PEER2PEER

TRANSBOUNDARY WATER SECURITY

Google Earth Engine for Water Resources

Annika Hjelmstad
Debora Yumi de Oliveira
University of Dar es Salaam, Tanzania, 2025

NSF

NATIONAL ACADEMIES Sciences Engineering Medicine

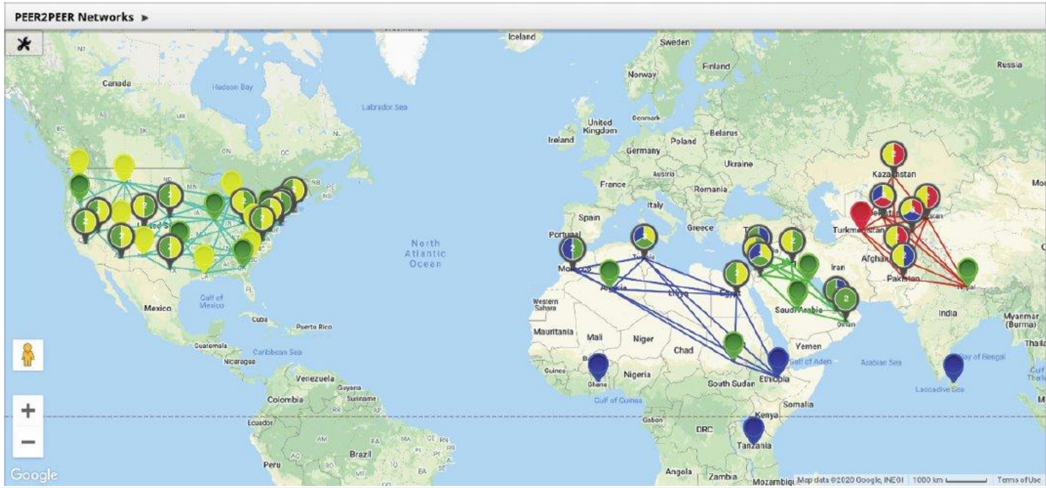
OSU

THE UNIVERSITY OF CALIFORNIA · IRVINE · LET THERE BE LIGHT

HEKIMA NI UHURU

About PEER2PEER

The goal of **PEER2PEER** is to create a **global network of regional water networks** to drive knowledge exchange and data sharing among US researchers and international partners to **advance transboundary water research** and prepare the next generation of water professionals.



About this workshop

- The focus of today's workshop is on using Google Earth Engine for hydrology and water resources, with a focus on developing the following skills:
- **Use remote sensing data for hydrology & water resource applications**
- **Work with large global datasets remotely using GEE servers**
- **Use the Python interface to Google Earth Engine to create an extensible computational environment in a Jupyter notebook**

Agenda

09:00 - 09:40
Introduction to GEE lecture and Jupyter notebook setup

Objectives: Learn what data is available on GEE
Understand how to access GEE resources remotely
Set up computing environment

09:40 - 10:00
Tutorial 1: Exploring GEE catalog

Objectives: Be able to run code in a Jupyter notebook
Learn how to read data from GEE image collections
Learn how to make a map of an image with `geemap`

10:00 - 11:00
Tutorial 2: Surface water change over time

Objectives: Use remote sensing data to characterize surface water
Understand limitations of remote sensing data
Analyze change in surface water area over time

11:00 - 12:00
Tutorial 3: Mapping drought indices

Objectives: ADD objectives

Training materials

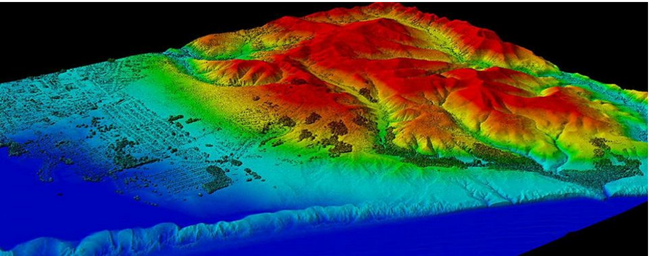
All training materials
are available at:

uci-chrs.github.io/GEE-Training-2025

Google Earth Engine: Remote Sensing Data

Remote Sensing

Pros	Cons
❑ Global coverage ❑ Temporal coverage ❑ Accessibility ❑ Consistency ❑ Low cost for end users	❑ Atmospheric and sun angle interferences ❑ High data volume ❑ Resolution limitations ❑ Sensor and calibration errors ❑ High operational costs



Goodchild et al. (2012):

PNAS

Next-generation Digital Earth

Michael F. Goodchild[a,1], Huadong Guo[b], Alessandro Annoni[c], Ling Bian[d], Kees de Bie[e], Frederick Campbell[f], Max Craglia[c], Manfred Ehlers[g], John van Genderen[e], Davina Jackson[h], Anthony J. Lewis[i], Martino Pesaresi[c], Gábor Remetey-Fülöpp[j], Richard Simpson[k], Andrew Skidmore[f], Changlin Wang[b], and Peter Woodgate[l]

[a]Department of Geography, University of California, Santa Barbara, CA 93106; [b]Center for Earth Observation and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China; [c]Joint Research Centre of the European Commission, 21027 Ispra, Italy; [d]Department of Geography, University at Buffalo, State University of New York, Buffalo, NY 14261; [e]Faculty of Geo-Information Science and Earth Observation, University of Twente, 7500 AE, Enschede, The Netherlands; [f]Fred Campbell Consulting, Ottawa, ON, Canada K2H 5G8; [g]Institute for Geoinformatics and Remote Sensing, University of Osnabrück, 49076 Osnabrück, Germany; [h]D_City Network, Newtown 2042, Australia; [i]Department of Geography and Anthropology, Louisiana State University, Baton Rouge, LA 70803; [j]Hungarian Association for Geo-Information, H-1122, Budapest, Hungary; [k]Nextspace, Auckland 1542, New Zealand; and [l]Cooperative Research Center for Spatial Information, Carlton South 3053, Australia

"The supply of geographic information from satellite-based and ground-based sensors has expanded rapidly, encouraging belief in a new, fourth, or "big data," paradigm of science that emphasizes **international collaboration, data-intensive analysis, huge computing resources, and high-end visualization.**"

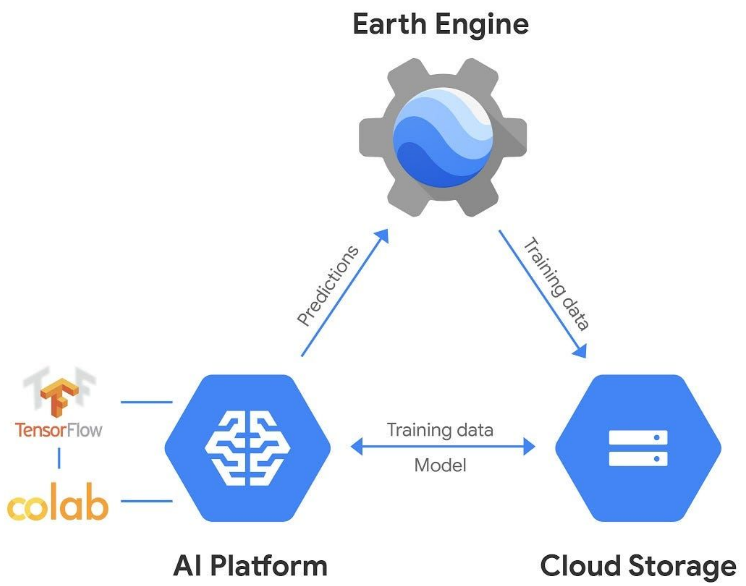
Why Google Earth Engine?

More than 50,000 terabytes of Earth observation data

Google computational infrastructure

Why Google Earth Engine?

- ❖ Powerful Python (and JavaScript) APIs that allow you to integrate large datasets into your analysis
- ❖ Easy Sharing and Collaboration
- ❖ Gridded Global Coverage of Time Series Data
- ❖ No High-End Hardware Required



Google Earth Engine: Data Catalog

Google Earth Engine

Data catalog

- Highlight water resources datasets
- Debora [5 min]
    - Qiusheng Wu website
    - The Awesome GEE (Google Earth Engine) Community Catalog
    - Aqua-monitor
    - Google Earth Engine for Water Resources Management

