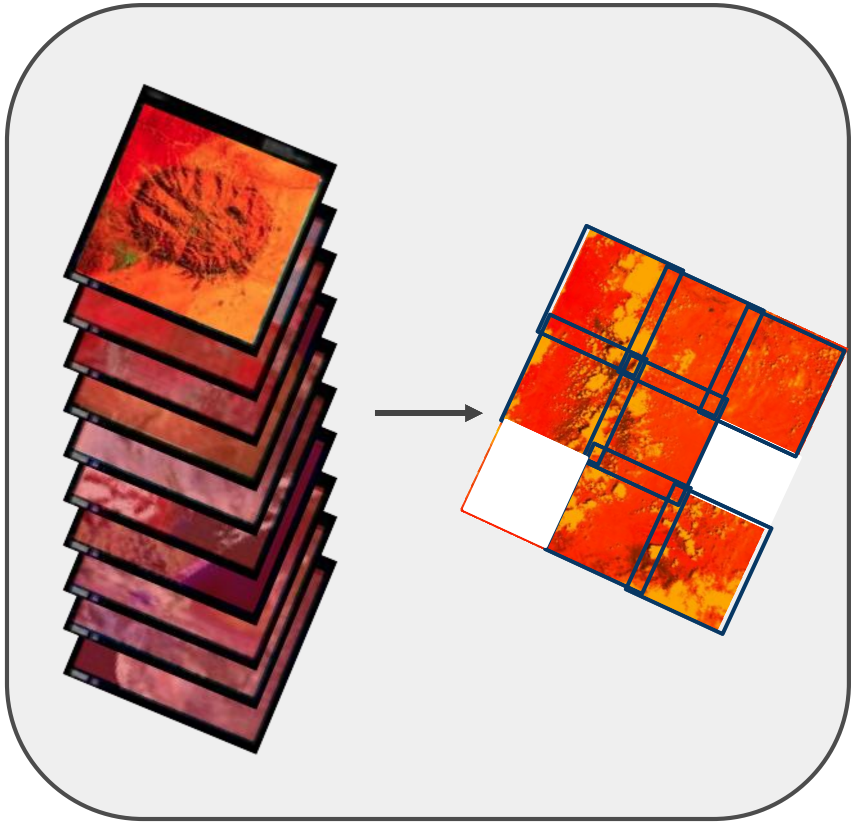
- **Image** - band math, clip, convolution, neighborhood, selection ...
- **Image Collection** - map, aggregate, filter, mosaic, sort ...
- **Feature** - buffer, centroid, intersection, union, transform ...
- **Feature Collection** - aggregate, filter, flatten, merge, sort ...
- **Filter** - by bounds, within distance, date, day-of-year, metadata ...
- **Reducer** - mean, linearRegression, percentile, histogram ....
- **Join** - simple, inner, outer, inverted ...
- **Kernel** - square, circle, gaussian, sobel, kirsch ...
- **Machine Learning -** CART, random forests, bayes, SVM, kmeans, cobweb ...
- **Projection** - transform, translate, scale ...

over 1000 data types and operators, and growing!

Reduce

Aggregate everything in a collection

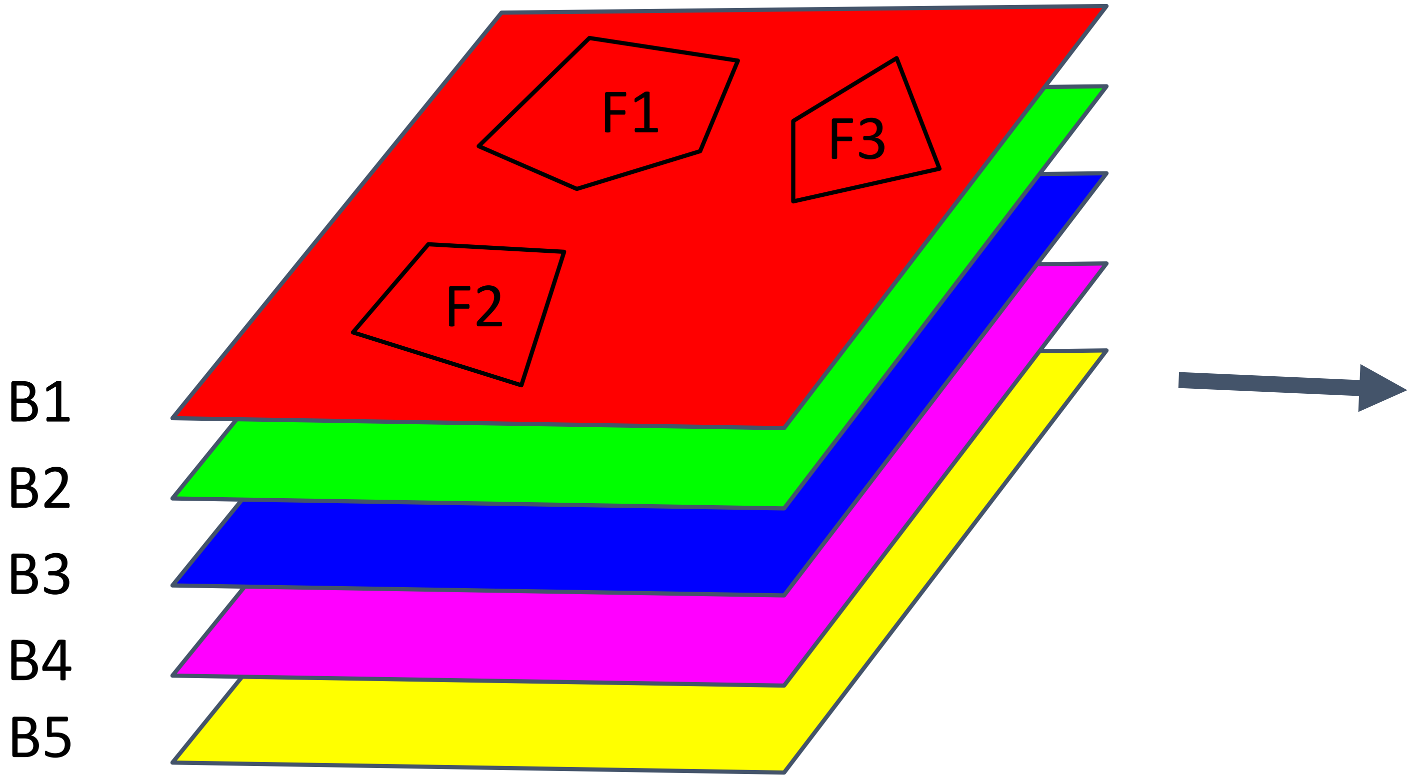
"Reduction"



