Introduction à Google Earth Engine

Cours pratiques divisés en études de cas

1

Introduction à Google Earth Engine

Cas 1 : Importer une ImageCollection, visualisation dans une région donnée

Cas 2 : Exporter un raster

Cas 3 : Exporter un tableau

Cas 4 : Créer une fonction, utiliser l'aggrégation spatiale et exporter un vecteur

Cas 5 : Indices de végétation et filtre simple

***Cas 6** : Algorithmes GEE et topographie*

Cas 7 : Images composites

Cas 8 : Détection des eaux de surface

Cas 9 : Classification

2

Utiliser Google Earth Engine Rappels

Lecture de la documentation

ee.Feature.buffer

Send feedback

Returns the input buffered by a given distance. If the distance is positive, the geometry is expanded, and if the distance is negative, the geometry is contracted.

Usage	Returns
`Feature.buffer(distance, maxError, proj)`	Feature

Argument	Type	Details
this: `feature`	Element	The feature the geometry of which is being buffered.
`distance`	Float	The distance of the buffering, which may be negative. If no projection is specified, the unit is meters. Otherwise the unit is in the coordinate system of the projection.
`maxError`	*ErrorMargin, default: null*	*The maximum amount of error tolerated when approximating the buffering circle and performing any necessary reprojection. If unspecified, defaults to 1% of the distance.*
`proj`	*Projection, default: null*	*If specified, the buffering will be performed in this projection and the distance will be interpreted as units of the coordinate system of this projection. Otherwise the distance is interpereted as meters and the buffering is performed in a spherical coordinate system.*

3

Utiliser Google Earth Engine Cas 1 – Importer et visualiser

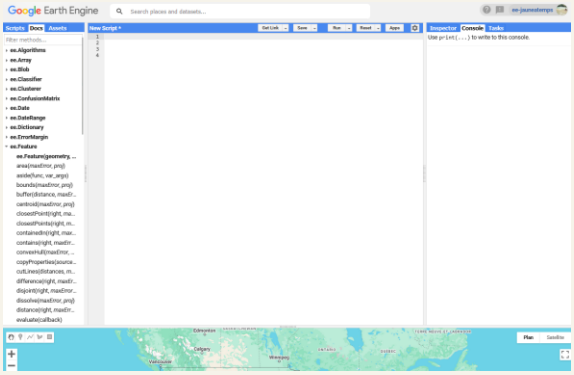
Importer une ImageCollection, visualisation dans une région donnée

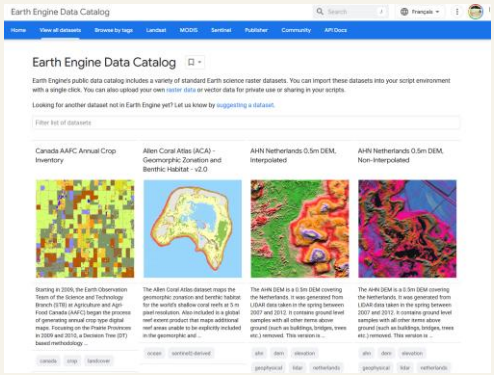
4

Utiliser Google Earth Engine Cas 1 – Importer et visualiser

Importer une ImageCollection, visualisation dans une région donnée

1. Ouvrir l'éditeur de code et le catalogue GEE



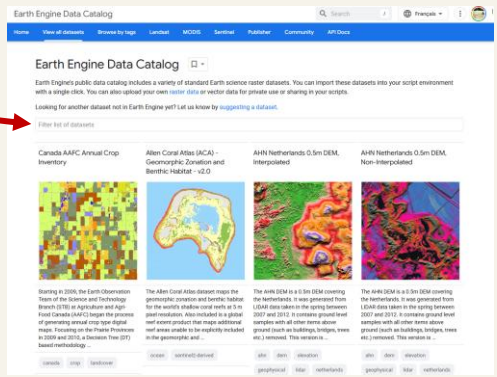


5

Utiliser Google Earth Engine Cas 1 – Importer et visualiser

Importer une ImageCollection, visualisation dans une région donnée

1. Ouvrir l'éditeur de code et le catalogue GEE
2. Chercher MOD11A2.061

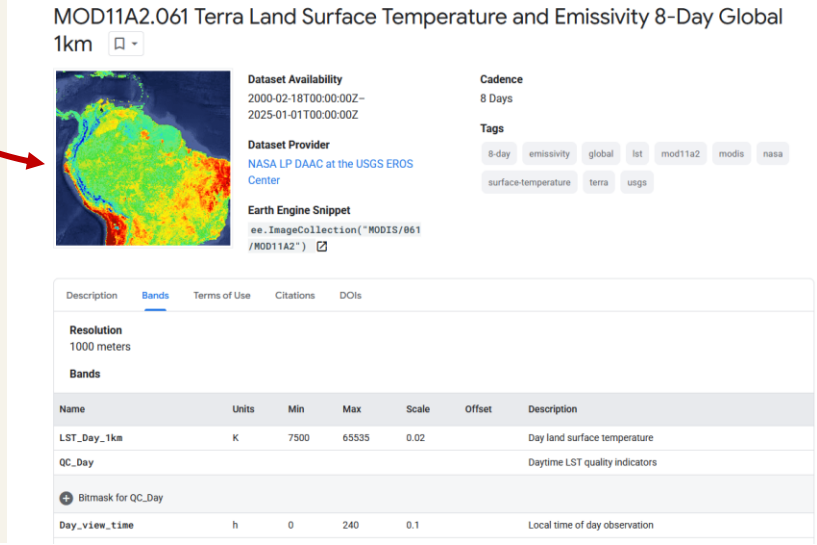


6

Utiliser Google Earth Engine Cas 1 – Importer et visualiser

Importer une ImageCollection, visualisation dans une région donnée

1. Ouvrir l'éditeur de code et le catalogue GEE
2. Chercher MOD11A2.061



7

Utiliser Google Earth Engine Cas 1 – Importer et visualiser

Importer une ImageCollection, visualisation dans une région donnée

1. Ouvrir l'éditeur de code et le catalogue GEE
2. Chercher MOD11A2.061
3. Importer l'ImageCollection dans l'éditeur de code

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
print(MOD11A2)
```

8

Utiliser Google Earth Engine Cas 1 – Importer et visualiser

Importer une ImageCollection, visualisation dans une région donnée

1. Ouvrir l'éditeur de code et le catalogue GEE
2. Chercher MOD11A2.061
3. Importer l'ImageCollection dans l'éditeur de code

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
```

4. Créer un polygone autour de Parakou comme vu hier, le renommer “site”

9

Utiliser Google Earth Engine Cas 1 – Importer et visualiser

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

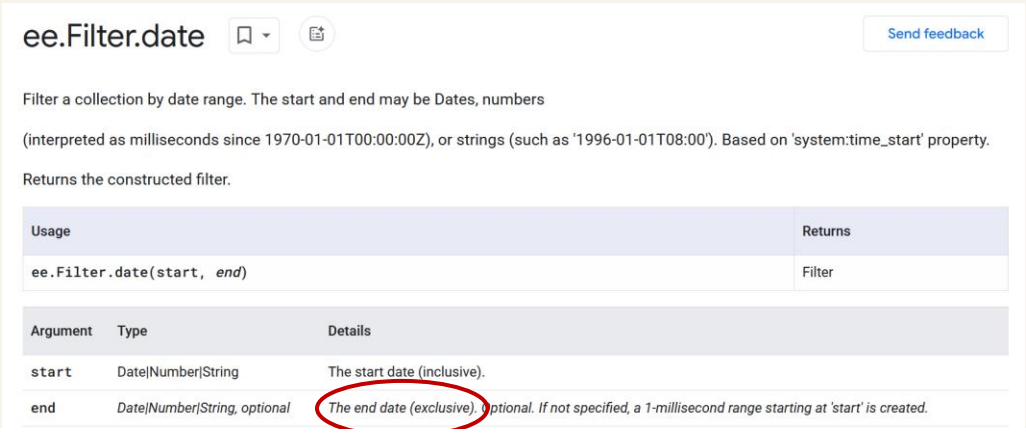
```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

10

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```



ee.Filter.date

Send feedback

Filter a collection by date range. The start and end may be Dates, numbers

(interpreted as milliseconds since 1970-01-01T00:00:00Z), or strings (such as '1996-01-01T08:00'). Based on 'system:time_start' property.

Returns the constructed filter.

Usage	Returns
`ee.Filter.date(start, end)`	Filter

Argument	Type	Details
`start`	Date|Number|String	The start date (inclusive).
`end`	*Date|Number|String, optional*	*The end date (exclusive). Optional. If not specified, a 1-millisecond range starting at 'start' is created.*

11

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

Les fonctions sont appliquées : Objet.fonction1().fonction2().fonction3();

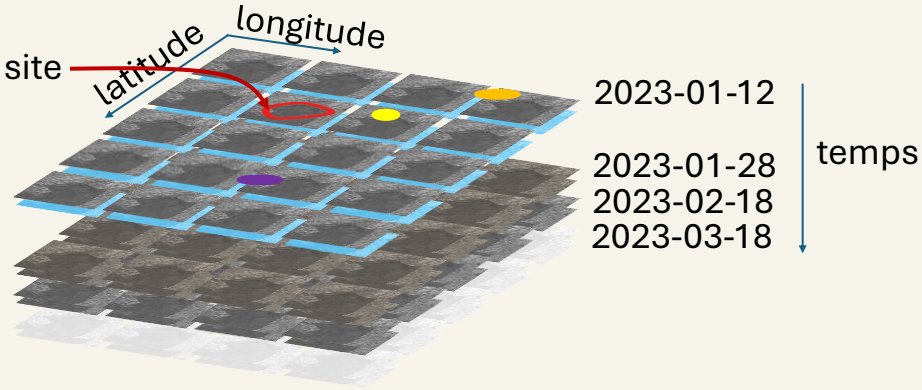
Et non pas fonction3(fonction2(fonction1(Objet)))

12

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

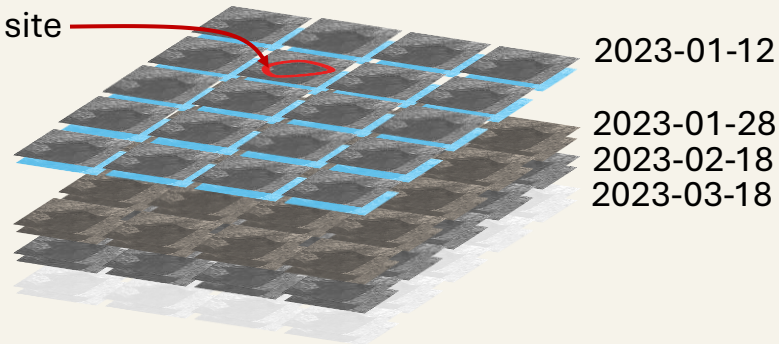


13

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

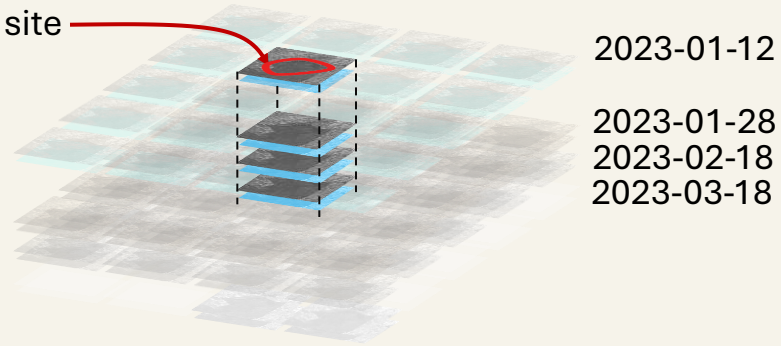


14

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

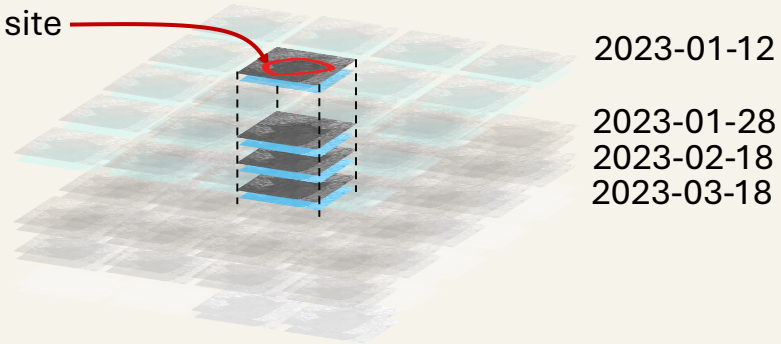
```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```



Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

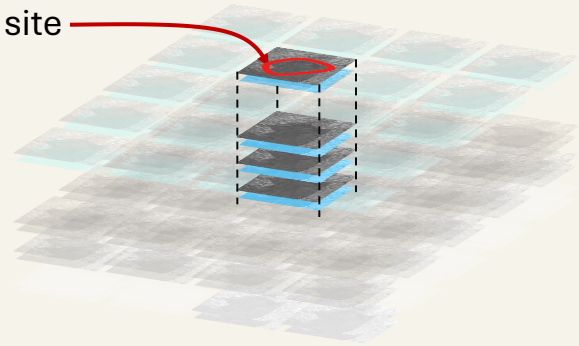


Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

début fin



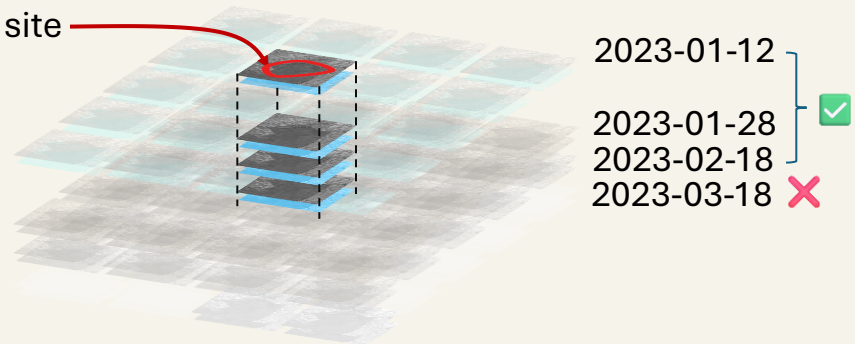
17

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

début fin



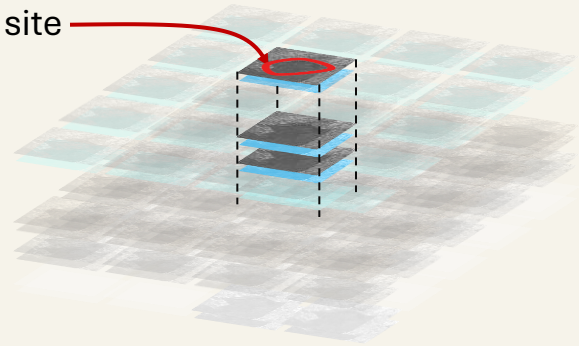
18

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

début fin



2023-01-12 ✅
2023-01-28 ✅
2023-02-18 ✅
2023-03-18 ❌

ImageCollection
de trois images

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

6. Regarder la taille de l'ImageCollection filtrée

Importer une ImageCollection, visualisation dans

5. Filtrer MOD11A2 : par date et pour le site d'int

```
var MOD11A2 = ee.ImageCollection
.filterBounds(site)
.filter(ee.Filter.date('2023-01-
```

6. Regarder la taille de l'ImageCollection filtrée

```
print(MOD11A2);
```

Bandes : valeurs des pixels

21

Importer une ImageCollection, visualisation dans

5. Filtrer MOD11A2 : par date et pour le site d'int

```
var MOD11A2 = ee.ImageCollection
.filterBounds(site)
.filter(ee.Filter.date('2023-01-
```

