

Introduction à Google Earth Engine

Cours pratique divisé en études de cas



1

Introduction à Google Earth Engine

Cas 8 : Détection des eaux de surface

Cas 9 : Classification

Cas 10 : Jointures spatiales

Cas 11 : Séries temporelles

Cas 12 : Détection de changement

Cas 13 : Régressions

Cas 14 : Tendances

Recommandations



2

Introduction à Google Earth Engine

Cas 8 : Détection des eaux de surface

Cas 9 : Classification

Cas 10 : Jointures spatiales

Cas 11 : Séries temporelles

Cas 12 : Détection de changement

Cas 13 : Régressions

Cas 14 : Tendances

Recommandations

Cas 9, 11-14 :

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



3

Introduction à Google Earth Engine

Cas 8 : Détection des eaux de surface

Cas 9 : Classification

Cas 10 : Jointures spatiales

Cas 11 : Séries temporelles

Cas 12 : Détection de changement

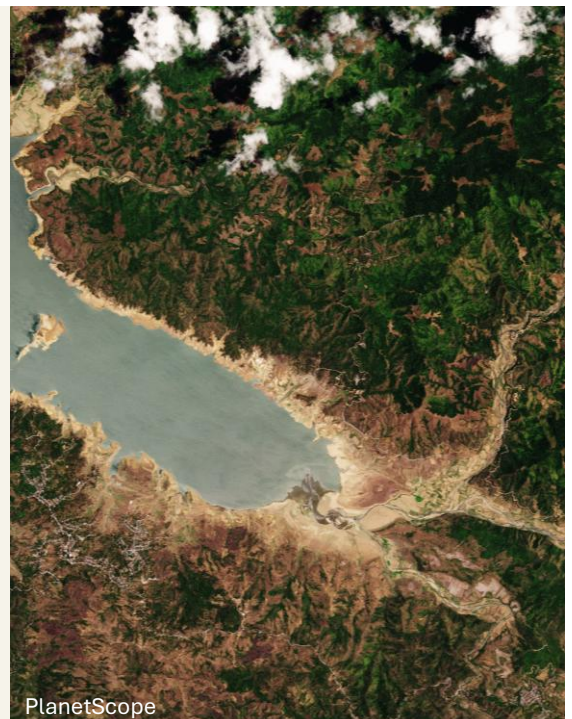
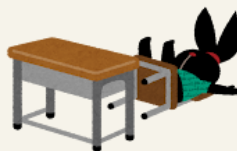
Cas 13 : Régressions

Cas 14 : Tendances

Recommandations

Cas 9, 11-14 :

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



4

Question 1

Quels types de données puis-je importer dans GEE?

- a) Rasters et vecteurs
- b) Raster, vecteurs, et vidéos
- c) Documents Word et rasters
- d) Tout type de donnée

5

Question 2

Comment est appelé un raster (.tif, ...) une fois importé dans GEE?

- a) Un googleraster ou googlerastercollection
- b) Une image ou imagecollection
- c) Un geotiff
- d) Un raster

6

Question 3

A quoi sert `ImageCollection.map`(fonction) ?

- a) A projeter une fonction sur la carte
- b) A appliquer une fonction sur l'`ImageCollection`
- c) A modifier une fonction
- d) A créer une fonction

7

Question 4

Sur une image avec des données entre 0 et 100, comment puis-je masquer les pixels inférieures à 49 ?

- a) Ce n'est pas possible sur GEE
- b) J'ajoute 49 à chaque valeur comme ça il ne reste plus de données < 49
- c) J'utilise un masque : `image.updateMask(image.lt(49))`
- d) J'utilise un masque : `image.updateMask(image.gt(48))`