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

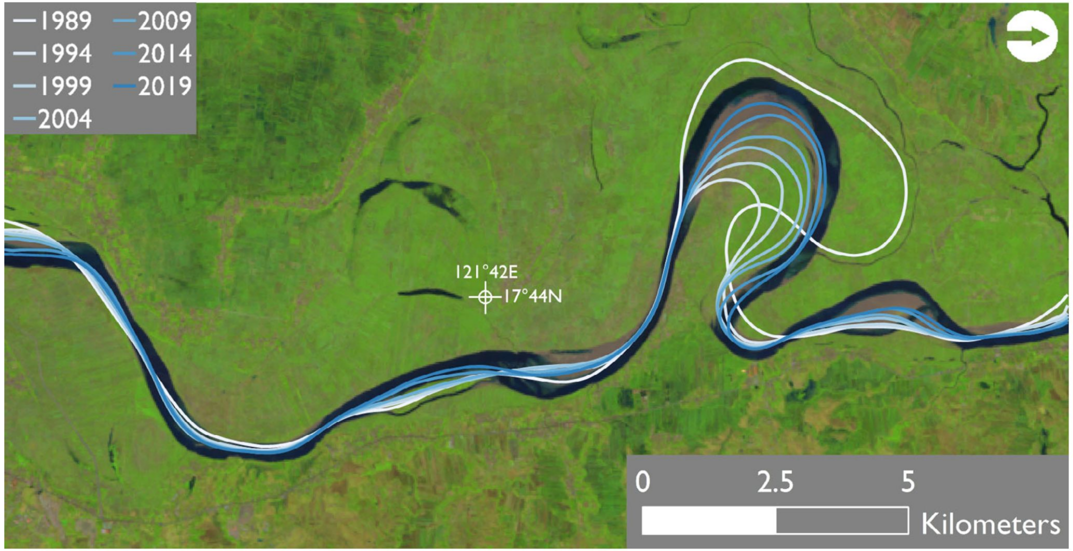
> 4000 new images every day

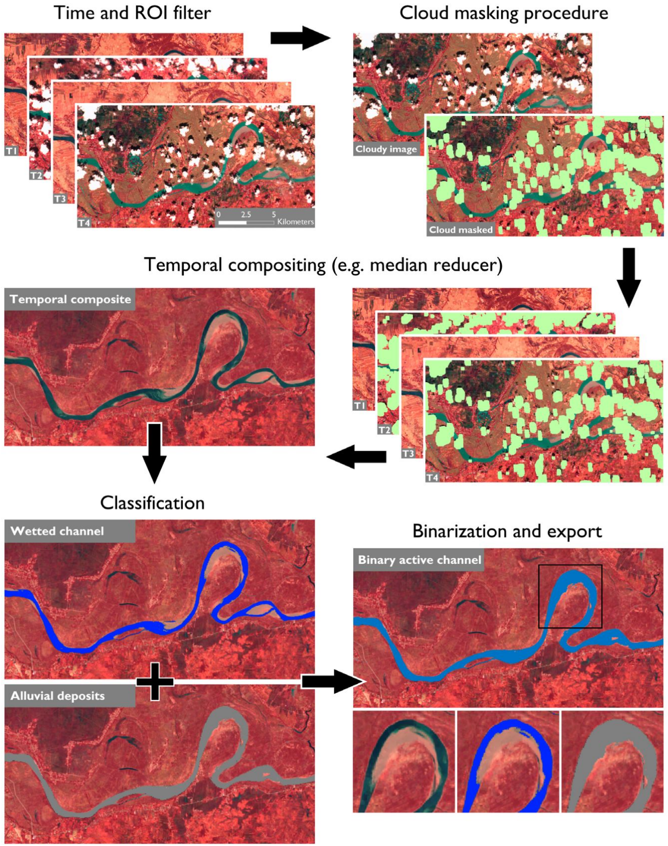
> 5 million images

> 7 petabytes of data

Google Earth Engine:
Hydrology and Water Resources Research Applications

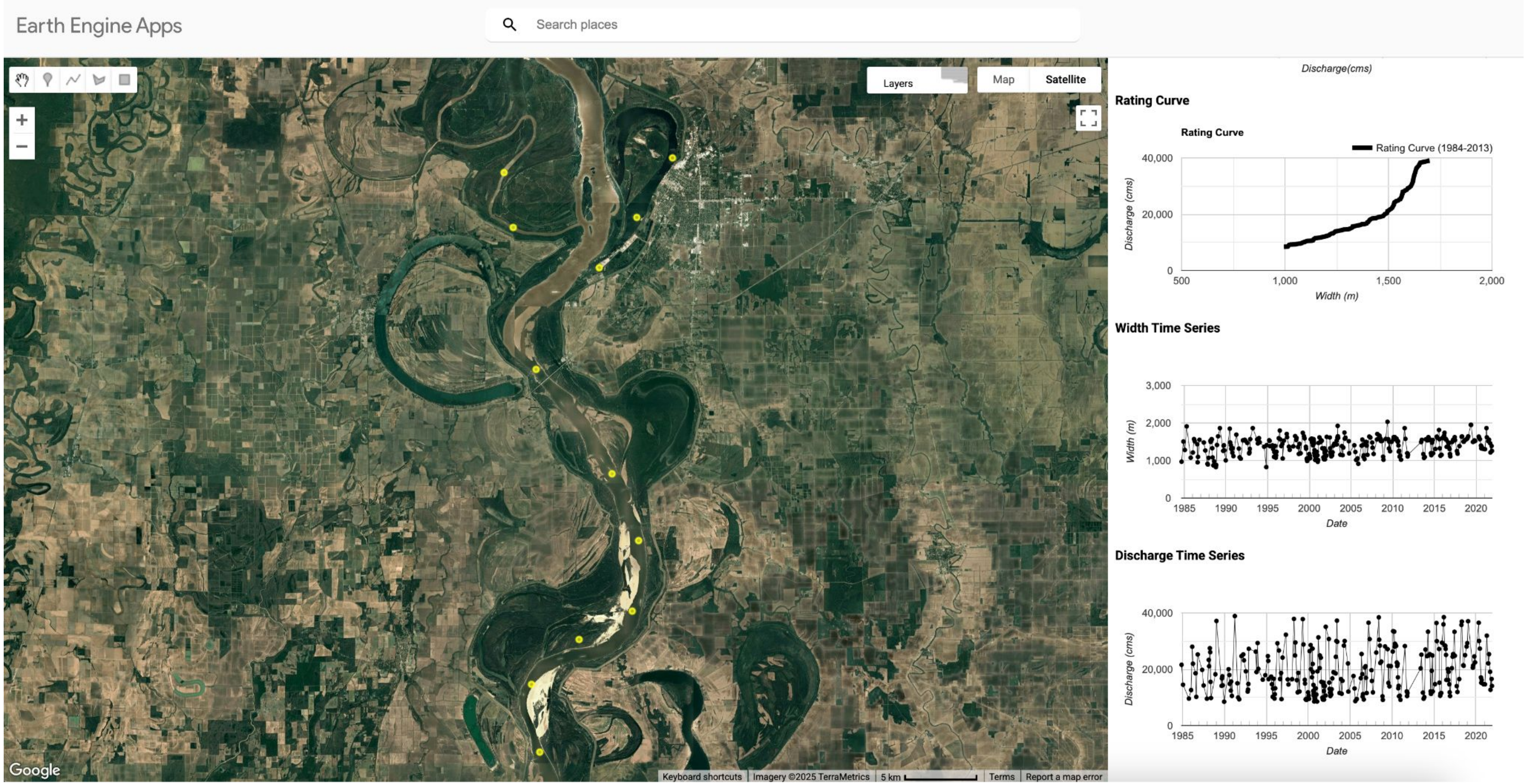
Detecting river channel change





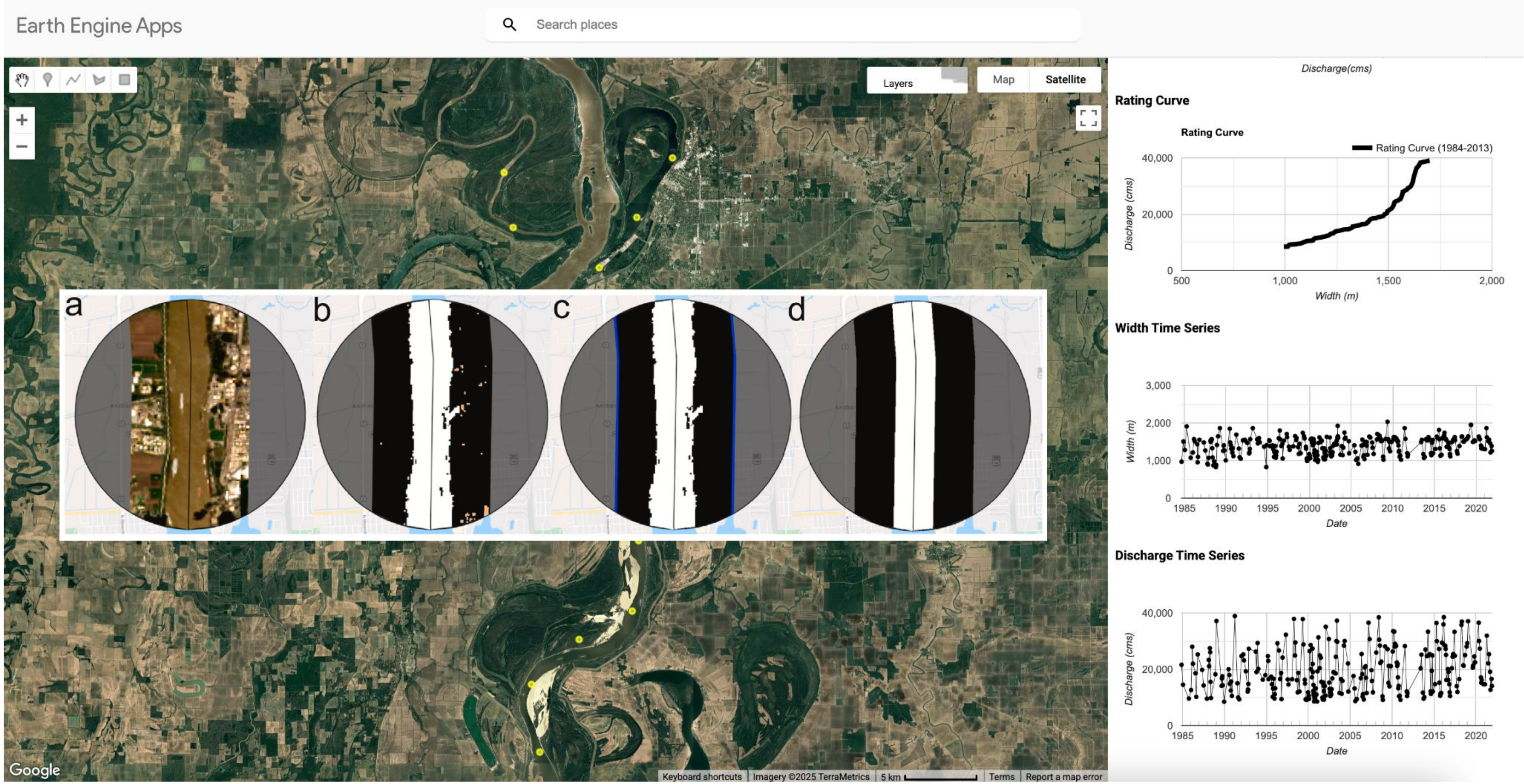
Boothroyd et al. (2021) *WIREs Water*

River discharge estimation



Riggs et al. (2022) *Environmental Modelling & Software*

River discharge estimation



Riggs et al. (2022) *Environmental Modelling & Software*

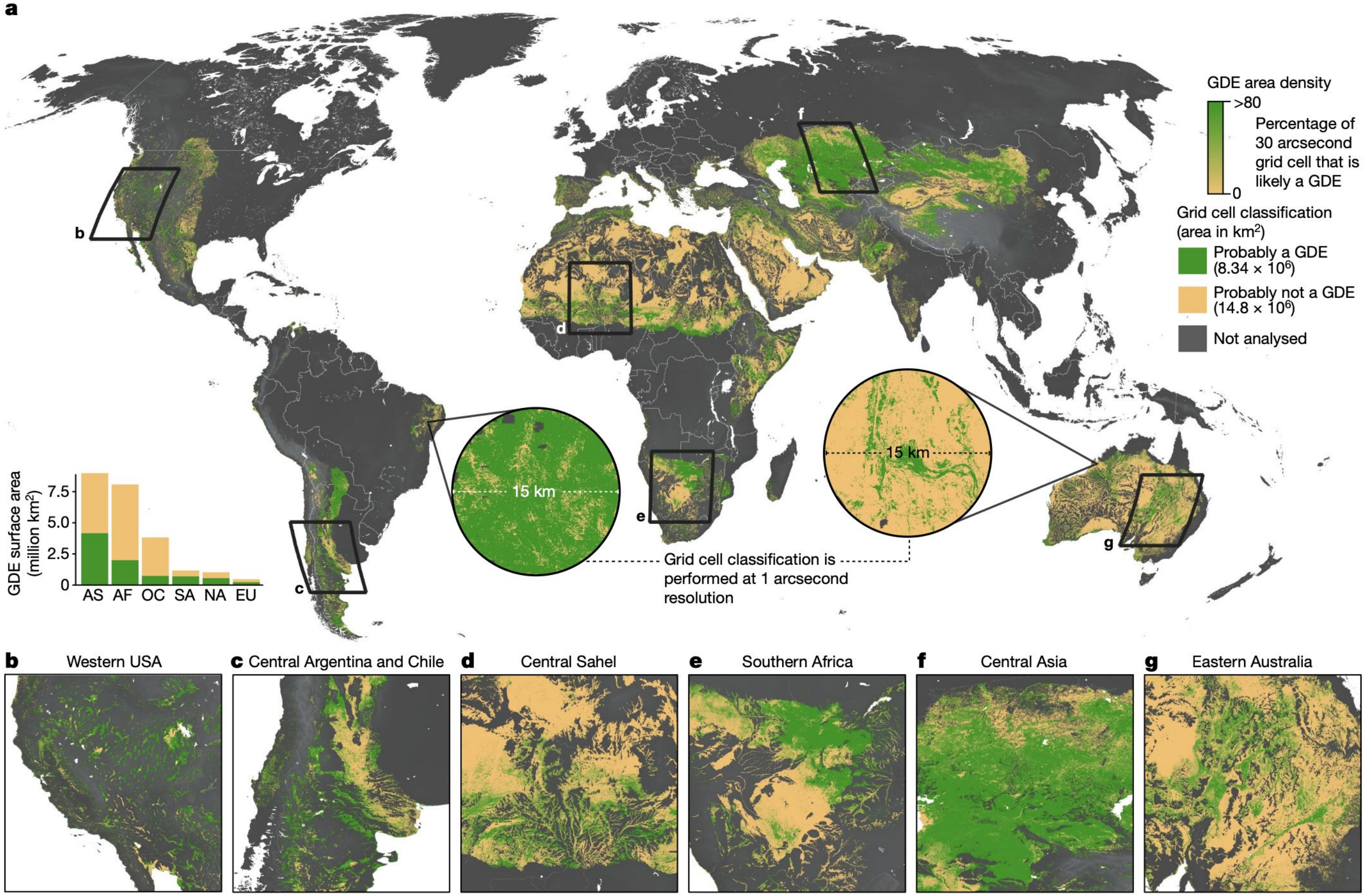
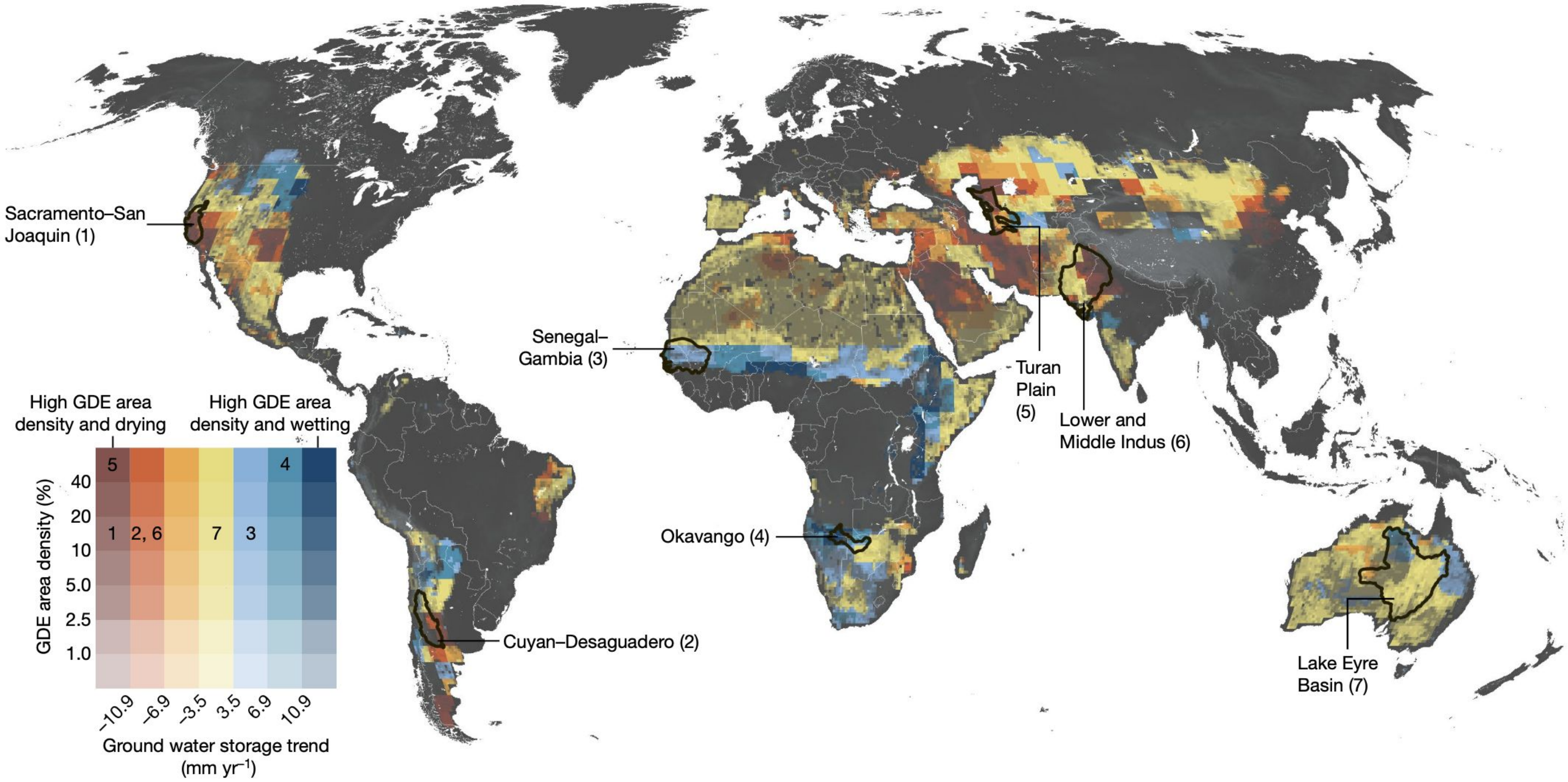


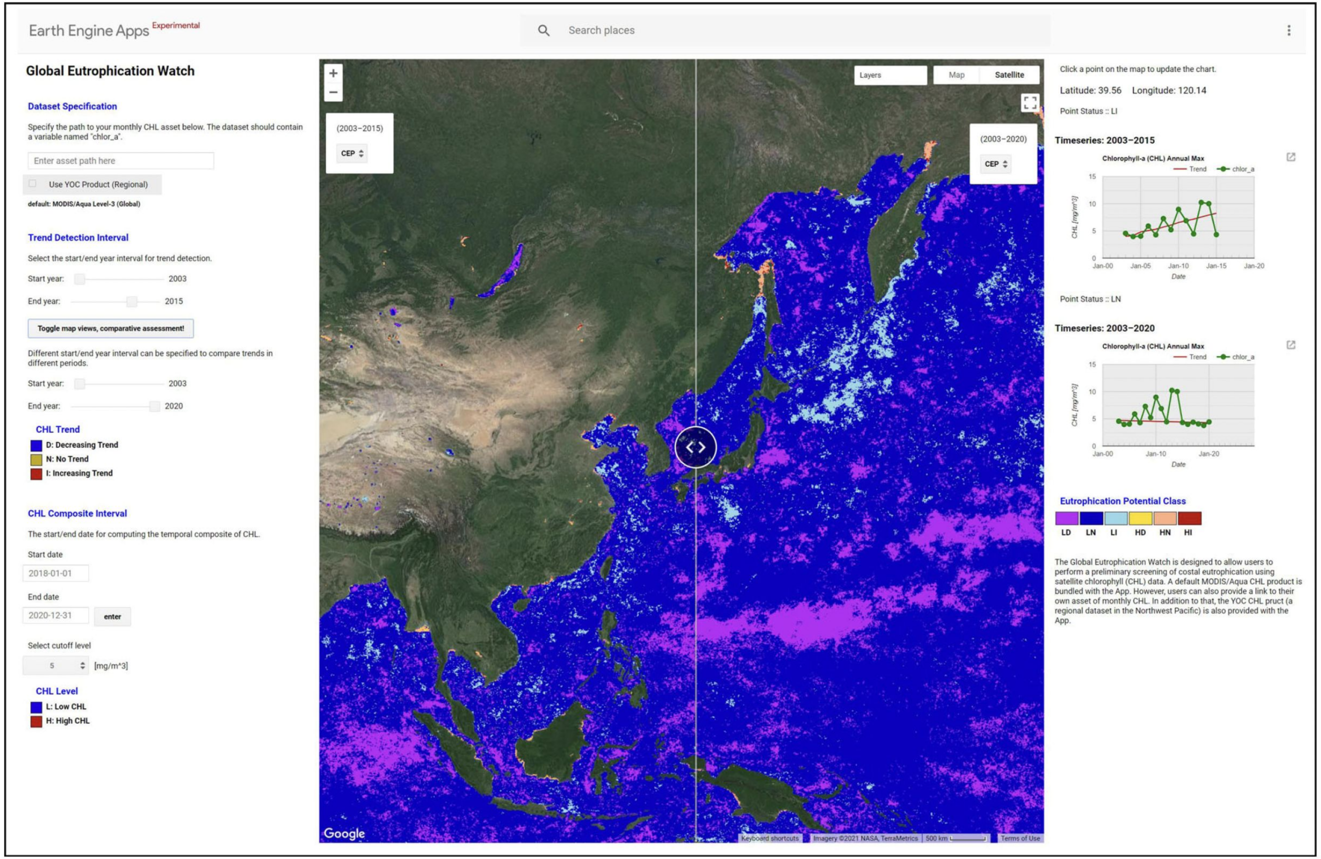
Fig. 1 | Global GDE map. **a**, Global map shows GDE area density at 30 arcsecond resolution (roughly 1 km grids). Call-out circles show binary GDE classification at the full 1 arcsecond resolution (roughly 30 m grids). Bar plot (bottom left) shows GDE surface area distribution across continents. AS, Asia; AF, Africa; OC, Oceania; SA, South America; NA, North America; EU, Europe. **b**–**g**, Regional maps shown at the full 1 arc second resolution for the western USA (**b**), central Argentina and Chile (**c**), the central Sahel region (**d**), southern Africa (**e**), central Asia (**f**) and eastern Australia (**g**). The global map is shown in the Robinson projection whereas all panel insets are shown in geographic projection (latitude and longitude) referenced to the World Geodetic System (WGS) 1984 datum. An interactive version of the full resolution map is available at https://codefornature.projects.earthengine.app/view/global-gde.

