Introduction to Google Earth Engine in Javascript

A Hands-on Workshop to get started with GEE

Ujaval Gandhi
ujaval@spatialthoughts.com
Spatial Thoughts

This presentation → https://bit.ly/intro-to-gee-js

Introduction

- Background in GIS and Remote Sensing
  - Intern at Indian Institute of Remote Sensing (IIRS), Dehradun, India
  - MS in Geospatial Information Engineering from University of Wisconsin - Madison, USA
- 15 years of professional experience
  - One of the early employees at Google Inc.
    - Moved to India in 2006 and established the maps team
    - Led the GIS and Aerial Imagery team in India from 2007-2019
  - Developed expertise in Google Earth Engine and trained 1000+ scientists and researchers across India
  - Left Google in March 2020.
- Presently
  - Focused full-time on teaching and building content at Spatial Thoughts
  - Active Earth Engine developer and trainer
  - Co-author of the open-access book *Earth Engine Fundamentals and Applications* (eefabook.org)

Ujaval Gandhi

TAs

- Emil Cherrington
- Samapriya Roy
- Ziu Lin

What is Google Earth Engine?

Google's cloud platform for easy petabyte-scale analysis

Data

An exhaustive catalog of remote sensing datasets, including multispectral, radar, aerial, climate, land cover, and vector.

Computation

Colocated data storage + computation

API

Simple, yet powerful JavaScript and Python API

Browser-based IDE

No software to download or keep up to date. All you need is a modest internet connection.

What can you do in Earth Engine?

- **Image Processing** (Map Algebra, Kernels and Convolutions, Spectral Unmixing, Pan-sharpening, Gap Filling, Data Fusion)
- **Vector Processing** (Zonal Statistics, Spatial Joins, Spatial Query etc.)
- **Terrain Processing** (Slope, Aspect, Hillsahde, Hill Shadow Analysis)
- **Time Series Analysis** (Extract Time-Series, Trend Analysis, Time-Series Smoothing, Temporal Segmentation etc.)
- **Object-based Image Analysis** (GLCM, Texture, Hotspots etc.)
- **Change Detection** (Spectral Distance, Change Classification, Class Transitions)
- **Machine Learning** (Supervised and Unsupervised Classification, Linear Regression, Principal Components Analysis etc.)
- **Deep Learning** (DNN, Object Detection etc. via Tensorflow)

What you CANNOT do in Earth Engine?

- Create Cartographic Outputs
- 3D visualization and analysis
- Run Hydrological models (i.e. Rainfall-runoff modeling) and analysis (i.e. watershed delineation, fill depression etc.)
- Photogrammetry (i.e. Orthorectification, Point-Clouds etc.)
- LIDAR processing
- SAR Interferometry

Workshop Overview

Part 1 (3 Hours)	Part 2 (2 Hours)
Data Discovery, Processing and Export	**Computation in Earth Engine**
Javascript Basics ImageCollections Filters Creating Composites FeatureCollections Export	Map/Reduce Programming Concepts Calculating Indices Computation on ImageCollections Time-series Charts

Workshop Structure

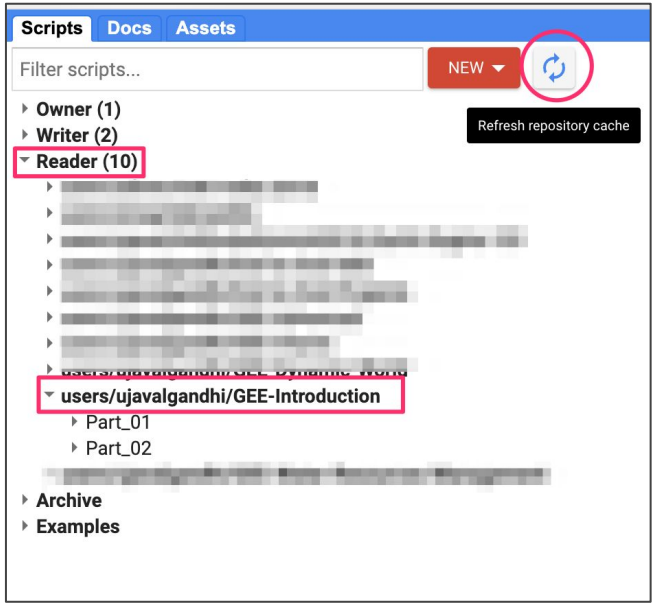
- We will have 10-15 minutes of explanation of a concept, followed by a 5-minute exercise
  - All participants are expected to complete the exercises during the class.
- You can ask questions at any point during the session.
- We will have lunch break from 12:30pm-1:30pm

Get the Workshop Materials

This presentation → https://bit.ly/intro-to-gee-js

You will need a Google Earth Engine account for this workshop. Sign-up if you do not have one.

