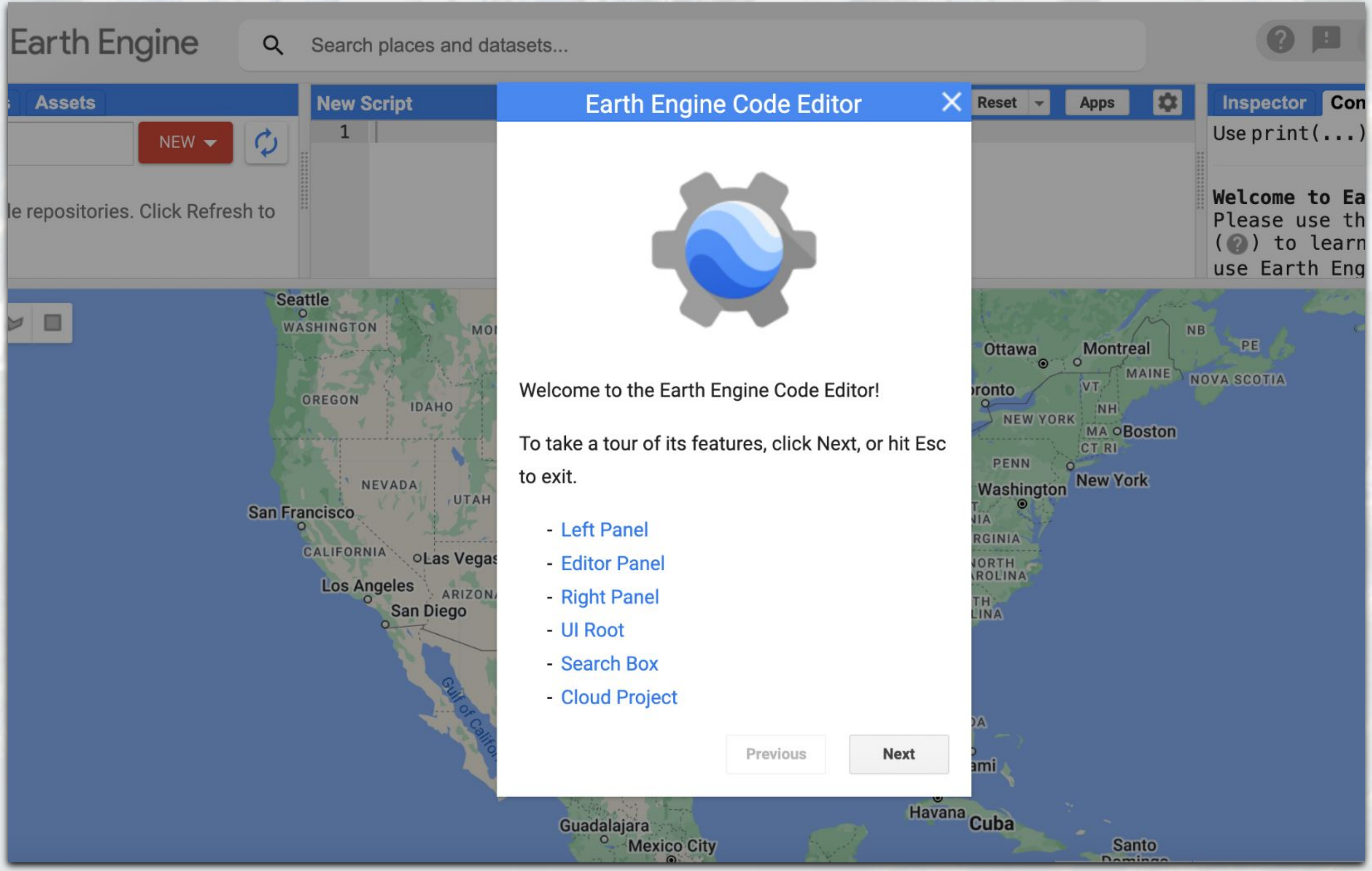
CS 106S Week 3

CS for Climate Change

cs106s.stanford.edu, Autumn 2024

Stanford | ENGINEERING
Computer Science

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

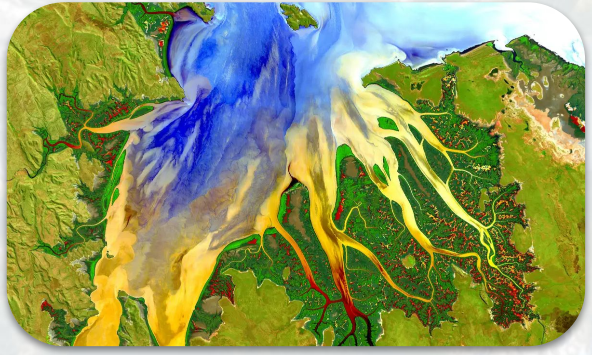


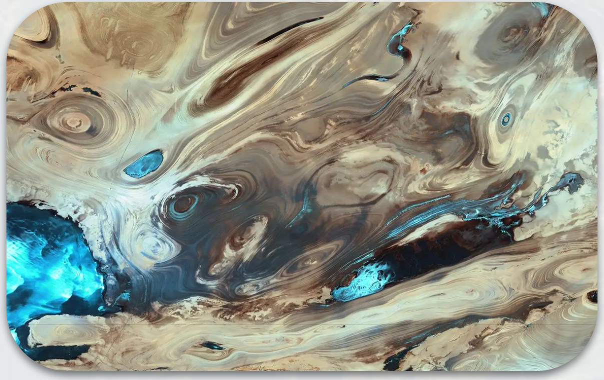
Earth Engine
Search places and datasets...
Assets
NEW
le repositories. Click Refresh to
New Script
Reset
Apps
Inspector
Con
Use print(...)
Welcome to Ea
Please use th
(?) to learn
use Earth Eng
Earth Engine Code Editor
Welcome to the Earth Engine Code Editor!
To take a tour of its features, click Next, or hit Esc to exit.
- Left Panel
- Editor Panel
- Right Panel
- UI Root
- Search Box
- Cloud Project
Previous
Next
Seattle
WASHINGTON
OREGON
IDAHO
NEVADA
UTAH
San Francisco
CALIFORNIA
Las Vegas
Los Angeles
San Diego
Guadalajara
Mexico City
Ottawa
Montreal
MAINE
NOVA SCOTIA
NB
PE
VT
NH
NEW YORK
MA
Boston
CT
RI
PENN
Washington
New York
Havana
Cuba
Santo

Agenda for Today

1. **Background:** What is remote sensing? What is the Earth Engine? How can satellite imagery be applied to help address climate change?
2. **Google Earth Engine Tutorial:** Get familiarized with its JavaScript API (e.g., loading & processing satellite images).
3. **Earth Art Gallery:** Javascript Exercise, fairly open-ended
4. **Reflections and Check-Off!**

