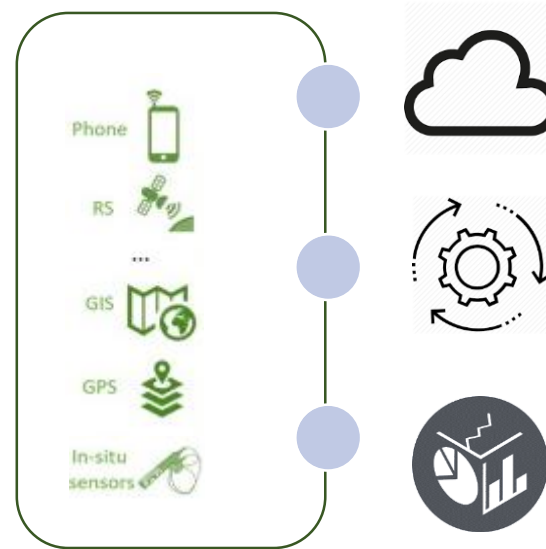


Geoinformatics and Big Data



Big Data Framework



Insights

Volume {Amount of space}

{Petabytes of archive data}

Variety {Diversity}

{heterogeneity in data and sensors}

Velocity {Momentum}

{Cloud storage and processing, API Services}

Veracity {Accuracy}

{Machine learning algorithms}

Big Data Tools

GBDX
aWhere
Google Earth Engine

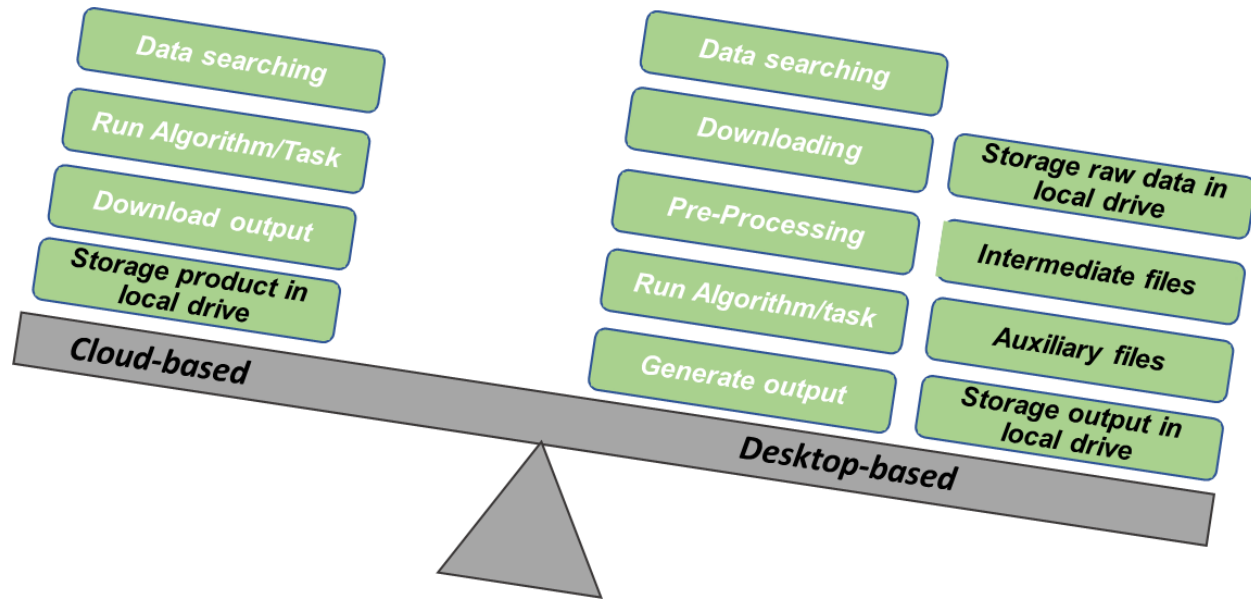
Python: Jupyter Notebook
R
Java script

HDFS
Spark
MongoDB



Why?

- Downloading dataset
- Processing huge dataset
- Sharing dataset



CGIAR Initiative

<https://bigdata.cgiar.org/>



Platform for
Big Data
in Agriculture

[ABOUT](#) ▾ [WHAT WE DO](#) ▾ [EVENTS](#) ▾ [NEWS AND MEDIA](#) ▾



Out Now: Our Year in Review

Building the Alliance for an
Agricultural Data Revolution



API

An Application Programming Interface (**API**) is a set of subroutine definitions, communication protocols, and tools for building software/application.



<https://www.youtube.com/watch?v=s7wmiS2mSXY>





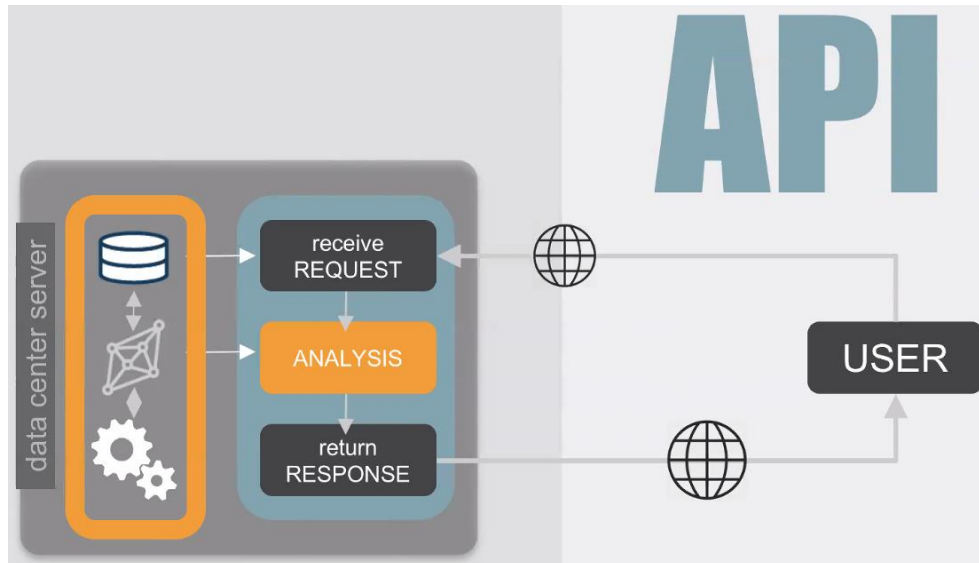
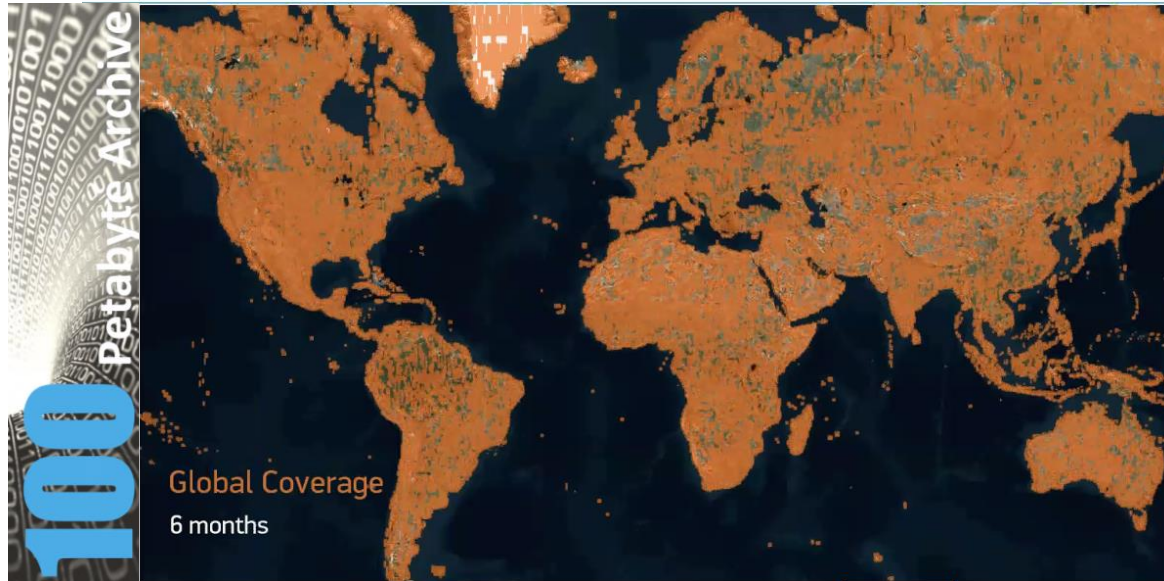
GBDX