6. Regarder la taille de l'ImageCollection filtrée

```
print(MOD11A2);
```

Type de données :
int = nombre entier (8/16/32/64 bits, signed ou unsigned)

float = decimal

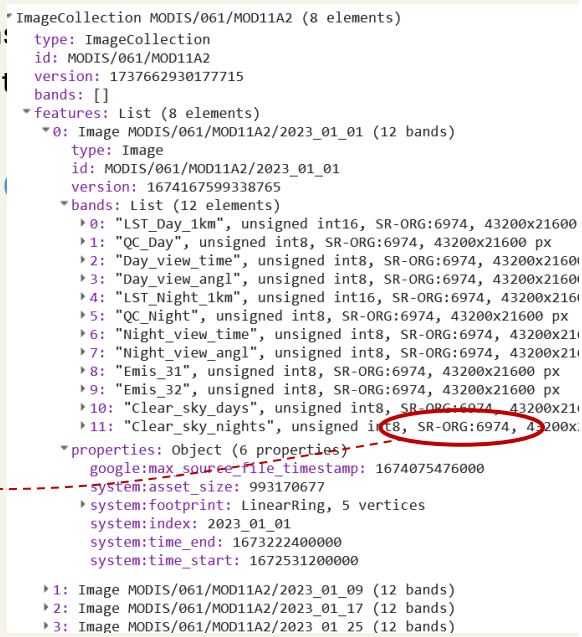
possible de convertir

22

Importer une ImageCollection, visualisation dans

5. Filtrer MOD11A2 : par date et pour le site d'int
```
var MOD11A2 = ee.ImageCollection
.filterBounds(site)
.filter(ee.Filter.date('2023-01-
```
6. Regarder la taille de l'ImageCollection filtrée
```
print(MOD11A2);
```

Projection



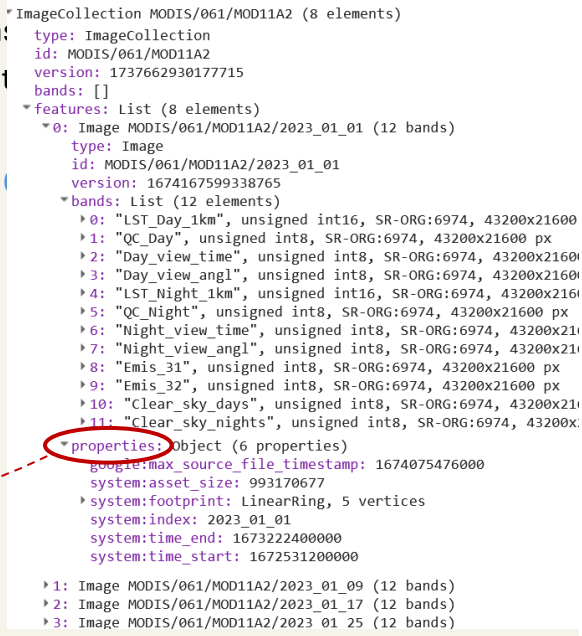
23

Importer une ImageCollection, visualisation dans

5. Filtrer MOD11A2 : par date et pour le site d'int
```
var MOD11A2 = ee.ImageCollection
.filterBounds(site)
.filter(ee.Filter.date('2023-01-
```
6. Regarder la taille de l'ImageCollection filtrée
```
print(MOD11A2);
```

Propriétés de l'image : date, …

Permet de filtrer une ImageCollection



24

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

6. Regarder la taille de l'ImageCollection filtrée

```
print(MOD11A2);
print(MOD11A2.size()); // la taille de l'ImageCollection
```

25

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

6. Regarder la taille de l'ImageCollection filtrée

```
print(MOD11A2);
print(MOD11A2.size()); // la taille de l'ImageCollection
```

7. Nous nous intéressons à la température uniquement : fonction pour sélectionner une bande

26

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

6. Regarder la taille de l'ImageCollection filtrée

```
print(MOD11A2);
print(MOD11A2.size()); // la taille de l'ImageCollection
```

7. Nous nous intéressons à la température uniquement : fonction pour sélectionner une bande

Name	Units	Min	Max	Scale	Offset	Description
LST_Day_1km	K	7500	65535	0.02		Day land surface temperature
QC_Day						Daytime LST quality indicators

27

Importer une ImageCollection, visualisation dans une région donnée

5. Filtrer MOD11A2 : par date et pour le site d'intérêt

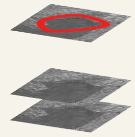
```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
.filterBounds(site)
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

6. Regarder la taille de l'ImageCollection filtrée

```
print(MOD11A2);
print(MOD11A2.size()); // la taille de l'ImageCollection
```

7. Nous nous intéressons à la température uniquement

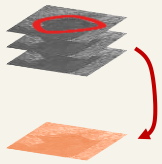
```
var MOD11A2 = MOD11A2.select('LST_Day_1km');
print(MOD11A2);
```



28

Importer une ImageCollection, visualisation dans une région donnée

8. Calcul de la température moyenne par pixel entre Janvier et Février



Se base sur l'application de la fonction **Reducer** : c'est l'application d'une statistique sur chaque pixel à travers une ImageCollection.

Il existe plusieurs types de Reducer : moyenne, maximum, minimum...
`imagecollection.reduce(ee.Reducer.mean())` ou bien `imagecollection.mean()`

ImageCollection à 3 Images

4	5
6	10

3	5
10	0

2	5
15	15

reducer mean (moyenne)

Image

3	5
10,3	8,3

29

Ut

Home > Products > Google Earth Engine > Reference

Was this helpful?

ee.ImageCollection.reduce

Send feedback

Applies a reducer across all of the images in a collection.

If the reducer has a single input, it will be applied separately to each band of the collection; otherwise it must have the same number of inputs as the collection has bands.

The reducer output names determine the names of the output bands: reducers with multiple inputs will use the output names directly, while reducers with a single input will prefix the output name with the input band name (e.g., '10_mean', '20_mean').

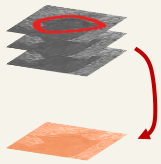
Usage	Returns
`ImageCollection.reduce(reducer, parallelScale)`	Image

Argument	Type	Details
this: `collection`	ImageCollection	The image collection to reduce.
`reducer`	Reducer	The reducer to apply to the given collection.
`parallelScale`	*Float, default: 1*	*A scaling factor used to limit memory use; using a larger parallelScale (e.g., 2 or 4) may enable computations that run out of memory with the default.*

30

Importer une ImageCollection, visualisation dans une région donnée

8. Calcul de la température moyenne par pixel entre Janvier et Février



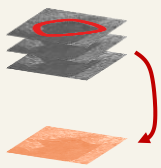
Se base sur l'application de la fonction **Reducer** : c'est l'application d'une statistique sur chaque pixel à travers une ImageCollection.

Il existe plusieurs types de Reducer : moyenne, maximum, minimum…
`imagecollection.reduce(ee.Reducer.mean())` ou bien `imagecollection.mean()`
Un reducer produit une image et non pas une ImageCollection

31

Importer une ImageCollection, visualisation dans une région donnée

8. Calcul de la température moyenne par pixel entre Janvier et Février



Se base sur l'application de la fonction **Reducer** : c'est l'application d'une statistique sur chaque pixel à travers une ImageCollection.

Il existe plusieurs types de Reducer : moyenne, maximum, minimum…
`imagecollection.reduce(ee.Reducer.mean())` ou bien `imagecollection.mean()`
Un reducer produit une image et non pas une ImageCollection

```
var MOD11A2moyenne = MOD11A2.reduce(ee.Reducer.mean());
print(MOD11A2moyenne);
```

32

Importer une ImageCollection, visualisation dans une région donnée

9. Changement d'unités et facteurs de conversion
   - Application du taux de conversion (scale)
   - Conversion K vers °C

Name	Units	Min	Max	Scale	Offset	Description
LST_Day_1km	K	7500	65535	0.02		Day land surface temperature
QC_Day						Daytime LST quality indicators

33

Importer une ImageCollection, visualisation dans une région donnée

9. Changement d'unités et facteurs de conversion
   - Application du taux de conversion (scale)
   - Conversion K vers °C

```
var MOD11A2moyenne_C = MOD11A2moyenne
.multiply(0.02).subtract(273.15)
```

Name	Units	Min	Max	Scale	Offset	Description
LST_Day_1km	K	7500	65535	0.02		Day land surface temperature
QC_Day						Daytime LST quality indicators

34

Importer une ImageCollection, visualisation dans une région donnée

10. Il ne nous reste qu'à afficher l'image produite

```
var couleur = {
  min: 25, max: 35,
  palette: ['blue', 'yellow', 'red']
};
```

GEE accepte les couleurs principales en “mots” ou alors toute la palette de couleurs en héxadécimal

Création d'un gradient de couleur allant du bleu au rouge en passant par le jaune

Toutes les valeurs < 25 seront de la même couleur (bleu ici),

> 35 seront rouges

35

Importer une ImageCollection, visualisation dans une région donnée

10. Il ne nous reste qu'à afficher l'image produite

```
var couleur = {
  min: 25, max: 35,
  palette: ['blue', 'yellow', 'red']
};

Map.addLayer(MOD11A2moyenne_C, couleur, 'T_moyenne_JanFev');
```

36

Importer une ImageCollection, visualisation dans une région donnée

10. Il ne nous reste qu'à afficher l'image produite

```
var couleur = {
   min: 25, max: 35,
   palette: ['blue', 'yellow', 'red']
};

Map.addLayer(MOD11A2moyenne_C, couleur, 'T_moyenne_JanFev');
```

=

```
Map.addLayer(MOD11A2moyenne_C, {min: 25, max: 35, palette: ['blue',
'yellow', 'red']}, 'T_moyenne_JanFev2');

Map.addLayer(MOD11A2moyenne_C, {}, 'T_moyenne_JanFev3');
```

Pas d'instructions = GEE utilisera une palette noir-blanc

37


```
var MOD11A2 = ee.ImageCollection('MODIS/061/MOD11A2')
     .filterBounds(site)
     .filter(ee.Filter.date('2023-01-01', '2023-03-01'))
     .select('LST_Day_1km');
var MOD11A2moyenne = MOD11A2.reduce(ee.Reducer.mean());
var MOD11A2moyenne_C = MOD11A2moyenne
     .multiply(0.02).subtract(273.15)
var couleur = {
     min: 25, max: 35,
     palette: ['blue', 'yellow', 'red']
   };
Map.addLayer(MOD11A2moyenne_C, couleur, 'T_moyenne_JanFev');
```

Exercice :
- chercher la valeur maximale
- de Décembre 2021 à Avril 2022 (compris)

38

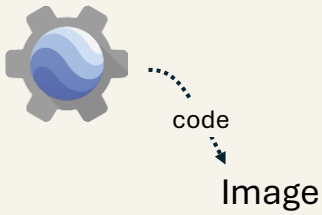
Exporter des données avec GEE

- Rasters
- Figures
- Assets GEE
- Tableaux
- Vidéos

39

Exporter un raster

Nécessairement une Image



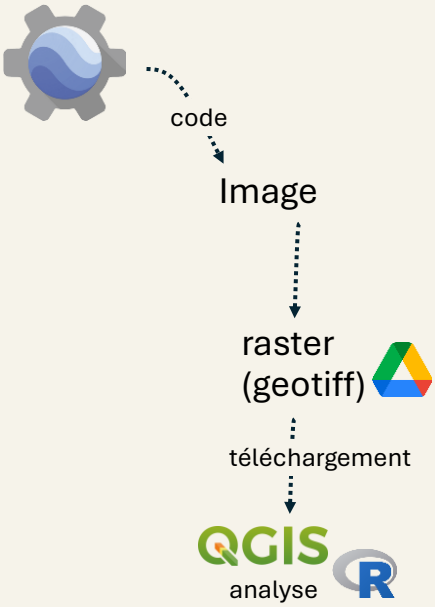
- Geotiff sur Google Drive
- Comme donnée sur Earth Engine (Asset)
- vers Cloud Storage

40

Exporter un raster

Nécessairement une image

- Geotiff sur Google Drive
- Comme donnée sur Earth Engine (Asset)
- vers Cloud Storage

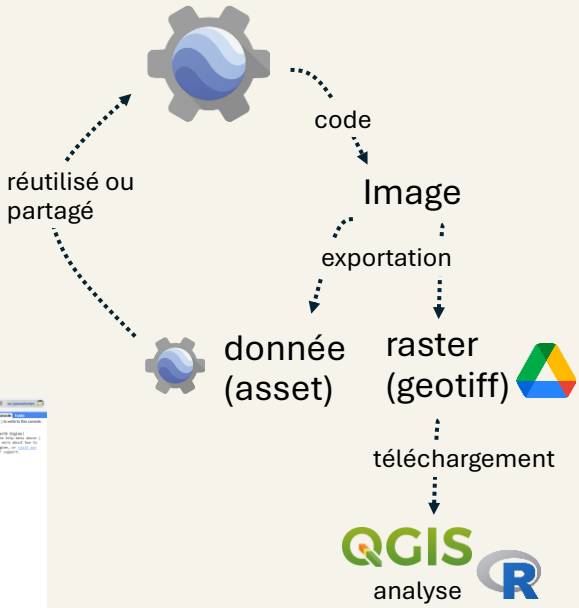


41

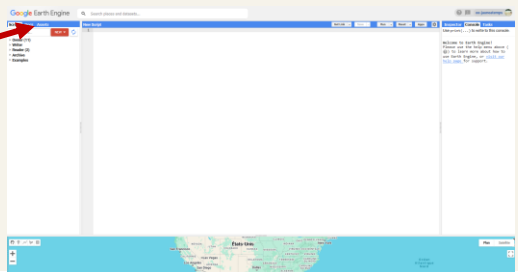
Exporter un raster

Nécessairement une image

- Geotiff sur Google Drive
- Comme donnée sur Earth Engine (Asset)
- vers Cloud Storage

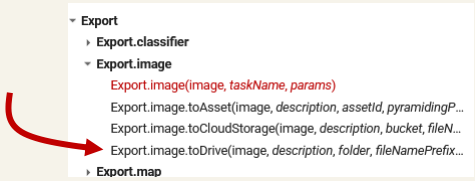


Les assets



42

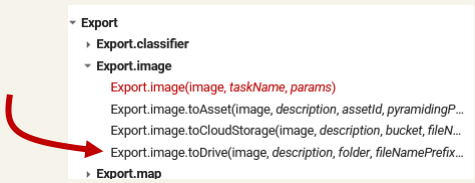
Geotiff sur Google Drive




```
[le reste du code précédent]
var MOD11A2moyenne_C = MOD11A2moyenne
     .multiply(0.02).subtract(273.15);
```

43

Geotiff sur Google Drive




