



Introduction to Google Earth Engine GEE工作坊

National Taiwan Normal University
2025/11/20



1

Lecture's content

Introduction

Raster, vectors

Signing in, web-based interface, datasets

How it works

Using GEE

3 case studies

- Creating a map of temperature anomalies, exporting
- Modeling species distribution based on external PA data
- Classifying land cover

Important notes

2

Lecture's content

Introduction

Raster, vectors

Signing in, web-based interface, datasets

How it works

Using GEE

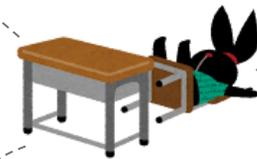
3 case studies

- Creating a map of temperature anomalies, exporting
- Modeling species distribution based on external PA data
- Classifying land cover

Important notes

Please let me know when something is not well explained

Copy-paste command lines from the webpage



Questions? Need for more information? Just tell me.

3

Lecture's content

Jonathan Peereman (jonathan.peereman@uliege.be)

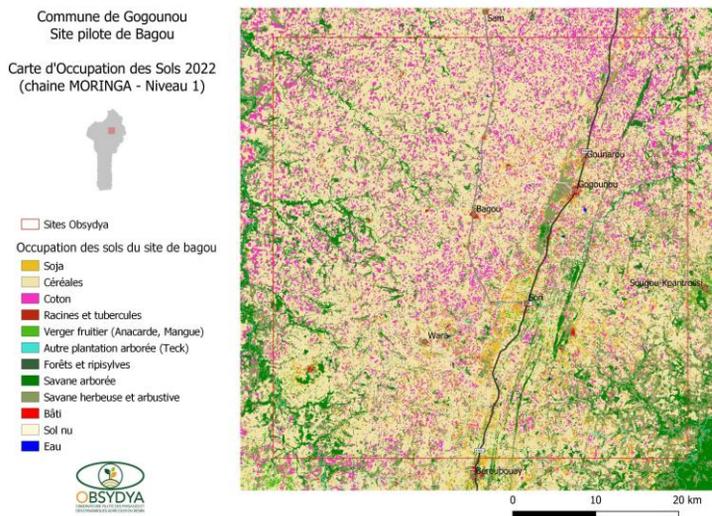
Ecology & remote sensing

- Disturbances
- Landscape dynamics
- Nature & Agriculture

4

Some applications

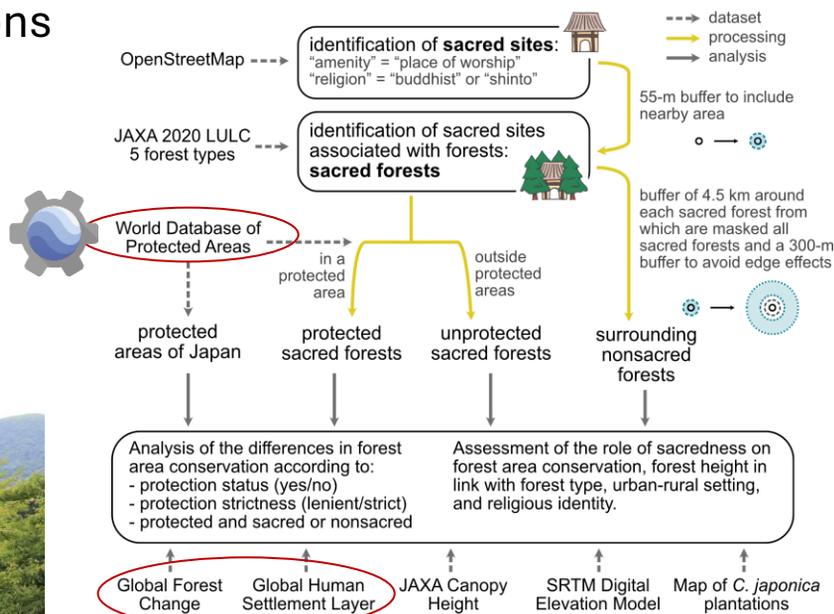
貝南 Benin



5

Some applications

Sacred forests of Japan



美しき日本

6

Remote sensing & GIS

Reminders

Remote sensing & GIS

- Important tools
- Based on available data
- Produce new data

7

Remote sensing & GIS

Reminders

Rasters

Picture that can be “seen” as a table where each cell has a value

Spatial resolution: size of the pixel



8

Rasters

Picture that can be “seen” as a table where each cell has a value

Spatial resolution: size of the pixel



9

Vectors

Spatial features: shape with spatial information to draw it + descriptive information (attribute data)



Rasters

Picture that can be “seen” as a table where each cell has a value

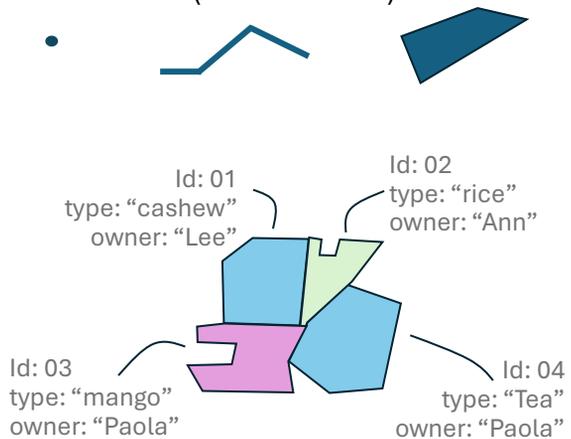
Spatial resolution: size of the pixel



10

Vectors

Spatial features: shape with spatial information to draw it + descriptive information (attribute data)



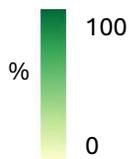
Remote sensing & GIS

Reminders

Rasters

Picture that can be “seen” as a table where each cell has a value

Rasters can be used to display very different information



11

Remote sensing & GIS

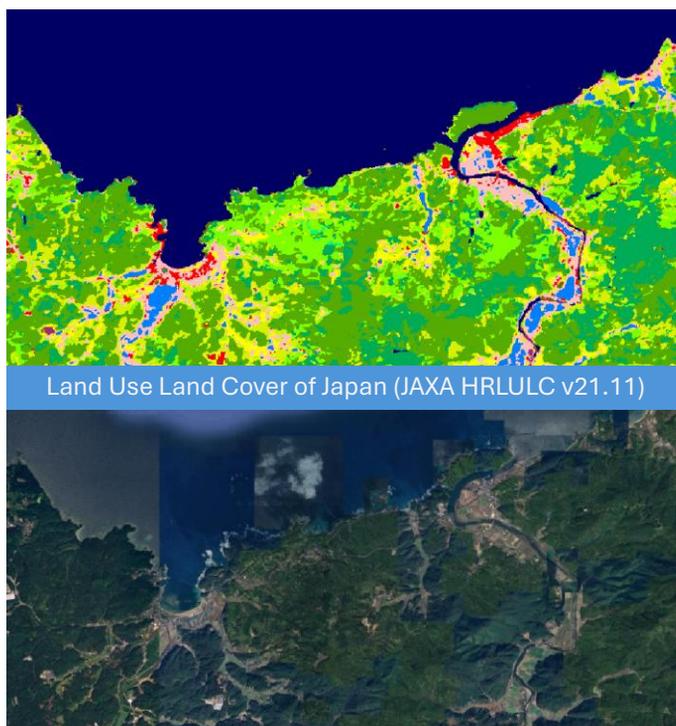
Reminders

Rasters

Picture that can be “seen” as a table where each cell has a value

Rasters can be used to display very different information

color	code	category
blue	1	water
red	2	urban
light blue	3	rice paddy
pink	4	crop
yellow	5	grass
light green	6	DBF
green	7	DNF
dark green	8	EBF
teal	9	ENF
brown	10	bare land
yellow-green	11	bamboo
purple	12	solar panel



12

Remote sensing & GIS

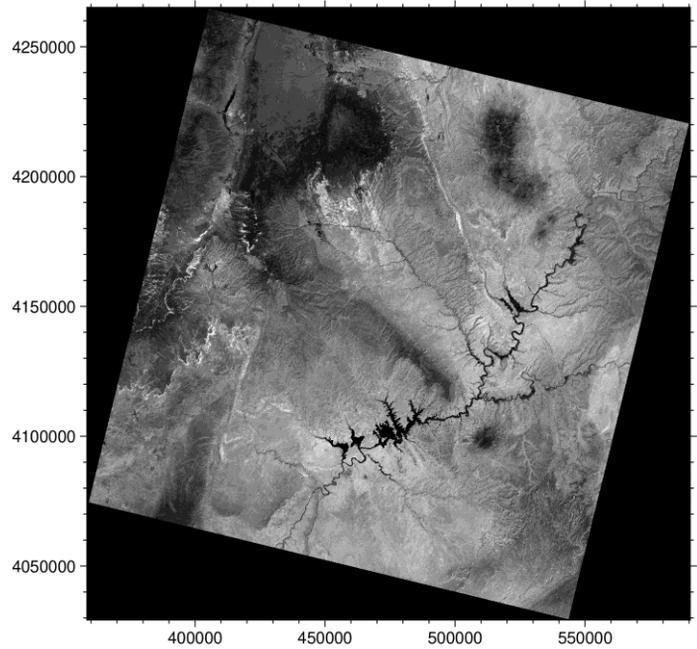
Reminders

Rasters

Picture that can be “seen” as a table where each cell has a value

Often more than one “band”

- Red, Green, Blue, ...
- Canopy height, canopy height SD, ...



13

Remote sensing & GIS

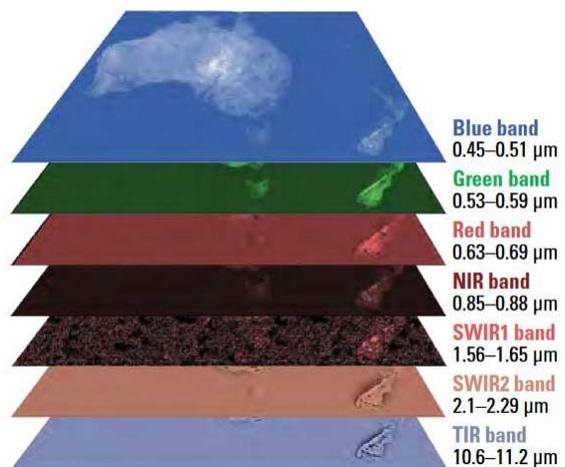
Reminders

Rasters

Picture that can be “seen” as a table where each cell has a value

Often more than one “band”

- Red, Green, Blue, ...
- Canopy height, canopy height SD, ...



GISGeography

14

Remote sensing & GIS

Reminders

Vectors

Spatial features: shape with spatial information to draw it + descriptive information (**attribute data**)

Protected areas



WDPAID	555593994.0000000000000000
WDPA_PID	555593994
PA_DEF	1
NAME	Chachayalaishan
ORIG_NAME	茶茶牙賴山野生動物重要棲息環境
DESIG	野生動物重要棲息環境
DESIG_ENG	Major Wildlife Habitat
DESIG_TYPE	National
IUCN_CAT	IV
INT_CRIT	Not Applicable
MARINE	0
REP_M_AREA	0
GIS_M_AREA	0
REP_AREA	20,068110000000001
GIS_AREA	20,071842017905698
NO_TAKE	Not Applicable
NO_TK_AREA	0
STATUS	Designated
STATUS_YR	2000
GOV_TYPE	Federal or national ministry or agency
OWN_TYPE	Not Reported
MANG_AUTH	Forestry Bureau
MANG_PLAN	Not Reported
VERIF	State Verified
METADATAID	1877
SUB_LOC	CN-71
PARENT_ISO	CHN
ISO3	TWN
SUPP_INFO	Not Applicable
CONS_OBJ	Not Applicable

15

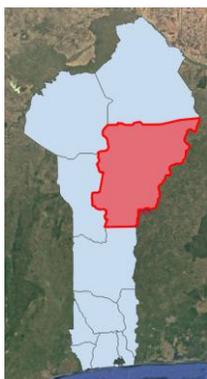
Remote sensing & GIS

Reminders

Vectors

Spatial features: shape with spatial information to draw it + descriptive information (**attribute data**)

Administrative units



▼ ben_admbnda_adm1_1m_salb_20190816	
adm1_name	Borgou
▶ (Derived)	
▶ (Actions)	
OBJECTID_1	4
adm1_name	Borgou
status	Member State
adm0_name	Benin
shape_Leng	7,73385771799
Type1	Département
ISO3	BEN
Shape_Le_1	7,73385771799
Shape_Area	2,11000613453
admin1Pcod	BJ04
admin0Pcod	BJ

16

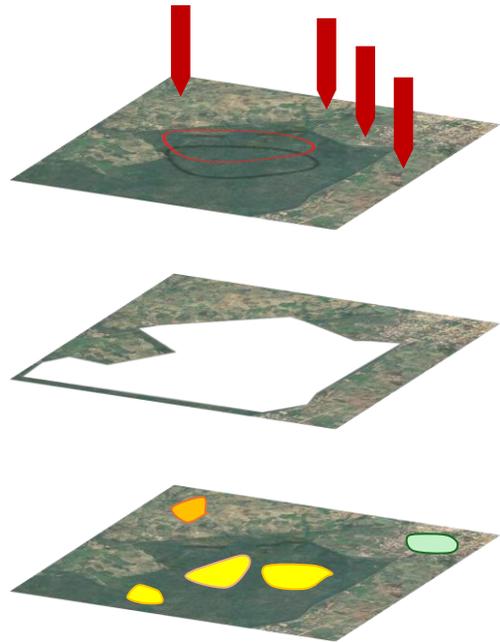
Remote sensing & GIS

Reminders

Rasters and vectors can be combined

- Extract values at sites or region of interest
- Mask an area
- Delimiting training areas of image classification

In QGIS, R, python, ...



17

Google Earth Engine

What is GEE ?

Interface

Data

Importing data

Javascript in GEE

Data catalogue

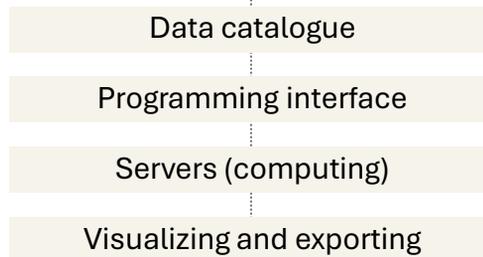
Functions (apidocs)

Planet



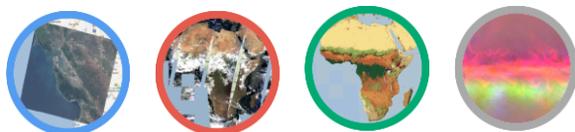
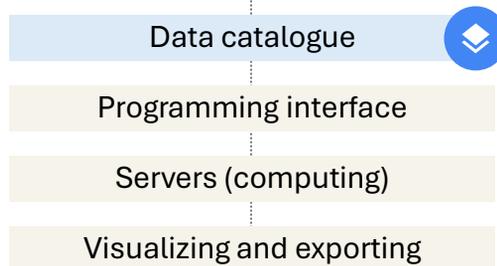
18

Google Earth Engine (GEE) Cloud-based spatial data processing and analyses



19

Google Earth Engine (GEE) Cloud-based spatial data processing and analyses



> 900 open datasets

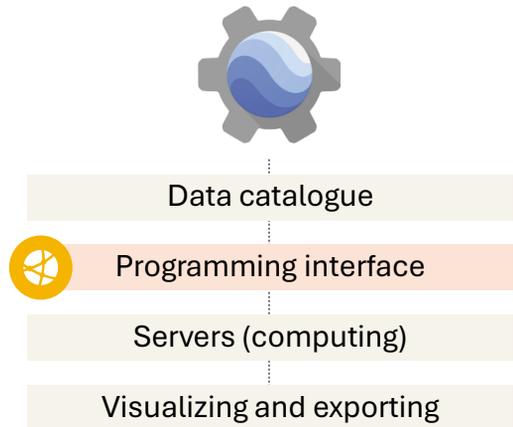


20

Google Earth Engine (GEE)

Importing, processing, analysing, and exporting based on programming languages (Javascript, python, R with *rgee*)

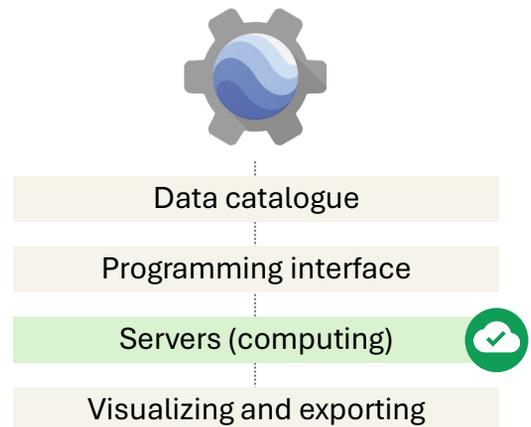
Collaborating, sharing scripts, setting up processing chains



21

Google Earth Engine (GEE)

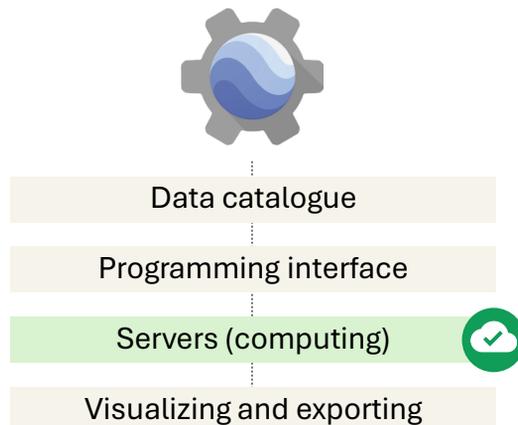
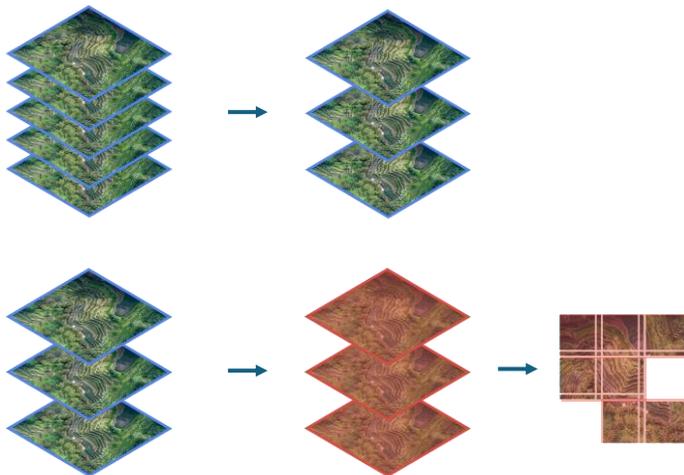
Little data is downloaded to the client unless we seek to visualize data



22

Google Earth Engine Introduction

Google Earth Engine (GEE)



Michael DeWitt & Katie Friis – Geo For Good 2022

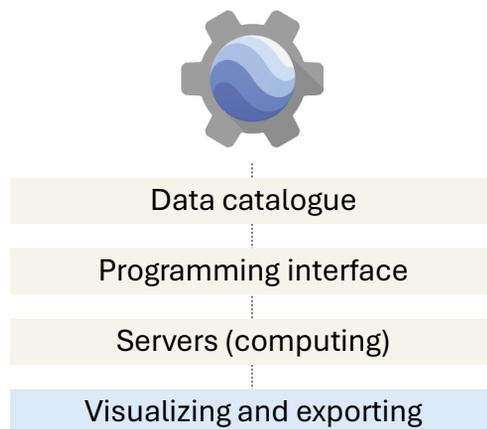
23

Google Earth Engine Introduction

Google Earth Engine (GEE)



Export: geotiff, .shp, tables, ...
In GEE: graph, maps, apps



Michael DeWitt & Katie Friis – Geo For Good 2022

24

Google Earth Engine Introduction

Examples

Combine numerous datasets

Global model

Article

Integrated global assessment of the natural forest carbon potential

<https://doi.org/10.1038/s41586-023-06723-z>

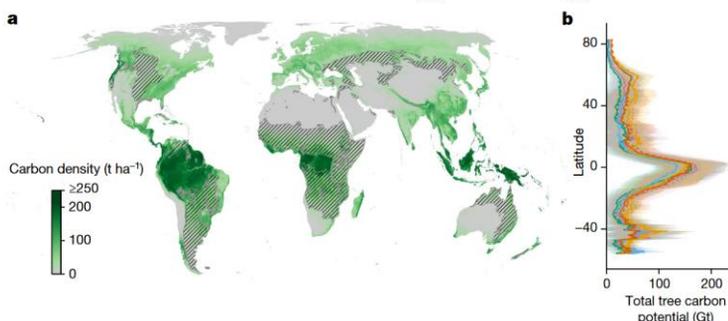
Received: 15 July 2022

Accepted: 6 October 2023

Published online: 13 November 2023

Forests are a substantial terrestrial carbon sink, but anthropogenic changes in land use and climate have considerably reduced the scale of this system¹. Remote-sensing estimates to quantify carbon losses from global forests^{2–5} are characterized by considerable uncertainty and we lack a comprehensive ground-sourced evaluation to benchmark these estimates. Here we combine several ground-sourced⁶ and satellite-

Mo et al. (2023)



25

Google Earth Engine Introduction

Examples

Screenshot of the Google Earth Engine interface. The main map shows a global view of forest cover change. The right sidebar contains the following information:

- Global Forest Change** (published by Hansen, Potapov, Moore, Miettinen et al.)
- University of Maryland**, Department of Geographical Sciences
- Results from time-series analysis of Landsat images characterizing forest extent and change.
- Trees are defined as vegetation taller than 5m in height and are expressed as a percentage per colour grid cell as 2000 Percent Tree Cover. Forest Cover Loss is defined as a stand-replacement disturbance, or a change from a forest to non-forest state, during the period 2000–2023.
- Forest Cover Gain is defined as the increase of trees, or a non-forest to forest change entirely, within the period 2000–2023. Forest Loss Year is a disaggregation of tree Forest Loss to annual time scales.
- Reference 2000 and 2023 imagery are median observations from a set of quality assessment-passed growing season observations.
- To share location copy URL
- Download the data
- ✓ Data Products
- Forest Loss Year (2023 Highlight)
 - 2023
 - 2022
 - 2020
 - No loss
 - Water or no data
- Other Data Layers
- 2000 Percent Tree Cover
- Background Imagery
- Year 2000 Bands 5/2/4
- Example Locations
- Wildfires in Canada

26

Google Earth Engine Getting ready

Creating a GEE account

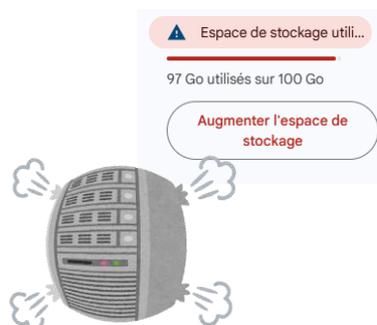
27

Google Earth Engine Getting ready

Exporting data: check available space

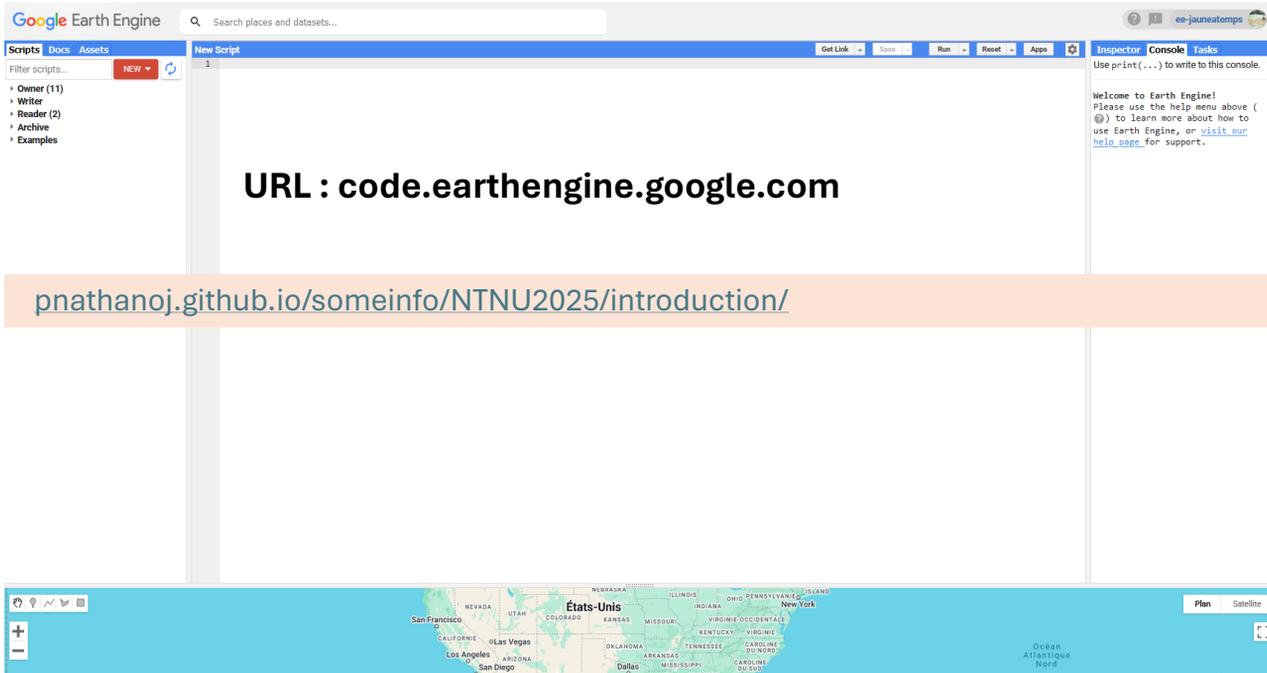
Exported data is found in Google Drive

<https://drive.google.com/drive/quota>



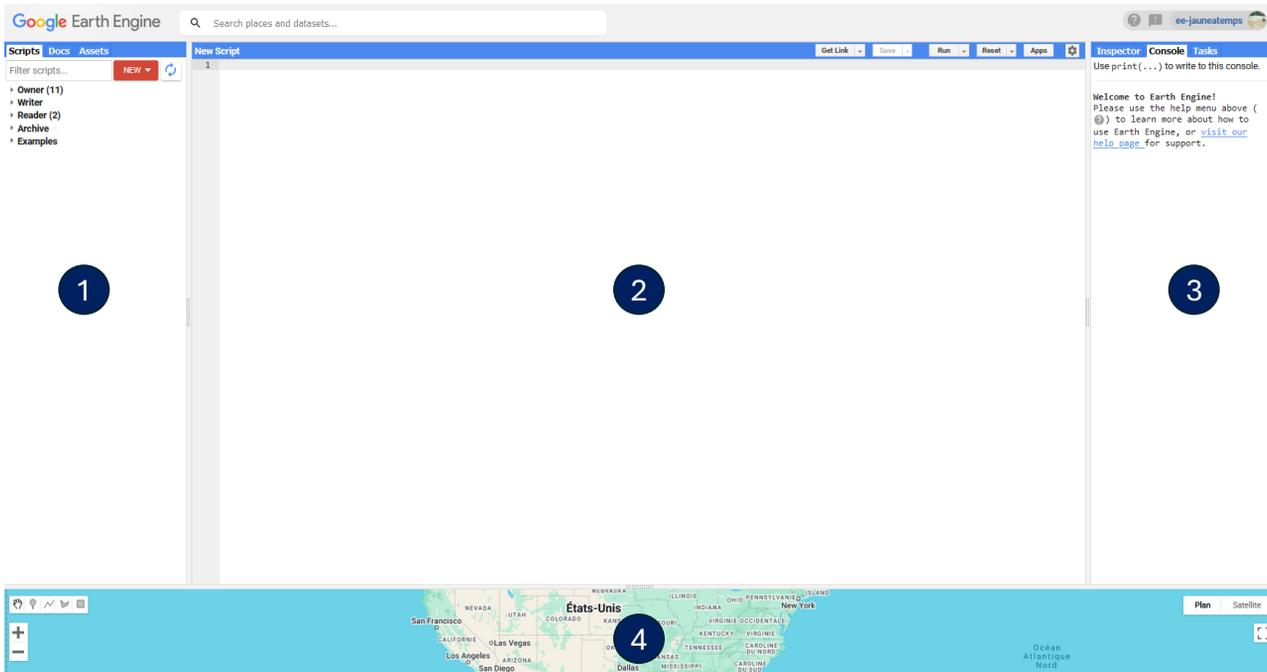
28

Google Earth Engine Getting ready



29

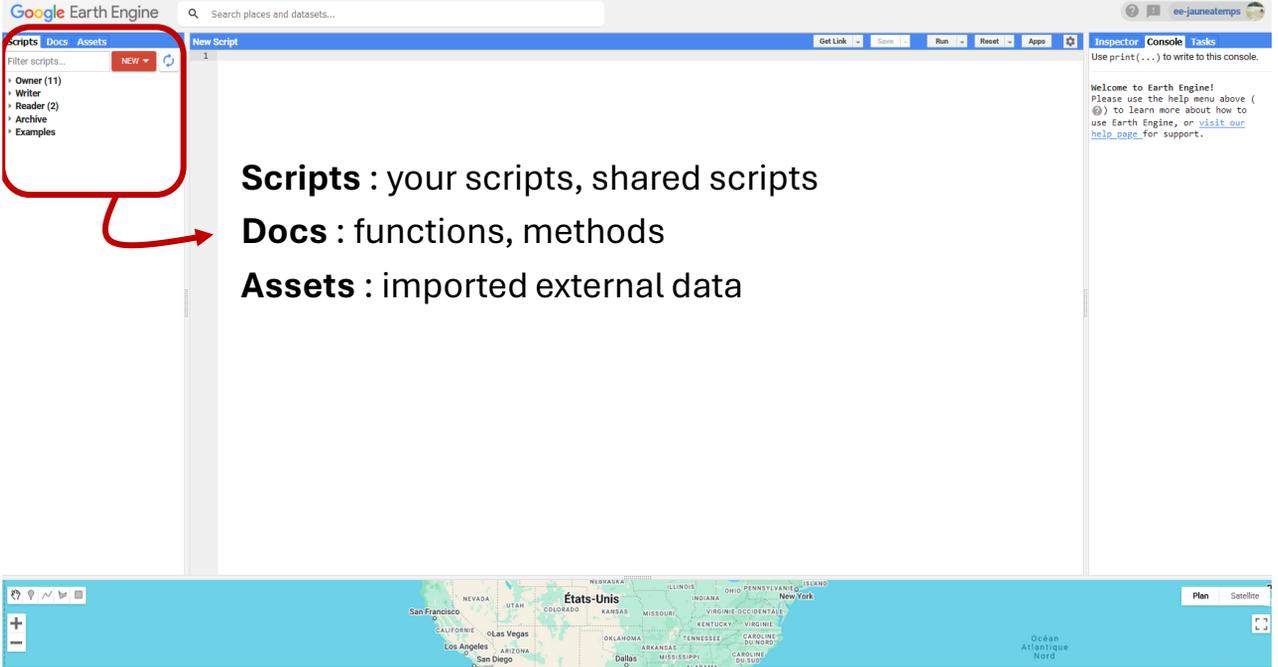
Google Earth Engine Getting ready



30

Google Earth Engine

Getting ready

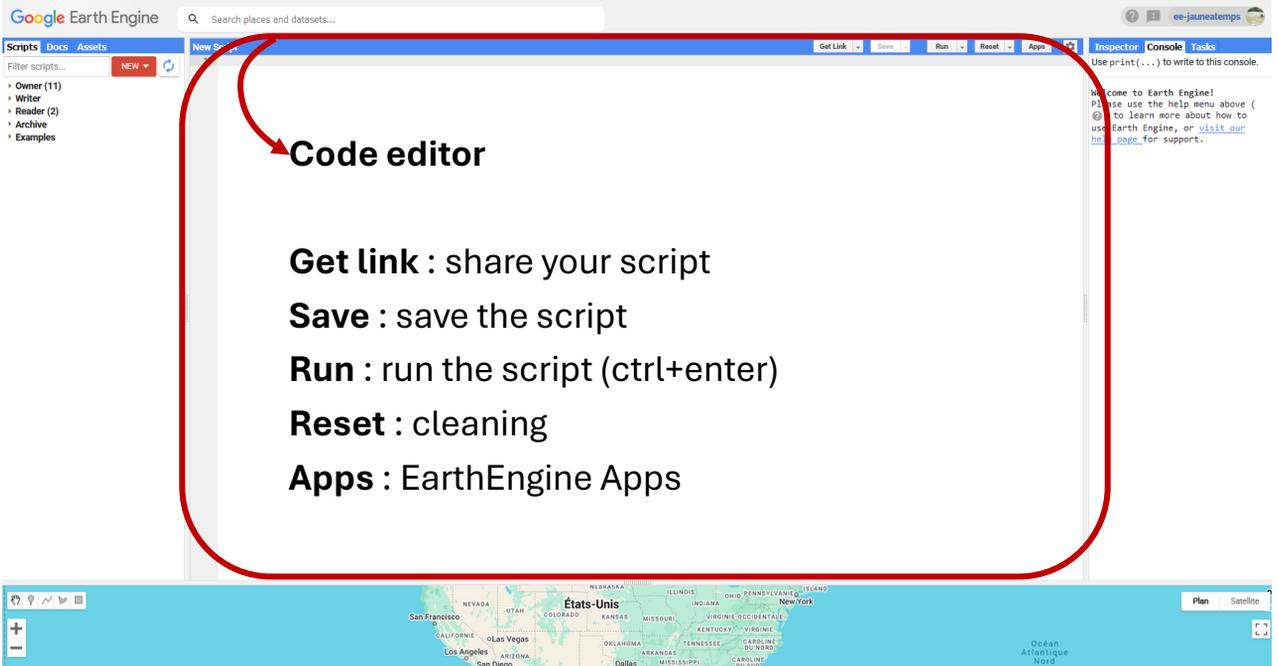


Scripts : your scripts, shared scripts
Docs : functions, methods
Assets : imported external data

