



PEER2PEER
TRANSBOUNDARY WATER SECURITY

Google Earth Engine for Water Resources

Annika Hjelmstad

Debora Yumi de Oliveira

University of Dar es Salaam, Tanzania, 2025

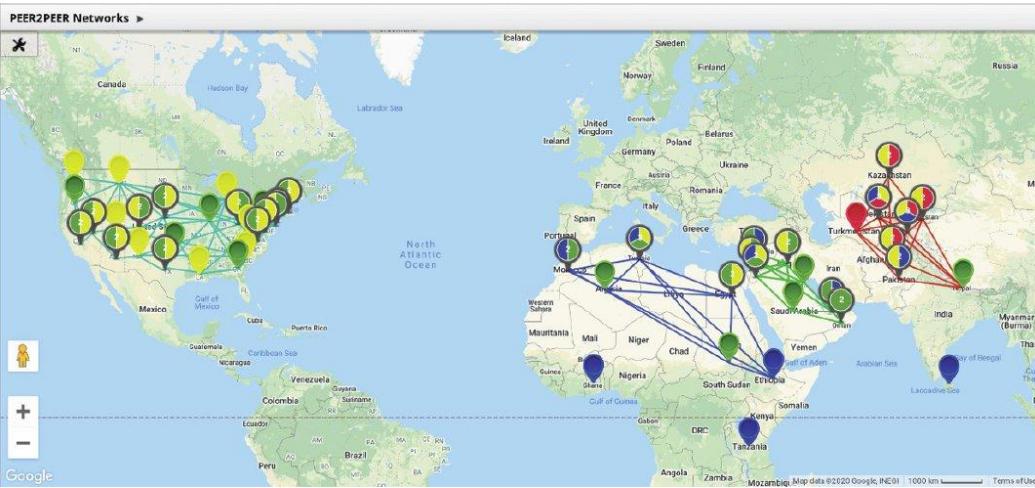


About PEER2PEER

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.

AccelNet: PEER2PEER International Convergence Research Networks in Transboundary Water Security

SHARE f t in e



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 in an extensible computational environment**



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: Water storage changes at the basin scale

Objectives: Analyze changes in total water storage at the basin scale

Training materials

All training materials
are available at:

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



Can also type in:
tinyurl.com/2ukc8v2e

Google Earth Engine: Remote Sensing Data

Google Earth Engine stores > 50 petabytes of earth observation data

Image source: <https://datacenters.google/photo-gallery/>



Image source:

[https://commons.m.wikimedia.org/wiki/File:NASA_Earth_satellites_currently_operating_\(9-2013\)_\(_8703587212\).jpg](https://commons.m.wikimedia.org/wiki/File:NASA_Earth_satellites_currently_operating_(9-2013)_(_8703587212).jpg)



GEE servers store and provide computational access to satellite imagery and other GIS data

Remote Sensing

Pros

- Global coverage
- Temporal coverage
- Accessibility
- Consistency
- Low cost for end users

Cons

- Atmospheric interference
- High data volume
- Resolution limitations
- Sensor and calibration errors
- High operational costs

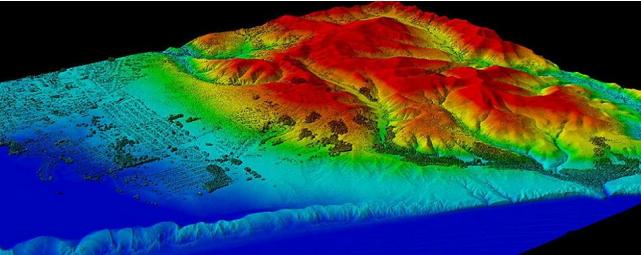


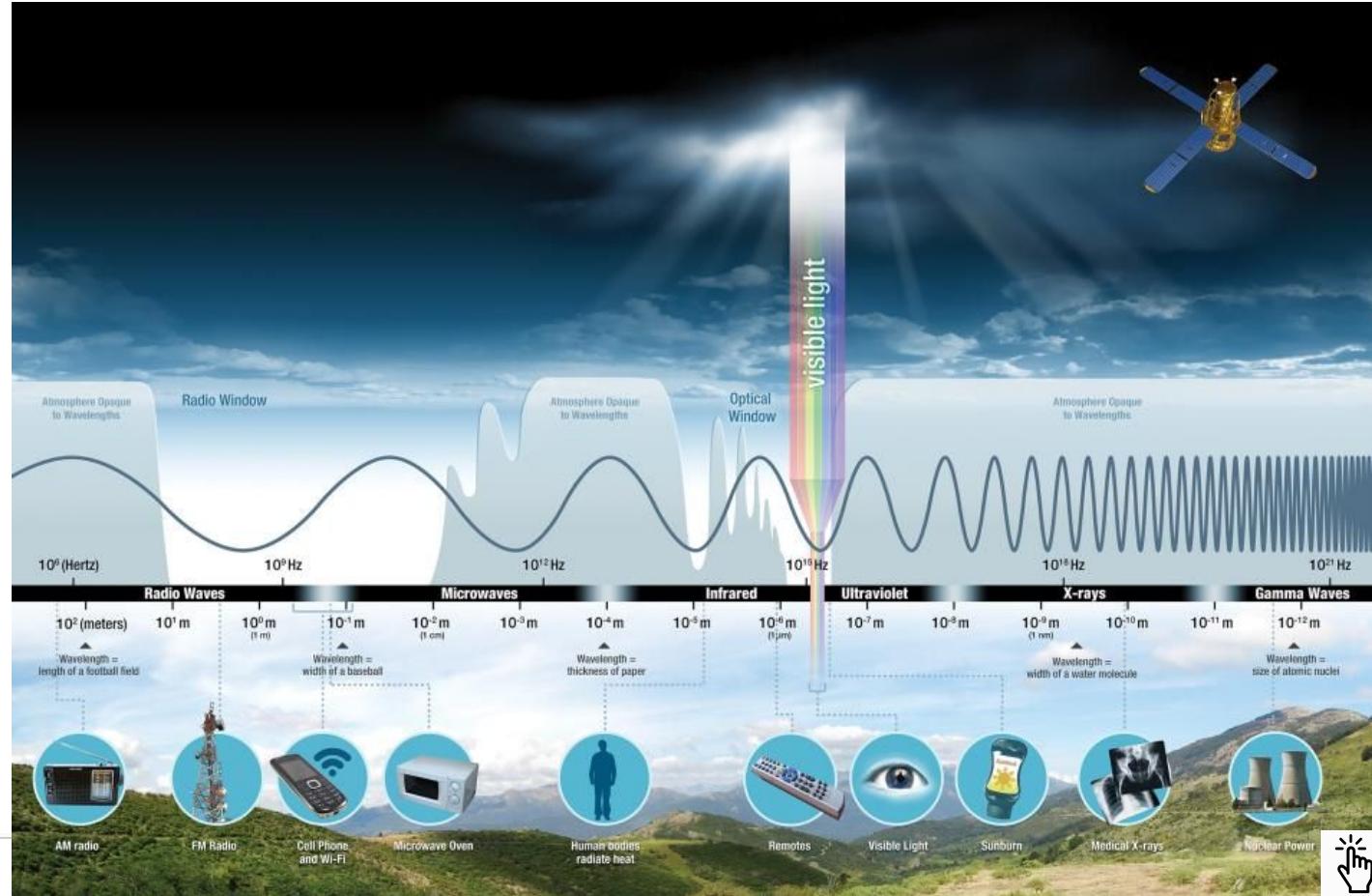
Image source:
<https://geodesy.noaa.gov/INFO/facts/remote-sensing.shtml>



Image source: Geospatialworld.net

Satellite remote sensing data: a primer

Image source: <https://www.earthdata.nasa.gov/learn/earth-observation-data-basics/remote-sensing>



Satellites measure **light*** reflected from the earth at different wavelengths, which we can convert into useful information

*electromagnetic radiation

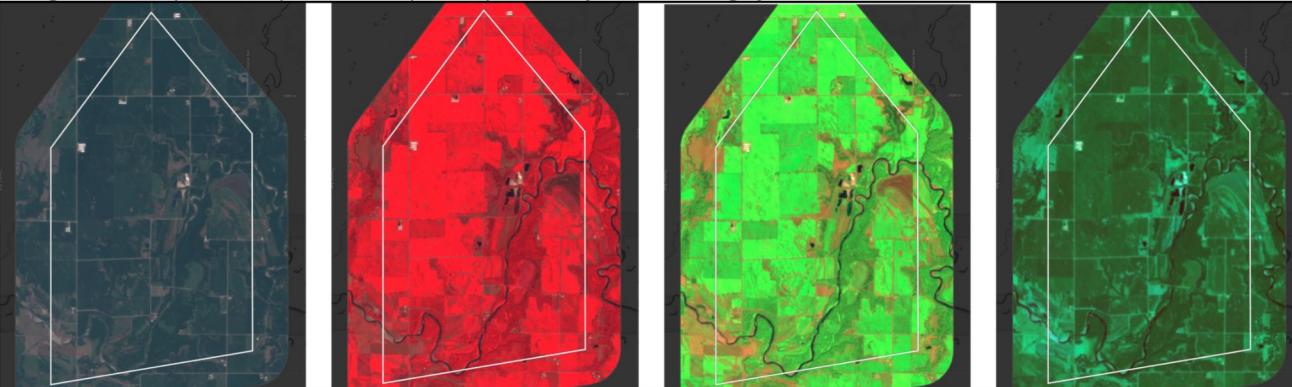


Satellite remote sensing data: a primer

Image source:

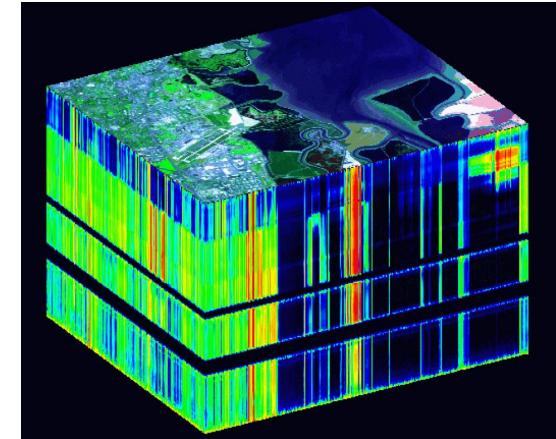
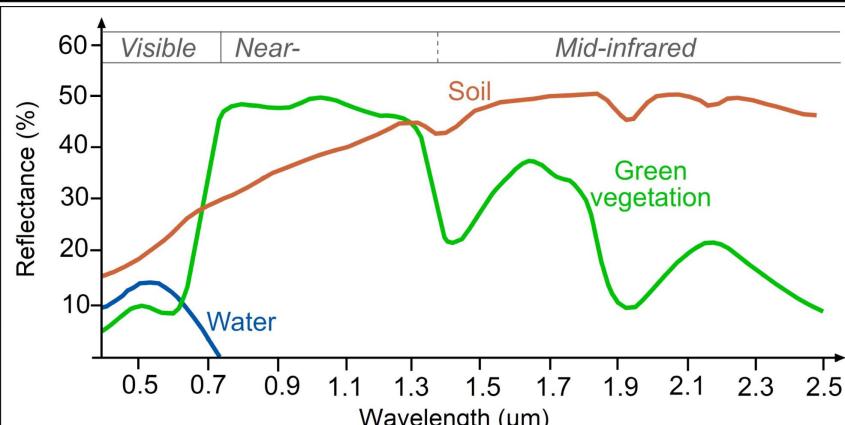
<https://www.earthdata.nasa.gov/learn/earth-observation-data-basics/remote-sensing>

Image source: <https://www.upstream.tech/posts/a-primer-on-false-color-imagery>



Truecolor; NIR, Red, Green; SWIR, NIR, Green; Blue, SWIR. Copernicus Sentinel data 2022.

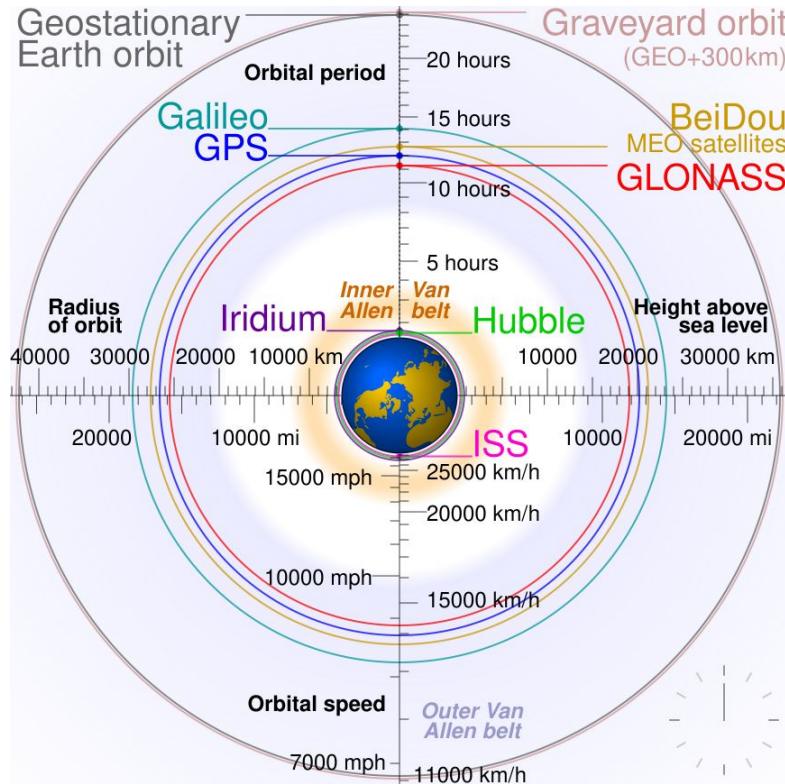
Image source:
<https://boisestatepressbooks.pub/chemistry/chapter/3-1-electromagnetic-energy/>



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

Types of satellites

Type & path of orbit



Sensing method

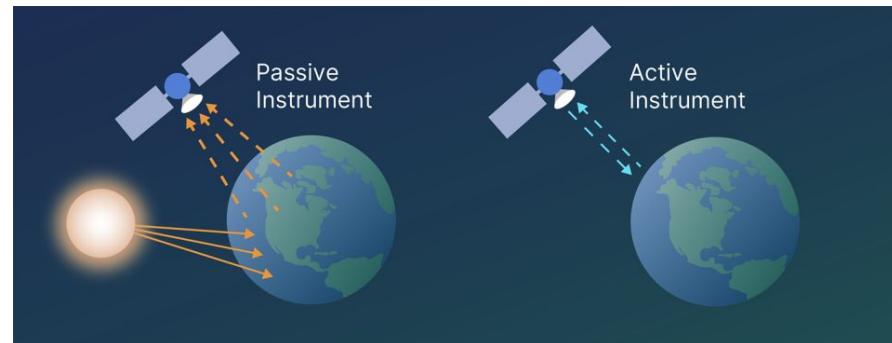
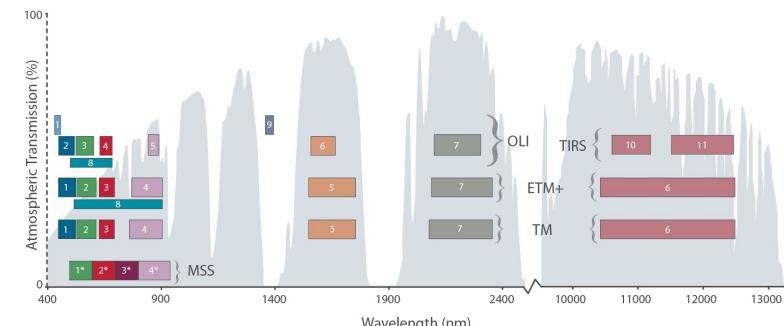
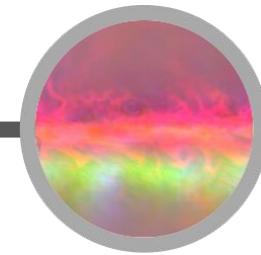
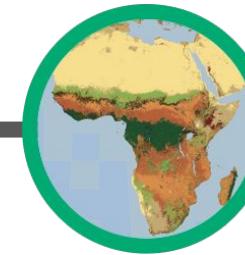


Image source: <https://www.earthdata.nasa.gov/learn/earth-observation-data-basics/remote-sensing>

Spectral bands



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, ...

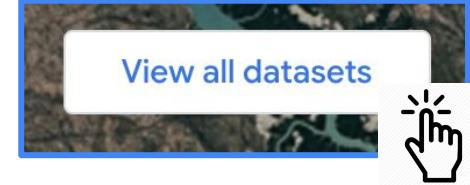
... and upload your own vectors and rasters