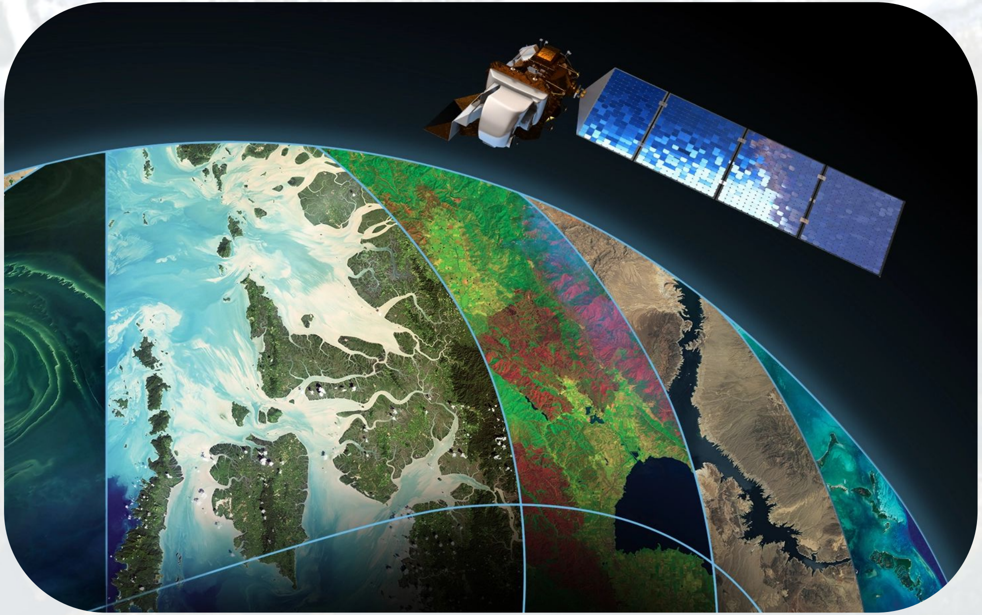
Remote Sensing





What is Remote Sensing?

Touching a rock to examine its texture?



The Landsat satellites (1972 – present) capturing images of the Earth’s surface.

Pigeons outfitted with cameras to take pictures over enemy territory (WW1).

Remote Sensing

“Remote sensing is acquiring information about an object from a distance.”
– NASA

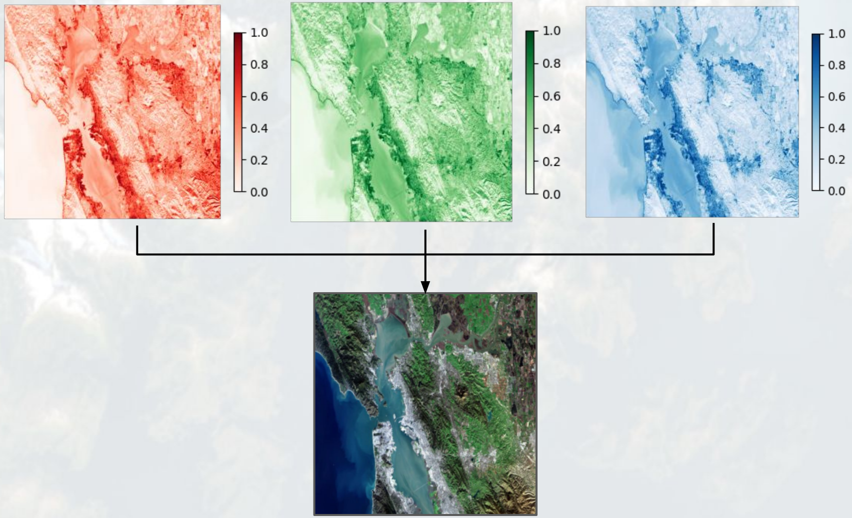
No physical contact with object!

It usually refers to observing Earth’s surface through sensors mounted on aircraft or satellites (maybe pigeons).