31

Google Earth Engine

Getting ready

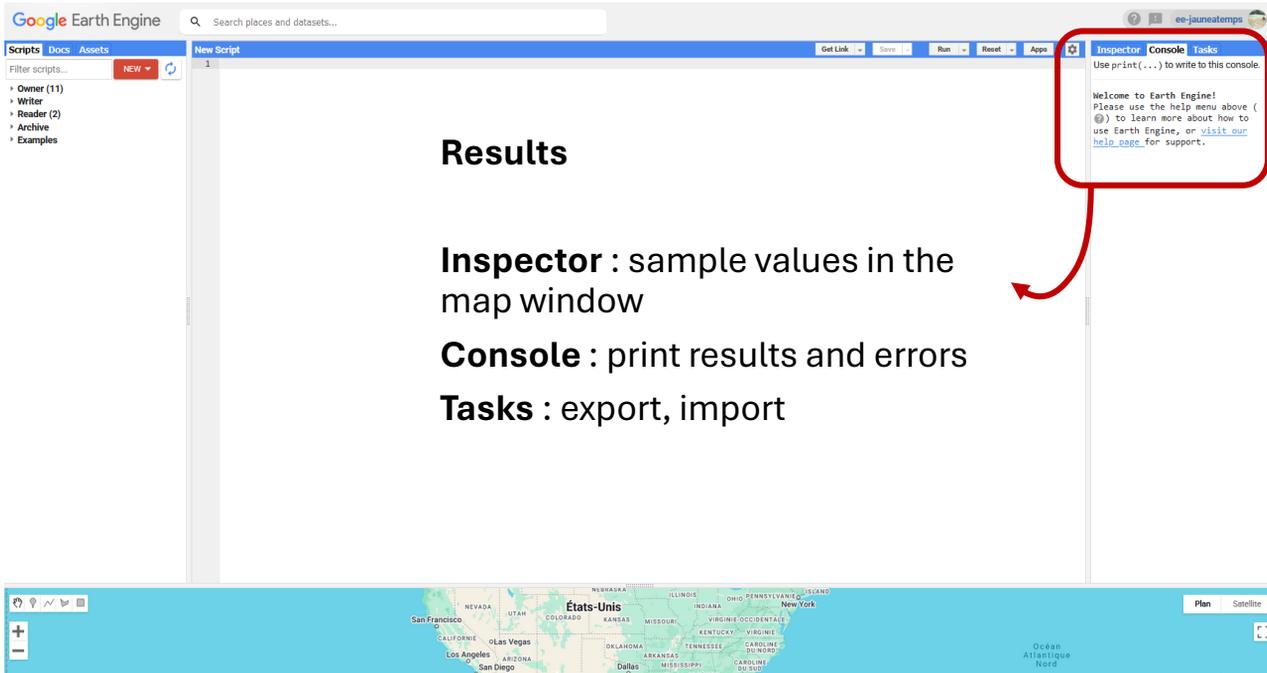


Code editor

Get link : share your script
Save : save the script
Run : run the script (ctrl+enter)
Reset : cleaning
Apps : EarthEngine Apps

32

Google Earth Engine Getting ready



Results

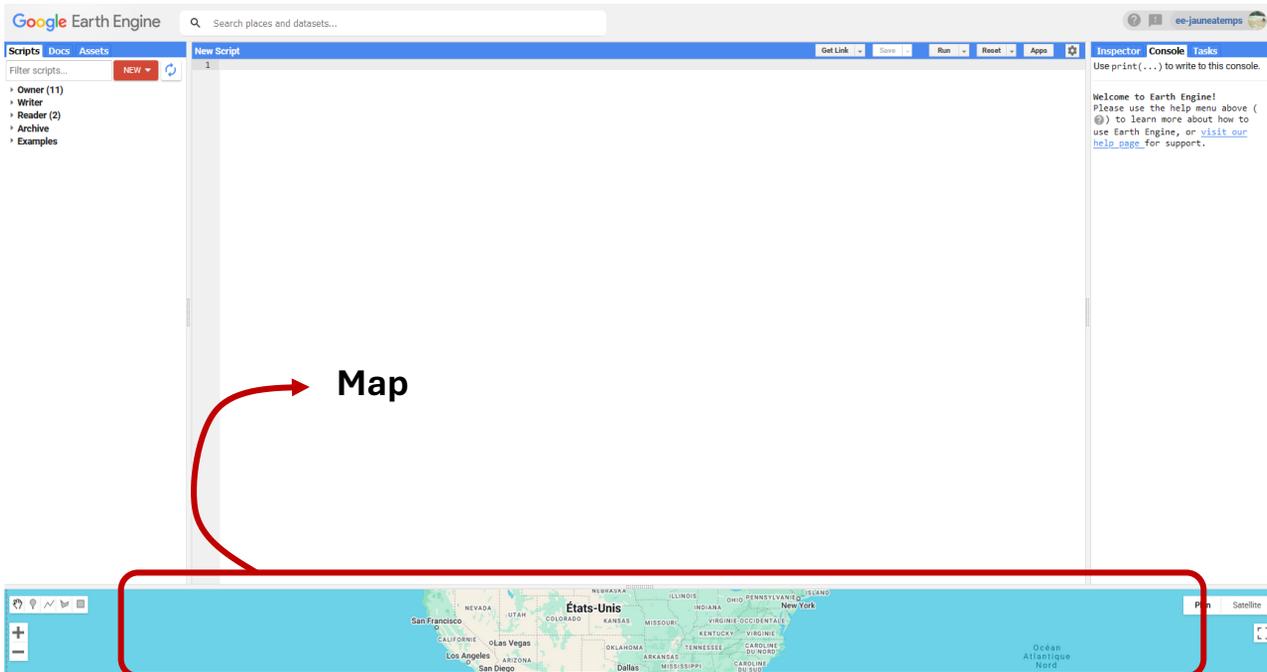
Inspector : sample values in the map window

Console : print results and errors

Tasks : export, import

33

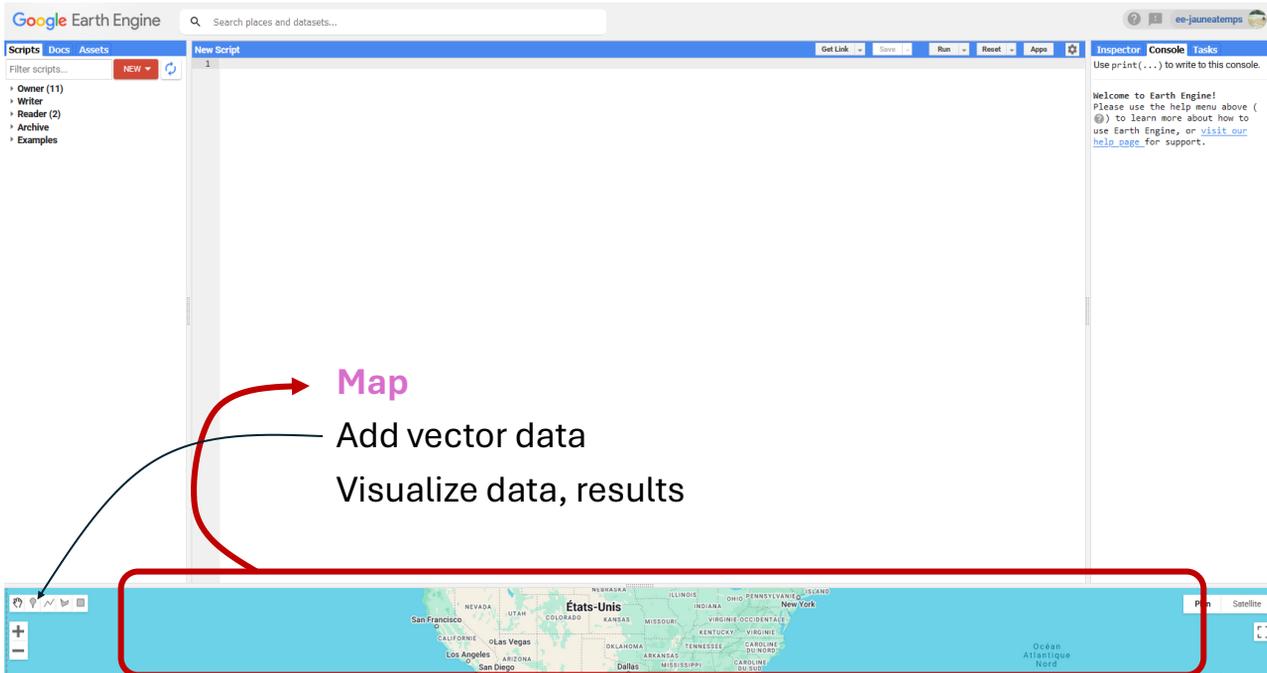
Google Earth Engine Getting ready



Map

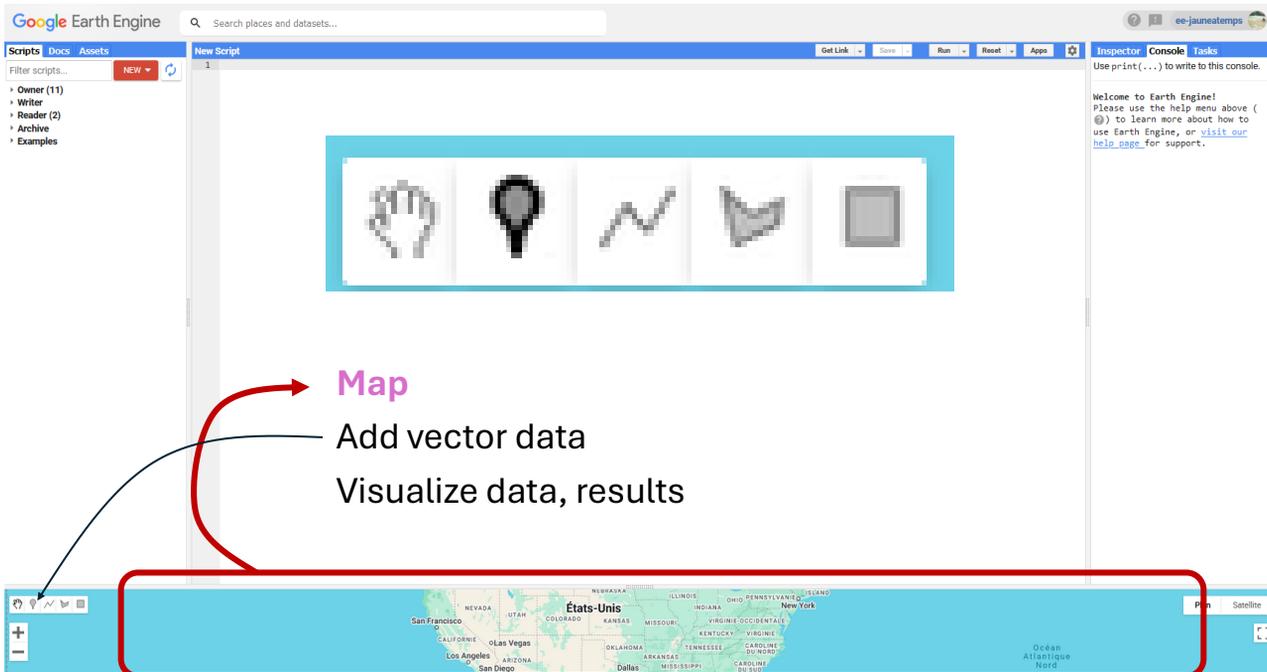
34

Google Earth Engine Getting ready



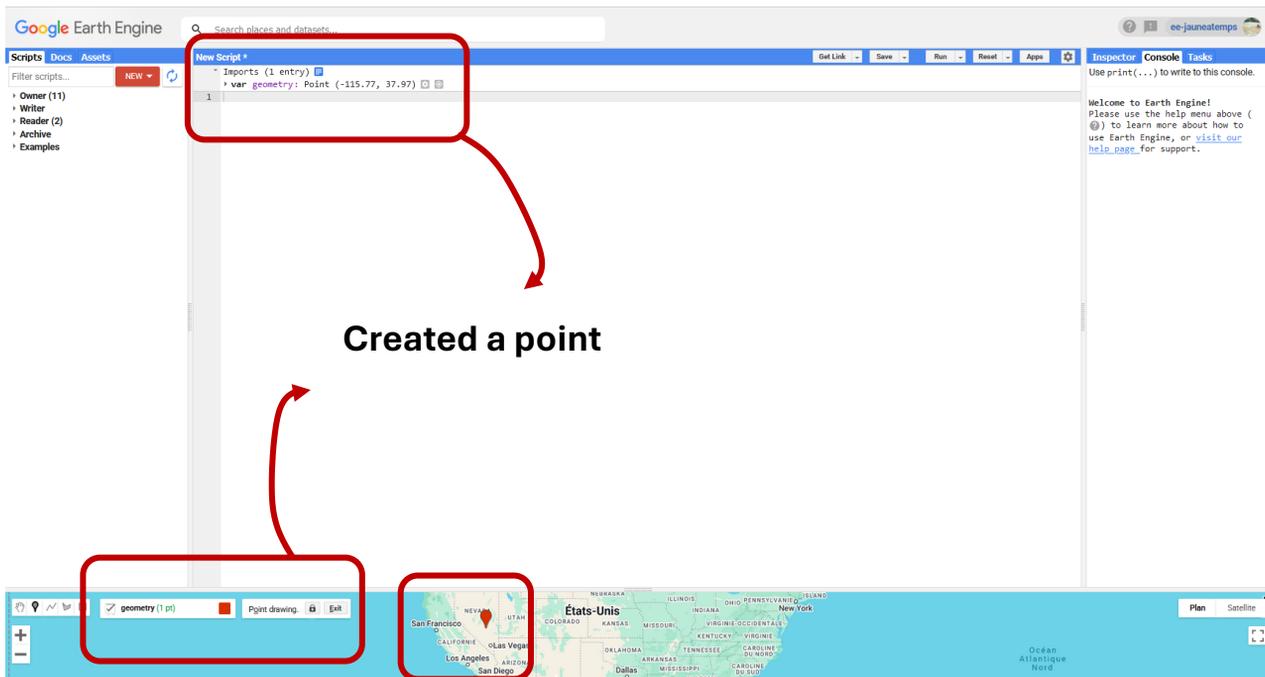
35

Google Earth Engine Getting ready



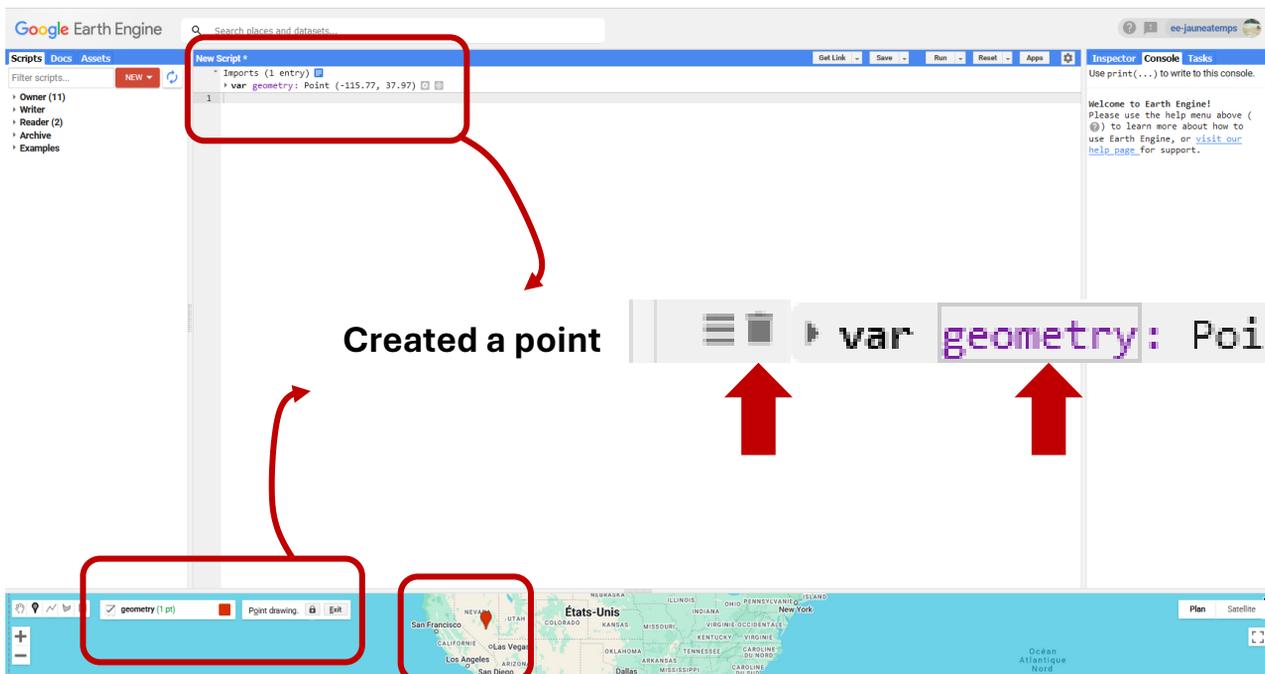
36

Google Earth Engine Getting ready



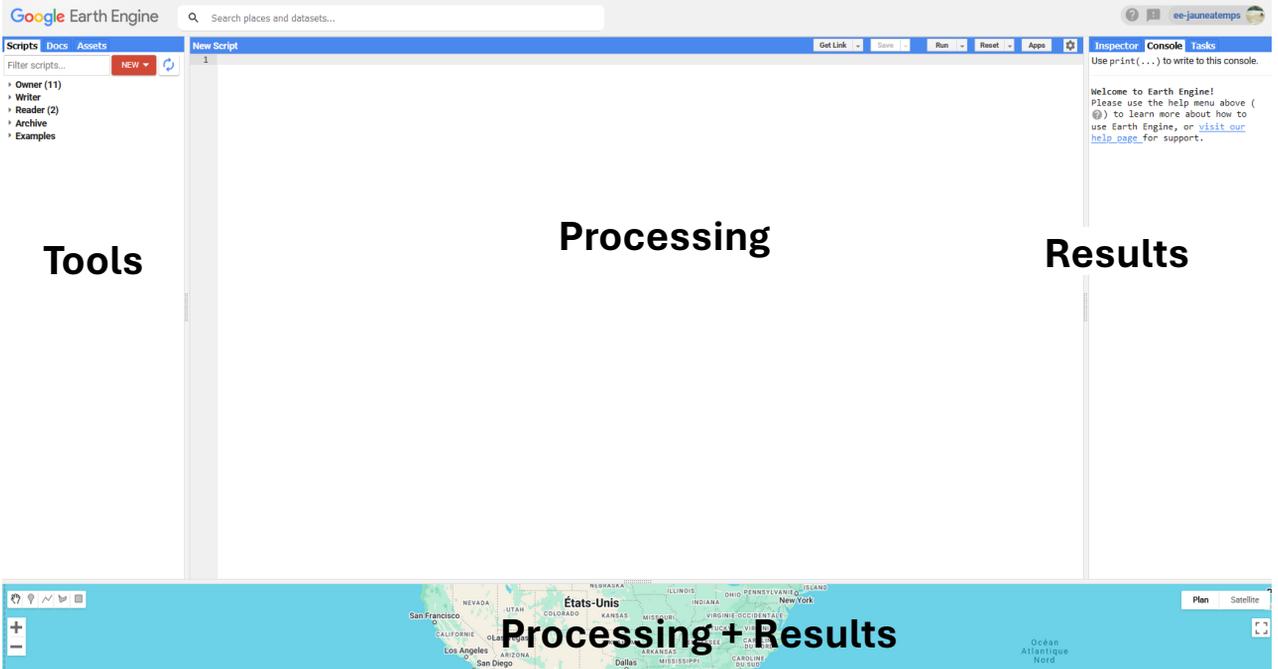
37

Google Earth Engine Getting ready



38

Google Earth Engine Getting ready



39

Google Earth Engine Getting ready

Programming

- 1) Create a folder to hold scripts
- 2) Create a new script
- 3) Test a few commands in Javascript

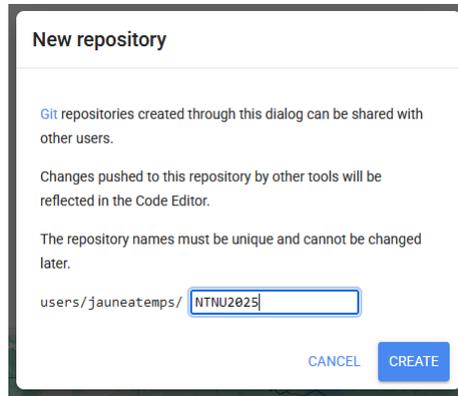
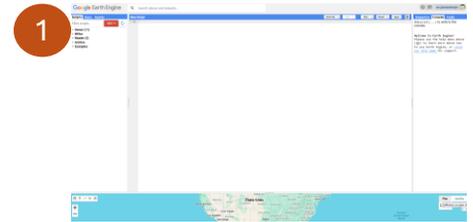
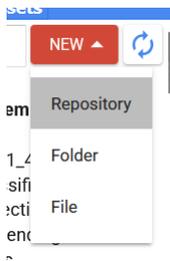
Case studies will be based on the github website that will allow you to copy-paste command lines

40

Google Earth Engine Getting ready

Programming

Create a repository (folder):
Script tab > New > Repository
Name it “NTNU2025”

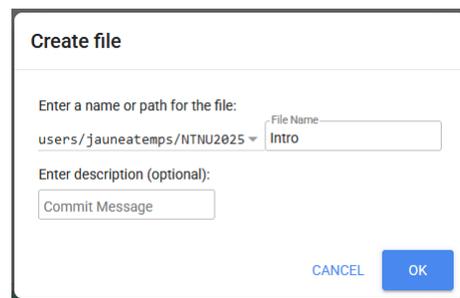
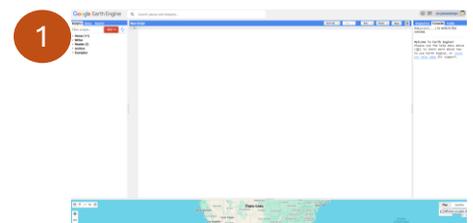
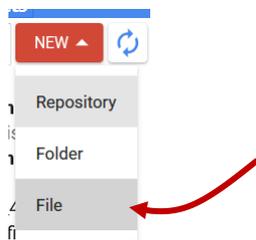


41

Google Earth Engine Getting ready

Programming

Create a new script/file, name it “Intro”
Script tab > New > File



42

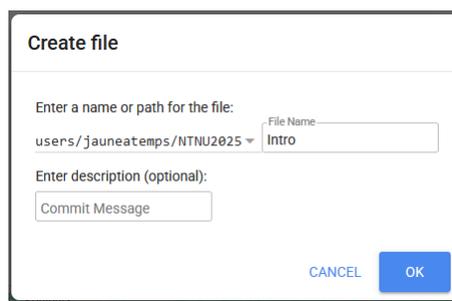
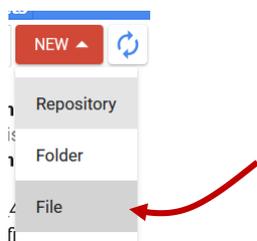
Google Earth Engine Getting ready

Programming

Create a new script/file, name it “Intro”

Script tab > New > File

Click on the new created file to open it!

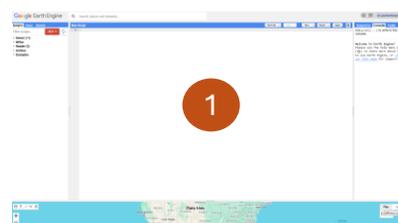


43

Google Earth Engine Getting ready

Programming

Open “Intro”



;

indicates the end of a command line

//

indicates that anything after does not need to be interpreted

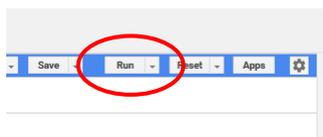
' ' or " "

can both be used

To run the script:

Ctrl+Enter

or



44

Google Earth Engine

Getting ready

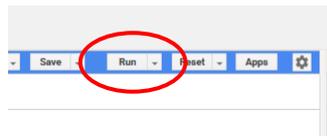
Programming

Write the following

```
// this is a comment  
var a1 = 234; // a1 is created
```

Run the script

The variable a1 was created and the value 234 was affected to this variable.



45

Google Earth Engine

Getting ready

Programming

Write the following

```
// this is a comment  
var a1 = 234; // a1 is created
```

Run the script

The variable a1 was created and the value 234 was affected to this variable.

! Reserved words can't be used to name variables ("var", "max", ...) because they are already used elsewhere (functions, ...).

46

Google Earth Engine Getting ready

Programming

Add the following

```
var a1 = 234;  
var a2 = ee.Number(a1);
```

47

Google Earth Engine Getting ready

Programming

Add the following

```
var a1 = 234; // Number object  
var a2 = ee.Number(a1);
```

We used the function `ee.Number()` to create a new variable `a` that is an object of type “Number”.

“ee.” functions are specific to GEE: GEE combine usual Javascript functions with dedicated functions.

48

Google Earth Engine

Getting ready

Programming

Add the following

```
var a1 = 234; // Number object
var a2 = ee.Number(a1);
```

! Putting the 234 value in ee.Number() allows to create a variable that will be stored in the server (away) and not in the client (here).