Examples

Summed area over all features
Median-pixel composite
Train a classifier

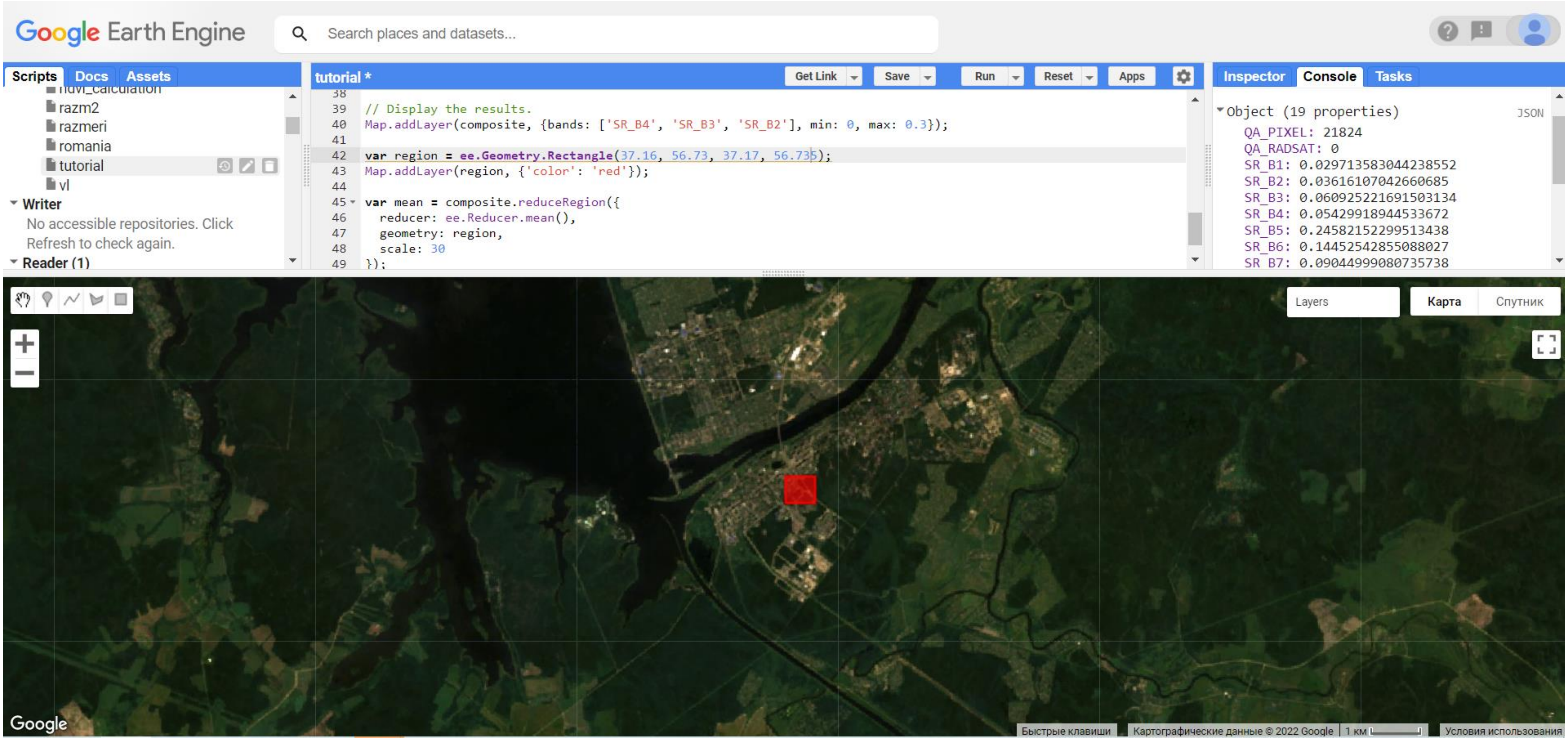
Reduce Regions



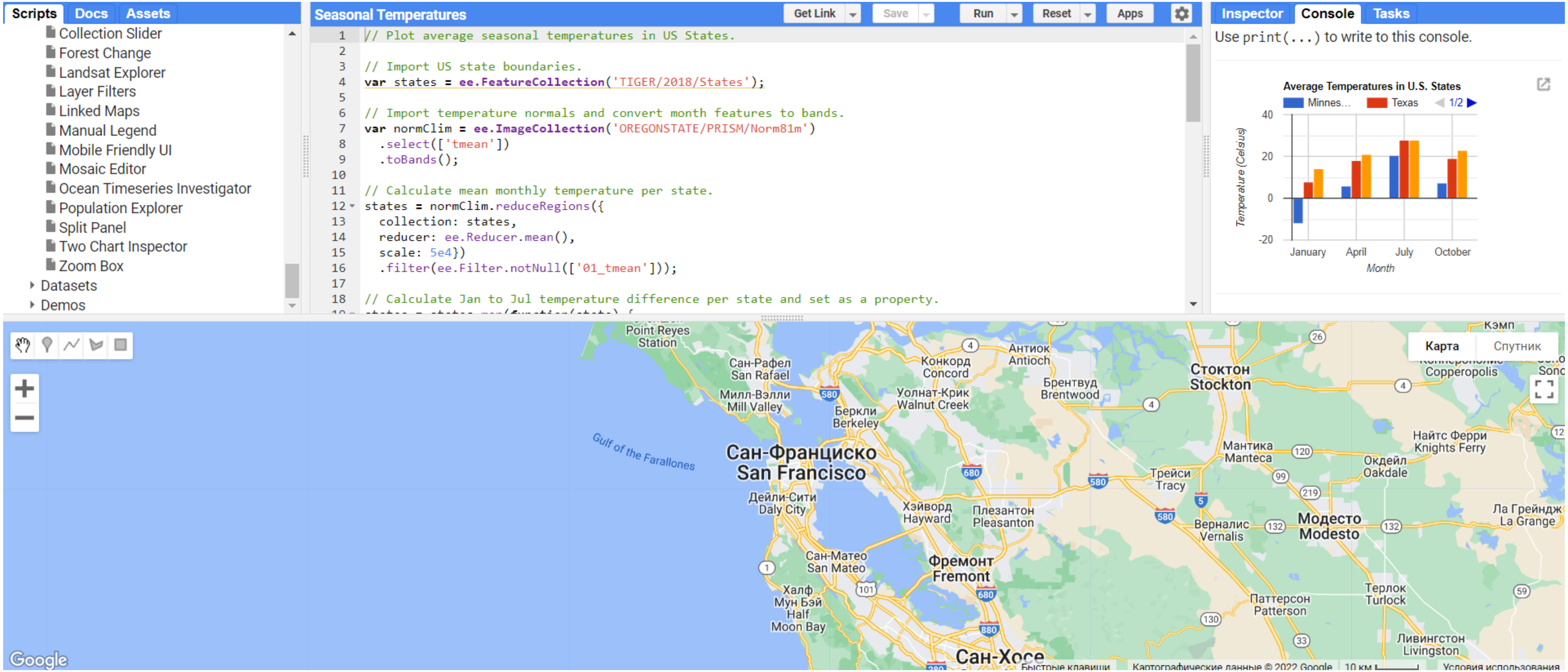
FeatureCollection

	B1	B2	B3	B4	B5
F1					
F2					
F3					

Reducer in action



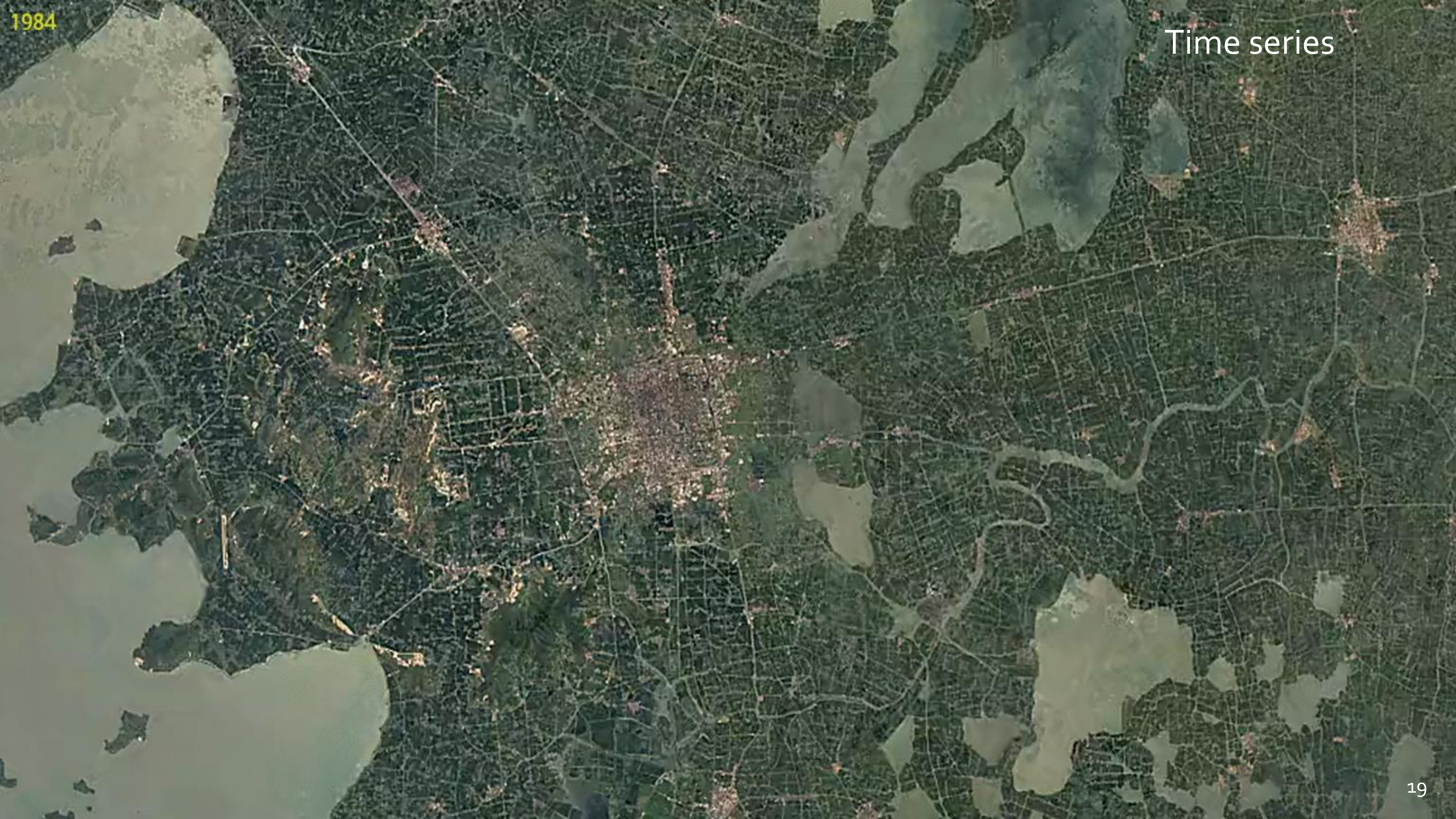
Editor options



Examples

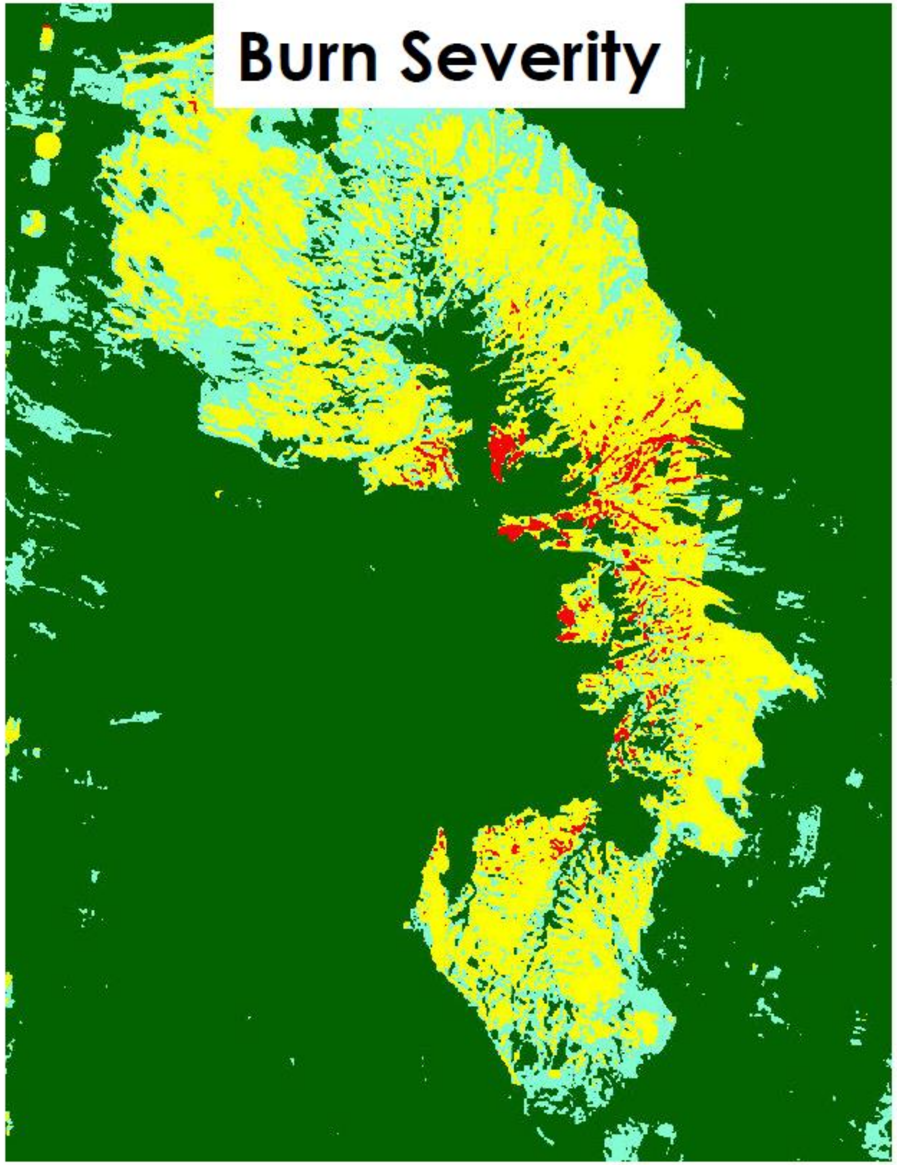
Time series

Time series

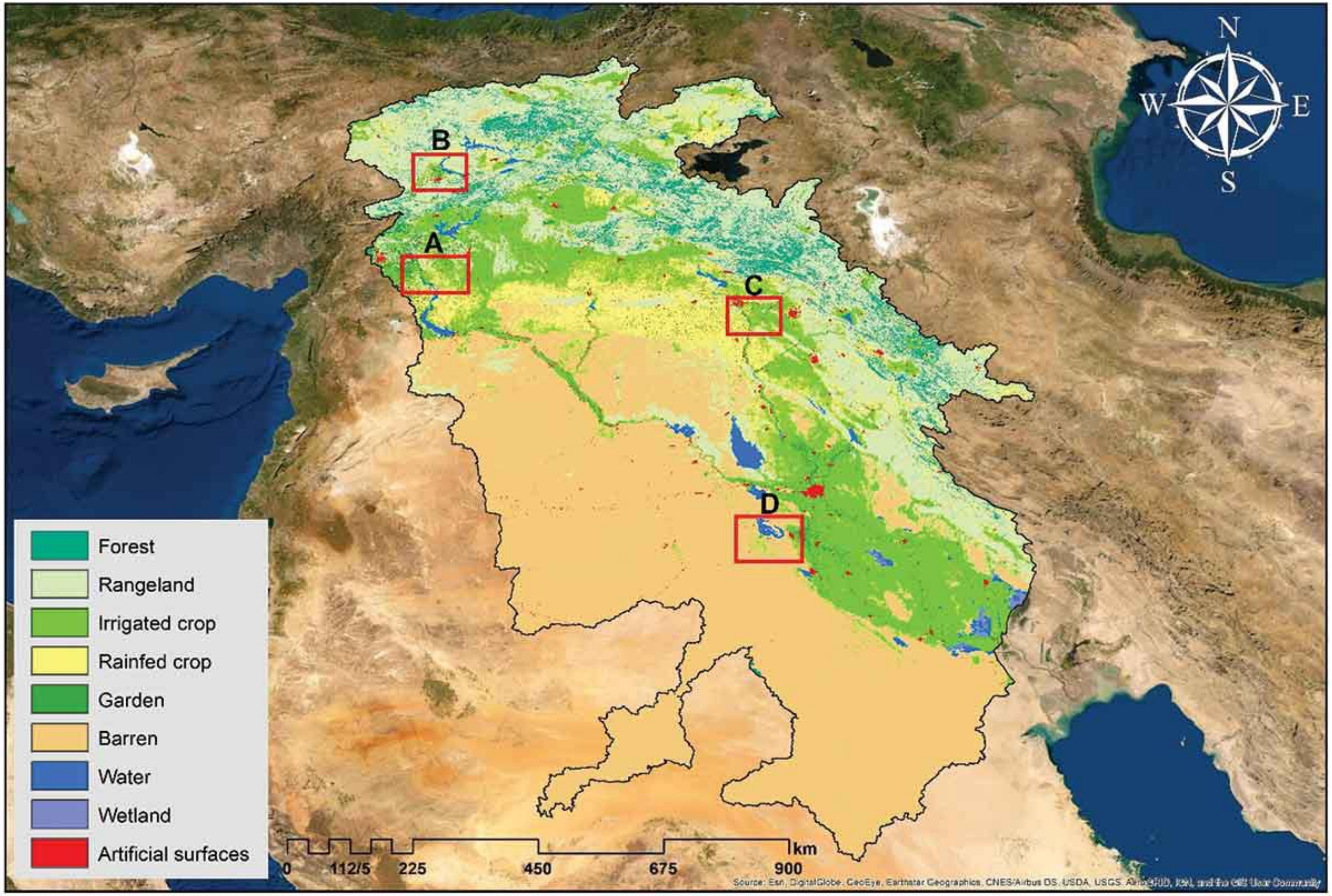


Observations for Pre-and Post-Fire Monitoring

- Burned area uses imagery to assess the extent of impacts on vegetation for a particular fire event.
- Burn severity compares burned area information to pre-fire imagery to assess relative magnitude of burn impacts.

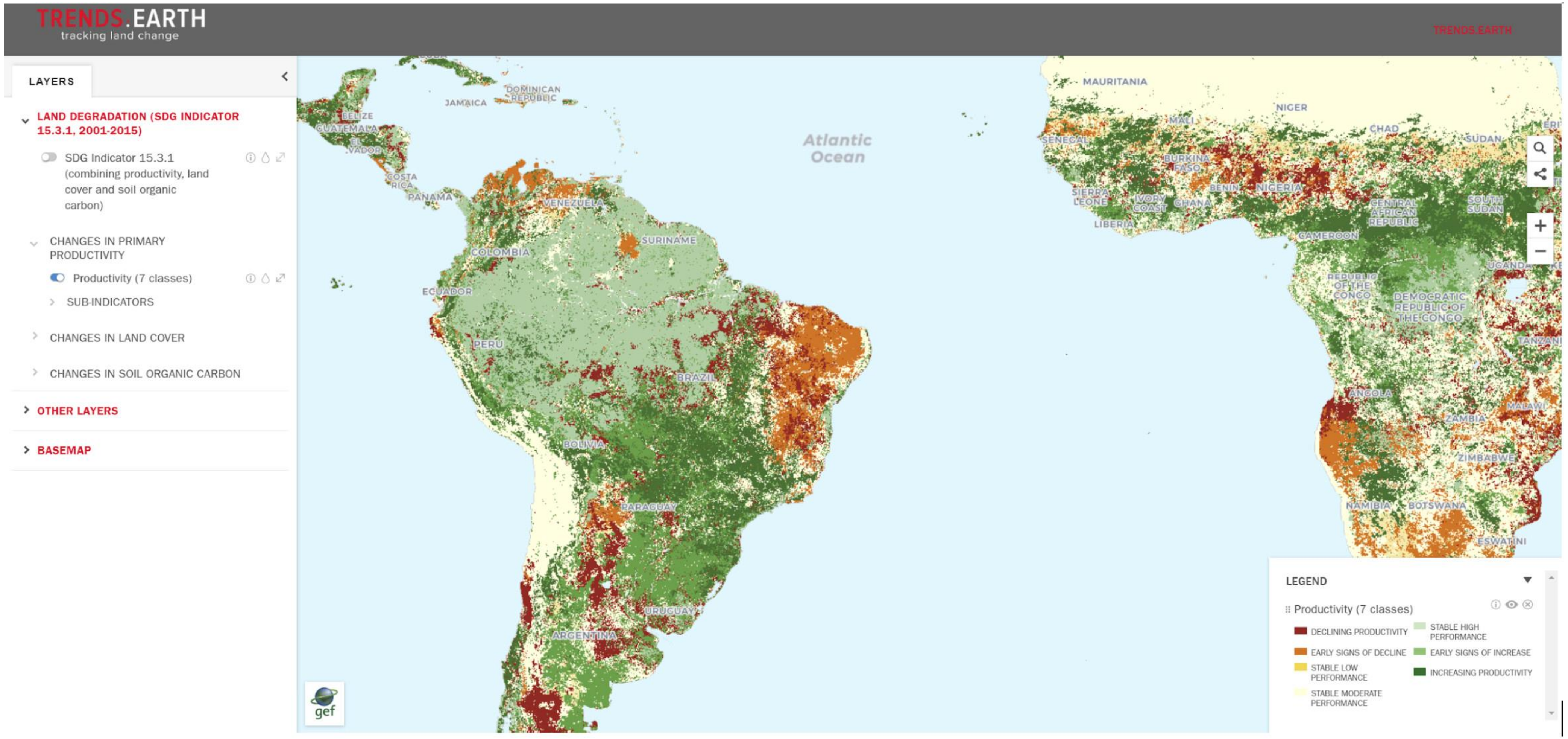


Soil Mapping and Classification



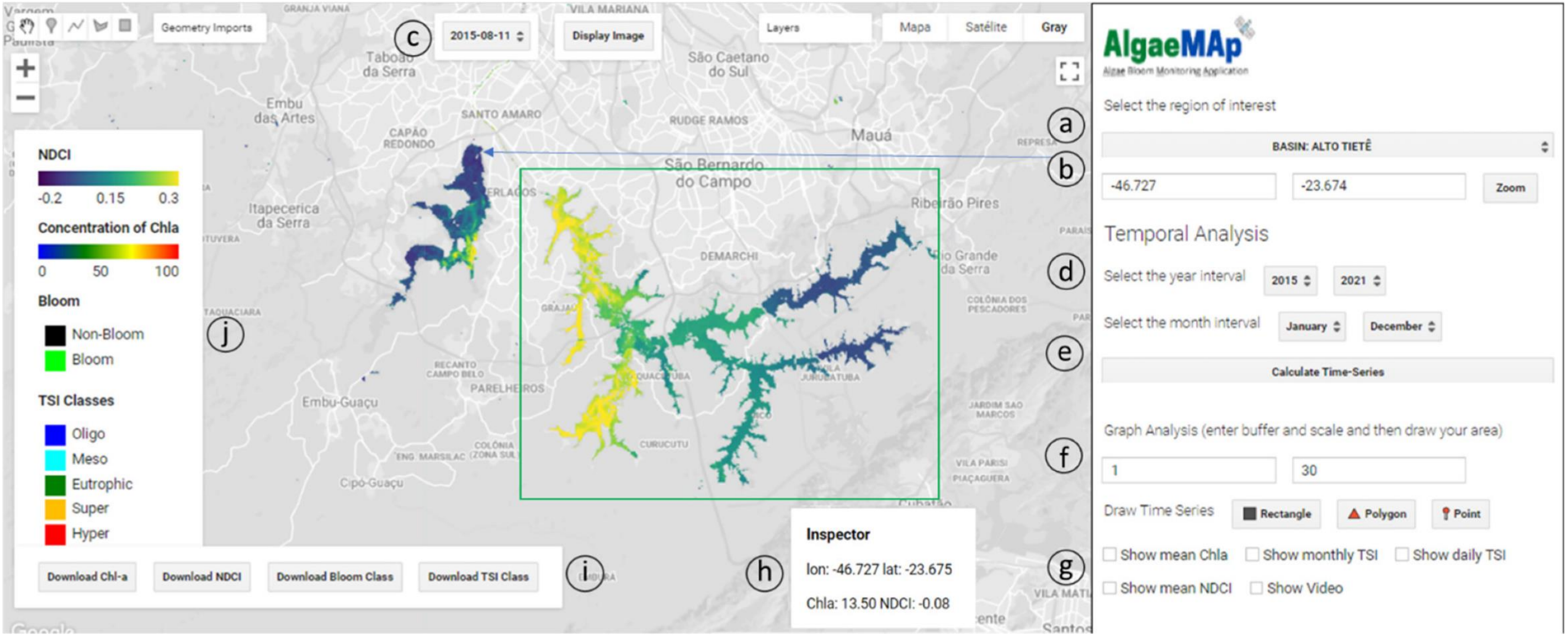
Trends.Earth

Through this project, we will develop a cloud-based platform dedicated to mapping land degradation which identifies potential land restoration opportunities at national to regional scale, allowing communities to prioritize areas to protect, manage, and restore in order to achieve land degradation neutrality.



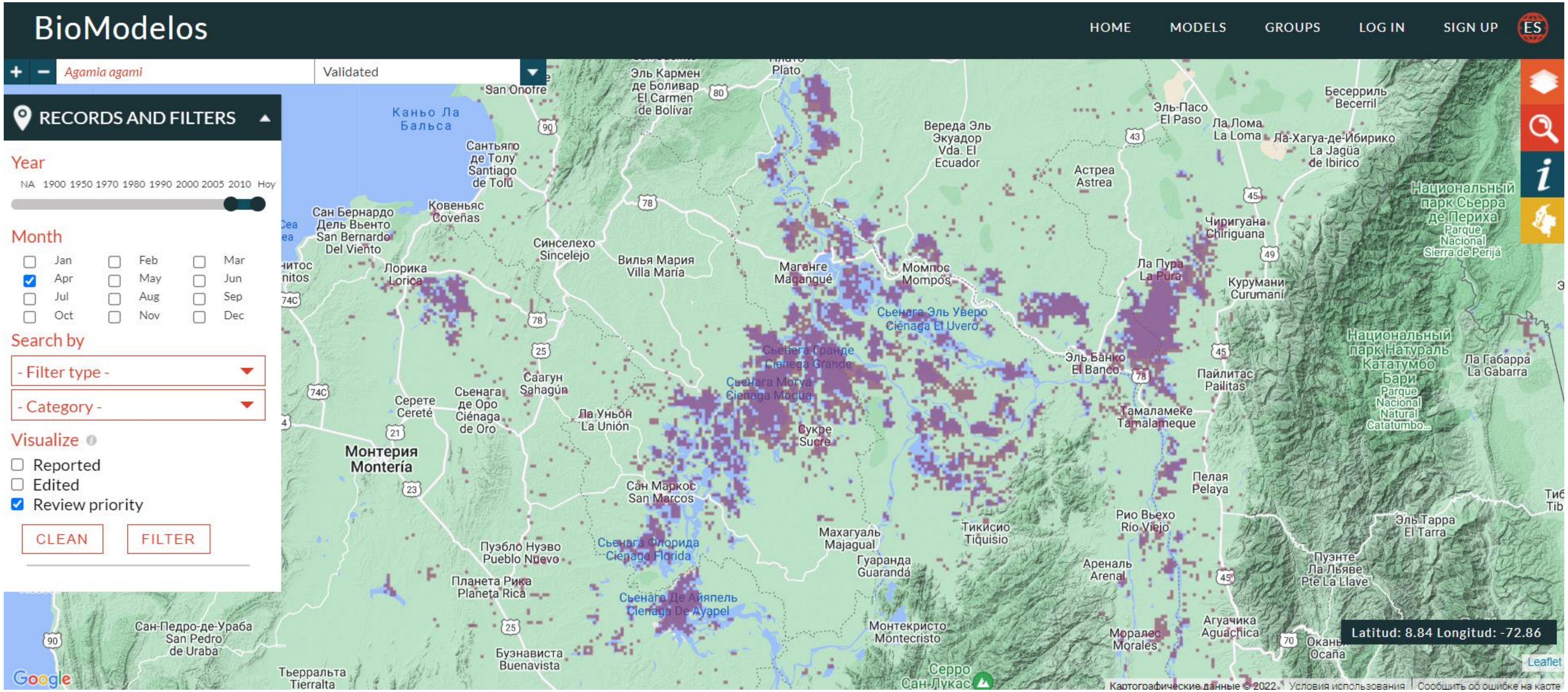
Alert System for Algal Bloom

Algae blooms occur when certain kinds of algae grow very quickly, forming patches, or "blooms," in the water. These blooms can be indicators of water degradation and emit powerful toxins that can endanger human and animal health.