> 200 public datasets

> 4000 new images every day

> 5 million images

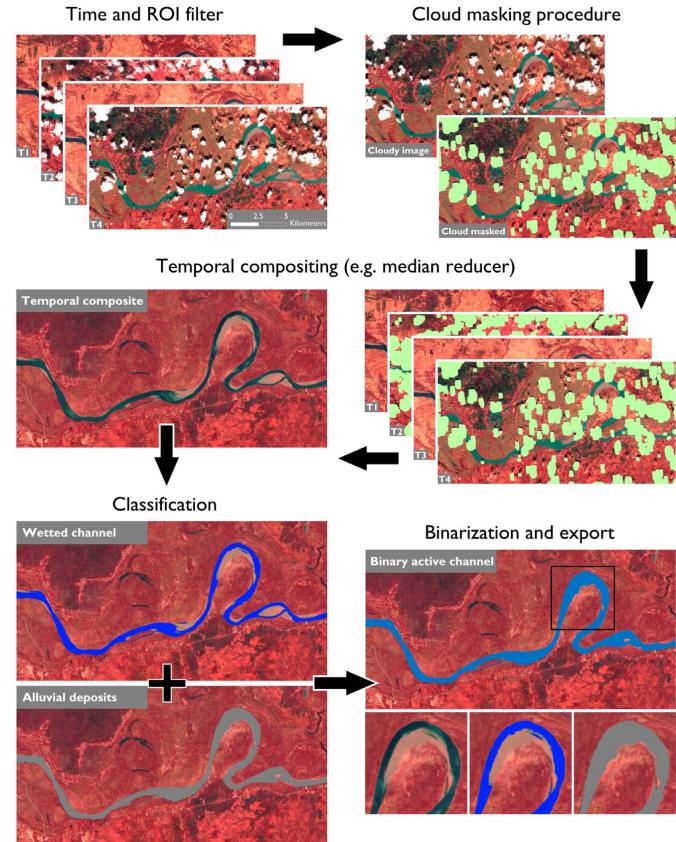
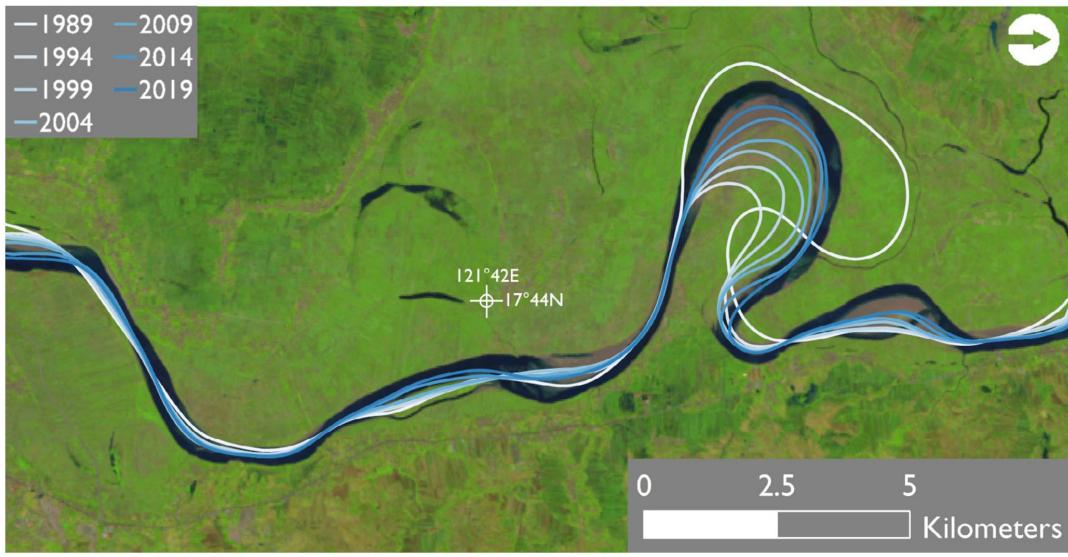
> 7 petabytes of data



Google Earth Engine: Hydrology and Water Resources Research Applications



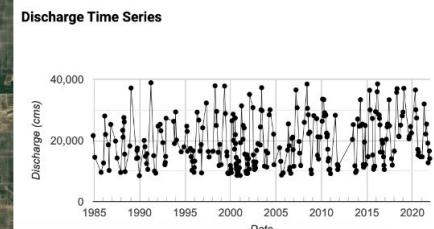
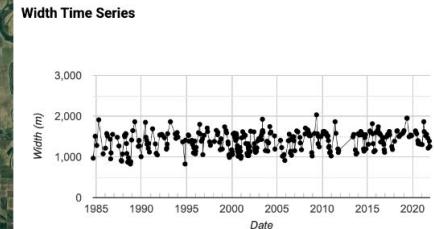
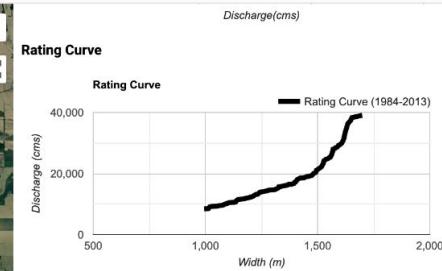
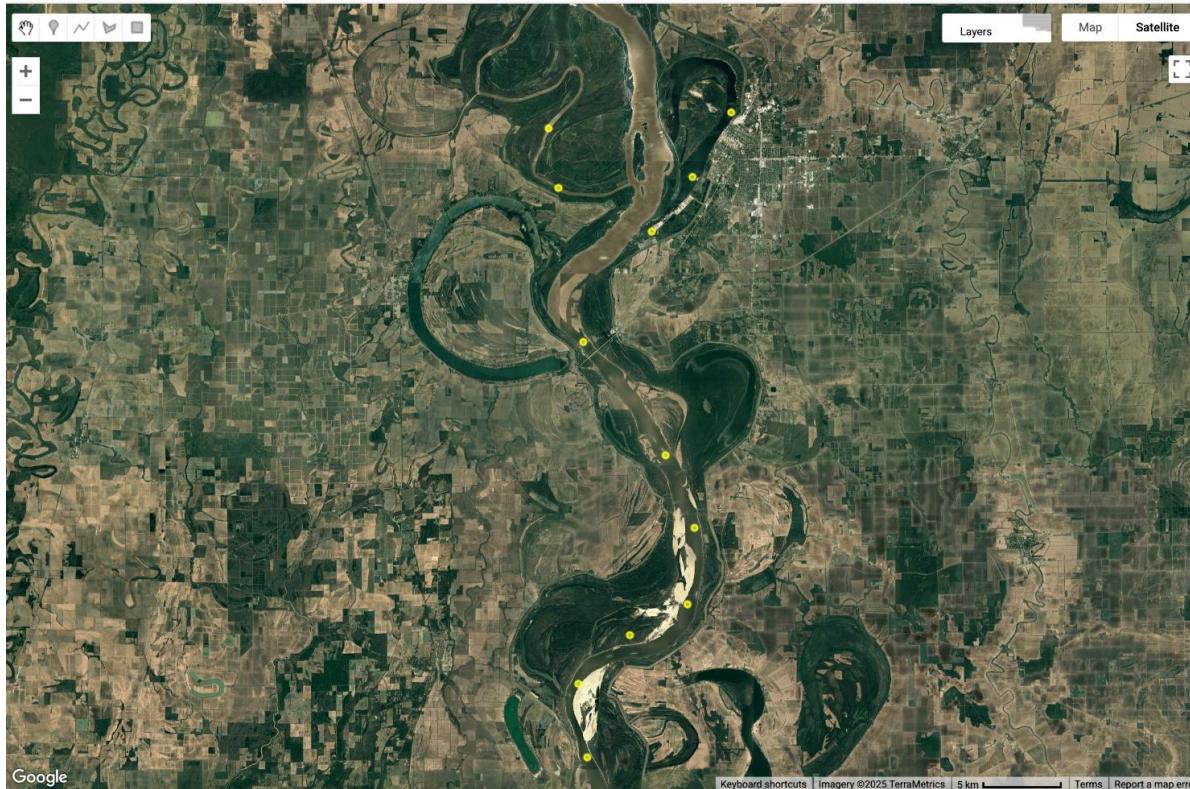
Detecting river channel change



River discharge estimation

Earth Engine Apps

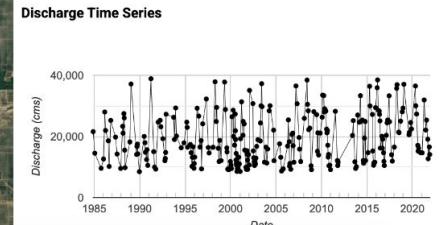
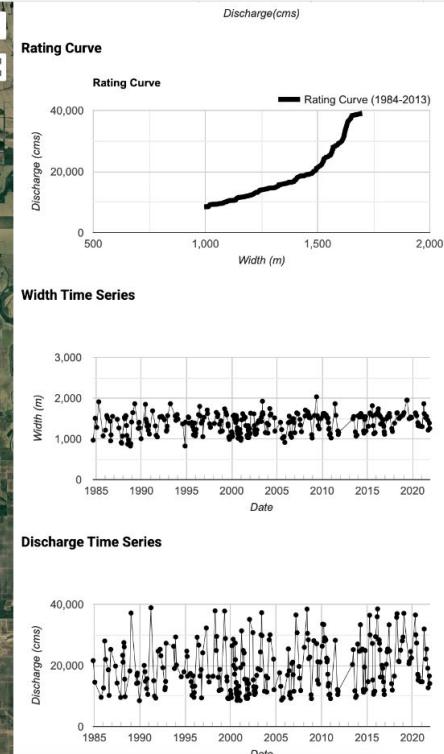
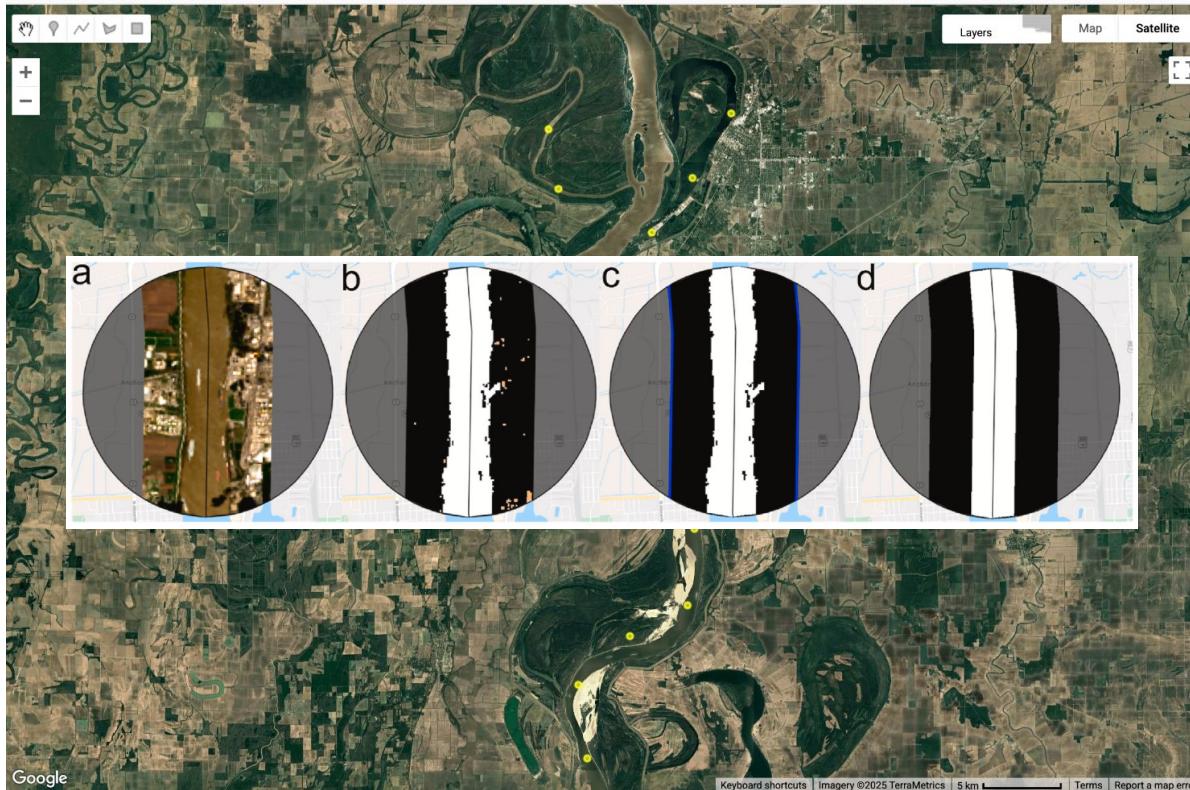
Search places