8

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

Détection des eaux de surface au Nord Bénin

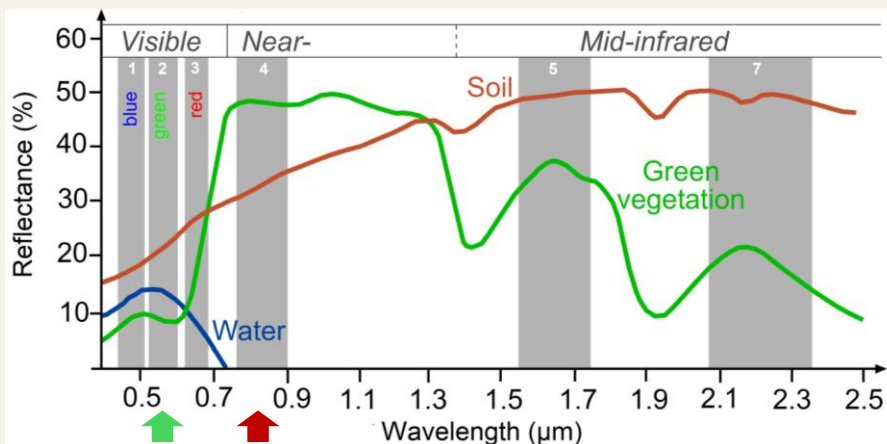


9

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

Réflectance de l'eau



Réflectance de l'eau
 ≠/≠ végétation, sol

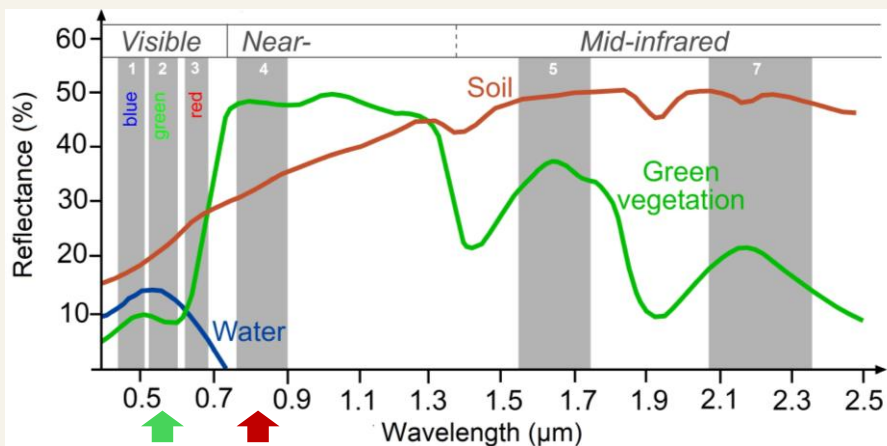
$$NDWI = \frac{Vert - PIR}{Vert + PIR}$$

10

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

Réflectance de l'eau



Réflectance de l'eau
 ≠ végétation, sol

$$NDWI = \frac{Vert - PIR}{Vert + PIR}$$

Seuil pour catégoriser l'eau ?

-0.1?

11

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

Détection des eaux de surface au Nord Bénin



```
var S2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
  .filterDate('2023-01-01', '2023-03-31')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
  .map(maskS2clouds);
```

12

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

```

var S2 = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
    .filterDate('2023-01-01', '2023-03-31')
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
    .map(maskS2clouds);

var NDWIfon = function(image) {
  var ndwi = image.normalizedDifference(['B3', 'B8']).rename('NDWI');
  return image.addBands(ndwi);
};
var S2ndwi = S2.map(NDWIfon).select('NDWI');
var S2ndwimax = S2ndwi.max();

```

13

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

```

var S2ndwi = S2.map(NDWIfon).select('NDWI');
var S2ndwimax = S2ndwi.max();

var seuil = ee.Number(-0.1);
var masque = S2ndwimax.gte(seuil);
var S2ndwimax_m = S2ndwimax.updateMask(masque);

Map.addLayer(S2ndwimax_m);
Map.setCenter(2.78527, 10.71729, 15); // ville de Sori

```

14

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

```

var S2ndwi = S2.map(NDWIfon).select('NDWI');
var S2ndwimax = S2ndwi.max();

var seuil = ee.Number(-0.1);
var masque = S2ndwimax.gte(seuil);
var S2ndwimax_m = S2ndwimax.updateMask(masque);

Map.addLayer(S2ndwimax_m);
Map.setCenter(2.78527, 10.71729, 15); // ville de Sori