49

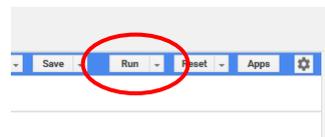
Google Earth Engine

Getting ready

Programming

Add the following

```
print(a1); // print the variable
```



And run the script

```
var a1 = ee.Number(123);
print(a1); // the value of a1 has changed
print(a1, 'the value of a');
```

50

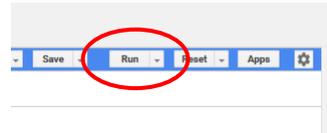
Google Earth Engine

Getting ready

Programming

Add the following

```
print(a); // print the variable
```



And run the script

```
var a = ee.Number(123);
print(a); // the value of a has changed
print(a, 'the value of a');
```

Do not use spaces, special characters (市, é, \$, ...) when naming variables

51

Google Earth Engine

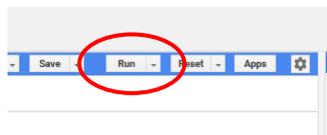
Getting ready

Programming

Add the following

```
var a = ee.Number(234);
var b = ee.Number(1000);
var c = a.add(b); //
print(c);
print(c.subtract(a))
```

And run the script



52

Google Earth Engine

Getting ready

Programming

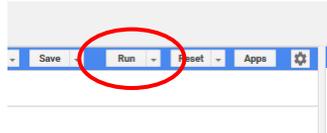
Add the following

```
var a = ee.Number(234);
var b = ee.Number(1000);
var c = a.add(b); //
print(c);
print(c.subtract(a))
```

A command can contain several functions in a row:

```
a.add(b).add(b).add(b)
= a + b + b + b
```

And run the script



53

Google Earth Engine

Getting ready

Programming

Create two variables

var1 = 32

and

var2 = 45.5

Compute and print the value of

var1/var2

Functions

```
+ .add()
- .subtract()
x .multiply()
/ .divide()
```

54

Google Earth Engine

Getting ready

Programming

How about *Strings*?

```
var site_1 = ee.String('Fushan Forest')  
print(site_1);
```

55

Google Earth Engine

Getting ready

Programming

How about *Strings*?

```
var site_1 = ee.String('Fushan Forest')  
print(site_1);
```

pnathanoj.github.io/someinfo/NTNU2025/introduction/

56

Google Earth Engine

Getting ready

Programming

Lists & Arrays: store data in GEE

Write and run the following

```
var a = ee.Number(10);
var b = ee.Number(20);
var c = ee.Number(530);
var List1 = ee.List([a, b, b, a, c]);
var Array1 = ee.Array([a, b, b, a, c]);

print(List1); print(Array1);
```

[and] to contain the elements (values)

57

Google Earth Engine

Getting ready

Programming

Lists & Arrays: store data in GEE

Possible to create list of lists, array of lists

```
var List2 = ee.List([[a, b, b, a, c], [a, c, c, c, a]]);
var Array2 = ee.Array([[a, b, b, a, c], [a, c, c, c, a]]);
```

58

Google Earth Engine Getting ready

Programming

Lists & Arrays: store data in GEE

Arrays and lists are used differently

```
var List2 = ee.List([[a, b, b, a, c], [a, c, c]]);
var Array2 = ee.Array([[a, b, b, a, c], [a, c, c]]);
```

All elements should have the same length in an Array

An array will only contain numerical values

```
var Array3 = ee.Array([[a, b, b, a, c], [a, c, c, c, 'yes']]);
```

59

Google Earth Engine Getting ready

Programming

Lists & Arrays: store data in GEE

Arrays and lists are used differently

An array can therefore contain a matrix: each column being a different list in the array, all columns have the same number of elements

	0	1	2	3	4
0	1.20	0.20	3.45	-2.1	3.22
1	-0.30	-8.1	2.0	40.3	18.0
2	0	3.4281	-10.0	3.21	5.0
3	7.39	8.01	-0.47	0.18	0.26
4	-3.0	1.20	-0.4	-0.1	0.56

60

Google Earth Engine

Getting ready

Programming

Lists & Arrays: store data in GEE

Arrays and lists are used differently

Lists can be modified to add elements, remove some, etc.

Arrays can be used to change values, filter values, etc.

```
var Array1 = ee.Array([[a, b, b, a, c], [a, c, c, c, a]]);
var Array3 = Array1.gte([[1, 2, 3, 11, 40],[10, 15, 45, 0, 9]]);
print(Array3);
```

function **gte** (greater or equal) : the new value = 1 if true, 0 if false.

61

Google Earth Engine

Getting ready

Programming

Lists & Arrays: store data in GEE

Arrays and lists are used differently

Lists can be modified to add elements, remove some, etc.

Arrays can be used to change values, filter values, etc.

```
var Array1 = ee.Array([[a, b, b, a, c], [a, c, c, c, a]]);
var Array3 = Array1.gte([[1, 2, 3, 11, 40],[10, 15, 45, 0, 9]]);
print(Array3);
```

function **gte** (greater or equal) : the new value = 1 if true, 0 if false.

62

Google Earth Engine Getting ready

Programming

Lists & Arrays: store data in GEE

Arrays and lists are used differently

Lists can be modified to add elements, remove some, etc.

Arrays can be used to change values, filter values, etc.

Used for linear models, matrices.

63

Google Earth Engine Getting ready

Programming

To summarize:

```
ee.Number()  
ee.String()  
print(variable, 'text')  
.add()  
.subtract()  
.multiply()  
.divide()  
ee.List()  
ee.Array()
```

64

Spatial data in GEE

Rasters

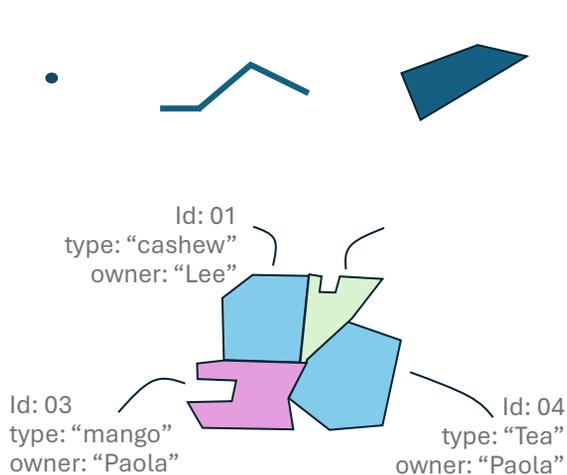
Resolution: size of the pixel

Extent: number of pixels

...

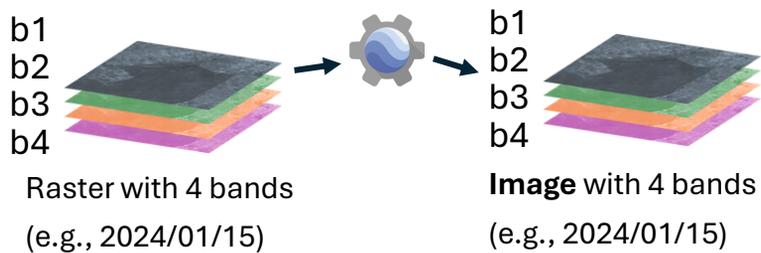


Vectors



65

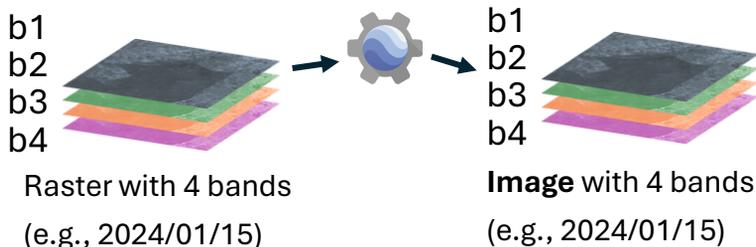
Spatial data in GEE

Raster > **Image** or **ImageCollection**

66

Spatial data in GEE

Raster > Image or ImageCollection



the Image can have several properties (~metadata)

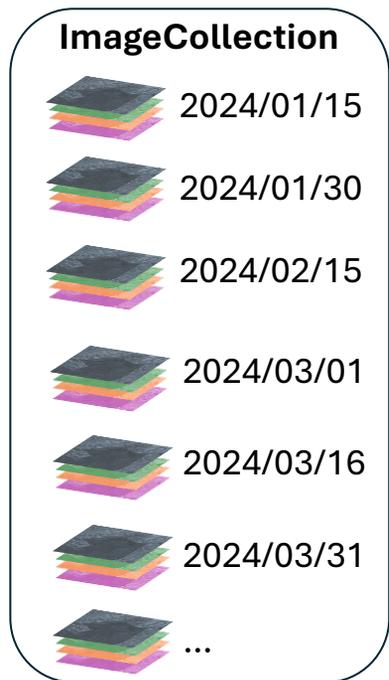
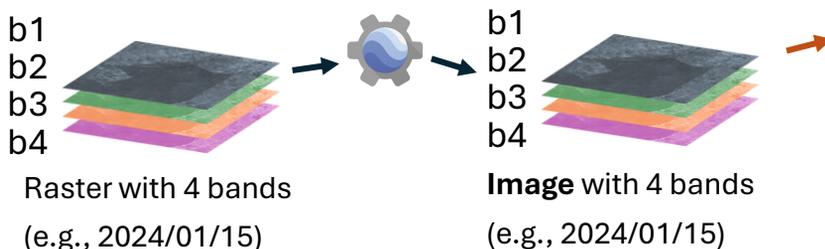
```

Inspector Console Tasks
▼ Image LANDSAT/LC08/C02/T1_L2/LC08
  type: Image
  id: LANDSAT/LC08/C02/T1_L2/LC08
  version: 1629837149647415
  ▼ bands: List (19 elements)
    ▶ 0: "SR_B1", unsigned int16,
    ▶ 1: "SR_B2", unsigned int16,
    ▶ 2: "SR_B3", unsigned int16,
    ▶ 3: "SR_B4", unsigned int16,
    ▶ 4: "SR_B5", unsigned int16,
    ▶ 5: "SR_B6", unsigned int16,
    ▶ 6: "SR_B7", unsigned int16,
    ▶ 7: "SR_QA_AEROSOL", unsigned
    ▶ 8: "ST_B10", unsigned int16,
    ▶ 9: "ST_ATRAN", signed int16,
    ▶ 10: "ST_CDIST", signed int16
    ▶ 11: "ST_DRAD", signed int16,
    ▶ 12: "ST_EMIS", signed int16,
    ▶ 13: "ST_EMSD", signed int16,
    ▶ 14: "ST_QA", signed int16, E
    ▶ 15: "ST_TRAD", signed int16,
    ▶ 16: "ST_URAD", signed int16,
    ▶ 17: "QA_PIXEL", unsigned int
    ▶ 18: "QA_RADSAT", unsigned in
  ▼ properties: Object (92 propert
    ALGORITHM_SOURCE_SURFACE_REF
    ALGORITHM_SOURCE_SURFACE_TEM
    CLOUD_COVER: 23.83
    CLOUD_COVER_LAND: 0
    COLLECTION_CATEGORY: T1
    COLLECTION_NUMBER: 2
    DATA_SOURCE_AIR_TEMPERATURE:
    DATA_SOURCE_ELEVATION: 61520
  
```

67

Spatial data in GEE

Raster > Image or ImageCollection



68

Spatial data in GEE

Raster > **Image** or **ImageCollection**

WHRC Pantropical National Level Carbon Stock Dataset



b1: biomass / hectare

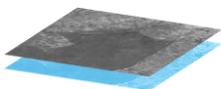
Image with 1 band

69

Spatial data in GEE

Raster > **Image** or **ImageCollection**

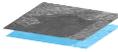
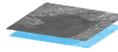
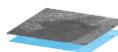
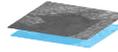
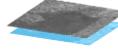
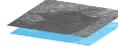
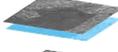
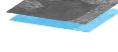
OpenET SIMS Monthly Evapotranspiration v2.0



b1: evapotranspiration
b2: number of clouds

Image with 2 bands
(monthly)
(e.g., 2023/01)

ImageCollection

-  2022/01
-  2022/02
-  2022/03
-  2022/04
-  2022/05
-  2022/06
-  2022/07
-  2022/08

... 2008-2023

Lot of data !

70

Spatial data in GEE

Vectors > **Feature** or **FeatureCollection**

id	name	date	agb
1	'site1'	'01-25'	52.6



id	name	date	agb
1	'site1'	'01-25'	52.6

Vector with 4 attributes
(e.g., sample sites)

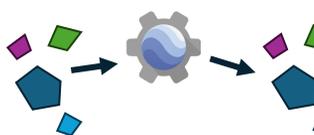
Feature with 4 attributes
called “**property**”

71

Spatial data in GEE

Vectors > **Feature** or **FeatureCollection**

id	name	date	agb
1	'site1'	'01-25'	52.6
2	'site2'	'01-28'	20.0
3	'site3'	'01-26'	38.4
4	'site4'	'01-25'	49.7
5	'site1'	'03-15'	51.8
6	'site2'	'03-15'	20.0



id	name	date	agb
1	'site1'	'01-25'	52.6
2	'site2'	'01-28'	20.0
3	'site3'	'01-26'	38.4
4	'site4'	'01-25'	49.7
5	'site1'	'03-15'	51.8
6	'site2'	'03-15'	20.0

Vector with 4 attributes
and 6 polygons
(e.g., sampling sites)

FeatureCollection with 4
properties and 6 features

72

Google Earth Engine Getting ready

Spatial data in GEE

Collections can be filtered based on their properties, and the Image or Feature location.

GEE functions do not work similarly on Image, Feature, ImageCollection, and FeatureCollection!

73

Google Earth Engine Getting ready

Data: 3 sources

Earth Engine Catalog

> 90 petabytes

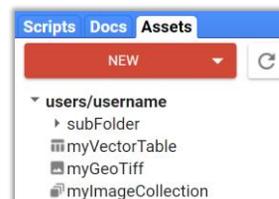
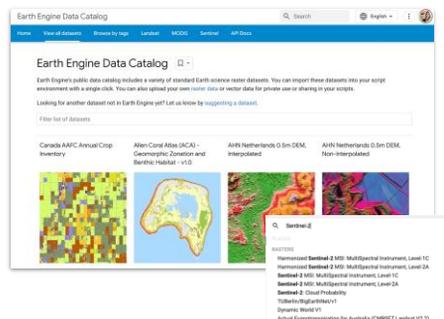
Managed by Google

Earth Engine Community Catalog

Earth Engine Assets

Our data (private, shared)

Limited space



Simon Ilyushchenko, Peter Davis, Sam Roy – Geo for Good 2024

74

Google Earth Engine Getting ready

Data: 3 sources

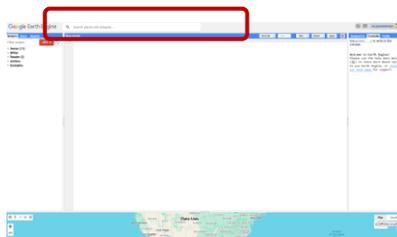
Earth Engine Catalog

> 90 petabytes

Managed by Google

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

or:



75

Google Earth Engine Getting ready

Data: 3 sources

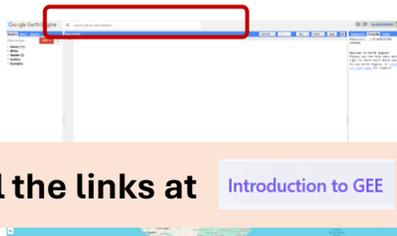
Earth Engine Catalog

> 90 petabytes

Managed by Google

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

or:



Open all the links at [Introduction to GEE](#)

76

Data: 3 sources

Home View all datasets Browse by tags Landsat MODIS Sentinel Publisher Community API Docs

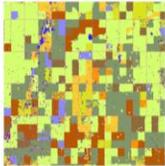
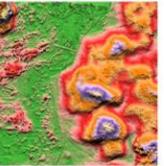
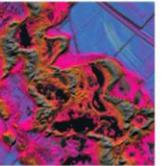
Earth Engine Data Catalog

Earth Engine's public data catalog includes a variety of standard Earth science raster datasets. You can import these datasets into your script environment 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](#).

search →

datasets →

Canada AAFC Annual Crop Inventory	Allen Coral Atlas (ACA) - Geomorphic Zonation and Benthic Habitat - v2.0	AHN Netherlands 0.5m DEM, Interpolated	AHN Netherlands 0.5m DEM, Non-Interpolated	AHN Netherlands 0.5m DEM, Raw Samples
				
Starting in 2009, the Earth Observation Team of the Science and Technology Branch (STB) at Agriculture and Agri-Food Canada (AAFC) began the process of generating annual crop type digital maps. Focusing on the Prairie Provinces in 2009 and 2010, a Decision Tree (DT) based methodology ...	The Allen Coral Atlas dataset maps the geomorphic zonation and benthic habitat for the world's shallow coral reefs at 5 m pixel resolution. Also included is a global reef extent product that maps additional reef areas unable to be explicitly included in the geomorphic and ...	The AHN DEM is a 0.5m DEM covering the Netherlands. It was generated from LIDAR data taken in the spring between 2007 and 2012. It contains ground level samples with all other items above ground (such as buildings, bridges, trees etc.) removed. This version is ...	The AHN DEM is a 0.5m DEM covering the Netherlands. It was generated from LIDAR data taken in the spring between 2007 and 2012. It contains ground level samples with all other items above ground (such as buildings, bridges, trees etc.) removed. This version is ...	The AHN DEM is a 0.5m DEM covering the Netherlands. It was generated from LIDAR data taken in the spring between 2007 and 2012. This version contains both ground level samples and items above ground level (such as buildings, bridges, trees etc.) The point cloud ...
canada crop landcover	ocean sentinel2-derived	ahn dem elevation geophysical lidar netherlands	ahn dem elevation geophysical lidar netherlands	ahn dem elevation geophysical lidar netherlands

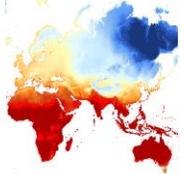
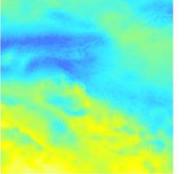
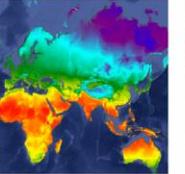
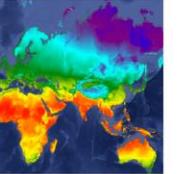
keywords →

77

Data: 3 sources

“Worldclim”

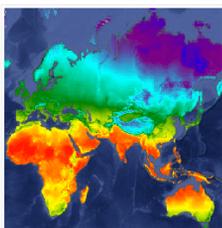
worldclim

TerraClimate: Monthly Climate and Climatic Water Balance for Global Terrestrial Surfaces, University of ...	OpenLandMap Precipitation Monthly	WorldClim BIO Variables V1	WorldClim Climatology V1
			
TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces. It uses climatically aided interpolation, combining high-spatial resolution climatological normals from the WorldClim dataset, with coarser spatial resolution, but time-varying data from CRU Ts4.0 and the Japanese 55-year Reanalysis (JRA55). ...	Monthly precipitation in mm at 1 km resolution based on SM2RAIN-ASCAT 2007-2018, IMERG, CHELSA Climate, and WorldClim. Downscaled to 1 km resolution using g0slwarp (cubic splines) and an average between WorldClim, CHELSA Climate, and IMERG monthly product (see, e.g., '3B-MO-LGIS-IMERG.20180601.V05B.tif'). 3x higher weight is given ...	WorldClim V1 Bioclim provides bioclimatic variables that are derived from the monthly temperature and rainfall in order to generate more biologically meaningful values. The bioclimatic variables represent annual trends (e.g., mean annual temperature, annual precipitation), seasonality (e.g., annual range in temperature and precipitation), and extreme ...	WorldClim version 1 has average monthly global climate data for minimum, mean, and maximum temperature and for precipitation. WorldClim version 1 was developed by Robert J. Hijmans, Susan Cameron, and Juan Paris, at the Museum of Vertebrate Zoology, University of California, Berkeley, in collaboration with ...
climate drought evapotranspiration geophysical global merced	enviromatrix imerg monthly openeohub openlandmap precipitation	berkeley climate monthly precipitation temperature weather	berkeley climate monthly precipitation temperature weather

78

Google Earth Engine Getting ready

Data: 3 sources



Dataset Availability
1960-01-01T00:00:00Z–1991-01-01T00:00:00Z

Dataset Provider
[University of California, Berkeley](#)

Earth Engine Snippet
`ee.Image("WORLDCLIM/V1/BIO")`

Tags

- berkeley
- climate
- monthly
- precipitation
- temperature
- weather
- worldclim
- bioclim
- coldest
- diurnal
- driest
- isothermality
- seasonality
- warmest
- wettest

Time period

producer

code to import in script

Description Bands Terms of Use Citations

WorldClim V1 Bioclim provides bioclimatic variables that are derived from the monthly temperature and rainfall in order to generate more biologically meaningful values.

The bioclimatic variables represent annual trends (e.g., mean annual temperature, annual precipitation), seasonality (e.g., annual range in temperature and precipitation), and extreme or limiting environmental factors (e.g., temperature of the coldest and warmest month, and precipitation of the wet and dry quarters).

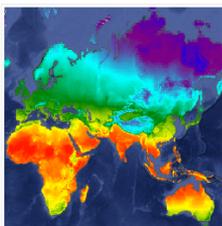
The bands scheme follows that of ANUCLIM, except that for temperature seasonality the standard deviation was used because a coefficient of variation does not make sense with temperatures between -1 and 1.

WorldClim version 1 was developed by Robert J. Hijmans, Susan Cameron, and Juan Parra, at the Museum of Vertebrate Zoology, University of California, Berkeley, in collaboration with Peter Jones and Andrew Jarvis (CIAT), and with Karen Richardson (Rainforest CRC).

79

Google Earth Engine Getting ready

Data: 3 sources



Dataset Availability
1960-01-01T00:00:00Z–1991-01-01T00:00:00Z

Dataset Provider
[University of California, Berkeley](#)

Earth Engine Snippet
`ee.Image("WORLDCLIM/V1/BIO")`

Tags

- berkeley
- climate
- monthly
- precipitation
- temperature
- weather
- worldclim
- bioclim
- coldest
- diurnal
- driest
- isothermality
- seasonality
- warmest
- wettest

Time period

producer

code to import in script

Description Bands Terms of Use Citations

WorldClim V1 Bioclim provides bioclimatic variables that are derived from the monthly temperature and rainfall in order to generate more biologically meaningful values.

The bioclimatic variables represent annual trends (e.g., mean annual temperature, annual precipitation), seasonality (e.g., annual range in temperature and precipitation), and extreme or limiting environmental factors (e.g., temperature of the coldest and warmest month, and precipitation of the wet and dry quarters).

The bands scheme follows that of ANUCLIM, except that for temperature seasonality the standard deviation was used because a coefficient of variation does not make sense with temperatures between -1 and 1.

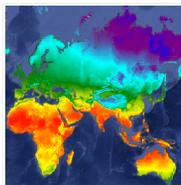
WorldClim version 1 was developed by Robert J. Hijmans, Susan Cameron, and Juan Parra, at the Museum of Vertebrate Zoology, University of California, Berkeley, in collaboration with Peter Jones and Andrew Jarvis (CIAT), and with Karen Richardson (Rainforest CRC).

80

Google Earth Engine Getting ready

Data: 3 sourc

“Worldclim”



Dataset Availability
1960-01-01T00:00:00Z–1991-01-01T00:00:00Z

Dataset Provider
University of California, Berkeley

Earth Engine Snippet
`ee.Image("WORLDCLIM/V1/BIO")`

Tags

berkeley climate monthly precipitation temperature weather worldclim bioclim
coldest diurnal driest isothermality seasonality warmest wettest

Description Bands Terms of Use Citations

Resolution
927.67 meters ← spatial resolution (“pixel size”, “scale”)

Bands

Name	Units	Min	Max	Scale	Description
bio01	°C	-29*	32*	0.1	Annual mean temperature
bio02	°C	0.9*	21.4*	0.1	Mean diurnal range (mean of monthly (max temp - min temp))
bio03	%	7*	96*		Isothermality (bio02/bio07 * 100)
bio04	°C	0.62*	227.21*	0.01	Temperature seasonality (Standard deviation * 100)
bio05	°C	-9.6*	49*	0.1	Max temperature of warmest month
bio06	°C	-57.3*	25.8*	0.1	Min temperature of coldest month
bio07	°C	5.3*	72.5*	0.1	Temperature annual range (bio05-bio06)

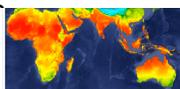
band names | units | min-max | conversion factor (x Scale)

81

Google Earth Engine Getting ready

Data: 3

“World



Earth Engine Snippet
`ee.Image("WORLDCLIM/V1/BIO")`

Description Bands Terms of Use Citations

WorldClim V1 Bioclim provides bioclimatic variables that are derived from the monthly temperature and rainfall in order to generate more biologically meaningful values.

The bioclimatic variables represent annual trends (e.g., mean annual temperature, annual precipitation), seasonality (e.g., annual range in temperature and precipitation), and extreme or limiting environmental factors (e.g., temperature of the coldest and warmest month, and precipitation of the wet and dry quarters).

The bands scheme follows that of ANUCLIM, except that for temperature seasonality the standard deviation was used because a coefficient of variation does not make sense with temperatures between -1 and 1.

WorldClim version 1 was developed by Robert J. Hijmans, Susan Cameron, and Juan Parra, at the Museum of Vertebrate Zoology, University of California, Berkeley, in collaboration with Peter Jones and Andrew Jarvis (CIAT), and with Karen Richardson (Rainforest CRC).

Explore with Earth Engine

```
Code Editor (JavaScript)
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');
```

82

Google Earth Engine

Getting ready

Data: 3 sources

“Worldclim”

paste this in the
code editor

Code Editor (JavaScript)

```

var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');

```

83

Google Earth Engine

Getting ready

Code Editor (JavaScript)

```

var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');

```

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
```

Import the Image WorldClim Bio in a new variable called “dataset”

84

Google Earth Engine

Getting ready

Code Editor (JavaScript)

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');
```

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
```

```
var annualMeanTemperature=dataset.select('bio01').multiply(0.1);
```

The band bio01 of dataset is selected and multiplied by 0.1 (the scale factor), according to the producer's instruction. This is a new variable called annualMeanTemperature.

85

Google Earth Engine

Getting ready

Code Editor (JavaScript)

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');
```

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
```

```
var annualMeanTemperature=dataset.select('bio01').multiply(0.1);
```

```
print(annualMeanTemperature);
```

We can look at the variable annualMeanTemperature.

86

Google Earth Engine

Getting ready

Code Editor (JavaScript)

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');
```

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature=dataset.select('bio01').multiply(0.1);
var visParams = {min: -23, max: 30, palette: ['blue', ...
'yellow', 'red'],};
```

To visualize an Image, it is often better to set a color scale with minimum and maximum values. Between {}: a dictionary that states that the argument min = -23, etc.

87

Google Earth Engine

Getting ready

Code Editor (JavaScript)

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');
```

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature=dataset.select('bio01').multiply(0.1);
var visParams = {min: -23, max: 30, palette: ['blue', ...
'yellow', 'red'],};
Map.setCenter(71.7, 52.4, 3);
```

This is not a new variable! We send an instruction to the Map, to focus to the 71.7 and 52.4 coordinates with a zoom of 3.

88

Functions of GEE are described in the apidocs!

Usage	Returns
<code>Map.setCenter(lon, lat, zoom)</code>	ui.Map

Argument	Type	Details
<code>lon</code>	Number	The longitude of the center, in degrees.
<code>lat</code>	Number	The latitude of the center, in degrees.
<code>zoom</code>	Number, optional	The zoom level, from 0 to 24.

```
Map.setCenter(71.7, 52.4, 3);
```

This is not a new variable! We send an instruction to the Map, to focus to the 71.7 and 52.4 coordinates with a zoom of 3.

89

Functions of GEE are described in the apidocs!

Usage	Returns
<code>Map.setCenter(lon, lat, zoom)</code>	ui.Map

Argument	Type	Details
<code>lon</code>	Number	The longitude of the center, in degrees.
<code>lat</code>	Number	The latitude of the center, in degrees.
<code>zoom</code>	Number, optional	The zoom level, from 0 to 24.