```
Export.image.toDrive({
  image: MOD11A2moyenne_C,
  description: 'temp_moyenne_site_1',
  folder: 'cours_GEE_UP',
  region: site,
  scale: 500,
  crs: 'EPSG:4326',
  fileFormat: 'GeoTIFF',
});
```

44

Utiliser Google Earth Engine Cas 2 – Exporter un raster

Geotiff sur Google Drive

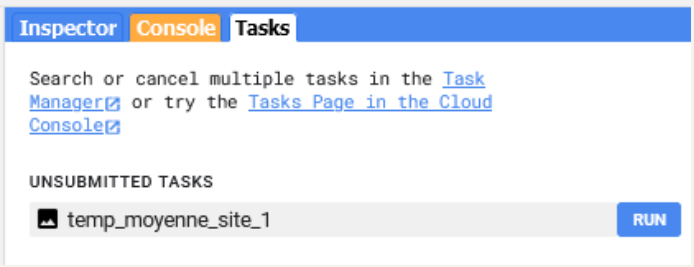
```
type: Image
id: LANDSAT/LC08/C02/T1_L2/LC08_192054_20140211
version: 1629939624650843
bands: List (6 elements)
  0: "blue", unsigned int16, EPSG:32631, 7591x7761 px
  1: "green", unsigned int16, EPSG:32631, 7591x7761 px
  2: "red", unsigned int16, EPSG:32631, 7591x7761 px
  3: "nir", unsigned int16, EPSG:32631, 7591x7761 px
  4: "swir1", unsigned int16, EPSG:32631, 7591x7761 px
  5: "swir2", unsigned int16, EPSG:32631, 7591x7761 px
```

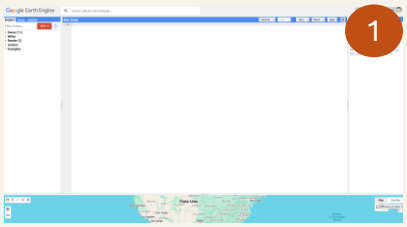
Les bandes doivent être du même pixelType (Int32, etc).

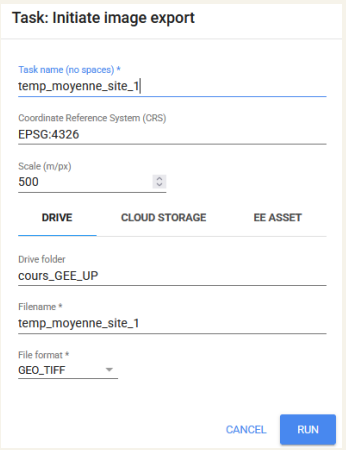
45

Utiliser Google Earth Engine Cas 2 – Exporter un raster

Geotiff sur Google Drive

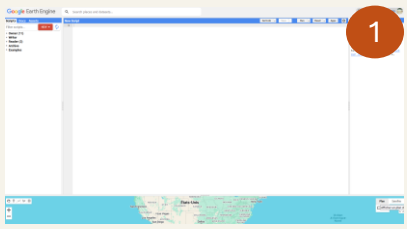


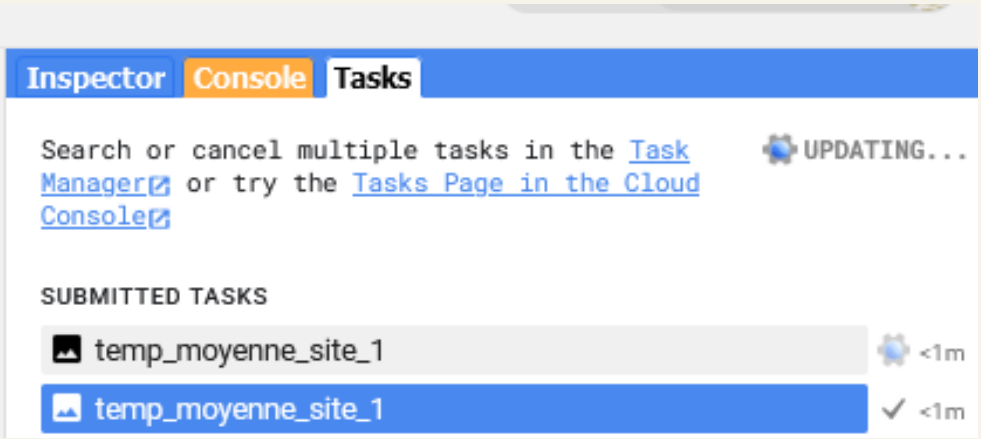




46

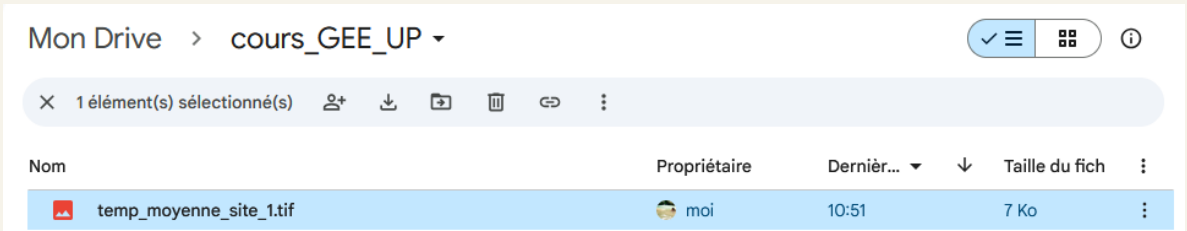
Geotiff sur Google Drive





47

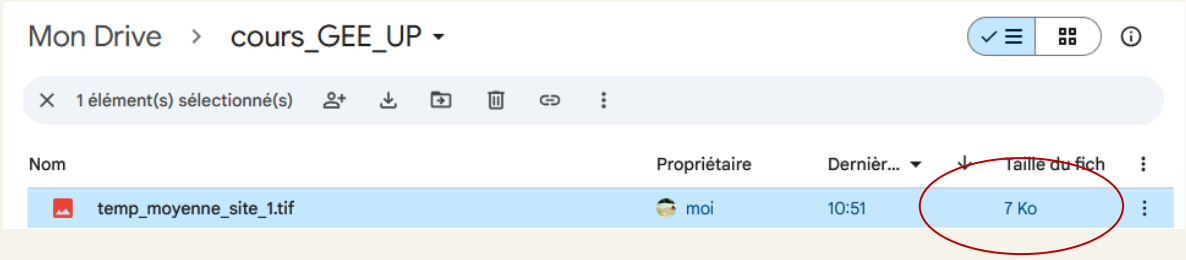
Geotiff sur Google Drive



48

Utiliser Google Earth Engine Cas 2 – Exporter un raster

Geotiff sur Google Drive



Changer le type de donnée avant l'extraction :

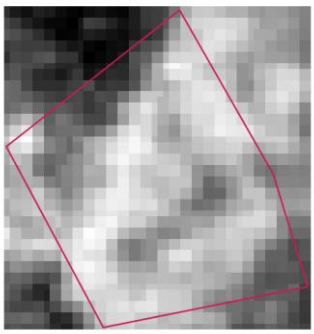
```
MOD11A2moyenne_C.float()
MOD11A2moyenne_C.double()
```

49

Utiliser Google Earth Engine Cas 2 – Exporter un raster

Geotiff sur Google Drive

Fonction image.**clip**(vecteur);



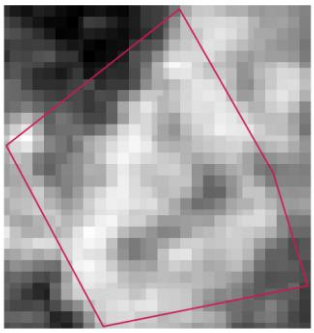
Usage	Returns
`Image.clip(geometry)`	Image

Argument	Type	Details
this: `image`	Image	The Image instance.
`geometry`	Feature|Geometry|Object	The Geometry or Feature to clip to.

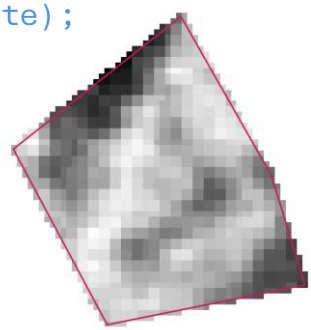
50

Geotiff sur Google Drive

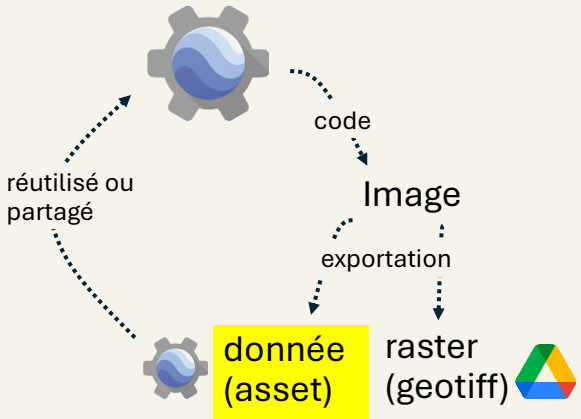
Fonction image.**clip**(vecteur);



```
var MOD11A2moyenne_C = MOD11A2moyenne
     .multiply(0.02).subtract(273.15);
var MOD11A2moyenne_C = MOD11A2moyenne_C.clip(site);
[export]
```



Comme donnée sur Earth Engine (Asset)

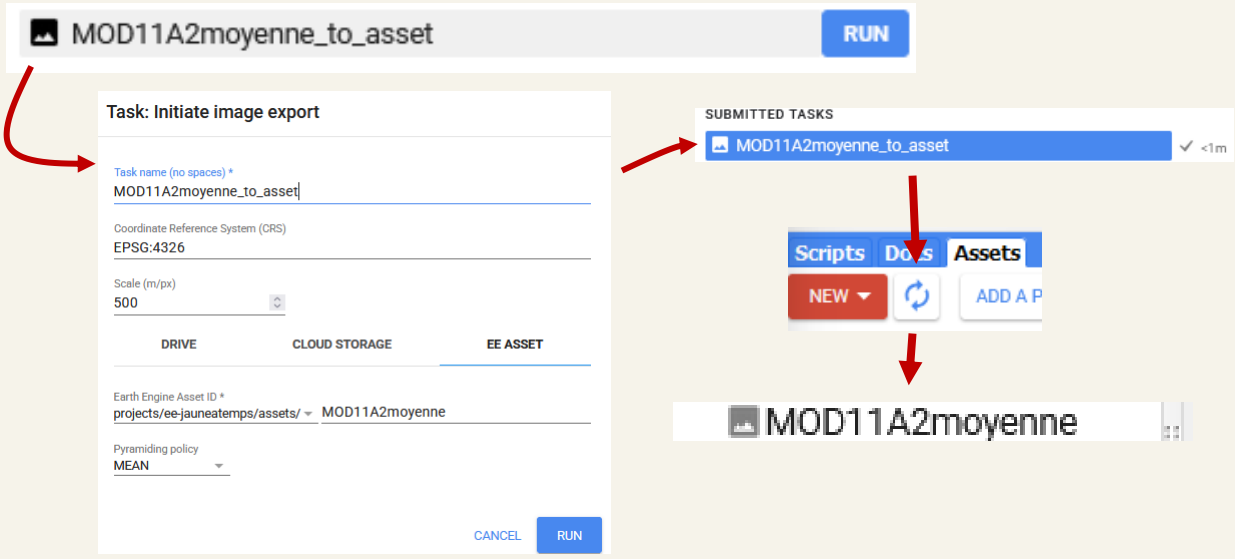


Comme donnée sur Earth Engine (Asset)

```
var MOD11A2moyenne_C = MOD11A2moyenne_C.clip(site);

Export.image.toAsset({
  image: MOD11A2moyenne,
  description: 'MOD11A2moyenne_to_asset',
  assetId: 'projects/NOMDUPROJET/assets/NOMDELASSET',
  region: site,
  scale: 500,
  crs: 'EPSG:4326'
});
```

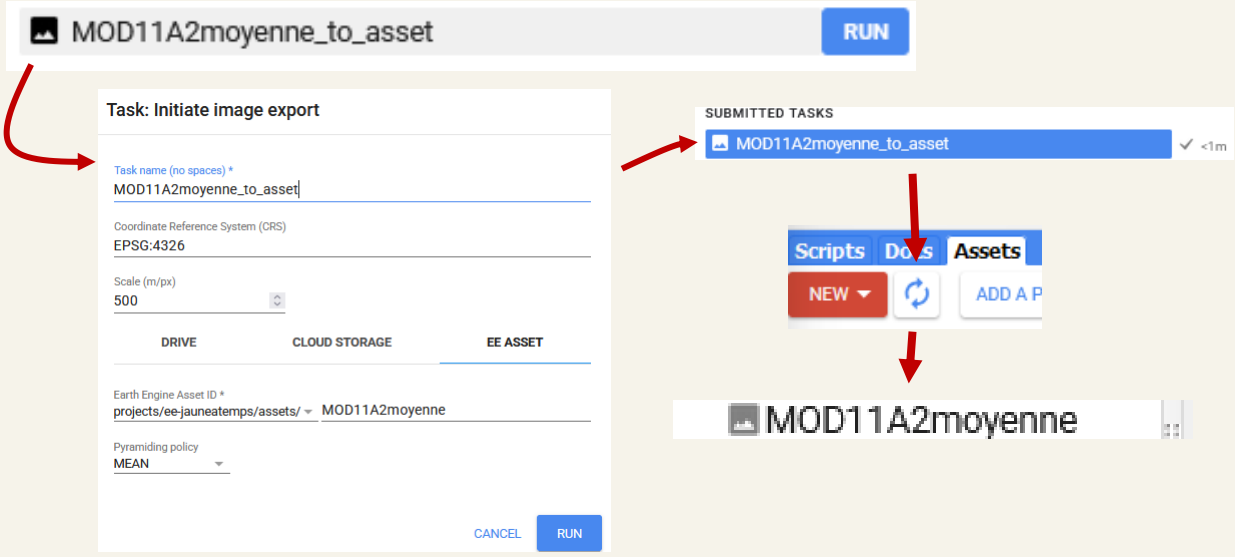
53

Comme donnée sur Earth Engine (Asset)



54

Comme donnée sur Earth Engine (Asset)



55

Comme donnée sur Earth Engine (Asset)

Pour ajouter cet asset à votre editeur de code

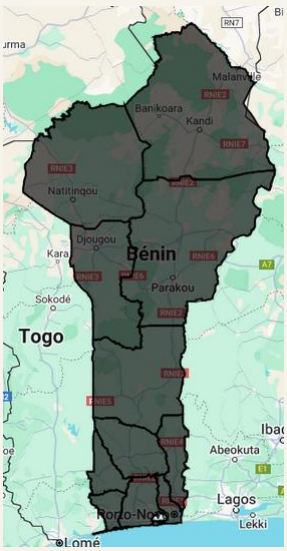
56

Conclusions

image.**clip**(vecteur)
Export.image.toDrive()
Export.image.toAsset()

57

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

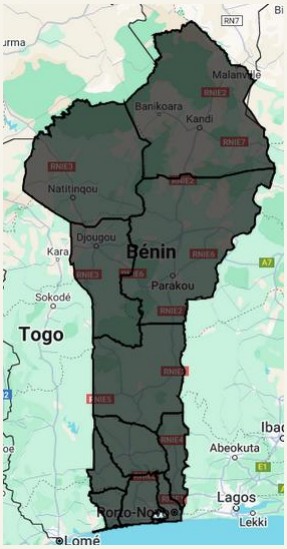


	Avril-Mai (1990-2020)	
Région	**Température moyenne (°C)**	**Précipitation moyenne (mm)**
Borgou	*x*	*x*
Atakora	*x*	*x*
Zou	*x*	*x*
…	…	…

58

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)



	Avril-Mai (1990-2020)	
Région	**Température moyenne (°C)**	**Précipitation moyenne (mm)**
Borgou	*x*	*x*
Atakora	*x*	*x*
Zou	*x*	*x*
…	…	…

Données :
limites administratives, → FeatureCollection
temperature, précipitation → Image ou ImageCollection

59

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

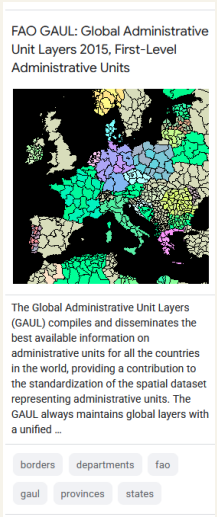
60

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

nouveau code: **“Cas_3”**

```
var BeninLevel1 =
ee.FeatureCollection('FAO/GAUL/2015/level1')
.filter(ee.Filter.eq('ADM0_NAME', 'Benin'));
```



61

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

nouveau code: **“Cas_3”**