Groundwater storage trends



Rohde et al. (2024) *Nature*

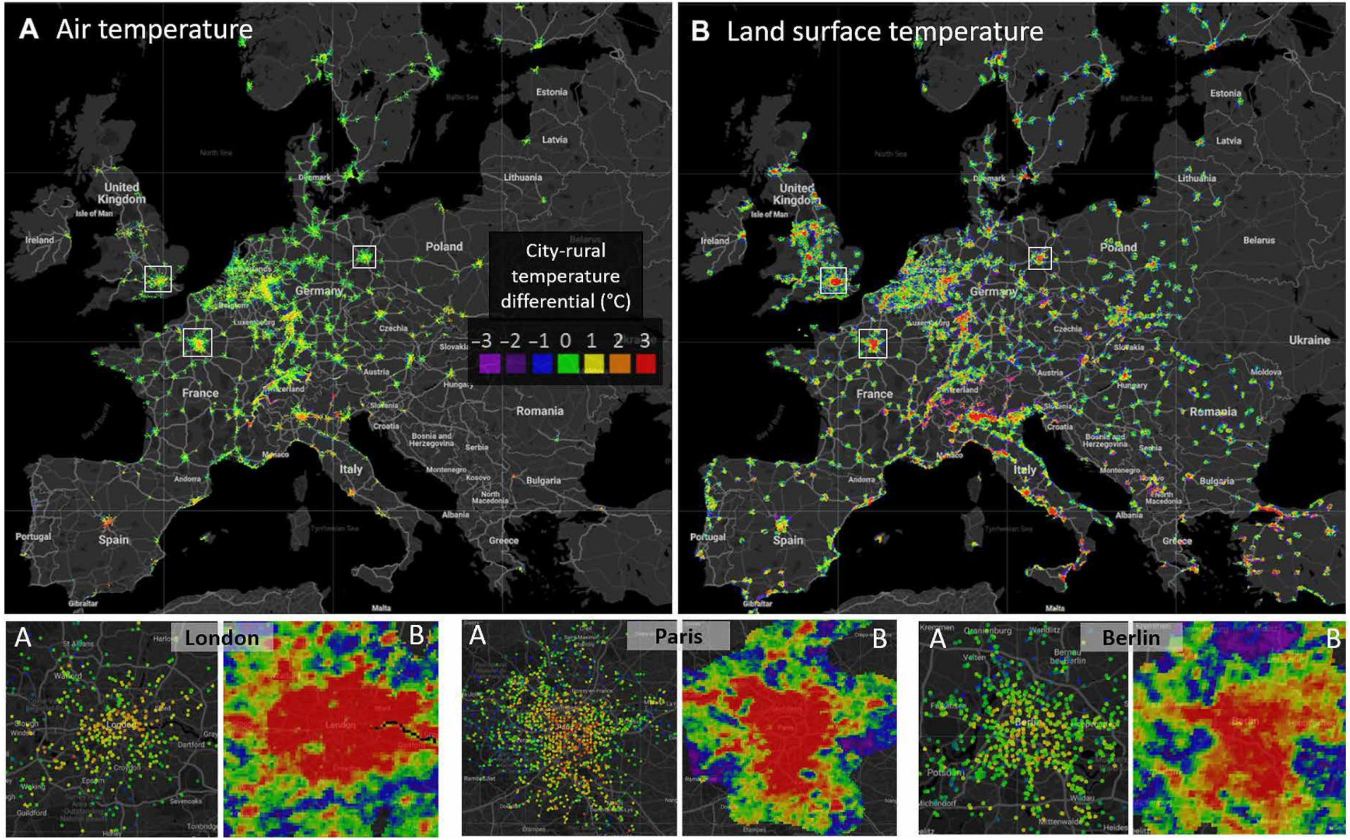
Tracking water quality parameters



Google Earth Engine

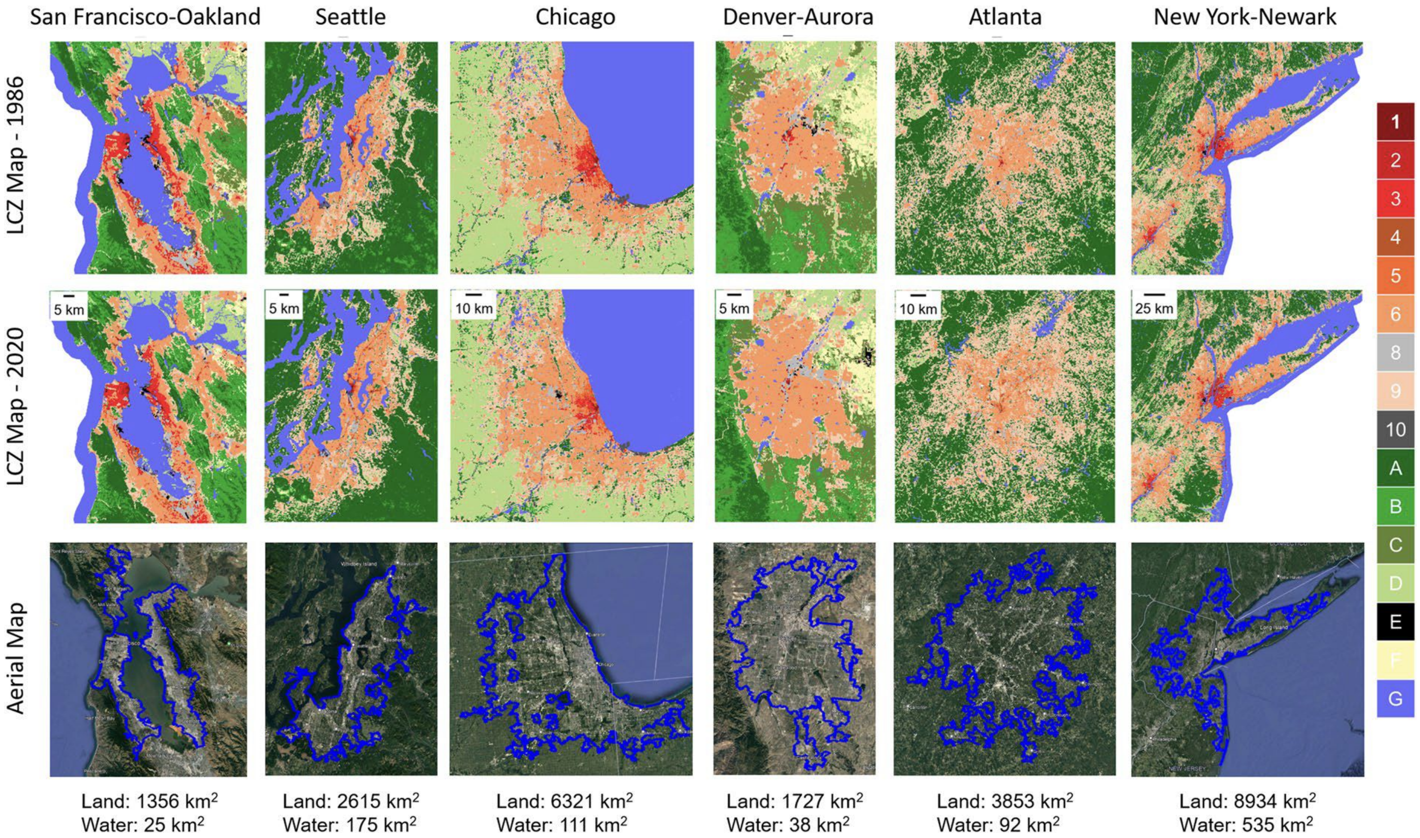
Maúre et al. (2021) *Nature Communications*

Mapping urban heat islands



Venter et al. (2021) *Science Advances*

Mapping urban form into local climate zones



Qi et al. (2024) *Scientific Data*

Morphometric analysis

Table 2
Watershed geometric characteristics.

Morphometric Characteristics	Formula and Description	Reference
Area of Watershed(A)	A = Area of Watershed in km^2	Horton (1945)
Perimeter of Watershed(P)	P=Perimeter of Watershed in km	Horton (1945)
Watershed Length (Lb)	$Lb = 1.312 \times A^{0.568}$, where Lb = Length of Watershed (km) A = Area of Watershed (km^2)	Schumm (1956)
Form Factor (Rf)	$Rf = A/Lb^2$, where A = Area of Watershed (km^2), Lb^2 = Square of the Watershed length	Horton (1945)
Shape Factor or Basin Shape (Bs)	$Bs = Lb^2/A$, where A = Area of Watershed (km^2), Lb^2 = Square of the Watershed length	Horton (1945)
Elongation Ratio (Re)	$Re = 2/Lb \text{ x } (A/\pi)^{0.5}$, where A = Area of Watershed (km^2), Lb = Length of Watershed (km)	Schumm (1956)
Circulatory Ratio (Rc)	$Rc = 4 \times \pi \text{ x } A/P^2$, where A = Area of Watershed (km^2), Lb = Length of Watershed (km)	Miller (1953)
Compactness Coefficient (Cc)	$Cc = P/2\ (\pi A)^{0.5}$, where P = Perimeter of the Watershed (km), A = Area of Watershed (km^2)	Gravelius (1914)
Compactness Constant	$Cc = 0.2821\ P/A^{0.5}$ where, Cc = Compactness Ratio A = Area of Watershed $(km)^2$ P = Perimeter of the Watershed (km)	Horton (1945)
Lemniscate Ratio	$k = Lb^2\ \pi/(4A)$, where Lb^2 = Square of the Watershed length, A = Area of Watershed (km^2)	Chorley (1957); Altıparmak and Türkoğlu (2018)

Google Earth Engine

Before we get started: Google Colab setup

Google Colab Setup

Please open
the instructions at:

uci-chrs.github.io/GEE-Training-2025/setup

Google Colab Setup

1. **Create a Google Cloud project** at https://console.cloud.google.com/earth-engine
    a. Remember your **project ID**
2. **Enable the Google Earth Engine API for your cloud project** at https://console.cloud.google.com/home/dashboard
3. **Register your Google Cloud project as non-commercial** at https://console.cloud.google.com/earth-engine
4. **Open a Colab notebook and authenticate with your Google Cloud project ID** at https://colab.research.google.com/

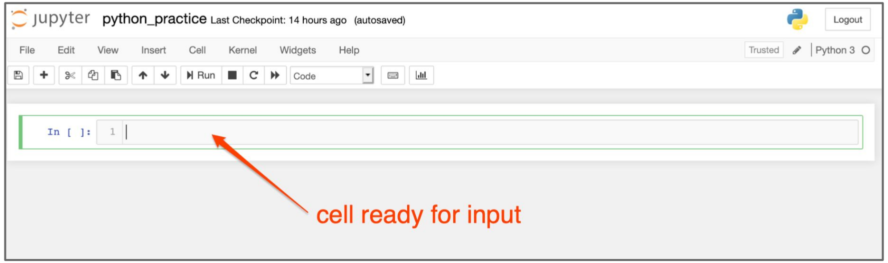
How does it work?