```
Map.setCenter(71.7, 52.4, 3);
```

This is not a new variable! We send an instruction to the Map, to focus to the 71.7 and 52.4 coordinates with a zoom of 3.

90

Google Earth Engine

Getting ready

Code Editor (JavaScript)

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = dataset.select('bio01').multiply(0.1);
var visParams = {
  min: -23,
  max: 30,
  palette: ['blue', 'purple', 'cyan', 'green', 'yellow', 'red'],
};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean Temperature');
```

```
var dataset = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature=dataset.select('bio01').multiply(0.1);
var visParams = {min: -23, max: 30, palette: ['blue', ...
'yellow', 'red'],};
Map.setCenter(71.7, 52.4, 3);
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean
Temperature');
```

91

Google Earth Engine

Getting ready

Usage		Returns
<code>Map.addLayer(eeObject, visParams, name, shown, opacity)</code>		<code>ui.Map.Layer</code>
Argument	Type	Details
<code>eeObject</code>	<code>Collection Feature Image RawMapId</code>	The object to add to the map.
<code>visParams</code>	<code>FeatureVisualizationParameters ImageVisualizationParameters, optional</code>	The visualization parameters. For Images and ImageCollection, see <code>ee.data.getMapId</code> for valid parameters. For Features and FeatureCollections, the only supported key is "color", as a CSS 3.0 color string or a hex string in "RRGGBB" format. Ignored when <code>eeObject</code> is a map ID.
<code>name</code>	<code>String, optional</code>	The name of the layer. Defaults to "Layer N".
<code>shown</code>	<code>Boolean, optional</code>	A flag indicating whether the layer should be on by default.
<code>opacity</code>	<code>Number, optional</code>	The layer's opacity represented as a number between 0 and 1. Defaults to 1.

```
Map.addLayer(annualMeanTemperature, visParams, 'Annual Mean
Temperature');
```

92

```

Map.addLayer(annualMeanTemperature, visParams, 'Name'); ✓
Map.addLayer(annualMeanTemperature, visParams, 'Name', false); ✓
Map.addLayer(annualMeanTemperature, visParams, 'Name', 0.5); ✗
Map.addLayer(annualMeanTemperature, visParams, 'Name', false, 0.5); ✓
Map.addLayer(annualMeanTemperature, {}, 'Name', false, 0.5); ✓

```

`Map.addLayer(eeObject, visParams, name, shown, opacity)`

Argument	Type	Details
<code>eeObject</code>	Collection Feature Image RawMapId	The object to add
<code>visParams</code>	<i>FeatureVisualizationParameters ImageVisualizationParameters, optional</i>	The visualization parameters. For a color string or a layer
<code>name</code>	<i>String, optional</i>	The name of the layer
<code>shown</code>	<i>Boolean, optional</i>	A flag indicating whether the layer is shown
<code>opacity</code>	<i>Number, optional</i>	The layer's opacity

93

```

Map.addLayer(annualMeanTemperature, visParams, 'Name'); ✓
Map.addLayer(annualMeanTemperature, visParams, 'Name', false); ✓
Map.addLayer(annualMeanTemperature, visParams, 'Name', 0.5); ✗
Map.addLayer(annualMeanTemperature, visParams, 'Name', false, 0.5); ✓
Map.addLayer(annualMeanTemperature, {}, 'Name', false, 0.5); ✓

```

Alternatively, precise the name of the arguments:

```

Map.addLayer({eeObject: annualMeanTemperature,
visParams: visParams,
name: 'Name',
opacity: 0.5}); ✓

```

`Map.addLayer(eeObject, visParams, name, shown, opacity)`

Argument	Type
<code>eeObject</code>	Collection Feature Image RawMapId
<code>visParams</code>	<i>FeatureVisualizationParameters ImageVisualizationParameters, optional</i>
<code>name</code>	<i>String, optional</i>
<code>shown</code>	<i>Boolean, optional</i>
<code>opacity</code>	<i>Number, optional</i>

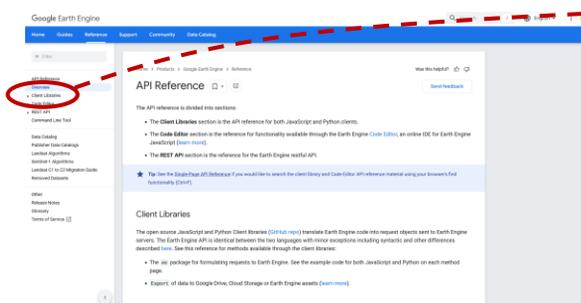
94

Google Earth Engine Getting ready

API Reference

<https://developers.google.com/earth-engine/apidocs/>
(left column > Client Libraries)

Detailed function documentation, code samples



Google Earth Engine

Home Guides Reference

Filter

API Reference

Overview

Client Libraries

JavaScript/Python

- ▶ ee.Algorithms
- ▶ ee.Array
- ▶ ee.Blob
- ▶ ee.Classifier
- ▶ ee.Clusterer
- ▶ ee.ConfusionMatrix
- ▶ ee.Date

95

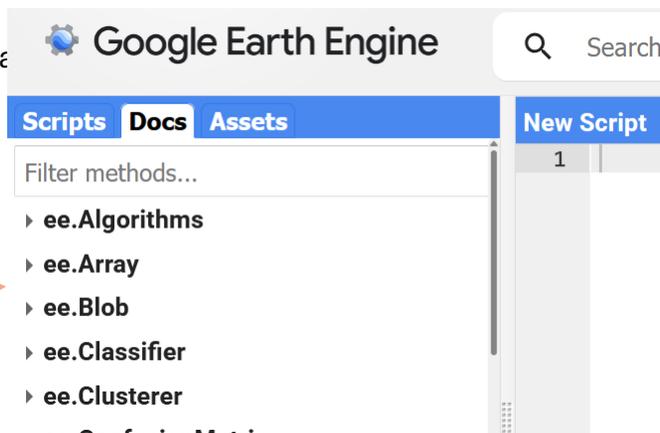
Google Earth Engine Getting ready

API Reference

<https://developers.google.com/earth-engine/apidocs/>
(left column > Client Libraries)

Detailed function documentation, code samples

Also available in the left panel of the code editor, but less information



96

Google Earth Engine Getting ready

API Reference

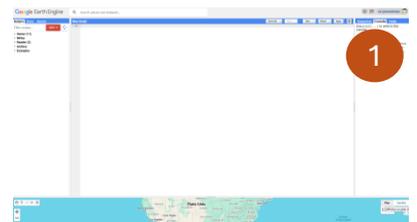
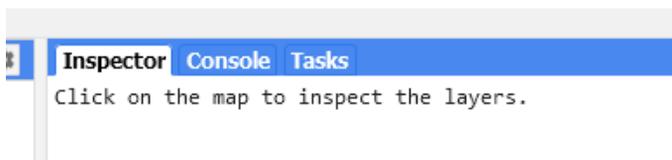
<https://developers.google.com/earth-engine/apidocs/>
(left column > Client Libraries)

Look at the function to *clamp* an Object of type Number

97

Google Earth Engine Getting ready

The Inspector Tool



98

Google Earth Engine

Getting ready

Data: 3 sources

Importing data

99

3 Case Studies

1. Importing data, looking at temperature anomalies, and their relationships with vegetation status in mangroves of the US' Southeast
2. Model potential species distribution based on external species PA data
3. Classify land cover



100

DOI: 10.1111/1365-2745.14440

RESEARCH ARTICLE

Journal of Ecology



Case Study 1

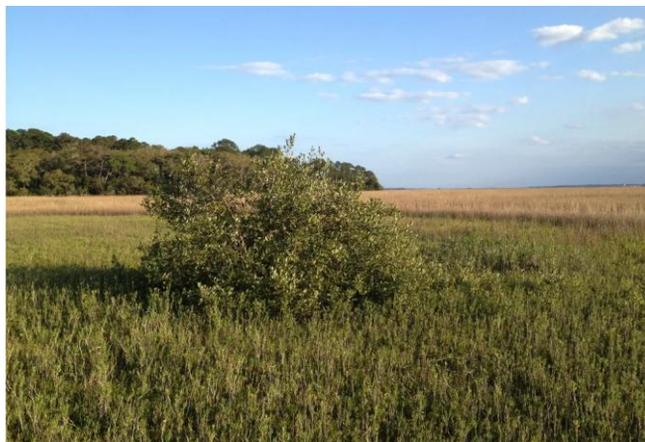
Low temperature: effects on the balance between mangroves and saltmarshes

Mangrove freeze resistance and resilience across a tropical-temperate transitional zone

Yiyang Kang¹ | David A. Kaplan² | Michael J. Osland³



Kyle C. Cavanaugh – Smithsonian magazine



101

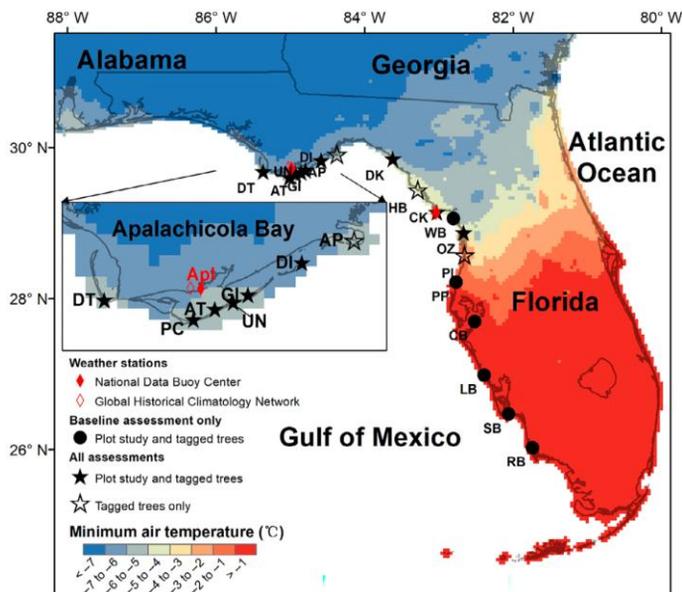
Case Study 1

Low temperature: effects on the balance between mangroves and saltmarshes

Kang et al. (2025) focused on 2022-2023 winter

Goals:

1. Import data
2. Produce the same map (→)
3. Identify mangroves



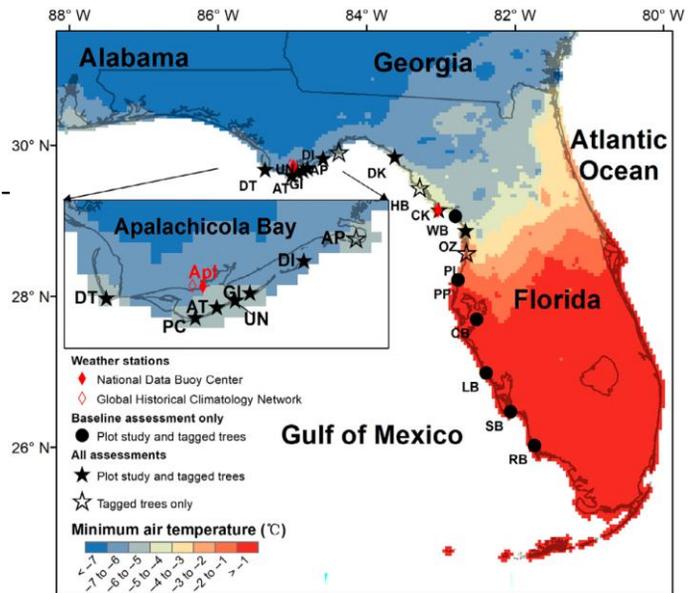
102

Case Study 1

1. Minimum air temperature map

They used PRISM temperature data (only for the USA) for November 2022- March 2023

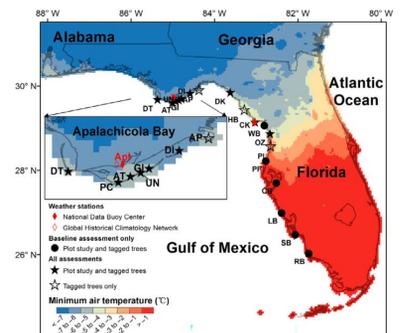
+ states' borders



103

Case Study 1

1. Minimum air temperature map



→ Open a new script

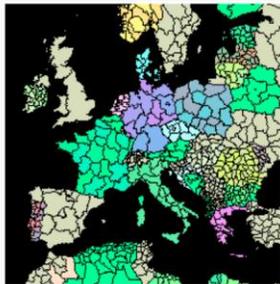
104

Case Study 1

1. Minimum air temperature map

A dataset with US states boundaries?

FAO GAUL: Global Administrative Unit Layers 2015, First-Level Administrative Units



Dataset Availability

2014-12-19T16:45:00Z–
2014-12-19T16:45:00Z

Dataset Provider

[FAO UN](#)

Earth Engine Snippet

FeatureCollection
ee.FeatureCollection
n("FAO/GAUL/2015/

Tags

borders

departments

fao

gaul

infrastructure-
boundaries

provinces

states

Table Schema		
Name	Type	Description
ADM0_CODE	INT	GAUL country code
ADM0_NAME	STRING	UN country name
DISP_AREA	STRING	Unsettled territory: 'Yes' or 'No'
STATUS	STRING	Status of the country
Shape_Area	DOUBLE	Shape area
Shape_Leng	DOUBLE	Shape length
ADM1_CODE	INT	GAUL code of administrative unit
ADM1_NAME	STRING	Name of administrative unit
EXP1_YEAR	INT	Expiry year of the administrative unit
STR1_YEAR	INT	Creation year of the administrative unit

105

Case Study 1

1. Minimum air temperature map

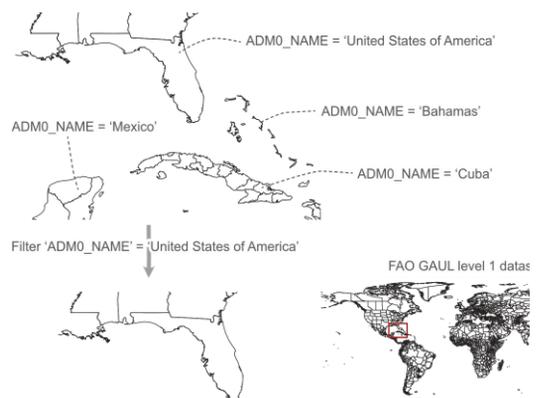
```
var FAOGAULL1 = ee.FeatureCollection('FAO/GAUL/2015/level1')
  .filter(ee.Filter.eq('ADM0_NAME', 'United States of America'));
```

```
.filter()
```

apply a filter to the
FeatureCollection

```
ee.Filter.eq()
```

this filter is of type *eq* (equal),
it only keeps Feature with
ADM0_NAME property
= 'United States of America'



106

Case Study 1

1. Minimum air temperature map

PRISM daily dataset: AN81d

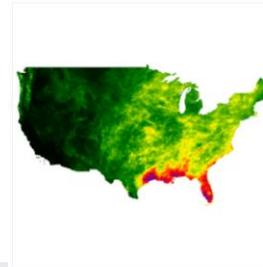
Pixel Size
4638.3 meters

Bands

Name	Units	Min	Max	Pixel Size	Description
ppt	mm	0*	731.65*	meters	Daily total precipitation (including rain and melted snow)
tmean	°C	-40.37*	45.98*	meters	Daily mean temperature (calculated as (tmin+tmx)/2)
tmin	°C	-47.56*	39.59*	meters	Daily minimum temperature
tmax	°C	-38.38*	54.13*	meters	Daily maximum temperature
tdmean	°C	-46.18*	31.61*	meters	Daily mean dew point temperature
vpdmin	hPa	0*	69.86*	meters	Daily minimum vapor pressure deficit
vpdmax	hPa	0*	142.42*	meters	Daily maximum vapor pressure deficit

* estimated min or max value

PRISM Daily Spatial Climate Dataset AN81d



Dataset Availability

1981-01-01T00:00:00Z–
2025-08-23T12:00:00Z

Dataset Provider

[PRISM / OREGONSTATE](#)

Earth Engine Snippet

```
ee.ImageCollection(
  "OREGONSTATE/PRISM/
  AN81d")
```

Cadence

1 Day

Tags

climate
geophysic
oregonsta
precipitati

107

Case Study 1

1. Minimum air temperature map

```
var PRISM_Temp = ee.ImageCollection('OREGONSTATE/PRISM/ANd')
  .filter(ee.Filter.date('2022-11-01', '2023-03-31'))
  .select('tmin');
```

`.filter()`

apply a filter to the ImageCollection

`ee.Filter.date()`

date filter, it only keeps Images that were recorded between two two dates

`.select()`

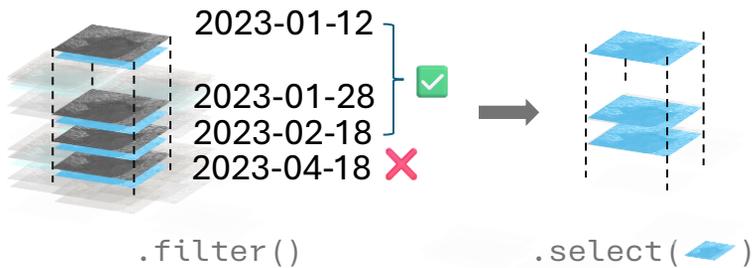
this select the band *tmin* across all Images in the ImageCollection

108

Case Study 1

1. Minimum air temperature map

```
var PRISM_Temp = ee.ImageCollection('OREGONSTATE/PRISM/ANd')
  .filter(ee.Filter.date('2022-11-01', '2023-03-31'))
  .select('tmin');
```



```
print('PRISM_Temp:', PRISM_Temp);
```

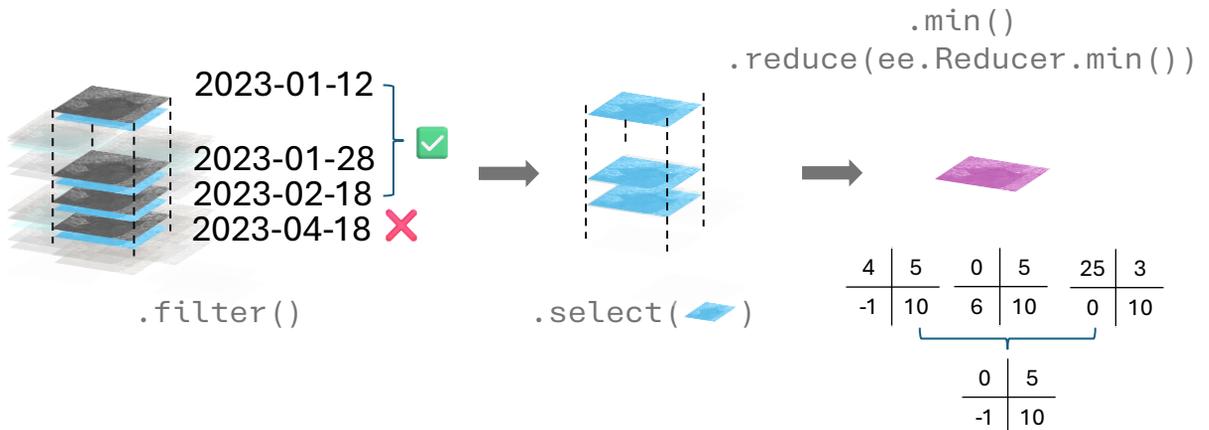
109

Case Study 1

1. Minimum air temperature map

Computing the minimum temperature value per pixel over the period

A **reducer** (max, mean, min, count...) is applied to the ImageCollection to produce a single Image



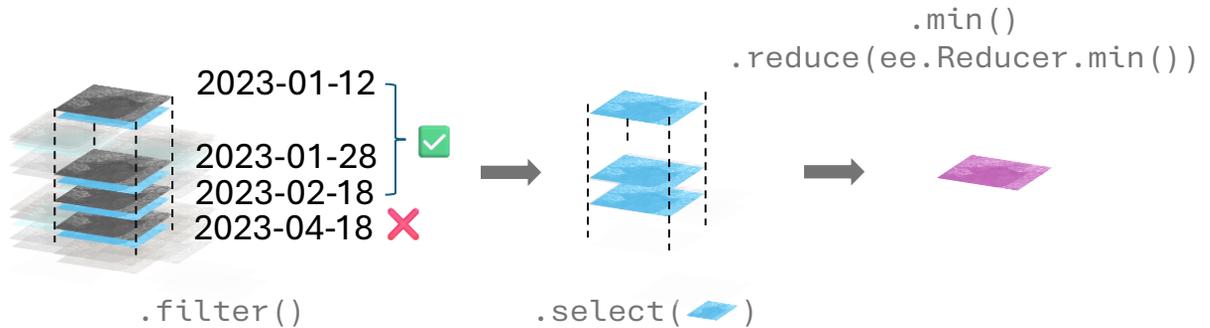
110

Case Study 1

1. Minimum air temperature map

Computing the minimum temperature value per pixel over the period

```
var Temp_min = PRISM_Temp.reduce(ee.Reducer.min());
print('Temp_min:', Temp_min);
```



111

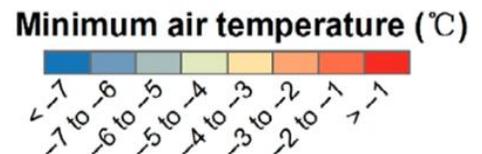
Case Study 1

1. Minimum air temperature map

To visualize these datasets, we first need to focus on the Southeast USA and produce the color gradient from Kang et al. (2025) for the minimum temperature

```
Map.setCenter(-82.295, 29.085, 7);
```

```
var TVis = {
  min: -7,
  max: 0,
  palette: ['#1475B9', '#6E9ABD', '#AABEBD', '#E2E9BF',
            '#FFE3A4', '#FFA672', '#FE6F47', '#F72B22'],
};
```



Hexadecimal color values

112

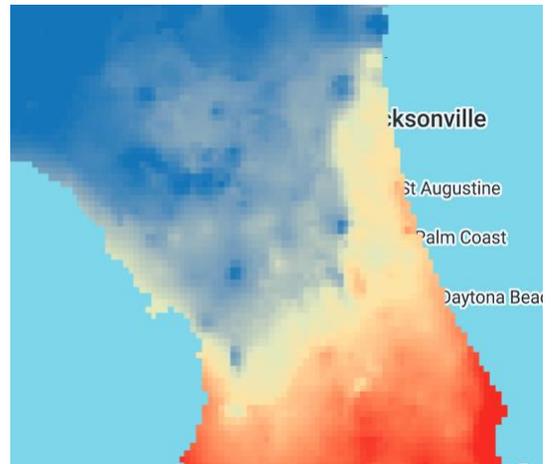
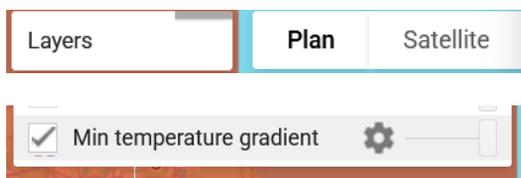
Case Study 1

1. Minimum air temperature map

We display the minimum temperature:

```
Map.addLayer(Temp_min, TVis, 'Min temperature gradient', false);
```

The **false** argument prevent the new layer to automatically load, you need to select it in the Layers tab of the map:



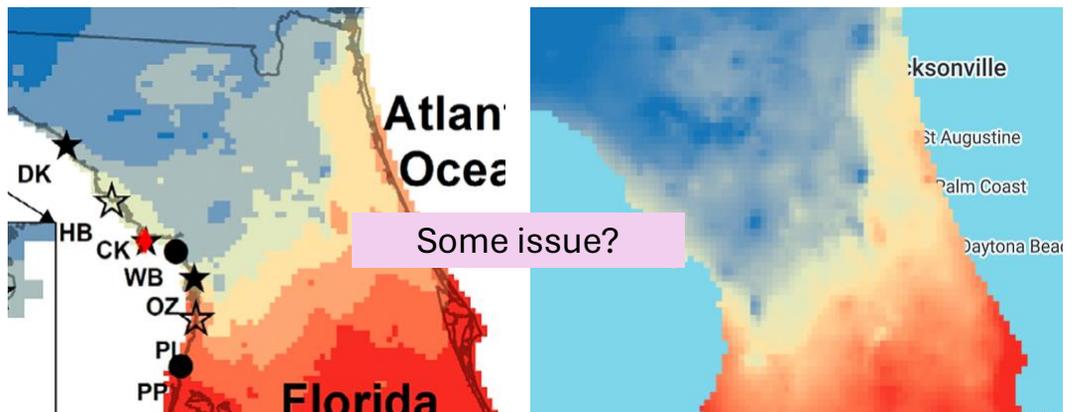
113

Case Study 1

1. Minimum air temperature map

We display the minimum temperature:

```
Map.addLayer(Temp_min, TVis, 'Min temperature gradient', false);
```



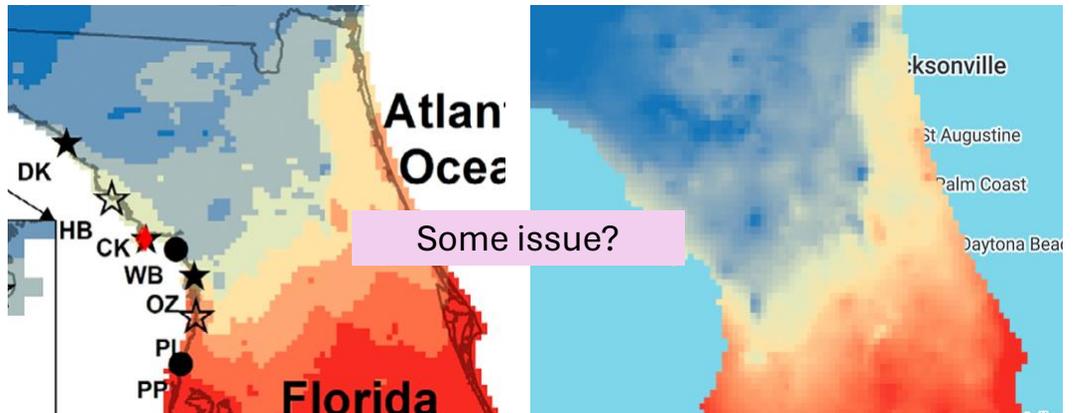
114

Case Study 1

1. Minimum air temperature map

Kang et al. (2025) figure is using categories (-3 to -4), not linear values (-3.56, -3.62, ...).

→ Convert decimal values (-3.56, -5.91) to integers (-3, -5).

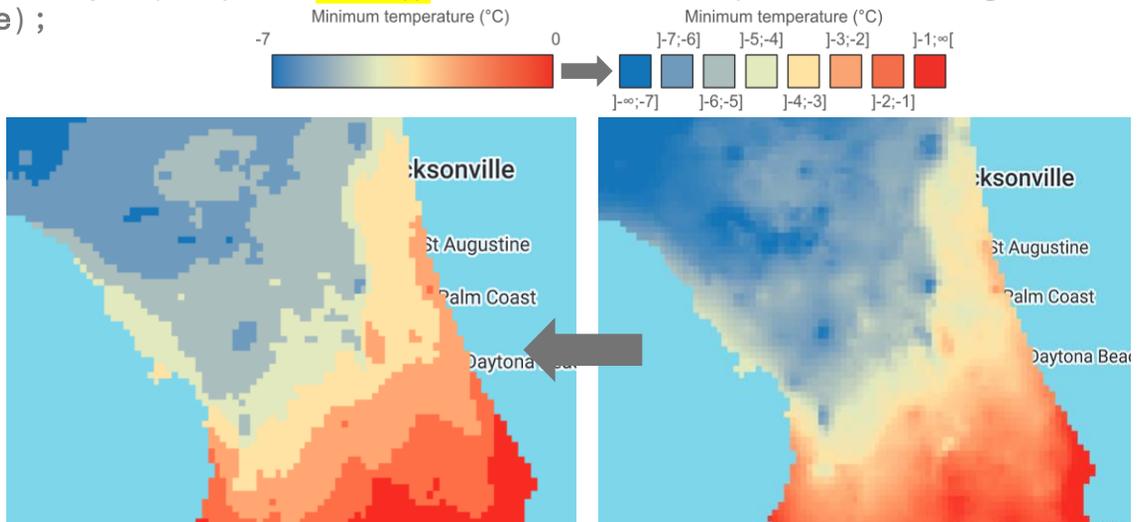


115

Case Study 1

1. Minimum air temperature map

```
Map.addLayer(Temp_min.int(), TVis, 'Min temperature categories', false);
```

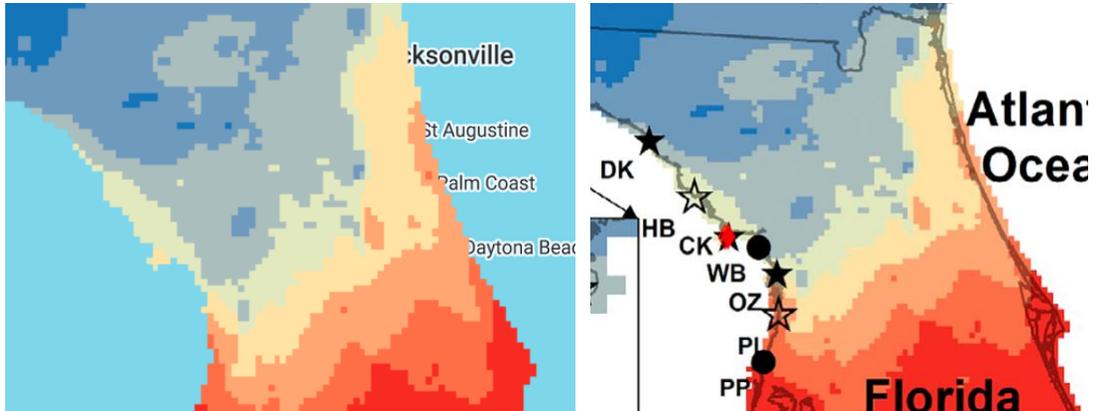


116

Case Study 1

1. Minimum air temperature map

```
Map.addLayer(Temp_min.int(), TVis, 'Min temperature categories', false);
```



117

Case Study 1

1. Minimum air temperature map

We display the state borders:

```
var BorderVis = {
  color: 'black',
  width: 1.0,
  fillColor: "#00000000"
};
```

Means transparent in GEE

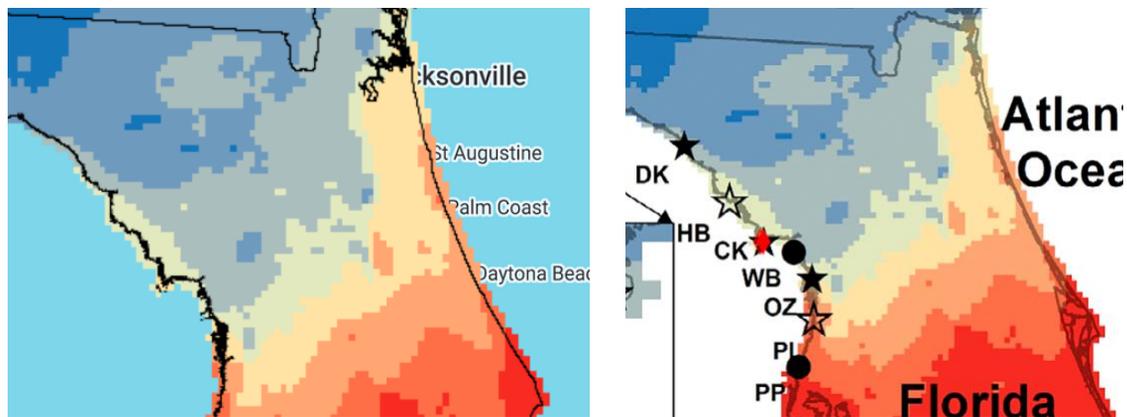
```
Map.addLayer(FAOGAULL1.style(BorderVis), {}, 'States', false);
```

118

Case Study 1

1. Minimum air temperature map

We display the state borders:



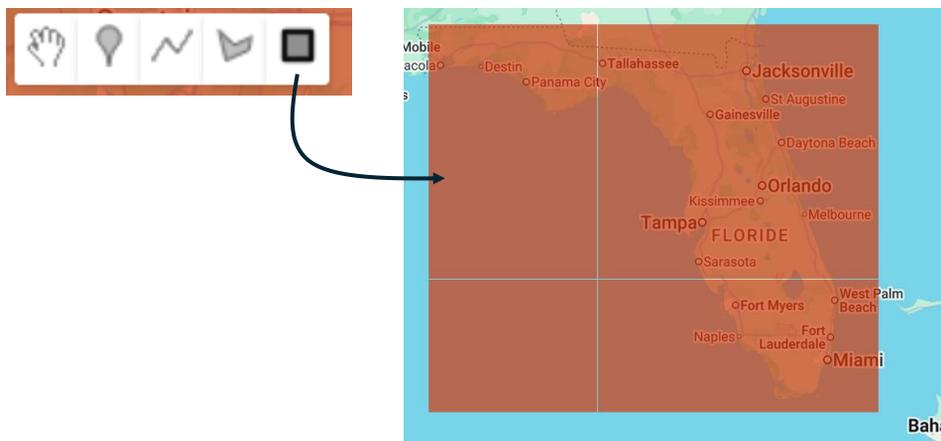
119

Case Study 1

1. Minimum air temperature map

To export the raster:

Create a polygon and name it “ROI”



120

Case Study 1

1. Minimum air temperature map

To export the raster:

Create a polygon and name it “ROI”

```
Export.image.toDrive({
  image: Temp_min.int(),
  description: 'Tempmin',
  folder: 'NTNU2025',
  region: ROI,
  scale: 4638,
  crs: 'EPSG:4326'
});
```

121

The screenshot shows the Google Cloud Console interface with the 'Tasks' tab selected. At the top, there are tabs for 'Inspector', 'Console', and 'Tasks'. Below the tabs, there is a search bar and a link to 'Task Manager' or 'Tasks Page in the Cloud Console'. The main content area is divided into three sections: 'UNSUBMITTED TASKS', 'SUBMITTED TASKS', and 'SUBMITTED TASKS'. In the 'UNSUBMITTED TASKS' section, there is a task named 'Tempmin' with a 'RUN' button. An arrow points down to the first 'SUBMITTED TASKS' section, which shows the 'Tempmin' task in progress, indicated by a snail icon and a progress bar. Another arrow points down to the second 'SUBMITTED TASKS' section, which shows the 'Tempmin' task completed, indicated by a checkmark and a '<1m' duration.