```
var BeninLevel1 =
ee.FeatureCollection('FAO/GAUL/2015/level1')
.filter(ee.Filter.eq('ADM0_NAME', 'Benin'));
```

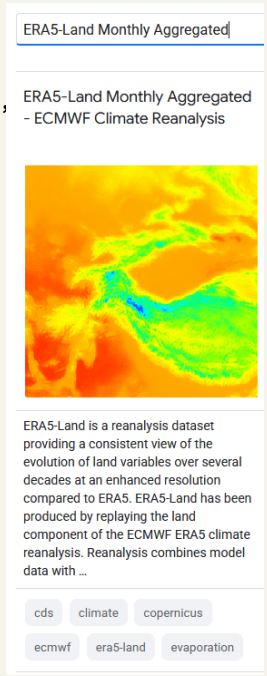
filtre — type de filtre : “eq” pour “equal” — propriété — valeur

62

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai,

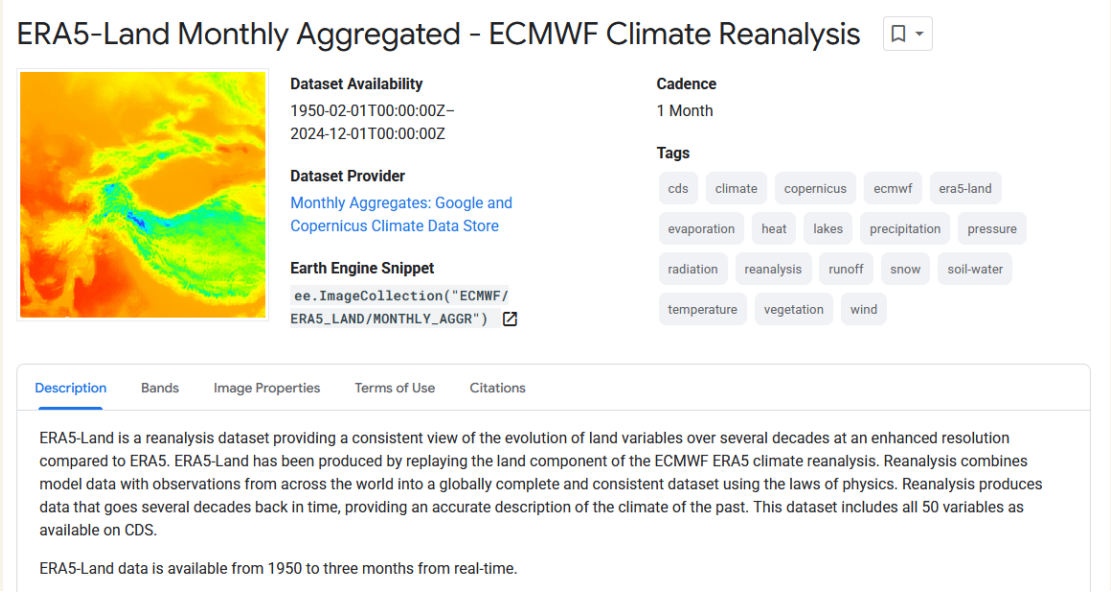
Données :
limites administratives,
temperature,
précipitation



63

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)



64

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Description Bands Image Properties Terms of Use Citations

Resolution
11132 meters

Bands

Name	Units	Description
dewpoint_temperature_2m	K	Temperature to which the air, at 2 meters above the surface of the Earth, would have to be cooled for saturation to occur. It is a measure of the humidity of the air. Combined with temperature and pressure, it can be used to calculate the relative humidity. 2m dew point temperature is calculated by interpolating between the lowest model level and the Earth's surface, taking account of the atmospheric conditions.
temperature_2m	K	Temperature of air at 2m above the surface of land, sea or in-land waters. 2m temperature is calculated by interpolating between the lowest model level and the Earth's surface, taking account of the atmospheric conditions.
skin_temperature	K	Temperature of the surface of the Earth. The skin temperature is the theoretical temperature that is required to satisfy the surface energy balance. It represents the temperature of the uppermost surface layer, which has no heat capacity and so can respond instantaneously to changes in surface fluxes. Skin temperature is calculated differently over land and sea

65

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

temperature_2m	K	Temperature of air at 2m above the surface of land, sea or in-land waters. 2m temperature is calculated by interpolating between the lowest model level and the Earth's surface, taking account of the atmospheric conditions.
total_precipitation_sum	m	Accumulated liquid and frozen water, including rain and snow, that falls to the Earth's surface. It is the sum of large-scale precipitation (that precipitation which is generated by large-scale weather patterns, such as troughs and cold fronts) and convective precipitation (generated by convection which occurs when air at lower levels in the atmosphere is warmer and less dense than the air above, so it rises). Precipitation variables do not include fog, dew or the precipitation that evaporates in the atmosphere before it lands at the surface of the Earth. This variable is

66

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

```
var climat =
ee.ImageCollection('ECMWF/ERA5_LAND/MONTHLY_AGGR')
.select('temperature_2m', 'total_precipitation_sum')
```

67

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

```
var climat =
ee.ImageCollection('ECMWF/ERA5_LAND/MONTHLY_AGGR')
.select('temperature_2m', 'total_precipitation_sum')
```

sélectionne ces deux bandes dans chaque Image

68

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

peu pratique dans ce cas

```
.filter(ee.Filter.date('2023-01-01', '2023-03-01'));
```

```
var climat =
ee.ImageCollection('ECMWF/ERA5_LAND/MONTHLY_AGGR')
.select('temperature_2m', 'total_precipitation_sum')
.filter(ee.Filter.calendarRange(1990, 2020, 'year'))
.filter(ee.Filter.calendarRange(4, 5, 'month'));
```

69

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Données :
limites administratives,
temperature,
précipitation

```
var climat =
ee.ImageCollection('ECMWF/ERA5_LAND/MONTHLY_AGGR')
.select('temperature_2m', 'total_precipitation_sum')
.filter(ee.Filter.calendarRange(1990, 2020, 'year'))
.filter(ee.Filter.calendarRange(4, 5, 'month')));
print(climat) // 31 ans x 2 mois = 62 images
```

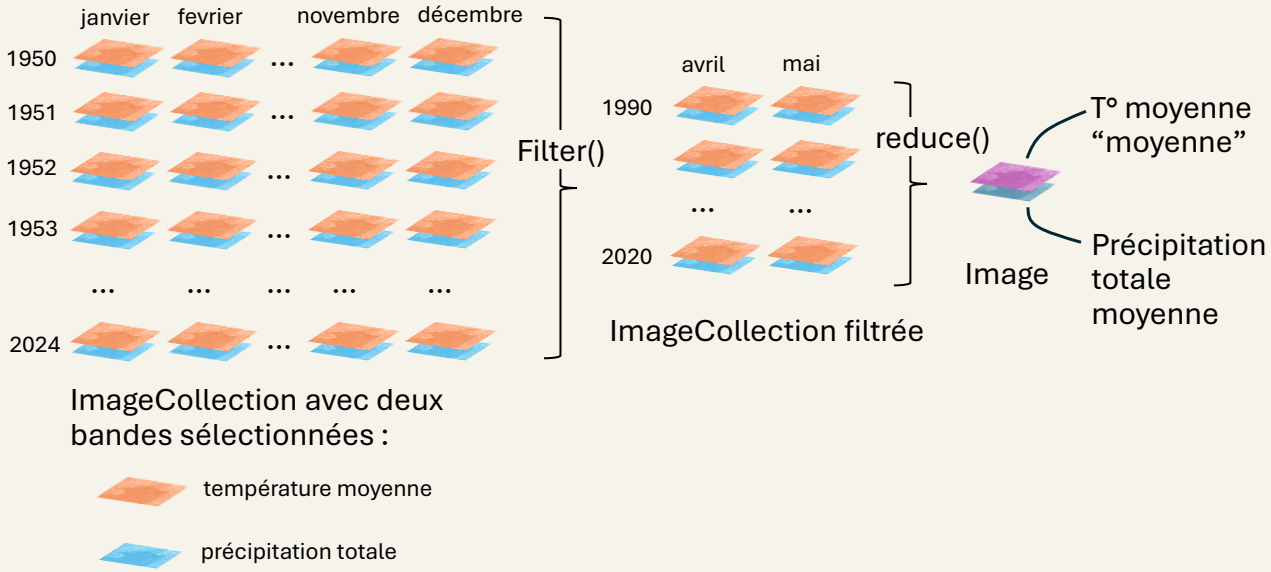
70

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var climat =
ee.ImageCollection('ECMWF/ERA5_LAND/MONTHLY_AGGR')
.select('temperature_2m', 'total_precipitation_sum')
.filter(ee.Filter.calendarRange(1990, 2020, 'year'))
.filter(ee.Filter.calendarRange(4, 5, 'month'));
print(climat) // 31 ans x 2 mois = 62 images

var climat_moyenne = climat.mean(); // produit image
print(climat_moyenne) // image à deux bandes
```

71

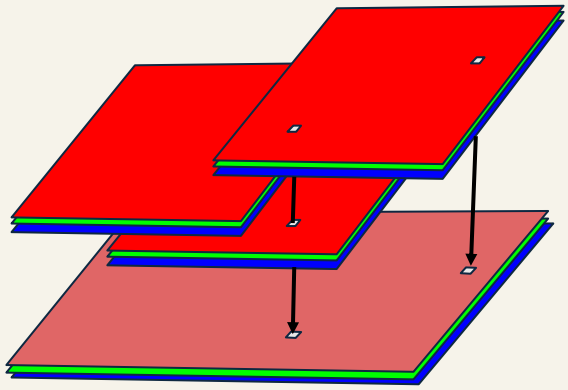
Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)



72

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

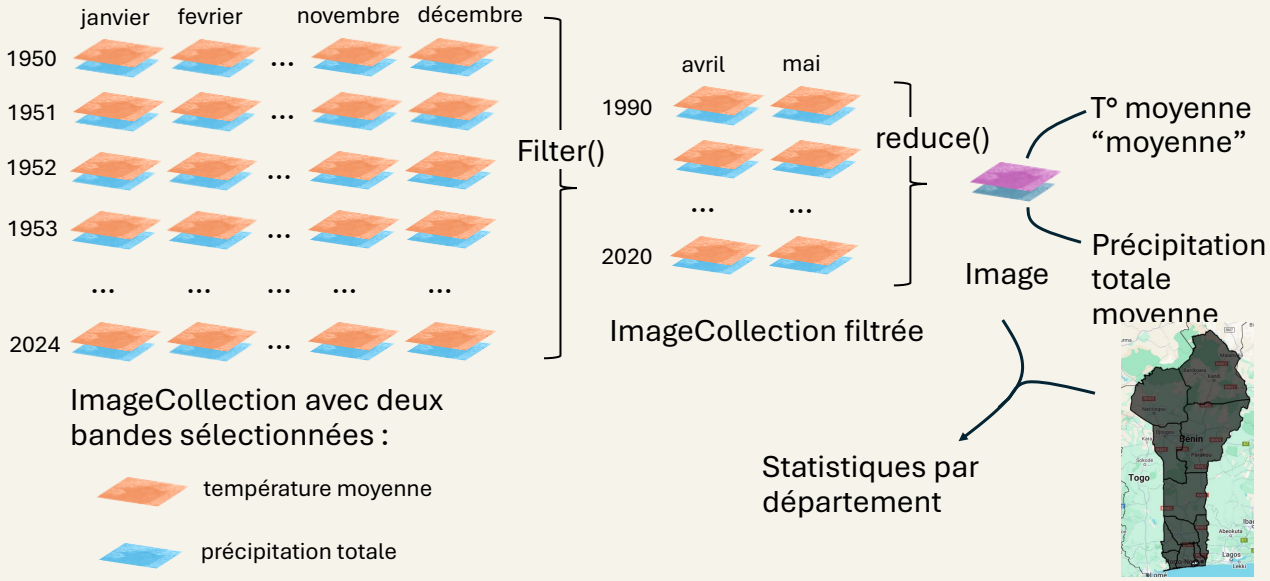


Noel Gorelick

73

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)



74

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var climat_moyenne = climat.mean(); // produit image
print(climat_moyenne) // image à deux bandes

print(BeninLevel1);
```

75

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var climat_moyenne = climat.mean(); // produit image
print(climat_moyenne) // image à deux bandes

print(BeninLevel1);

print(BeninLevel1.aggregate_array("ADM1_NAME"));
```

les fonctions aggregate permettent de travailler sur les propriétés des ImageCollection et des FeatureCollection

76

Temp s des unites administratives (Avril-Mai, 1990-2020)

```
var  imat.mean(); // produit image
prir / image à deux bandes

prir

prir gate_array("ADM1_NAME"));
```

ee.FeatureCollection
aggregate_array
aggregate_count
aggregate_count_distinct
aggregate_first
aggregate_histogram
aggregate_max
aggregate_mean
aggregate_min
aggregate_product
aggregate_sample_sd
aggregate_sample_var
aggregate_stats
aggregate_sum
aggregate_total_sd
aggregate_total_var

les fonctions aggregate permettent de travailler sur les propriétés des ImageCollection et des FeatureCollection

77

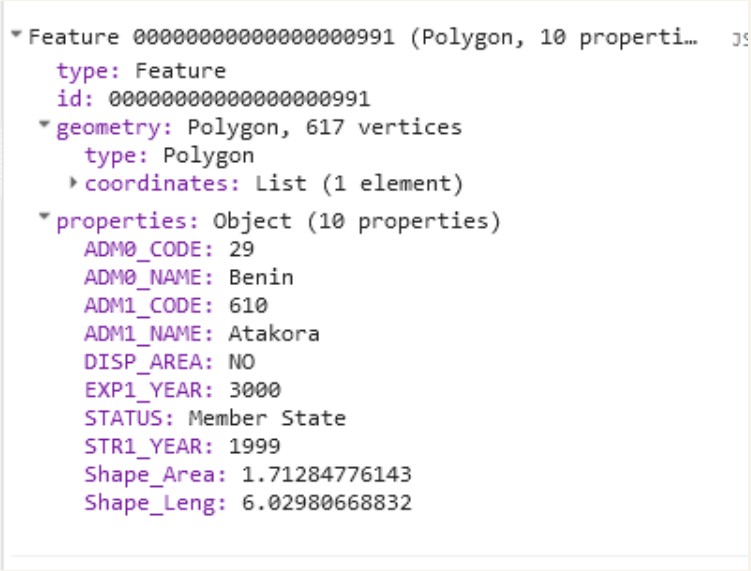
Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
print(BeninLevel1);
print(BeninLevel1.aggregate_array("ADM1_NAME"));

var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'));
print(Atakora); // FeatureCollection avec un seul Feature
print(Atakora.first()); // Feature
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();
```

78

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
print(BeninLevel1);
print(BeninLevel1.aggrega

var Atakora = BeninLevel
.filter(ee.Filter.eq('ADM
print(Atakora); // Featu
print(Atakora.first());
var Atakora = BeninLevel
.filter(ee.Filter.eq('ADM
.first();
```



79

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats = climat_moyenne.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: Atakora,
  scale: 11000,
  crs: 'EPSG:4326'
});
```

80

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

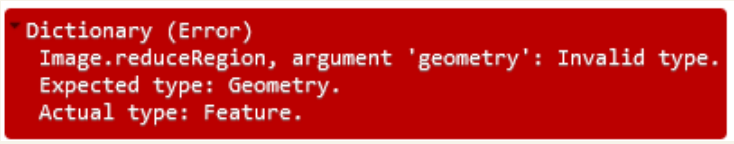
```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats = climat_moyenne.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: Atakora,
  scale: 11000,
  crs: 'EPSG:4326'
});
```

reduceRegion ne s'intéresse qu'à une seule region (vecteur)

81

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats = climat_moyenne.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: Atakora,
  scale: 11000,
  crs: 'EPSG:4326'
});
```



82

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats = climat_moyenne.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: Atakora.geometry(), ✅
  scale: 11000,
  crs: 'EPSG:4326'
});
```

83

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats = climat_moyenne.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: Atakora.geometry(),
  scale: 11000,
  crs: 'EPSG:4326'
});
```

```
▾ Object (2 properties)
    temperature_2m: 303.18130162681416
    total_precipitation_sum: 0.0693768725399788
```

84

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', '
.first();

var stats = climat_moyenne.reduceRe
  reducer: ee.Reducer.mean(),
  geometry: Atakora.geometry(),
  scale: 11000,
  crs: 'EPSG:4326'
});
```

ee.Reducer.max(*numInputs*)
ee.Reducer.mean()
ee.Reducer.median(*maxBuckets, min..*
ee.Reducer.min(*numInputs*)
ee.Reducer.minMax()
ee.Reducer.mode(*maxBuckets, minB..*
ee.Reducer.or()
ee.Reducer.pearsonsCorrelation()
ee.Reducer.percentile(percentiles, o...