```

Beaucoup de pixels ne semblent pas appartenir à de l'eau, trop petits groupes

15

Utiliser Google Earth Engine

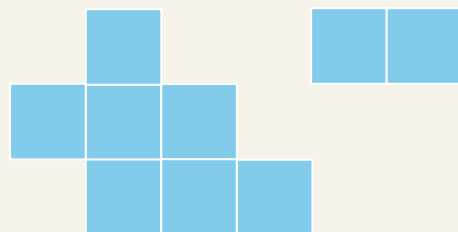
Cas 8 – Détection des eaux de surface

```
Image.connectedPixelCount();
```

Usage

```
Image.connectedPixelCount(maxSize, eightConnected)
```

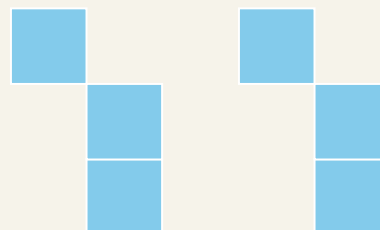
Argument	Type	Details
this: input	Image	The input image.
maxSize	Integer, default: 100	The maximum size of the neighborhood in pixels.
eightConnected	Boolean, default: true	Whether to use 8-connected rather 4-connected rules.



eightConnected:

false

true



16

Utiliser Google Earth Engine

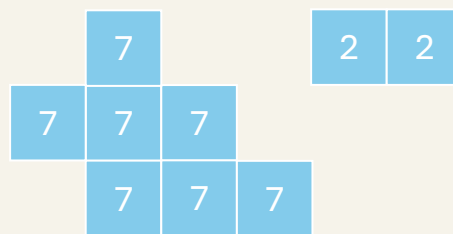
Cas 8 – Détection des eaux de surface




```
Image.connectedPixelCount();
```

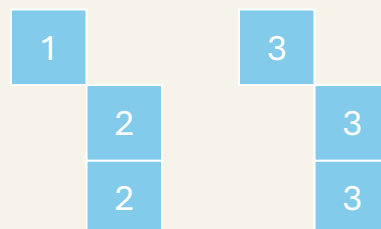
Usage

```
Image.connectedPixelCount(maxSize, eightConnected)
```

Argument	Type	Details
this: input	Image	The input image.
maxSize	Integer, default: 100	The maximum size of the neighborhood in pixels.
eightConnected	Boolean, default: true	Whether to use 8-connected rather 4-connected rules.



eightConnected: 
false  true 



17

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

```
var S2ndwimax_m_px_lgrp = S2ndwimax_m.int()
    .connectedPixelCount(100, true)
    .reproject({
      crs: 'EPSG:4326',
      scale: 10
    });
```

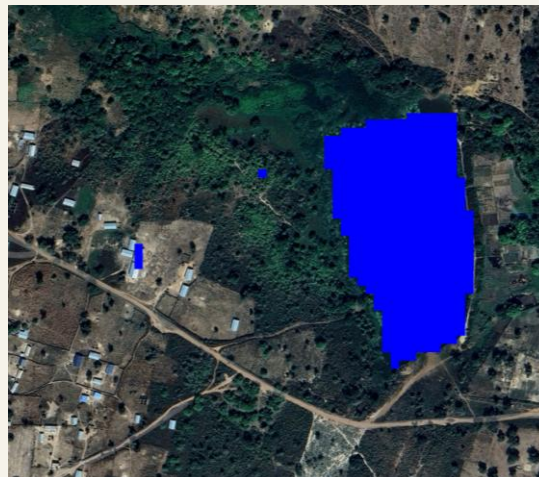
```
Map.addLayer(S2ndwimax_m_px_lgrp, {}, 'connectedPixelCount');
var masque_grp = S2ndwimax_m_px_lgrp.gte(5); // masque groups < 5 pixels
var S2ndwimax_m_g = S2ndwimax_m.updateMask(masque_grp);
Map.addLayer(S2ndwimax_m_g);
```

18

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

Zones identifiées comme de l'eau



19

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

Détection des eaux de surface au Nord Bénin

Question :

Quel type de surface est confondu avec l'eau ?

Comment remédier à ce problème ?

20

Utiliser Google Earth Engine

Cas 8 – Détection des eaux de surface

Détection des eaux de surface au Nord Bénin

Exercice :

Utiliser le NDWI moyen au lieu du NDWI maximum.