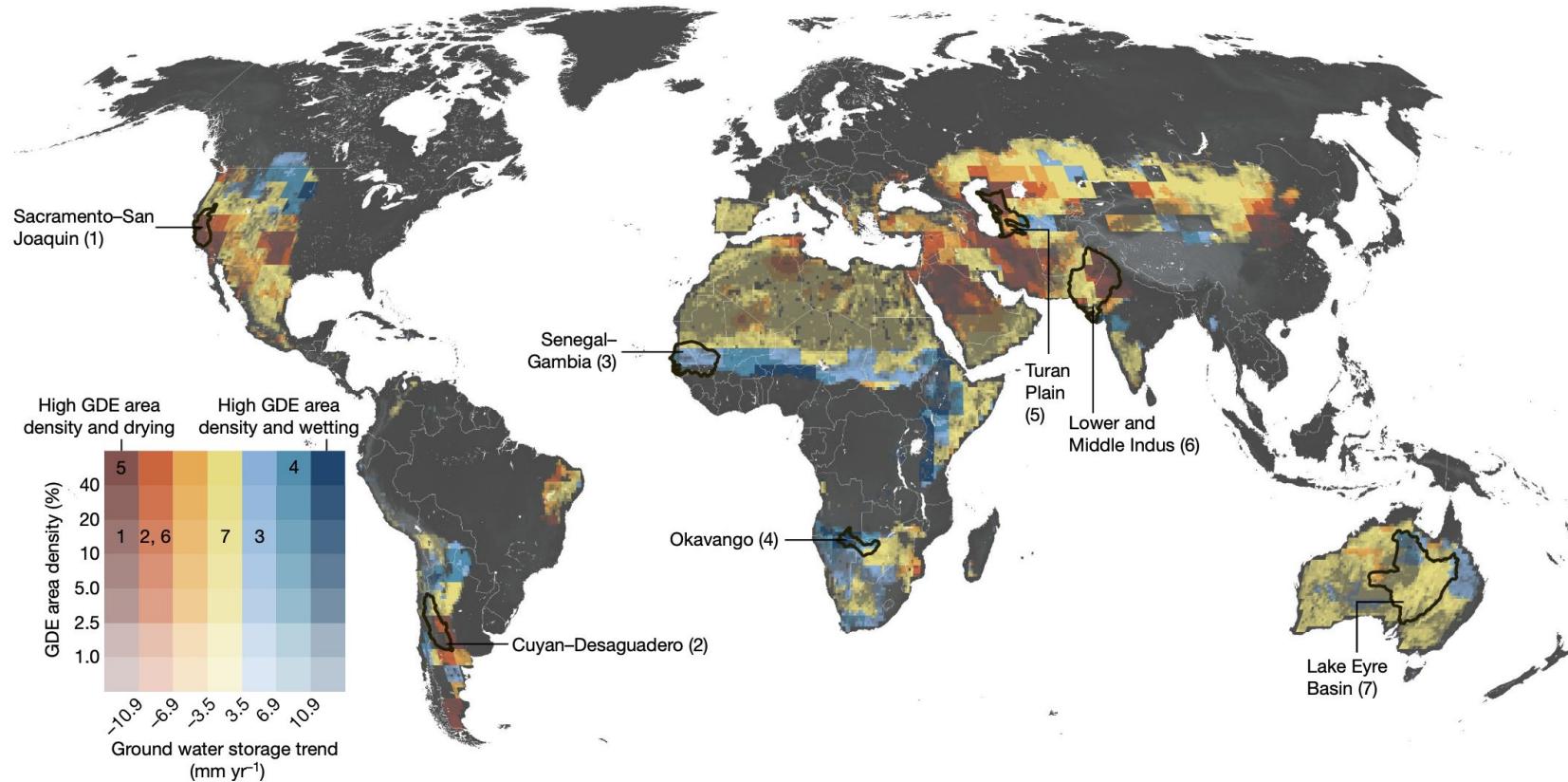
River discharge estimation

Earth Engine Apps

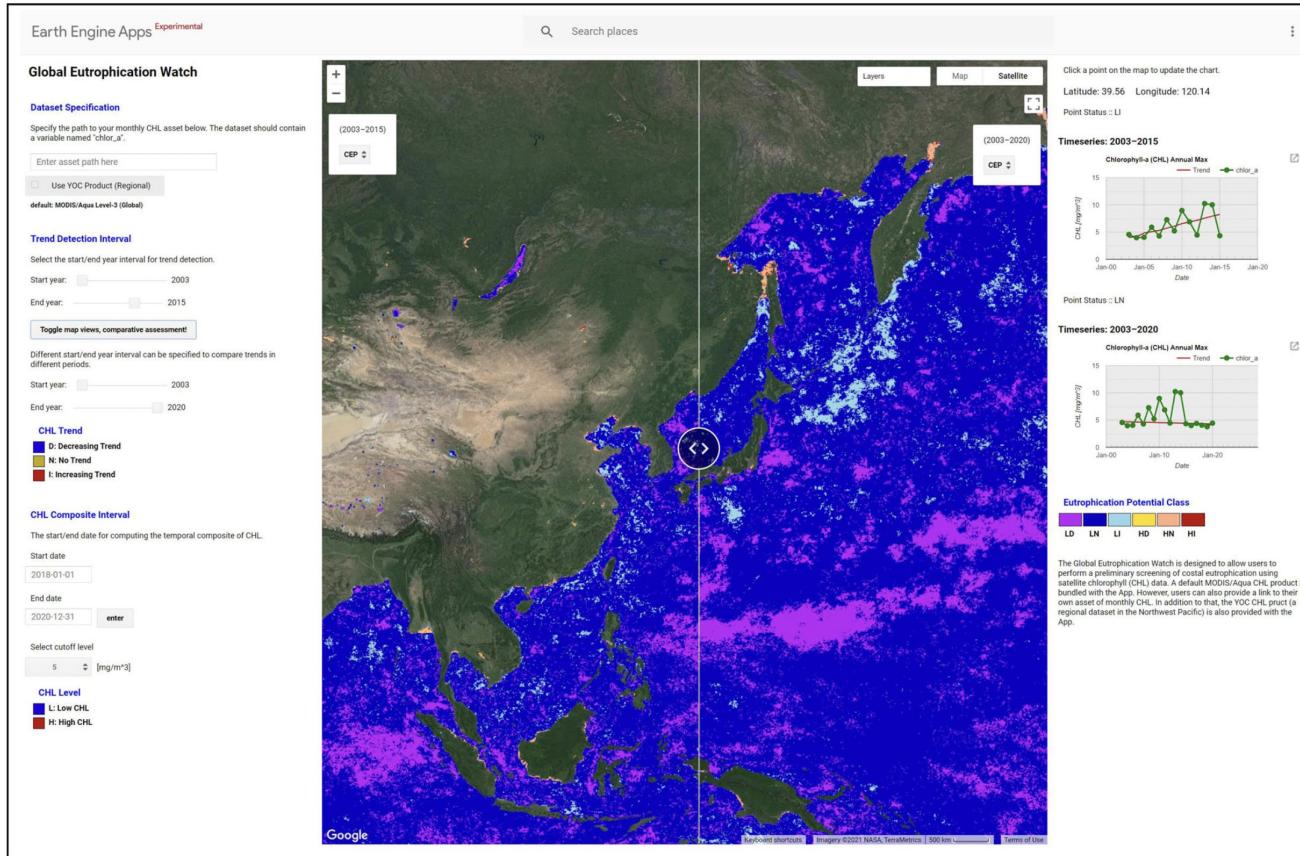
Search places



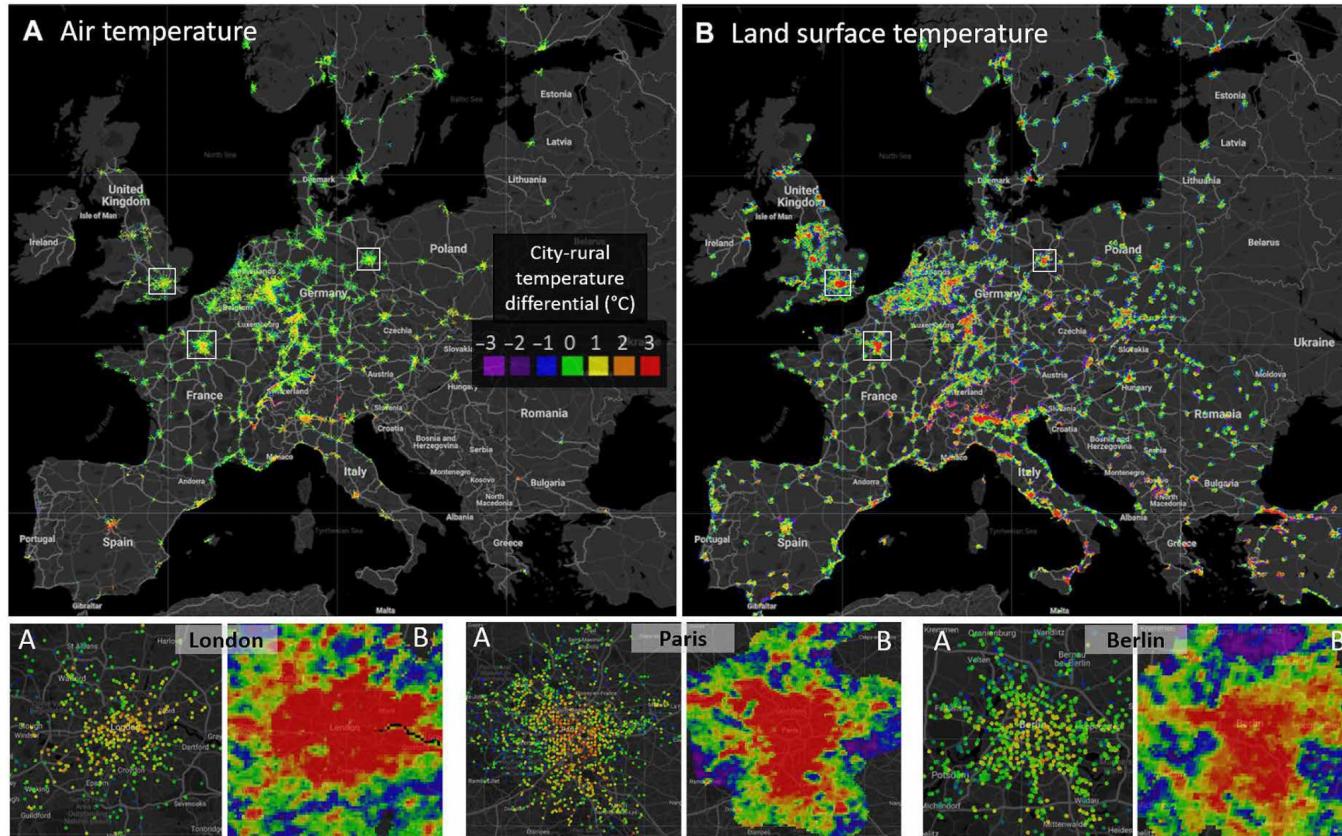
Groundwater storage trends



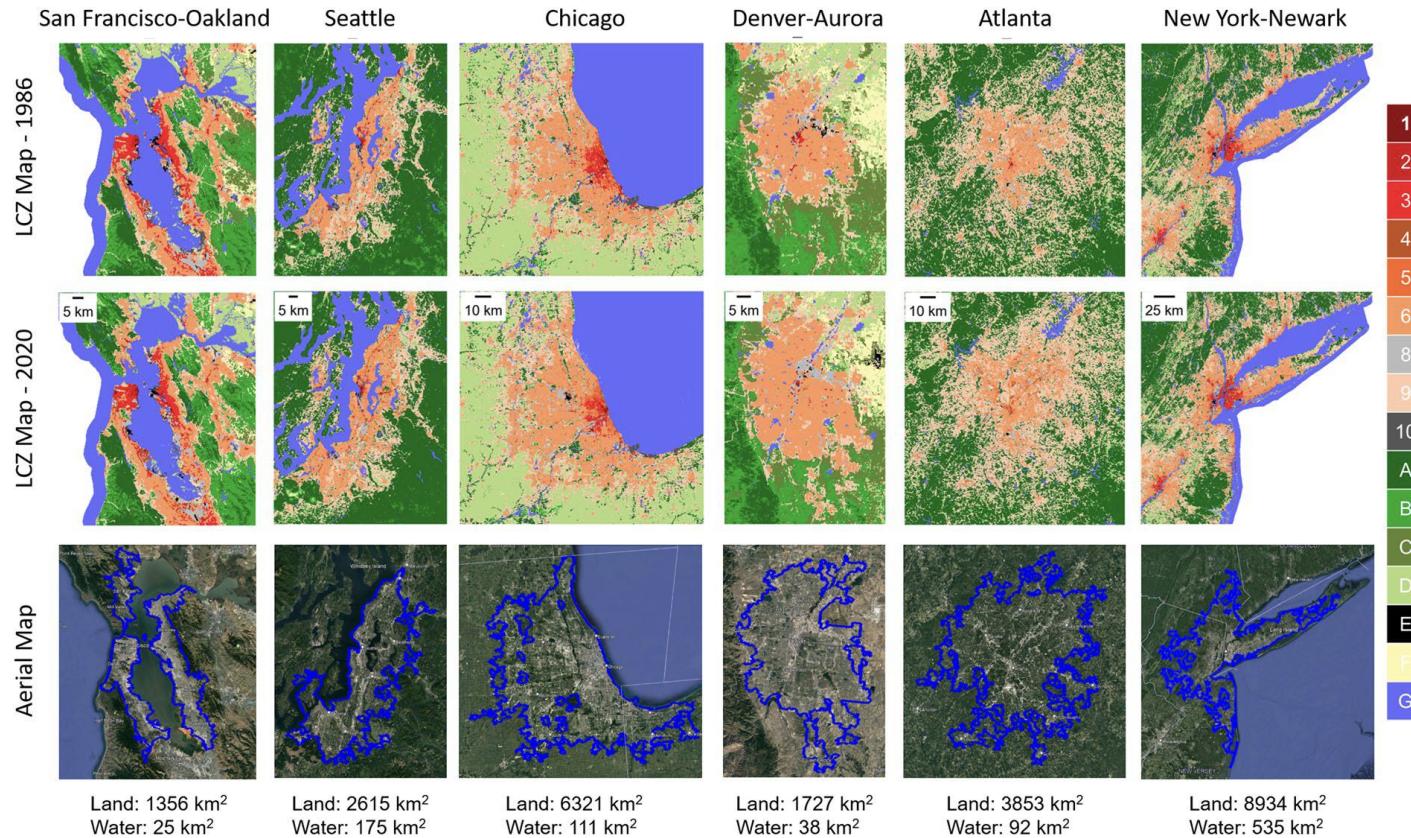
Tracking water quality parameters



Mapping urban heat islands



Mapping urban form into local climate zones



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

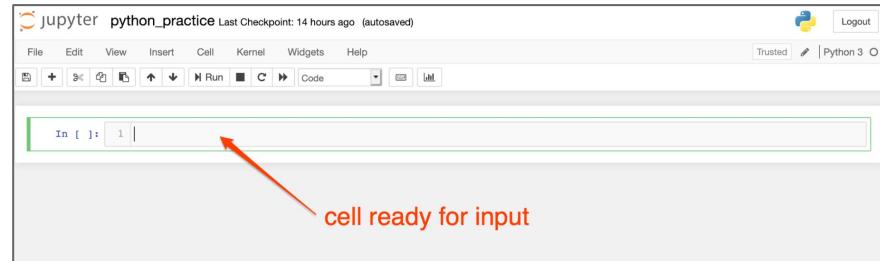
<https://console.cloud.google.com/earth-engine>

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 by 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

The screenshot shows a section of the 'Client vs. Server' page. It includes a navigation bar with Home, Products, Google Earth Engine, and Guides. A note at the top states: '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.' Below this is a note about the `getInfo()` method: '★ 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`, 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.' At the bottom, there's a note about distinguishing between Earth Engine objects and other Python or JavaScript objects.

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

Data Catalog

Visit <https://developers.google.com/earth-engine/datasets/> and click "View all datasets". You can browse or search available datasets.

Earth Engine Data Catalog

A planetary-scale platform for Earth science data & analysis

Earth Engine's public data archive includes more than forty years of historical imagery and scientific datasets, updated and expanded daily.

[View all datasets](#)

Earth Engine Data Catalog

Search

Home Categories All datasets All tags Landsat MODIS Sentinel Publisher Community API Doc

Earth Engine Data Catalog Filter list of datasets

Earth Engine's public data catalog includes a variety of standard Earth science raster datasets. You can import these with a single click. You can also upload your own [raster data](#) or vector data for private use or sharing in your scripts.

Looking for another dataset not in Earth Engine yet? Let us know by [suggesting a dataset](#).