Satellite Imagery

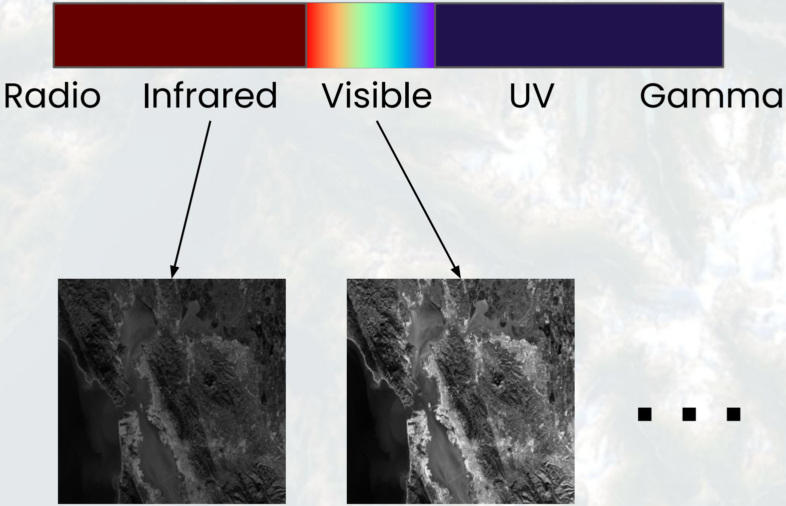
RGB

Red Green Blue



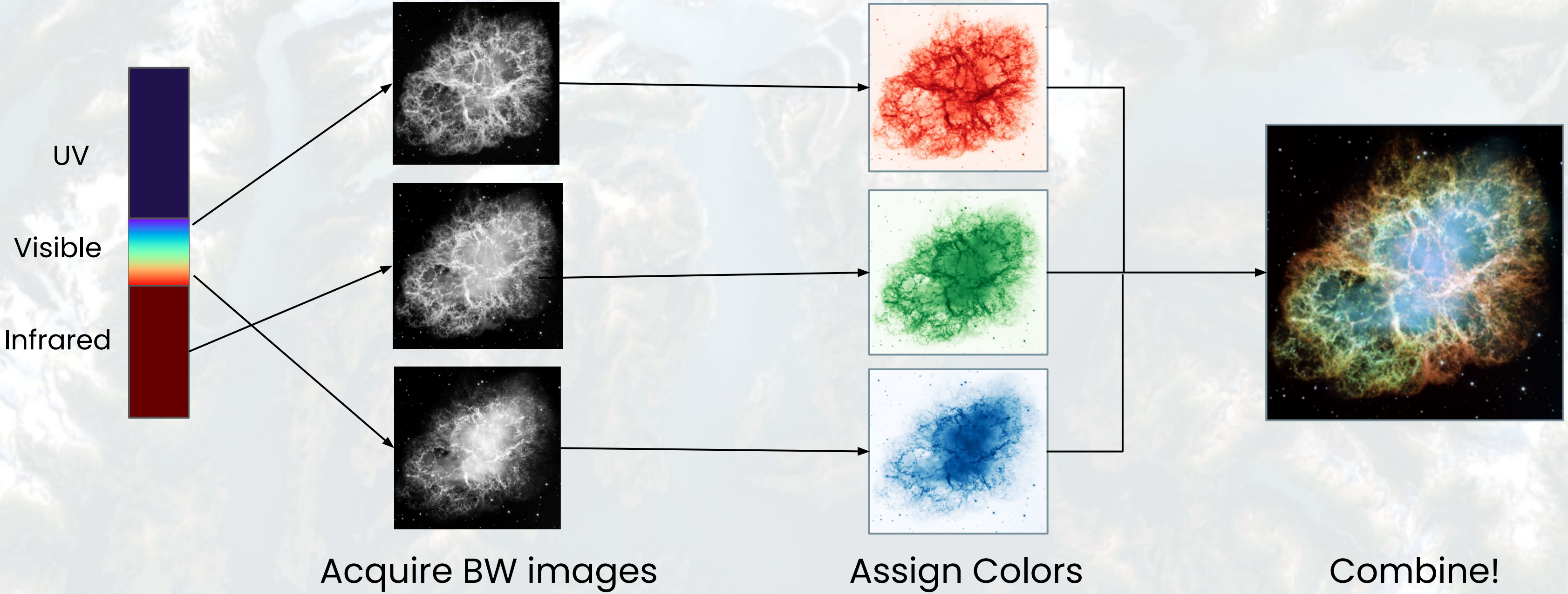
3 Bands

Multispectral



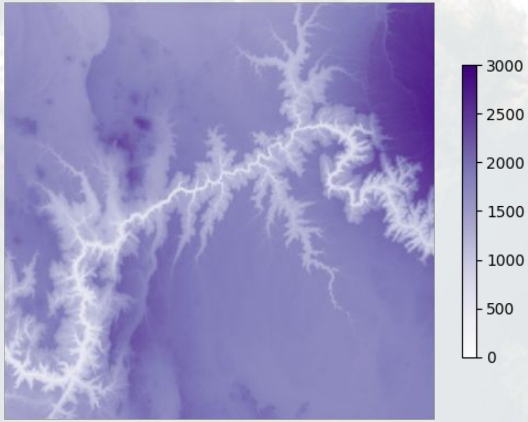
Varying # Of Bands

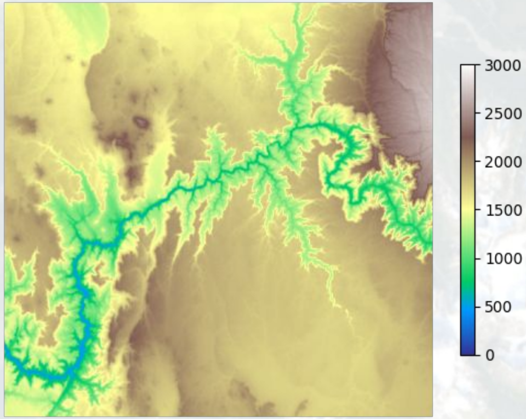
False-Color / Pseudocolor Imagery



Telescope photos are all truly in black and white! Scientists assign visible colors to different wavelengths of light.

False-Color / Pseudocolor Imagery

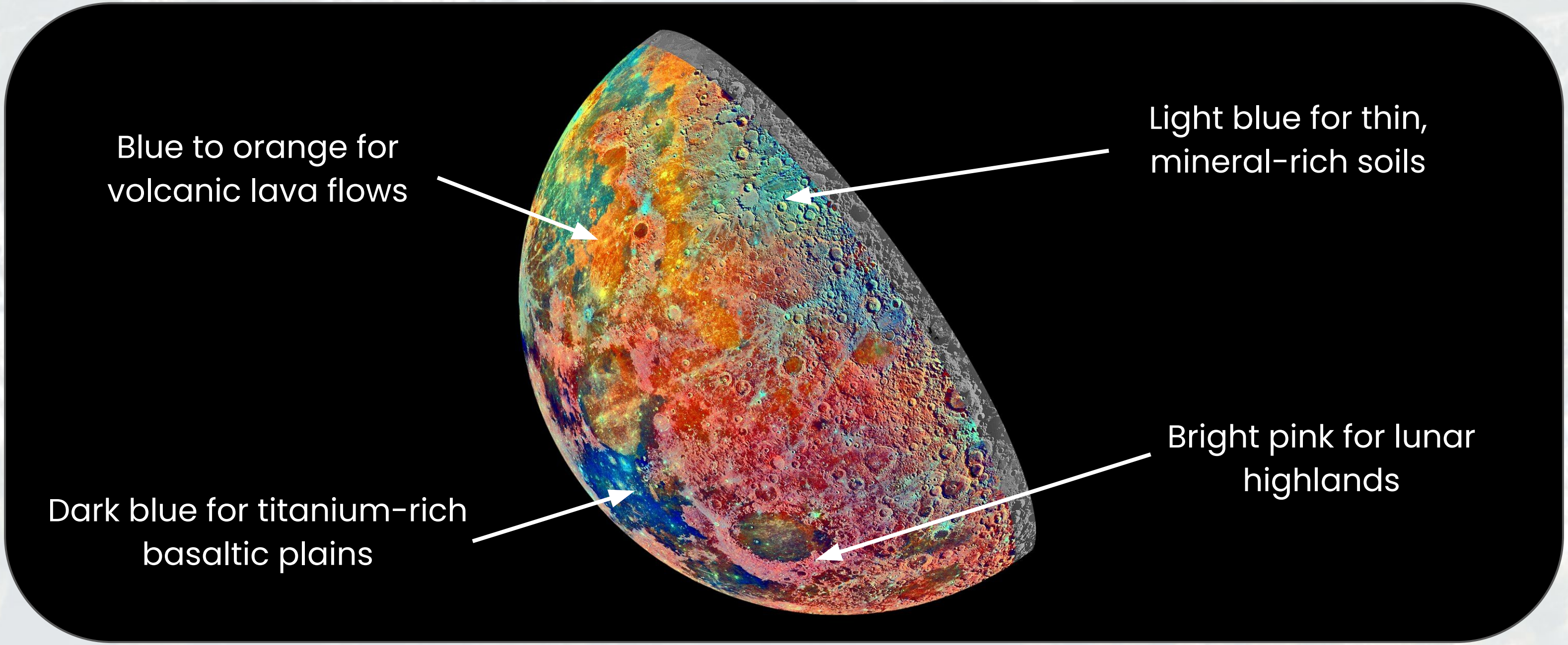




Coloring an Elevation Map

Why might it be useful to depict an image in colors that differ from what our human eyes would normally see?

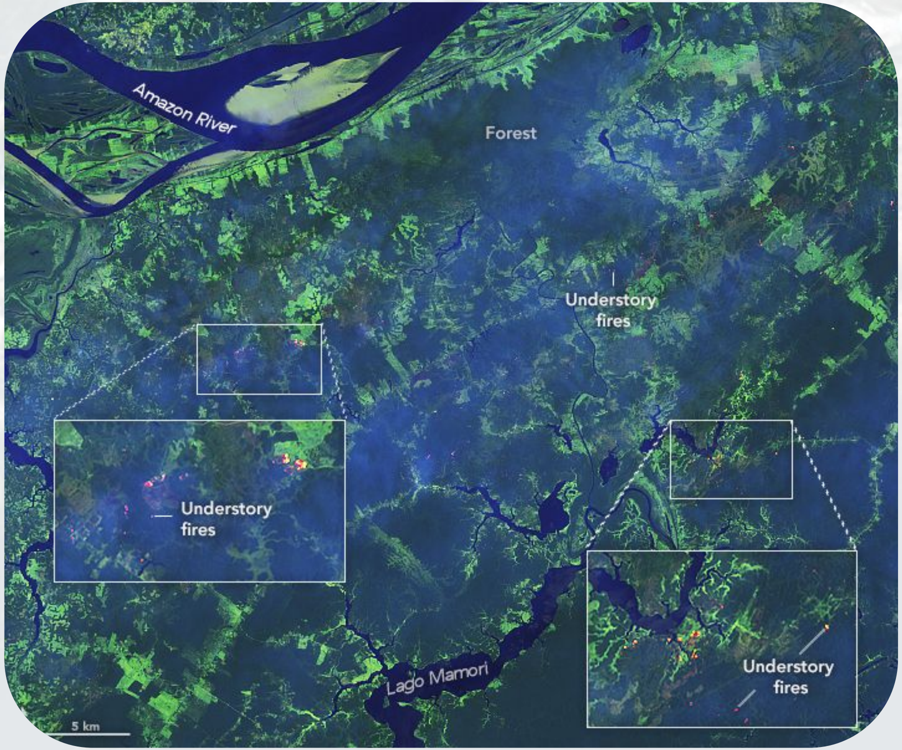
Moon Crescent Mosaic



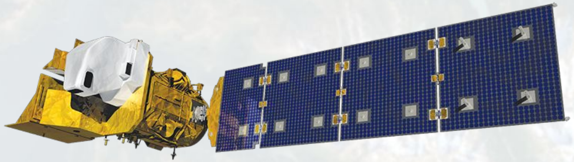
False-Color Composite of 53 Images!