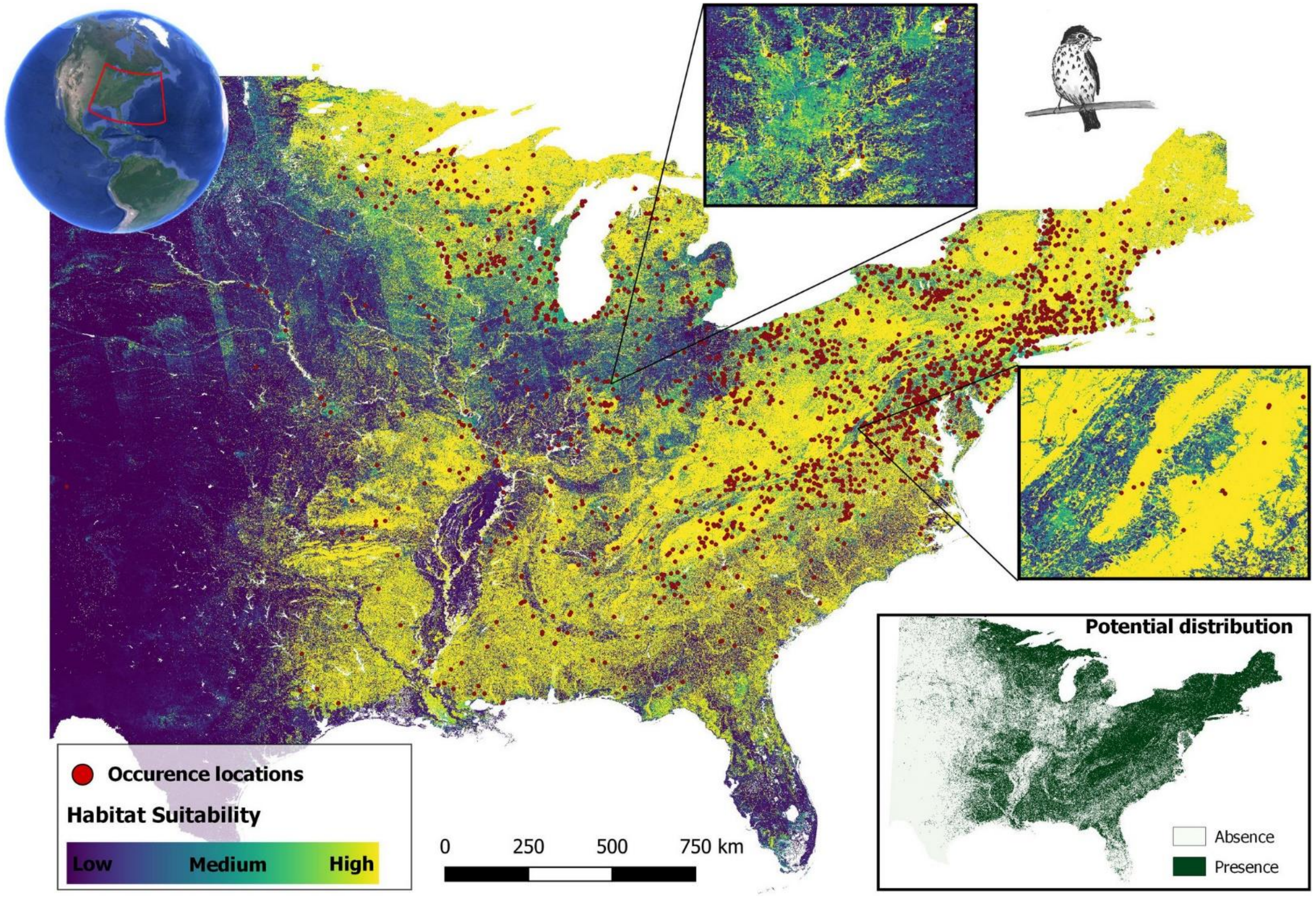
BioModelos: a collaborative online system to map species distributions

BioModelos, a modeling approach supported by an online system and a core team, whereby a network of experts contributes to the development of species distribution models by assessing the quality of occurrence data, identifying potentially limiting environmental variables, establishing species' accessible areas and validating qualitatively modeling predictions. Models developed through BioModelos become publicly available once validated by experts, furthering their use in conservation applications.



Essential Biodiversity Variables - ScaleUp

We implemented a workflow for species distribution modelling in GEE that includes importing species occurrence data into the GEE platform, selecting and preparing predictor variables, and performing model fitting with spatial or temporal split-block cross-validation techniques.



Fire Danger Warning System

This **app** could be used to visualize the **Australia Forest** Fire using Landsat 8 and Sentinel 2 composites

Wildfire Susceptibility Prediction Fusion with Remote Sensing Data of Different Spatial Resolutions

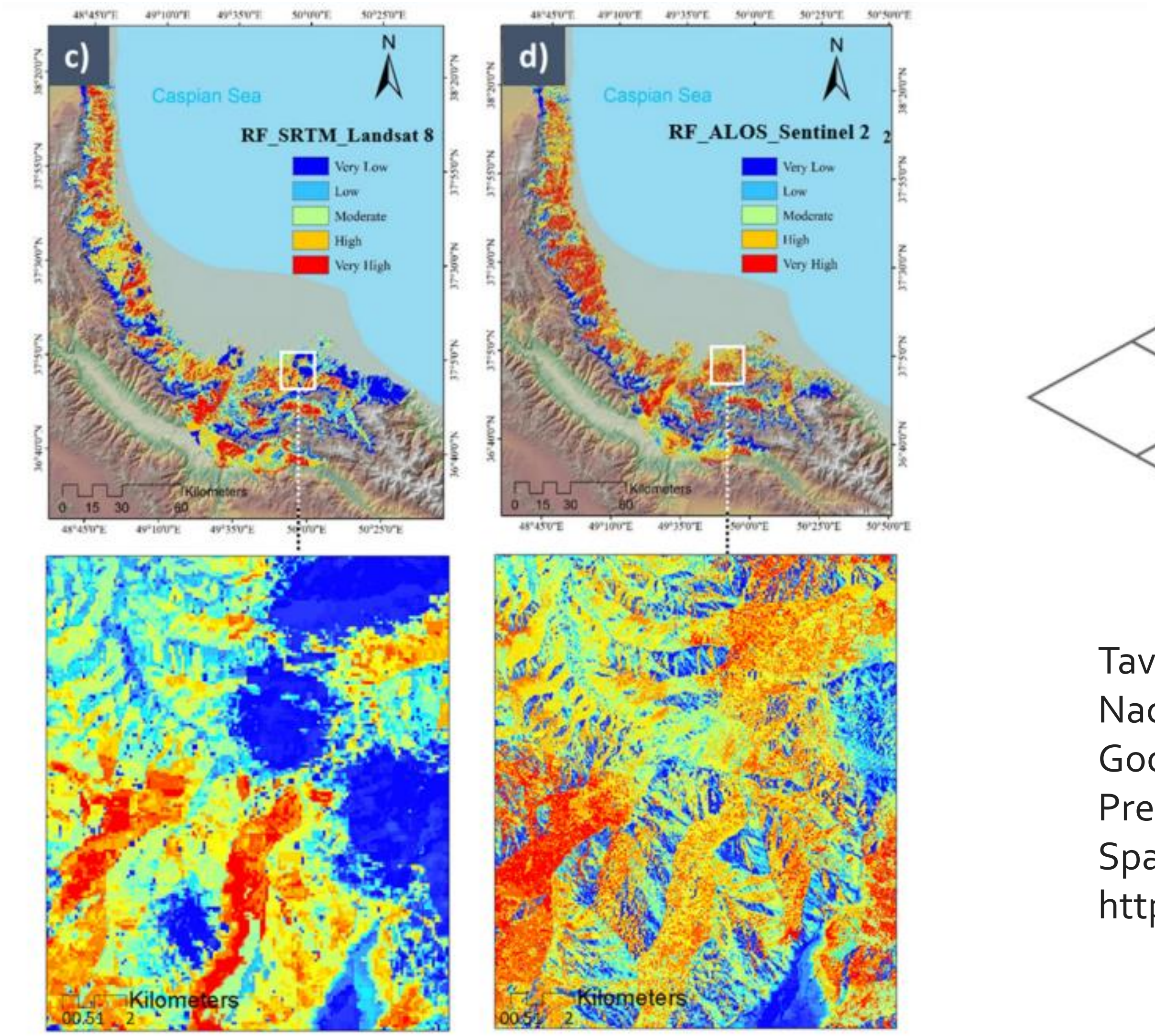
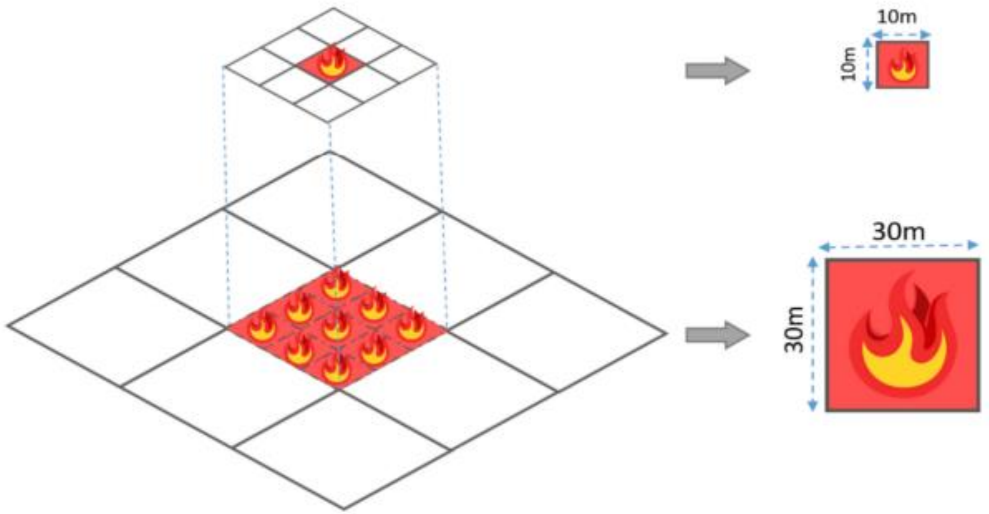


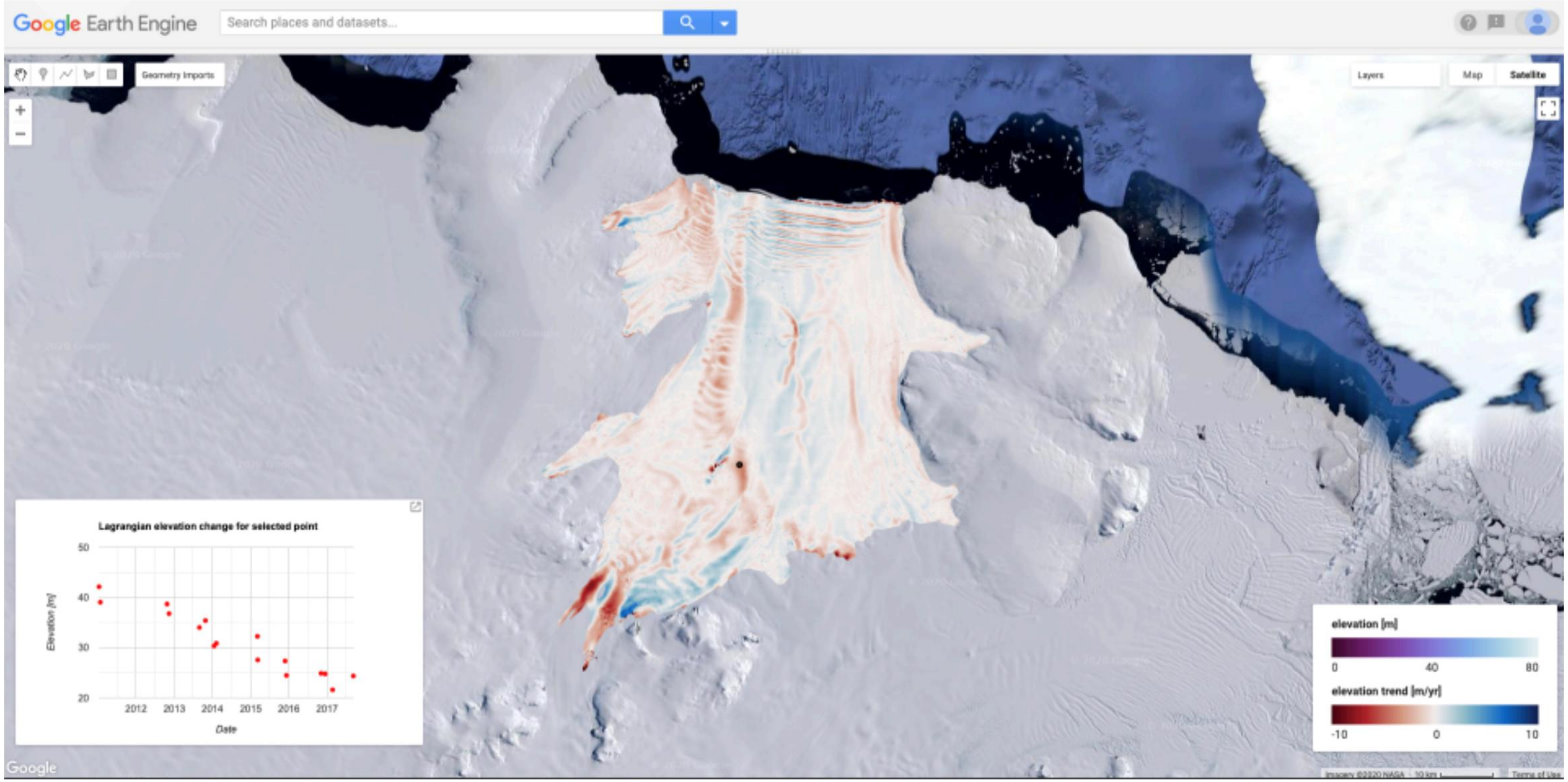
Figure 10. The resulting wildfire susceptibility prediction (WSP) maps from (**a**) SVM_{30}, (**b**) SVM_{10}, (**c**) RF_{10}, (**d**) RF_{30}.



Tavakkoli Piralilou, S.; Einali, G.; Ghorbanzadeh, O.; Nachappa, T.G.; Gholamnia, K.; Blaschke, T.; Ghamisi, P. A Google Earth Engine Approach for Wildfire Susceptibility Prediction Fusion with Remote Sensing Data of Different Spatial Resolutions. *Remote Sens.* **2022**, *14*, 672. https://doi.org/10.3390/rs14030672

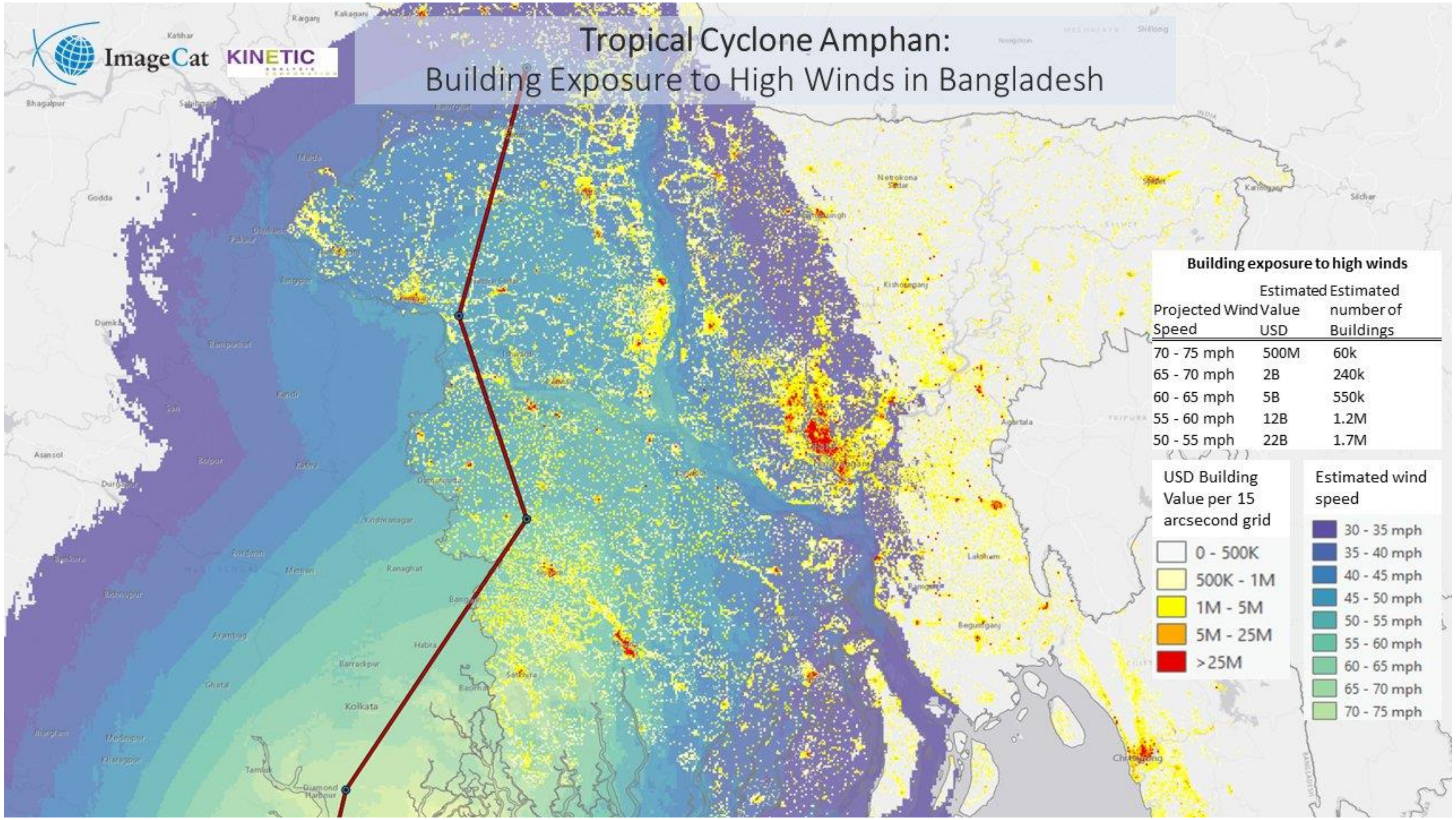
GEE-interface for analyzing basal melt over Dotson ice shelf

A platform to analyze and demonstrate the use of satellite data over Antarctic ice shelves. This will allow insight in the surface, subsurface and basal conditions of these ice shelves, which are major sources of uncertainty in future sea level projections from Antarctica.