21

Utiliser Google Earth Engine

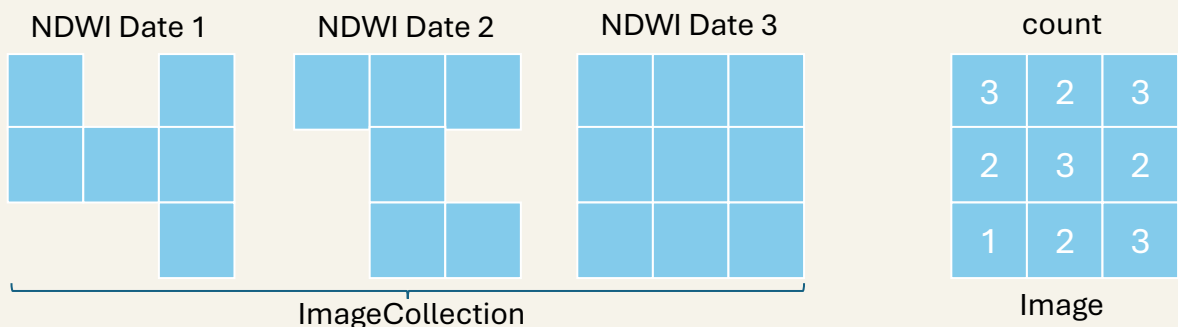
Cas 8 – Détection des eaux de surface

Ajout d'une étape lors de l'utilisation de la moyenne :
ne considérer que les pixels avec au moins x observations sur la période d'étude.

22

Ajout d'une étape lors de l'utilisation de la moyenne :
ne considérer que les pixels avec au moins x observations sur la période d'étude.

La fonction `ImageCollection.count()`



23

Ajout d'une étape lors de l'utilisation de la moyenne :
ne considérer que les pixels
avec au moins x observations sur la période d'étude.

La fonction `ImageCollection.count()`

```
var S2ndwicount = S2ndwi.count();
var masque_count = S2ndwicount.gte(3); //au moins 3 observations

var S2ndwimax_m_g_c = S2ndwimax_m_g.updateMask(masque_count);
Map.addLayer(S2ndwimax_m_g_c);
```

24

Exercice :

Modifier le code pour détecter l'eau entre Juin et Septembre avec un seuil de 6 observations au minimum.

25

Vectorisation

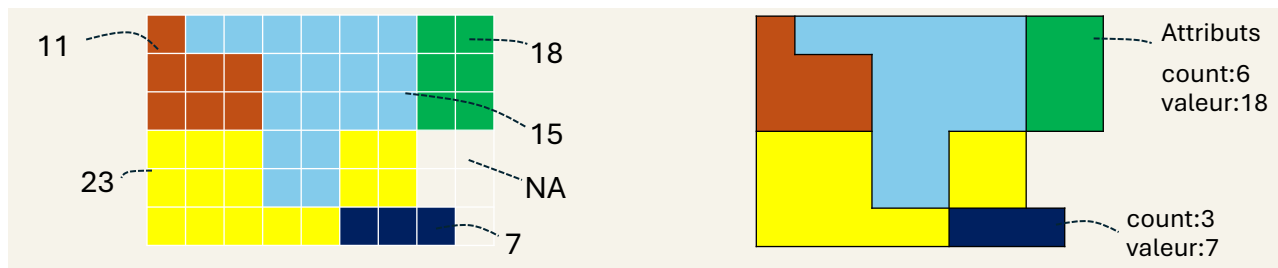
`Image.reduceToVectors()`

ee.Image.reduceToVectors


[Send feedback](#)

Convert an image to a feature collection by reducing homogeneous regions. Given an image containing a band of labeled segments and zero or more additional bands, runs a reducer over the pixels in each segment producing a feature per segment.

Either the reducer must have one fewer inputs than the image has bands, or it must have a single input and will be repeated for each band.