How can satellite imagery be applied for humanitarian and sustainability causes?

Mapping Forest Fires in the Amazon



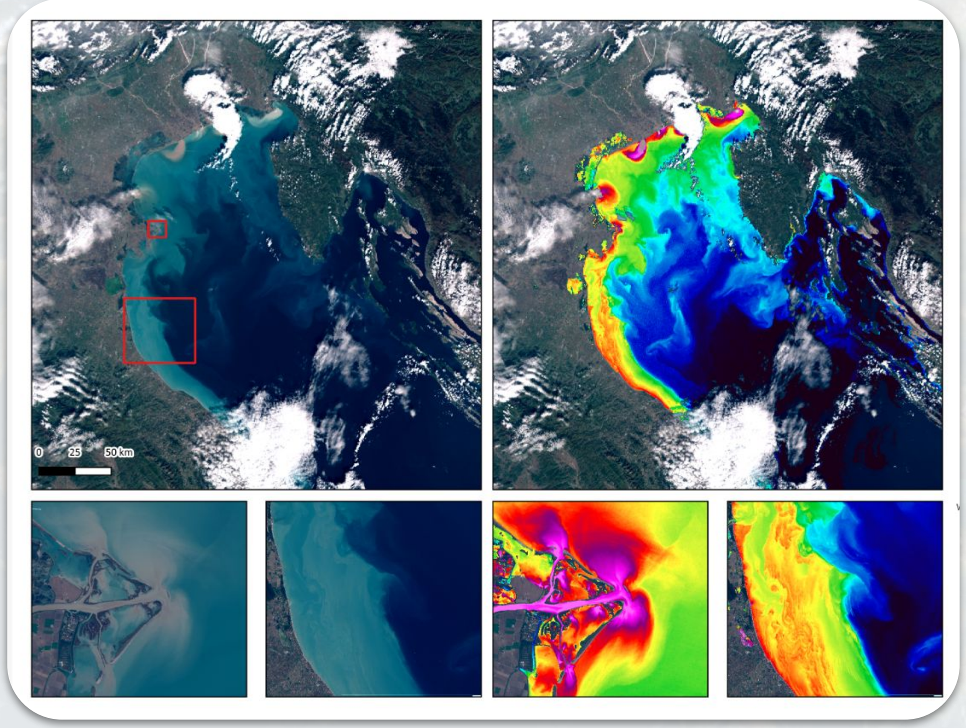
False-Color Image Using Shortwave Infrared Signals



Landsat 9 (Operating Land-Imager 2 Sensor)

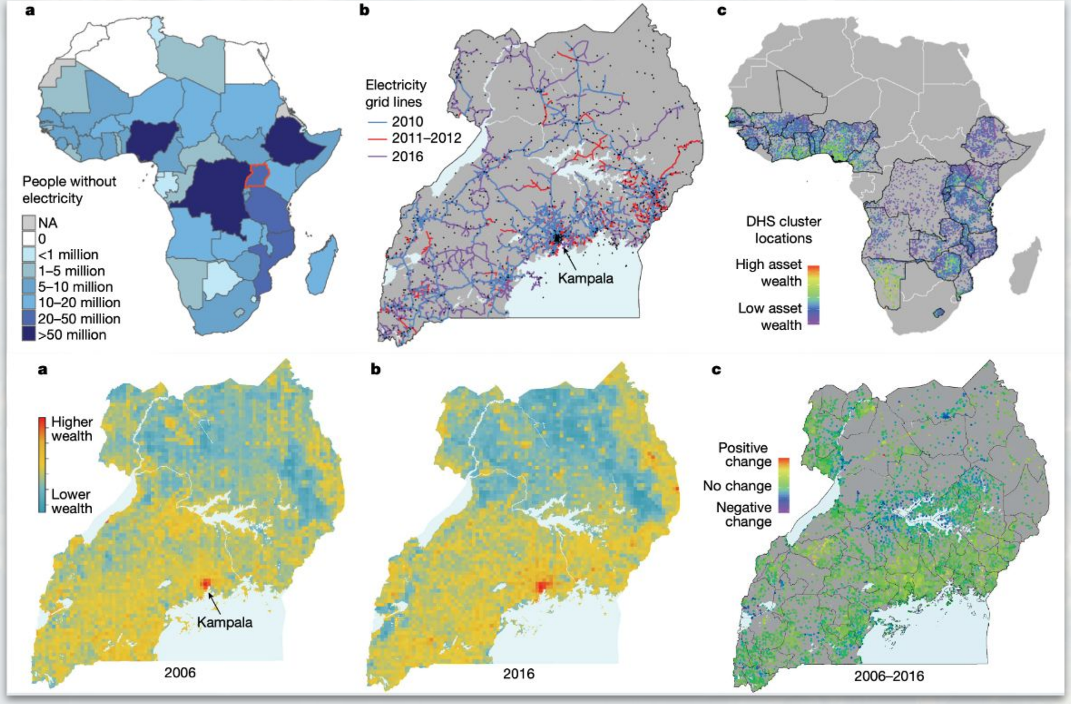
https://landsat.visibleearth.nasa.gov/view.php?id=151965

Monitoring Flooding Near Adriatic Sea, Italy



https://www.eomap.com/using-satellite-data-for-flood-monitoring/

Economic Well-Being and Electricity Access in Uganda



https://www.nature.com/articles/s41586-022-05322-8

How can we ensure remote satellite analyses are aligned with on-ground community needs?