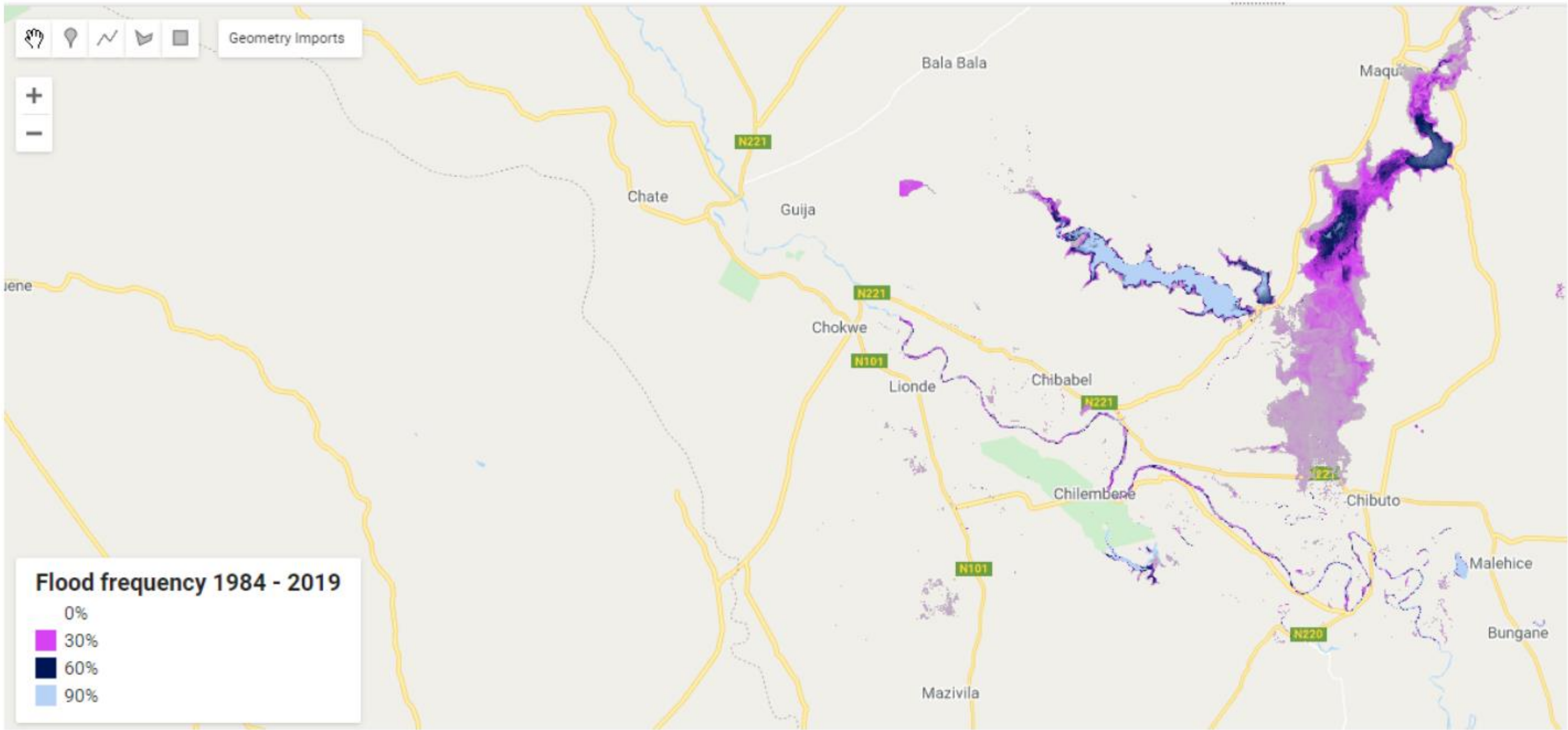
ImageCat - Disaster Forecasting, Mitigation and Response

ImageCat, Inc., is a Long Beach, CA. company specializing in innovative solutions to natural hazard risk assessment and management. ImageCat offers solutions towards building a more resilient future using earth observation data and geospatial technologies.



Supporting water and disaster risk management in Mozambique

The project HydroPC focused on co-development and application of innovative data technologies and comprehensive training of beneficiaries to support water and disaster risk management in Mozambique.



Earth Engine Costs

It's free

An excerpt from their website:

"Why is Google working on Earth Engine?

Google's mission is to organize the world's information and make it universally accessible and useful. In line with this mission, Earth Engine organizes geospatial information and makes it available for analysis. More generally, Google strives to make the world a better place through the use of technology. Earth Engine's technical infrastructure powers humanitarian, scientific, and environmental initiatives which Google is proud to support."

When to use EE?

EE Benefits:

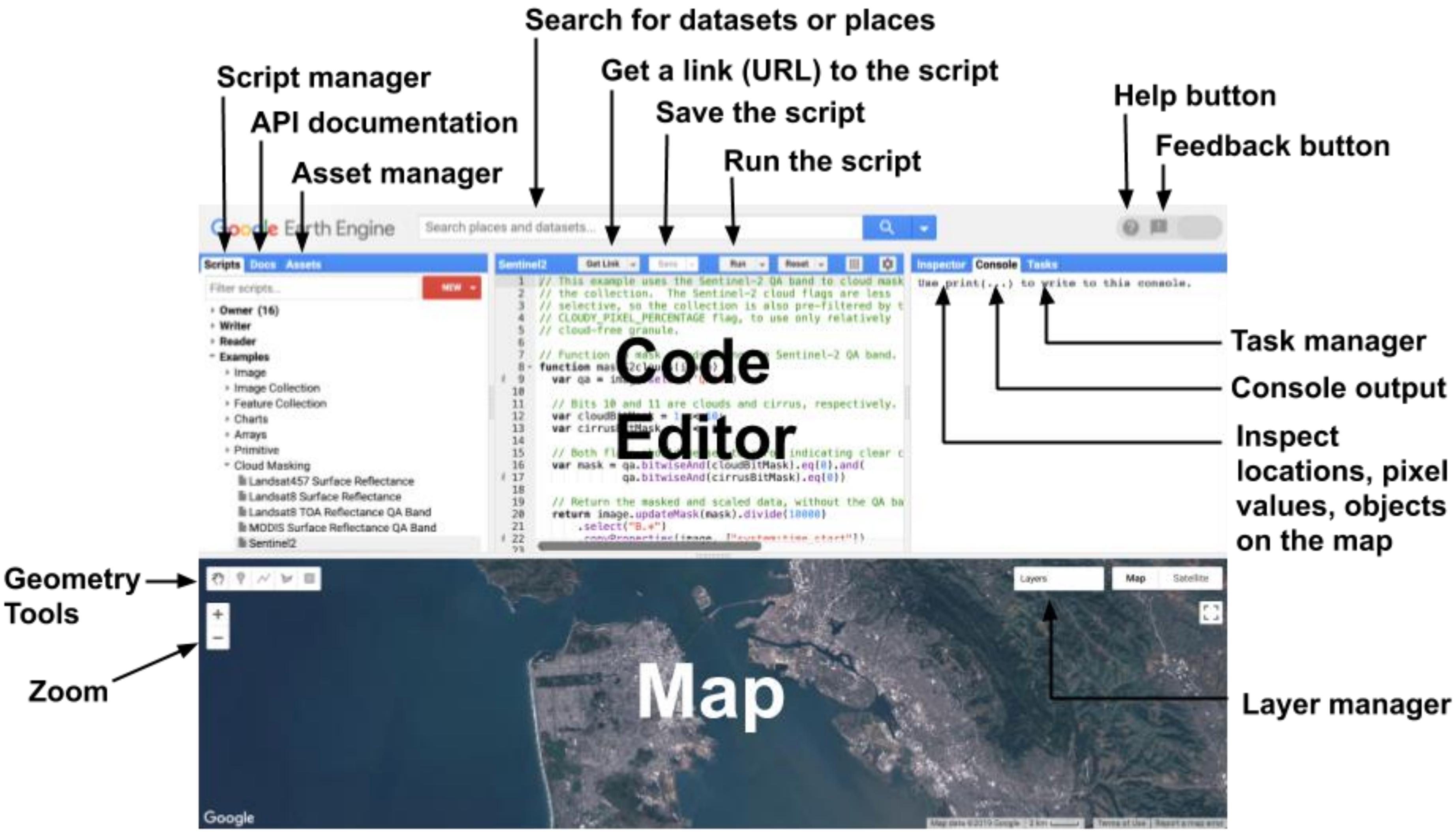
Good for projects that requires:

- Data coverage for a large region
- Extensive data library
- High speed, intensive processing capacity
- Advanced raster processing tools

EE Limitations:

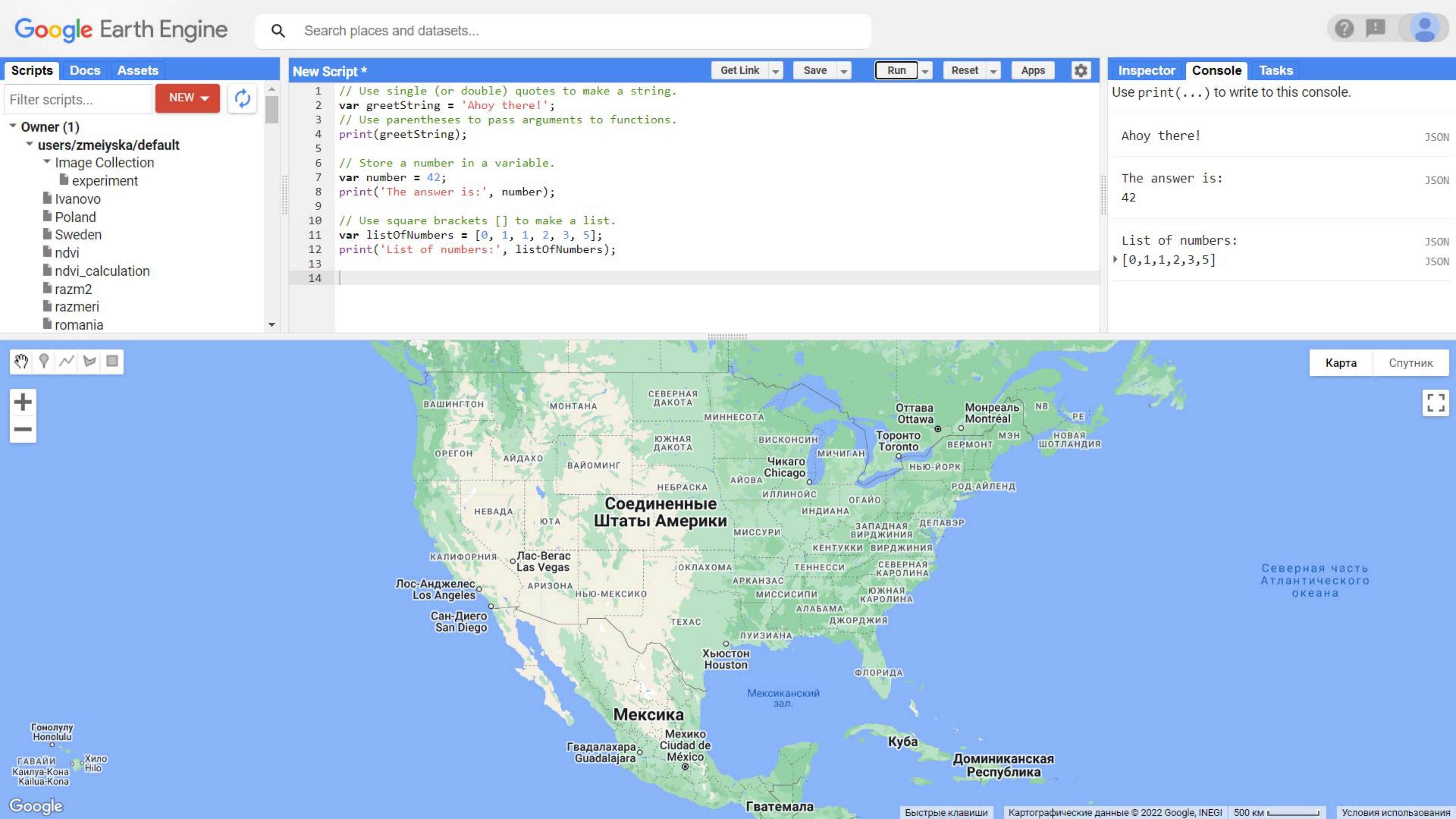
- Better suited to image analyses than vector-based analyses
- Analysis based on pixel spatial relations are harder to complete (because of the processing on multiple CPU's). Image segmentation and hydrologic modeling options are limited

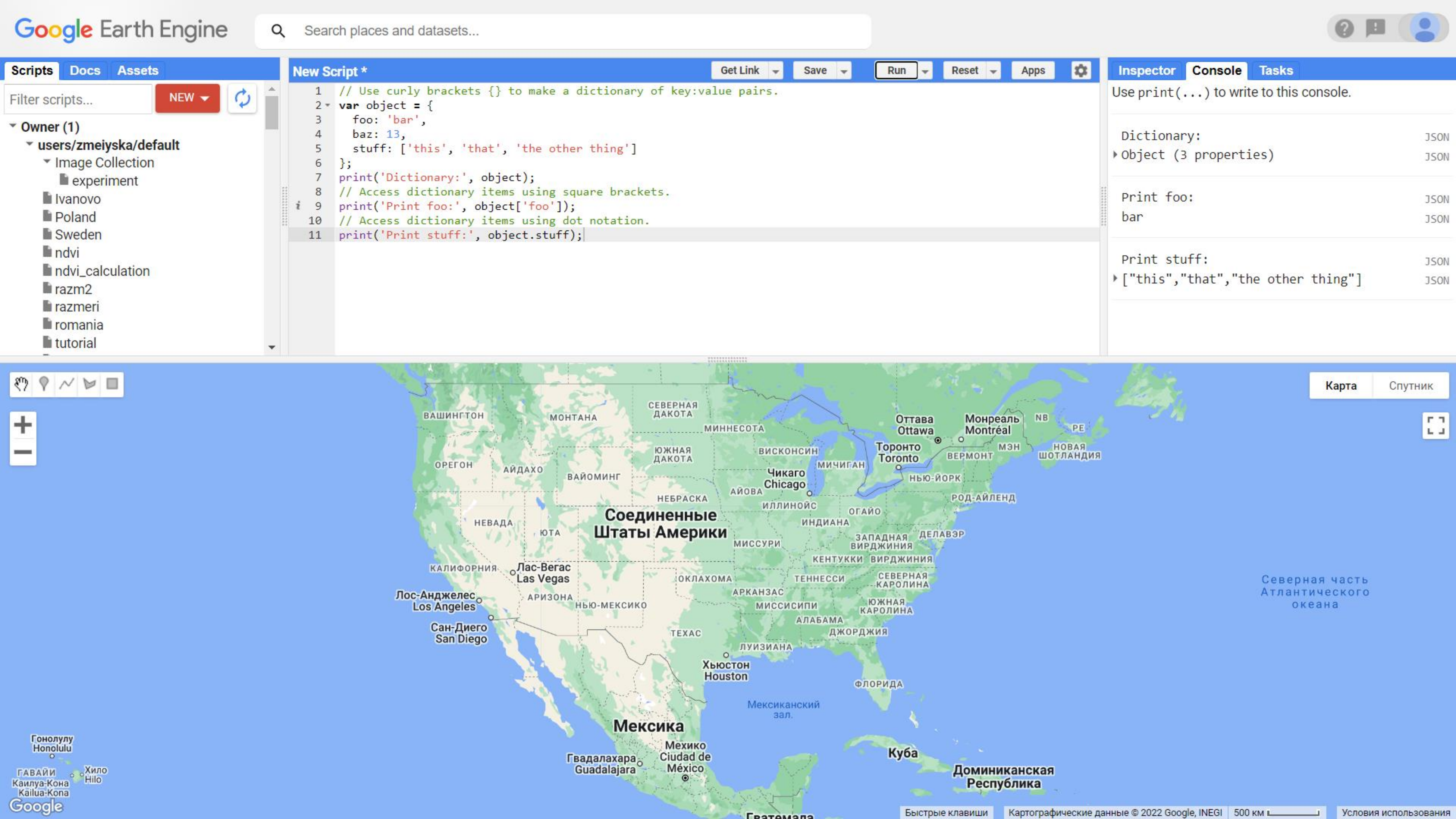
https://developers.google.com/earth-engine/guides/playground#:~:text=The%20Earth%20Engine%20(EE)%20Code,JavaScript%20code%20editor