26

Vectorisation

`Image.reduceToVectors()`

```
var S2ndwimax_vect = S2ndwimax_m_grp.reduceToVectors({
  geometry: region,
  crs: 'EPSG:4326',
  scale: 20,
  eightConnected: true,
  labelProperty: 'valeur'
});
```

Créer un polygone “region” autour de Sori

27

Détection des eaux de surface au Nord Bénin : Conclusions

Production d'indices

`Image.connectedPixelCount()`

`ImageCollection.count()`

`Image.reduceToVectors()`

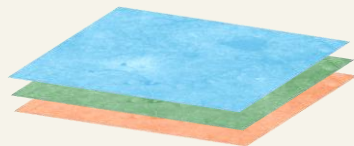
28

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Classification d'image

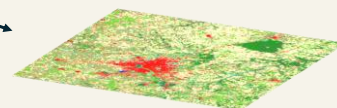
données multispectrales



bleu
vert
rouge
infrarouge proche...



classification

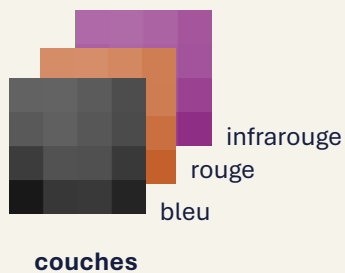


29

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Méthodes



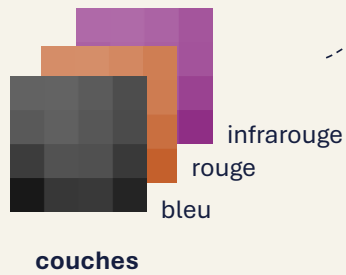
Frohn & Lopez (2018)

30

Utiliser Google Earth Engine

Cas 9 – Classification d'image

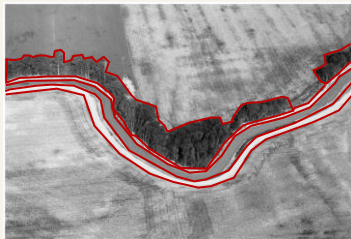
Méthodes



délimitation
d'éléments /
occupation
du sol

entière

**classification
manuelle**



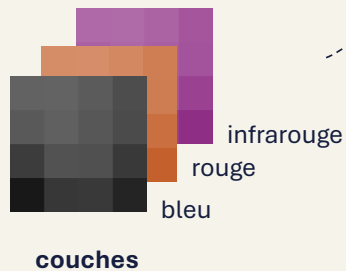
Frohn & Lopez (2018)

31

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Méthodes



délimitation
d'éléments /
occupation
du sol

entière

**classification
manuelle**

**classification
non-supervisée**



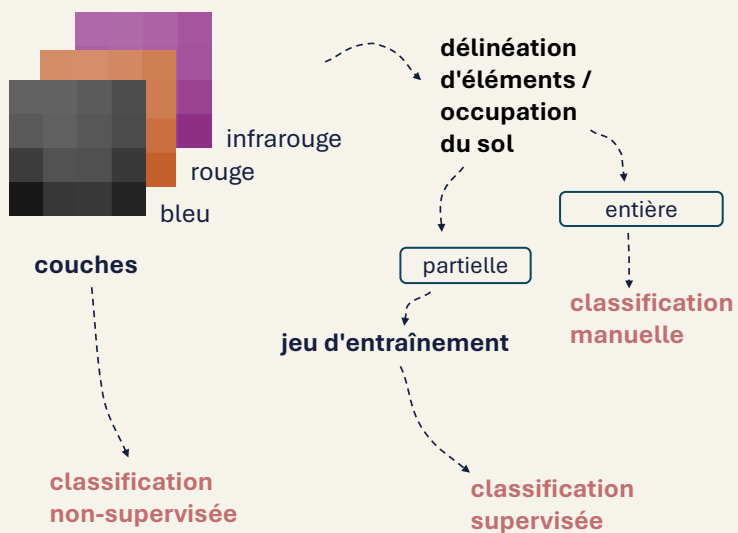
Frohn & Lopez (2018)