Case Study 1

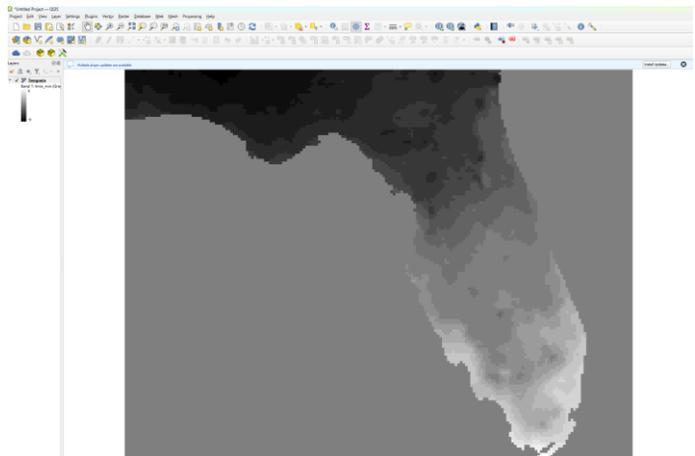
1. Minimum air temperature map

To export the raster:

Create a polygon and name it “ROI”

```
Export.image.toDrive({
  image: Temp_min.int(),
  description: 'Tempmin',
  folder: 'NTNU2025',
  region: ROI,
  scale: 4638,
  crs: 'EPSG:4326'
});
```

122



Case Study 1

Conclusion

```

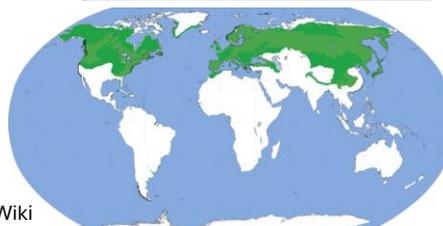
filter()
select()
reduce() + reducers
Map.setCenter()
Map.addLayer()
int()
Export.image.toDrive()

```

123

Case Study 2

Modeling potential species
distribution based on
species PA data



Jaguarondi, snowy owl, *Betula* spp. - Wiki

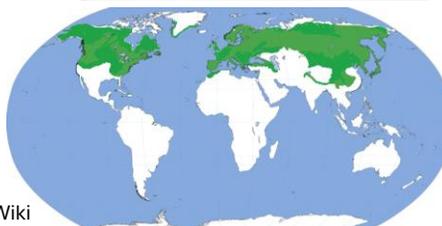
124

Case Study 2

Modeling potential species distribution based on species PA data



→ Create a new script

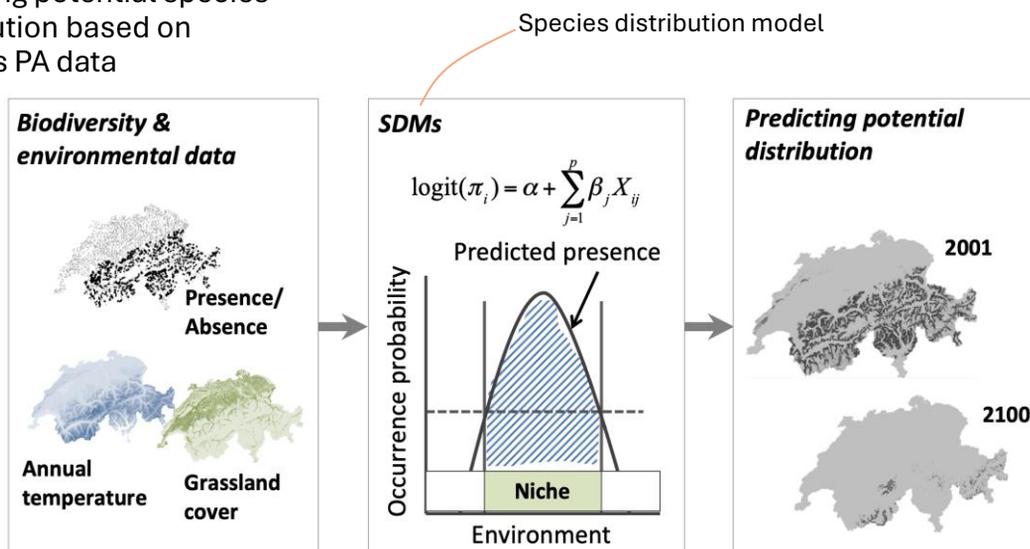


Jaguarondi, snowy owl, *Betula* spp. - Wiki

125

Case Study 2

Modeling potential species distribution based on species PA data

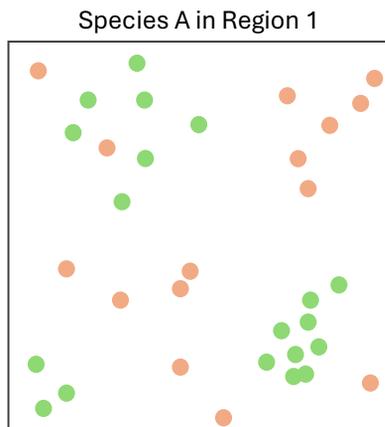


Damaris Zurell – <https://damariszurell.github.io/SDM-Intro/>

126

Case Study 2

Presence-Absence (PA) data



● Present
● Absent

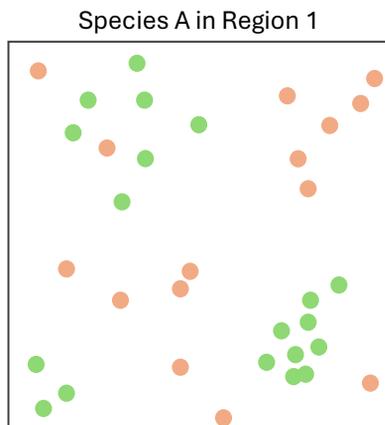
In GEE: FeatureCollection with a *property* stating whether it is present or absent

ID	Sp.	Date	Present	Obs
1	A	2025	1 ●	Marco
2	A	2025	0 ●	Marco
3	A	2025	1 ●	Lianchu

127

Case Study 2

Presence-Absence (PA) data



● Present
● Absent

To model the Species A distribution across the entire Region 1:

Parameters that could explain Species A distribution such as

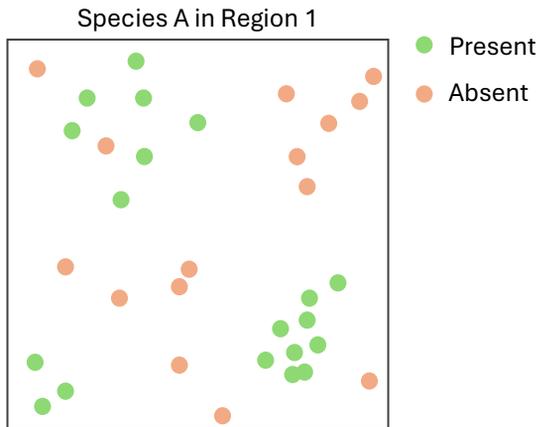
- Temperature
- Precipitation
- Topography
- Human presence
- Barriers
- ...

For each point PA ● ●, the parameters are sampled and used in a model produce a map of Region 1's potential distribution of Species A

128

Case Study 2

Presence-Absence (PA) data



Many approaches to model distribution...

GEE has only one function to model species distribution, and it requires PA data:
Maxent (Maximum Entropy)

129

Case Study 2

Presence-Absence (PA) data

Biodiversity Informatics, 15, 2020, pp. 69-80

PRESENCE-ONLY AND PRESENCE-ABSENCE DATA FOR COMPARING SPECIES DISTRIBUTION MODELING METHODS

JANE ELITH^{1*}, CATHERINE H. GRAHAM², ROOZBEH VALAVI¹, MEINRAD ABEGG², CAROLINE BRUCE³, SIMON FERRIER⁴, ANDREW FORD⁵, ANTOINE GUISSAN⁶, ROBERT J. HIJMANS⁷, FALK HUETTSMANN⁸, LUCIA LOHMANN⁹, BETTE LOISELLE¹⁰, CRAIG MORITZ¹¹, JAKE OVERTON¹², A. TOWNSEND PETERSON¹³, STEVEN PHILLIPS¹⁴, KAREN RICHARDSON⁵, STEPHEN E. WILLIAMS⁵, SUSAN K. WISER¹⁷, THOMAS WOHLGEMUTH², NIKLAUS E. ZIMMERMANN²

SA	Continental Brazil, Ecuador, Colombia, Bolivia, and Peru, South America	12223.17		11 (0)	1 000 vascular plants: 30
----	---	----------	--	--------	---------------------------

<https://osf.io/kwc4v/files> > data > Records > test_pa > SAtest_pa.csv

Please download the file

130

Case Study 2

Presence-Absence (PA) data

Biodiversity Informatics, 15, 2020, pp. 69-80

PRESENCE-ONLY AND PRESENCE-ABSENCE DATA FOR COMPARING SPECIES DISTRIBUTION MODELING METHODS

SA Continental Brazil, Ecuador, Colombia, Bolivia, and Peru, South America

12223.17



11 (0) 1 000 vascular plants: 30

JANE ELITH^{1*}, CATHERINE H. GRAHAM², ROOZBEH VALAVI¹, MEINRAD ABEGG³, CAROLINE BRUCE³, SIMON FERRIER⁴, ANDREW FORD⁵, ANTOINE GUISSAN⁶, ROBERT J. HIJMANS⁷, FALK HUETTMMANN⁸, LUCIA LOHMANN⁹, BETTE LOISELLE¹⁰, CRAIG MORITZ¹¹, JAKE OVERTON¹², A. TOWNSEND PETERSON¹³, STEVEN PHILLIPS¹⁴, KAREN RICHARDSON¹⁵, STEPHEN E. WILLIAMS¹⁶, SUSAN K. WISER¹⁷, THOMAS WOHLGEMUTH¹, NIKLAUS E. ZIMMERMANN²

<https://osf.io/kwc4v/files/osfstorage> > data > Records > test_pa > SAtest_pa.csv

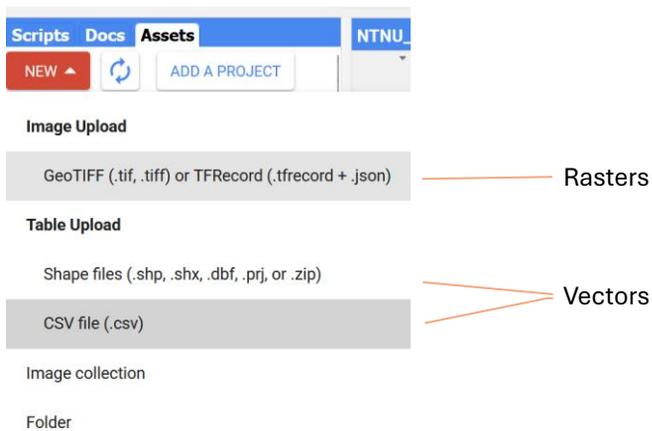
group	siteid	x	y	sa01	sa02	sa03	sa04	sa05	sa06	...
plant	allpahua	-73.4167	-3.95	0	0	0	1 ●	0	0	0 ...
plant	alterdoc	-54.9667	-2.5	0	0	0	0 ●	0	0	0
plant	altodem	-73.8333	10.9167	0	0	0	0	0	0	0
plant	amotape	-80.6167	-4.15	0	0	0	0	0	0	0
plant	anchicay	-76.8333	3.75	0	0	0	0	0	0	0
plant	antado	-75.9167	7.25	0	0	0	0	0	0	0
plant	araracua	-72.3333	-0.4167	0	0	0	0	0	0	0 ...

coordinates
30 species

131

Case Study 2

Importing external data into GEE



Scripts Docs **Assets** NTNU

NEW ↻ ADD A PROJECT

Image Upload

GeoTIFF (.tif, .tiff) or TFRecord (.tfrecord + .json) → Rasters

Table Upload

Shape files (.shp, .shx, .dbf, .prj, or .zip) → Vectors

CSV file (.csv) → Vectors

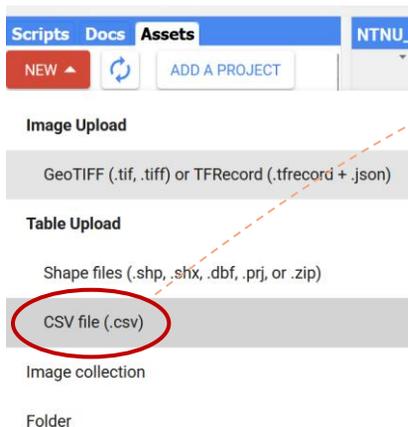
Image collection

Folder

132

Case Study 2

Importing external data into GEE



Rasters

Vectors

Upload a new csv asset

Source files

SELECT

Please drag and drop or select a file for this asset.
Allowed extension: csv.

SAtest_pa.csv

Asset ID

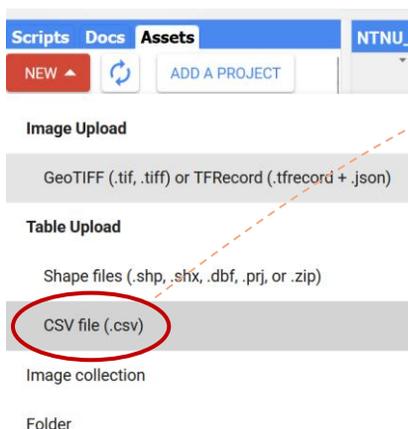
projects/ee-jauneatemp/assets/ others/SA_spPA

Folder name / file name

133

Case Study 2

Importing external data into GEE



Rasters

Vectors

Upload a new csv asset

Source files

SELECT

Please drag and drop or select a file for this asset.
Allowed extension: csv.

SAtest_pa.csv

Asset ID

projects/ee-jauneatemp/assets/ others/SA_spPA

Folder name / file name

X column
longitude x

Y column
latitude y

Explanation

CRS
EPSG:4326

Explanation

CSV delimiter
,

Geodesic

Infer geodesic from...

CSV qualifier
"

134

Case Study 2

Importing external data into GEE

Image Upload
GeoTIFF (.tif, .tiff) or TFRecord (.tfrecord + .json) — Rasters

Table Upload
Shape files (.shp, .shx, .dbf, .prj, or .zip) — Vectors
CSV file (.csv) — Vectors
 Image collection
 Folder

Upload a new csv asset

Source files
SELECT
Please drag and drop or select a file for this asset. Allowed extension: csv.
SAtest_pa.csv

Asset ID
projects/ee-jauneatemp/assets/... others/SA_spPA
Folder name / file name

X column longitude X
Y column latitude y

CRS EPSG:4326
Geodesic Infer geodesic from...
CSV delimiter
CSV qualifier

CANCEL UPLOAD

135

Case Study 2

Importing external data into GEE

Inspector Console Tasks

Search or cancel multiple tasks in the [Task Manager](#) or try the [Tasks Page in the Cloud Console](#)

UNSUBMITTED TASKS
 (100.00%) SAtest_pa upload

SUBMITTED TASKS
 Ingest table: "projects/ee-jauneatemp/assets/other..." <1m

Server processing step (no need to stay connected)

Uploading step (takes time for large files)

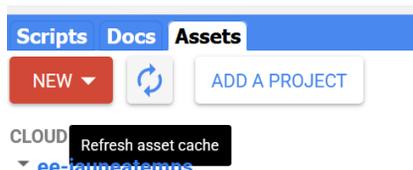
ok errors

SUBMITTED TASKS
 Ingest table: "projects/ee-jauneatemp/assets/other..." ✓ <1m
 Ingest table: "projects/ee-jauneatemp/assets/other..." ▲ 1m

136

Case Study 2

Importing external data into GEE



Directly import it in the script

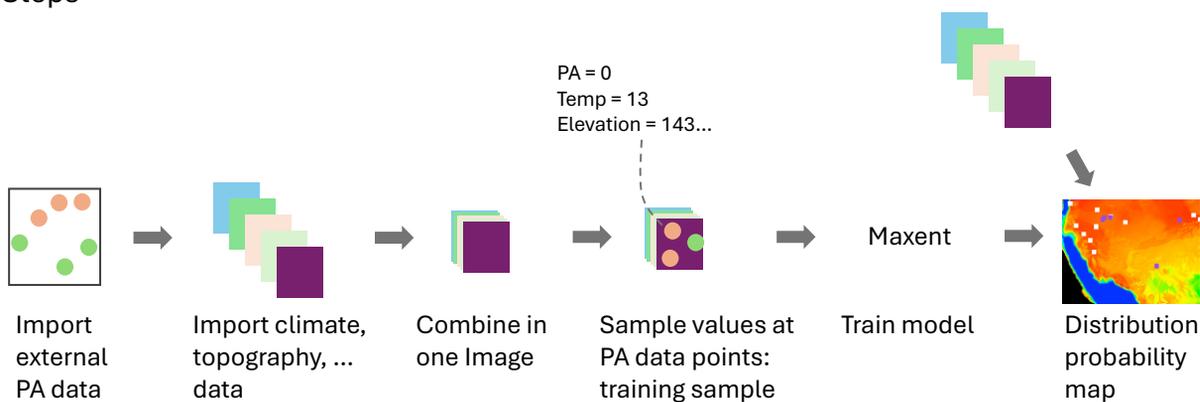


Rename table:
testPA

137

Case Study 2

Steps

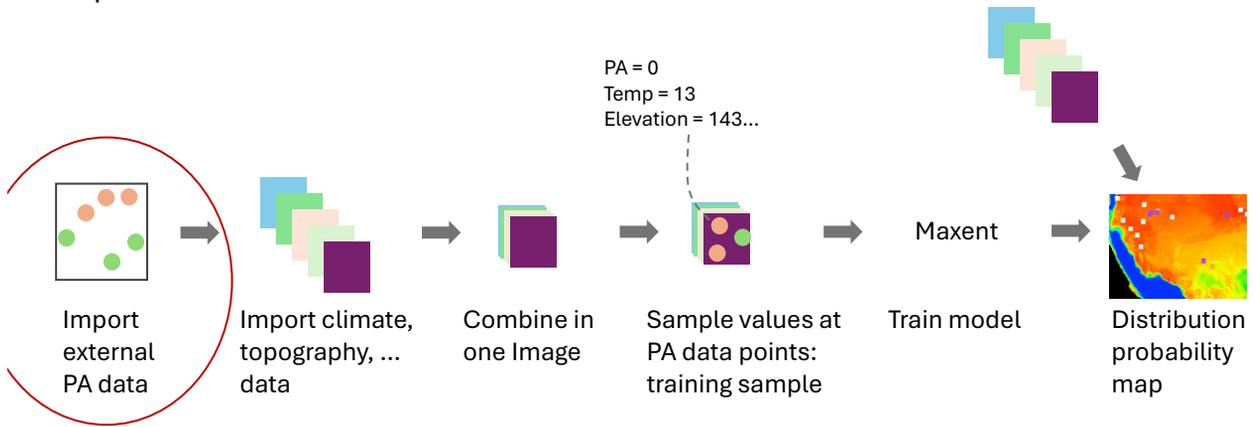


Maxent - biodiversityinformatics.amnh.org/open_source/maxent/

138

Case Study 2

Steps



Maxent - biodiversityinformatics.amnh.org/open_source/maxent/

139

Case Study 2

Visualize PA data

```
Map.addLayer(speciesPA.filter('sa10 == 0'),
  {color: 'red'},
  'Training data (sp. absent)');
```

Only Features for column sa10 = 0
(species 10 is absent)

```
Map.addLayer(speciesPA.filter('sa10 == 1'),
  {color: 'blue'},
  'Training data (sp. present)');
```

Only Features for column sa10 = 0
(species 10 is absent)

```
Map.centerObject(speciesPA, 4);
```

140

Case Study 2

Visualize PA data

Only Features for column sa10 = 0
(species 10 is absent)

```
Map.addLayer(speciesPA.filter('sa10 == 0'),
  {color: 'red'},
  'Training data (sp. absent)');
```

```
Map.addLayer(speciesPA.filter('sa10 == 1'),
  {color: 'blue'},
  'Training data (sp. present)');
```

```
Map.centerObject(speciesPA, 4);
```



141

Case Study 2

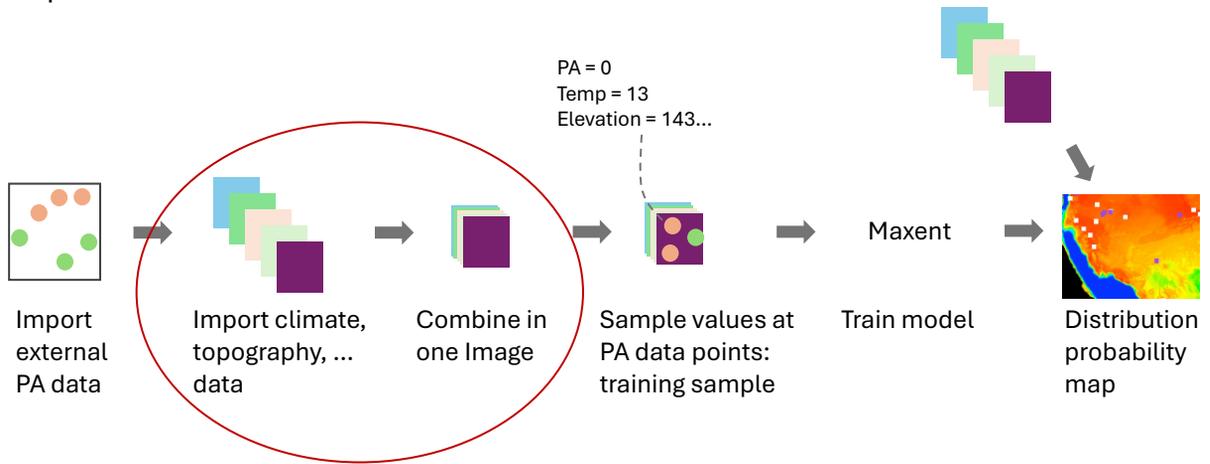
Visualize PA data

```
var trainingData = ee.FeatureCollection(speciesPA);
```

142

Case Study 2

Steps



Maxent - biodiversityinformatics.amnh.org/open_source/maxent/

143

Case Study 2

Parameters

- Mean temperature
- Temperature variation
- Minimum temperature
- Precipitation



- Elevation
- Soil type
- Slope
- ...

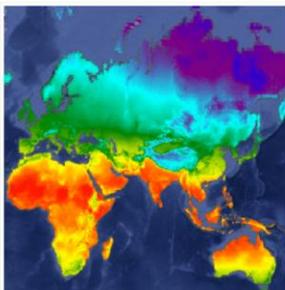
Begonia spp. – Karl Magnacca

144

Case Study 2

Parameters

WorldClim BIO Variables V1



Dataset Availability

1960-01-01T00:00:00Z–
1991-01-01T00:00:00Z

Dataset Provider

[University of California, Berkeley](#)

Earth Engine Snippet

```
ee.Image("WORLDCLIM/V1/BIO")
```



Pixel Size

927.67 meters

Bands

Name	Units	Min	Max	Scale	Pixel Size	Description
bio01	°C	-29*	32*	0.1	meters	Annual mean temperature
bio02	°C	0.9*	21.4*	0.1	meters	Mean diurnal range (mean of monthly (max temp - min temp))
bio03	%	7*	96*		meters	Isothermality (bio02/bio07 * 100)
bio04	°C	0.62*	227.21*	0.01	meters	Temperature seasonality (Standard deviation * 100)
bio05	°C	-9.6*	49*	0.1	meters	Max temperature of warmest month
bio06	°C	-57.3*	25.8*	0.1	meters	Min temperature of coldest month
bio07	°C	5.3*	72.5*	0.1	meters	Temperature annual range (bio05-bio06)
bio08	°C	-28.5*	37.8*	0.1	meters	Mean temperature of wettest quarter
bio09	°C	-52.1*	36.6*	0.1	meters	Mean temperature of driest quarter
bio10	°C	-14.3*	38.3*	0.1	meters	Mean temperature of warmest quarter
bio11	°C	-52.1*	28.9*	0.1	meters	Mean temperature of coldest quarter
bio12	mm	0*	11401*		meters	Annual precipitation

What are the bands for:

Annual mean temperature

Annual precipitation

Driest month precipitation

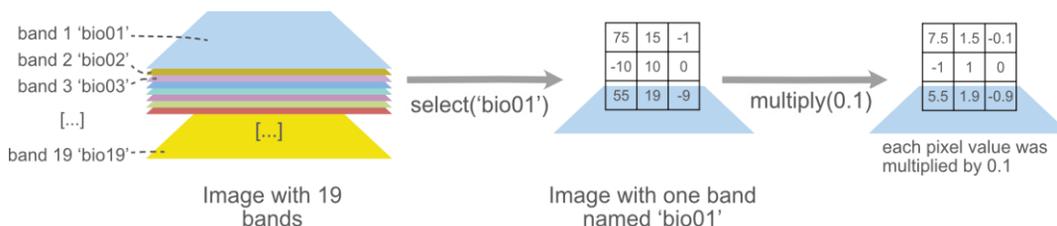
Temperature seasonality?

145

Case Study 2

Parameters

```
var worldclim = ee.Image('WORLDCLIM/V1/BIO');
var annualMeanTemperature = worldclim.select('bio01')
  .multiply(0.1);
var annualPrecipitation = worldclim.select('bio12');
var driestMonthPrecipitation = worldclim.select('bio14');
var temperatureSeasonality = worldclim.select('bio04')
  .multiply(0.01);
```



146

Case Study 2

Parameters

Description					
Bands					
Name	Units	Min	Max	Pixel Size	Description
elevation	m	-10*	6500*	30 meters	Elevation
* estimated min or max value					

NASA SRTM Digital Elevation 30m 



Dataset Availability

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

```
var elevation = ee.Image('USGS/SRTMGL1_003')
  .select('elevation');
```

147

Case Study 2

Parameters

```
var allData = ee.Image([elevation, annualMeanTemperature,
  annualPrecipitation, driestMonthPrecipitation,
  temperatureSeasonality]);
```

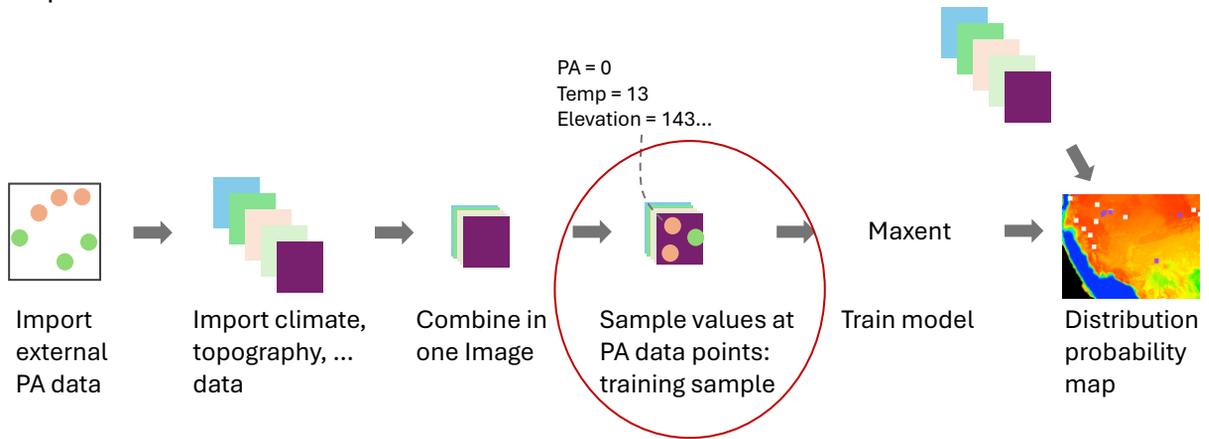


Combine in
one Image

148

Case Study 2

Steps



Maxent - biodiversityinformatics.amnh.org/open_source/maxent/

149

Case Study 2

Sampling values at PA points

```
var trainingSample = allData.sampleRegions({
  collection: trainingData,
  scale: 500});
```

```
print(trainingSample)
```

Usage	Returns
<code>Image.sampleRegions(collection, properties, scale, projection, tileScale, geometries)</code>	FeatureCollection geometries

Argument	Type	Details
<code>this: image</code>	Image	The image to sample.
<code>collection</code>	FeatureCollection	The regions to sample over.
<code>properties</code>	List, default: null	The list of properties to copy from each input feature. Defaults to all non-system properties.
<code>scale</code>	Float, default: null	A nominal scale in meters of the projection to sample in. If unspecified, the scale of the image's first band is used.
<code>projection</code>	Projection, default: null	The projection in which to sample. If unspecified, the projection of the image's first band is used. If specified in addition to scale, rescaled to the specified scale.
<code>tileScale</code>	Float, default: 1	A scaling factor used to reduce aggregation tile size; using a larger tileScale (e.g., 2 or 4) may enable computations that run out of memory with the default.
<code>geometries</code>	Boolean, default: false	If true, the results will include a point geometry per sampled pixel. Otherwise, geometries will be omitted (saving memory).