```
Object (4 properties)
  temperature_2m_max: 305.69443665234417
  temperature_2m_min: 300.505260777548
  total_precipitation_sum_max: 0.10498726971850521
  total_precipitation_sum_min: 0.03645013468082153
```

Beaucoup de possibilités

85

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Que signifient ici les maximum et minimum ?

ee.Reducer.max(*numInputs*)
ee.Reducer.mean()
ee.Reducer.median(*maxBuckets, min..*
ee.Reducer.min(*numInputs*)
ee.Reducer.minMax()
ee.Reducer.mode(*maxBuckets, minB..*
ee.Reducer.or()
ee.Reducer.pearsonsCorrelation()
ee.Reducer.percentile(percentiles, o...

Object (4 properties)
temperature_2m_max: 305.69443665234417
temperature_2m_min: 300.505260777548
total_precipitation_sum_max: 0.10498726971850521
total_precipitation_sum_min: 0.03645013468082153

86

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats = climat_moyenne.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: Atakora.geometry(),
  scale: 11000,
  crs: 'EPSG:4326'
});
```

87

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats_benin = climat_moyenne.reduceRegions({
  reducer: ee.Reducer.mean(),
  collection: BeninLevel1,
  scale: 11000,
  crs: 'EPSG:4326'
});
```

reduceRegions s'intéresse à plusieurs régions (vecteurs, soit une FeatureCollection)

88

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

```
var Atakora = BeninLevel1
.filter(ee.Filter.eq('ADM1_NAME', 'Atakora'))
.first();

var stats_benin = climat_moyenne.reduceRegions({
  reducer: ee.Reducer.mean(),
  collection: BeninLevel1,
  scale: 11000,
  crs: 'EPSG:4326'
});
print(stats_benin); // FeatureCollection
```

reduceRegions s'intéresse à plusieurs régions (vecteurs, soit une FeatureCollection)

89

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Extraire un tableau

```
Export.table.toDrive({
  collection: stats_benin,
  description: 'temperatureprecipitation19902020Benin',
  folder: 'cours_GEE_UP',
  fileFormat: 'CSV',
  selectors: ['ADM1_NAME', 'temperature_2m',
'total_precipitation_sum']
});
```

90

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Extraire un tableau

```
Export.table.toDrive({
  collection: stats_benin,
  description: 'temperatureprecipitation19902020Benin',
  folder: 'cours_GEE_UP',
  fileFormat: 'CSV',
  selectors: ['ADM1_NAME', 'temperature_2m',
'total_precipitation_sum']
});
```

91

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Extraire un tableau

```
Export.table.toDrive({
  collection: stats_benin,
  description: 'temperatureprecipitation19902020Benin',
  folder: 'cours_GEE_UP',
  fileFormat: 'CSV',
  selectors: ['ADM1_NAME', 'temperature_2m',
'total_precipitation_sum']
});
```

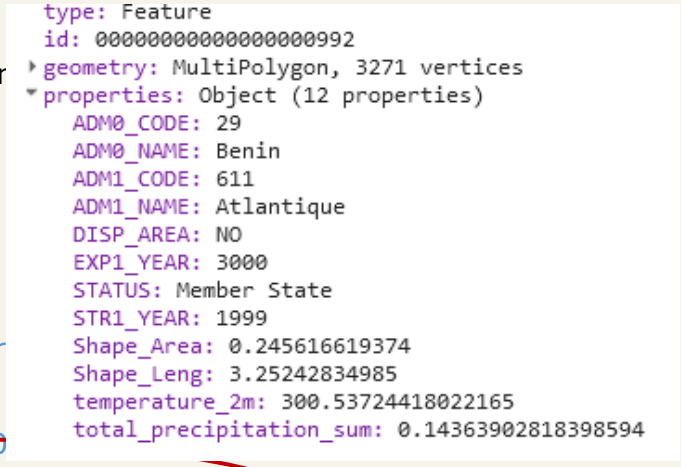
92

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des ur

Extraire un tableau

```
Export.table.toDrive({
  collection: stats_benin,
  description: 'temperaturepr
  folder: 'cours_GEE_UP',
  fileFormat: 'CSV',//'GeoJSO
  selectors: ['ADM1_NAME', 'temperature_2m',
'total_precipitation_sum']
});
```



93

Utiliser Google Earth Engine Cas 3 – Exporter un tableau

Température et précipitation dans des unites administratives (Avril-Mai, 1990-2020)

Extraire un tableau

```
Export.table.toDrive({
  collection: stats_benin,
  description: 'temperaturepre
  folder: 'cours_GEE_UP',
  fileFormat: 'CSV',//'GeoJSO
  selectors: ['ADM1_NAME', 'te
'total_precipitation_sum']
});
```

	A	B	C
1	ADM1_NAME	temperature_2m	total_precipitation_s
2	Alibori	305.2598568	0.04272398691
3	Atakora	303.1813016	0.06937687254
4	Atlantique	300.5372442	0.1436390282
5	Borgou	302.0028001	0.08980707526
6	Collines	301.0477647	0.1211123042
7	Couffo	300.5086218	0.1287314729
8	Donga	300.9605953	0.1240269296
9	Littoral	300.8260128	0.1325597818
10	Mono	300.6685771	0.1482974
11	Oueme	300.5565595	0.1491322794
12	Plateau	300.602534	0.1274833416
13	Zou	300.86598	0.1195695778

94

Production d'une fonction pour retirer des décimales

Les fonctions dans GEE
- Celles déjà existantes
- Nouvelles ajoutées dans le code

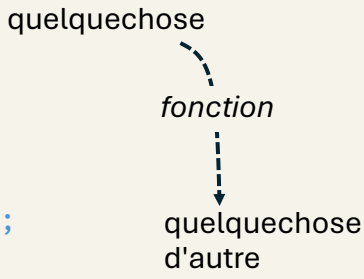
95

Production d'une fonction pour retirer des décimales

Les fonctions dans GEE
- Celles déjà existantes
- Nouvelles ajoutées dans le code

Une fonction se créée comme :

```
var mafonction = function(parametre){
     var nouvellevaleur = parametre.add(15);
     return nouvellevaleur
};
print(mafonction(ee.Number(14)));
```

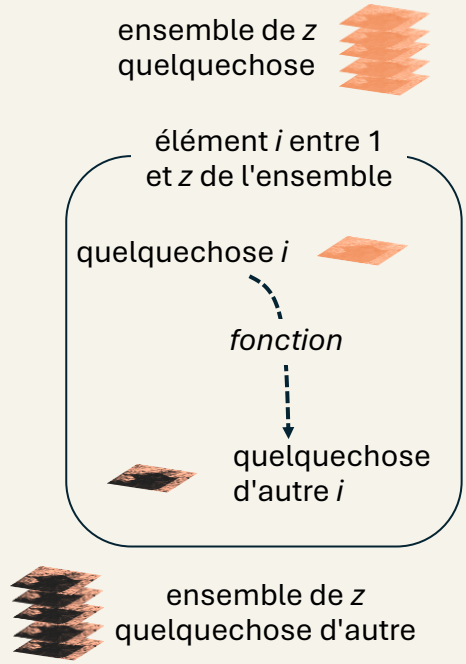


96

Production d'une fonction pour retirer des décimales

Les fonctions dans GEE
- Celles déjà existantes
- Nouvelles ajoutées dans le code

Pour appliquer une fonction à plusieurs éléments d'un groupe (par exemple chaque feature d'une FeatureCollection) il faut l'appeler dans **une boucle**.



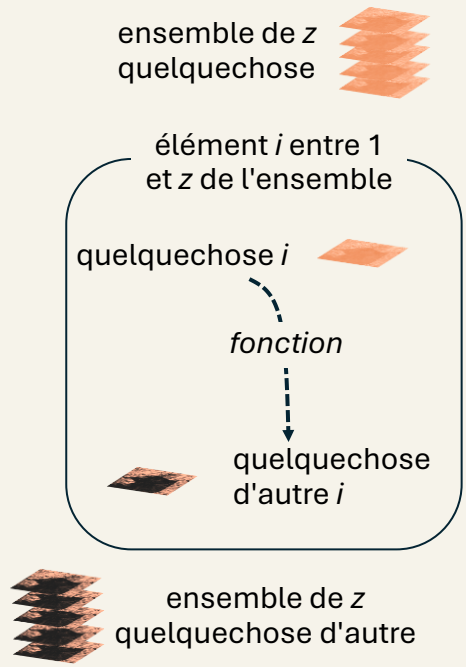
97

Production d'une fonction pour retirer des décimales

Les fonctions dans GEE
- Celles déjà existantes
- Nouvelles ajoutées dans le code

Pour appliquer une fonction à plusieurs éléments d'un groupe (par exemple chaque feature d'une FeatureCollection) il faut l'appeler dans **une boucle**.

Exemples :
- *Cliper* chaque image ou ajouter une bande à chaque image d'une ImageCollection,
- Convertir des valeurs dans une FeatureCollection



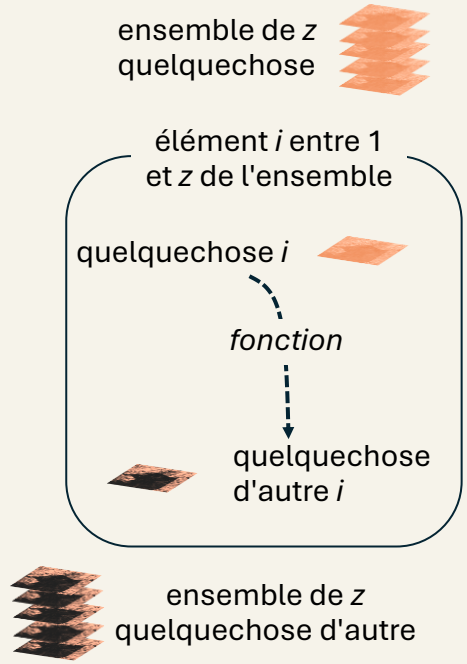
98

Production d'une fonction pour retirer des décimales

Les fonctions dans GEE
- Celles déjà existantes
- Nouvelles ajoutées dans le code

Dans GEE, il est possible d'utiliser la méthode *for* mais ce n'est pas conseillé.

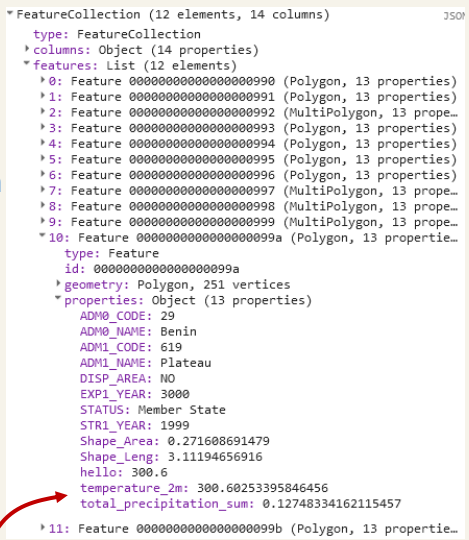
Dans GEE, les boucles se font avec .**map***(fonction)*



99

Production d'une fonction pour retirer des décimales

```
print(stats_benin); // FeatureCollection
```



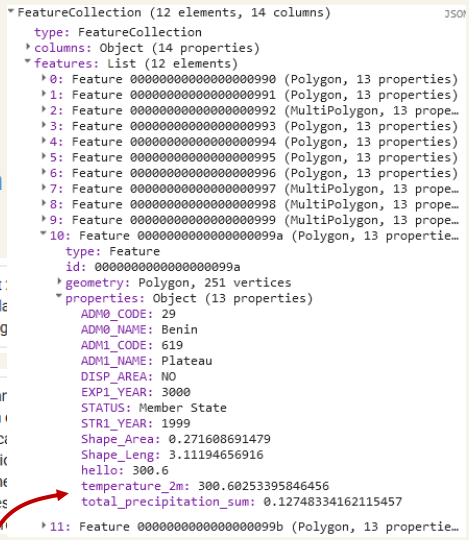
but de la boucle : retirer des décimales, convertir en °C ou mm

100

Production d'une fonction pour retirer des décimales

```
print(stats_benin); // FeatureCollection
```

temperature_2m	K	Temperature of air at 1 temperature is calcula Earth's surface, taking
total_precipitation_sum	m	Accumulated liquid ar surface. It is the sum (generated by large-sca convective precipitatio levels in the atmosphe Precipitation variables the atmosphere befo

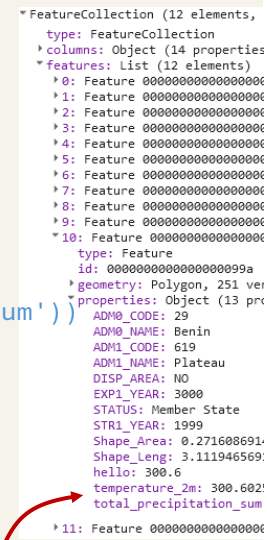


but de la boucle : retirer des décimales, convertir en °C ou mm

101

Production d'une fonction pour retirer des décimales

```
print(stats_benin); // FeatureCollection
var funcor = function(feature){
      var temp = ee.Number(feature.get('temperature_2m'))
      var prec = ee.Number(feature.get('total_precipitation_sum'))

};
```

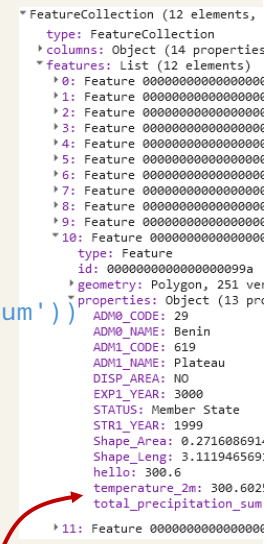


but de la boucle : retirer des décimales, convertir en °C ou mm

102

Production d'une fonction pour retirer des décimales

```
print(stats_benin); // FeatureCollection
var funcor = function(feature){
      var temp = ee.Number(feature.get('temperature_2m'))
      var prec = ee.Number(feature.get('total_precipitation_sum'))
      var tempcor =
temp.subtract(273.15).multiply(100).round().divide(100)
      var preccor =
prec.multiply(1000).multiply(100).round().divide(100)
};
```



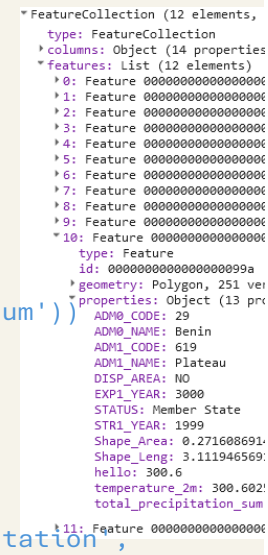
but de la boucle : retirer des décimales, convertir en °C ou mm

103

Utiliser Google Earth Engine Cas 4 – Fonctions

Production d'une fonction pour retirer des décimales