- Get the workshop repository → Click this link to open Google Earth Engine code editor and add the repository to your account.
- If successful, you will have a new repository named **users/ujavalgandhi/GEE-Introduction** in the Scripts tab in the Reader section.
- If you do not see the repository in the Reader section, click Refresh repository cache button in your Scripts tab.



Let’s get coding

What is your favorite programming language?

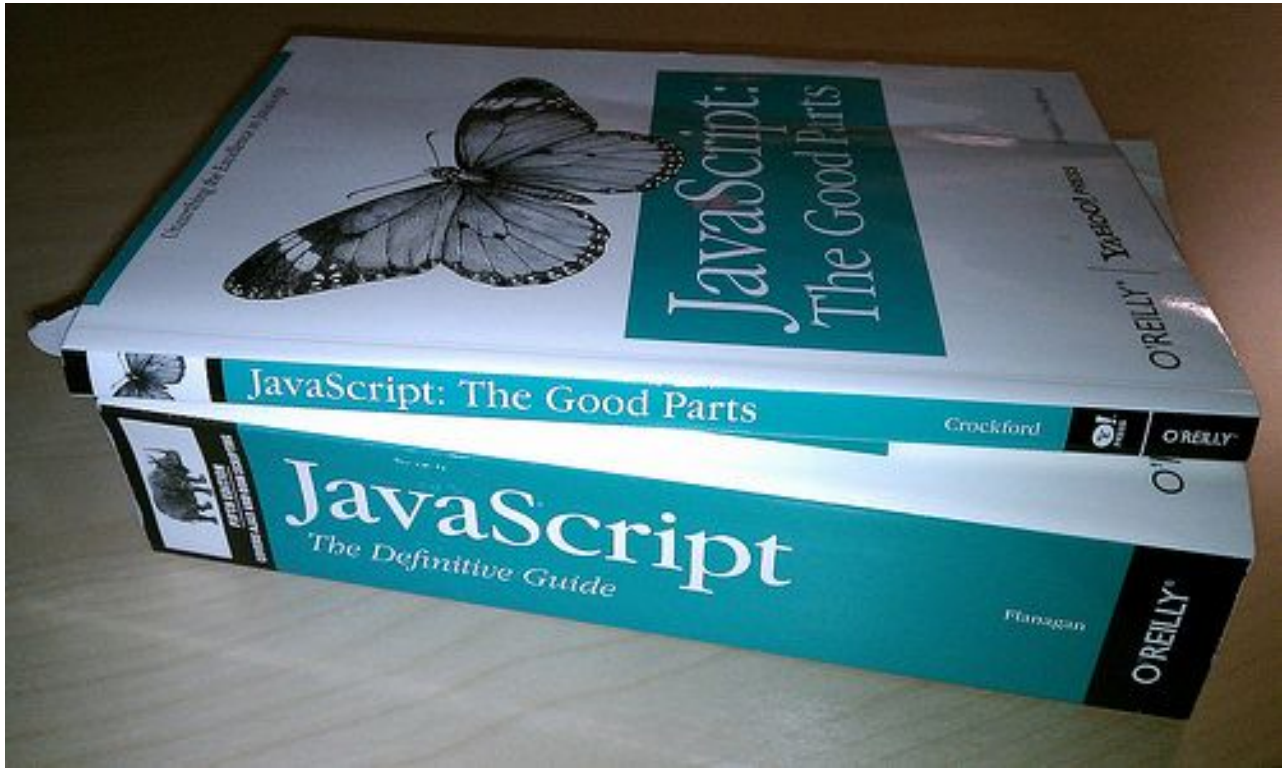


Image Source: Reddit

Javascript vs. Python

- You are learning the Earth Engine API
- The API is exactly the same regardless of the language you choose
  - With a few small caveats
- Javascript API is the most mature and easiest to get started.
  - No installation required
  - No need to worry about authentication
  - Very easy to share scripts, ask for help
  - Building and deploying apps is very easy
- Python API is much more powerful
  - Integrate with other data science libraries for data processing and plotting
  - Automate launching and managing Exports
- We will focus on Javascript for this workshop.
  - You can automatically convert any Javascript code to Python [tutorial]