150

Case Study 2

Sampling values at PA points

```
var trainingSample = allData.sampleRegions({
  collection: trainingData,
  scale: 500});

print(trainingSample)
```

```
▼ FeatureCollection (152 elements, 0 columns)
  type: FeatureCollection
  columns: Object (0 properties)
  ▼ features: List (152 elements)
    ▼ 0: Feature 00000000000000000001_0
      type: Feature
      id: 00000000000000000001_0
      geometry: null
      ▼ properties: Object (37 properties)
        bio01: 26.1
        bio04: 5.5600000000000005
        bio12: 1980
        bio14: 33
        elevation: 1
        group: plant
        sa01: 0
        sa02: 0
        sa03: 0
        sa04: 0
```

JSON

151

Case Study 2

Sampling values at PA points

```
var trainingSample = allData.sampleRegions({
  collection: trainingData,
  scale: 500});

print(trainingSample)
```

```
▼ FeatureCollection (152 elements, 0 columns)
  type: FeatureCollection
  columns: Object (0 properties)
  ▼ features: List (152 elements)
    ▼ 0: Feature 00000000000000000001_0
      type: Feature
      id: 00000000000000000001_0
      geometry: null
      ▼ properties: Object (37 properties)
        bio01: 26.1
        bio04: 5.5600000000000005
        bio12: 1980
        bio14: 33
        elevation: 1
        group: plant
        sa01: 0
        sa02: 0
        sa03: 0
        sa04: 0
```

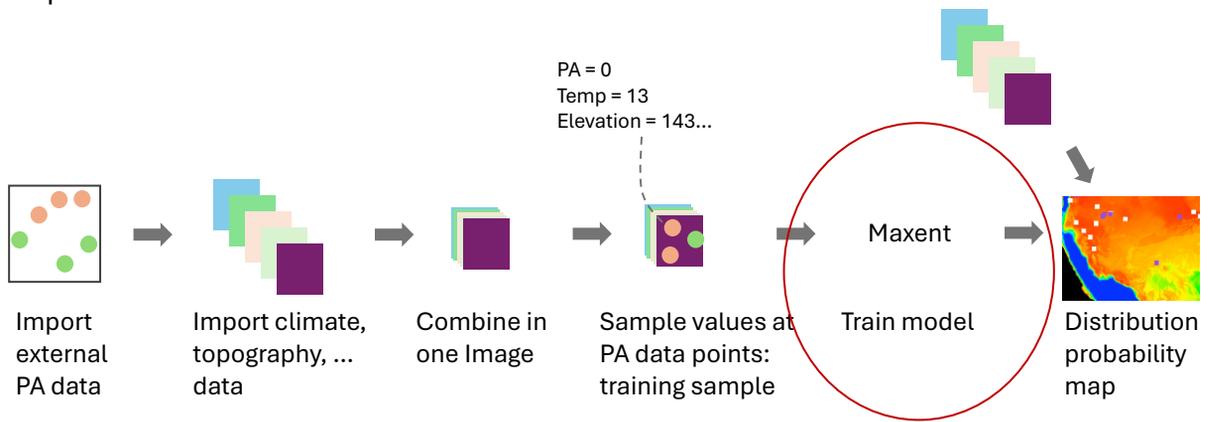
JSON

can be exported as a
shapefile or table (csv)

152

Case Study 2

Steps



Maxent - biodiversityinformatics.amnh.org/open_source/maxent/

153

Case Study 2

Training

Define the classifier that is used + train it with a sample

154

Case Study 2

Training

Define the classifier that is used + train it with a sample

- ▼ ee.Classifier
 - amnhMaxent**
 - confusionMatrix
 - decisionTree
 - decisionTreeEnsemble
 - explain
 - libsvm
 - load
 - minimumDistance
 - mode
 - schema
 - setOutputMode
 - smileCart
 - smileGradientTreeBoost
 - smileKNN
 - smileNaiveBayes
 - smileRandomForest
 - spectralRegion
 - train

155

Case Study 2

Training

Define the classifier that is used + train it with a sample

ee.Classifier.amnhMaxent

[Send feedback](#)

On this page
Examples

Creates a Maximum Entropy classifier. Maxent is used to model species distribution probabilities using environmental data for locations of known presence and for a large number of 'background' locations. For more information and to cite, see: https://biodiversityinformatics.amnh.org/open_source/maxent/ and the reference publication: Phillips, et. al., 2004 A maximum entropy approach to species distribution modeling, Proceedings of the Twenty-First International Conference on Machine Learning. The output is a single band named 'probability', containing the modeled probability, and an additional band named 'clamp' when the 'writeClampGrid' argument is true.

Usage	Returns
<pre>ee.Classifier.amnhMaxent(<i>categoricalNames</i>, <i>outputFormat</i>, <i>autoFeature</i>, <i>linear</i>, <i>quadratic</i>, <i>product</i>, <i>threshold</i>, <i>hinge</i>, <i>hingeThreshold</i>, <i>l2lqThreshold</i>, <i>lq2lqptThreshold</i>, <i>addSamplesToBackground</i>, <i>addAllSamplesToBackground</i>, <i>betaMultiplier</i>, <i>betaHinge</i>, <i>betaLq</i>, <i>betaCategorical</i>, <i>betaThreshold</i>, <i>extrapolate</i>, <i>doClamp</i>, <i>writeClampGrid</i>, <i>randomTestPoints</i>, <i>seed</i>)</pre>	Classifier

amnh: American
Museum of
Natural History

Maxent:
Maximum Entropy

156

Case Study 2

Training

In GEE: Define the classifier that is used + train it with a sample

```
var classifier = ee.Classifier.amnhMaxent().train({
  features: trainingSample, ----- Sampled data (T°, PA, elevation...)
  classProperty: 'sa10', ----- column (property) used to identify PA (0/1)
  inputProperties: allData.bandNames()
});
```

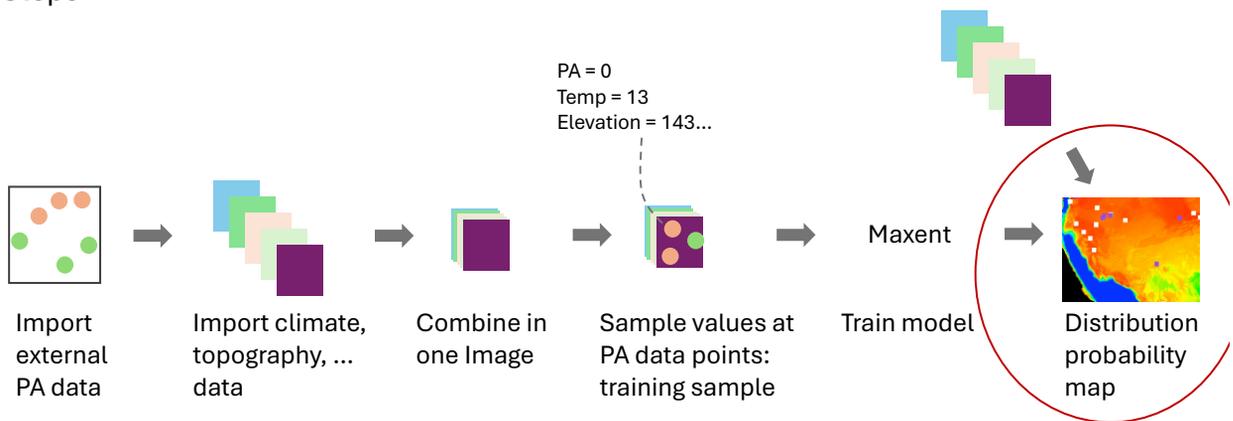
The names of the columns that are used as model parameters: temperature variables, elevation, precipitation



157

Case Study 2

Steps



Maxent - biodiversityinformatics.amnh.org/open_source/maxent/

158

Case Study 2

Predicting species distribution

```
var imageClassified = allData.classify(classifier);
```

ee.Image.classify



[Send feedback](#)

On this page

Examples

Classifies an image.

Usage	Returns
<code>Image.classify(classifier, <i>outputName</i>)</code>	Image

159

Case Study 2

Predicting species distribution

```
var imageClassified = allData.classify(classifier);
```

```
print(imageClassified, 'Classified');
```

```
Map.addLayer(imageClassified,
  {bands: 'probability', min: 0, max: 1},
  'Probability');
```

Try with different species: sa01, sa02, ...

160

Case Study 2

Conclusion

```

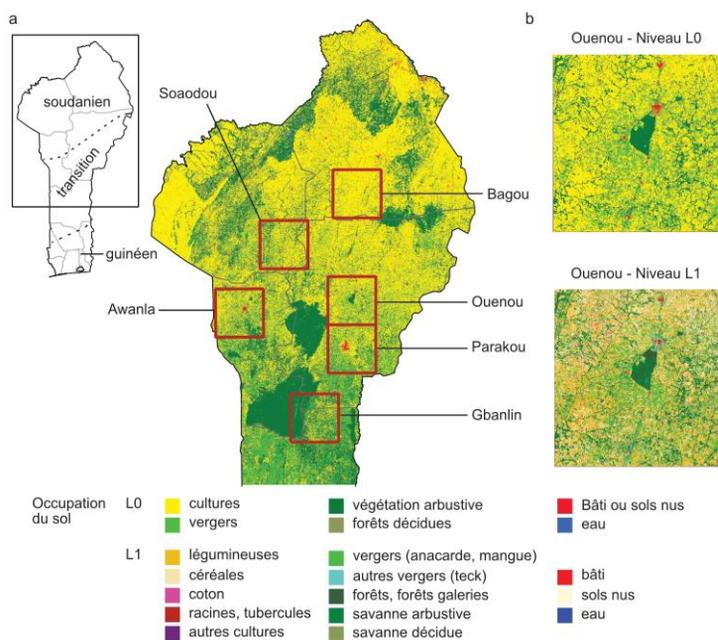
filter()
sampleRegions()
bandNames()
ee.Classifier.amnhMaxent()
train()
classify()

```

161

Case Study 3

Land cover classification and vegetation trends

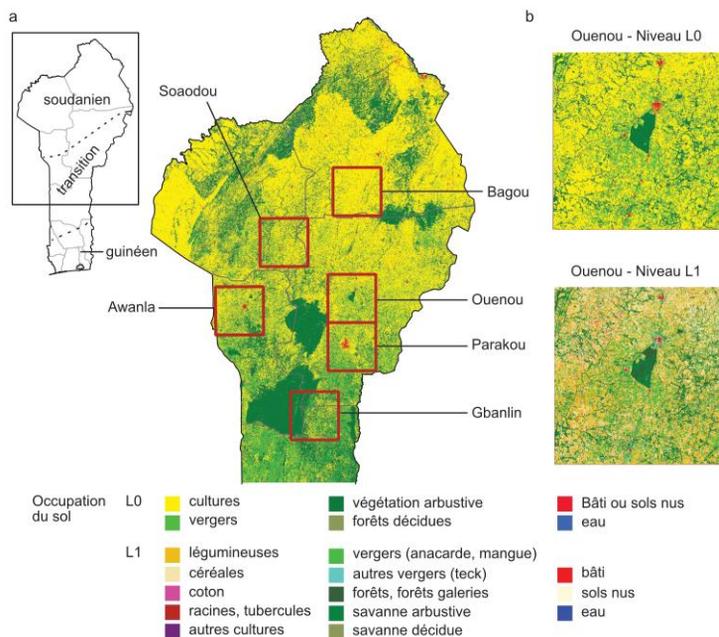


162

Case Study 3

Land cover classification and vegetation trends

→ Create a new script

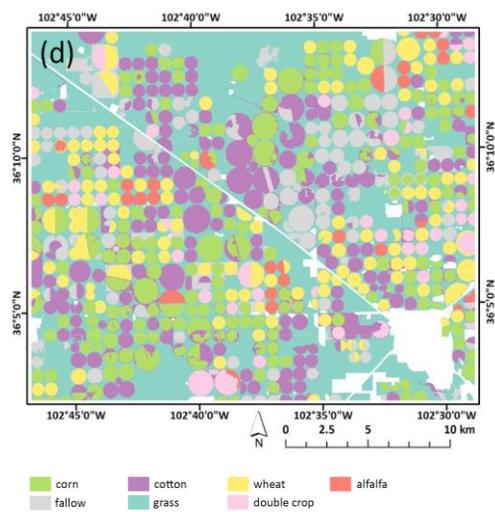
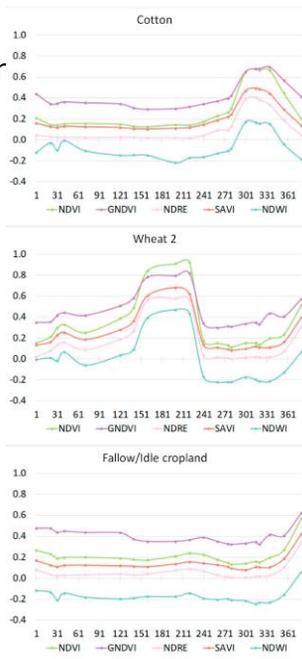


163

Case Study 3

Land cover classification and vegetation trends

Land use, land cover (LUCL) maps



Csilik et al. (2019)

164

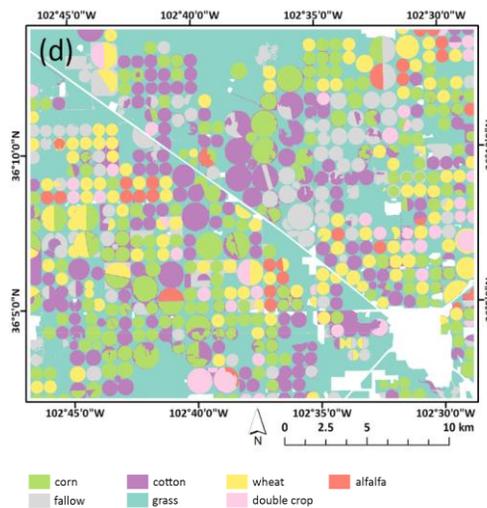
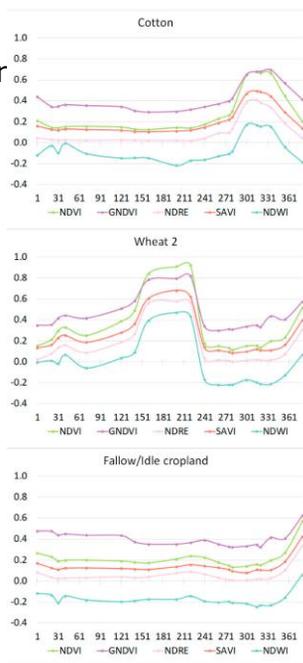
Case Study 3

Land cover classification and vegetation trends

Land use, land cover (LUCL) maps

Not always adapted:

- classes
- spatial resolution
- period
- etc



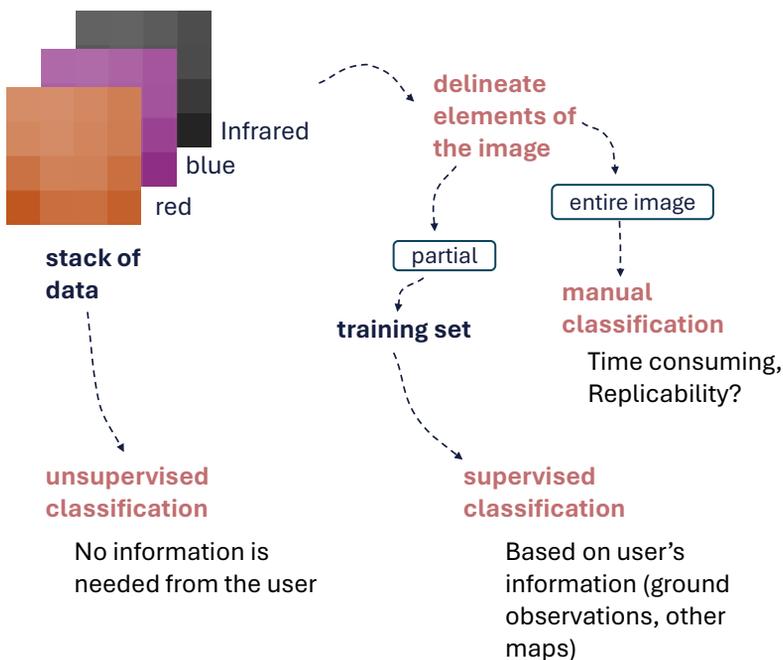
Csilik et al. (2019)

165

Case Study 3

Land cover classification and vegetation trends

Different approaches

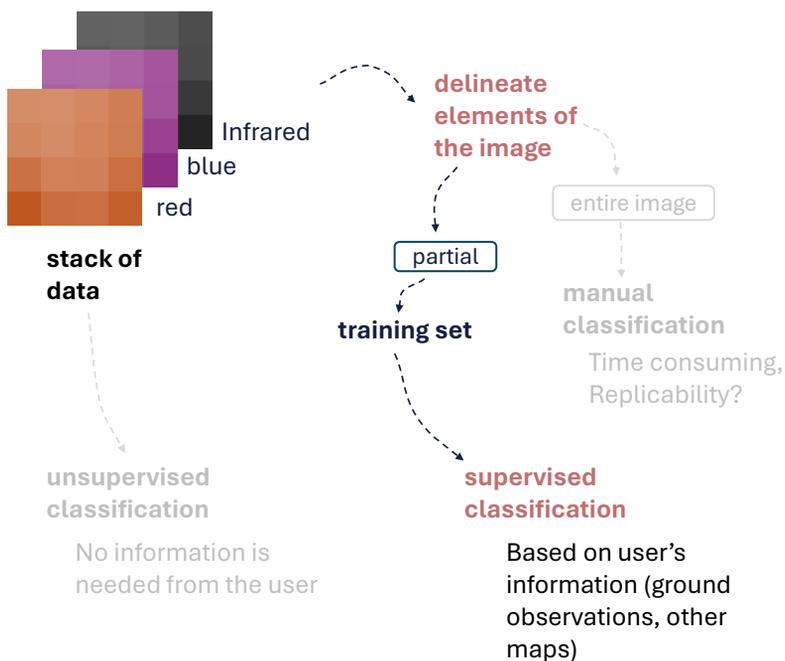


166

Case Study 3

Land cover classification and vegetation trends

Different approaches



167

Case Study 3

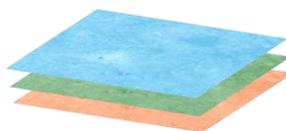
Supervised classification



Area of interest

- Forests
- Built-up
- Crops
- Water
- Bare soil
- etc.

Classes



Spatial information

Model

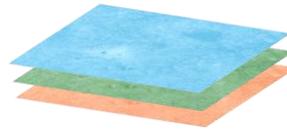
168

Case Study 3

Supervised classification



- Forests
- Built-up
- Crops
- Water
- Bare soil
- etc.



Spatial information

Model

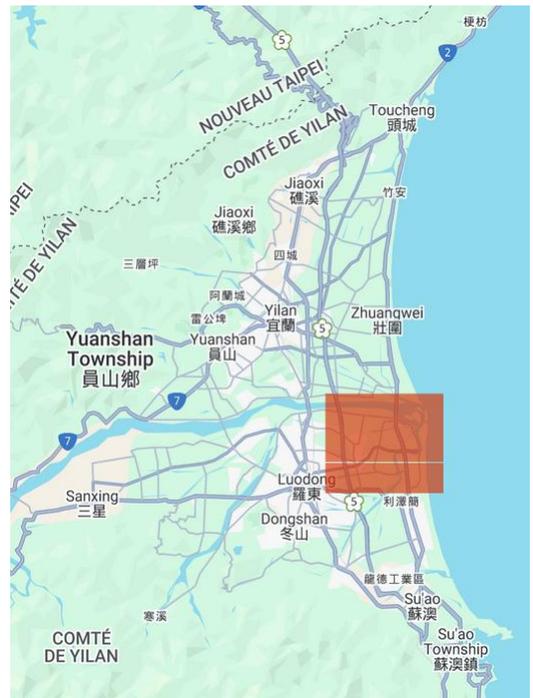
169

Case Study 3

Area of interest

Use the geometry tool to create a polygon near Yilan

Rename it “study_site”



170

Case Study 3

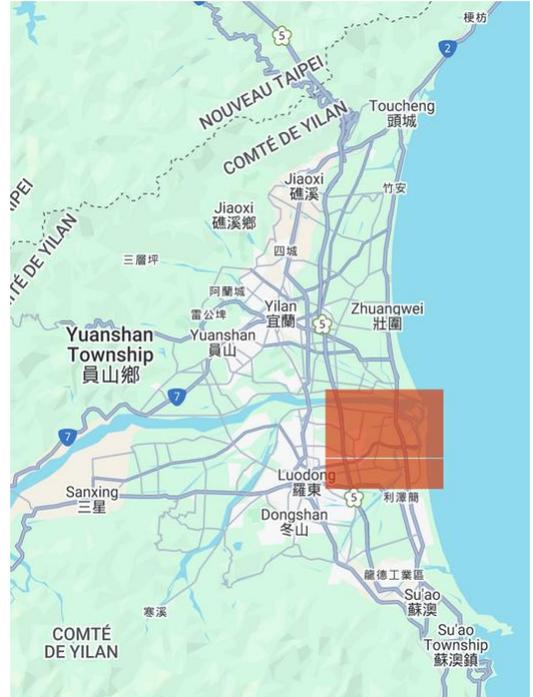
Area of interest

Use the geometry tool to create a polygon near Yilan

Rename it “study_site”

Main land cover classes:

- built-up
- water
- natural vegetation
- farms
- bare soil

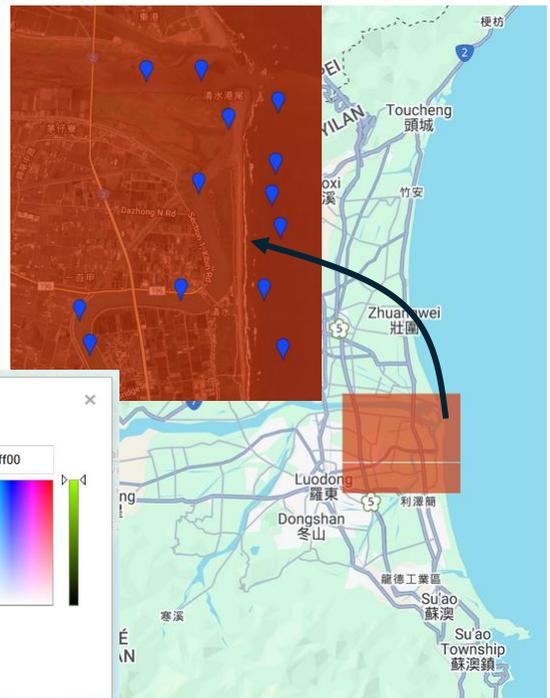
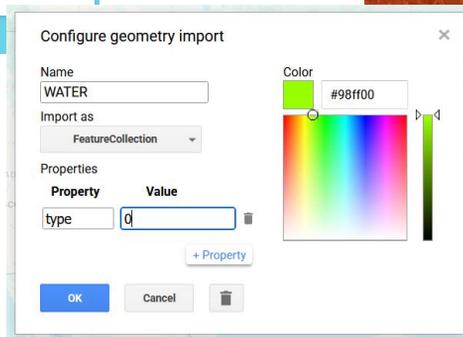
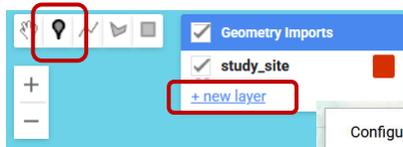


171

Case Study 3

Land cover classes: “sample” points

Use the geometry tool to create new points for each land cover class

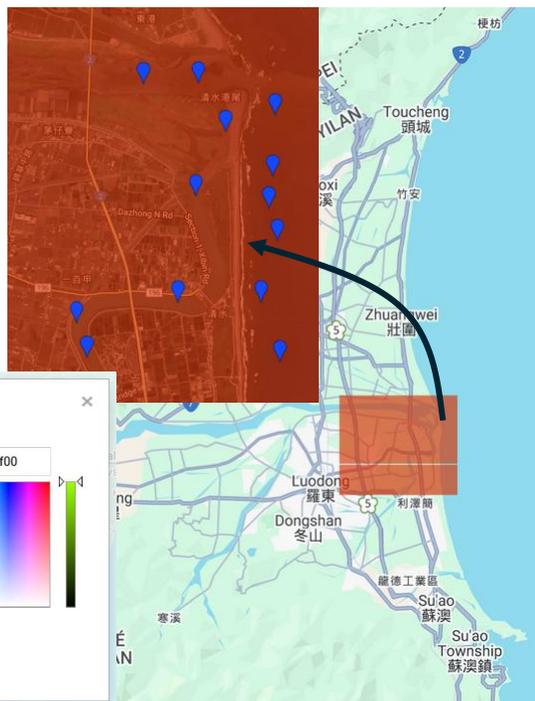
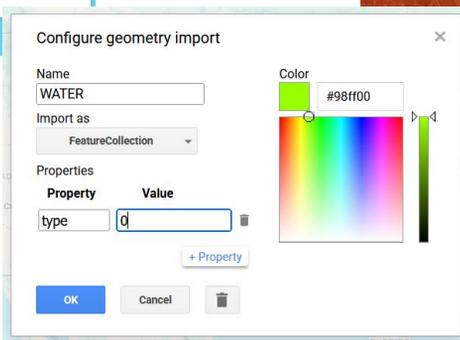
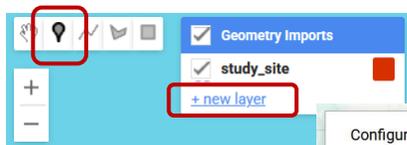


172

Case Study 3

Land cover classes: “sample” points

Use the geometry tool to create new points for each land cover class



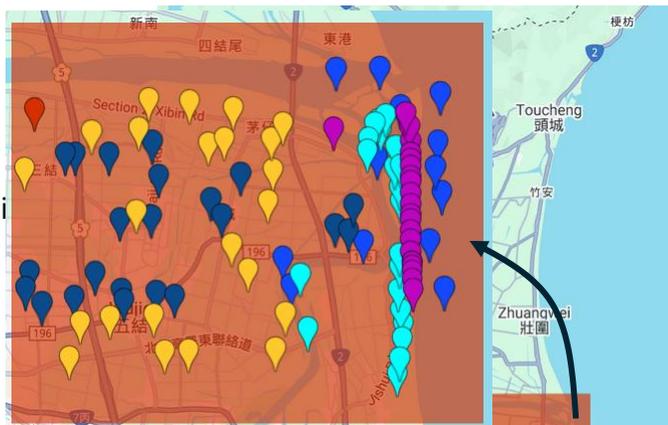
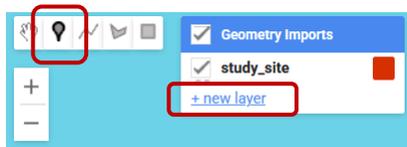
- WATER 0
- BUILTUP 1
- FARMS 2
- NATURALVEG 3
- BARESOIL 4

173

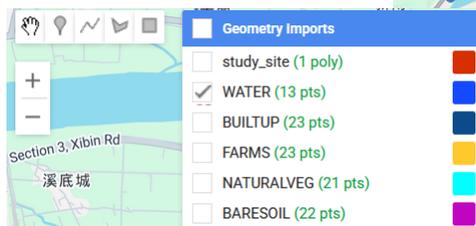
Case Study 3

Land cover classes: “sample” points

Use the geometry tool to create new points for each land cover class



- WATER 0
- BUILTUP 1
- FARMS 2
- NATURALVEG 3
- BARESOIL 4



174

Case Study 3

Spatial information

USGS Landsat 8 Level 2, Collection 2, Tier 1 🔍 📄 ⌵

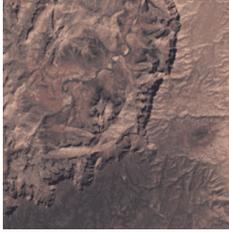
🔹 AI-generated Key Takeaways

Description Bands

Pixel Size
30 meters

Bands

Name	Units	Min	Max	Scale	Offset	Pixel Size	Wavelength	Description
SR_B1		1	65455	2.75e-05	-0.2	meters	0.435-0.451 μm	Band 1 (ultra blue, coastal aerosol) surface reflectance
SR_B2		1	65455	2.75e-05	-0.2	meters	0.452-0.512 μm	Band 2 (blue) surface reflectance
SR_B3		1	65455	2.75e-05	-0.2	meters	0.533-0.590 μm	Band 3 (green) surface reflectance
SR_B4		1	65455	2.75e-05	-0.2	meters	0.636-0.673 μm	Band 4 (red) surface reflectance
SR_B5		1	65455	2.75e-05	-0.2	meters	0.851-0.879 μm	Band 5 (near infrared) surface reflectance



Dataset Availability
2013-03-18T15:58:14Z–
2025-09-30T13:38:04.421000Z

Dataset Provider
[USGS](#)

Earth Engine Snippet
ee.ImageCollection("LANDSAT/LC08/C02/T1_L2") [🔗](#)

Revisit Interval
16 Days

Tags

cfmask cloud

satellite-imagery

175

Case Study 3

Spatial information

```
var landsat = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
  .filterBounds(study_site)
  .filterDate('2023-01-01', '2023-05-31')
  .sort('CLOUD_COVER')
  .first();
```

176

Case Study 3

Spatial information

```
var landsat_vis = {
  bands: ['SR_B4', 'SR_B3', 'SR_B2'],
  min: 7000,
  max: 15000
};
```

Name	Units	Min	Max	Scale	Offset	Pixel Size	Wavelength	Description
SR_B1		1	65455	2.75e-05	-0.2	meters	0.435-0.451 μm	Band 1 (ultra blue, coastal aerosol) surface reflectance
SR_B2		1	65455	2.75e-05	-0.2	meters	0.452-0.512 μm	Band 2 (blue) surface reflectance
SR_B3		1	65455	2.75e-05	-0.2	meters	0.533-0.590 μm	Band 3 (green) surface reflectance
SR_B4		1	65455	2.75e-05	-0.2	meters	0.636-0.673 μm	Band 4 (red) surface reflectance
SR_B5		1	65455	2.75e-05	-0.2	meters	0.851-0.879 μm	Band 5 (near infrared) surface reflectance

177

Case Study 3

Spatial information

```
Map.centerObject(study_site, 12);
Map.addLayer(landsat, landsat_vis, 'L8 image');
```

178

Case Study 3

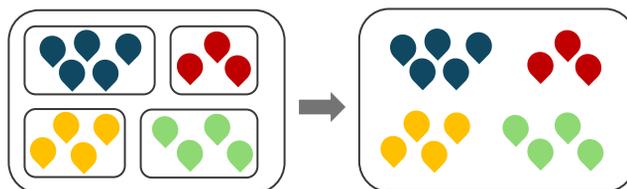
Training data

```
var training_points = ee.FeatureCollection([WATER, BUILTUP,
FARMS, NATURALVEG, BARESOIL]).flatten();
```

```
print(training_points);
```

Transforms a FeatureCollection
of FeatureCollection into a
FeatureCollection

 Feature of FeatureCollection
"WATER"



179

Case Study 3

Sampling

```
var bands = ['SR_B1', 'SR_B2', 'SR_B3', 'SR_B4', 'SR_B5',
'SR_B6', 'SR_B7', 'ST_B10'];
```