```
print(stats_benin); // FeatureCollection
var funcor = function(feature){
      var temp = ee.Number(feature.get('temperature_2m'))
      var prec = ee.Number(feature.get('total_precipitation_sum'))
      var tempcor =
temp.subtract(273.15).multiply(100).round().divide(100)
      var preccor =
prec.multiply(1000).multiply(100).round().divide(100)

      return feature.set('temperature', tempcor).set('precipitation',
preccor)
};
```



104

Production d'une fonction pour retirer des décimales

```
print(stats_benin); // FeatureCollection
var funcor = function(feature){
      var temp = ee.Number(feature.get('temperature_2m'))
      var prec = ee.Number(feature.get('total_precipitation_sum'))
      var tempcor =
temp.subtract(273.15).multiply(100).round().divide(100)
      var preccor =
prec.multiply(1000).multiply(100).round().divide(100)

      return feature.set('temperature', tempcor).set('precipitation',
preccor)
};
var stats_benin_corr = stats_benin.map(funcor)
```

105

Production d'une fonction pour retirer des déc

```
print(stats_benin_corr);
```

106

Production d'une fonction pour retirer des décimales

Exercice :
Copiez et modifiez le code pour extraire un csv (tableau) de la **temperature et précipitation** moyennes des mois de **Juillet et aout** entre **1980 et 1995** pour le **Togo**

107

Production d'une fonction pour retirer des décimales

Exercice :
Copiez et modifiez le code pour extraire un csv (tableau) de la **temperature et précipitation** moyennes des mois de **Juillet et aout** entre **1980 et 1995** pour le **Togo**

Résultat :

	A	B	C
1	ADM1_NAME	temperature	precipitation
2	Centrale	23.4	294
3	Kara	24.17	253.54
4	Maritime	24.37	140.9
5	Plateaux	23.57	243.7
6	Savanes	25.42	197.09

108

Cas 3 et 4 : conclusions

FeatureCollection.**filter**(**ee.Filter.eq**())
ee.Filter.calendarRange()
ImageCollection ou FeatureCollection.**first**()
ImageCollection ou FeatureCollection.**aggregate_**()
image.**reductionRegion**()
image.**reductionRegions**()
Export.table.toDrive()
function(quelquechose){}
.**map**(function)
.round()
feature.**get**('nomdelavariable')
feature.**set**('nom', valeur)

109

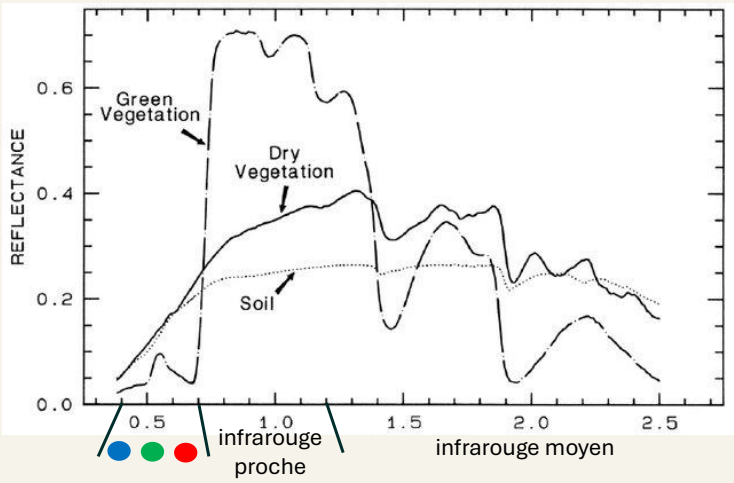
Calcul d'un indice de vegetation sur des zones cultivées

Rappel indices de végétation

110

Utiliser Google Earth Engine Cas 5 – Indices

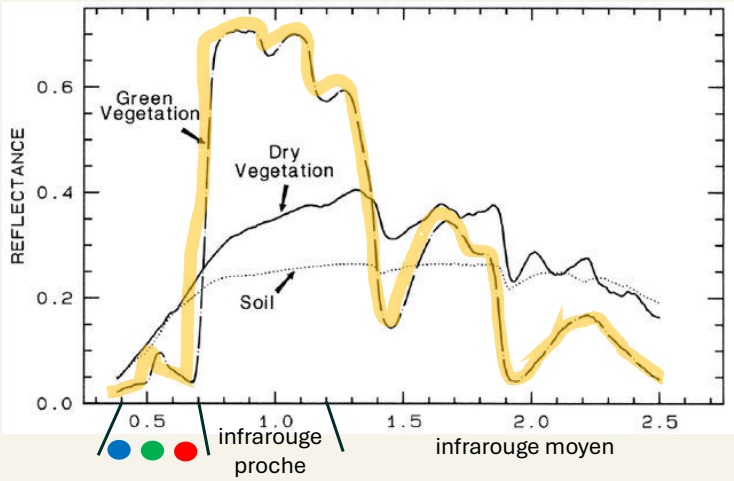
Rappel



111

Utiliser Google Earth Engine Cas 5 – Indices

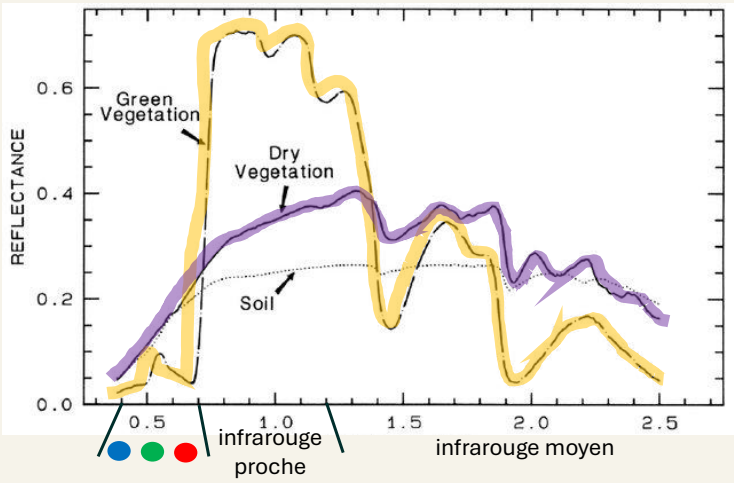
Rappel



Végétation saine:
Rouge = 0,05, PIR = 0,65

112

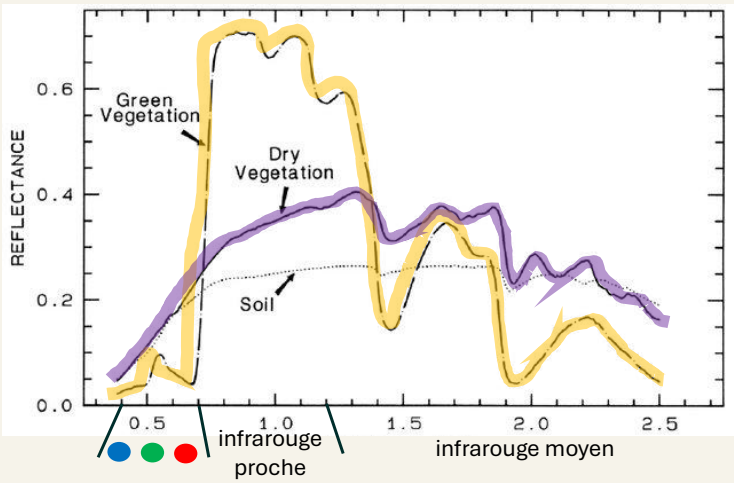
Rappel



Végétation saine:
Rouge = 0,05, PIR = 0,65

Végétation sèche:
Rouge = 0,2, PIR = 0,35

Rappel



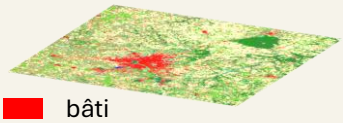
Végétation saine:
Rouge = 0,05, PIR = 0,65
→ NDVI = 0,85

Végétation sèche:
Rouge = 0,2, PIR = 0,35
→ NDVI = 0,27

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

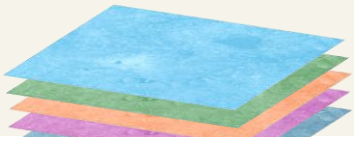
Calcul d'un indice de vegetation sur des zones cultivées

occupation du sol



classes
- bâti
- forêt
- eau
- culture
- ...

données multispectrales

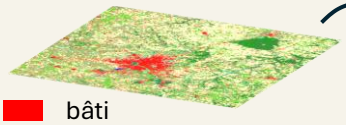


- bleu
- vert
- rouge
- infrarouge proche...

115

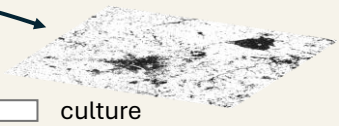
Calcul d'un indice de vegetation sur des zones cultivées

occupation du sol



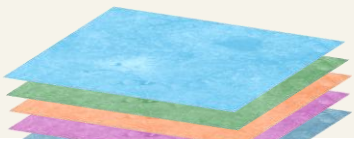
classes
- bâti
- forêt
- eau
- culture
- ...

masque des cultures



classes
- culture
- autre

données multispectrales

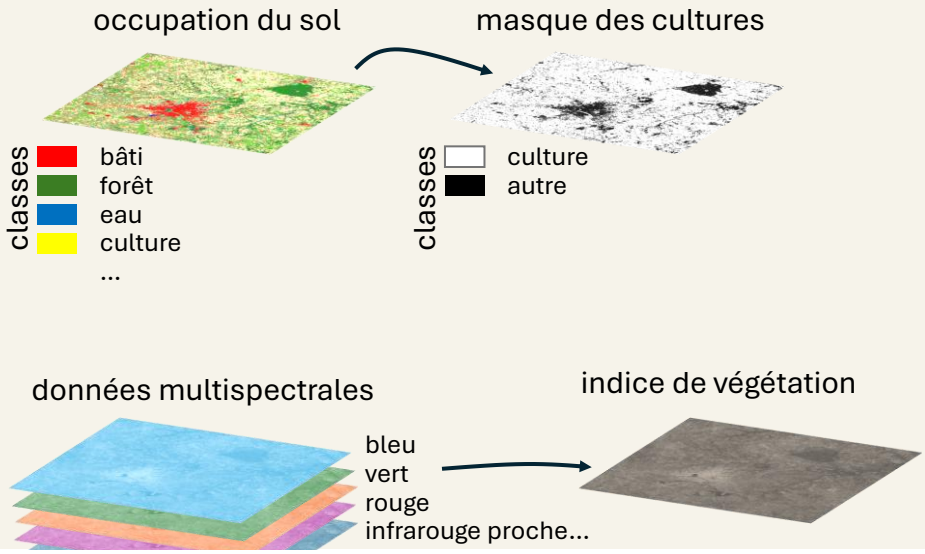


- bleu
- vert
- rouge
- infrarouge proche...

116

Utiliser Google Earth Engine Cas 5 – Indices

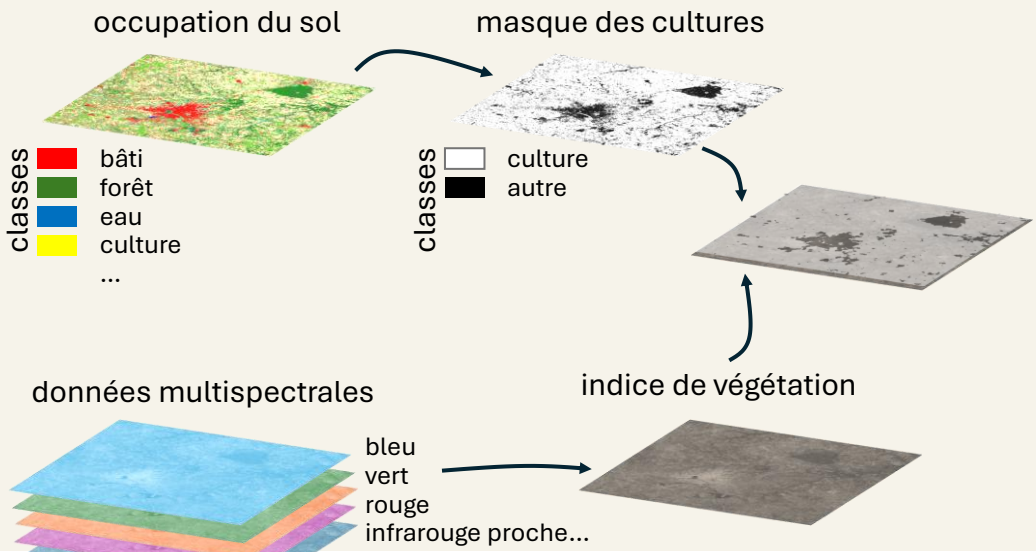
Calcul d'un indice de vegetation sur des zones cultivées



117

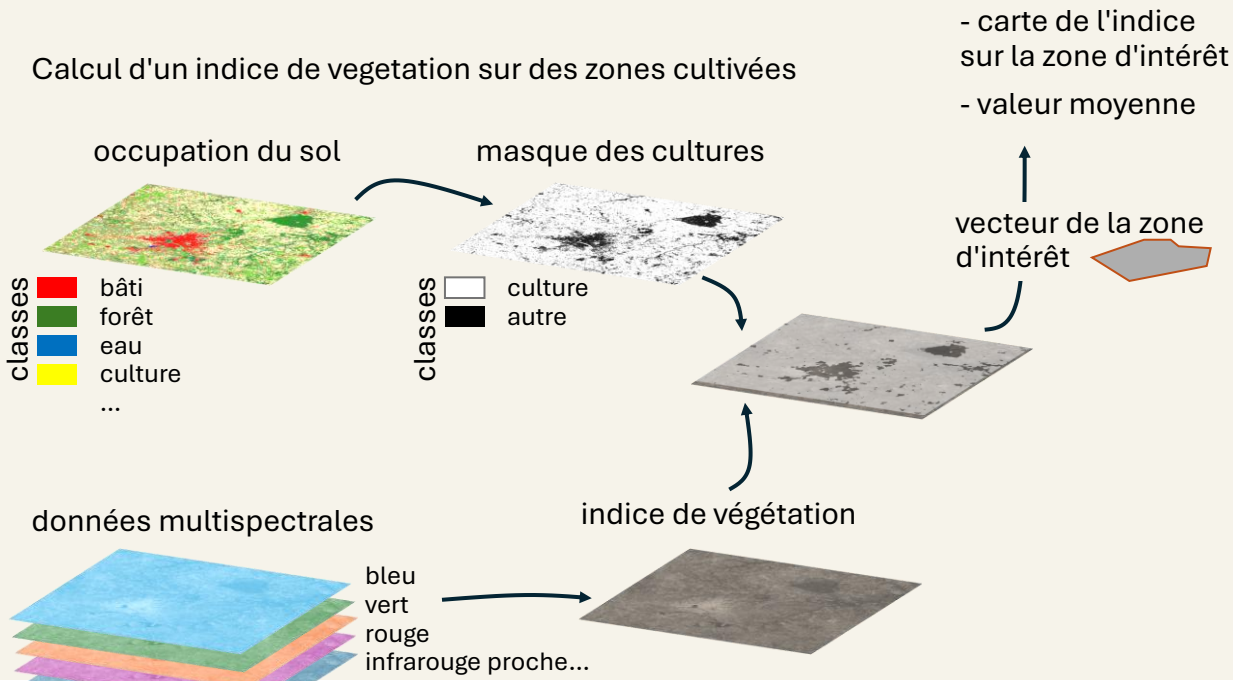
Utiliser Google Earth Engine Cas 5 – Indices

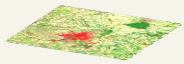
Calcul d'un indice de vegetation sur des zones cultivées



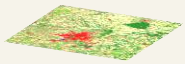
118

Calcul d'un indice de vegetation sur des zones cultivées



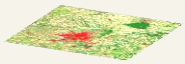
Importation de la carte d'occupation du sol 