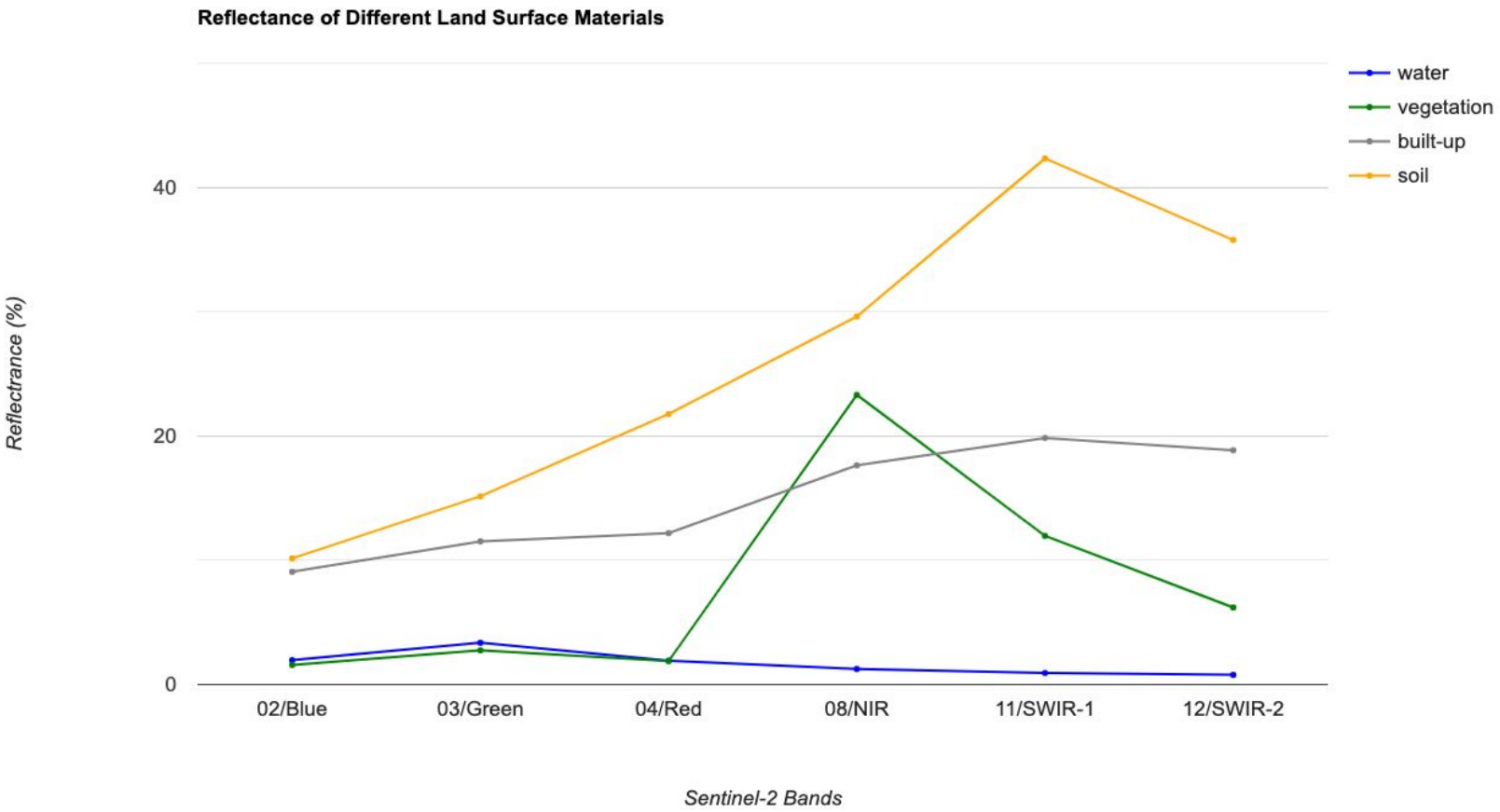
Part 1:
Data Discovery, Processing and Export