Name	Units	Min	Max	Scale	Offset	Pixel Size	Wavelength	Description
SR_B1		1	65455	2.75e-05	-0.2	meters	0.435-0.451 µm	Band 1 (ultra blue, coastal aerosol) surface reflectance
SR_B2		1	65455	2.75e-05	-0.2	meters	0.452-0.512 µm	Band 2 (blue) surface reflectance
SR_B3		1	65455	2.75e-05	-0.2	meters	0.533-0.590 µm	Band 3 (green) surface reflectance
SR_B4		1	65455	2.75e-05	-0.2	meters	0.636-0.673 µm	Band 4 (red) surface reflectance
SR_B5		1	65455	2.75e-05	-0.2	meters	0.851-0.879 µm	Band 5 (near infrared) surface reflectance

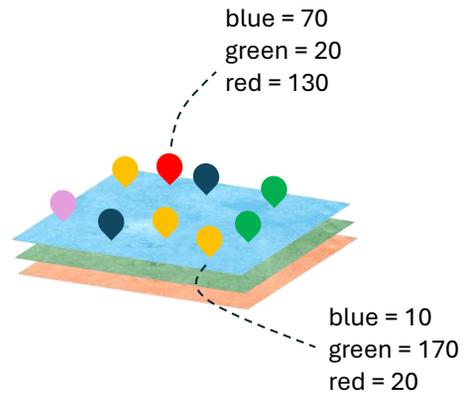
180

Case Study 3

Sampling

```
var sample = landsat.select(bands)
  .sampleRegions({
    collection: training_points,
    properties: ['type'],
    scale: 30
  });

print(sample);
```

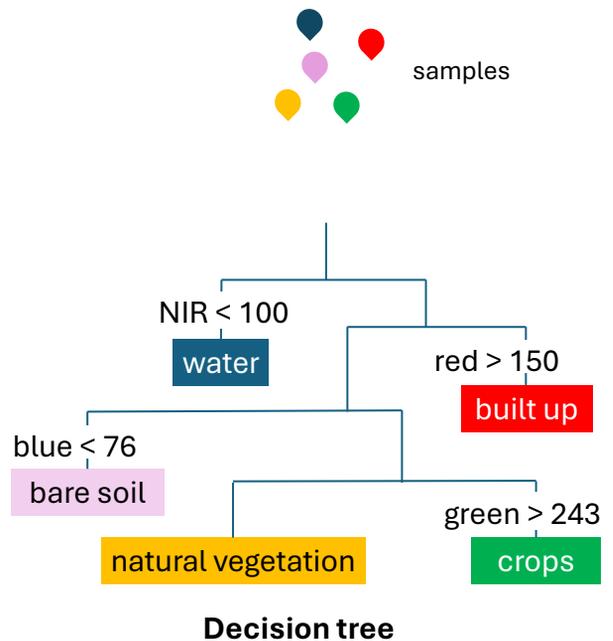


181

Case Study 3

Train model

```
ee.Classifier
ee.Classifier.amnhMaxent(cat...)
ee.Classifier.decisionTree(tree...)
ee.Classifier.decisionTreeEnsemble(...)
ee.Classifier.libsvm(decisionPr...)
ee.Classifier.load(id)
ee.Classifier.minimumDistance(...)
ee.Classifier.smileCart(maxNo...)
ee.Classifier.smileGradientTree(...)
ee.Classifier.smileKNN(k, sear...)
ee.Classifier.smileNaiveBayes(...)
ee.Classifier.smileRandomForest(...)
ee.Classifier.spectralRegion(c...)
confusionMatrix()
explain()
```



182

Case Study 3

Train model

```
var classifier = ee.Classifier.smileCart().train({
  features: sample,
  classProperty: 'type',
  inputProperties: bands
});
```

183

Case Study 3

Classification

```
var classification = landsat.select(bands)
  .classify(classifier);

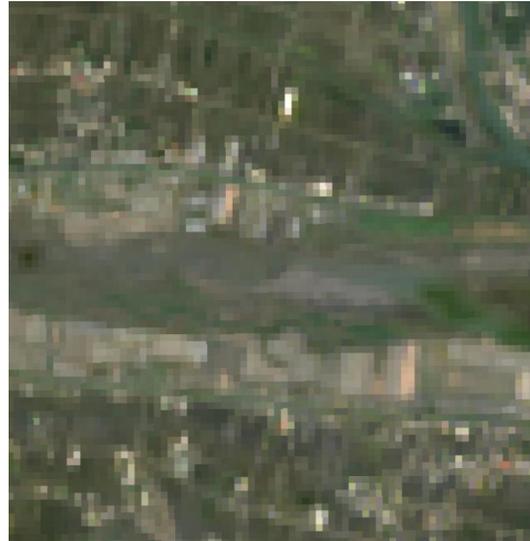
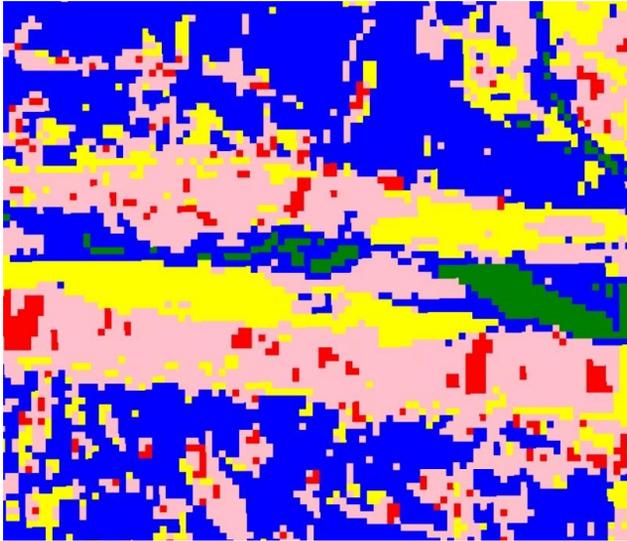
var classification_vis = {
  min: 0,
  max: 4,
  palette: ['green', 'red', 'blue', 'yellow', 'pink']
};

Map.addLayer(classification, classification_vis,
  'classificationsupervCART');
```

184

Case Study 3

Classification



185

Case Study 3

Classification

[Export the land cover map to Google Drive](#)

186

Case Study 3

Evaluating the classification

For each class (e.g., water)

$$\bullet \text{ Accuracy} = \frac{TP+TN}{Total} = \frac{50+35}{100}$$

$$\bullet \text{ Producer accuracy} = \frac{TP}{TP+FN} = \frac{50}{50+5}$$

$$\bullet \text{ User accuracy} = \frac{TP}{TP+FP} = \frac{50}{50+10}$$

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	True positive	False positive
	Negative	False negative	True negative

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	50	10
	Negative	5	35

187

Case Study 3

Evaluating the classification

For each class (e.g., water)

$$\bullet \text{ Accuracy} = \frac{TP+TN}{Total} = \frac{50+35}{100}$$

$$\bullet \text{ Producer accuracy} = \frac{TP}{TP+FN} = \frac{50}{50+5}$$

$$\bullet \text{ User accuracy} = \frac{TP}{TP+FP} = \frac{50}{50+10}$$

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	True positive	False positive
	Negative	False negative	True negative

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	50	10
	Negative	5	35

188

Case Study 3

Evaluating the classification

For each class (e.g., water)

$$\bullet \text{ Accuracy} = \frac{TP+TN}{Total} = \frac{50+35}{100}$$

$$\bullet \text{ Producer accuracy} = \frac{TP}{TP+FN} = \frac{50}{50+5} \quad \text{---} \rightarrow \text{Omission error} = 100 - \text{Producer}$$

$$\bullet \text{ User accuracy} = \frac{TP}{TP+FP} = \frac{50}{50+10} \quad \text{---} \rightarrow \text{Commission error} = 100 - \text{User}$$

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	True positive	False positive
	Negative	False negative	True negative

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	50	10
	Negative	5	35

189

Case Study 3

Evaluating the classification

For each class (e.g., water)

$$\bullet \text{ Accuracy} = \frac{TP+TN}{Total} = \frac{50+35}{100}$$

$$\bullet \text{ Producer accuracy} = \frac{TP}{TP+FN} = \frac{50}{50+5} \quad \text{---} \rightarrow \text{Omission error} = 100 - \text{Producer}$$

$$\bullet \text{ User accuracy} = \frac{TP}{TP+FP} = \frac{50}{50+10} \quad \text{---} \rightarrow \text{Commission error} = 100 - \text{User}$$

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	True positive	False positive
	Negative	False negative	True negative

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	50	10
	Negative	5	35

Coefficient Kappa [-1; 1]

190

Case Study 3

Evaluating the classification

Split of the dataset into:

- Training set
- Validation set

A new classification is produced using the training set and tested using the validation set.

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	True positive	False positive
	Negative	False negative	True negative

blue = 70
green = 20
red = 130



191

Case Study 3

Evaluating the classification

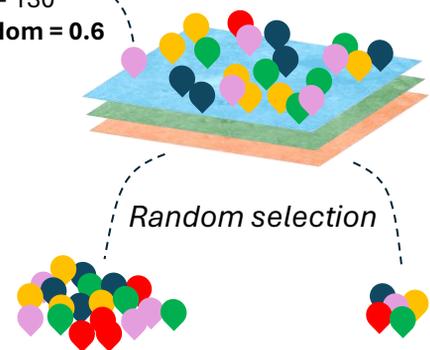
Split of the dataset into:

- Training set (**80%**)
- Validation set (**20%**)

A new classification is produced using the training set and tested using the validation set.

		Observed values	
		Positive	Negative
Predicted values (classification)	Positive	True positive	False positive
	Negative	False negative	True negative

blue = 70
green = 20
red = 130
random = 0.6



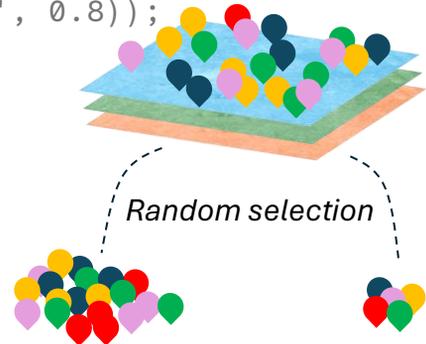
192

Case Study 3

Evaluating the classification

```
var traintest = sample.randomColumn();
var train_group =
traintest.filter(ee.Filter.lt('random', 0.8));
var test_group =
traintest.filter(ee.Filter.gte('random', 0.8));
```

```
print(train_group);
print(test_group);
```



193

Case Study 3

Evaluating the classification

```
var classifier2 = ee.Classifier.smileCart().train({
  features: train_group,
  classProperty: 'type',
  inputProperties: bands
});
```

194

Case Study 3

Evaluating the classification

```
var confusion_matrix = test_group.classify(classifier2)
    .errorMatrix({
        actual: 'type',
        predicted: 'classification'
    });
```

Computes a 2D error matrix for a collection by comparing two columns of a collection: one containing the actual values, and one containing predicted values. The values are expected to be small contiguous integers, starting from 0. Axis 0 (the rows) of the matrix correspond to the actual values, and Axis 1 (the columns) to the predicted values.

Usage	Returns
<code>FeatureCollection.errorMatrix(actual, predicted, order)</code>	ConfusionMatrix

195

Case Study 3

Evaluating the classification

```
print('Confusion Matrix:', confusion_matrix);
print('Accuracy:', confusion_matrix.accuracy());
print('Producer Accuracy:',
confusion_matrix.producersAccuracy());
print('User Accuracy:', confusion_matrix.consumersAccuracy());
print('Kappa:', confusion_matrix.kappa());
```

Confusion Matrix:

▼ List (5 elements)

```
▶ 0: [3,0,4,0,0]
▶ 1: [0,2,0,0,0]
▶ 2: [1,0,7,0,0]
▶ 3: [0,0,0,6,0]
▶ 4: [0,1,0,0,3]
```

▼ ee.ConfusionMatrix

ee.ConfusionMatrix(array, ord...

accuracy()

array()

consumersAccuracy()

fscore(beta)

kappa()

order()

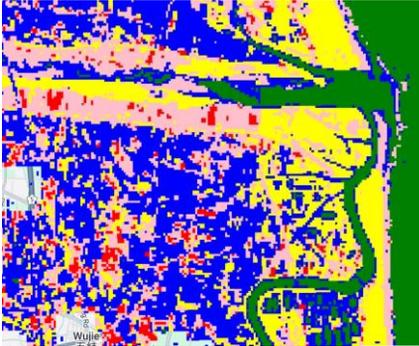
producersAccuracy()

-

196

Case Study 3

Share of natural vegetation cover



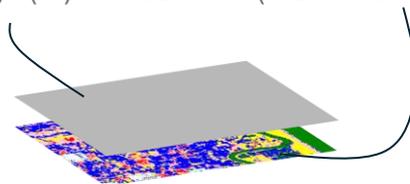
Estimate the % of natural vegetation in the study area

197

Case Study 3

Share of natural vegetation cover

```
var classification2 = ee.Image(1).addBands(classification);
```



```
var vegcover = classification2.reduceRegion(
  ee.Reducer.sum().group({groupField: 1}), study_site, 30);
```

```
print(vegcover, "vegcover");
```

198

Case Study 3

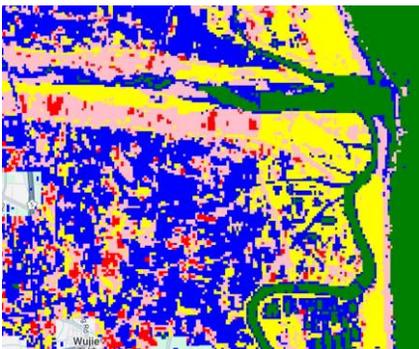
Share of natural vegetation cover

```
var pixelcount_class = ui.Chart.image.byClass({
  image: classification2,
  classBand: 'classification',
  region: study_site,
  scale: 30,
  reducer: ee.Reducer.count()
});
print(pixelcount_class);
```

199

Case Study 3

NDVI trends



Estimate the % of
natural vegetation
in the study area



Follow the NDVI
change in
cultivated areas

200

Case Study 3

NDVI trends

```
var HLSL_1525 = ee.ImageCollection("NASA/HLS/HLSL30/v002")
  .filterDate('2015-01-01', '2025-01-01')
  .filterBounds(study_site)
  .select(['B4', 'B5'])
  .filter(ee.Filter.lt('CLOUD_COVERAGE', 50));
```

201

Case Study 3

NDVI trends

```
var HLSL_1525 = ee.ImageCollection("NASA/HLS/HLSL30/v002")
  .filterDate('2015-01-01', '2025-01-01')
  .filterBounds(study_site)
  .select(['B4', 'B5'])
  .filter(ee.Filter.lt('CLOUD_COVERAGE', 50));
```

- Compute NDVI for each Image between 2000 and 2024
- Focus only on the cultivated areas

202

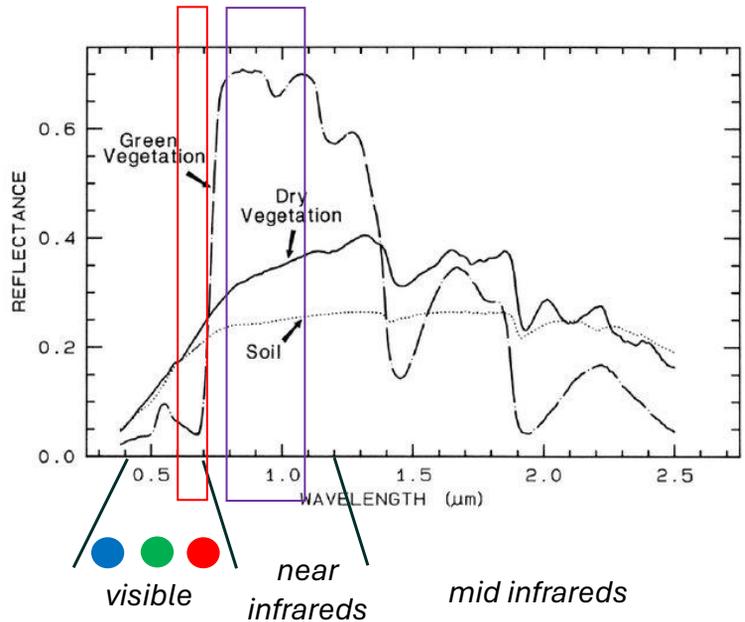
Case Study 3

NDVI trends

Using reflectance change at different wavelengths to monitor vegetation health:

$$NDVI = \frac{NIR - R}{NIR + R}$$

near infrared
red



203

Case Study 3

NDVI trends

Custom functions + map ()

204

Case Study 3

NDVI trends

Custom functions + `map()`

The function contains instructions to modify, create objects.
`map()` applies the function to each element of a collection, list.

ImageCollection



205

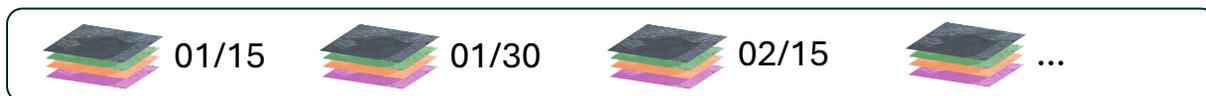
Case Study 3

NDVI trends

Custom functions + `map()`

The function contains instructions to modify, create objects.
`map()` applies the function to each element of a collection, list.

ImageCollection



`function(Compute NDVI, ...)`

New ImageCollection



206

Case Study 3

NDVI trends

```

var farmsfun = function(image) {
  var imagemasked = image.updateMask(classification.eq(2));
  var ndvi = imagemasked.normalizedDifference(['B5', 'B4'])
    .rename('ndvi');
  return imagemasked.addBands(ndvi);
};

```

element

NIR

R

207

Case Study 3

NDVI trends

```

var farmsfun = function(image) {
  var imagemasked = image.updateMask(classification.eq(2));
  var ndvi = imagemasked.normalizedDifference(['B5', 'B4'])
    .rename('ndvi');
  return imagemasked.addBands(ndvi);
};

var HLSL_1525_NDVI = HLSL_1525.map(farmsfun);
print(HLSL_1525_NDVI, 'HLSL_1525_NDVI');

```

208

Case Study 3

NDVI trends

```
var graph = ui.Chart.image.series({
  imageCollection: HLSL_1525_NDVI.select('ndvi'),
  region: study_site,
  reducer: ee.Reducer.median(),
  scale: 30
});
print(graph);
```

209

Case Study 3

NDVI trends

```
var graph = ui.Chart.image.series({
  imageCollection: HLSL_1525_NDVI.select('ndvi'),
  region: study_site,
  reducer: ee.Reducer.median(),
  scale: 30
});
print(graph);
```

Test other reducers (e.g. `ee.Reducer.min()`, ...)

210

Case Study 3

NDVI trends

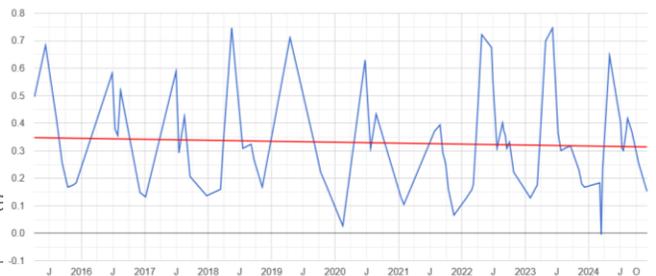
```
var graph_trend = ui.Chart.image.series({
  imageCollection: HLSL_1525_NDVI.select('ndvi'),
  region: study_site,
  reducer: ee.Reducer.median(),
  scale: 30
}).setOptions({
  trendlines: {0: {type: 'linear', color: 'red',
  visibleInLegend: true}}
});
print(graph_trend);
```

211

Case Study 3

NDVI trends

```
var graph_trend = ui.Chart.image.series({
  imageCollection: HLSL_1525_NDVI.select('ndvi'),
  region: study_site,
  reducer: ee.Reducer.median(),
  scale: 30
}).setOptions({
  trendlines: {0: {type: 'linear', color: 'red',
  visibleInLegend: true}}
});
print(graph_trend);
```



212

Important notes

Collaborating with GEE

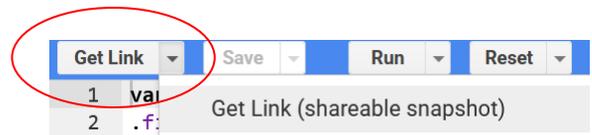
- Sharing scripts
- Sharing data (assets)
- Sharing repositories
- Tracking script changes



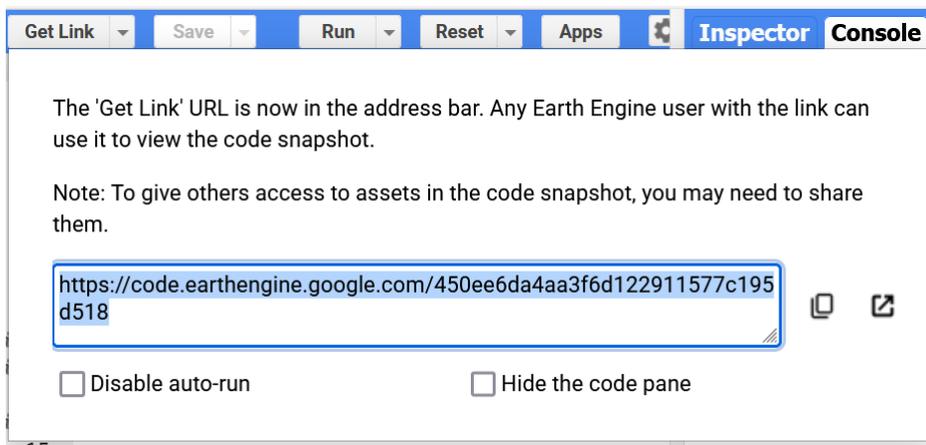
213

Important notes

Collaborating with GEE



To share a script:



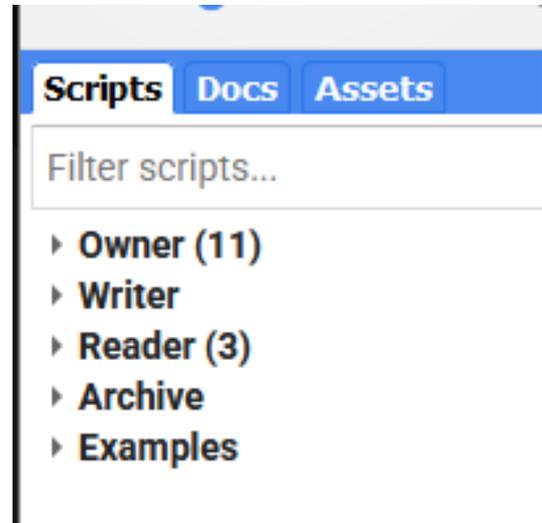
214

Important notes

Collaborating with GEE

To share a repository

Owner/Writer/Reader



215

Important notes

Collaborating with GEE

Share Repo: users/jauneatemps/course

jauneatemps@gmail.com Owner ▾

Email or domain Reader ▾ ADD ACCESS

Anyone can read

Reader

Writer

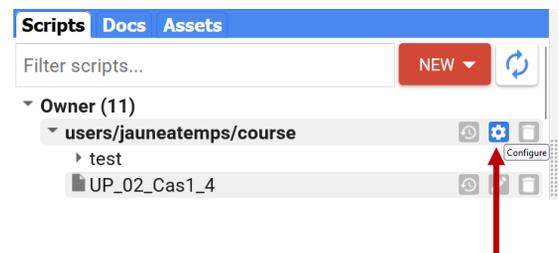
Users with access to the repository can add it to the Code Editor using:

https://code.earthengine.google.com/?accept_repo=users/jauneatemps/course

Clone the [Git](#) repository by running the following command in a terminal:

```
git clone https://earthengine.google.com/users/jauneatemps/course
```

DONE



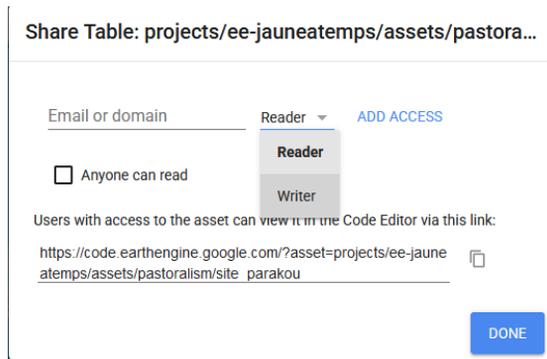
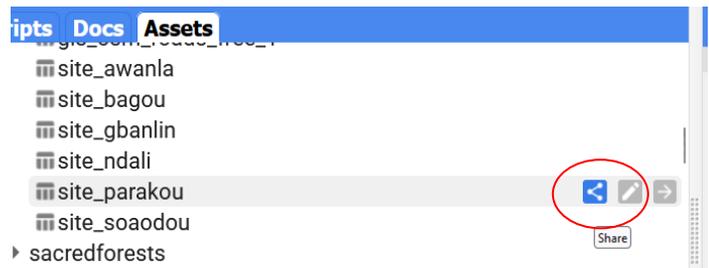
Writer allows script editing

216

Important notes

Collaborating with GEE

To share assets:



217

Important notes

Collaborating with GEE

To use other user's assets:

```
var Lcmap = ee.Image('projects/ee-jauneatemps/assets/pastoralism/OBSYDYA_OS_2023_L1_Awanla')
```

Many datasets not included in the official GEE catalogue, introduced in peer-reviewed articles:

e.g. *Monitoring road development in Congo Basin forests with multi-sensor satellite imagery and deep learning* (Slagter et al., 2024)

And in other catalogues such as <https://gee-community-catalog.org/projects/>

218

Important notes

Collaborating with GEE

Track changes (repository)

▼ Owner (11)

▼ users/jauneatemps/course

Revision history:

Time	Author	Message
Jan 30, 11:30	jauneatemps	Updated UP_03_deternding (added 1...
Jan 30, 10:36	jauneatemps	Updated UP_03_detectionchangeme...
Jan 30, 10:30	jauneatemps	Updated UP_03_detectionchangeme...
Jan 30, 09:56	jauneatemps	Updated UP_03_deternding (added 3...
Jan 30, 09:52	jauneatemps	Updated UP_03_deternding (added 1...
Jan 30, 09:45	jauneatemps	Updated UP_03_deternding (added 2...
Jan 30, 09:45	jauneatemps	Updated UP_03_deternding (added 9...
Jan 29, 14:42	jauneatemps	Updated UP_03_deternding (added 4...
Jan 29, 14:35	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 14:34	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 14:34	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 14:17	jauneatemps	Updated UP_03_deternding (added 2...
Jan 29, 14:16	jauneatemps	Updated UP_03_deternding (added 3...
Jan 29, 14:05	jauneatemps	Updated UP_03_deternding (added 4...
Jan 29, 11:17	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 11:04	jauneatemps	Updated UP_03_deternding (added 2...
Jan 29, 11:03	jauneatemps	Updated UP_03_deternding (added 2...
Jan 29, 11:02	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 10:32	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 10:32	jauneatemps	Updated UP_03_deternding (added 4...
Jan 29, 10:28	jauneatemps	Updated UP_03_deternding (added 5...
Jan 29, 10:19	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 09:56	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 09:55	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 09:52	jauneatemps	Updated UP_03_deternding (added 1...
Jan 29, 09:25	jauneatemps	Updated UP_03_deternding (added 1...
Jan 28, 17:23	jauneatemps	Updated UP_03_deternding (added 2...
Jan 28, 17:22	jauneatemps	Updated UP_03_deternding (added 1...
Jan 28, 17:22	jauneatemps	Updated UP_03_deternding (added 1...
Jan 28, 17:20	jauneatemps	Updated UP_03_deternding (added 1...
Jan 28, 17:19	jauneatemps	Updated UP_03_deternding (added 4...

ending (added 1 ...
ctionchangeme... **Revert**
ctionchangeme... **Done**

219

Important notes

Collaborating with GEE

Track changes (script)

Revision history: UP_03_detectionchangement

Time	Author	Message
Jan 30, 10:36	jauneatemps	Updated UP_03_detectionchangeme...
Jan 30, 10:30	jauneatemps	Updated UP_03_detectionchangeme... Compare
Jan 27, 17:20	jauneatemps	Updated UP_03_detectionchangeme...
Jan 27, 17:15	jauneatemps	Updated UP_03_detectionchangeme...
Jan 27, 17:10	jauneatemps	Updated UP_03_detectionchangeme...

```

< Jan 30, 10:30 vs latest commit
50 50
51 51 var thresholdGain = 0.05;
52 52 var thresholdLoss = -0.05;
53 53
54 54 var diffClassified = ee.Image(0);
55 55
56 56 var diffClassified = diffClassified.where(deltaNDVI.lte(thresholdLoss), 2);
57 57 var diffClassified = diffClassified.where(deltaNDVI.gte(thresholdGain), 1);
58 58
59 59 var methodeAlternative = deltaNDVI.gte(thresholdGain).where(deltaNDVI.lte(thresholdLoss),
60 60
61 61 var changeVis = {
62 62   palette: 'blue,red',
63 63   min: 1,
64 64   max: 2
65 65 };
66 66
67 67 Map.addLayer(diffClassified.selfMask(),
68 68   changeVis,
69 69   'change classified by threshold');
70 70
71 71 Map.addLayer(methodeAlternative.selfMask(),
72 72   changeVis,
73 73   'change classified by threshold 2');
74 74
75 75
76 76
  
```

CANCEL REVERT

220

Important notes

Collaborating with GEE

Track changes (script)

Revision history: UP_03_detectionchangement

Time	Author	Message
Jan 30, 10:36	jauneatemps	Updated UP_03_detectionchangeme...
Jan 30, 10:30	jauneatemps	Updated UP_03_detectionchangeme... Compare
Jan 27, 17:20	jauneatemps	Updated UP_03_detectionchangeme...
Jan 27, 17:15	jauneatemps	Updated UP_03_detectionchangeme...

```

< Jan 30, 10:30 vs latest commit
50 50
51 51 var thresholdGain = 0.05;
52 52 var thresholdLoss = -0.05;
53 53
54 54 var diffClassified = ee.Image(0);
55 55
56 56 var diffClassified = diffClassified.where(deltaNDVI.lte(thresholdLoss), 2);
57 57 var diffClassified = diffClassified.where(deltaNDVI.gte(thresholdGain), 1);
58 58
59 59 var methodeAlternative = deltaNDVI.gte(thresholdGain).where(deltaNDVI.lte(thresholdLoss),
60 60
61 61 var changeVis = {
62 62   palette: 'blue,red',
63 63   min: 1,
64 64   max: 2
65 65 };
66 66
67 67 Map.addLayer(diffClassified.selfMask(),
68 68   changeVis,
69 69   'change classified by threshold');
70 70
71 71 Map.addLayer(methodeAlternative.selfMask(),
72 72   changeVis,
73 73   'change classified by threshold 2');
74 74
75 75
76 76
  