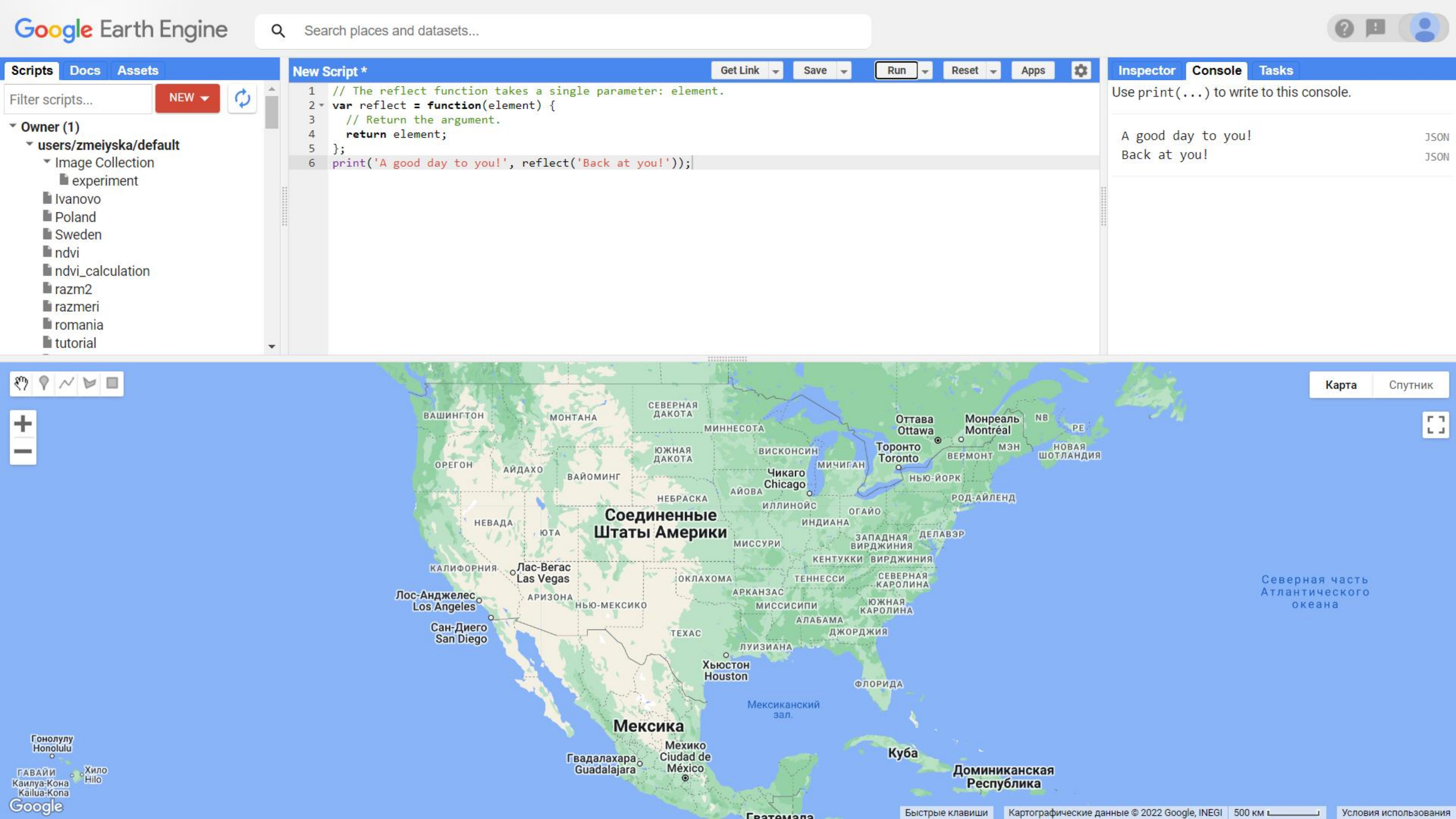


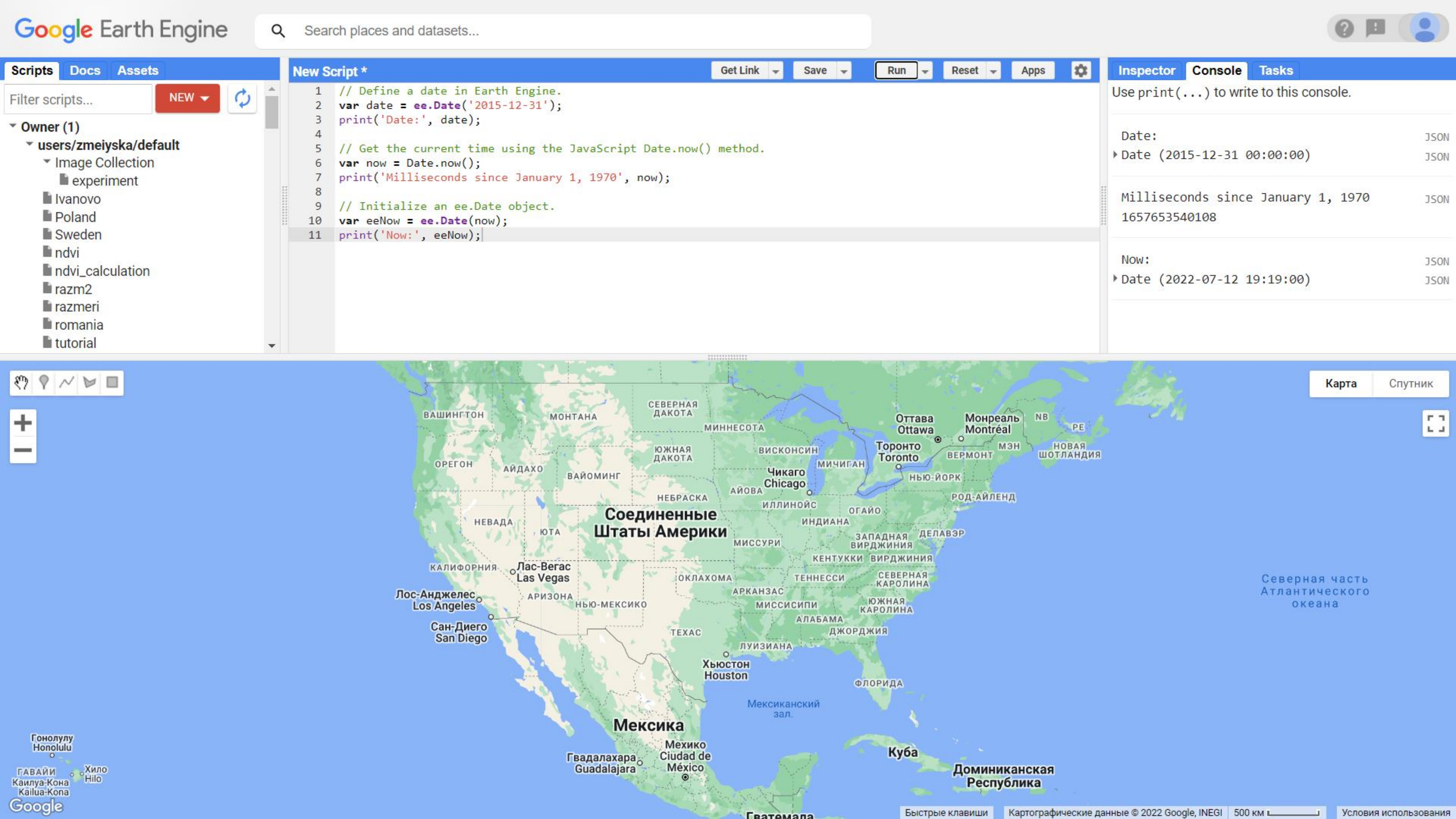
Basics

https://code.earthengine.google.com/6b24f55c3c60991a09352ce552fd310a


Google Earth Engine
Search places and datasets...
Scripts
Docs
Assets
Filter scripts...
NEW
Owner (1)
users/zmeiyska/default
Image Collection
experiment
Ivanovo
Poland
Sweden
ndvi
ndvi_calculation
razm2
razmeri
romania
New Script *
Get Link
Save
Run
Reset
Apps
// Use single (or double) quotes to make a string.
var greetString = 'Ahoy there!';
// Use parentheses to pass arguments to functions.
print(greetString);
// Store a number in a variable.
var number = 42;
print('The answer is:', number);
// Use square brackets [] to make a list.
var listOfNumbers = [0, 1, 1, 2, 3, 5];
print('List of numbers:', listOfNumbers);
Inspector
Console
Tasks
Use print(...) to write to this console.
Ahoy there!
JSON
The answer is:
42
JSON
List of numbers:
JSON
[0,1,1,2,3,5]
JSON
Карта
Спутник
Соединенные Штаты Америки
Мексика
Северная часть Атлантического океана
Мексиканский зал.
Куба
Доминиканская Республика
Гватемала
Быстрые клавиши
Картографические данные © 2022 Google, INEGI
500 км
Условия использования

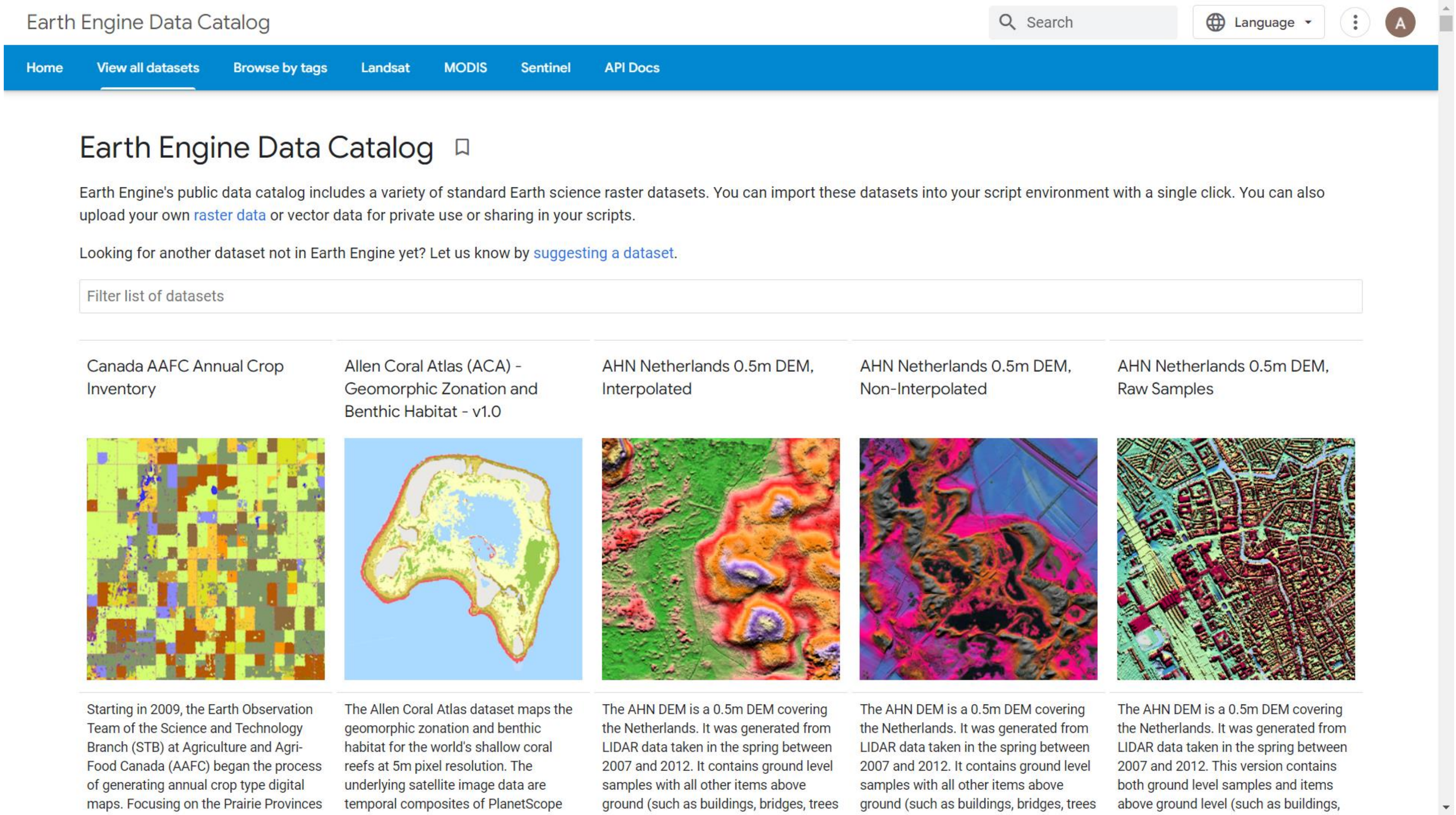

Google Earth Engine
Search places and datasets...
Scripts Docs Assets
Filter scripts...
NEW
Owner (1)
users/zmeiyska/default
Image Collection
experiment
Ivanovo
Poland
Sweden
ndvi
ndvi_calculation
razm2
razmeri
romania
tutorial
New Script *
Get Link
Save
Run
Reset
Apps
// Use curly brackets {} to make a dictionary of key:value pairs.
var object = {
  foo: 'bar',
  baz: 13,
  stuff: ['this', 'that', 'the other thing']
};
print('Dictionary:', object);
// Access dictionary items using square brackets.
print('Print foo:', object['foo']);
// Access dictionary items using dot notation.
print('Print stuff:', object.stuff);
Inspector Console Tasks
Use print(...) to write to this console.
Dictionary:
JSON
Object (3 properties)
JSON
Print foo:
JSON
bar
JSON
Print stuff:
JSON
["this","that","the other thing"]
JSON
Карта
Спутник
Соединенные Штаты Америки
Мексика
Куба
Доминиканская Республика
Мексиканский зал.
Северная часть Атлантического океана
Оттава Ottawa
Монреаль Montréal
Торонто Toronto
Чикаго Chicago
Лас-Вегас Las Vegas
Лос-Анджелес Los Angeles
Сан-Диего San Diego
Хьюстон Houston
Мехико Ciudad de México
Гвадалахара Guadalajara
Гонолулу Honolulu
Быстрые клавиши
Картографические данные © 2022 Google, INEGI
500 км
Условия использования


Google Earth Engine
Search places and datasets...
Scripts
Docs
Assets
Filter scripts...
NEW
Owner (1)
users/zmeiyska/default
Image Collection
experiment
Ivanovo
Poland
Sweden
ndvi
ndvi_calculation
razm2
razmeri
romania
tutorial
New Script *
Get Link
Save
Run
Reset
Apps
// The reflect function takes a single parameter: element.
var reflect = function(element) {
// Return the argument.
return element;
};
print('A good day to you!', reflect('Back at you!'));
Inspector
Console
Tasks
Use print(...) to write to this console.
A good day to you!
JSON
Back at you!
JSON
Карта
Спутник
Соединенные Штаты Америки
Мексика
Северная часть Атлантического океана
Мексиканский зал.
Куба
Доминиканская Республика
Быстрые клавиши
Картографические данные © 2022 Google, INEGI
500 км
Условия использования


Google Earth Engine
Search places and datasets...
Scripts
Docs
Assets
Filter scripts...
NEW
Owner (1)
users/zmeiyska/default
Image Collection
experiment
Ivanovo
Poland
Sweden
ndvi
ndvi_calculation
razm2
razmeri
romania
tutorial
New Script *
Get Link
Save
Run
Reset
Apps
// Define a date in Earth Engine.
var date = ee.Date('2015-12-31');
print('Date:', date);
// Get the current time using the JavaScript Date.now() method.
var now = Date.now();
print('Milliseconds since January 1, 1970', now);
// Initialize an ee.Date object.
var eeNow = ee.Date(now);
print('Now:', eeNow);
Inspector
Console
Tasks
Use print(...) to write to this console.
Date:
Date (2015-12-31 00:00:00)
Milliseconds since January 1, 1970
1657653540108
Now:
Date (2022-07-12 19:19:00)
JSON
Карта
Спутник
Соединенные Штаты Америки
Мексика
Северная часть Атлантического океана
Мексиканский зал.
Куба
Доминиканская Республика
Оттава Ottawa
Монреаль Montréal
Торонто Toronto
Чикаго Chicago
Лас-Вегас Las Vegas
Лос-Анджелес Los Angeles
Сан-Диего San Diego
Хьюстон Houston
Гвадалахара Guadalajara
Мехико Ciudad de México
Гонолулу Honolulu
Хило Hilo
Каилуа-Кона Kailua-Kona
Быстрые клавиши
Картографические данные © 2022 Google, INEGI
500 км
Условия использования

Data Catalog

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

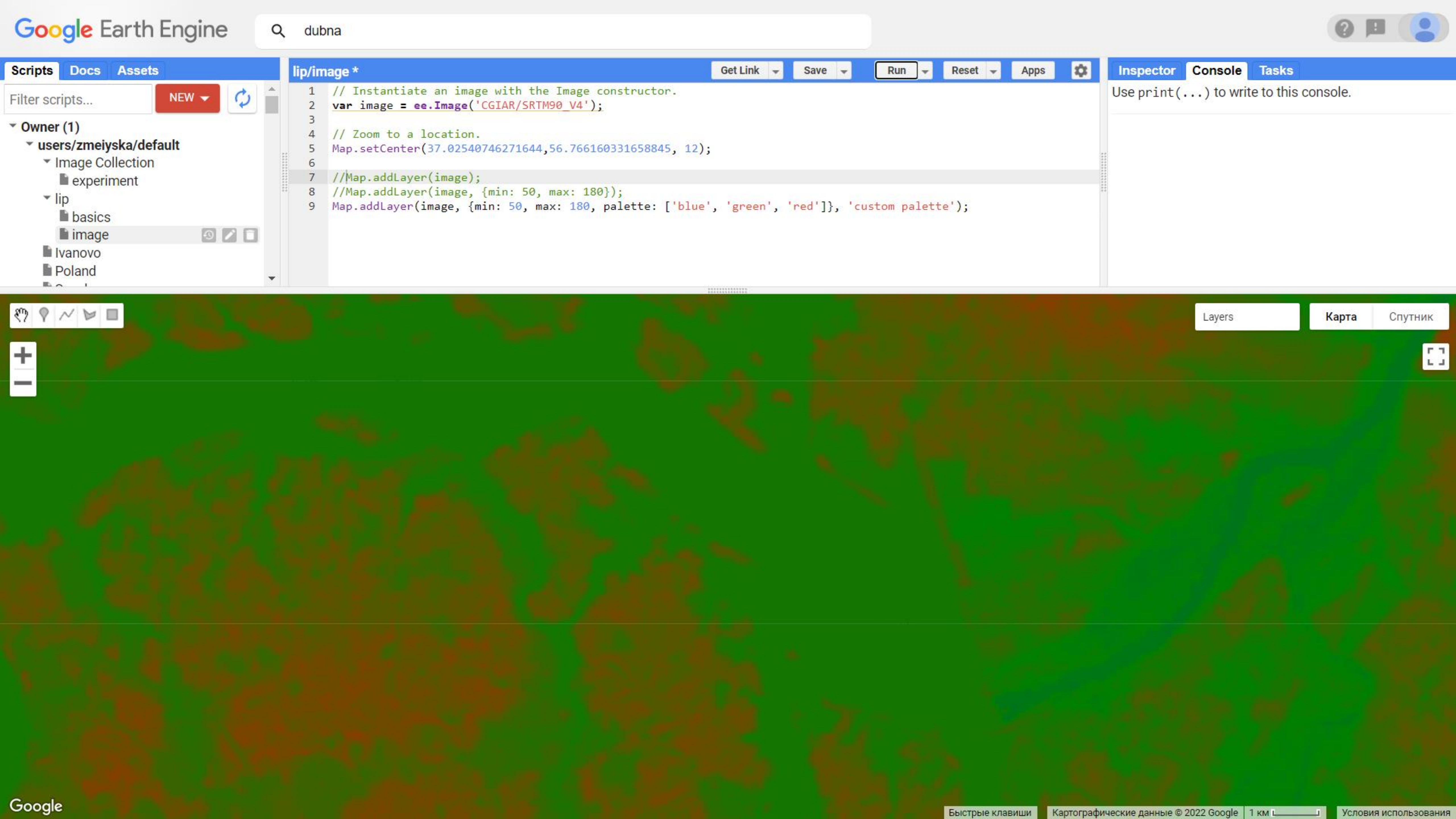


Images

https://developers.google.com/earth-engine/datasets/catalog/CGIAR_SRTM90_V4?hl=en

https://code.earthengine.google.com/31f672aec5b2a3f8bc7d1fbf23450935

```
// Instantiate an image with the Image constructor.
var image = ee.Image('CGIAR/SRTM90_V4');

// Zoom to a location.
Map.setCenter(37.02540746271644,56.766160331658845, 12);

Map.addLayer(image);
//Map.addLayer(image, {min: 50, max: 180});
//Map.addLayer(image, {min: 50, max: 180, palette: ['blue', 'green', 'red']}, 'custom palette');
```


Google Earth Engine
dubna
Scripts
Docs
Assets
Filter scripts...
NEW
Owner (1)
users/zmeiyska/default
Image Collection
experiment
lip
basics
image
Ivanovo
Poland
lip/image *
Get Link
Save
Run
Reset
Apps
// Instantiate an image with the Image constructor.
var image = ee.Image('CGIAR/SRTM90_V4');
// Zoom to a location.
Map.setCenter(37.02540746271644,56.766160331658845, 12);
//Map.addLayer(image);
//Map.addLayer(image, {min: 50, max: 180});
Map.addLayer(image, {min: 50, max: 180, palette: ['blue', 'green', 'red']}, 'custom palette');
Inspector
Console
Tasks
Use print(...) to write to this console.
Layers
Карта
Спутник
Google
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

Images computation

https://code.earthengine.google.com/a4b27505dfd223c7f5878e97ca42d06b

https://developers.google.com/earth-engine/apidocs/ee-terrain-slope

```
// Instantiate an image with the Image constructor.
var image = ee.Image('CGIAR/SRTM90_V4');

// Apply an algorithm to an image.
var slope = ee.Terrain.slope(image);

// Zoom to a location.
Map.setCenter(37.02540746271644,56.766160331658845, 12);

Map.addLayer(slope, {min: 0, max :5}, 'slope');
```