```
var carteOS = ee.Image('
projects/ee-
jauneatemps/assets/pastoralism/OBSYDYA_OS_2023_L1_Awanla')
Map.centerObject(carteOS)
Map.addLayer(carteOS)
```

Importation de la carte d'occupation du sol 

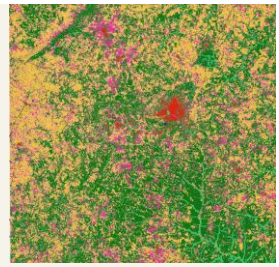
```
var carteOS = ee.Image('
projects/ee-
jauneatemps/assets/pastoralism/OBSYDYA_OS_2023_L1_Awanla')
Map.centerObject(carteOS)
Map.addLayer(carteOS)
```

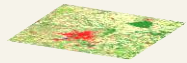
121

Importation de la carte d'occupation du sol 

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


var couleursOS = ['eebc1d', 'f1e5ac', 'f531c1', 'bb270d',
'4cbb17', '40e0d0', '355e3b', '008000', '8a9a5b', 'ff0000',
'fff8dc', '0000ff']; // liste de couleurs en hexadécimal
Map.addLayer(carteOS,
{min: 11, max: 61, palette: couleursOS},
'carteOS', false);
```

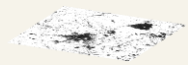
122

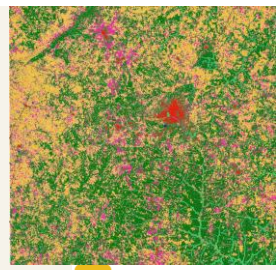


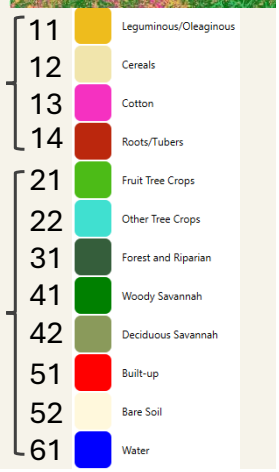
Importation de la carte d'occupation du sol 

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


var couleursOS = ['eebc1d', 'f1e5ac', 'f531c1', 'bb270d',
'4cbb17', '40e0d0', '355e3b', '008000', '8a9a5b', 'ff0000',
'fff8dc', '0000ff']; // liste de couleurs en hexadécimal
Map.addLayer(carteOS,
{min: 11, max: 61, palette: couleursOS},
'carteOS', false);
```

123

Production du masque culture/non-culture 





124

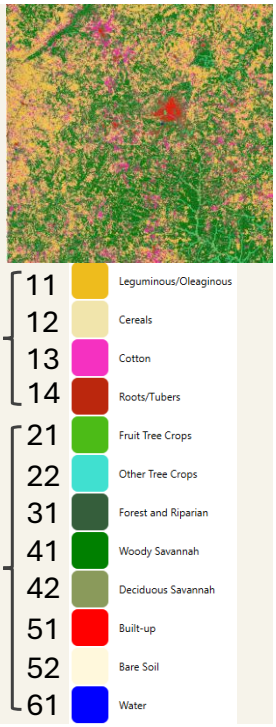
Utiliser Google Earth Engine Cas 5 – Indices

Production du masque culture/non-culture

```
var carteOSmask = carteOS.lte(14);
```

lower than or equal ≤

autres : .eq(), .lt(), .gt(), .gte()

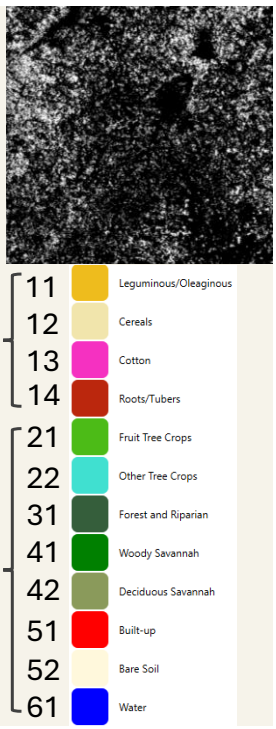


125

Utiliser Google Earth Engine Cas 5 – Indices

Production du masque culture/non-culture

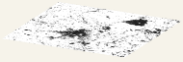
```
var carteOSmask = carteOS.lte(14);
Map.addLayer(carteOSmask, {}, 'mask');
```

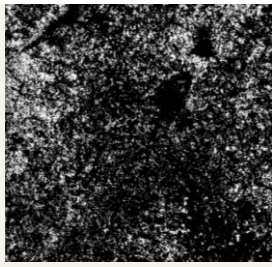


126

Utiliser Google Earth Engine Cas 5 – Indices

Production du masque culture/non-culture



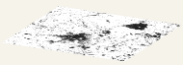


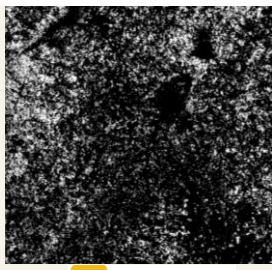
```
var carteOSmask = carteOS.lte(14);
Map.addLayer(carteOSmask, {}, 'mask');

var carteOSmasquee = carteOS.updateMask(carteOSmask);
Map.addLayer(carteOSmasquee, {min: 11, max: 61, palette:
couleursOS}, 'carteOSmasquee', false);
```

127

Utiliser Google Earth Engine Cas 5 – Indices

Production du masque culture/non-culture






```
var carteOSmask = carteOS.lte(14);
Map.addLayer(carteOSmask, {}, 'mask');
```

Autres cas pour masquer :

```
var mask = carteOS.eq(31).or(carteOS.eq(21));
var mask = carteOS.gte(21).and(carteOS.lte(42));
```

128

Utiliser Google Earth Engine Cas 5 – Indices

Production d'un indice de végétation

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

Dataset Availability
2015-06-27T00:00:00Z–2025-01-20T16:26:09Z

Revisit Interval
5 Days

Dataset Provider
European Union/ESA/Copernicus

Tags
copernicus esa eu msi radiance sentinel

Earth Engine Snippet
`ee.ImageCollection("COPERNICUS/S2_HARMONIZED")`

Description Bands Image Properties Terms of Use

Bands

Name	Scale	Pixel Size	Wavelength	Description
B1	0.0001	60 meters	443.9nm (S2A) / 442.3nm (S2B)	Aerosols
B2	0.0001	10 meters	496.6nm (S2A) / 492.1nm (S2B)	Blue
B3	0.0001	10 meters	560nm (S2A) / 559nm (S2B)	Green
B4	0.0001	10 meters	664.5nm (S2A) / 665nm (S2B)	Red
B5	0.0001	20 meters	703.9nm (S2A) / 703.8nm (S2B)	Red Edge 1
B6	0.0001	20 meters	740.2nm (S2A) / 739.1nm (S2B)	Red Edge 2
B7	0.0001	20 meters	782.5nm (S2A) / 779.7nm (S2B)	Red Edge 3
B8	0.0001	10 meters	835.1nm (S2A) / 833nm (S2B)	NIR

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

After 2022-01-25, Sentinel-2 scenes with PROCESSING_BASELINE '04.00' or above have their DN (value) range shifted by 1000. The HARMONIZED collection shifts data in newer scenes to be in the same range as in older scenes. Sentinel-2 is a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus ...

copernicus esa eu msi radiance sentinel

129

Utiliser Google Earth Engine Cas 5 – Indices

Production d'un indice de végétation

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

Dataset Availability
2015-06-27T00:00:00Z–2025-01-21T18:08:41Z

Revisit Interval
5 Days

Dataset Provider
European Union/ESA/Copernicus

Tags
copernicus esa eu msi radiance sentinel

Earth Engine Snippet
`ee.ImageCollection("COPERNICUS/S2_HARMONIZED")`

Description Bands Image Properties Terms of Use

Image Properties

Name	Type	Description
CLOUDY_PIXEL_PERCENTAGE	DOUBLE	Granule-specific cloudy pixel percentage taken from the original metadata
CLOUD_COVERAGE_ASSESSMENT	DOUBLE	Cloudy pixel percentage for the whole archive that contains this granule. Taken from the original metadata
DATASTRIP_ID	STRING	Unique identifier of the datastrip Product Data Item (PDI)
DATATAKE_IDENTIFIER	STRING	Uniquely identifies a given Datatake. The ID contains the Sentinel-2 satellite, start date and time, absolute orbit number, and processing baseline.
DATATAKE_TYPE	STRING	MSI operation mode
DEGRADED_MSI_DATA_PERCENTAGE	DOUBLE	Percentage of degraded MSI and ancillary data

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

After 2022-01-25, Sentinel-2 scenes with PROCESSING_BASELINE '04.00' or above have their DN (value) range shifted by 1000. The HARMONIZED collection shifts data in newer scenes to be in the same range as in older scenes. Sentinel-2 is a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus ...

copernicus esa eu msi radiance sentinel

130

Production d'un indice de végétation

Copier le code de la fonction et le code pour importer le dataset

Explore with Earth Engine

Code Editor (JavaScript) Colab (Python)

```
/**
 * Function to mask clouds using the Sentinel-2 QA band
 * @param {ee.Image} image Sentinel-2 image
 * @return {ee.Image} cloud masked Sentinel-2 image
 */
function maskS2clouds(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);
}

// Map the function over a month of data and take the median.
// Load Sentinel-2 TOA reflectance data (adjusted for processing changes
// that occurred after 2022-01-25).
var dataset = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
                  .filterDate('2022-01-01', '2022-01-31')
                  // Pre-filter to get less cloudy granules.
                  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
                  .map(maskS2clouds);
```

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

After 2022-01-25, Sentinel-2 scenes with PROCESSING_BASELINE '04.00' or above have their DN (value) range shifted by 1000. The HARMONIZED collection shifts data in newer scenes to be in the same range as in older scenes. Sentinel-2 is a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus ...

copernicus esa eu msi radiance sentinel

131

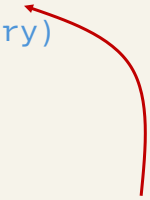
Production d'un indice de vegetation

```
var sentinel2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
      .filterDate('2023-02-01', '2023-03-31')
      .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
      .filterBounds(geometry)
      .map(maskS2clouds);
```

132

Production d'un indice de vegetation

```
var sentinel2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
      .filterDate('2023-02-01', '2023-03-31')
      .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
      .filterBounds(geometry)
      .map(maskS2clouds);
```

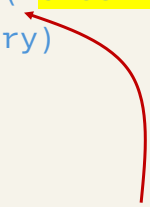


lower than <

133

Production d'un indice de vegetation

```
var sentinel2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
      .filterDate('2023-02-01', '2023-03-31')
      .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
      .filterBounds(geometry)
      .map(maskS2clouds);
```



lower than <

134

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

Dataset Availability
2015-06-27T00:00:00Z–2025-01-21T18:08:41Z

Revisit Interval
5 Days

Dataset Provider
European Union/ESA/Copernicus

Tags
copernicus esa eu msi radiance sentinel

Earth Engine Snippet
`ee.ImageCollection("COPERNICUS/S2_HARMONIZED")`

Description Bands Image Properties Terms of Use

Image Properties

Name	Type	Description
CLOUDY_PIXEL_PERCENTAGE	DOUBLE	Granule-specific cloudy pixel percentage taken from the original metadata
CLOUD_COVERAGE_ASSESSMENT	DOUBLE	Cloudy pixel percentage for the whole archive that contains this granule. Taken from the original metadata
DATASTRIP_ID	STRING	Unique identifier of the datastrip Product Data Item (PDI)
DATATAKE_IDENTIFIER	STRING	Uniquely identifies a given Datatake. The ID contains the Sentinel-2 satellite, start date and time, absolute orbit number, and processing baseline.
DATATAKE_TYPE	STRING	MSI operation mode
DEGRADED_MSI_DATA_PERCENTAGE	DOUBLE	Percentage of degraded MSI and ancillary data

Production d'un indice de vegetation

```
var sentinel2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
      .filterDate('2023-02-01', '2023-03-31')
      .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
      .filterBounds(geometry)
      .map(maskS2clouds);
var bandeNDVI = function(image) {
      var ndvi = image.normalizedDifference(['B8', 'B4'])
      .rename('NDVI');
      return image.addBands(ndvi);
};
```

Production d'un indice de vegetation

```
var bandeNDVI = function(image) {
     var ndvi = image.normalizedDifference(['B8', 'B4'])
     .rename('NDVI');
     return image.addBands(ndvi);
};

var sentinel2_ndvi = sentinel2.map(bandeNDVI).select('NDVI');
```

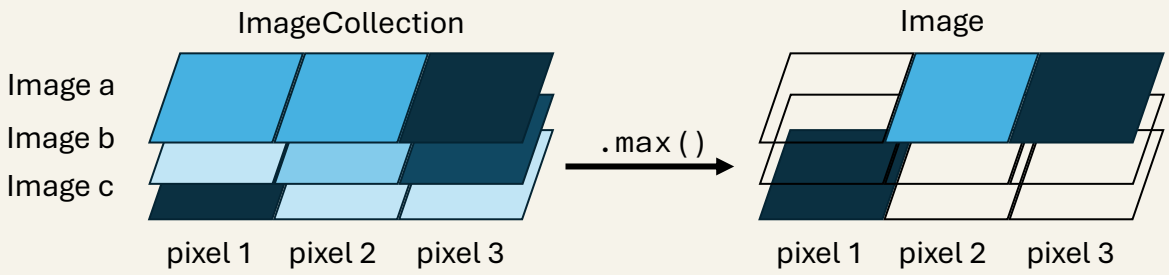
137

Production d'un indice de vegetation

```
var bandeNDVI = function(image) {
     var ndvi = image.normalizedDifference(['B8', 'B4'])
     .rename('NDVI');
     return image.addBands(ndvi);
};

var sentinel2_ndvi = sentinel2.map(bandeNDVI).select('NDVI');

var sentinel2_ndvi_max = sentinel2_ndvi.max();
```

138

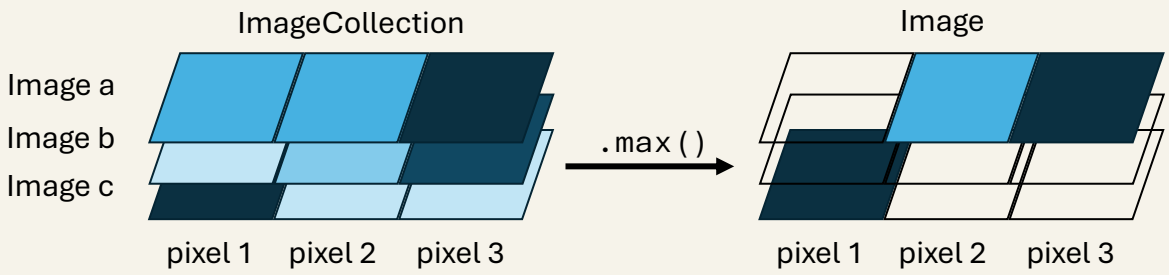
Production d'un indice de vegetation




```
var sentinel2_ndvi_max = sentinel2_ndvi.max();
```

139

Production d'un indice de vegetation



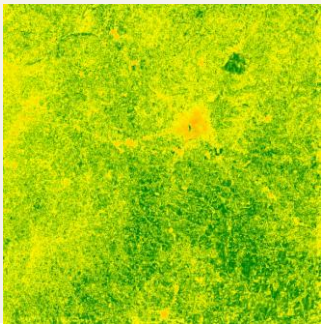