```

CANCEL REVERT

221

Important notes

Collaborating with GEE: Import other functions

require

Use functions from other users, or developed by yourself

222

Important notes

Collaborating with GEE: Import other functio

Create a new script
'S2cloudmask' in
'NTNUGEE' repository



Harmonized Sentinel-2 MSI: MultiSpectral Instrument, Level-1C

Disponibilité des ensembles de données
2015-06-27T00:00:00Z-2025-01-31T17:26:43Z

Fournisseur d'ensembles de données
Union européenne/ESA/Copernicus

Extrait Earth Engine
ee.ImageCollection("COPERNICUS/S2_HARMONIZED")

Intervalle de nouvelle visite
5 jours

Tags
copernicus esa es mei radance
sentinel

```
exports.maskS2clouds = function (image) {
  var qa = image.select('QA60');
  // Bits 10 and 11 are clouds and cirrus, respectively.
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;
  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
    .and(qa.bitwiseAnd(cirrusBitMask).eq(0));
  return image.updateMask(mask).divide(10000);
}
```

223

Important notes

Collaborating with GEE: Import other functions

```
var cloudmaskfun = require(
  'users/jauneatemps/NTNUGEE:S2cloudmask');

var S2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
  .filterDate('2024-10-01', '2024-12-01')
  .filterBounds(geometry)
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
  .map(cloudmaskfun.maskS2clouds);
```

224

Important notes

Collaborating with GEE

- Scripts
- Data (assets)
- Repositories

- Reader / Writer

Git: <https://earthengine.google.com> (other user's scripts)

225

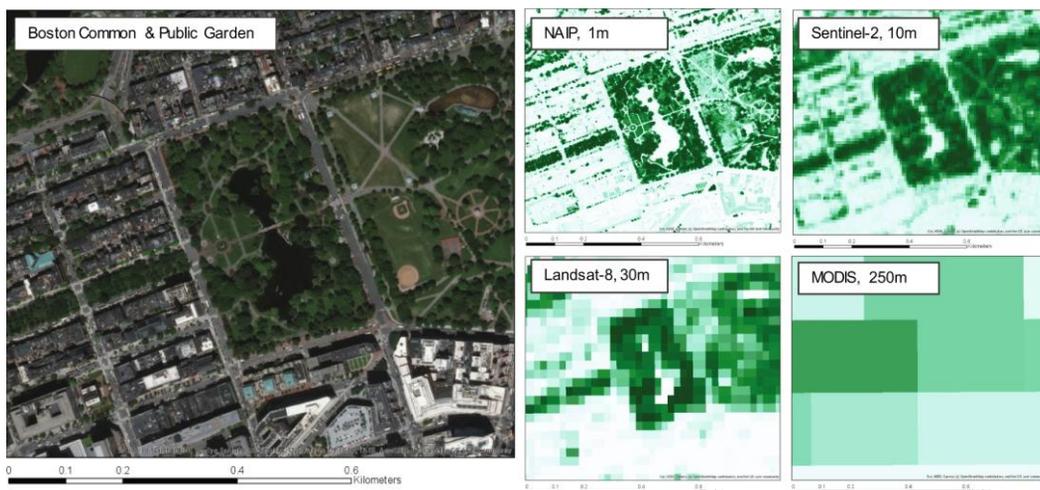
Important notes

Spatial resolution and projection

226

Important notes

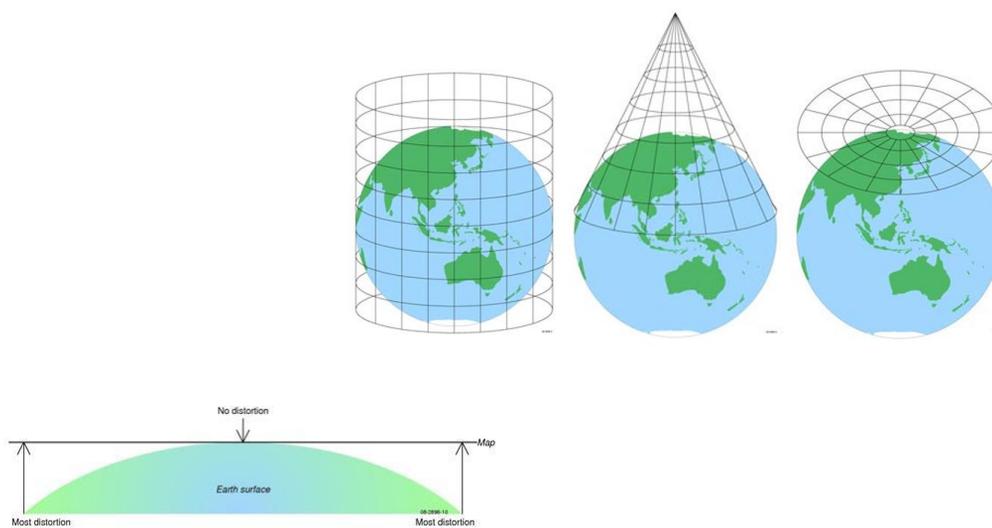
Spatial resolution and projection



227

Important notes

Spatial resolution and **projection**



ICSM.gov.au; psu.edu

228

Important notes

Spatial resolution and projection

GEE handle spatial resolution based on the user's requirements:

- Display an image
- Extract a raster
- Compute statistics
- ...

Specified using *scale*.

For example in `ee.Image.reduceRegions`

`ee.Image.reduceRegions`  

Apply a reducer over the area of each feature in the given collection.

The reducer must have the same number of inputs as the input image has bands.

Returns the input features, each augmented with the corresponding reducer outputs.

Usage

```
Image.reduceRegions(collection, reducer, scale, crs, crsTransform, tileScale)
```

229

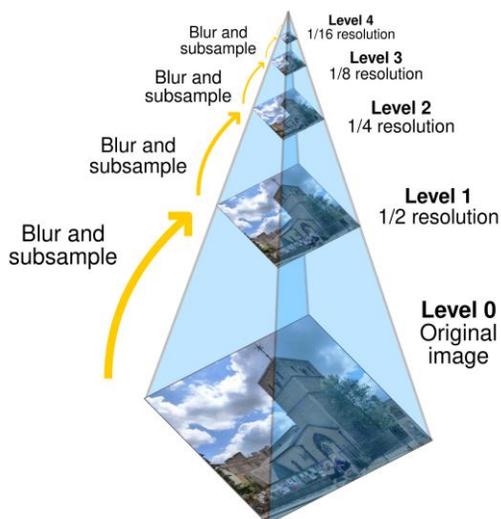
Important notes

Spatial resolution and projection

Pyramiding policy of GEE

Resampling:

- Mean (continuous variables)
- One of the existing values (discrete)



230

Important notes

Spatial resolution and projection

Pyramiding policy of GEE

Resampling:

- Mean (continuous variables)
- One of the existing values (discrete)

Upload a new image asset

Source files

 Please drag and drop or select files for this asset.
 Allowed extensions: tiff, tif, json, tfrecord or tfrecord.gz.

Asset ID
 projects/ee-jauneatemps/assets/ +

Properties
 Metadata properties about the asset which can be edited during asset upload and after ingestion. The "system:time_start" property is used as the primary date of the asset.

Advanced options (circled in red)

Pyramiding policy
 MEAN

Resampling method
 None

[Learn more](#) about how uploaded files are processed.

231

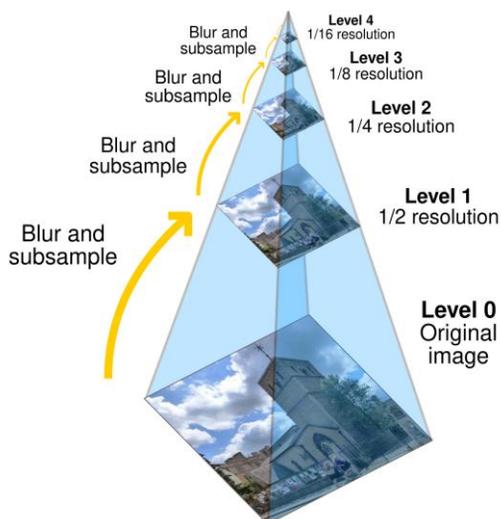
Important notes

Spatial resolution and projection

Pyramiding policy of GEE

To display images on the map, or to compute statistics, GEE use the map's zoom level to select a level of the pyramid, if no *scale* value is defined in the script!

The desired output will lead GEE to use a specific pyramid level.



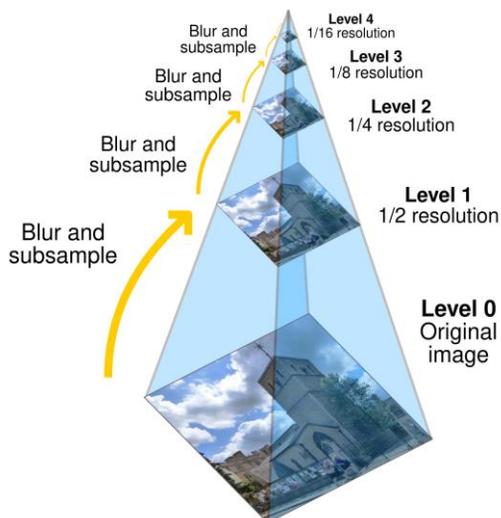
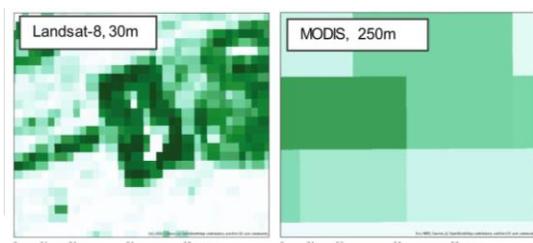
232

Important notes

Spatial resolution and projection

Pyramiding policy of GEE

If *scale* is used, GEE identify the closest pyramid level and then resample it to fit the scale.



233

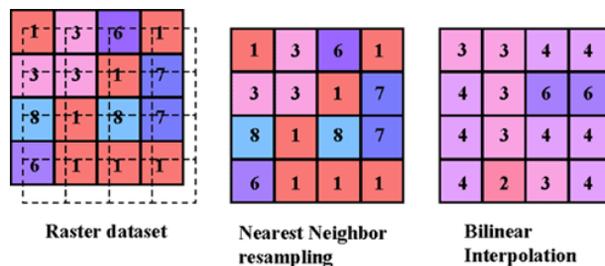
Important notes

Spatial resolution and projection

Pyramiding policy of GEE

If *scale* is used, GEE identify the closest pyramid level and then resample it to fit the scale.

The default resample approach is *nearest neighbor*



documentation ArcGIS

234

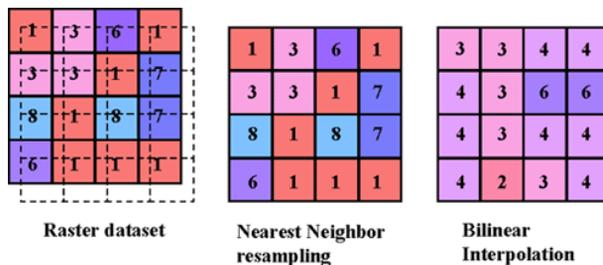
Important notes

Spatial resolution and projection

Pyramiding policy of GEE

If *scale* is used, GEE identify the closest pyramid level and then resample it to fit the scale.

The default resample approach is *nearest neighbor*



```
Image.resample('bilinear')
Image.resample('bicubic')
```

documentation ArcGIS

235

Important notes

Spatial resolution and projection

Open

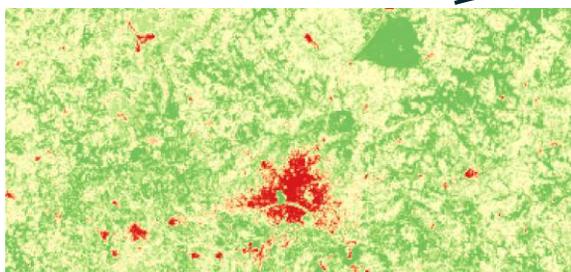
236

Important notes

Spatial resolution and projection

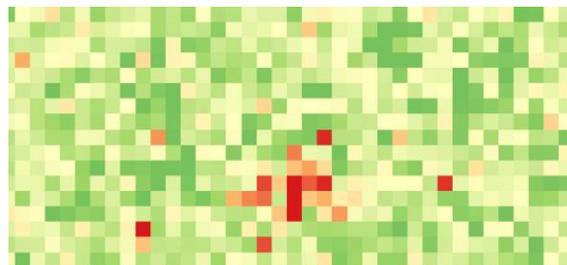
100m to 1000m resolution

Without specifying a resampling method (= nearest neighbor)

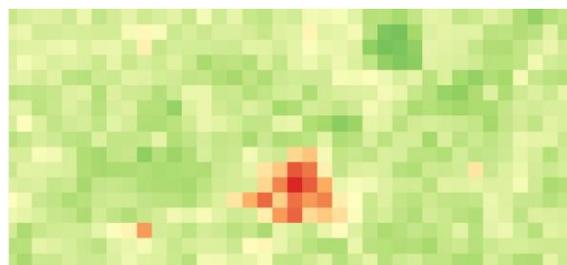


100m

Using `.reduceResolution()` and `mean` as the reducer



1000m



237

Important notes

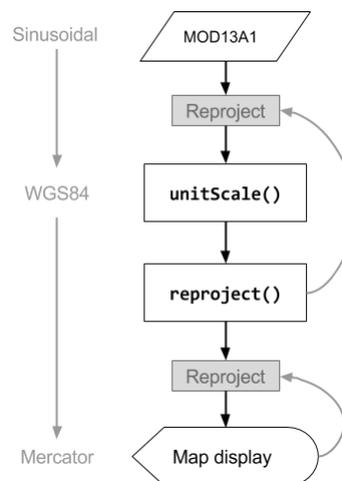
Spatial resolution and projection

Spatial resolution and projection can be handled by GEE even if they are specified at the end of the script (e.g., statistics, export)

`/\` Displaying an Image with very high resolution can lead to errors: too much data

`/\` if you use `reproject()` based on an Image, it may be affected by the map's zoom level

`/\` don't use many reprojections

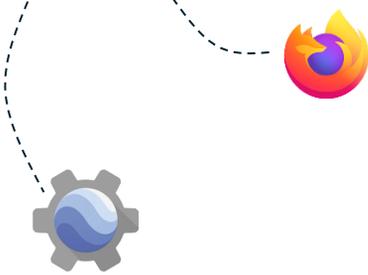


documentation GEE

238

Important notes

Server and client



Objects are either EE objects or Javascript objects

EE objects: “ee.”

239

Important notes

Server and client

```
var clientString = 'string client'
print(type of client_string)
```

```
var serverString = ee.String('not a string')
print(type of server_string)
```

`ee.x(content)`: the content is known by the client unless it is requested (using `print()` , etc)

240

Important notes

Server and client

```
var clientString = 'string client'
print(type of client_string)
```

```
var serverString = ee.String('not a string')
print(type of server_string)
```

`ee.x(content)`: the content is known by the client unless it is requested (using `print()`, etc)

`getInfo()` → will pause code processing until the requested value is sent to the client

241

Important notes

Server and client

More efficient to have everything server-side

for i loops are not recommended

`.map()` instead

same for *if else*

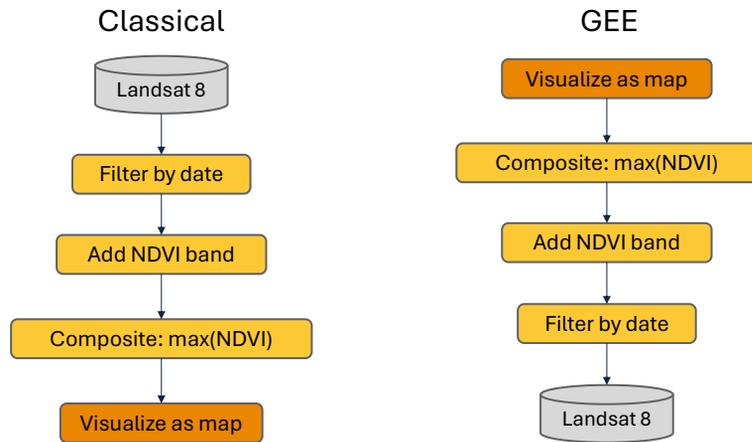
`ee.Algorithms.If()` or `Filter()` instead

242

Important notes

Server and client

Deferred execution



Michael DeWitt & Katie Friis – Geo For Good 2022

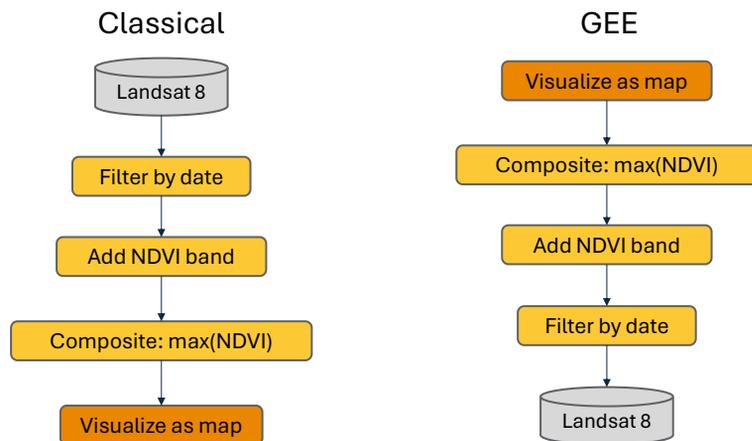
243

Important notes

Server and client

Deferred execution

- Parallelization
- Efficient
- /!\ Displayed images will depend on the map zoom level, etc.

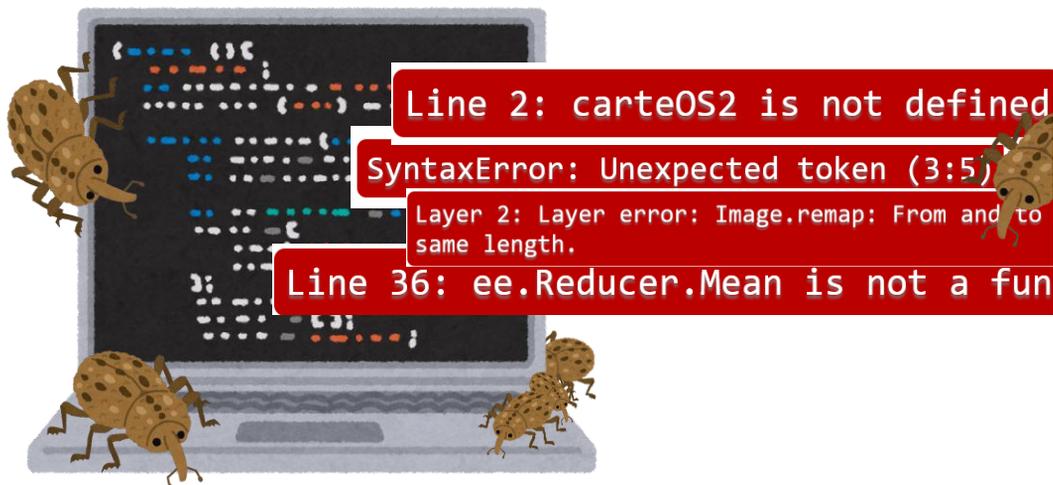


Michael DeWitt & Katie Friis – Geo For Good 2022

244

Important notes

Debugging



245

Important notes

Debugging

```

8 ▾ var carteOSremap = carteOS.remap({
9   from: fromList,
10  to: toList,
11  defaultValue: 0
12 });

```

Expected '}' to match '{' from line 8 and instead saw ')'.
 Expected ')' and instead saw ';'.
 Missing semicolon.

Syntax: missing (), {}, name of the arguments, ...

```

8 ▾ var carteOSremap = carteOS.remap({
9   fromList,
10  to: toList
11 });

```

Expected ':' and instead saw ','.

246

Important notes

Debugging

```
var carteOSremap = carteOS.remap({  
  from: fromlist,  
  to: toList,  
  defaultValue: 0  
});
```

Line 9: fromlist is not defined

247

Important notes

Debugging tools

print()

Map.addLayer()

Inspector

collection.filter()

collection.limit()

collection.first()

Common errors: Image is actually an ImageCollection, missing bands, different names, ...

248

Important notes

Debugging tools

Not recommended

- Using loops and getInfo (client-side)
- Working with very large areas, numbers...
- Unnecessary steps (converting values, ...)

249

Important notes

Debugging tools

Not recommended

- Using loops and getInfo (client-side)
- Working with very large areas, numbers...
- Unnecessary steps (converting values, ...)

User Memory Limit Exceeded	}	Too much data to handle, etc.
Computation Timeout		
Too Many Aggregations		
Internal Server Error		

250

Important notes

Debugging tools

Not recommended

- Using loops and getInfo (client-side)
- Working with very large areas, numbers...
- Unnecessary steps (converting values, ...)

User Memory Limit Exceeded

Computation Timeout

Too Many Aggregations

Internal Server Error

} Too much data to handle, etc.

--- Export rasters, smaller area...

251

Important notes

Debugging tools

- Read GEE's error messages
- print(), export...
- Use search engines, many of the errors have already been reported and solved

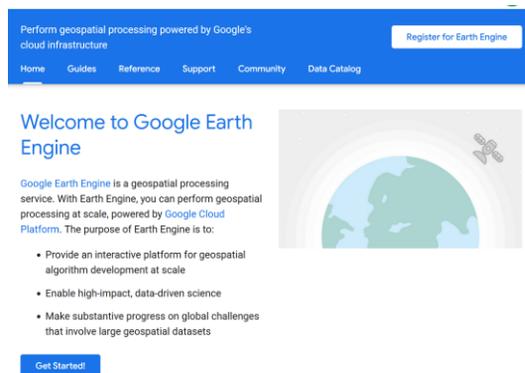
252

Important notes

Resources



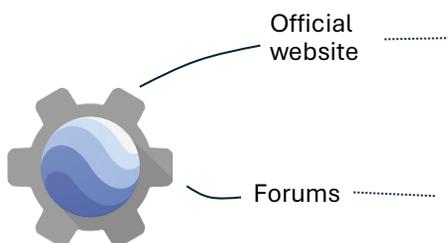
- Guides
- Tutorials
- Detailed information (+ tutorials) about specific products such as Sentinel 1, Landsat



253

Important notes

Resources

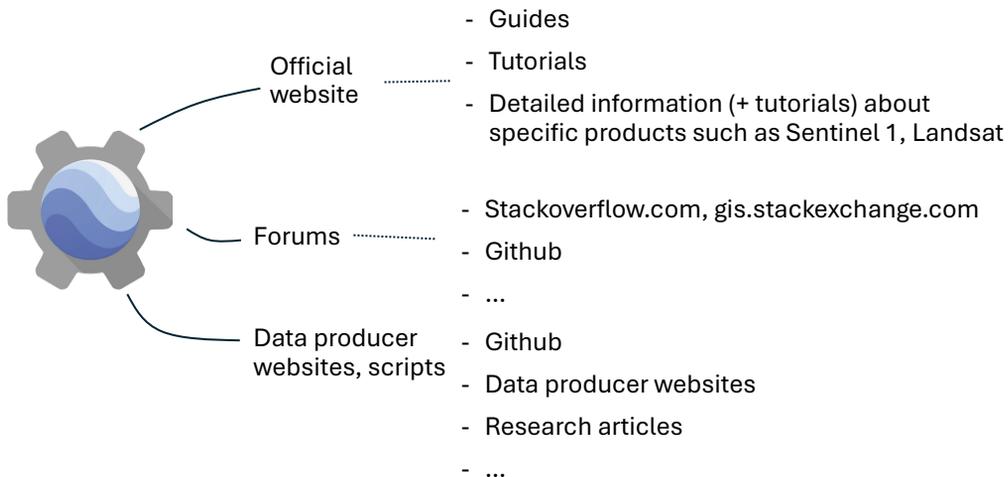


- Guides
- Tutorials
- Detailed information (+ tutorials) about specific products such as Sentinel 1, Landsat
- Stackoverflow.com, gis.stackexchange.com
- Github
- ...

254

Important notes

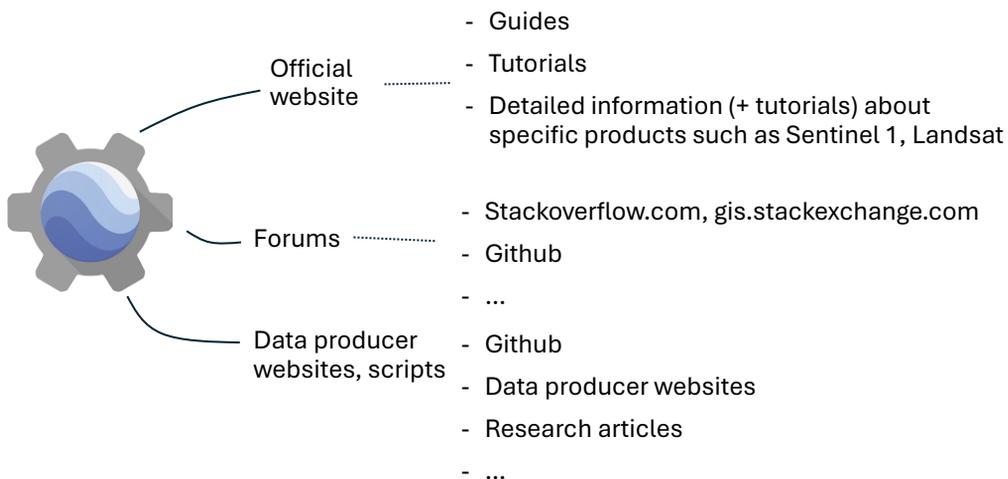
Resources



255

Important notes

Resources



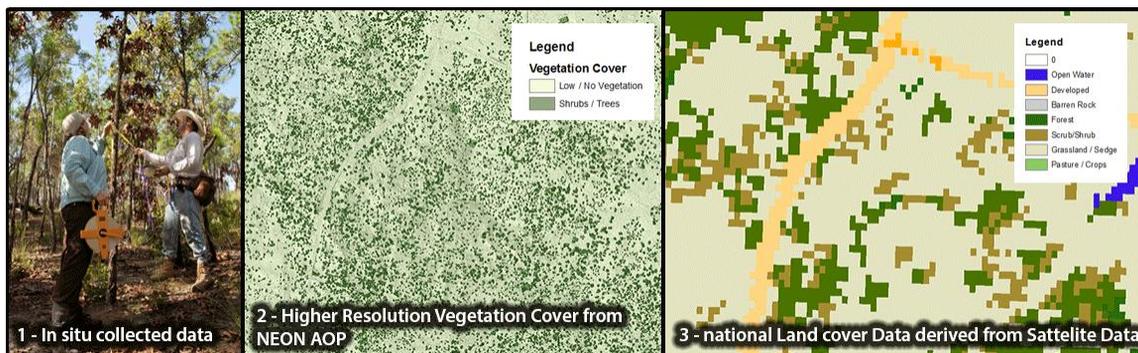
256

Important notes

Resources

NEON network (USA)

<https://www.neonscience.org/resources/learning-hub/tutorials/intro-aop-gee-image-collections>



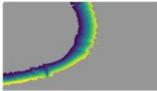
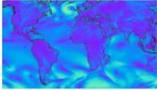
257

Important notes

Publisher Data Catalogs <https://developers.google.com/earth-engine/datasets/publisher>

Publisher Data Catalogs

Publisher Data Catalogs are created by the publisher of the dataset for use by the larger Google Earth Engine community and shared publicly as collections of Earth Engine assets. These Catalogs are not created by Google Earth Engine.

 <p>Forest Data Partnership</p> <p>4 datasets</p> <p>The Forest Data Partnership strengthens collaboration and facilitates improved global monitoring of commodity-driven deforestation, forest degradation and restoration efforts across the globe.</p>	 <p>Geoscience Australia</p> <p>6 datasets</p> <p>Geoscience Australia provides Earth observation services, expert advice, and information for decision makers. Program objectives are:</p> <ul style="list-style-type: none"> • ensure continuity of access to remotely sensed information in support of government priorities. • promote use and value of satellite imagery within government, industry and the wider community. • advance remote sensing science through extracting value and use of EO's to its full potential. • provide data stewardship to ensure access to data, and • provide national emergency coordination for disasters and emergencies. 	 <p>Land & Carbon Lab Global Pasture Watch</p> <p>3 datasets</p> <p>The Land & Carbon Lab, co-managed by the Australian Institute of Land Use Science (AILUS) and the Insoo Earth Fund, established the Global Pasture Watch research consortium. The consortium, which is made-up of experts in geospatial monitoring, machine learning, ecology and agriculture across some of the world's leading research institutions, is developing global products for grasslands and livestock grazing in the 21st century.</p>
 <p>National Ecological Observatory Network</p> <p>5 datasets</p> <p>The U.S. National Science Foundation's National Ecological Observatory Network (NEON) is a continental-scale observation facility operated by federal and designed to collect long-term open access ecological data to better understand how U.S. ecosystems are changing. Its additional</p>	 <p>Planet</p> <p>3 datasets</p> <p>Planet provides daily satellite data that helps business, government, researchers, and journalists understand the physical world and take action. Through Planet's International Climate & Forests Initiative (ICFI), users can now access Planet's high resolution, analysis-ready mosaic of the world's tropics in order to help reduce and reverse the</p>	 <p>WeatherNext</p> <p>2 datasets</p> <p>WeatherNext is a growing suite of next-generation weather forecasting AI models developed by Google.</p>

258

Important notes

1. Variable's names reflecting the variable's purposes
2. Run the script regularly, use print() to check the results
3. Carefully read the documentation (functions, data, ...)
4. Online resources
5. Save often
6. Comment your code
7. LLMs can be useful but...

259

Thank you!

<https://www.eefabook.org/>

Online and free book
with tutorials

Planet



260