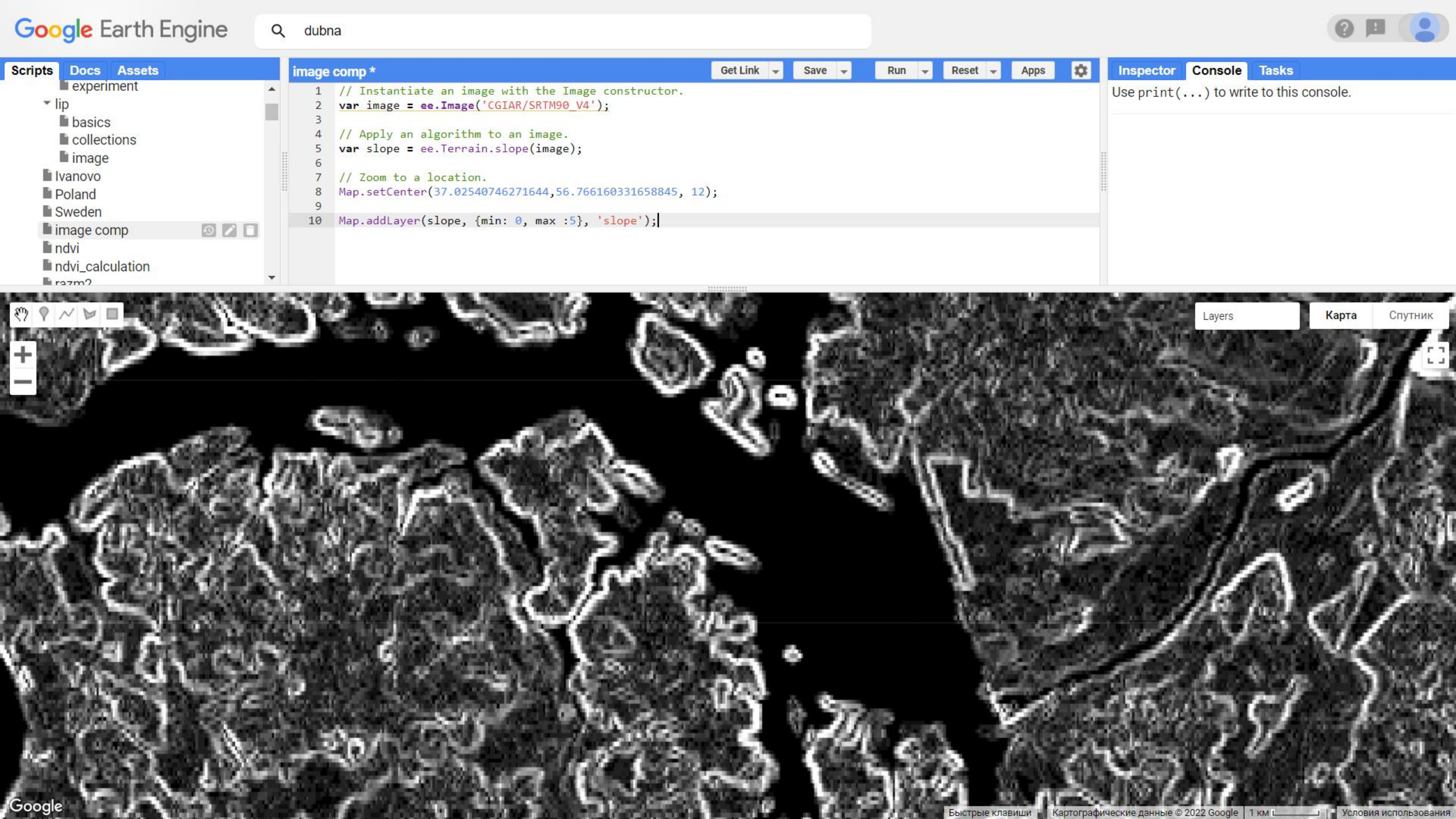

Google Earth Engine
dubna
Scripts
Docs
Assets
experiment
lip
basics
collections
image
Ivanovo
Poland
Sweden
image comp
ndvi
ndvi_calculation
image comp *
Get Link
Save
Run
Reset
Apps
// Instantiate an image with the Image constructor.
var image = ee.Image('CGIAR/SRTM90_V4');
// Apply an algorithm to an image.
var slope = ee.Terrain.slope(image);
// Zoom to a location.
Map.setCenter(37.02540746271644,56.766160331658845, 12);
Map.addLayer(slope, {min: 0, max :5}, 'slope');
Inspector
Console
Tasks
Use print(...) to write to this console.
Layers
Карта
Спутник
Google
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

Image statistics

https://code.earthengine.google.com/37d0c570ad54bf3d0232386d868be168

```
// Instantiate an image with the Image constructor.
var image = ee.Image('CGIAR/SRTM90_V4');

// Zoom to a location.
Map.setCenter(37.02540746271644,56.766160331658845, 12);
var p1 = ee.Geometry.Point(37.02540746271644,56.766160331658845);
var p2 = ee.Geometry.Point(37.03502049982581,56.770864050772836);

var mean = image.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: p1,
  scale: 90
});

print(mean);

var mean = image.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: p2,
  scale: 90
});

print(mean);

Map.addLayer(image, {min: 50, max: 180, palette: ['blue', 'green', 'red']}, 'custom palette');
Map.addLayer(p1);
Map.addLayer(p2);
```

Google Earth Engine

dubna

Scripts | Docs | Assets

- experiment
- lip
  - basics
  - collections
  - image
  - image comp
  - image statistics
- Ivanovo
- Poland
- Sweden
- ndvi
- ndvi_calculation

image statistics *

Get Link | Save | Run | Reset | Apps

```
// Instantiate an image with the Image constructor.
var image = ee.Image('CGIAR/SRTM90_V4');

// Zoom to a location.
Map.setCenter(37.02540746271644,56.766160331658845, 12);
var p1 = ee.Geometry.Point(37.02540746271644,56.766160331658845);
var p2 = ee.Geometry.Point(37.03502049982581,56.770864050772836);

var mean = image.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: p1,
  scale: 30
});

```

Inspector | Console | Tasks

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

```
▾ Object (1 property)                JSON
    elevation: 134

▾ Object (1 property)                JSON
    elevation: 120
```

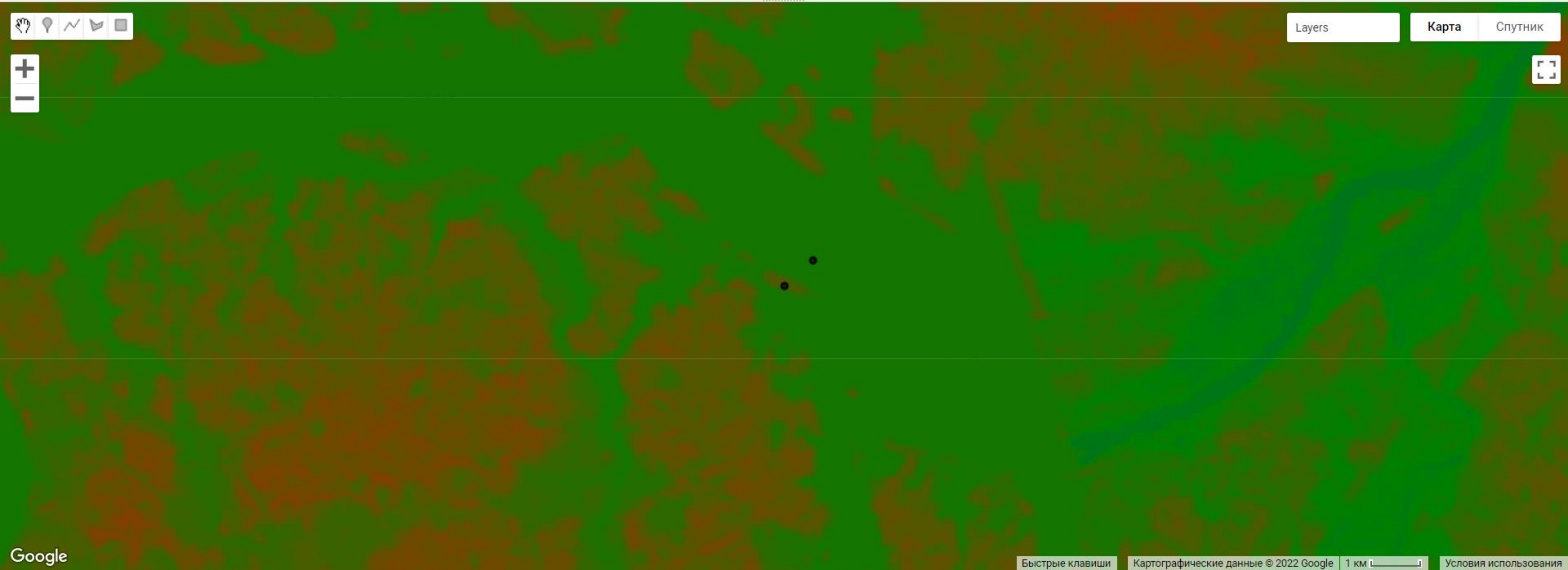
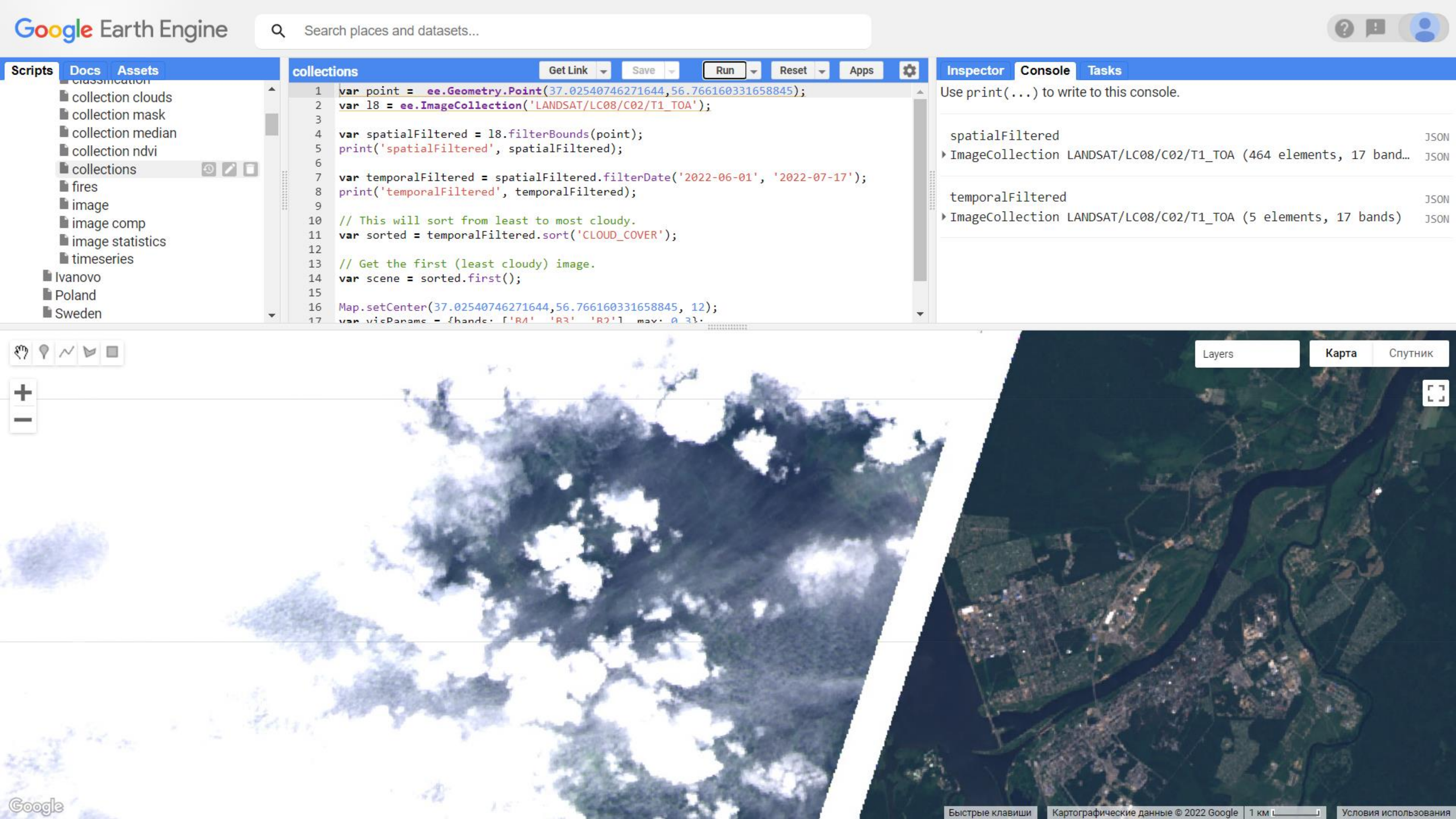


Image Collections

https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LC08_C02_T1_TOA

https://code.earthengine.google.com/92fc966d883b11f9850b36504e6ea5eb

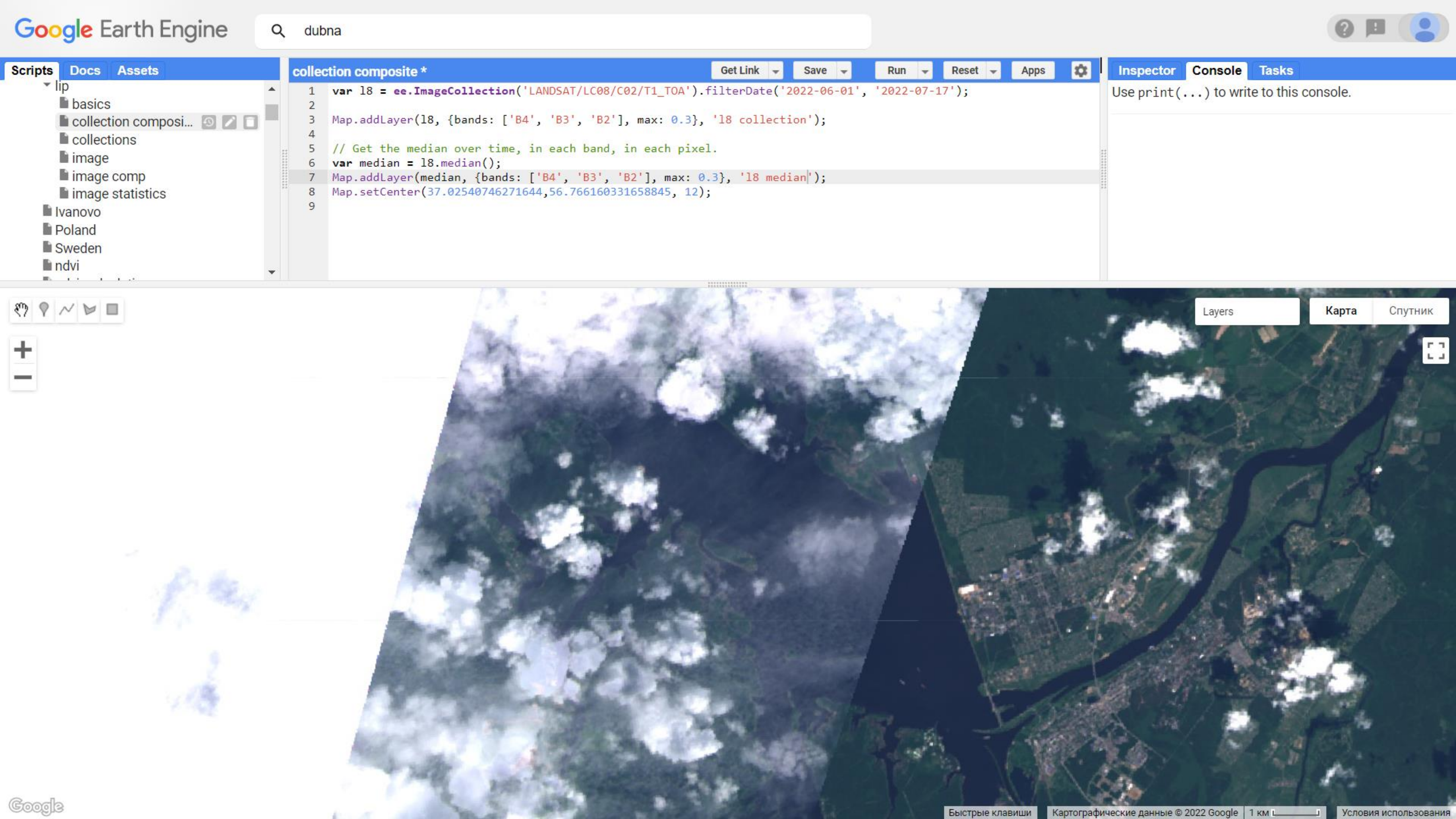
```
var point =  ee.Geometry.Point(37.02540746271644,56.766160331658845);
var l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA');

var spatialFiltered = l8.filterBounds(point);
print('spatialFiltered', spatialFiltered);

var temporalFiltered = spatialFiltered.filterDate('2022-06-01', '2022-07-17');
print('temporalFiltered', temporalFiltered);

// This will sort from least to most cloudy.
var sorted = temporalFiltered.sort('CLOUD_COVER');

// Get the first (least cloudy) image.
var scene = sorted.first();

Map.setCenter(37.02540746271644,56.766160331658845, 12);
var visParams = {bands: ['B4', 'B3', 'B2'], max: 0.3};
Map.addLayer(scene, visParams, 'true-color composite less clouds');
Map.addLayer(temporalFiltered, visParams, 'true-color composite');
```


Google Earth Engine
Search places and datasets...
Scripts
Docs
Assets
collection clouds
collection mask
collection median
collection ndvi
collections
fires
image
image comp
image statistics
timeseries
Ivanovo
Poland
Sweden
collections
Get Link
Save
Run
Reset
Apps
var point = ee.Geometry.Point(37.02540746271644,56.766160331658845);
var l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA');
var spatialFiltered = l8.filterBounds(point);
print('spatialFiltered', spatialFiltered);
var temporalFiltered = spatialFiltered.filterDate('2022-06-01', '2022-07-17');
print('temporalFiltered', temporalFiltered);
// This will sort from least to most cloudy.
var sorted = temporalFiltered.sort('CLOUD_COVER');
// Get the first (least cloudy) image.
var scene = sorted.first();
Map.setCenter(37.02540746271644,56.766160331658845, 12);
Inspector
Console
Tasks
Use print(...) to write to this console.
spatialFiltered
JSON
ImageCollection LANDSAT/LC08/C02/T1_TOA (464 elements, 17 band…
JSON
temporalFiltered
JSON
ImageCollection LANDSAT/LC08/C02/T1_TOA (5 elements, 17 bands)
JSON
Layers
Карта
Спутник
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

Median image

https://code.earthengine.google.com/e06d18ab4bda8ae5c6500407d8bd97d5

```
var l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA').filterDate('2022-06-01', '2022-07-17');

Map.addLayer(l8, {bands: ['B4', 'B3', 'B2'], max: 0.3}, 'l8 collection');

// Get the median over time, in each band, in each pixel.
var median = l8.median();
Map.addLayer(median, {bands: ['B4', 'B3', 'B2'], max: 0.3}, 'l8 median');
Map.setCenter(37.02540746271644,56.766160331658845, 12);
```