32

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Classification d'image



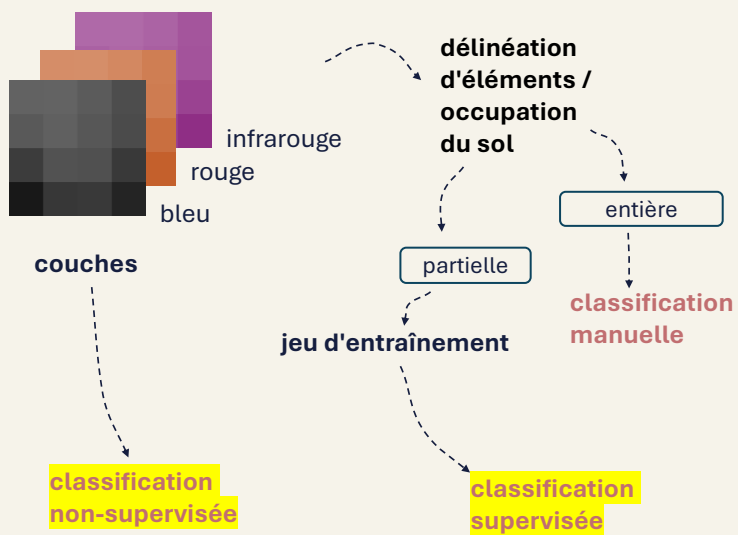
Frohn & Lopez (2018)

33

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Méthodes



Frohn & Lopez (2018)

34

Utiliser Google Earth Engine

Cas 9 – Classification d'image

```
var carteOS = ee.Image('
projects/ee-jauneatemps/assets/pastoralism/OBSYDYA_OS_2023_L1_Awanla')
var site_etude = carteOS.geometry();
```

35

Utiliser Google Earth Engine

Cas 9 – Classification d'image

```
var carteOS = ee.Image('
projects/ee-jauneatemps/assets/pastoralism/OBSYDYA_OS_2023_L1_Awanla')
var site_etude = carteOS.geometry();

var landsat = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
  .filterBounds(site_etude )
  .filterDate('2023-01-01', '2023-03-31')
  .sort('CLOUD_COVER')
  .first();
```

36

Utiliser Google Earth Engine

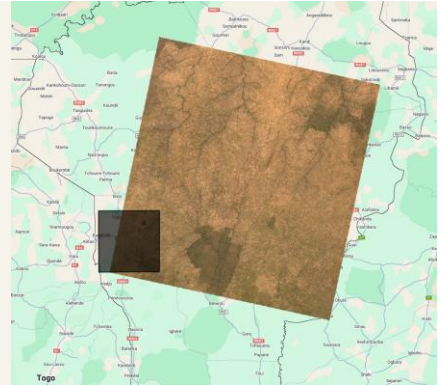
Cas 9 – Classification d'image

```

var landsat_vis = {
  bands: ['SR_B4', 'SR_B3', 'SR_B2'],
  min: 7000,
  max: 15000
};
Map.centerObject(site_etude, 10);
Map.addLayer(landsat, landsat_vis, 'Image L8');
Map.addLayer(site_etude)

```

= l'image utilisée pour l'entraînement

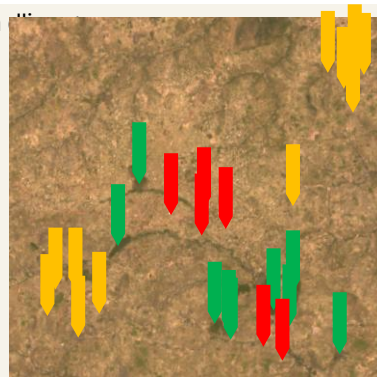


37

Utiliser Google Earth Engine

Cas 9 – Classification

Création des points d'entraînement



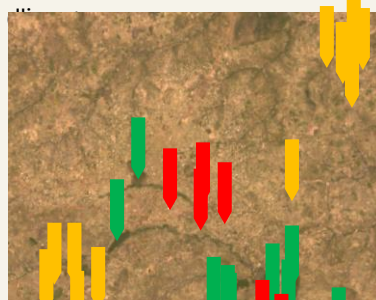
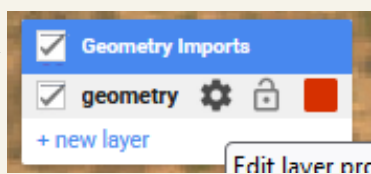
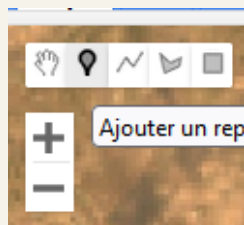
classe	code
Forêt	0
Bâti	1
Eau	2
Végétation basse	3
Sol nu	4