❖ Each code cell in the jupyter notebook executes Python code

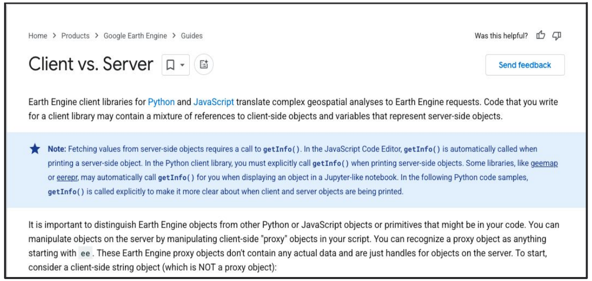
❖ Every operation using the **ee** prefix is a **proxy object** for data and operations on the **Google Earth Engine servers**, not on your computer. **ee** objects and code shouldn't be mixed with regular Python code!

❖ As late as possible, we request the computed data from the GEE servers and bring it into our analysis as a Python data type instead of an ee proxy object calling **.getInfo()** on an **ee** object

❖ Once we've called **.getInfo()**, we have a Python object and the rest of our analysis is done as a regular Python script.



For more information, see the **Client vs. Server page** of the GEE documentation



Home › Products › Google Earth Engine › Guides

Was this helpful?

Client vs. Server

Send feedback

Earth Engine client libraries for Python and JavaScript translate complex geospatial analyses to Earth Engine requests. Code that you write for a client library may contain a mixture of references to client-side objects and variables that represent server-side objects.

★ **Note:** Fetching values from server-side objects requires a call to `getInfo()`. In the JavaScript Code Editor, `getInfo()` is automatically called when printing a server-side object. In the Python client library, you must explicitly call `getInfo()` when printing server-side objects. Some libraries, like geemap or eerepr, may automatically call `getInfo()` for you when displaying an object in a Jupyter-like notebook. In the following Python code samples, `getInfo()` is called explicitly to make it more clear about when client and server objects are being printed.

It is important to distinguish Earth Engine objects from other Python or JavaScript objects or primitives that might be in your code. You can manipulate objects on the server by manipulating client-side "proxy" objects in your script. You can recognize a proxy object as anything starting with `ee`. These Earth Engine proxy objects don't contain any actual data and are just handles for objects on the server. To start, consider a client-side string object (which is NOT a proxy object):

How does it work?

- Each code cell in the jupyter notebook executes Python code
- Every operation using the **ee** prefix is a **proxy object** for data and operations on the **Google Earth Engine servers**, not on your computer. **ee** objects and code shouldn't be mixed with regular Python code!
- As late in the analysis as possible, we request the computed data from the GEE servers and bring it into our analysis as a Python data type instead of an ee proxy object calling **.getInfo()** on an **ee** object
- Once we've called **.getInfo()**, we have a regular Python object and the rest of our analysis

Remember

Anything with the **ee** prefix is a **proxy object** for something on the **GEE server**

After the **.getInfo()** call, you are working with **regular Python code** in your computing environment.

Tutorial 1

Exploring the Data Catalog

Google Earth Engine

Jupyter Notebook for the Tutorial

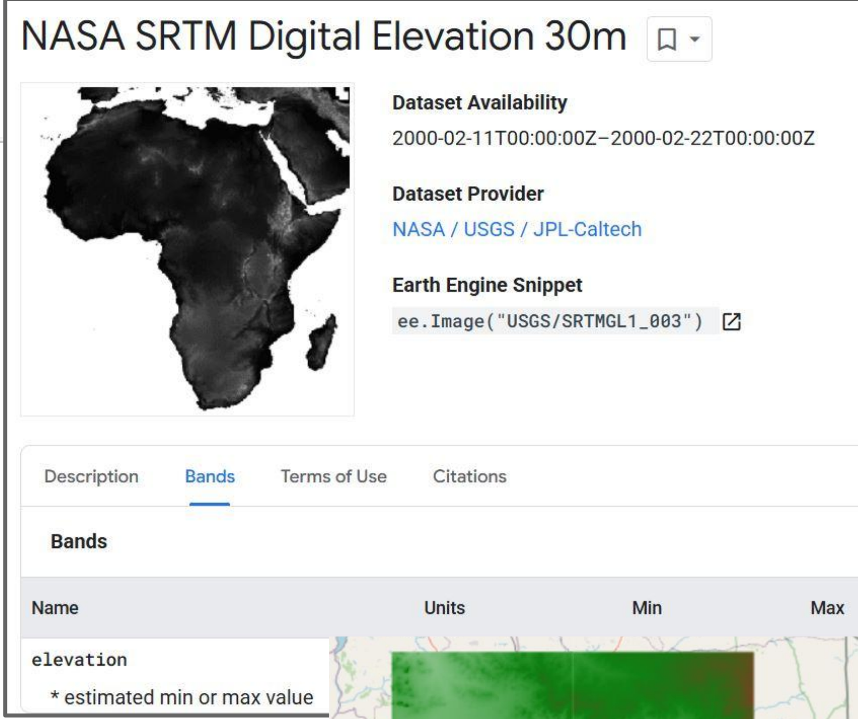
Please open the
Jupyter notebook
here

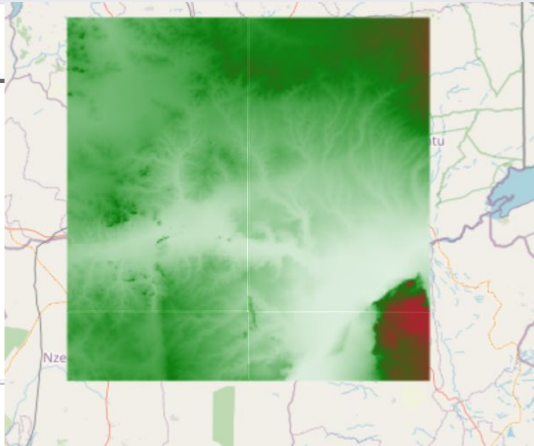
And click **Open in Colab** at the top

Elevation

The **Shuttle Radar Topography Mission** (SRTM) digital elevation data is an international research effort that obtained digital elevation models on a near-global scale and is provided by NASA JPL at a resolution of 30 meters.

Dataset: NASA SRTM Digital Elevation
Coverage: Global
Spatial Resolution: 30 meters
Earth Engine Snippet:
`ee.Image("USGS/SRTMGL1_003")`





Land cover & land use

The **European Space Agency (ESA) WorldCover 10 m** 2021 product provides a global land cover map at a high spatial resolution of 10 meters, derived from Sentinel-1 and Sentinel-2 satellite data. This dataset includes 11 distinct land cover classes.

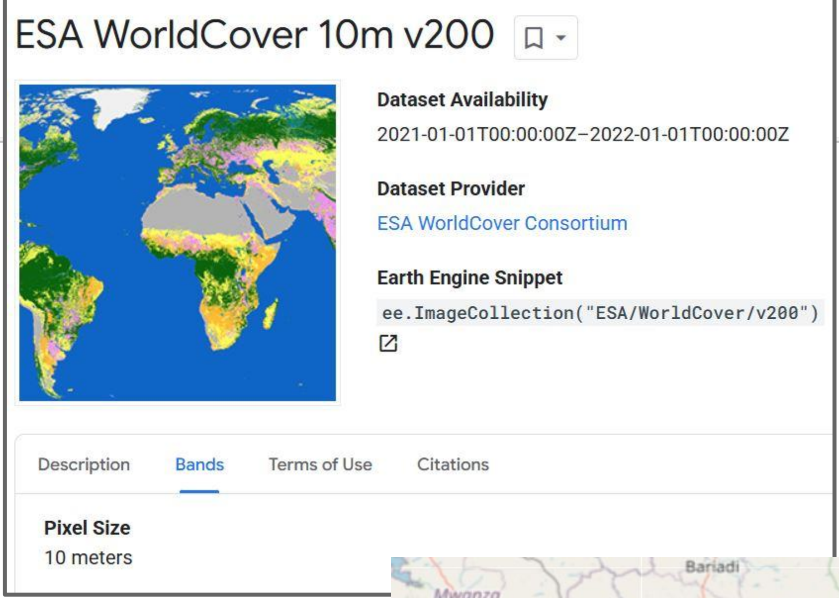
Dataset: ESA WorldCover 10m v200

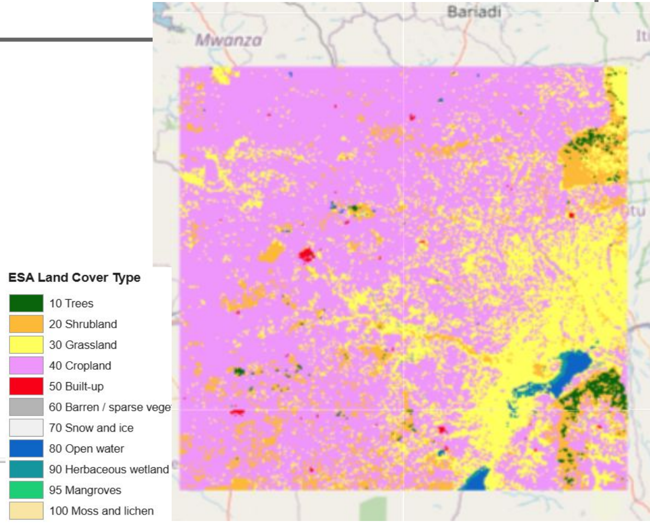
Coverage: Global

Spatial Resolution: 10 meters

Earth Engine Snippet:

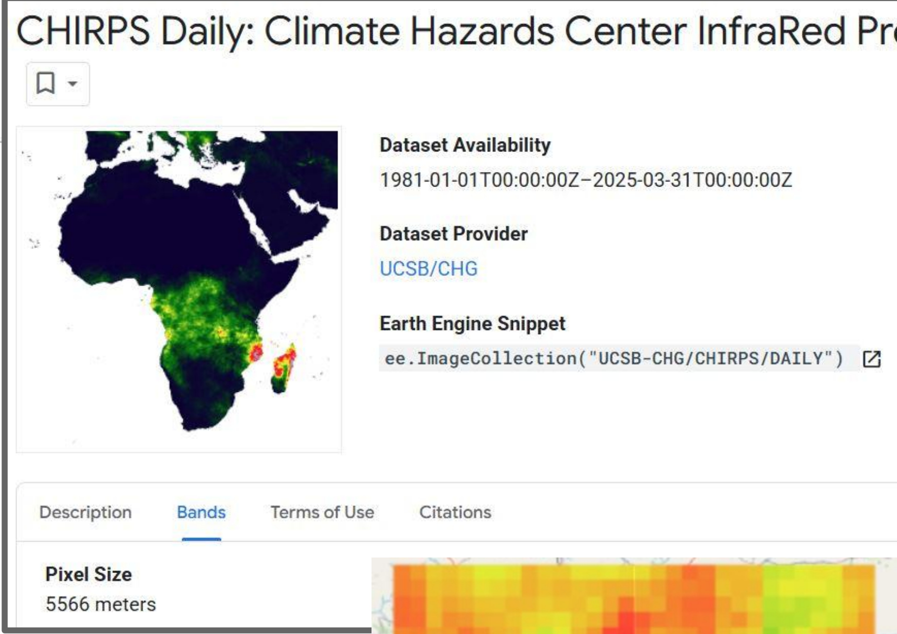
```
ee.ImageCollection("ESA/WorldCover/v200")
```





Precipitation

The **Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS)** is a long-term, high-resolution rainfall dataset developed to support drought early warning and climate monitoring efforts.