Quiz 1

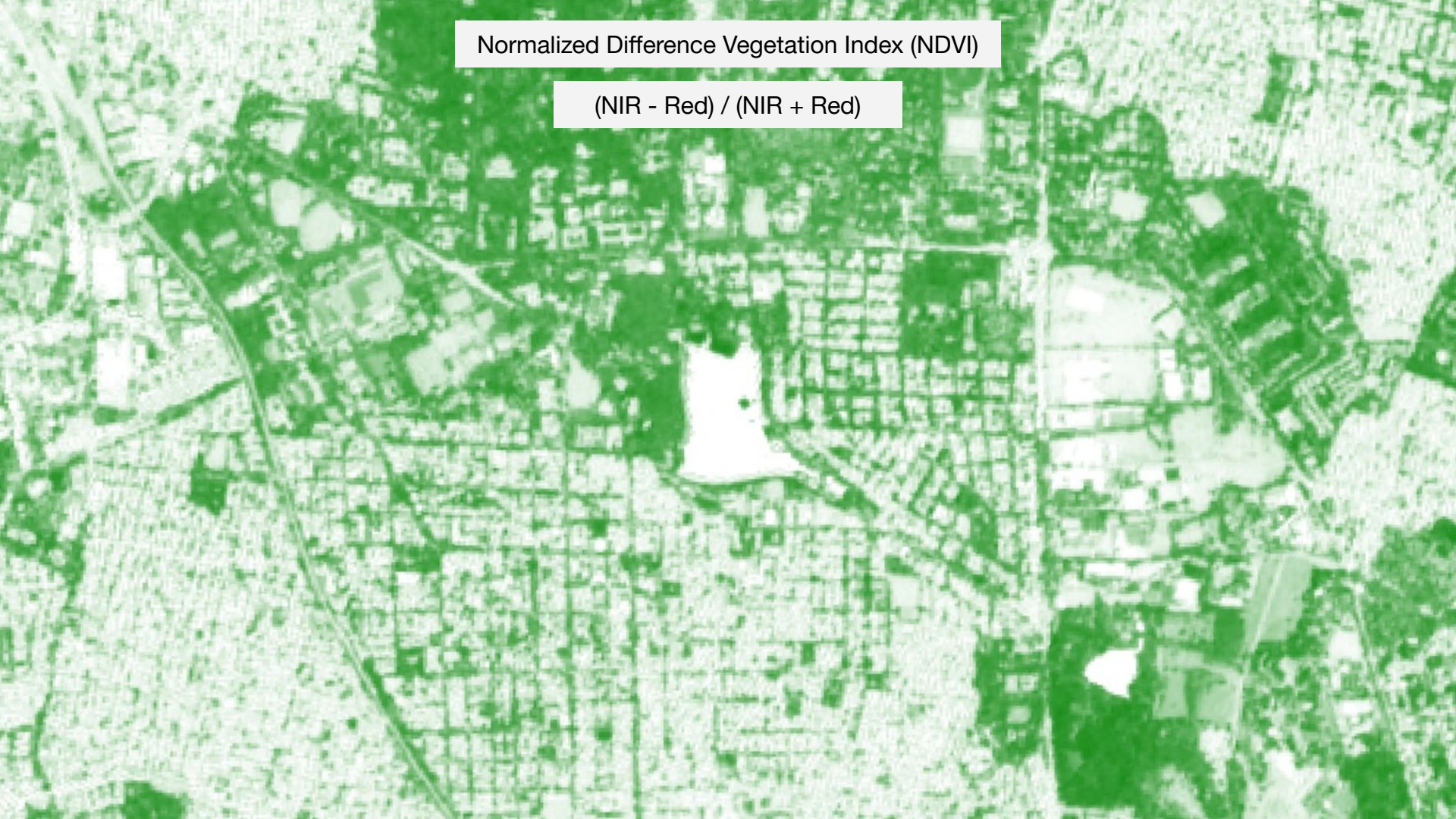
This presentation → https://bit.ly/intro-to-gee-js