Geo Big Data Ecosystem

- Platform and Workflow
 - GitHub and Python Jupyter Notebook
- Very High resolution data and potential of very high resolution data
 - Different sensor data and their characteristics
 - Data Searching and Few basic syntax
 - Case study
 - Discussion

GBDX Platform

How much Information DigitalGlobe Collects



GBDX APIs



AWS - Amazon Web Services

on demand cloud computing platform

Docker

software containerization platform

S3 - Simple Scalable Storage

store and retrieve any amount of data

Docker Image

standardized wrapper for your algorithm

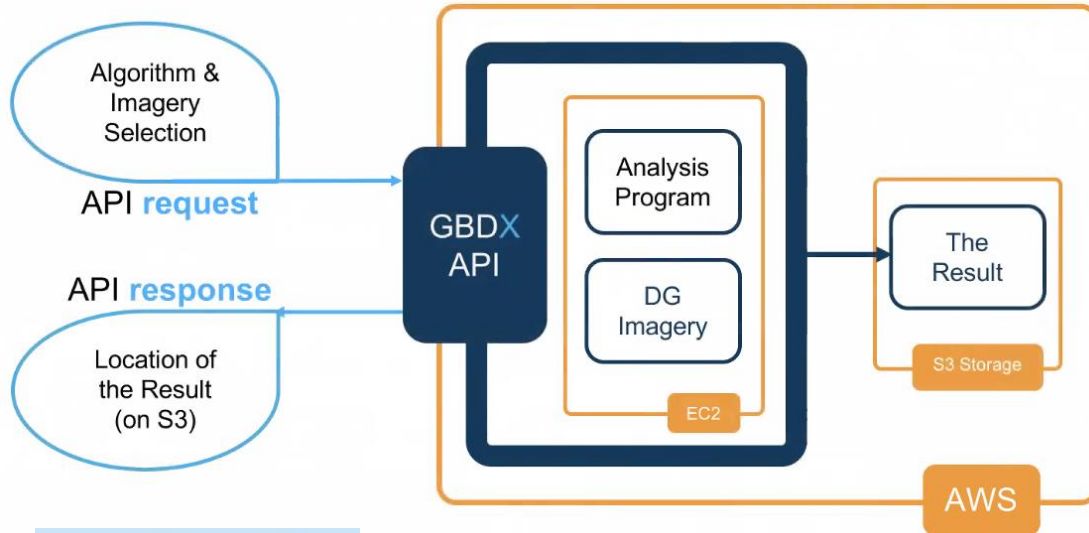
EC2 - Elastic Cloud Compute

scalable computing capacity

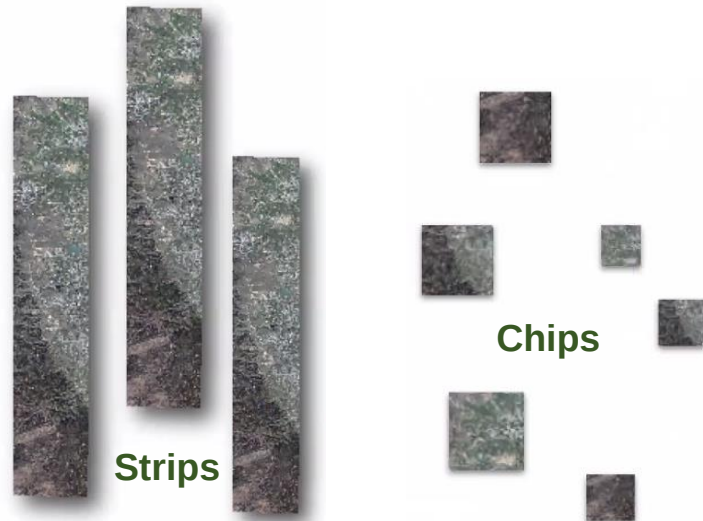
Docker Hub

a repository of images

How it works

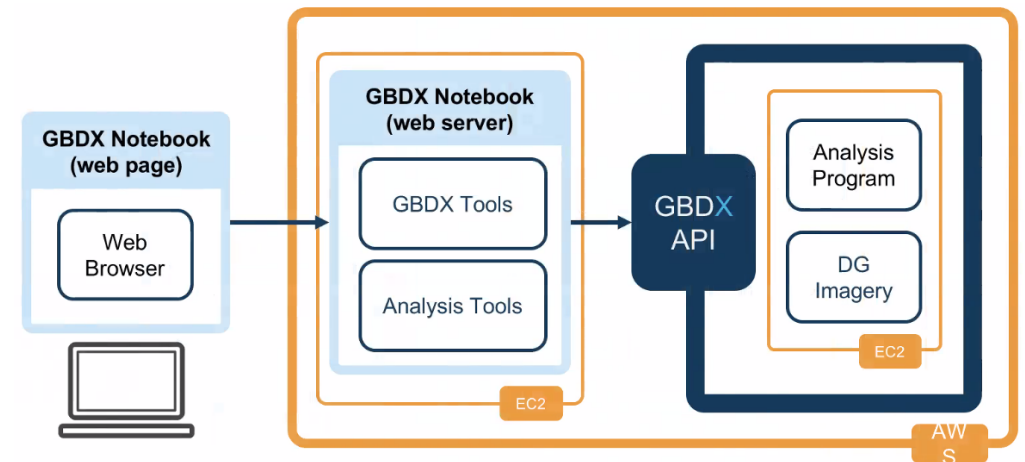
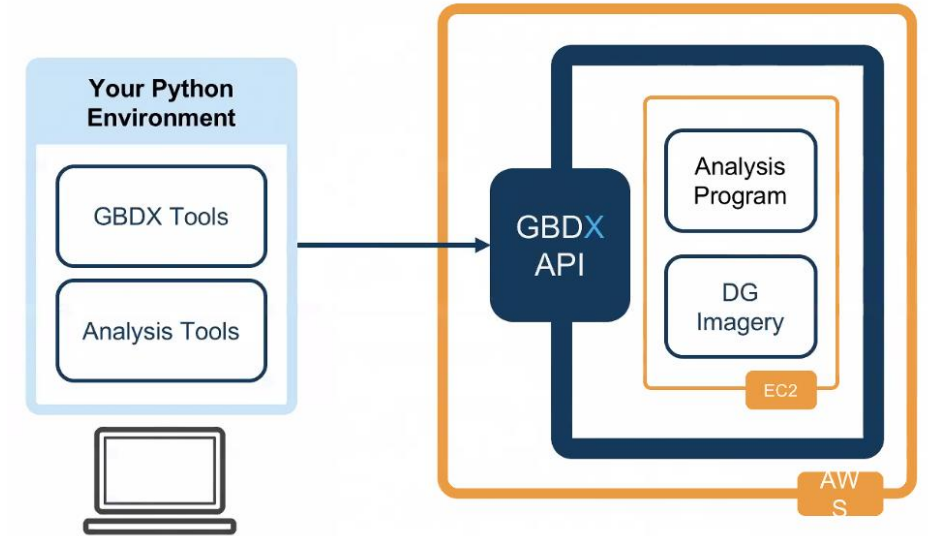


User defined workflow



Strips and Chips: Processing time and Data volume

Two Ways to make an API Call





Available data

DigitalGlobe

- WorldView-3
- WorldView-2
- WorldView-1
- GeoEye-1
- IKONOS
- QuickBird

MDA

- RADARSAT-2

NASA

- Landsat-8

GBDX University

Sign Up Docs API Reference Release Notes Status Page Contact Us Log In

Docs > Get Started with GBDX APIs

Q Search

GETTING STARTED

Get Started with GBDX APIs

GBDX Overview

Evaluation Plan

Postman Instructions & Collections

gbdxtools

GBDX OPERATIONS

Platform Status Page

Deprecation Schedule

GBDX Compute Hours

GBDX Supported Domains

GBDX Security

GBDX STORIES

Get Started with GBDX APIs

SUGGEST EDITS

Learn how to use GBDX APIs and services using this quick tutorial.