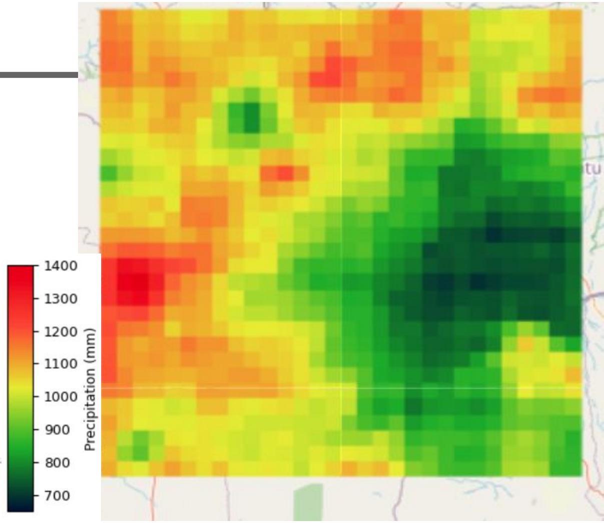
Dataset: CHIRPS Daily

Coverage: Global 50°S to 50°N, 1981-

Spatial Resolution: 5.5 kilometers

Earth Engine Snippet:

`ee.ImageCollection("UCSB-CHG/CHIRPS/DAILY")`



Land Surface Temperature

The MODIS satellite's **MOD11A1 V6.1** product provides daily land surface temperature (LST) and emissivity data. The product includes both daytime and nighttime LST bands and their associated quality assurance layers, as well as MODIS thermal bands 31 and 32.

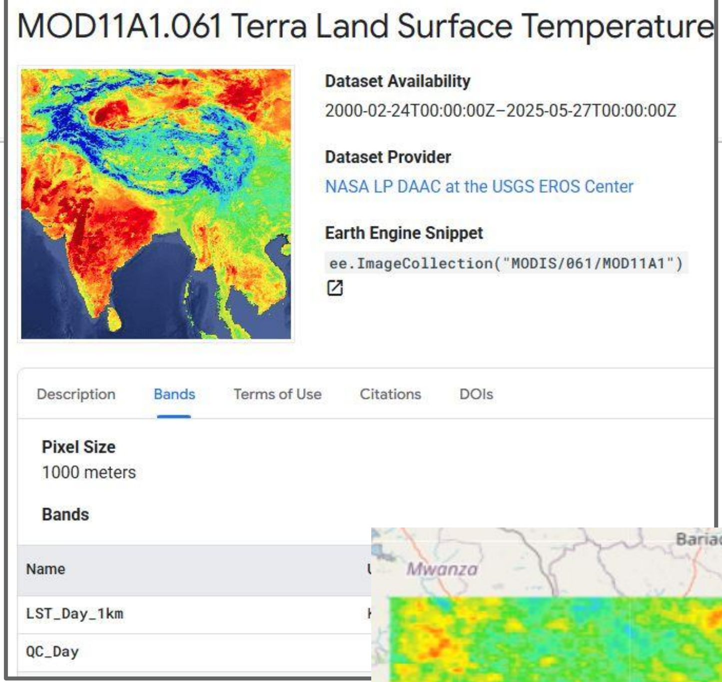
Dataset: MOD11A1 V6.1

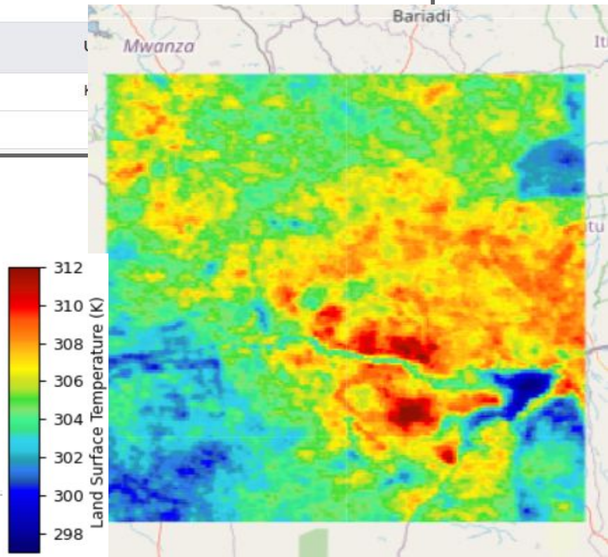
Coverage: Global, 2000-

Spatial Resolution: 1 kilometer

Earth Engine Snippet:

```
ee.ImageCollection("MODIS/061/MOD11A1")
```





Soil moisture

The **Soil Moisture Active Passive (SMAP)** mission is a NASA satellite observatory designed to measure the amount of water in the top 5 cm of soil across the globe, enabling detection of soil moisture changes driven by weather events, seasonal variation, and climatic shifts.

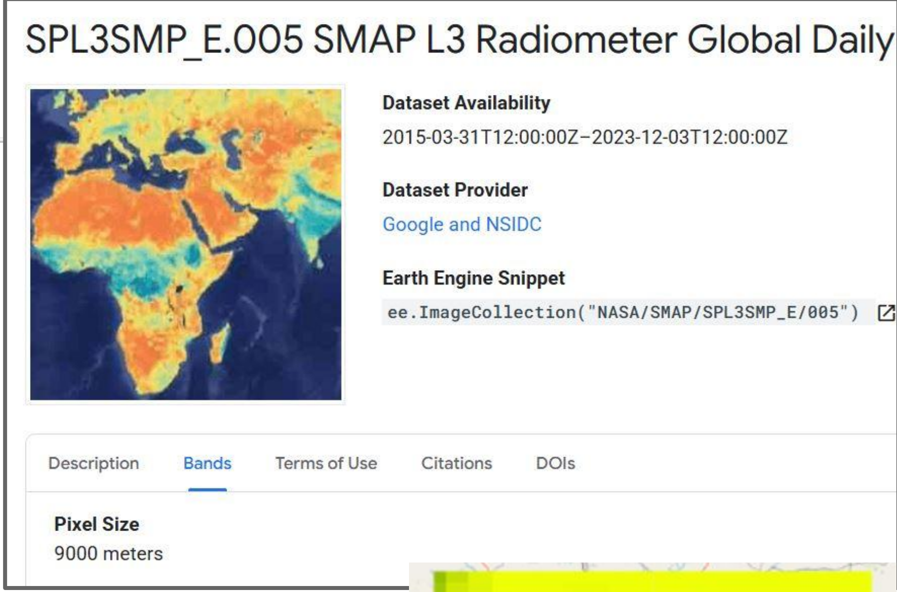
Dataset: SMAP L3 Radiometer

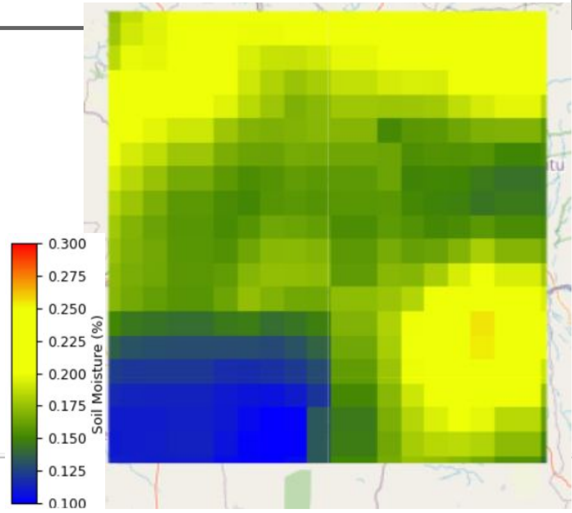
Coverage: Global, 2015-2023

Spatial Resolution: 9 kilometers

Earth Engine Snippet:

```
ee.ImageCollection("NASA/SMAP/SPL3SMP_E/005")
```





How to add Legends

1. Categorical

```
# Add legend to the map
legend_dict = {
    "Low (1000m)": "ffffff",
    "Medium (~1300m)": "008000",
    "High (1600m+)": "8b4513"}

# Generate and attach legend
Map1.add_legend(
    title="SRTM Elevation",
    legend_dict=legend_dict,
    draggable=False,
    output="srtm_legend.html",
)
```

2. Built-in Legend

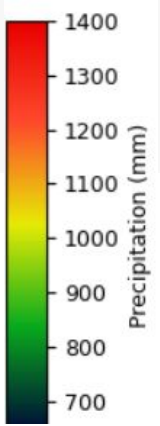
```
# Add the buil-in legend to the map
Map2.add_legend(
    title="ESA Land Cover Type",
    builtin_legend="ESA_WorldCover",
    draggable=False,
    position="bottomleft",
    style={"bottom": "5px"},
)

#Display the Map
Map2
```

3. Colorbar Legend

```
# Visualization parameters
precip_vis = {'min': 650,'max': 1400,
    'palette': ['001137', '0aab1e',
                'e7eb05', 'ff4a2d', 'e90000'],}

# Add a vertical colorbar Legend to the map
Map3.add_colorbar(
    precip_vis,
    label="Precipitation (mm)",
    layer_name="Yearly Rainfall",
    orientation="vertical",
    position="bottomleft",
    transparent_bg=True,
)
```



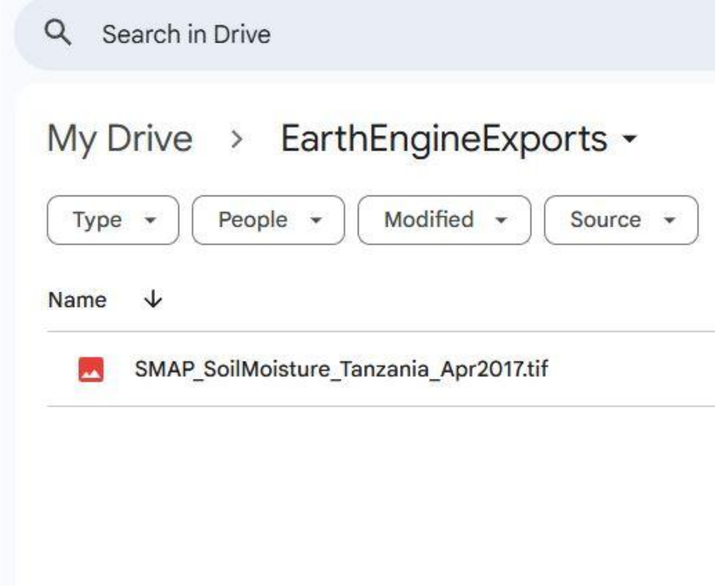
Downloading Datasets to Google Drive

After processing remote sensing datasets in Google Earth Engine (GEE), you can export the results to Google Drive using the **Export.image.toDrive()** function in Python. Follow the steps given below to access the download the data for your study region.