2000 Greenland Mosaic - Greenland Ice Mapping Project (GIMP)

AG100: ASTER Global Emissivity Dataset 100-meter V003

ahn Netherlands 0.5m DEM, Interpolated

2000 Greenland Mosaic - Greenland Ice Mapping Project (GIMP)

AG100: ASTER Global Emissivity Dataset 100-meter V003

ahn Netherlands 0.5m DEM, Interpolated

Jupyter Notebook for the Tutorial

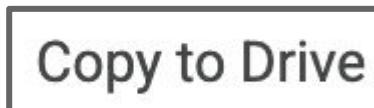
Please open the
Jupyter notebook

[here](#)



And click  at the top

Then





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")
```

NASA SRTM Digital Elevation 30m



Dataset Availability

2000-02-11T00:00:00Z–2000-02-22T00:00:00Z

Dataset Provider

NASA / USGS / JPL-Caltech

Earth Engine Snippet

```
ee.Image("USGS/SRTMGL1_003")
```

Description

Bands

Terms of Use

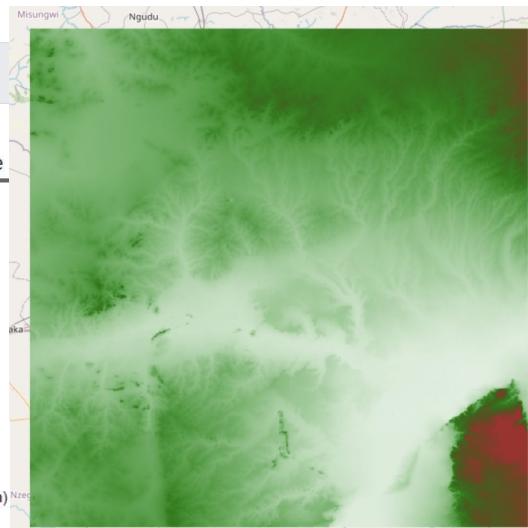
Citations

Bands

Name

elevation

* estimated min or max value



SRTM Elevation

Low (1000m)
Medium (~1300m)
High (1600m+)

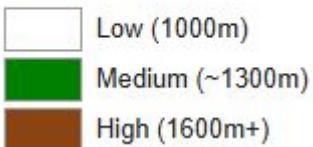
How to add Legends

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",
)
```

SRTM Elevation



2. Built-in Legend

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

)
```

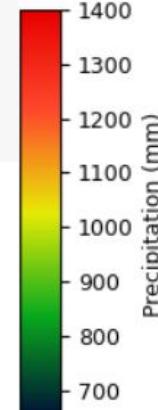
#Display the Map
Map2



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,
```



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

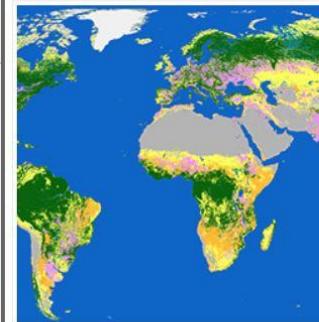
Coverage: Global

Spatial Resolution: 10 meters

Earth Engine Snippet:

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

ESA WorldCover 10m v200



Dataset Availability

2021-01-01T00:00:00Z–2022-01-01T00:00:00Z

Dataset Provider

[ESA WorldCover Consortium](#)

Earth Engine Snippet

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

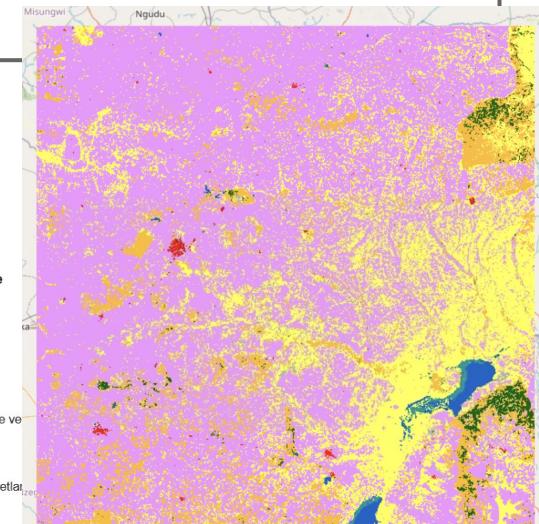
Description

Bands

Terms of Use

Citations

Pixel Size
10 meters



Precipitation

PERSIANN-CDR is a daily quasi-global precipitation product that spans the period from 1983-01-01 to present. The data is produced quarterly, with a typical lag of three months.

Dataset: PERSIANN-CDR

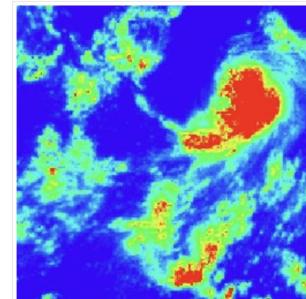
Coverage: Global 60°S to 60°N, 1983-

Spatial Resolution: 25 kilometers

Earth Engine Snippet:

```
ee.ImageCollection("NOAA/PERSIANN-CDR")
```

PERSIANN-CDR: Precipitation Estimation From Rain Networks-Climate Data Record



Dataset Availability

1983-01-01T00:00:00Z-2024-09-30T00:00:00Z

Dataset Provider

NOAA NCDC

Earth Engine Snippet

```
ee.ImageCollection("NOAA/PERSIANN-CDR")
```

Description

Bands

Terms of Use

Citations

DOIs

Pixel Size

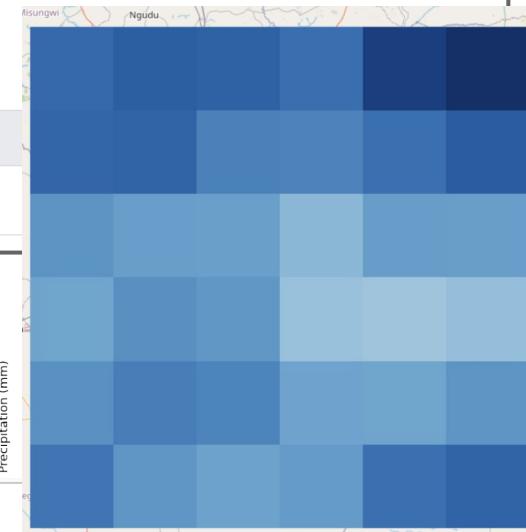
27830 meters

Bands

Name

precipitation

* estimated min or max value



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.

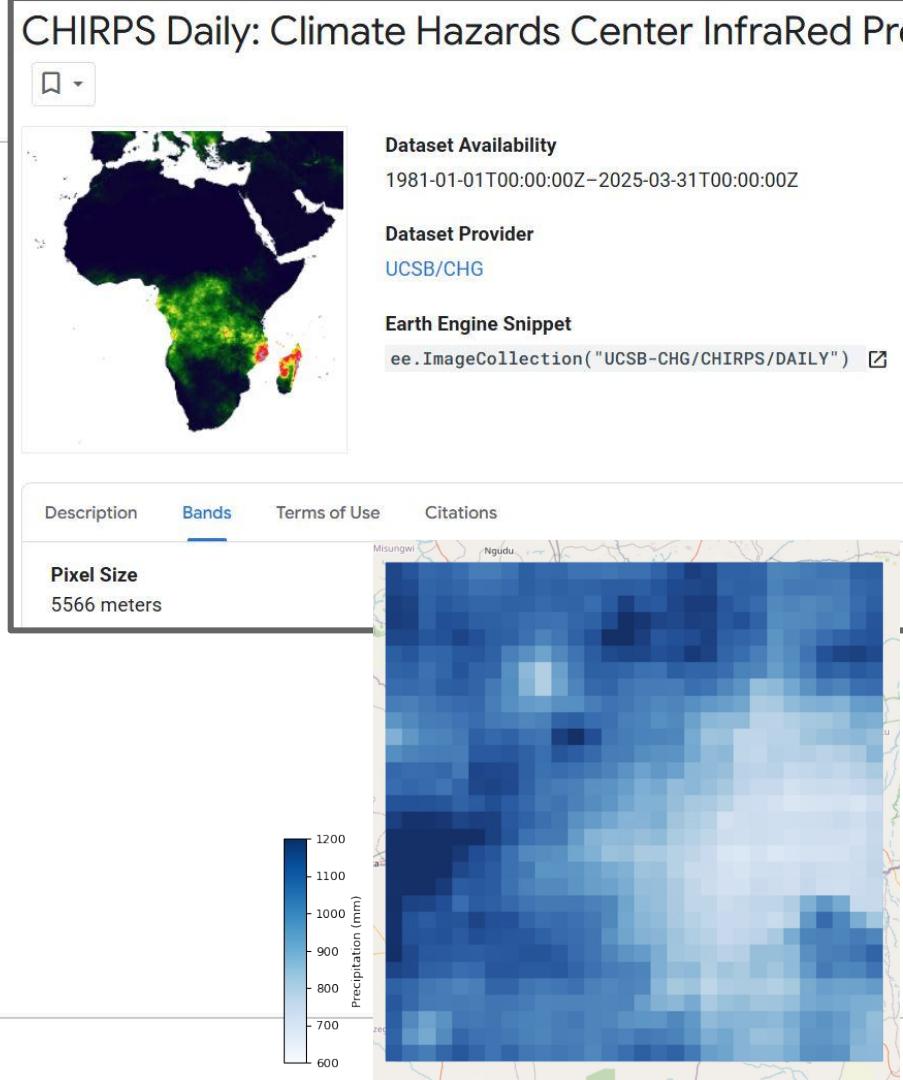
Dataset: CHIRPS Daily

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

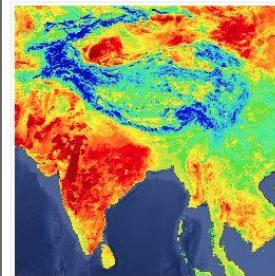
Coverage: Global, 2000-

Spatial Resolution: 1 kilometer

Earth Engine Snippet:

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

MOD11A1.061 Terra Land Surface Temperature



Dataset Availability

2000-02-24T00:00:00Z–2025-05-27T00:00:00Z

Dataset Provider

NASA LP DAAC at the USGS EROS Center

Earth Engine Snippet

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



Description

Bands

Terms of Use

Citations

DOIs

Pixel Size

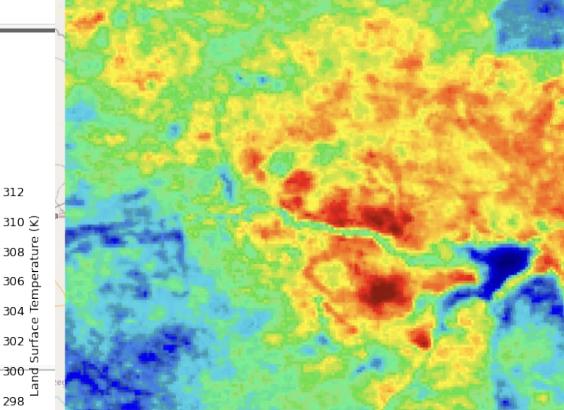
1000 meters

Bands

Name

LST_Day_1km

QC_Day



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

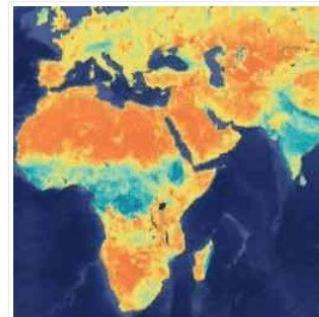
Coverage: Global, 2015-2023

Spatial Resolution: 9 kilometers

Earth Engine Snippet:

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

SPL3SMP_E.005 SMAP L3 Radiometer Global Daily



Dataset Availability

2015-03-31T12:00:00Z–2023-12-03T12:00:00Z

Dataset Provider

Google and NSIDC

Earth Engine Snippet

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

Description

Bands

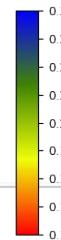
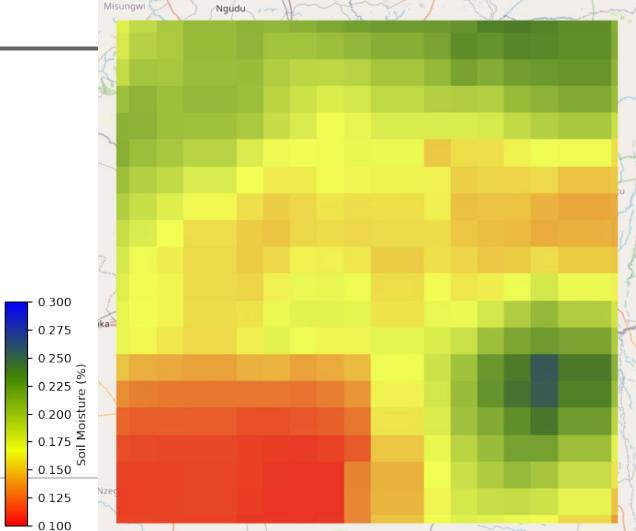
Terms of Use

Citations

DOIs

Pixel Size

9000 meters

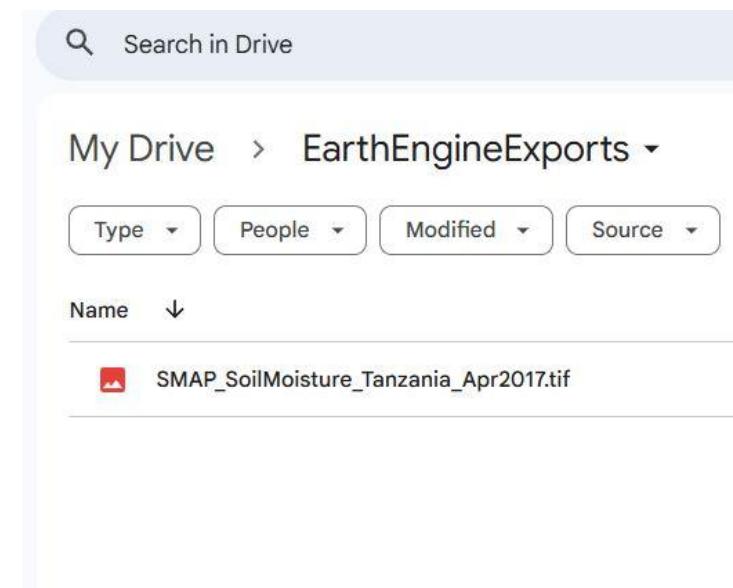


Downloading Datasets to Google Drive

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

```
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()
```



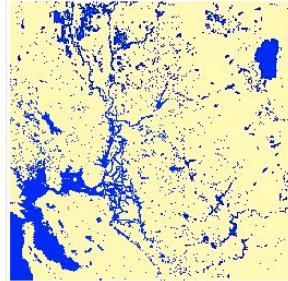
Tutorial 2

Surface Water Change Over Time

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							
	<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td>water</td><td>Water detection for the month.</td></tr></tbody></table>	Name	Description	water	Water detection for the month.			
Name	Description							
water	Water detection for the month.							
	<p>■ Bitmask for water</p> <ul style="list-style-type: none">• Bits 0-1: Water detection<ul style="list-style-type: none">• 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



Image source: <https://landsat.gsfc.nasa.gov/satellites/landsat-9/>

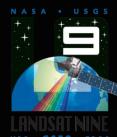


Image source:
<https://landsat.gsfc.nasa.gov/apps/YourNameInLandsat-main/index.html>

Jupyter Notebook for the Tutorial

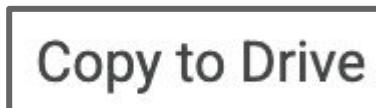
Please open the
Jupyter notebook

here



And click  at the top

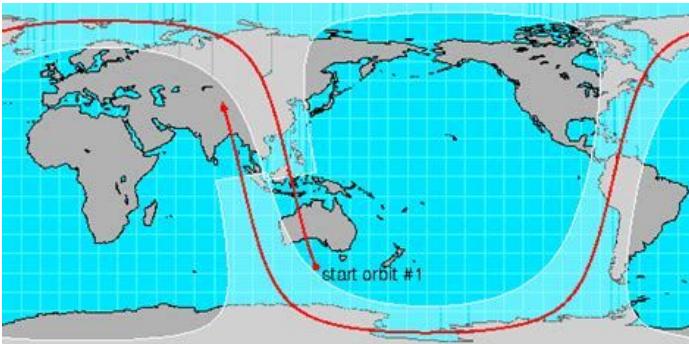
Then



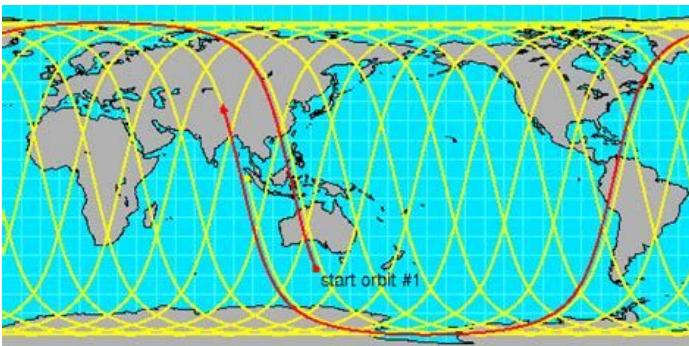


What makes satellites prone to errors and missing data?

- ❖ 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



Images source:
https://apollo.nvu.vsc.edu/classes/remote/lecture_notes/satellite/platforms/