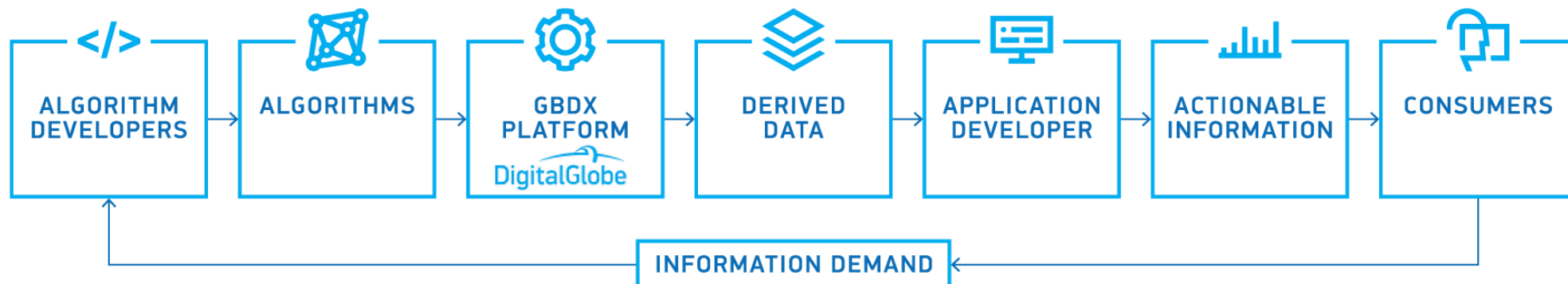
For this tutorial, we'll use the Postman client to make API calls. The GBDX Postman collection features a section called "Getting Started." The requests in this collection complement the steps in this guide. Step 2 will explain how to set up Postman.

Table of Contents

Section	Description
Step 1: Get your account credentials	Learn how to get your account credentials
Step 2: Set up the Postman client	Set up the Postman client and import the GBDX environment and collection files
Step 3: Get a token	GBDX authentication is token-based. Learn how to get a token that will allow you to make API requests

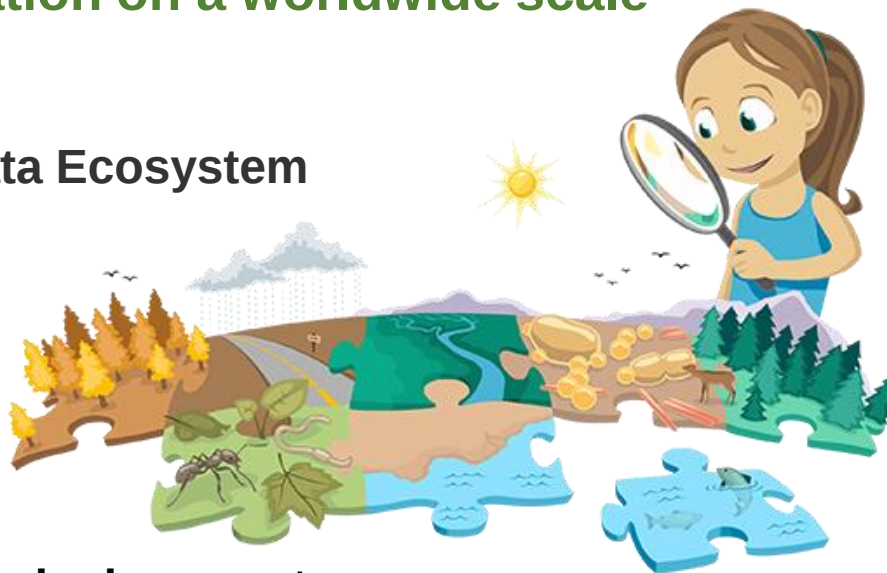
<https://gbdxdocs.digitalglobe.com/docs/getting-started>

Moving Beyond the Pixel



The GBDX platform enables to **build, access and run advanced workflows and tools** that extract **actionable information on a worldwide scale**

Geo Big Data Ecosystem



Enables data Producers and Developers to **build solutions**

Networked ecosystem

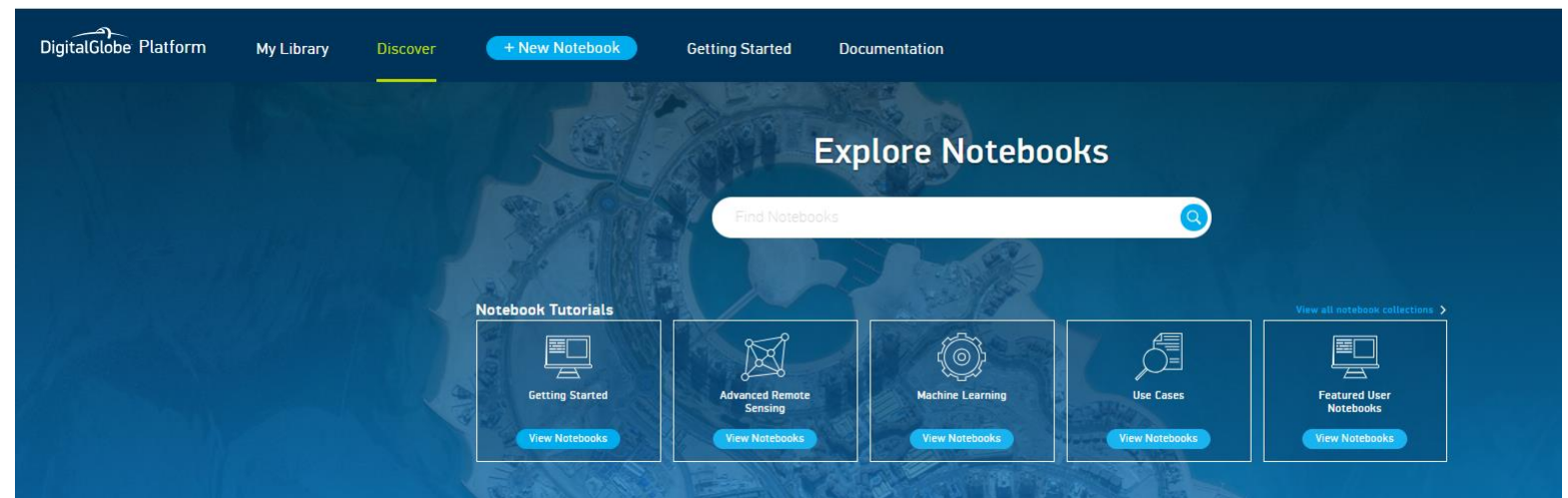
GBDX Notebook




Notebooks

[Don't remember your password?](#)

[LOG IN >](#)



<https://notebooks.geobigdata.io/>

Image Searching

+ New Notebook

Getting Started Documentation

Start with Imagery
Recommended

Blank
Advanced

Explore Notebooks

Find Notebooks

Notebook Tutorials

- Getting Started
View Notebooks
- Advanced Remote Sensing
View Notebooks
- Machine Learning
View Notebooks
- Use Cases
View Notebooks
- Featured User Notebooks
View Notebooks

DigitalGlobe Platform My Library Discover + New Notebook Getting Started Documentation

jupyter introduction_to_imagery_access_in_gbdxtools (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

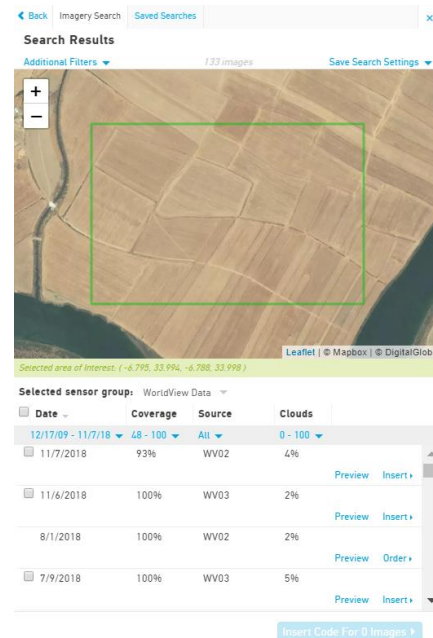
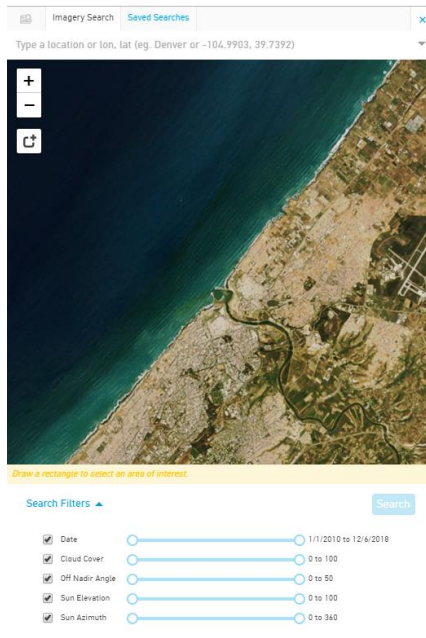
Not Trusted Python 3