Export the image to your Google Drive

```python
task = ee.batch.Export.image.toDrive(
    image=soil_moisture_am,
    description='SMAP_SoilMoisture_Apr2017',
    folder='EarthEngineExports',  # Your Drive folder name
    fileNamePrefix='SMAP_SoilMoisture_Tanzania_Apr2017',
    region=aoi,
    scale=10000,  # SMAP native resolution ~10 km
    crs='EPSG:4326',
    maxPixels=1e13
)

# Start the export task
task.start()
```



Tutorial 2

Surface Water Change Over Time

Google Earth Engine

Jupyter Notebook for the Tutorial

Please open the Jupyter notebook **here**

And click Open in Colab at the top

Surface water dataset

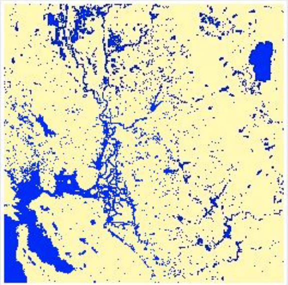
❖ Derived from over 4 million images from the **Landsat** 5, 7, and 8 missions

❖ **Temporal coverage**: Monthly, 1984-2021

❖ **Spatial coverage**: 30m, Global (except for the poles)

❖ For more details, see the [corresponding paper in the journal Nature]()

JRC Monthly Water History, v1.4



Dataset Availability

1984-03-16T00:00:00Z–2022-01-01T00:00:00Z

Dataset Provider

EC JRC / Google

Earth Engine Snippet

`ee.ImageCollection("JRC/GSW1_4/MonthlyHistory")`

Description | Bands | Image Properties | Terms of Use | Citations

Pixel Size

30 meters

Bands

Name	Description
`water`	Water detection for the month.

Bitmask for water

- Bits 0-1: Water detection
  - 0: No data
  - 1: Not water
  - 2: Water

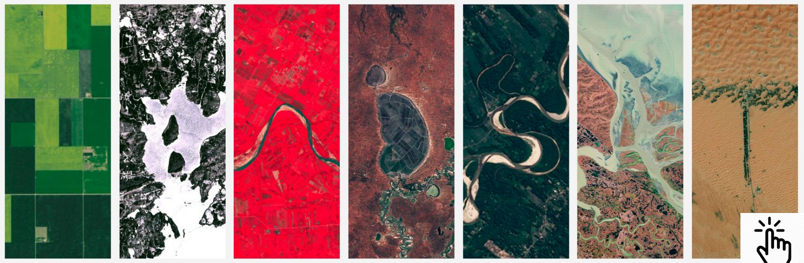
Landsat

- The Landsat program (led by NASA and the USGS) is a set of low-earth orbit, passive **satellite Earth imaging missions** that date back to 1972.
- Landsat provides top-of-atmosphere reflectance & brightness temperature images