Part 2:
Computation in Earth Engine

Spectral Indices and Band Ratios

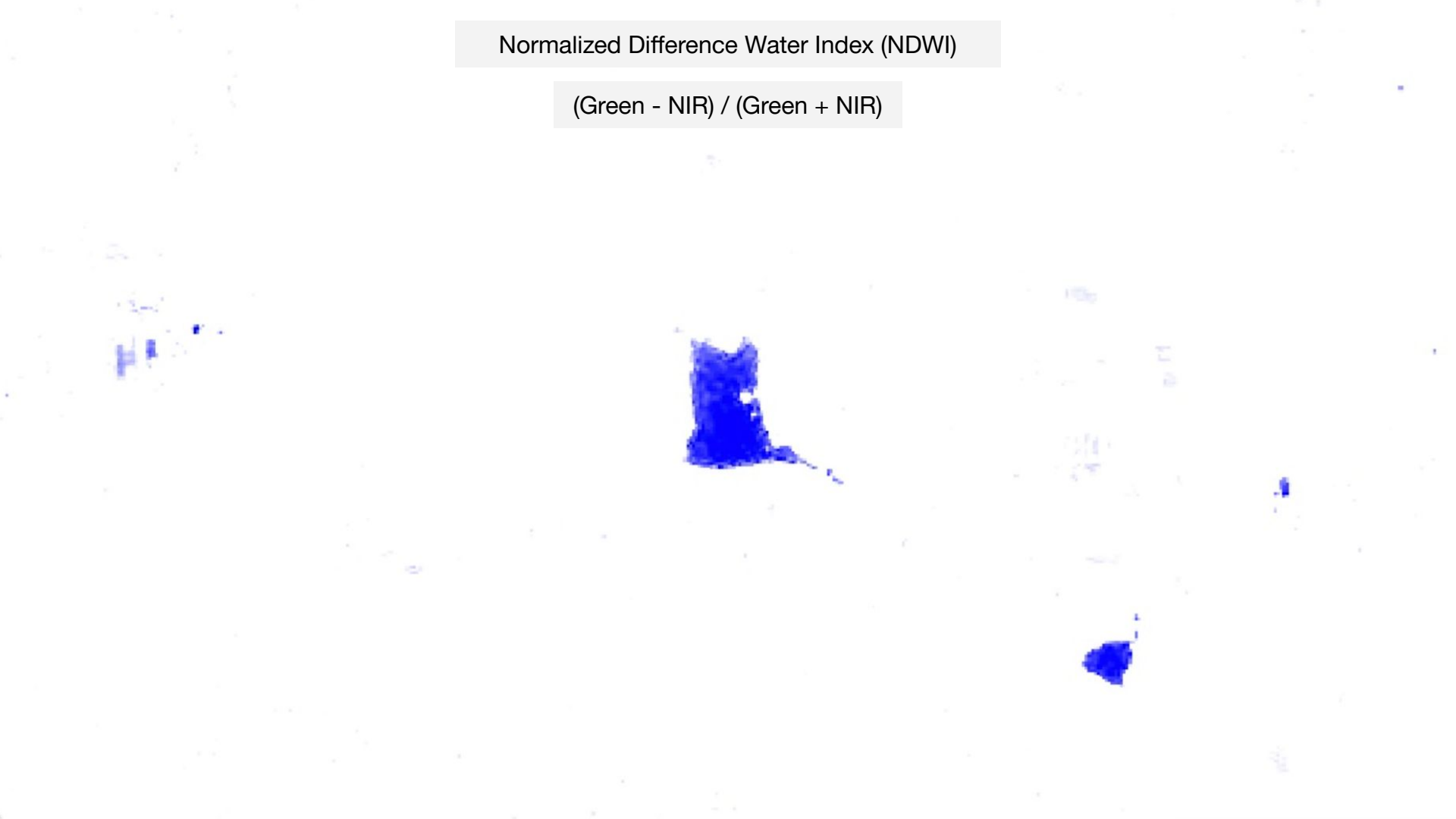

Reflectance of Different Land Surface Materials
water
vegetation
built-up
soil
40
20
0
Reflectrance (%)
02/Blue
03/Green
04/Red
08/NIR
11/SWIR-1
12/SWIR-2
Sentinel-2 Bands