Publish Notebook

Hide Tips Code Samples

DigitalGlobe Notebook Buttons Explained

- Imagery Button**
Choose an area of interest, sort and filter imagery, and then add an image or collections of images to your notebook.
Open
- Open Data Button**
Explore open imagery around disaster events.
Open
- Orders Button**
Place orders for unordered imagery.
Open
- Deploy Button**
Deploy this notebook as a GBDX task. This will allow you to run your notebook at larger scales in just a few easy steps.
Open



```
In [1]: from gbdxtools.task import env
from gbdxtools import CatalogImage
%matplotlib inline

catalog_id = env.inputs.get('catalog_id', '103001008888E800')
bbox = env.inputs.get('bbox', '-6.795043945639919, 33.99424688250682, -6.787877082497289, 33.99817847460809')

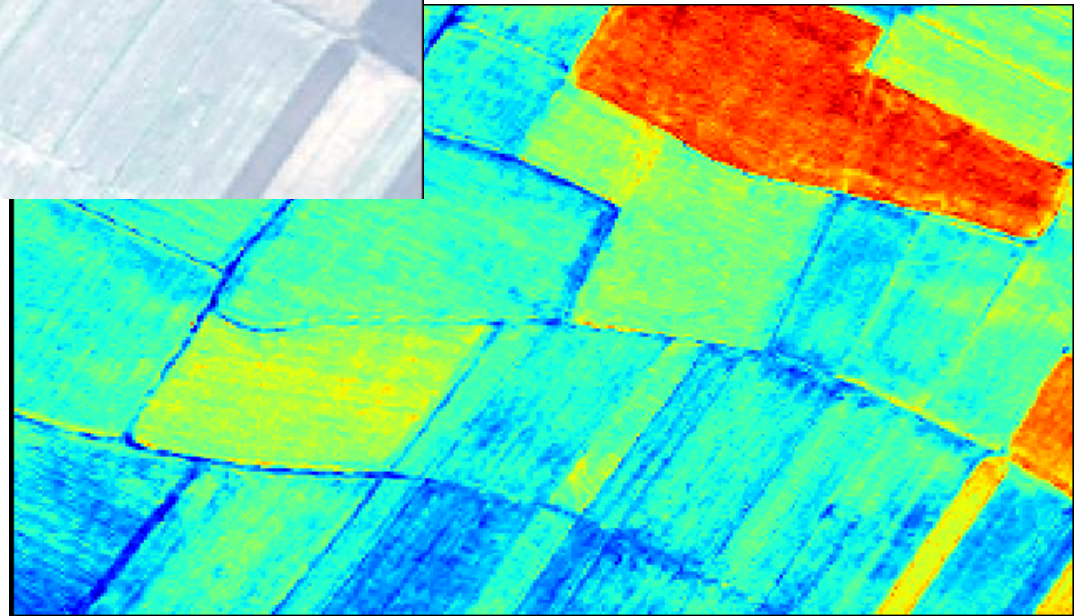
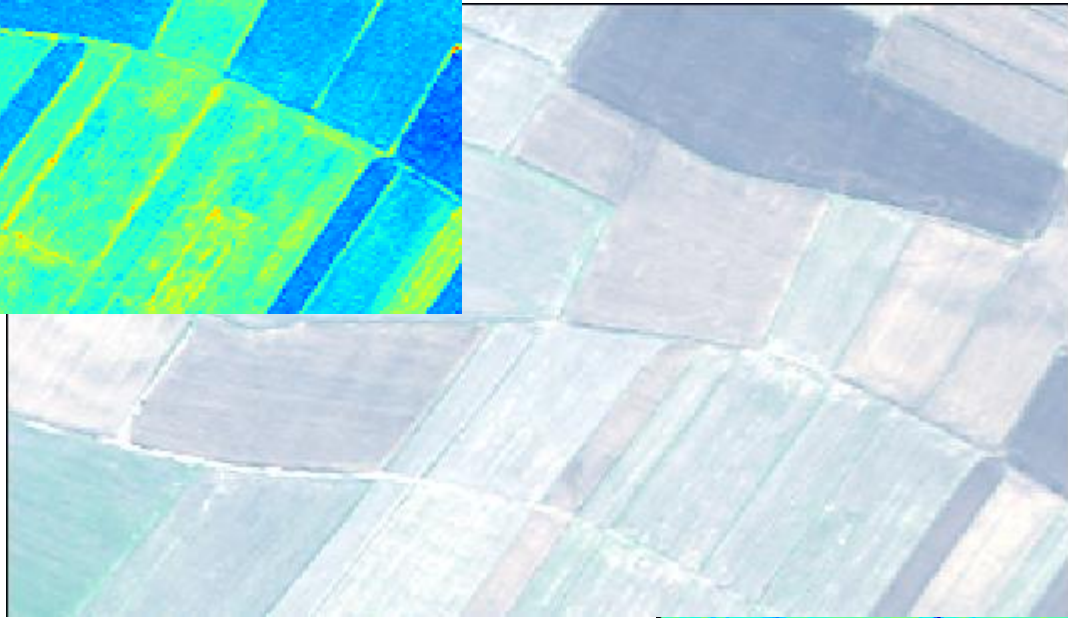
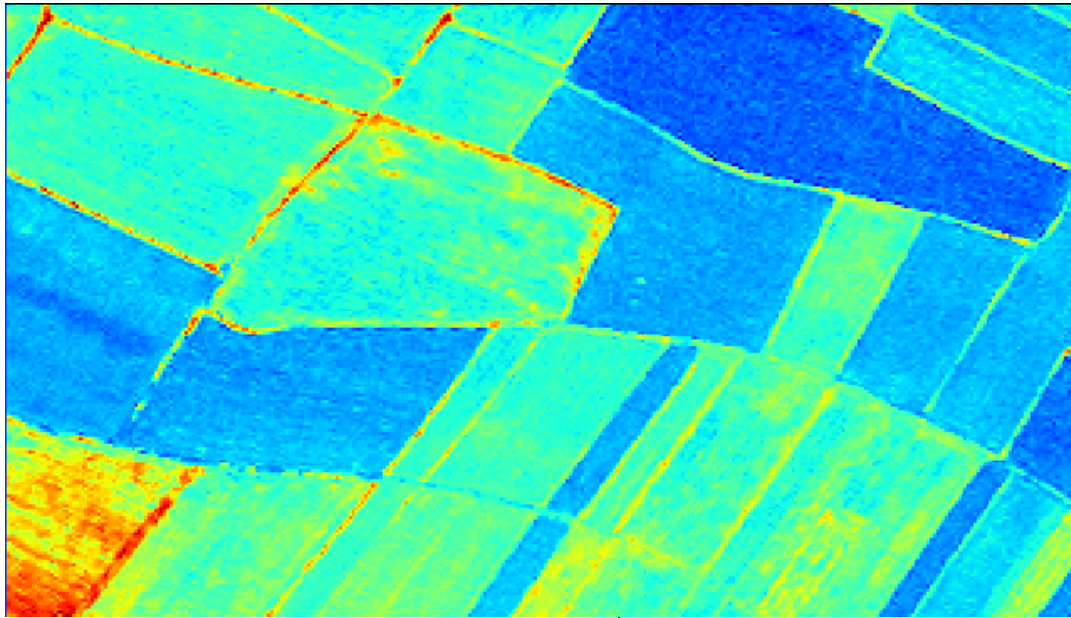
image = CatalogImage(catalog_id, band_type="MS", bbox=map(float, bbox.split(",")))
image.plot(w=10, h=10)
```





Case Study

Hands-on





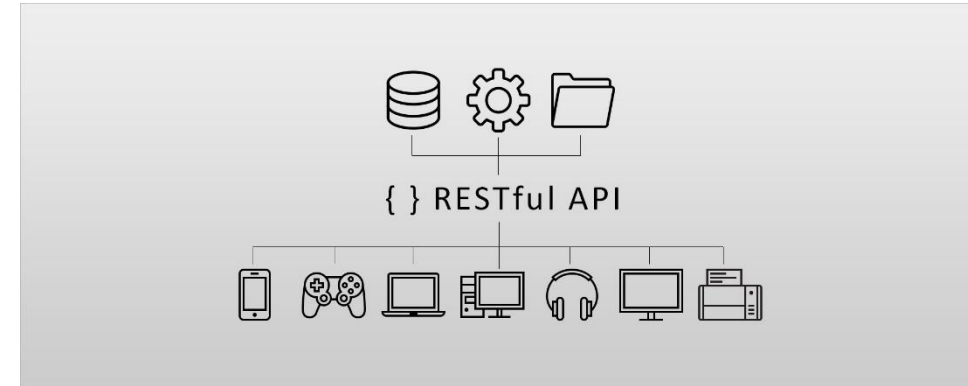
- Platform and Workflow
 - R and RStudio
- Libraries and API fetching
 - Case study
 - Discussion

aWhere Services

aWhere provides Ag applications and big data insights.