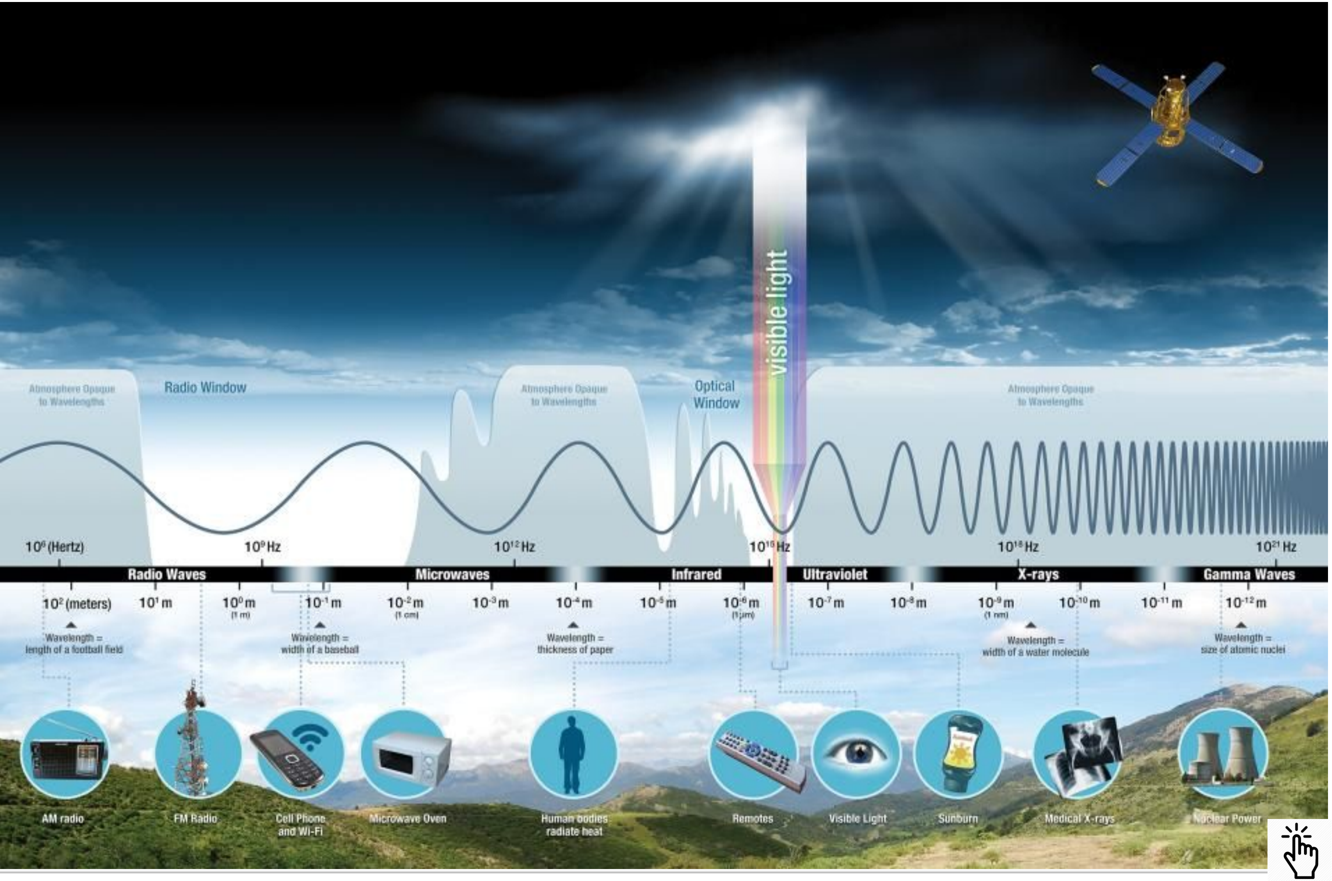
USGS
science for a changing world

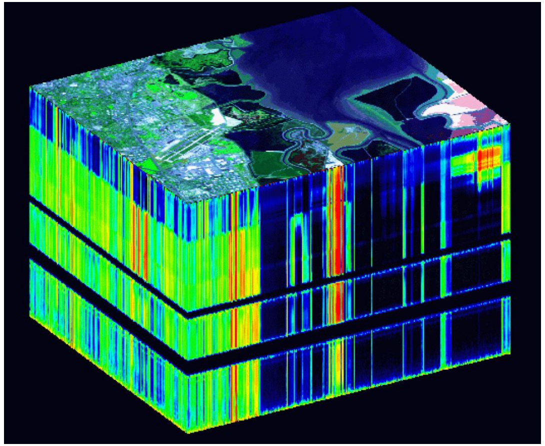
NASA Landsat Science

visible earth
Landsat Gallery



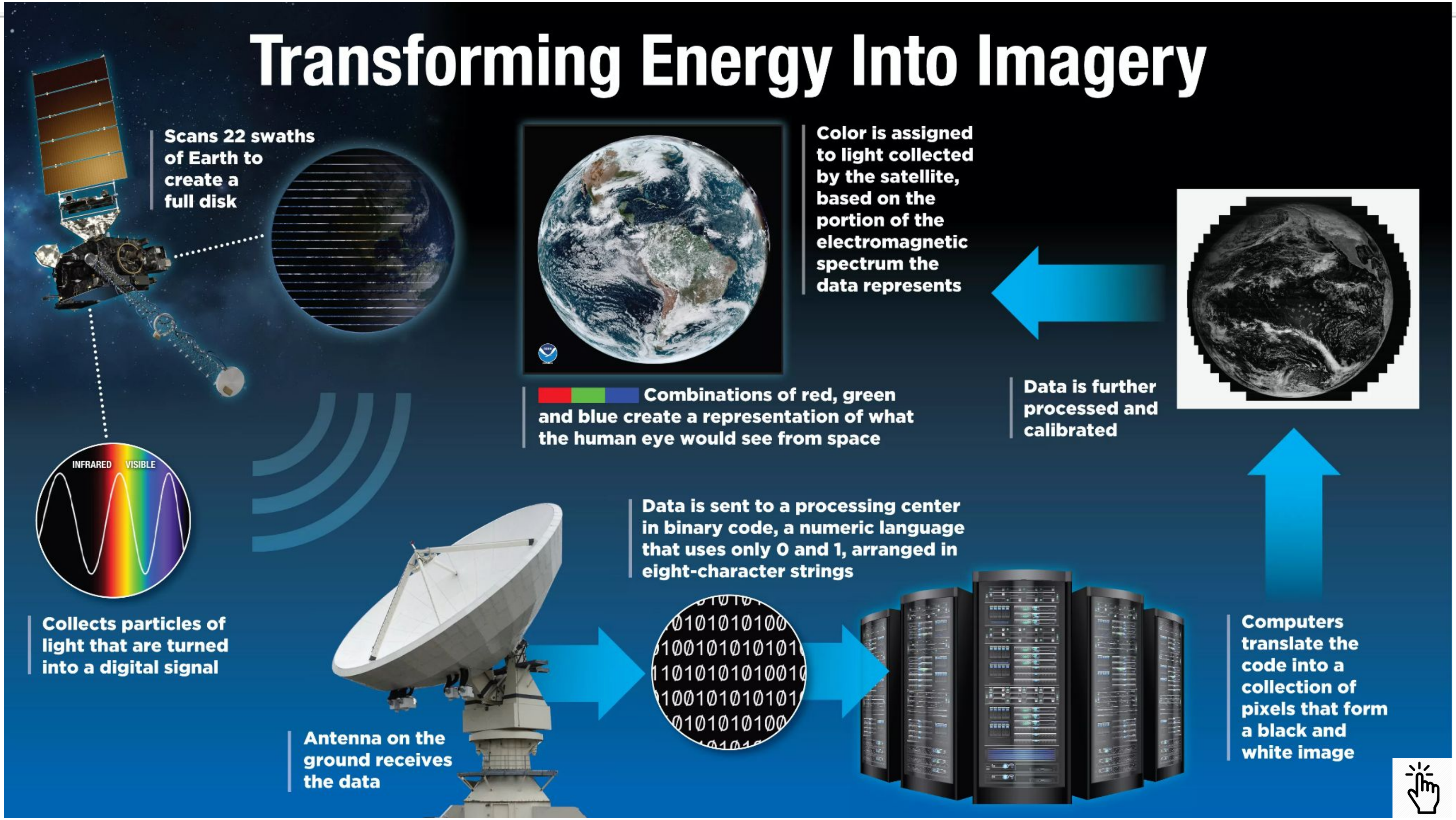
Satellite remote sensing data: a primer





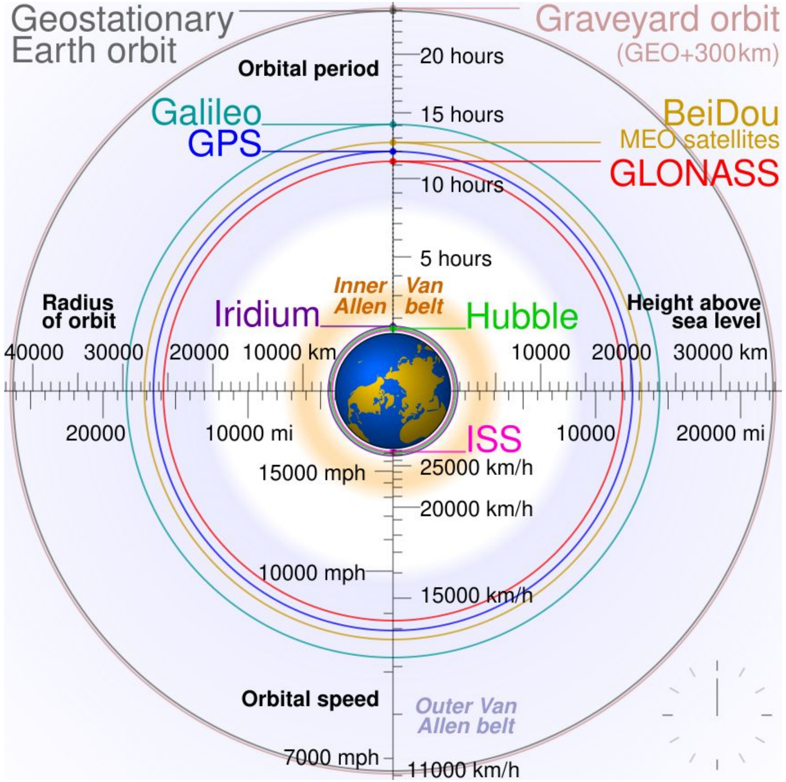
Any 3 channels can substitute the visible red, green, and blue to produce a **false color** image, which can reveal details not available in the visible spectrum

Satellite remote sensing data: a primer

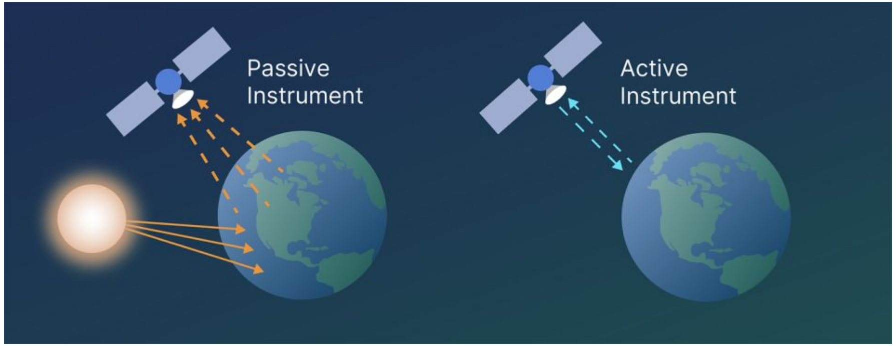


Types of satellites

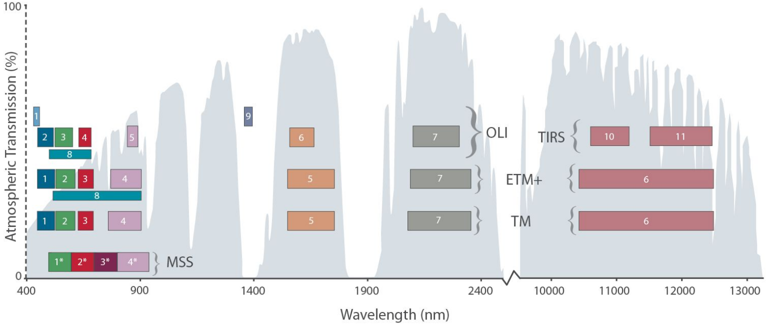
Type & path of orbit



Sensing method



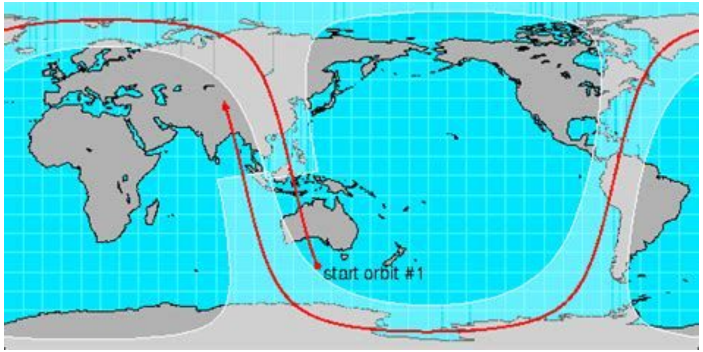
Spectral bands



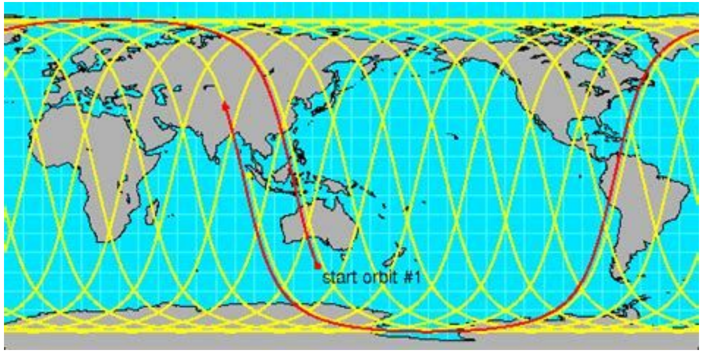
* MSS bands 1–4 were known as bands 4–7, respectively, on Landsats 1–3

What makes satellites prone to errors and missing data?

- Equipment failure
- Less coverage & resolution in early satellite missions
- Limited scan coverage depending on the type and orbital path of a satellite
- Satellites only directly measure **electromagnetic radiation** (EMR)
  - We use a variety of algorithms to process EMR data from different bands into estimates of different kinds of earth observations. These models are imperfect.



Polar orbiting satellite path



Tutorial 3

Mapping Drought

Jupyter Notebook for the Tutorial

Please open the
Jupyter notebook

ADD URL **here**

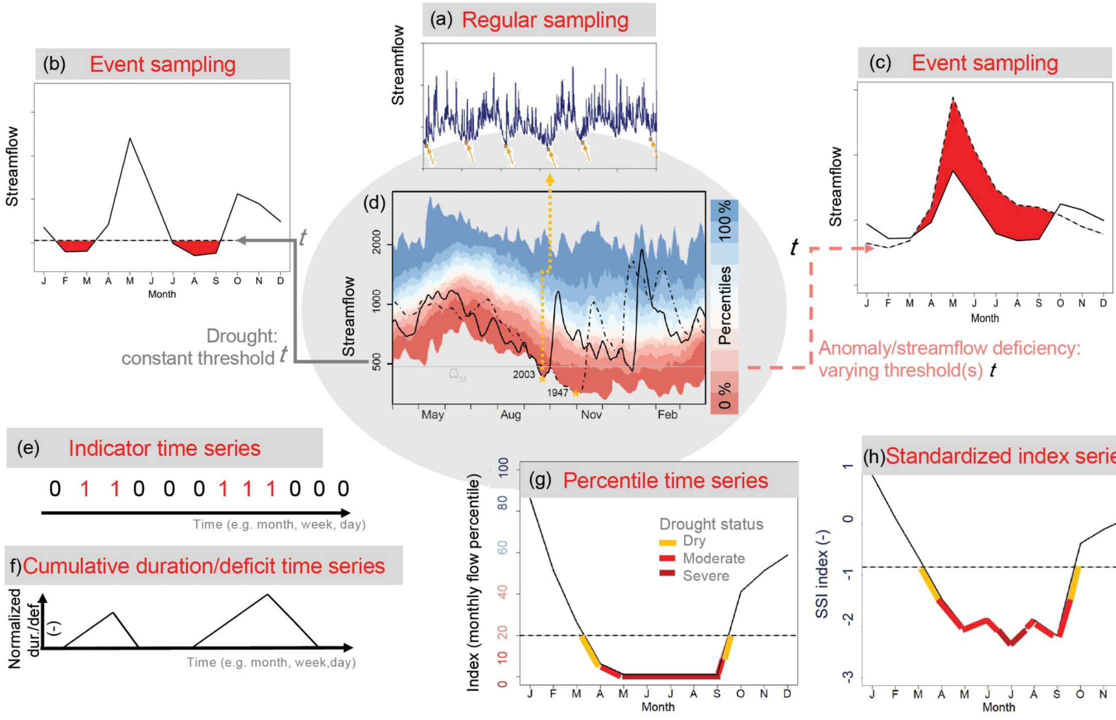
And click Open in Colab at the top

Types of droughts

- Agricultural
- Meteorological
- Hydrological
- Socioeconomic

Drought indicators

- SPI
- SPEI
- SSI
- etc.



Lena Tallaksen Dooge Medal's Talk

- Drought type matters
- Drought definition matters
- Drought index derivation matters
- Catchment storage matters
- Climate change matters

Standardized Precipitation Index (SPI)

[How to compute]

[Scale]

- D0
- D1
- D2
- D3
- D4

Total Water Storage Anomaly

Sum of all above and below surface water storages

- Canopy water
- Rivers and lakes
- Soil moisture
- Groundwater

[GRACE]

[Scale]

- Drier than normal
- Near normal
- Wetter than normal