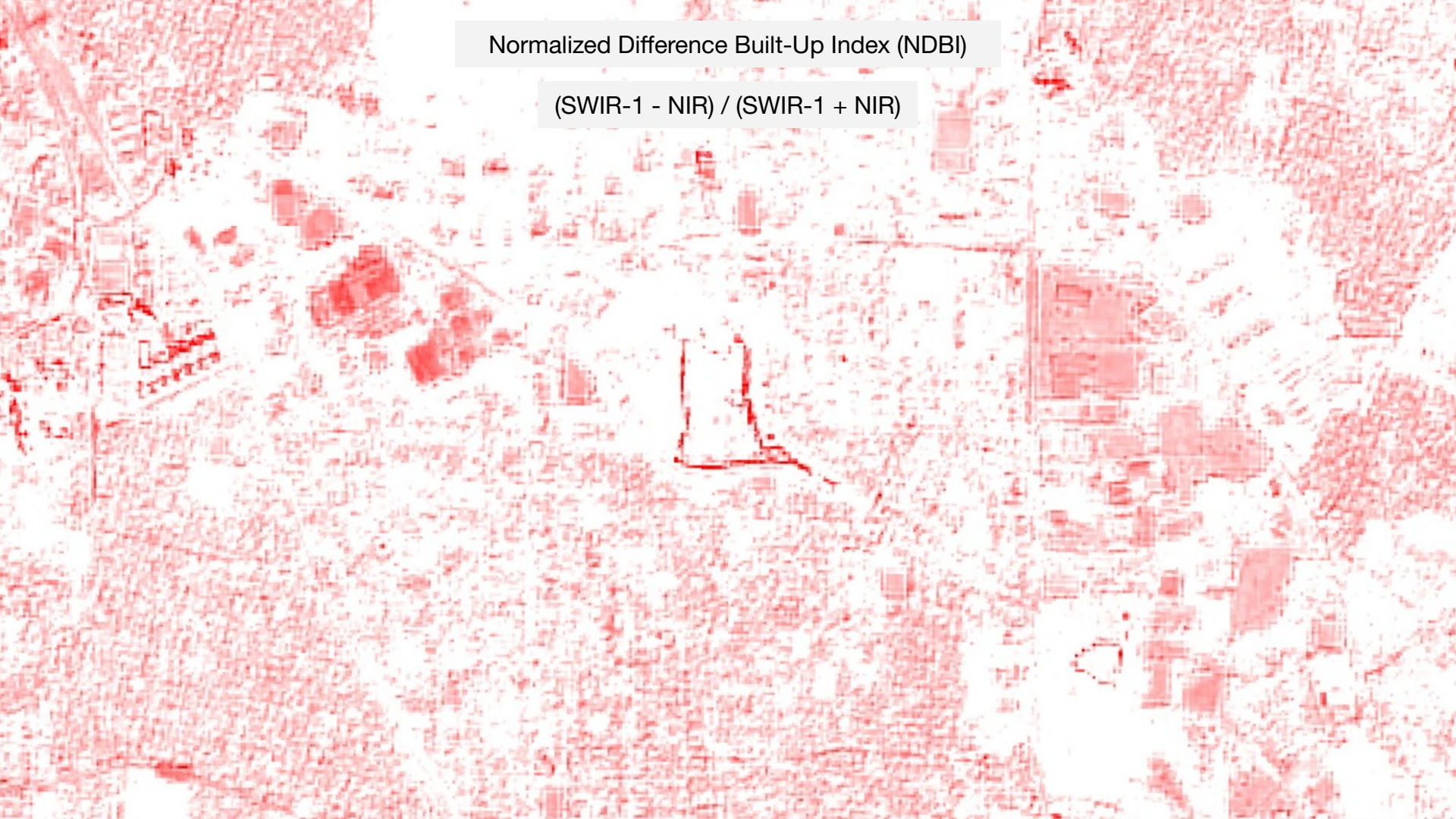

Sentinel-2 SR Image


Normalized Difference Vegetation Index (NDVI)
(NIR - Red) / (NIR + Red)

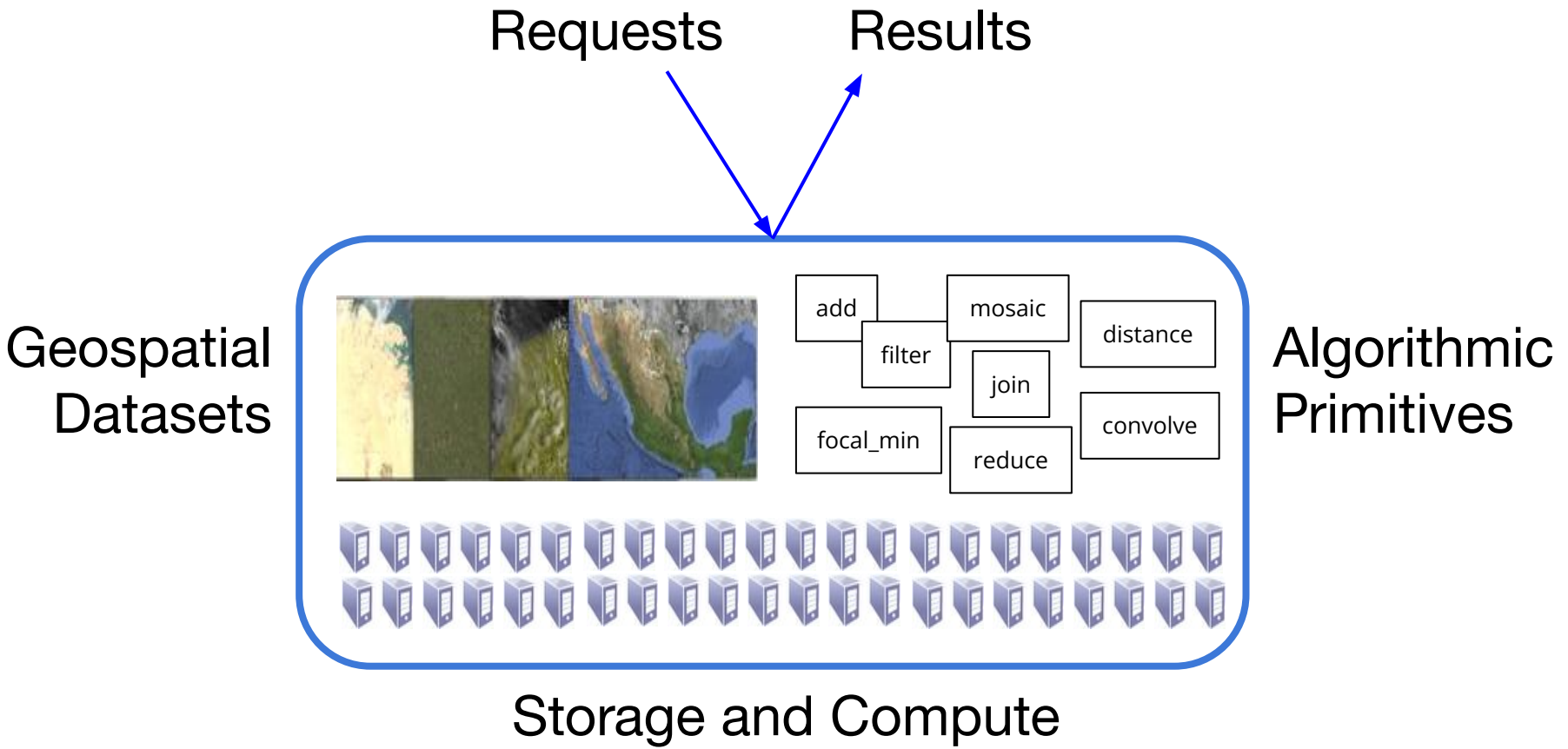
Normalized Difference Water Index (NDWI)