```
var sentinel2_ndvi_max = sentinel2_ndvi.max();
```

→ *Maximum Value Composite*

140

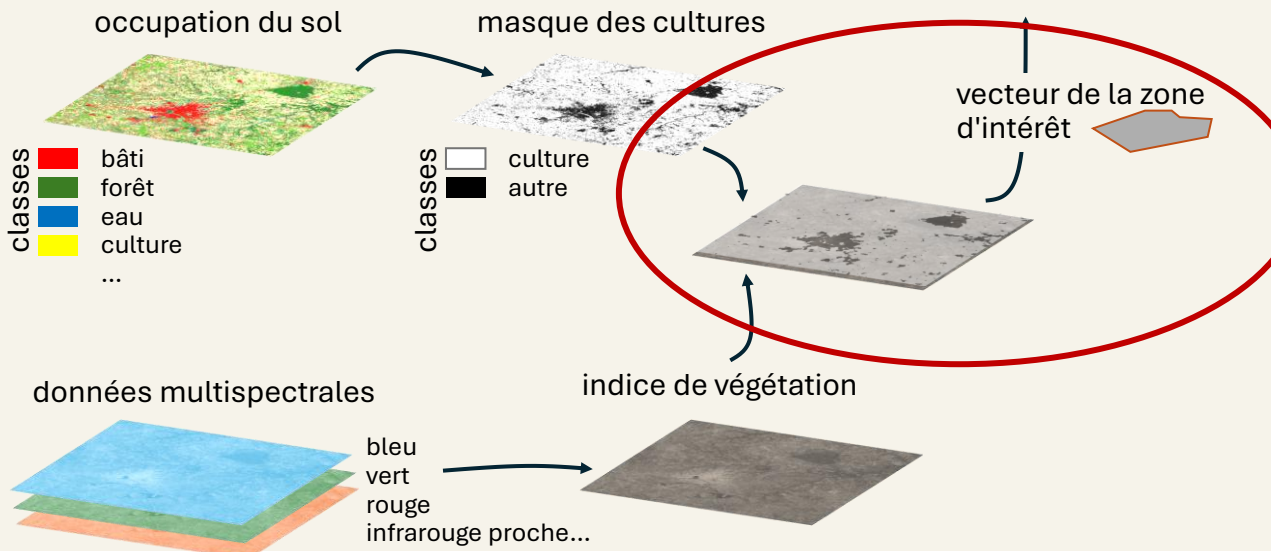
Production d'un indice de vegetation

```
var sentinel2_ndvi_max = sentinel2_ndvi.max();

Map.addLayer(sentinel2_ndvi_max, {min: 0, max: 0.5,
palette: ['orange', 'yellow', 'green']}, 'sentinel2_ndvi_max',
false);
```



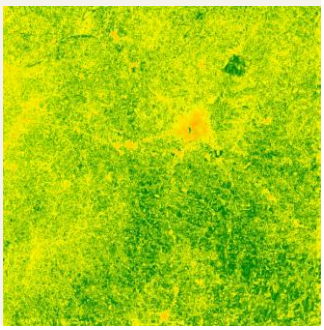
Calcul d'un indice de vegetation sur des zones cultivées

- carte de l'indice sur la zone d'intérêt
- valeur moyenne



Production d'un indice de vegetation

```
var sentinel2_ndvi_max = sentinel2_ndvi.max();


Map.addLayer(sentinel2_ndvi_max, {min: 0, max: 0.5,
palette: ['orange', 'yellow', 'green']}, 'sentinel2_ndvi_max',
false);


var sentinel2_ndvi_max_masque =
sentinel2_ndvi_max.updateMask(carteOSmask);
```



143

Production d'un indice de vegetation

```
var sentinel2_ndvi_max = sentinel2_ndvi.max();


Map.addLayer(sentinel2_ndvi_max, {min: 0, max: 0
palette: ['orange', 'yellow', 'green']}, 'sentin
false);


var sentinel2_ndvi_max_masque =
sentinel2_ndvi_max.updateMask(carteOSmask);
Map.addLayer(sentinel2_ndvi_max_masque, {min: 0, max: 0.5,
palette: ['orange', 'yellow', 'green']},
'sentinel2_ndvi_max_masque', false);
```

144

Extraction avec une zone d'intérêt

```
var sentinel2_ndvi_max_masque =
sentinel2_ndvi_max.updateMask(carteOSmask);
```

145

Extraction avec une zone d'intérêt

```
var sentinel2_ndvi_max_masque =
sentinel2_ndvi_max.updateMask(carteOSmask);


var zoneinteret = carteOS.geometry()
Map.addLayer(zoneinteret, {}, 'limites')
```

146

Extraction avec une zone d'intérêt

```
var zoneinteret = carteOS.geometry()
Map.addLayer(zoneinteret, {}, 'limites')

var NDVIcultures = sentinel2_ndvi_max_masque.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: zoneinteret,
  scale: 10,
  crs: 'EPSG:4326',
});
```

147

Extraction avec une zone d'intérêt

```
var zoneinteret = carteOS.geometry()
Map.addLayer(zoneinteret, {}, 'limites')

var NDVIcultures = sentinel2_ndvi_max_masque.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: zoneinteret,
  scale: 10,
  crs: 'EPSG:4326',
});
```

```
Dictionary (Error)
  Image.reduceRegion: Too many pixels in the region. Found 25528256, but
  maxPixels allows only 10000000.
  Ensure that you are not aggregating at a higher resolution than you
  intended; that is a frequent cause of this error. If not, then you may
  set the 'maxPixels' argument to a limit suitable for your computation;
  set 'bestEffort' to true to aggregate at whatever scale results in
  'maxPixels' total pixels; or both.
```

148

Extraction avec une zone d'intérêt

```
var zoneinteret = carteOS.geometry()
Map.addLayer(zoneinteret, {}, 'limites')

var NDVIcultures = sentinel2_ndvi_max_masque.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: zoneinteret,
  scale: 10,
  crs: 'EPSG:4326',
  maxPixels: 1e13
});
```

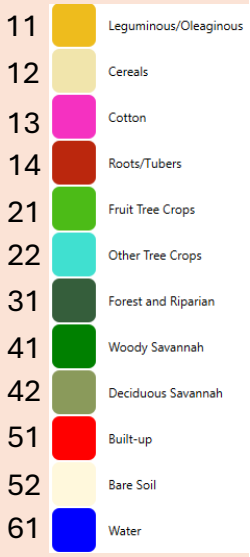
```
Dictionary (Error)
Image.reduceRegion: Too many pixels in the region. Found 25528256, but maxPixels allows only 10000000.
Ensure that you are not aggregating at a higher resolution than you intended; that is a frequent cause of this error. If not, then you may set the 'maxPixels' argument to a limit suitable for your computation; set 'bestEffort' to true to aggregate at whatever scale results in 'maxPixels' total pixels; or both.
```

149

Calcul d'un indice de vegetation sur des zones cultivées

Exercice :

Modifiez le code pour extraire le NDVI moyen des **vergers** ou des **cultures de céréales** des mois de **janvier et février** ou de **juillet et aout** en 2023



150

Notes sur la production d'indices

```
image.normalizedDifference(['B8', 'B4'])
```

$$\frac{B8 - B4}{B8 + B4}$$

```
var imageB8 = image.select('B8');
var imageB4 = image.select('B4');
imageB8.subtract(imageB4).divide(imageB8.add(imageB4))

image.expression('(NIR-RED)/(NIR+RED)', {
     'NIR' : image.select('B8'),
     'RED' : image.select('B4')
})
```

151

Animation simple du NDVI dans une région d'intérêt

Créer un polygone sur Parakou

```
var couleurs = {min: 0, max: 0.5, palette: ['orange', 'yellow',
'green']};

var cartes_ndvi = sentinel2_ndvi.map(function(image) {
  return image
    .visualize(couleurs)
    .clipToCollection(ee.FeatureCollection(geometry))
    .copyProperties(image, image.propertyNames());
});
```

152

Animation simple du NDVI dans une région d'intérêt

Créer un polygone sur Parakou

```
var videoArgs = {
  dimensions: 1000,
  region: geometry,
  framesPerSecond: 6,
  crs: 'EPSG:4326'
};

print(ui.Thumbnail(cartes_ndvi, arguments_video));
```


Calcul d'un indice de vegetation sur des zones cultivées : Conclusions

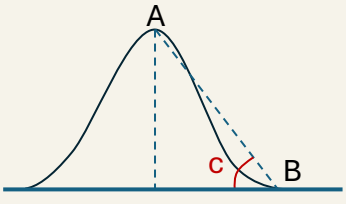
image.**lte**(nombre), mais aussi **lt**(), **gte**(), **gt**(), **eq**()
image.**updateMask**()
image.**addBands**()
image.**normalizedDifference**([])
image.**geometry**()
image.**rename**()

Algorithmes déjà présents dans GEE : topographie

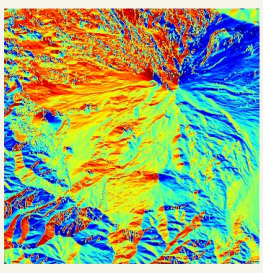
155

Algorithmes déjà présents dans GEE : topographie

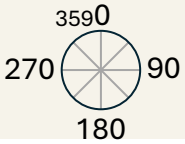
modèle numérique de terrain (MNT ou DEM)



pente moyenne (c)



aspect



estompage de pente (hillshade)

156

Algorithmes déjà présents dans GEE : topographie

“dem”

SRTM Digital Elevation Data Version 4

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

Dataset Provider
NASA/CGIAR

Earth Engine Snippet
`ee.Image("CGIAR/SRTM90_V4")`

Tags
dem elevation geophysical srtm topo
cgiar

Description Bands Terms of Use Citations

Bands

Name	Units	Min	Max	Pixel Size	Description
elevation	m	-444*	8806*	90 meters	Elevation

* estimated min or max value

157

Algorithmes déjà présents dans GEE : topographie

```
var mnt = ee.Image('CGIAR/SRTM90_V4');
var topographie = ee.Algorithms.Terrain(mnt);
print(topographie);
```

Home > Products > Google Earth Engine > Reference

Ce contenu vous a-t-il été utile ?

ee.Algorithms.Terrain

Send feedback

Calculates slope, aspect, and a simple hillshade from a terrain DEM.

Expects an image containing either a single band of elevation, measured in meters, or if there's more than one band, one named 'elevation'. Adds output bands named 'slope' and 'aspect' measured in degrees plus an unsigned byte output band named 'hillshade' for visualization. All other bands and metadata are copied from the input image. The local gradient is computed using the 4-connected neighbors of each pixel, so missing values will occur around the edges of an image.

158

Utiliser Google Earth Engine Cas 6 – Topographie

```
var mnt = ee.Image('CGIAR/SRTM90_V4');
var topographie = ee.Algorithms.Terrain(mnt);
print(topographie);
Map.addLayer(topographie.select('slope'),
{min: 0, max: 90, palette: ['white', 'black']});
```

159

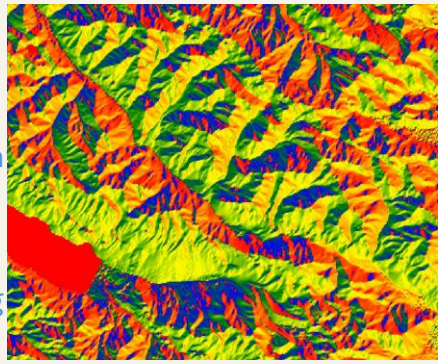
Utiliser Google Earth Engine Cas 6 – Topographie

```
var mnt = ee.Image('CGIAR/SRTM90_V4');
var topographie = ee.Algorithms.Terrain(mnt);
print(topographie);
Map.addLayer(topographie.select('aspect'),
{min: 0, max: 360, palette: ['red', 'orange',
'yellow', 'green', 'blue']});
```

160

Utiliser Google Earth Engine Cas 6 – Topographie

```
var mnt = ee.Image('CGIAR/SRTM90_V4');
var topographie = ee.Algorithms.Terrain(mn
print(topographie);
Map.addLayer(topographie.select('aspect'),
{min: 0, max: 360, palette: ['red', 'orang
'yellow', 'green', 'blue']});
```

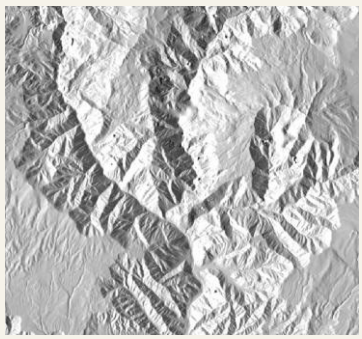


161

Utiliser Google Earth Engine Cas 6 – Topographie

Algorithmes déjà présents dans GEE : topographie

```
var mnt = ee.Image('CGIAR/SRTM90_V4');
var topographie = ee.Algorithms.Terrain(mnt);
print(topographie);
Map.addLayer(topographie.select('hillshade'));
```



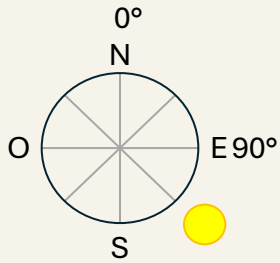
162

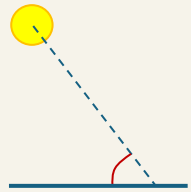
- ee.Terrain
  - aspect
  - fillMinima
  - hillShadow
  - hillshade
  - products
  - slope

Algorithmes déjà présents dans GEE : topographie

Il existe `ee.Algorithms.Terrain(mnt)`, mais aussi des fonctions dédiées :

```
ee.Terrain.aspect(mnt);
ee.Terrain.slope(mnt);
ee.Terrain.hillshade(mnt, azimuth, elevation);
```





Algorithmes déjà présents dans GEE : topographie

ee.Terrain.hillshade

Send feedback

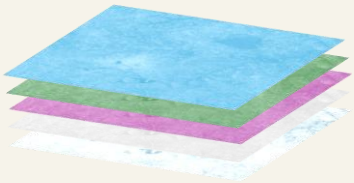
Computes a simple hillshade from a DEM.

Usage	Returns
`ee.Terrain.hillshade(input, azimuth, elevation)`	Image

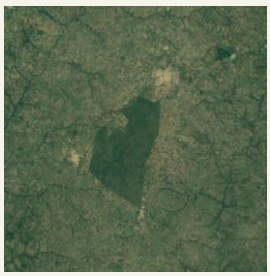
Argument	Type	Details
`input`	Image	An elevation image, in meters.
`azimuth`	*Float, default: 270*	*The illumination azimuth in degrees from north.*
`elevation`	*Float, default: 45*	*The illumination elevation in degrees.*

Images en vraie couleur

données multispectrales



bleu
vert
rouge
infrarouge proche...



165

Images en vraie couleur

données multispectrales : Sentinel 2 1C


```
/**
 * Function to mask clouds using the Sentinel-2 QA band
 * @param {ee.Image} image Sentinel-2 image
 * @return {ee.Image} cloud masked Sentinel-2 image
 */
function maskS2clouds(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);
}
```

166

Images en vraie couleur

```
var benin = ee.FeatureCollection('FAO/GAUL/2015/level0')
.filter(ee.Filter.eq('ADM0_NAME', 'Benin'));

var S2_2024 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
     .filterDate('2024-10-01', '2024-12-01')
     .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
     .map(maskS2clouds);
```

167

Images en vraie couleur

```
var composite = S2_2024.median().clip(benin);
```

168

Images en vraie couleur


```
var RVBVis = {
  min: 0.0,
  max: 0.3,
  bands: ['B4', 'B3', 'B2'],
};

Map.centerObject(benin, 7);
Map.addLayer(composite, RVBVis, 'VraieCouleur');
```

169

Essayer avec

```
.max()
.mean()
.min()
```

```
var composite = S2_2024.median().clip(benin);
```

170