RESTful APIs

aWhere's API platform is built as REST-based web services.



Our APIs

Easy Integration

Community

Help

Store

Search

Sign In

Get Started >

Our APIs

Overview

Get Started

About Our Data >

Conventions

Authentication

API Reference ▼

Fields & Plantings

aWhere's Agriculture Intelligence API Platform

The API reference contains all the details you need for accessing our new, next-generation API platform. The new platform is a comprehensive agriculture intelligence system that is both powerful and flexible, designed for a wide variety of use cases and applications.

► Fields and Plantings

The Fields API is how you manage your field locations in the aWhere Platform, and Plantings record the crop and field activity to empower agronomic calculations and modeling. | [View Documentation](#) >

► Weather

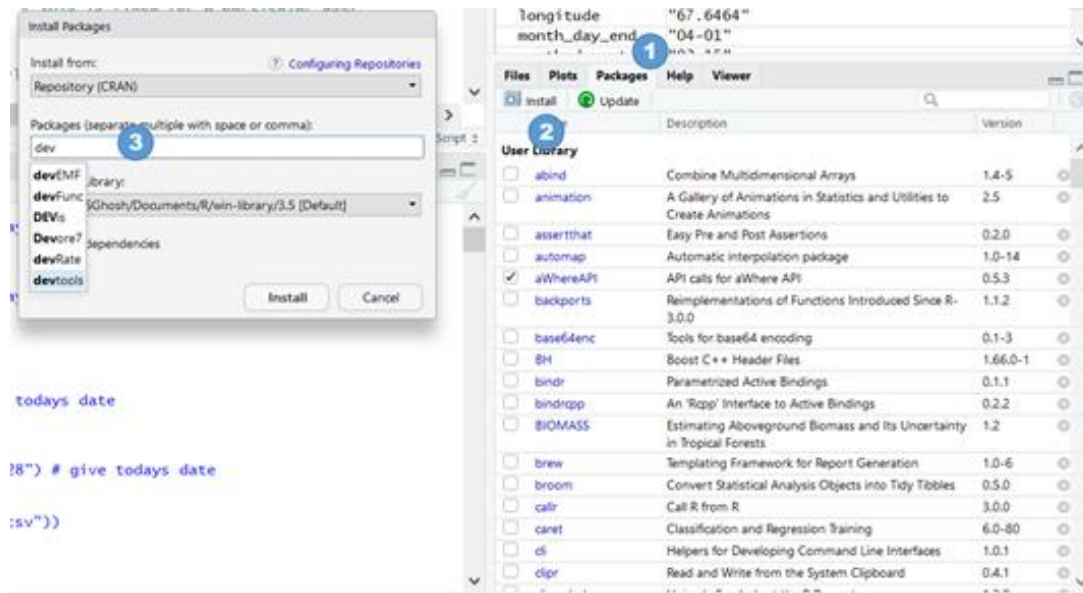
Privacy

<https://developer.awhere.com/api/reference>

Installation aWhere API

Requirement: R and/or Rstudio

Install **devtools**



Files Plots Packages Help Viewer

R: API calls for aWhere API Find in Topic

API calls for aWhere API

Documentation for package 'aWhereAPI' version 0.5.3

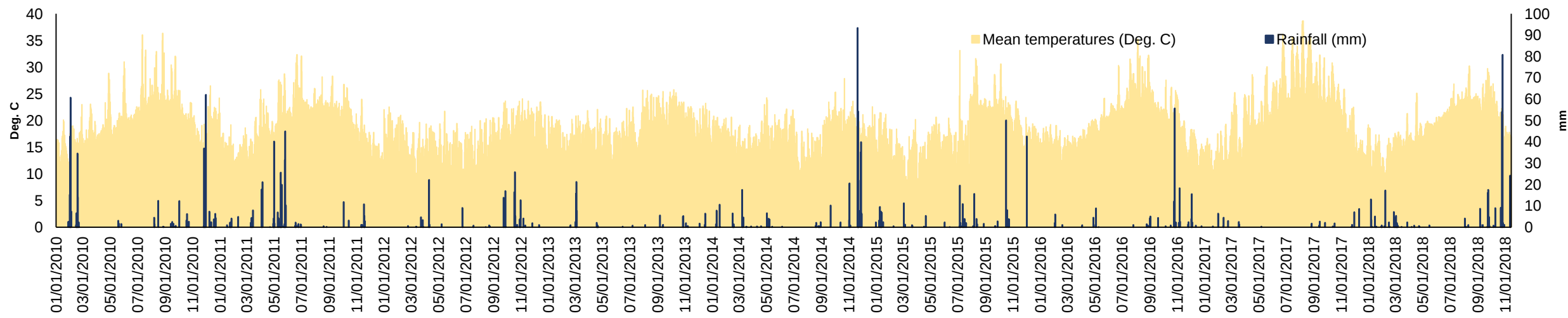
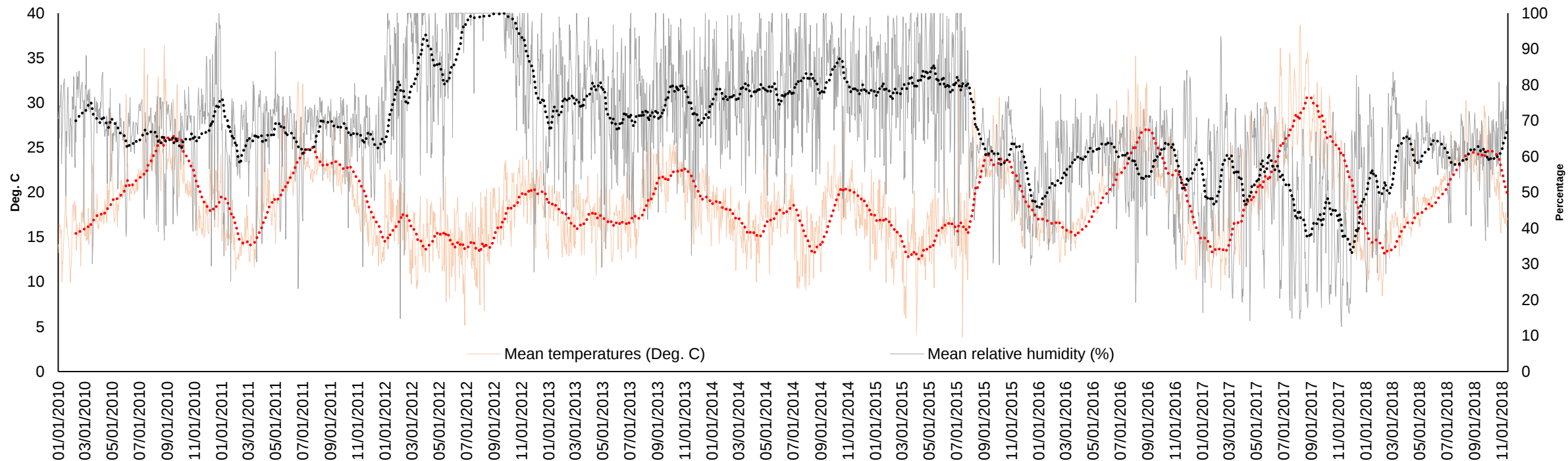
- [DESCRIPTION file.](#)
- [User guides, package vignettes and other documentation.](#)