Google Earth Engine
dubna
Scripts
Docs
Assets
lip
basics
collection composi...
collections
image
image comp
image statistics
Ivanovo
Poland
Sweden
ndvi
collection composite *
Get Link
Save
Run
Reset
Apps
var l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA').filterDate('2022-06-01', '2022-07-17');
Map.addLayer(l8, {bands: ['B4', 'B3', 'B2'], max: 0.3}, 'l8 collection');
// Get the median over time, in each band, in each pixel.
var median = l8.median();
Map.addLayer(median, {bands: ['B4', 'B3', 'B2'], max: 0.3}, 'l8 median');
Map.setCenter(37.02540746271644,56.766160331658845, 12);
Inspector
Console
Tasks
Use print(...) to write to this console.
Layers
Карта
Спутник
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

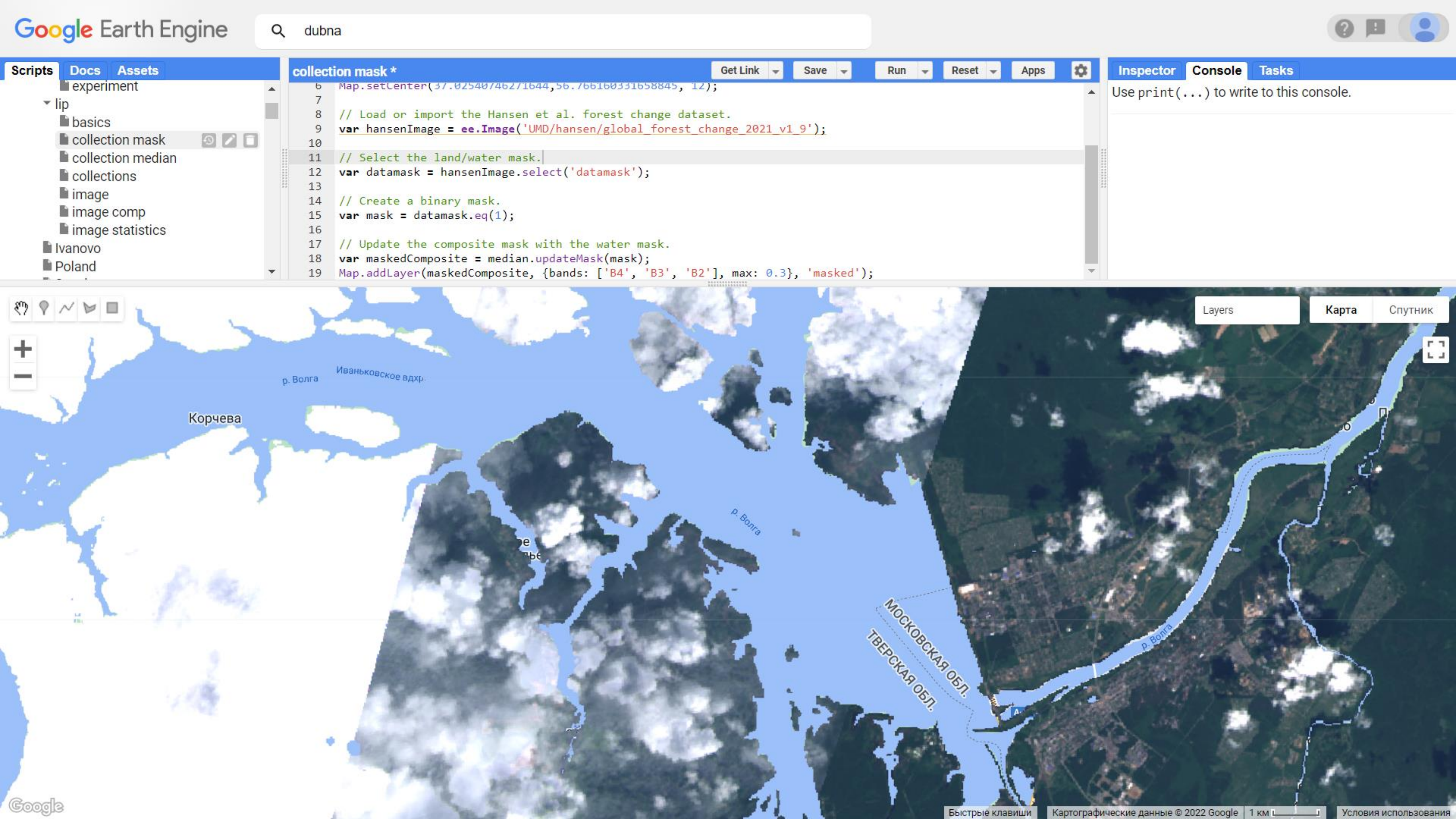
Masking

https://developers.google.com/earth-engine/apidocs/ee-image-updatemask

https://developers.google.com/earth-engine/datasets/catalog/UMD_hansen_global_forest_change_2021_v1_9?hl=en

https://code.earthengine.google.com/976753b4bfb128c1cc8293ac60fe1b3f

```
var l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA').filterDate('2022-06-01', '2022-07-17');

// Get the median over time, in each band, in each pixel.
var median = l8.median();
Map.addLayer(median, {bands: ['B4', 'B3', 'B2'], max: 0.3}, 'l8 median');
Map.setCenter(37.02540746271644,56.766160331658845, 12);

// Load or import the Hansen et al. forest change dataset.
var hansenImage = ee.Image('UMD/hansen/global_forest_change_2021_v1_9');

// Select the land/water mask.
var datamask = hansenImage.select('datamask');

// Create a binary mask.
var mask = datamask.eq(1);

// Update the composite mask with the water mask.
var maskedComposite = median.updateMask(mask);
Map.addLayer(maskedComposite, {bands: ['B4', 'B3', 'B2'], max: 0.3}, 'masked');
```


Google Earth Engine
dubna
Scripts
Docs
Assets
experiment
lip
basics
collection mask
collection median
collections
image
image comp
image statistics
Ivanovo
Poland
collection mask *
Get Link
Save
Run
Reset
Apps
Map.setCenter(37.02540746271644,56.766160331658845, 12);
// Load or import the Hansen et al. forest change dataset.
var hansenImage = ee.Image('UMD/hansen/global_forest_change_2021_v1_9');
// Select the land/water mask.
var datamask = hansenImage.select('datamask');
// Create a binary mask.
var mask = datamask.eq(1);
// Update the composite mask with the water mask.
var maskedComposite = median.updateMask(mask);
Map.addLayer(maskedComposite, {bands: ['B4', 'B3', 'B2'], max: 0.3}, 'masked');
Inspector
Console
Tasks
Use print(...) to write to this console.
Layers
Карта
Спутник
р. Волга
Иваньковское вдхр.
Корчева
р. Волга
МОСКОВСКАЯ ОБЛ.
ТВЕРСКАЯ ОБЛ.
р. Волга
Google
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

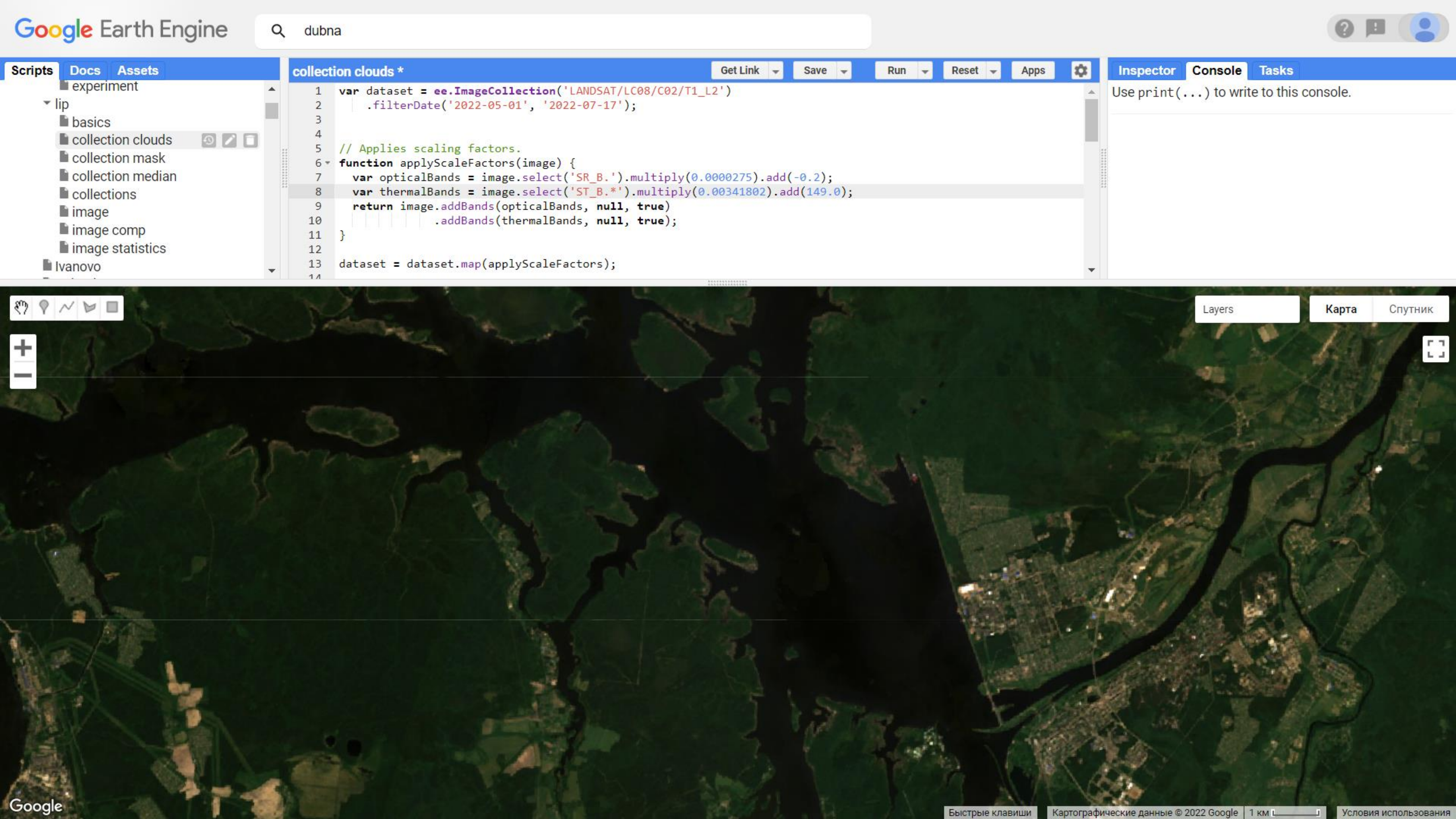
Clouds

https://code.earthengine.google.com/36ce292efce7e367f53aac53ed970fbf

https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LC08_C02_T1_L2#description

```
var dataset = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
    .filterDate('2022-05-01', '2022-07-17');


// Applies scaling factors.
function applyScaleFactors(image) {
  var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);
  var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
  return image.addBands(opticalBands, null, true)
            .addBands(thermalBands, null, true);
}

dataset = dataset.map(applyScaleFactors);

var visualization = {
  bands: ['SR_B4', 'SR_B3', 'SR_B2'],
  min: 0.0,
  max: 0.3,
};

Map.setCenter(37.02540746271644,56.766160331658845, 12);

Map.addLayer(dataset, visualization, 'clouds');
```

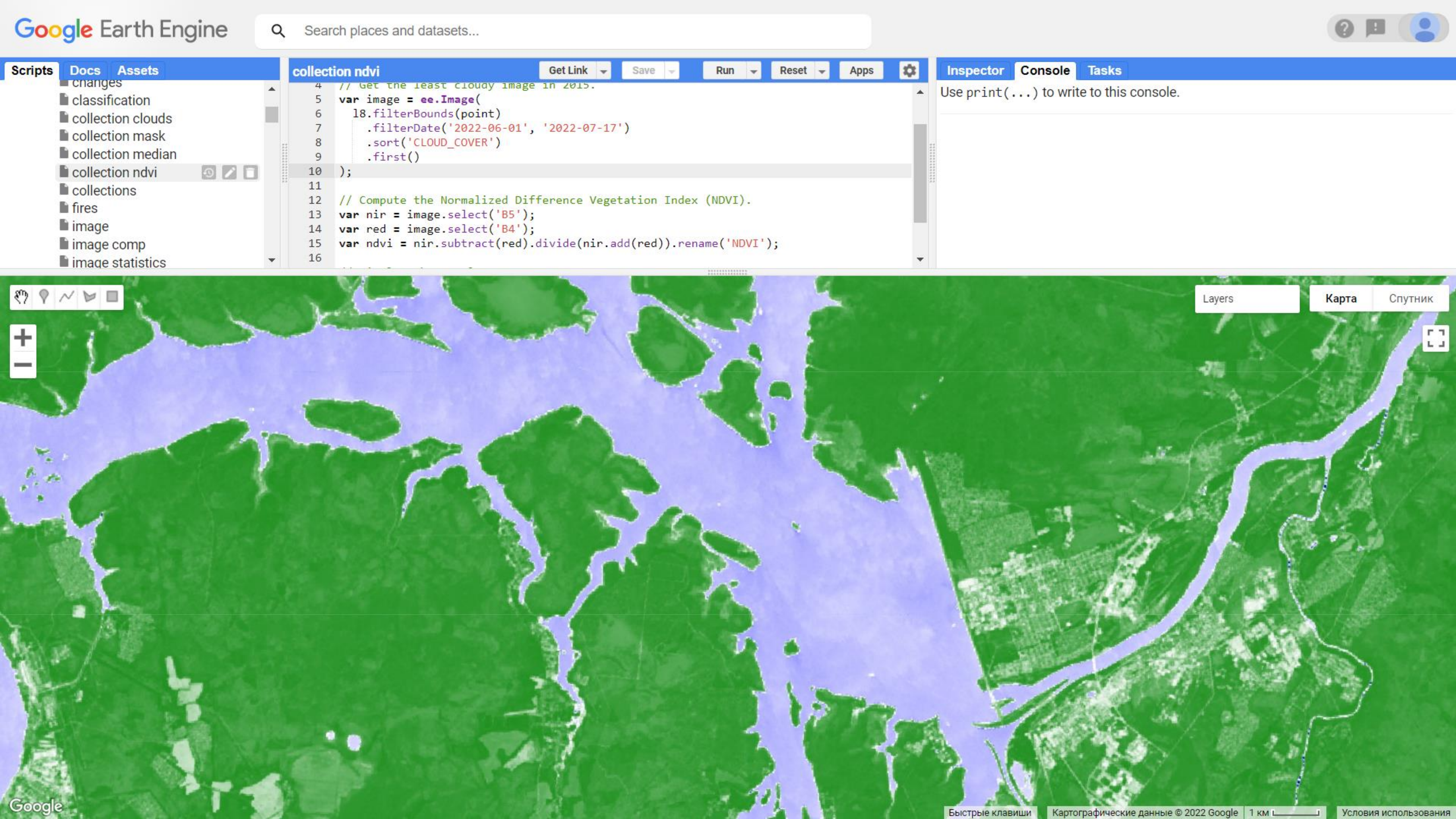
```
function maskL8sr(image) {
  // Bit 0 - Fill
  // Bit 1 - Dilated Cloud
  // Bit 2 - Cirrus
  // Bit 3 - Cloud
  // Bit 4 - Cloud Shadow
  var qaMask = image.select('QA_PIXEL').bitwiseAnd(parseInt('11111', 2)).eq(0);
  var saturationMask = image.select('QA_RADSAT').eq(0);

  // Apply the scaling factors to the appropriate bands.
  var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);

  // Replace the original bands with the scaled ones and apply the masks.
  return image.addBands(opticalBands, null, true)
      .updateMask(qaMask)
      .updateMask(saturationMask);
}

// Map the function over one year of data.
var collection = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
                 .filterDate('2022-05-01', '2022-07-17')
                 .map(maskL8sr);

var composite = collection.median();

// Display the results.
Map.addLayer(composite, {bands: ['SR_B4', 'SR_B3', 'SR_B2'], min: 0, max: 0.3}, 'no clouds');
```


Google Earth Engine
dubna
Scripts
Docs
Assets
experiment
lip
basics
collection clouds
collection mask
collection median
collections
image
image comp
image statistics
Ivanovo
collection clouds *
Get Link
Save
Run
Reset
Apps
var dataset = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
.filterDate('2022-05-01', '2022-07-17');
// Applies scaling factors.
function applyScaleFactors(image) {
var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);
var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
return image.addBands(opticalBands, null, true)
.addBands(thermalBands, null, true);
}
dataset = dataset.map(applyScaleFactors);
Inspector
Console
Tasks
Use print(...) to write to this console.
Layers
Карта
Спутник
Google
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