(Green - NIR) / (Green + NIR)




Normalized Difference Built-Up Index (NDBI)
(SWIR-1 - NIR) / (SWIR-1 + NIR)

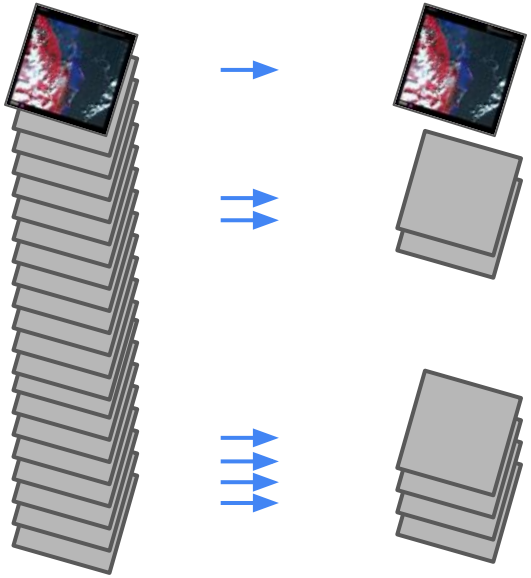
Map/Reduce Programming Concepts

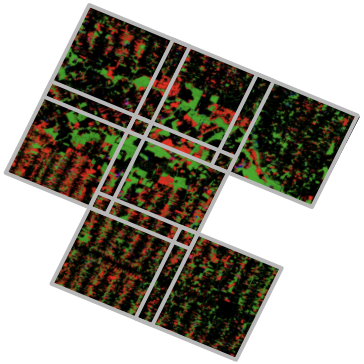

Requests
Results
Geospatial Datasets
add
mosaic
distance
filter
join
convolve
focal_min
reduce
Algorithmic Primitives
Storage and Compute

filter()

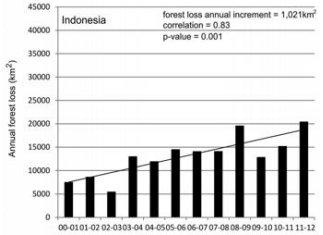
map()

reduce()





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



How to write parallel computing code?

- Filter your collection to get a collection with desired data
- Write a function to process a single image or feature
- `map()` that function on your filtered collection
- `reduce()` the resulting collection

map()