TECHNOLOGY

Satellite Images Can Harm the Poorest Citizens

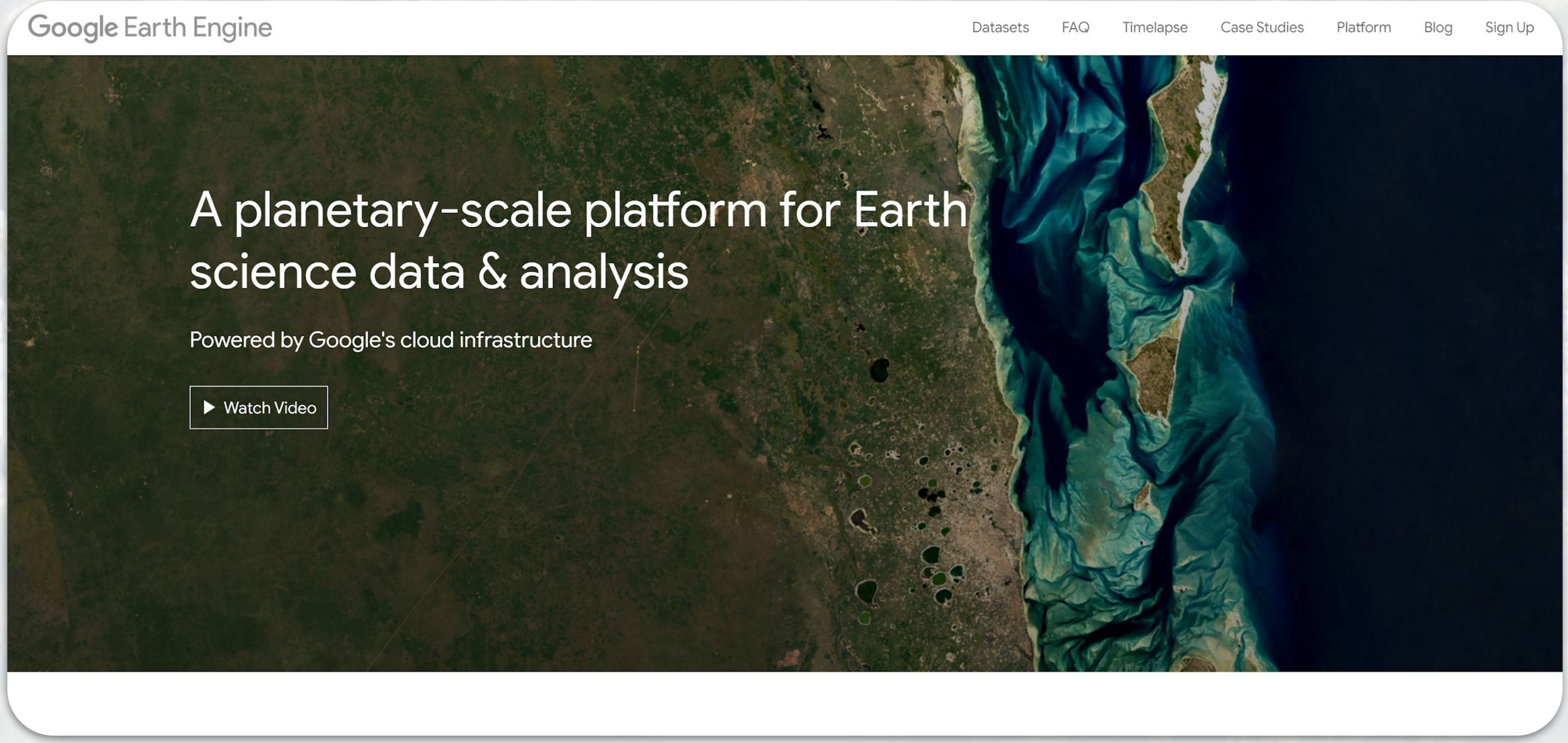
In Ho Chi Minh City, computer analysis of orbital images overlooks some urban communities. To represent them, cities will have to put boots on the ground.

By Annette M. Kim

https://www.theatlantic.com/technology/archive/2018/06/satellite-images-can-harm-the-poorest-citizens/561920/

What is the Google Earth Engine?

Google Earth Engine



Google Earth Engine

Datasets

Petabyte-scale catalog of public and free-to-use geospatial datasets.

Explore the Data Catalog

Compute

Leverage Google's cloud platform for planetary-scale analysis of Earth science data.

Read the publication

APIs

Full-featured JavaScript, Python and REST APIs.

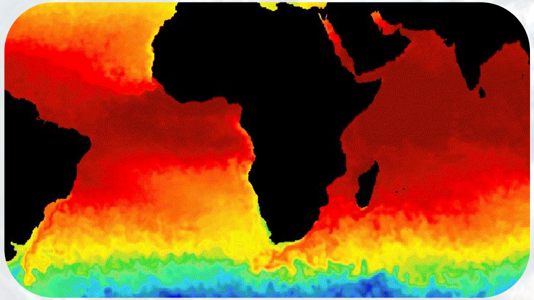
Developer guides

Apps

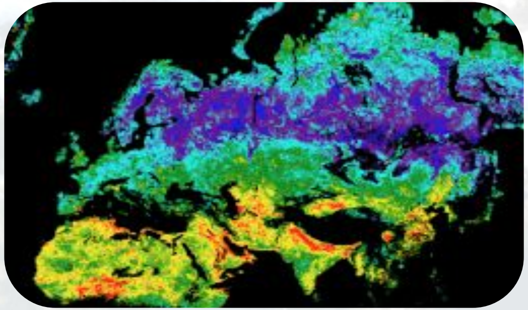
Dynamic, publicly accessible user interfaces for Earth Engine analyses.

Apps gallery

GEE Data Catalog



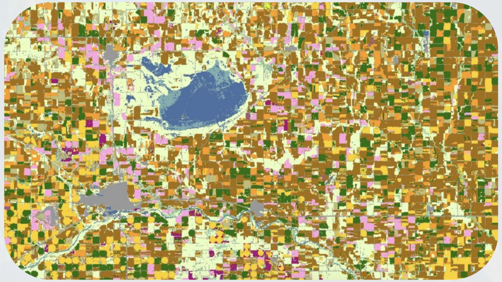
Global Surface Temperature



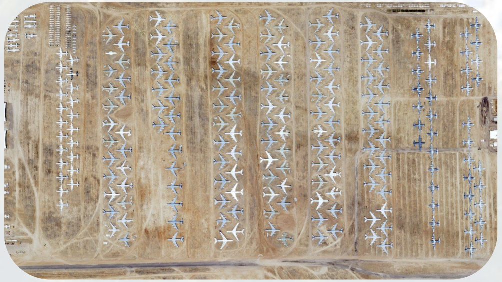
Climate (e.g., methane levels)

Sentinel-2 Multispectral

Terrain (e.g., elevation maps)