Help Pages

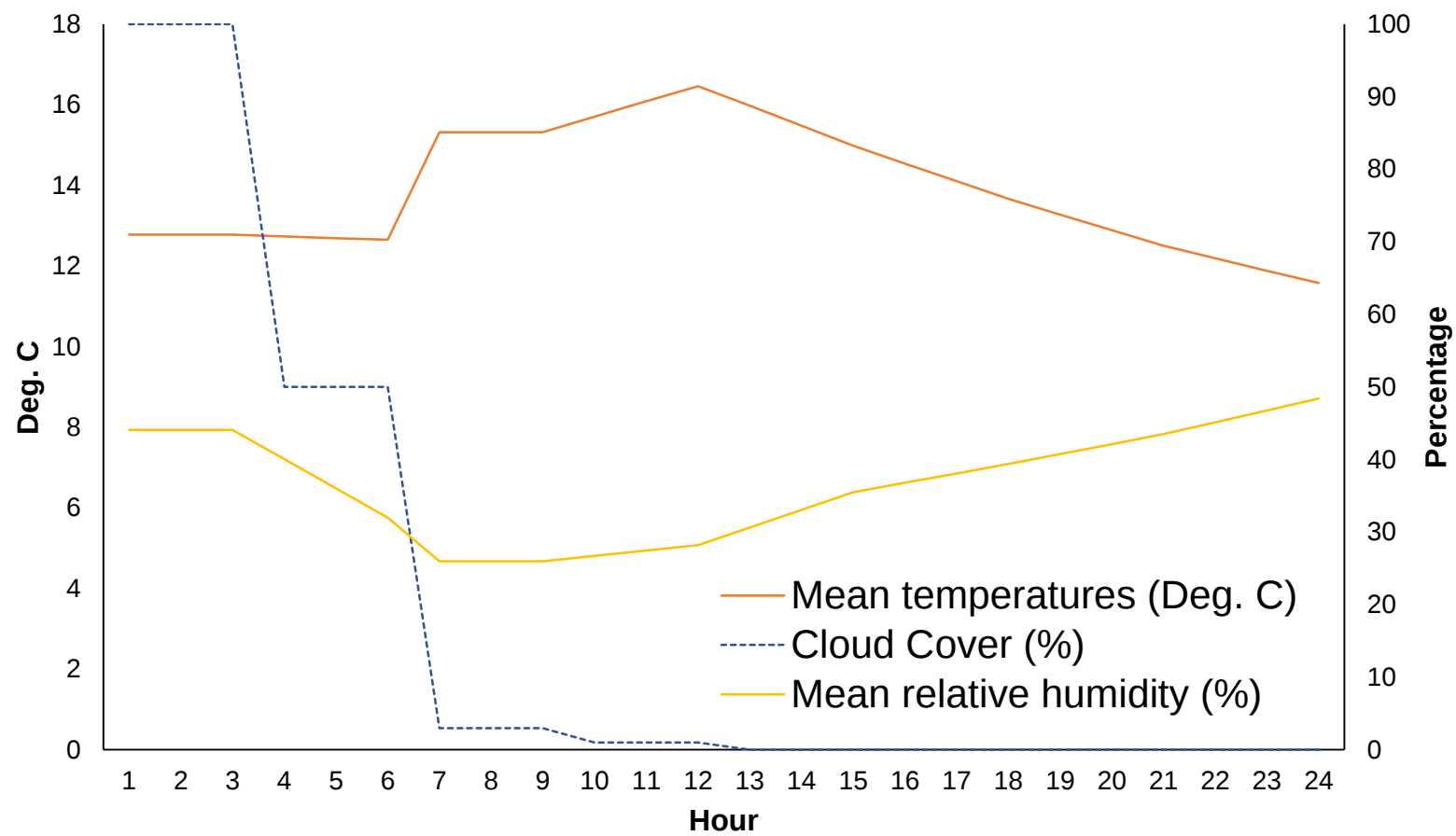
agronomic_norms_fields	agronomic_norms_fields
agronomic_norms_latlng	agronomic_norms_latlng
agronomic_values_fields	agronomic_values_fields
agronomic_values_latlng	agronomic_values_latlng
checkAccumulationStartDate	Check Accumulation Start Date

```
devtools::install_github("aWhereAPI/aWhere-R-Library")
```

Historical data (30.53,-9.34)

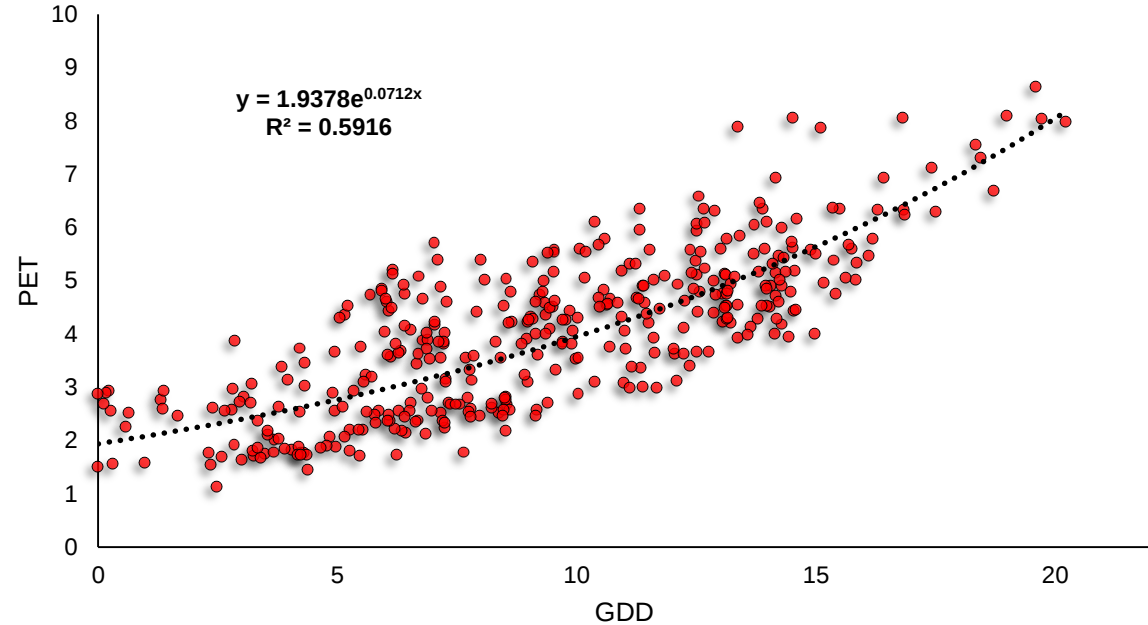


Now casting (15/11/2018)



Growing Degree-Days (GDDs)

Growing Degree-Days are a standard metric for ambient heat critical to plant growth.



Potential Evapotranspiration (PET)

Potential Evapotranspiration is a metric in agronomy that can drive some powerful insights especially around irrigation decisions.



Case Study

Hands-on



- **Platform and Workflow**
- **Libraries and API fetching**
 - **Case study**
 - **Discussion**

Google Earth Engine

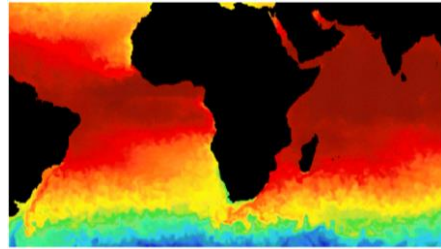
<https://earthengine.google.com/>

DATASETS FAQ TIMELAPSE CASE STUDIES PLATFORM BLOG SIGN UP

A planetary-scale platform
for Earth science data &
analysis

Powered by Google's cloud infrastructure

▶ WATCH VIDEO



Google Earth Engine is a cloud-based platform for planetary-scale environmental data analysis.