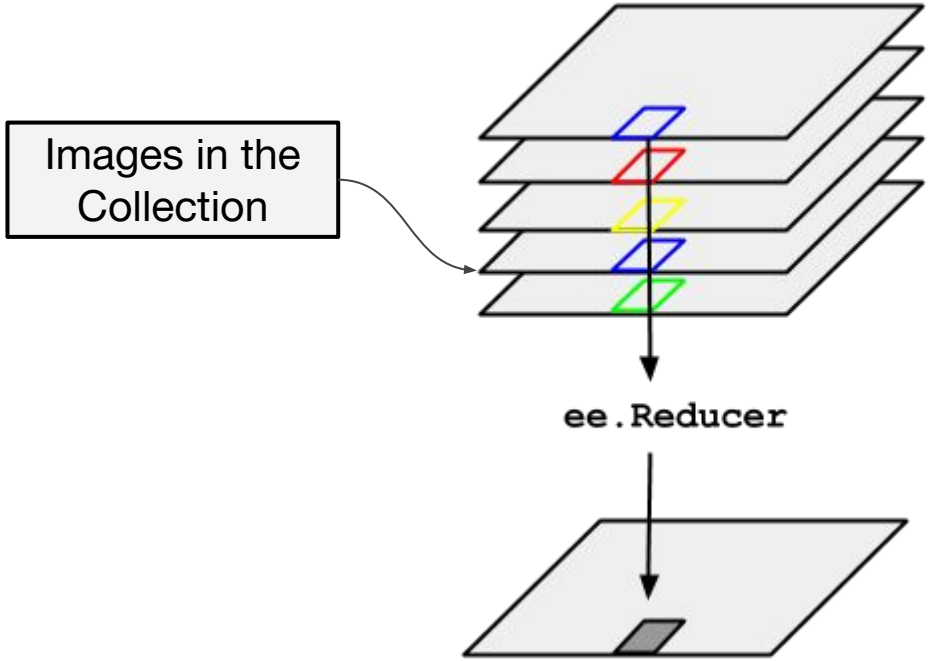
- map() works on following Earth Engine objects
  - List
  - Dictionary
  - ImageCollection
  - FeatureCollection
- You can run a map operation by calling the `map()` function

reduce()

- Earth Engine provides built-in reducers (ee.Reducer…)
- Reducers work on following Earth Engine objects
  - List
  - Image
  - ImageCollection
  - FeatureCollection
- You run a reducer by calling the `reduce()` function

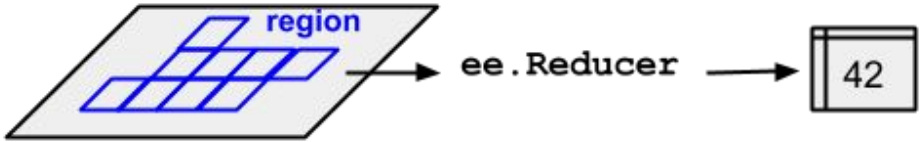
Reducing Image Collection

- Creates an Image
- Reducer is applied per pixel across images
- Example
  - Creating composites



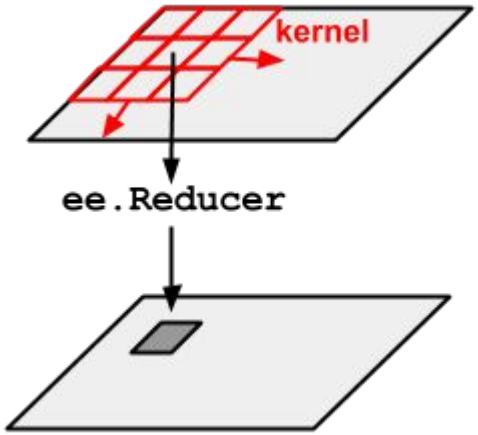
Reducing Image Region

- Creates a number of a dictionary
- Reducer is applied on all pixels of each band
- Example
  - Calculating area
  - Calculating average NDVI in a farm



Other types of reducers

- Convolutions `(reduceNeighborhood)`
- Zonal Statistics `(reduceRegions)`
- Statistics on Attribute Tables `(reduceColumn)`
- Raster to Vector conversion `(reduceToVectors)`
- Vector to Raster conversion `(reduceToImage)`
- Linear Regression `(linearFit, linearRegression)`



Tips for programming

- If/else conditions
  - filter / map / merge / reduce
  - See example
- Learn more about Reducers
  - https://courses.spatialthoughts.com/gee-advanced.html#reducers

Quiz 2

Feedback

Please fill this 2-minute anonymous feedback survey,

Resources

- Earth Engine User Guide
- Awesome GEE Community Catalog
- Earth Engine Fundamentals and Applications (EEFA)
  - Open access book [PDF] [Text] [Videos]
  - 50 Chapters, 100+ authors
  - Covering both GEE concepts and domain-specific applications
- Spatial Thoughts OpenCourseWare
  - End-to-End Google Earth Engine
  - Google Earth Engine for Water Resources Management
  - Creating Publication Quality Charts with GEE
  - Earth Engine Advanced Concepts

Cloud-Based Remote Sensing with Google Earth Engine: Fundamentals and Applications | SpringerLink

Notes

- S2 BRDF correction [paper] [code]
- S2 Phenology Parameters [github]