Cropland



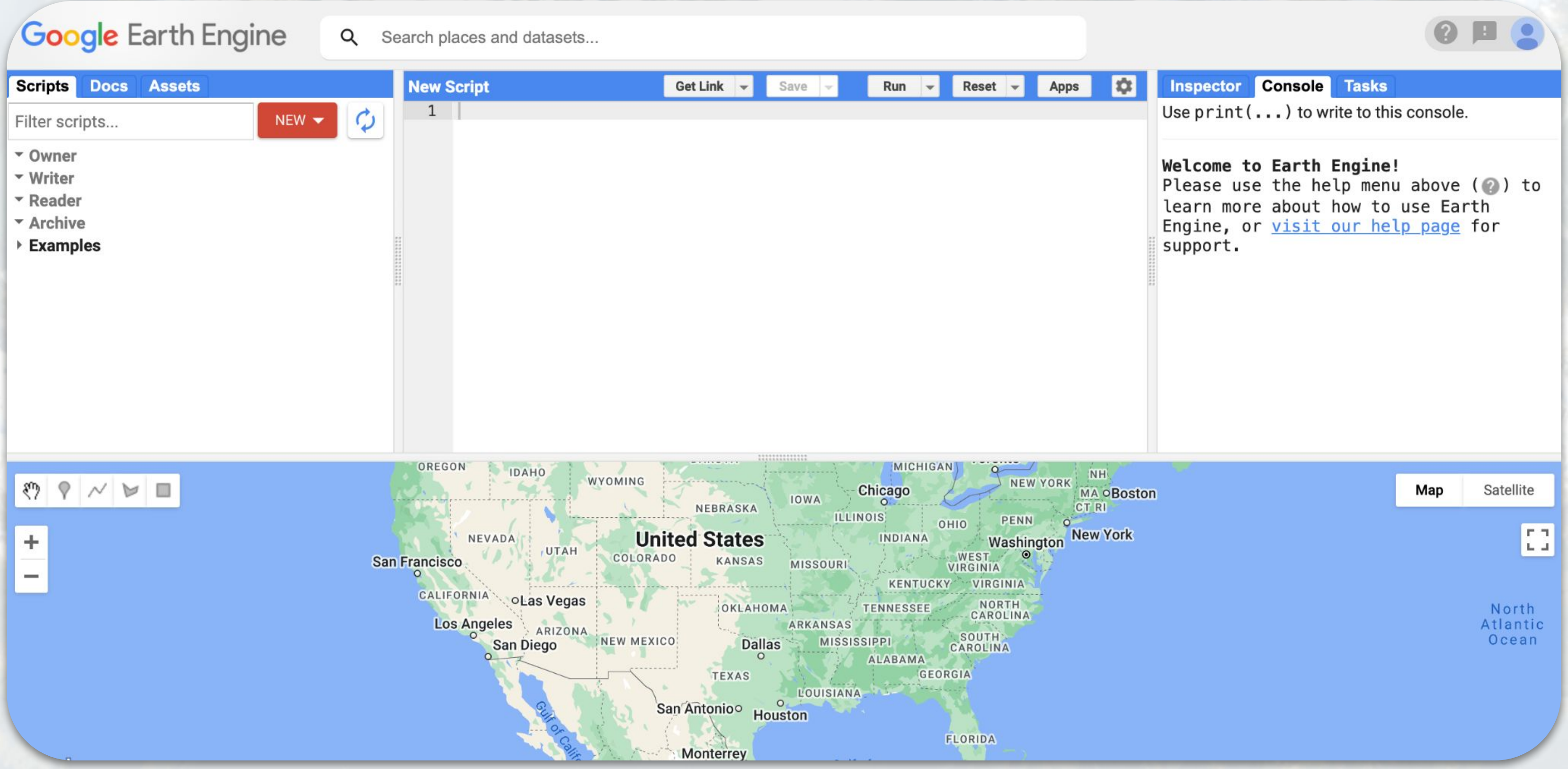
High-Resolution Imagery

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

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

Make sure to use the Google account—ideally, Stanford—you registered with!

https://tinyurl.com/cs106s-w6-code


Google Earth Engine
Search places and datasets...
Scripts
Docs
Assets
Filter scripts...
NEW
Owner
Writer
Reader
Archive
Examples
New Script
Get Link
Save
Run
Reset
Apps
1
Inspector
Console
Tasks
Use print(...) to write to this console.
Welcome to Earth Engine!
Please use the help menu above (?) to learn more about how to use Earth Engine, or visit our help page for support.
Map
Satellite
United States
North Atlantic Ocean
San Francisco
Las Vegas
Los Angeles
San Diego
Chicago
Boston
New York
Washington
Dallas
San Antonio
Houston
Monterrey
OREGON
IDAHO
WYOMING
NEVADA
UTAH
COLORADO
CALIFORNIA
ARIZONA
NEW MEXICO
NEBRASKA
KANSAS
OKLAHOMA
TEXAS
IOWA
ILLINOIS
MISSOURI
ARKANSAS
LOUISIANA
MISSISSIPPI
ALABAMA
GEORGIA
FLORIDA
TENNESSEE
KENTUCKY
INDIANA
OHIO
MICHIGAN
WEST VIRGINIA
VIRGINIA
NORTH CAROLINA
SOUTH CAROLINA
PENN
NEW YORK
NH
MA
CT
RI

New Script * | Get Link | Save | Run | Reset | Apps

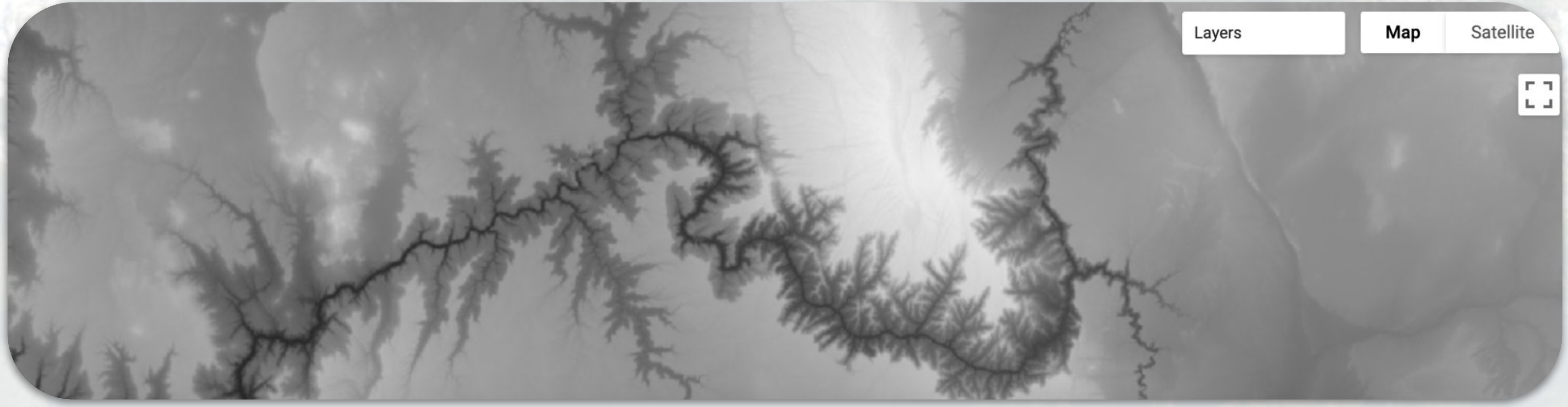
```javascript
// instantiate an image (here, a global elevation map)
var image = ee.Image('CGIAR/SRTM90_V4');

// add the image to the Map below
Map.addLayer(image);
```

https://developers.google.com/earth-engine/tutorials/tutorial_api_02

New Script * Get Link Save Run Reset Apps

```
// instantiate an image (here, a global elevation map)
var image = ee.Image('CGIAR/SRTM90_V4');

// center the map at the Grand Canyon and zoom in
Map.setCenter(-112.8598, 36.2841, 9);

// add the image to the Map below (set grayscale range to [0,3000])
Map.addLayer(image, {min: 0, max: 3000});
```

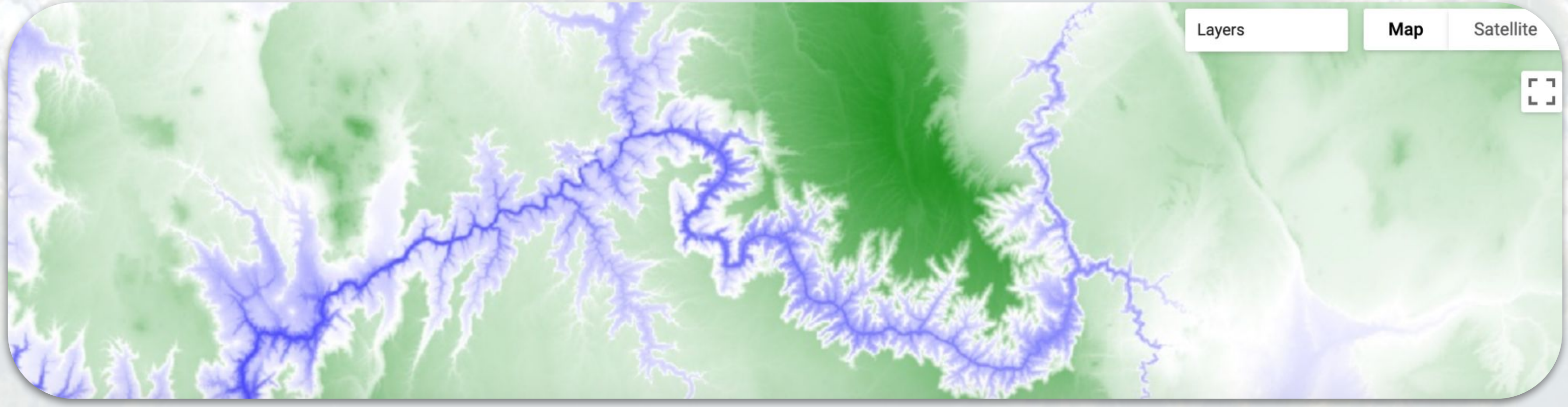
36.2841° **N**, 112.8598° **W**



New Script *

Get Link | Save | Run | Reset | Apps

```
// instantiate an image (here, a global elevation map)
var image = ee.Image('CGIAR/SRTM90_V4');

// center the map at the Grand Canyon and zoom in
Map.setCenter(-112.8598, 36.2841, 9);

// add the image to the Map below, and use a color palette
// going from blue to white to green in the elevation range [0,3000]
Map.addLayer(image, {min: 0, max: 3000,
            palette: ["blue","white","green"]});
```