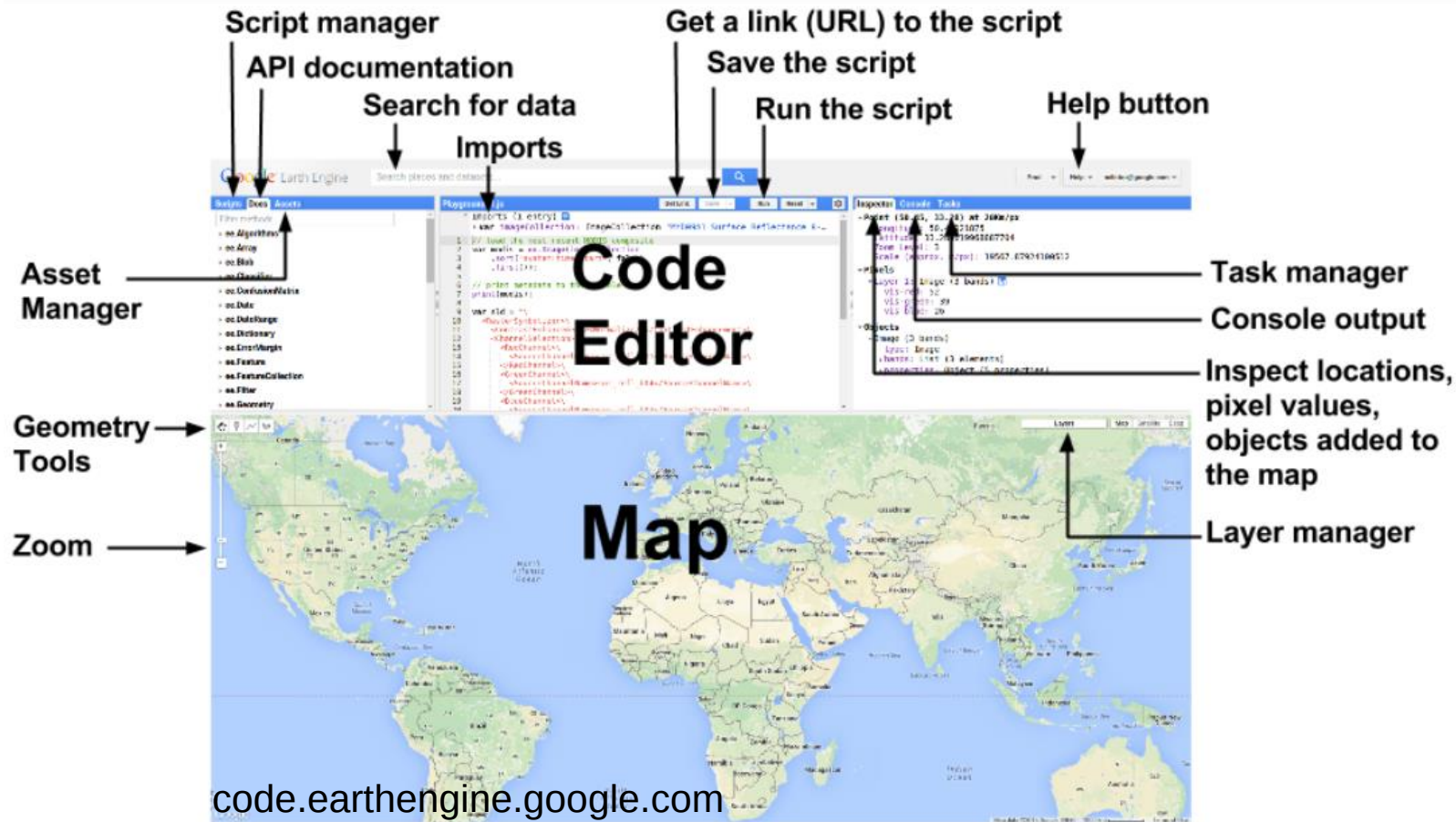
- Datasets:** A petabyte-scale archive of publicly available remotely sensed imagery and other data.

- Compute power:** Google's computational infrastructure optimized for parallel processing of geospatial data.

- APIs:** APIs for JavaScript and Python ([hosted on GitHub](#)) for making requests to the Earth Engine servers. These docs focus on JavaScript.

- Code Editor:** An online Integrated Development Environment (IDE) for rapid prototyping and visualization of complex spatial analyses using the Javascript API.

Earth Engine Code Editor



- JavaScript code editor
- Map display for visualizing geospatial datasets
- API reference documentation (Docs tab)
- Git-based script manager (Scripts tab)
- Console output (Console tab)
- Task manager (Tasks tab) to handle long-running queries
- Interactive map query (Inspector tab)
- Search of the data archive or saved scripts
- Geometry drawing tools



Thank You





Sources of images used in this presentation

<https://depositphotos.com/43981793/stock-photo-word-cloud-data-analysis.html>

<http://rkinsight.com/insights-discovery/>

<https://www.vutlan.com/blog/new-api-made-public-b34.html>

<https://bigdata.cgiar.org/>

https://github.com/GeoBigData/gbdx-training/blob/master/gbdx_notebooks_module

<https://www.youtube.com/watch?v=s7wmiS2mSXY>

<https://gbdxdocs.digitalglobe.com/docs/getting-started>

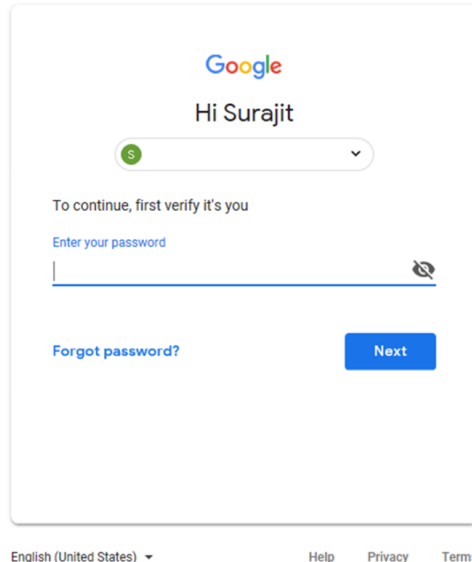
<https://developer.awhere.com/api/reference>

<https://earthengine.google.com/>

Module 1. Step by step instructions for Google Earth Engine Setup

Step 1.

Signup for Earth Engine (<https://signup.earthengine.google.com>).



Google

Hi Surajit

To continue, first verify it's you

Enter your password

Forgot password? Next

English (United States) Help Privacy Terms

Once you have been accepted, you will receive an email with additional information.

Step 2.

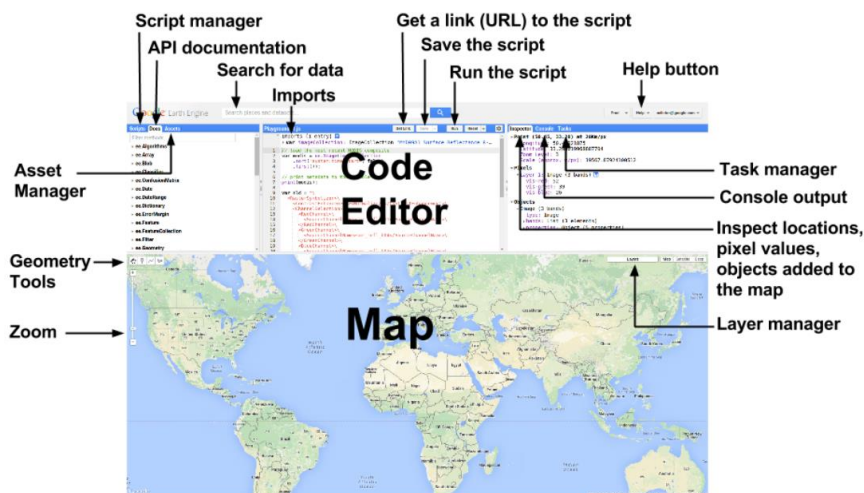
Go through JavaScript for Earth Engine.

https://developers.google.com/earth-engine/tutorial_js_01

Step 3.