NDVI

https://code.earthengine.google.com/ec9e2f81b63ad006475b263f5f0cd661

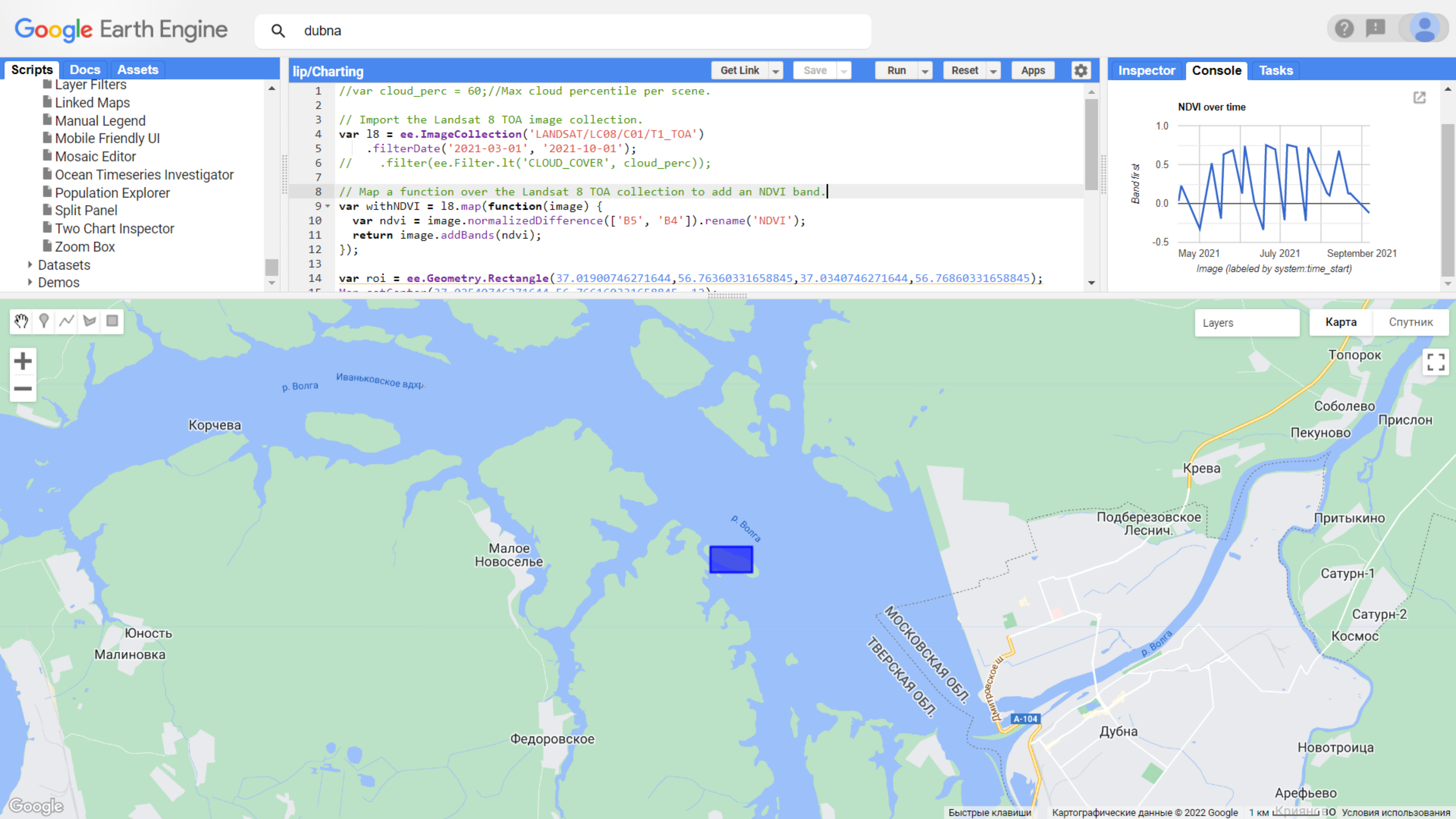
```
var point =  ee.Geometry.Point(37.02540746271644,56.766160331658845);
var l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA');

// Get the least cloudy image in 2015.
var image = ee.Image(
  l8.filterBounds(point)
    .filterDate('2022-06-01', '2022-07-17')
    .sort('CLOUD_COVER')
    .first()
);

// Compute the Normalized Difference Vegetation Index (NDVI).
var nir = image.select('B5');
var red = image.select('B4');
var ndvi = nir.subtract(red).divide(nir.add(red)).rename('NDVI');

// Display the result.
Map.setCenter(37.02540746271644,56.766160331658845, 12);
var ndviParams = {min: -1, max: 1, palette: ['blue', 'white', 'green']};
Map.addLayer(ndvi, ndviParams, 'NDVI image');
```


Google Earth Engine
Search places and datasets...
Scripts
Docs
Assets
changes
classification
collection clouds
collection mask
collection median
collection ndvi
collections
fires
image
image comp
image statistics
collection ndvi
Get Link
Save
Run
Reset
Apps
// Get the least cloudy image in 2015.
var image = ee.Image(
l8.filterBounds(point)
.filterDate('2022-06-01', '2022-07-17')
.sort('CLOUD_COVER')
.first()
);
// Compute the Normalized Difference Vegetation Index (NDVI).
var nir = image.select('B5');
var red = image.select('B4');
var ndvi = nir.subtract(red).divide(nir.add(red)).rename('NDVI');
Inspector
Console
Tasks
Use print(...) to write to this console.
Layers
Карта
Спутник
Google
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

Charts

https://code.earthengine.google.com/ec976be7da2129036839e1cf2f21cbbb

```
//var cloud_perc = 60;//Max cloud percentile per scene.

// Import the Landsat 8 TOA image collection.
var l8 = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
    .filterDate('2021-03-01', '2021-12-01');
//    .filter(ee.Filter.lt('CLOUD_COVER', cloud_perc));

// Map a function over the Landsat 8 TOA collection to add an NDVI band.
var withNDVI = l8.map(function(image) {
  var ndvi = image.normalizedDifference(['B5', 'B4']).rename('NDVI');
  return image.addBands(ndvi);
});

var roi = ee.Geometry.Rectangle(37.01900746271644,56.76360331658845,37.0340746271644,56.76860331658845);
Map.setCenter(37.02540746271644,56.766160331658845, 12);
Map.addLayer(roi, {color: 'blue'}, 'clouds');


// Create a chart.
var chart = ui.Chart.image.series({
  imageCollection: withNDVI.select('NDVI'),
  region: roi,
  reducer: ee.Reducer.median(),
  scale: 30
}).setOptions({title: 'NDVI over time'});

// Display the chart in the console.
print(chart);
```



Google Earth Engine
dubna
Scripts
Docs
Assets
Layer Filters
Linked Maps
Manual Legend
Mobile Friendly UI
Mosaic Editor
Ocean Timeseries Investigator
Population Explorer
Split Panel
Two Chart Inspector
Zoom Box
Datasets
Demos
lip/Charting
Get Link
Save
Run
Reset
Apps
//var cloud_perc = 60;//Max cloud percentile per scene.
// Import the Landsat 8 TOA image collection.
var l8 = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
.filterDate('2021-03-01', '2021-10-01');
// .filter(ee.Filter.lt('CLOUD_COVER', cloud_perc));
// Map a function over the Landsat 8 TOA collection to add an NDVI band.
var withNDVI = l8.map(function(image) {
var ndvi = image.normalizedDifference(['B5', 'B4']).rename('NDVI');
return image.addBands(ndvi);
});
var roi = ee.Geometry.Rectangle(37.01900746271644,56.76360331658845,37.0340746271644,56.76860331658845);
Inspector
Console
Tasks
NDVI over time
1.0
0.5
0.0
-0.5
Band first
May 2021
July 2021
September 2021
Image (labeled by system:time_start)
Layers
Карта
Спутник
р. Волга
Иваньковское вдхр.
Корчева
Малое Новоселье
Юность
Малиновка
Федоровское
р. Волга
Топорок
Соболево
Прислон
Пекуново
Крева
Подберезовское Леснич.
Притыкино
Сатурн-1
Сатурн-2
Космос
МОСКОВСКАЯ ОБЛ.
ТВЕРСКАЯ ОБЛ.
Дмитровское ш.
A-104
р. Волга
Дубна
Новотроица
Арефьево
Google
Быстрые клавиши
Картографические данные © 2022 Google
1 км
Условия использования

Burn Areas

https://developers.google.com/earth-engine/datasets/catalog/FIRMS?hl=en

https://code.earthengine.google.com/e9c2c3dd7cfb47bb55111557da6102d0

```
var dataset = ee.ImageCollection('FIRMS').filter(
    ee.Filter.date('2022-07-01', '2022-07-16'));
var fires = dataset.select('T21');
var firesVis = {
  min: 325.0,
  max: 400.0,
  palette: [
    'ff0000', 'fd4100', 'fb8200', 'f9c400', 'f2ff00', 'b6ff05',
    '7aff0a', '3eff0f', '02ff15', '00ff55', '00ff99', '00ffdd',
    '00ddff', '0098ff', '0052ff', '0210ff', '3a0dfb', '7209f6',
    'a905f1', 'e102ed', 'ff00cc', 'ff0089', 'ff0047', 'ff0004'
  ]
};
Map.setCenter(37.02540746271644,56.766160331658845, 7);
Map.addLayer(fires, firesVis, 'Fires');
```

Google Earth Engine

Search places and datasets...

Scripts | Docs | Assets

- collection mask
- collection median
- collection ndvi
- collections
- fires
- image
- image comp
- image statistics
- Ivanovo
- Poland
- Sweden
- TOA corrections

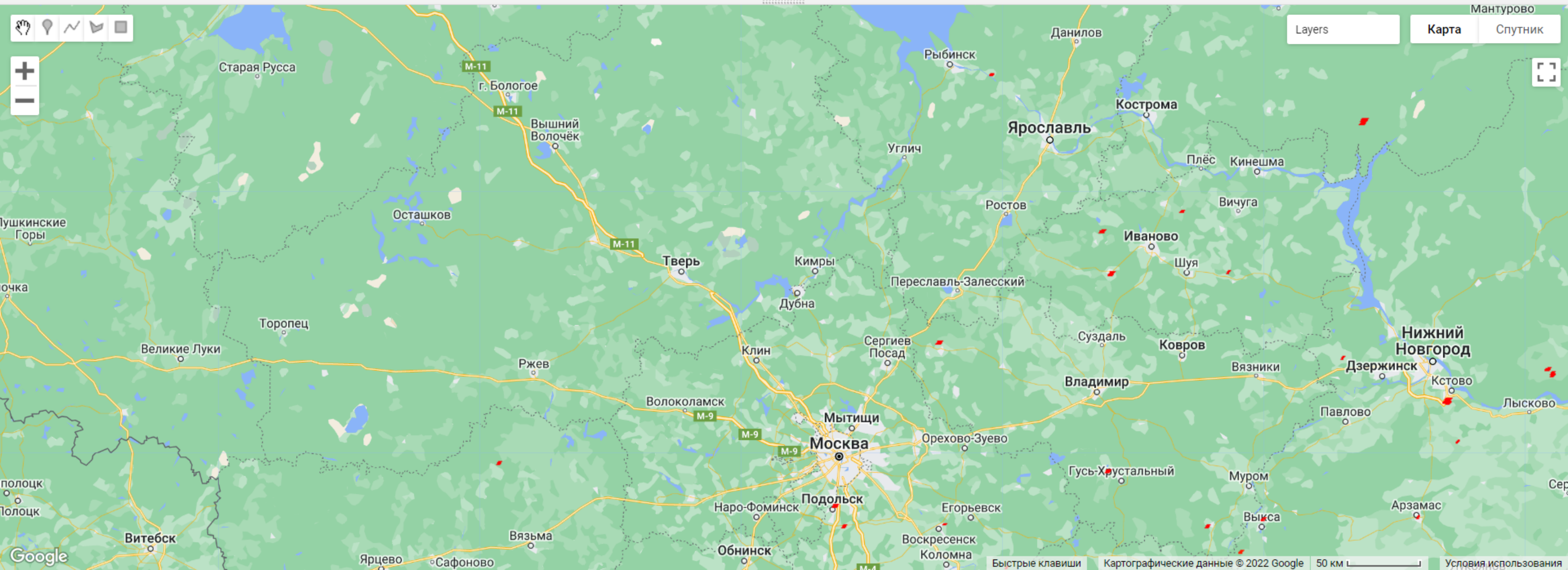
lip/fires *

Get Link | Save | Run | Reset | Apps

```
var fires = dataset.select('T21');
var firesVis = {
  min: 325.0,
  max: 400.0,
  palette: [
    'ff0000', 'fd4100', 'fb8200', 'f9c400', 'f2ff00', 'b6ff05',
    '7aff0a', '3eff0f', '02ff15', '00ff55', '00ff99', '00ffdd',
    '00ddff', '0098ff', '0052ff', '0210ff', '3a0dfb', '7209f6',
    'a905f1', 'e102ed', 'ff00cc', 'ff0089', 'ff0047', 'ff0004'
  ]
};
Map.setCenter(37.02540746271644,56.766160331658845, 7);
Map.addLayer(fires, firesVis, 'Fires');
```

Inspector | Console | Tasks

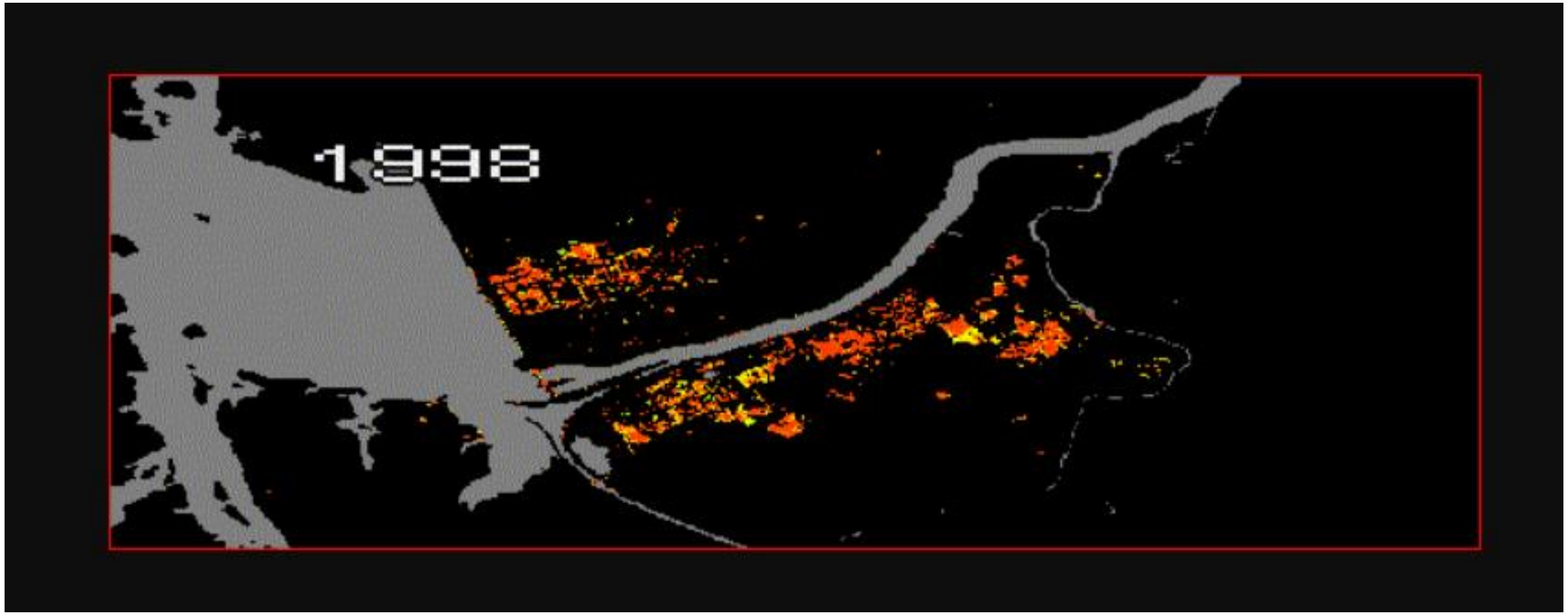
Use print(...) to write to this console.



Timeseries

https://developers.google.com/earth-engine/datasets/catalog/Tsinghua_FROM-GLC_GAIA_v10?hl=en#description

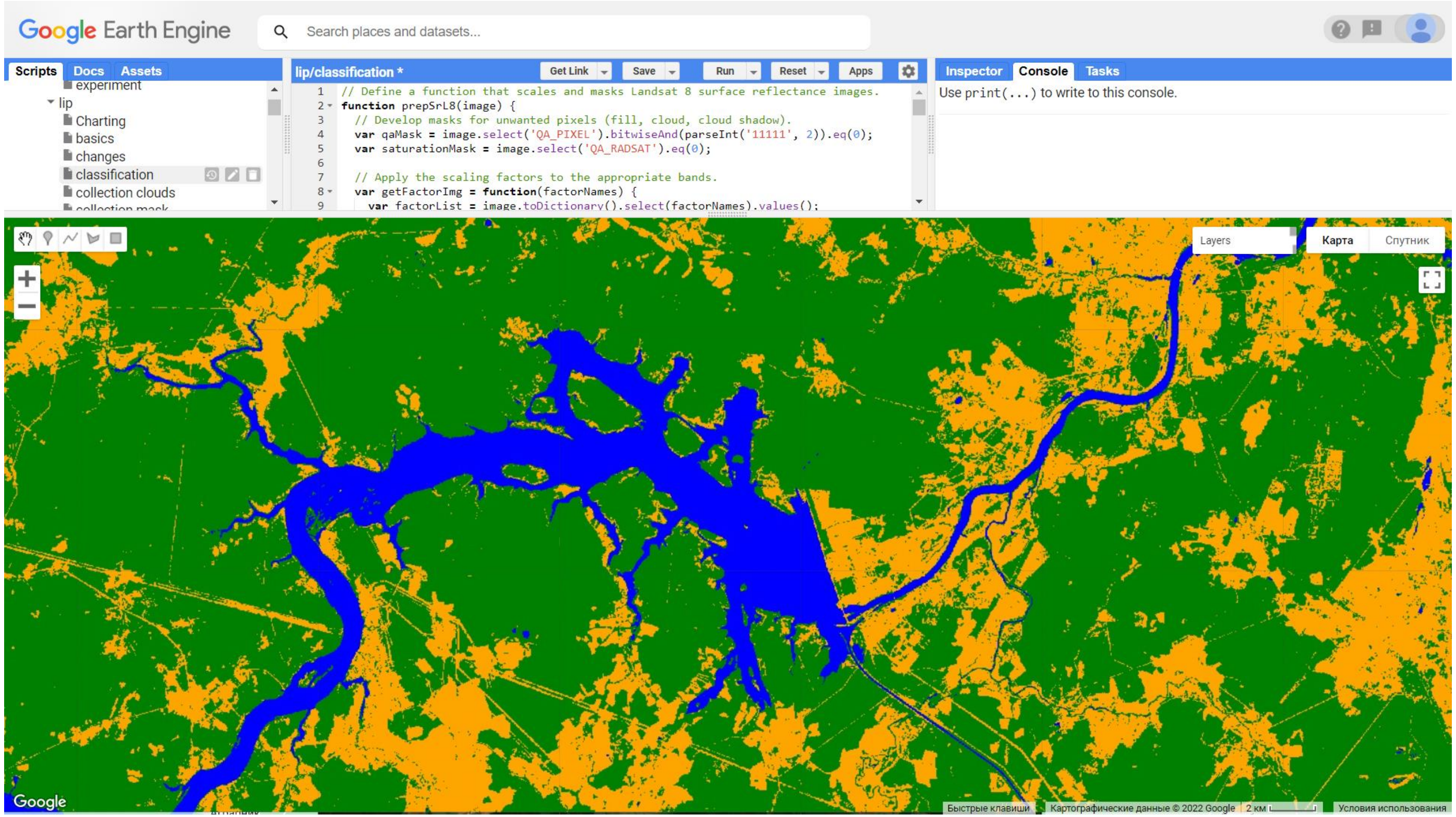
https://code.earthengine.google.com/cdd574d1d7466208b1eb5bff5fb7c02e



Land use classification

https://developers.google.com/earth-engine/apidocs/ee-classifier-libsvm

https://code.earthengine.google.com/1828d8cf8bd900884f4a9746ee214252




That's all Folks!