New Script *

Get Link | Save | Run | Reset | Apps

```javascript
/* NOTE: this may take about a minute to run */
// retrieve 8-band images captured using the Landsat satellite
var landsat = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA');

// take the median of image values over the 2016 year
var median = landsat.filterDate('2016-01-01', '2016-12-31').median();

// display the composite image, using bands B4 (red),
// B3 (green), and B2 (blue) in the red, green, and blue channels,
// respectively --- a natural-color RGB image
Map.addLayer(median, {bands: ['B4', 'B3', 'B2'], max: 0.3});
```

Landsat 8/9 Operational Land Image (OLI) and Thermal Infrared Sensor (TIRS)

Band	Wavelength	Useful for mapping
Band 1 - coastal aerosol	0.43-0.45	Coastal and aerosol studies
Band 2 - blue	0.45-0.51	Bathymetric mapping, distinguishing soil from vegetation and deciduous from coniferous vegetation
Band 3 - green	0.53-0.59	Emphasizes peak vegetation, which is useful for assessing plant vigor
Band 4 - red	0.64-0.67	Discriminates vegetation slopes
Band 5 - Near Infrared (NIR)	0.85-0.88	Emphasizes biomass content and shorelines
Band 6 - Short-wave Infrared (SWIR) 1	1.57-1.65	Discriminates moisture content of soil and vegetation; penetrates thin clouds
Band 7 - Short-wave Infrared (SWIR) 2	2.11-2.29	Improved moisture content of soil and vegetation; penetrates thin clouds
Band 8 - Panchromatic	0.50-0.68	15 meter resolution, sharper image definition
Band 9 - Cirrus	1.36-1.38	Improved detection of cirrus cloud contamination
Band 10 - TIRS 1	10.60-11.19	100 meter resolution, thermal mapping and estimated soil moisture
Band 11 - TIRS 2	11.50-12.51	100 meter resolution, improved thermal mapping and estimated soil moisture

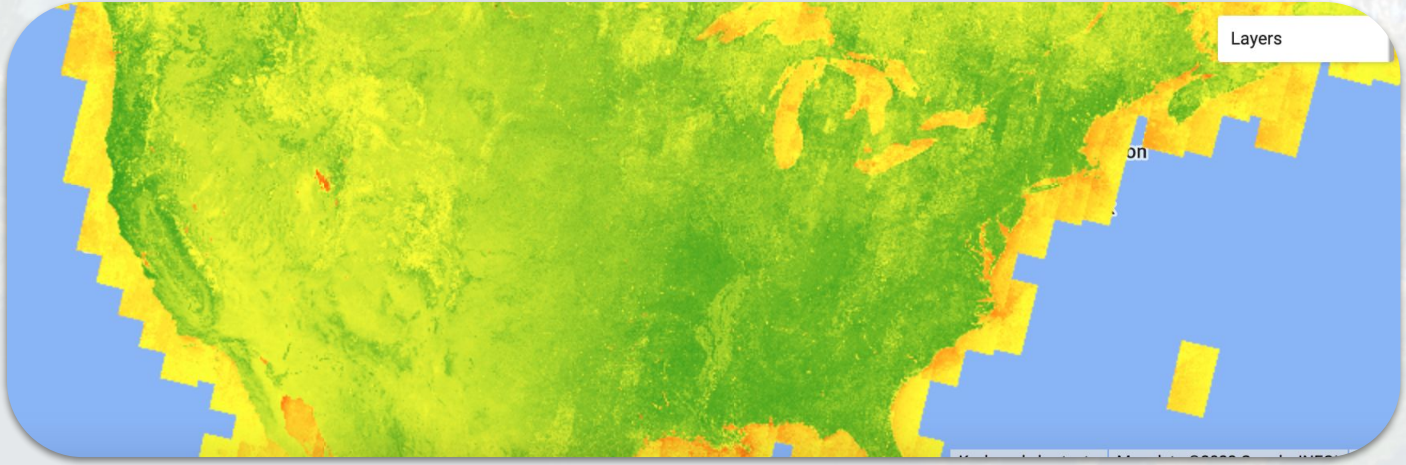
https://www.usgs.gov/faqs/what-are-band-designations-landsat-satellites

New Script * | Get Link | Save | Run | Reset | Apps

```javascript
/* NOTE: this may take about a minute to run */
// retrieve 8-band images captured using the Landsat satellite
var landsat = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA');

// take the median of image values over the 2016 year
var aggregate = landsat.filterDate('2016-01-01', '2016-12-31').median();

// calculate the vegetation index: (NIR - R) / (NIR + R)
var vegetation = aggregate.expression(
  '(NIR - R) / (NIR + R)',
  {"NIR": aggregate.select('B5'), "R": aggregate.select("B4")})

Map.addLayer(vegetation, {min: -1, max: 1,
    palette: ['red', 'yellow', 'green']});
```

Band Glossary

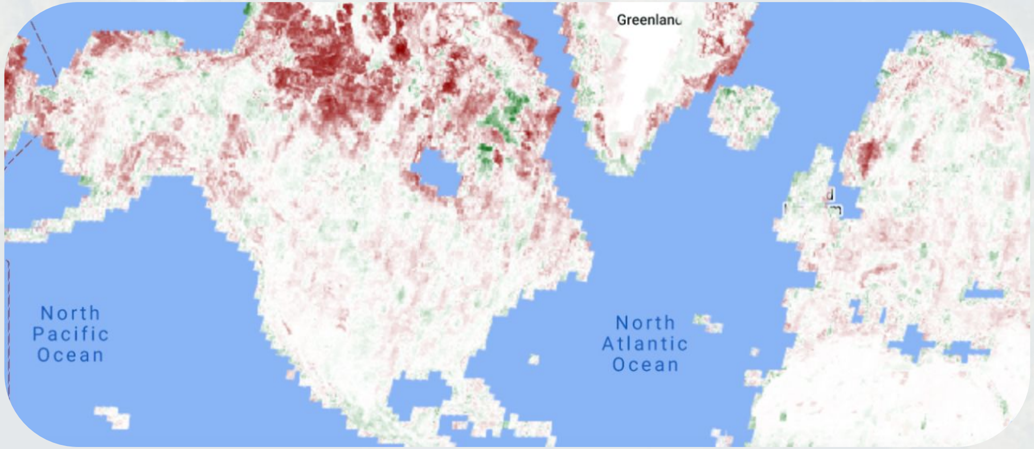
B4: R (Red)
B5: NIR (Near Infrared)