Explore Earth Engine Code Editor (the IDE for writing Earth Engine JavaScript code in a web browser)

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



Module 2. TRMM time series data analysis

//Author: GU, ICARDA

```
//https://developers.google.com/earth-engine/datasets/catalog/TRMM_3B43V7
var dataset = ee.ImageCollection('TRMM/3B43V7') //call the data set
    .filter(ee.Filter.date('1998-01-01', '2018-12-31')); //set the time range

//precipitation unit mm/hr
var precipitation = dataset.select('precipitation');
//Set you area of interest (region)
var geom = ee.FeatureCollection('users/point/Rabat');
//Set you area of interest (point with a buffer around a point.)
//var geom = ee.Geometry.Point(-6.8604577,33.9803603).buffer(2000); //(longitude, latitude)
format

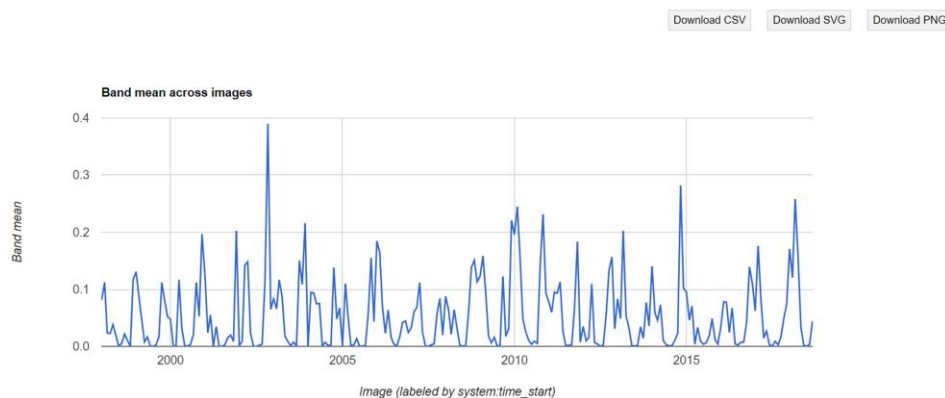
Map.addLayer(ee.Image().paint(geom, 0, 2), {}, 'Area of Interest');
Map.setCenter(-6.8604577,33.9803603, 7);

// Create and print time series chart.
print(ui.Chart.image.series(precipitation, geom, ee.Reducer.mean()));

/*
var precipitationVis = {
  min: 0.1,
  max: 1.2,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};

Map.addLayer(precipitation, precipitationVis, 'Precipitation');
*/
```

Output



Module 3. Vegetation analysis

```
//Author: GU, ICARDA

//Getting the S2 image collection of a year
var S2 = ee.ImageCollection('COPERNICUS/S2')
    .filterDate('2015-10-01', '2018-12-11'); //

//Computing the NDVI using band8 and band4
var ndvi = S2.map(function(image) {
    return image.select().addBands(image.normalizedDifference(['B8', 'B4']));
});

var vis = {min: 0, max: 1, palette: [
    'FFFFFF', 'CE7E45', 'FCD163', '66A000', '207401',
    '056201', '004C00', '023B01', '012E01', '011301'
]};

Map.addLayer(ndvi, vis, 'NDVI');
Map.setCenter(-6.8857609, 33.9167924, 8);

// Create a panel to hold our widgets.
var panel = ui.Panel();
panel.style().set('width', '300px');

// Create an intro panel with labels.
var intro = ui.Panel([
    ui.Label({
        value: 'Chart',
        style: {fontSize: '20px', fontWeight: 'bold'}
    }),
    ui.Label('Click on the map to generate chart')
]);
panel.add(intro);

var lon = ui.Label();
var lat = ui.Label();
panel.add(ui.Panel([lon, lat], ui.Panel.Layout.flow('horizontal')));

Map.onClick(function(coords) {
    lon.setValue('lon: ' + coords.lon.toFixed(2)),
    lat.setValue('lat: ' + coords.lat.toFixed(2));

    var point = ee.Geometry.Point(coords.lon, coords.lat);
    var dot = ui.Map.Layer(point, {color: 'FF0000'});
    Map.layers().set(1, dot);

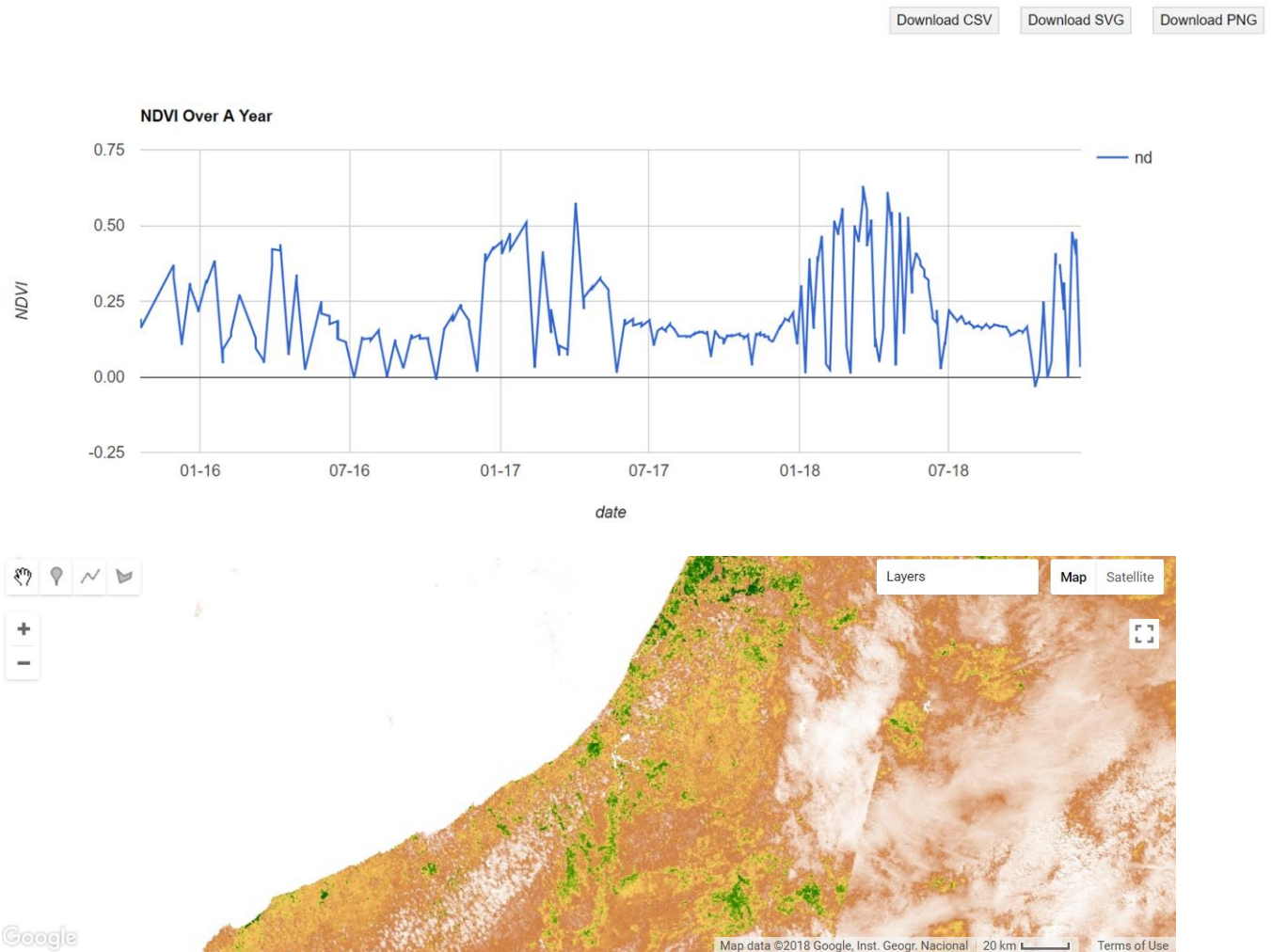
    // NDVI chart.
    var ndviChart = ui.Chart.image.series(ndvi, point, ee.Reducer.mean(), 500);
    ndviChart.setOptions({
        title: 'NDVI Over A Year',
        vAxis: {title: 'NDVI'},
        hAxis: {title: 'date', format: 'MM-yy', gridlines: {count: 7}},
    });
});
```

```
});
panel.widgets().set(2, ndviChart);

//Get the NDVI time series from the clicked point region
var y = ndvi.getRegion(point, 250).slice(1)
    .map(function(item) { return ee.Number(ee.List(item).get(-1)).divide(1e4) });

});

Map.style().set('cursor', 'crosshair');
// Add the panel to the ui.root.
ui.root.insert(0, panel);
```



Useful links

<https://developers.google.com/earth-engine/>
<https://youtu.be/4E6yQLoGO2o>
<https://americaview.org/program-areas/education/google-earth-engine-tutorials/>
https://geoscripting-wur.github.io/Earth_Engine/