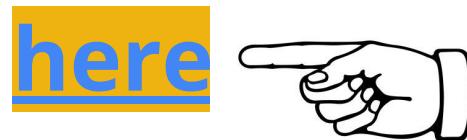
Tutorial 3

Storage changes at the basin scale



Jupyter Notebook for the Tutorial

Please open the
Jupyter notebook



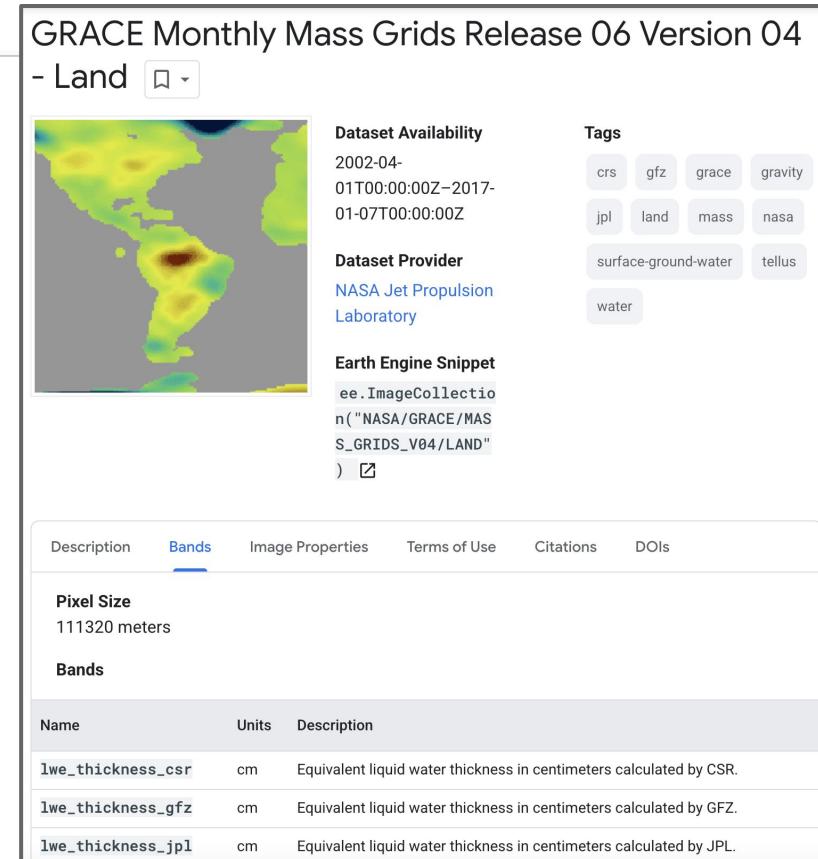
And click  Open in Colab at the top

Then



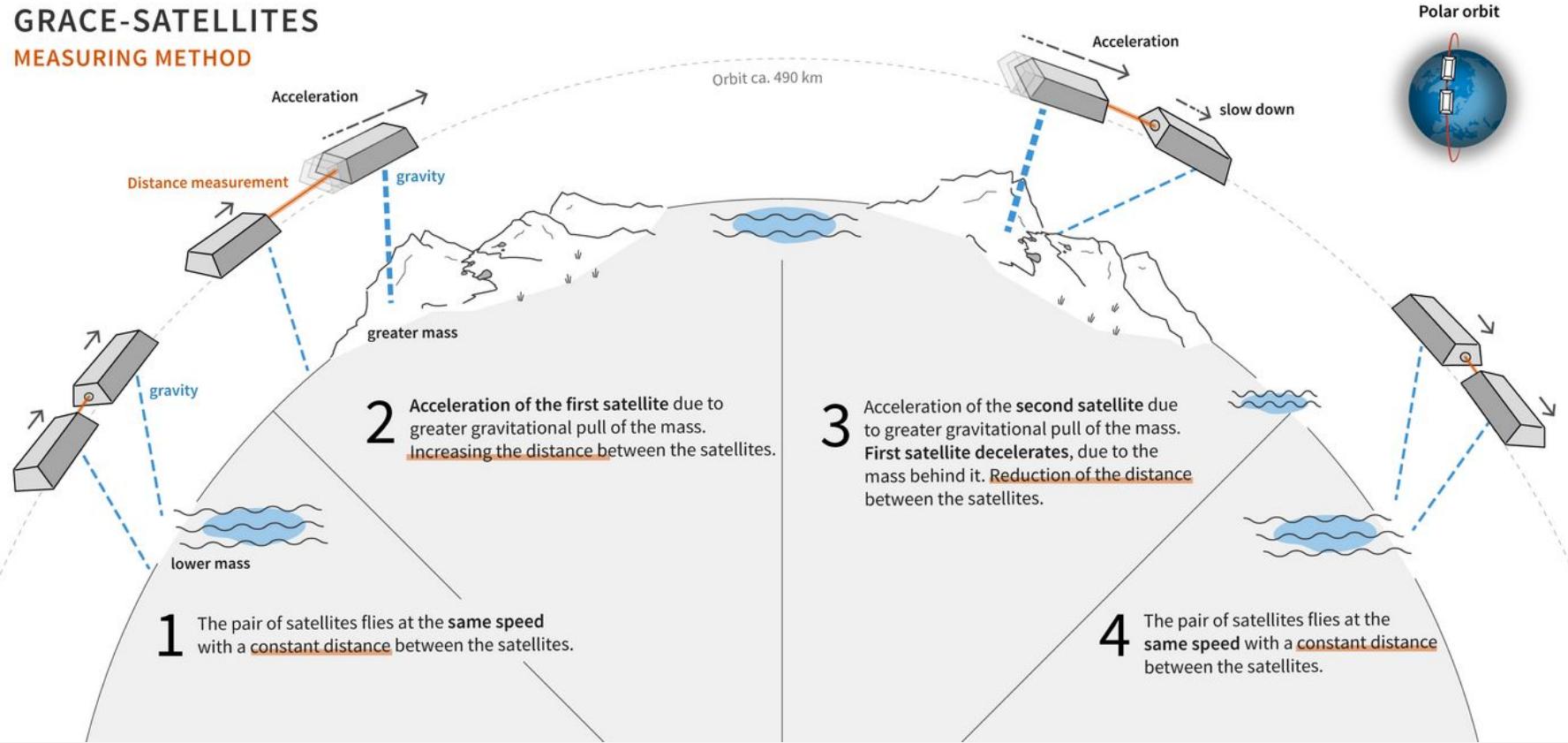
TWSA dataset

- ❖ Sum of all above and below surface water storages
 - Canopy water
 - Rivers and lakes
 - Soil moisture
 - Groundwater
- ❖ Derived from GRACE & GRACE-FO
- ❖ **Temporal coverage:** Monthly, 2002-2017
- ❖ **Spatial coverage:** 100 km, Global



GRACE

GRACE-SATELLITES MEASURING METHOD



Basin boundaries

- ❖ Watersheds range from level 1 (coarse) to level 12 (detailed), using Pfastetter codes
- ❖ HydroBASINS is a component of HydroSHEDS

WWF HydroATLAS Basins Level 03



Dataset Availability
2000-02-22T00:00:00Z–2000-02-22T00:00:00Z

Dataset Provider
[WWF](#)

Earth Engine Snippet
`ee.FeatureCollection("WWF/HydroATLAS/v1/Basins/level03")`

WWF HydroATLAS Basins Level 04



Dataset Availability
2000-02-22T00:00:00Z–2000-02-22T00:00:00Z

Dataset Provider
[WWF](#)

Earth Engine Snippet
`ee.FeatureCollection("WWF/HydroATLAS/v1/Basins/level04")`

Thank you!

- ❖ For additional resources on learning GEE and Python, please see
uci-chrs.github.io/GEE-Training-2025/additional-resources
- ❖ For further questions, feel free to contact Annika Hjelmstad at



ahjelmst@uci.edu

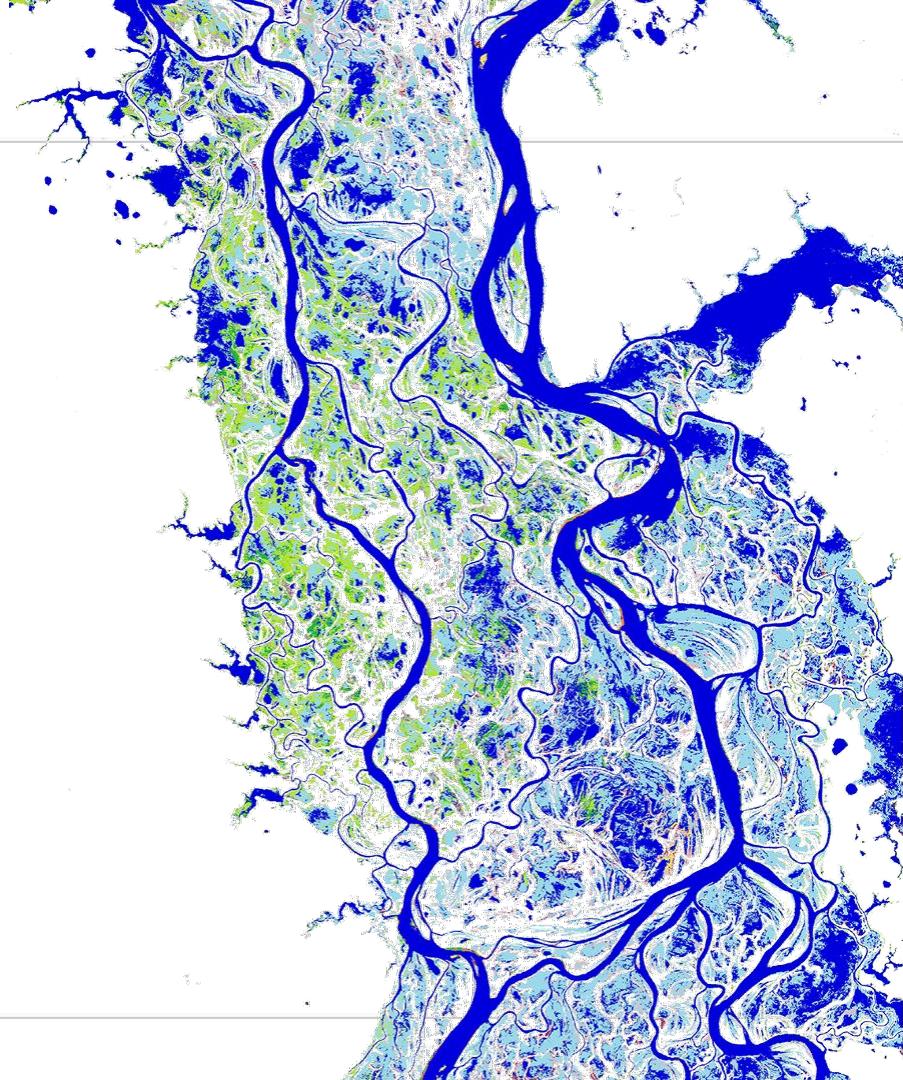


Image source: <https://global-surface-water.appspot.com/>