New Script *

Get Link | Save | Run | Reset | Apps

```
// load an already created Landsat composite of the year 1999
var landsat_1999 = ee.Image('LANDSAT/LE7_TOA_1YEAR/1999');

// load an already created Landsat composite of the year 2008
var landsat_2008 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2008');

// get the NDVI (vegetation index) of each
function get_vegetation(image){
  return image.expression('(NIR - R) / (NIR + R)',
  {"NIR": image.select('B5'), "R": image.select("B4")});
}
var vegetation_1999 = get_vegetation(landsat_1999);
var vegetation_2008 = get_vegetation(landsat_2008);

// get the difference in vegetation and Map it
var vegetation_diff = vegetation_2008.subtract(vegetation_1999);
Map.addLayer(vegetation_diff, {min: -1, max: 1,
    palette: ['darkred', 'white', 'darkgreen']});
```



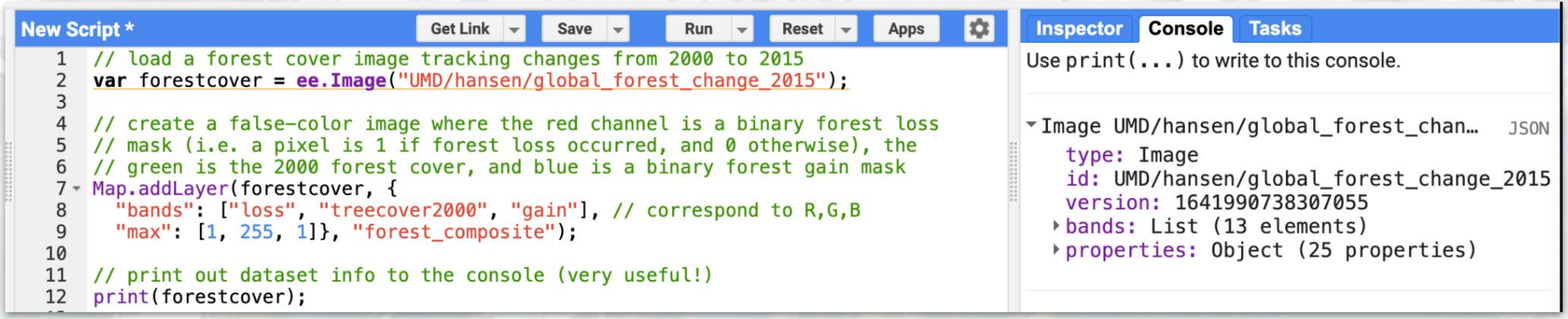
New Script * | Get Link | Save | Run | Reset | Apps

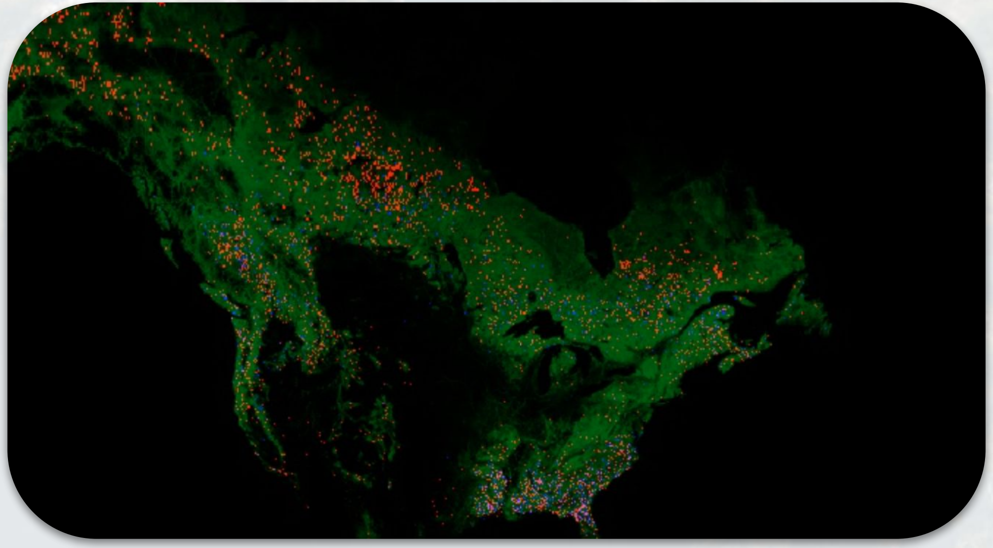
```
// load a forest cover image tracking changes from 2000 to 2015
var forestcover = ee.Image("UMD/hansen/global_forest_change_2015");

// plot the tree cover in the year 2000 (light green for high forestation)
Map.addLayer(forestcover, {"bands": ["treecover2000"],
  "palette": ["black","lightgreen"]}, "treecover2000");

// print out dataset info to the console (very useful!)
print(forestcover);
```

Inspector | Console | Tasks

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

```
▾ Image UMD/hansen/global_forest_chan…    JSON
    type: Image
    id: UMD/hansen/global_forest_change_2015
    version: 1641990738307055
  ▸ bands: List (13 elements)
  ▸ properties: Object (25 properties)
```


New Script *
Get Link
Save
Run
Reset
Apps
// load a forest cover image tracking changes from 2000 to 2015
var forestcover = ee.Image("UMD/hansen/global_forest_change_2015");
// create a false-color image where the red channel is a binary forest loss
// mask (i.e. a pixel is 1 if forest loss occurred, and 0 otherwise), the
// green is the 2000 forest cover, and blue is a binary forest gain mask
Map.addLayer(forestcover, {
"bands": ["loss", "treecover2000", "gain"], // correspond to R,G,B
"max": [1, 255, 1]}, "forest_composite");
// print out dataset info to the console (very useful!)
print(forestcover);
Inspector
Console
Tasks
Use print(...) to write to this console.
Image UMD/hansen/global_forest_chan…
JSON
type: Image
id: UMD/hansen/global_forest_change_2015
version: 1641990738307055
bands: List (13 elements)
properties: Object (25 properties)

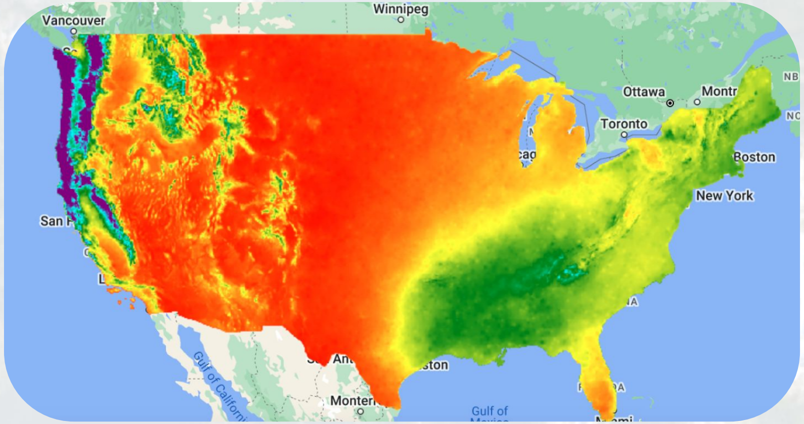


Explore more data at
https://developers.google.com/earth-engine/datasets!

ee.Image("UMD/hansen/global_forest_change_2016_v1_4")

MODIS Reflectance Imagery
ee.ImageCollection(**"MODIS/061/MCD43A4"**);

https://developers.google.com/earth-engine/datasets/catalog/MODIS_061_MCD43A4



Precipitation Data
ee.ImageCollection(**"OREGONSTATE/PRISM/Norm91m"**);

https://developers.google.com/earth-engine/datasets/catalog/OREGONSTATE_PRISM_Norm91m

Earth Art Gallery

JavaScript Exercise

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

Check-Off Form

Fill out this week's attendance form on the cs106s.stanford.edu website !