38

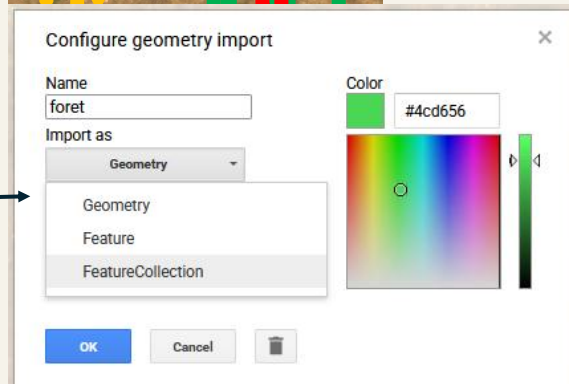
Utiliser Google Earth Engine

Cas 9 – Classification

Création des points d'entraînement



classe	code
Forêt	0
Bâti	1
Eau	2
Végétation basse	3
Sol nu	4

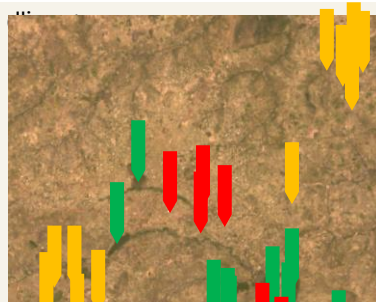
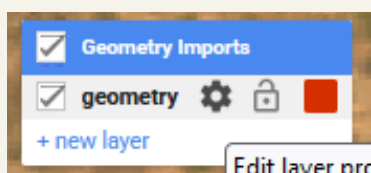
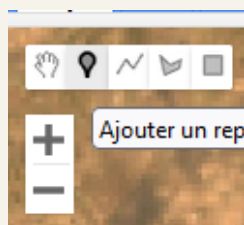


39

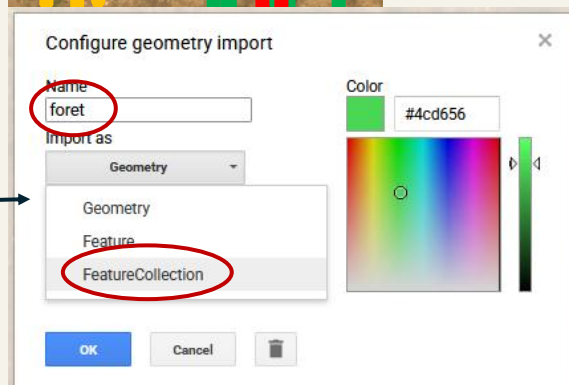
Utiliser Google Earth Engine

Cas 9 – Classification

Création des points d'entraînement



classe	code
Forêt	0
Bâti	1
Eau	2
Végétation basse	3
Sol nu	4

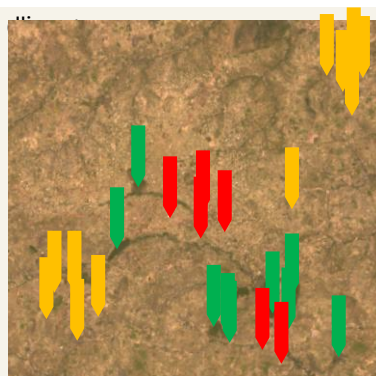


40

Utiliser Google Earth Engine

Cas 9 – Classification

Création des points d'entraînement



classe	code
Forêt	0
Bâti	1
Eau	2
Végétation basse	3
Sol nu	4

Configure geometry import

Name:

Color:

Import as:

Properties:

Properties

Property	Value
classe	<input type="text" value="0"/>

41

Utiliser Google Earth Engine

Cas 9 – Classification d'image

New Script *

Imports (1 entry) `var foret: FeatureCollection (0 elements)`

Configure geometry import

Name:

Color:

Import as:

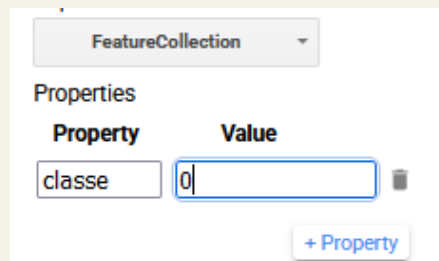
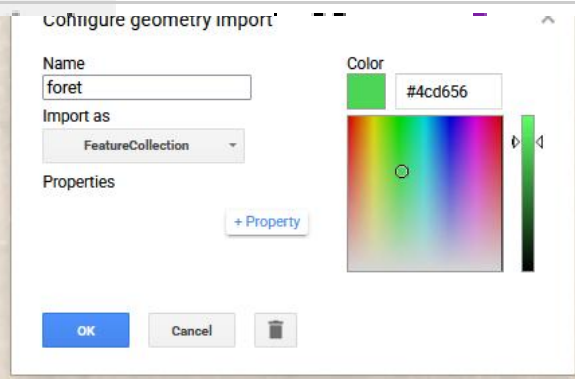
Properties:

Properties

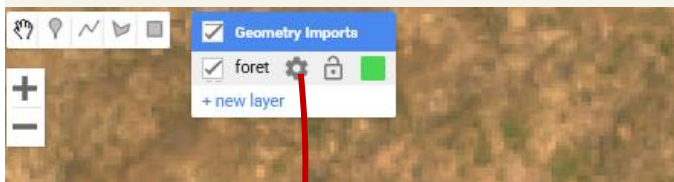
Property	Value
classe	<input type="text" value="0"/>

42

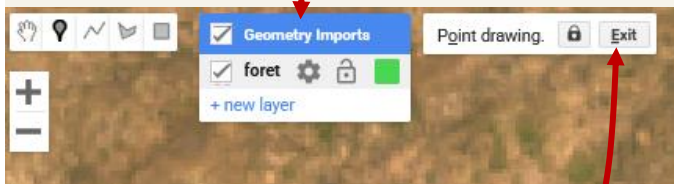
New Script *

Imports (1 entry) var foret: FeatureCollection (0 elements) 

43



activation du mode d'ajout de points

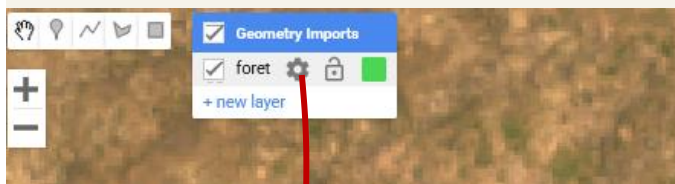


pour terminer l'ajout de points

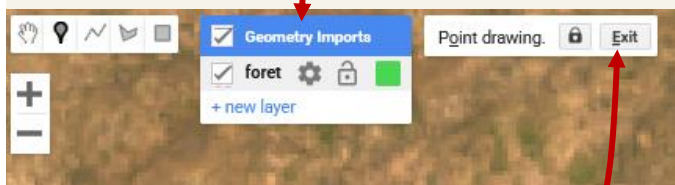


L'outil "main" permet de sélectionner un point et de le déplacer ou supprimer

44



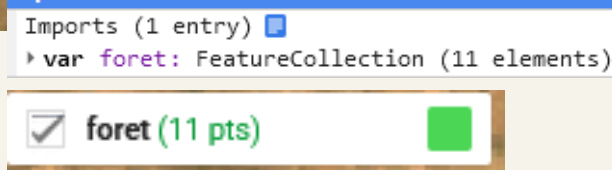
activation du mode d'ajout de points



pour terminer l'ajout de points

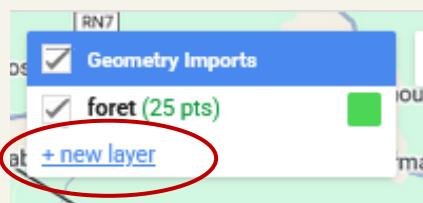







L'outil "main" permet de sélectionner un point et de le déplacer ou supprimer



45





Création des points d'entraînement

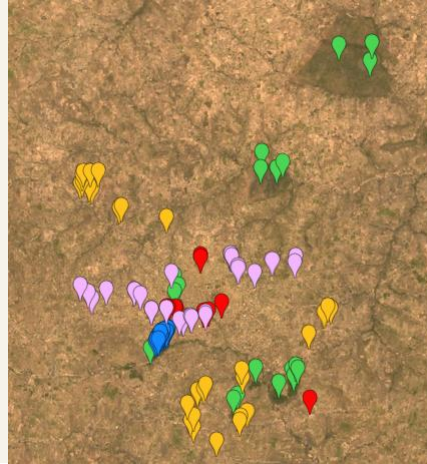
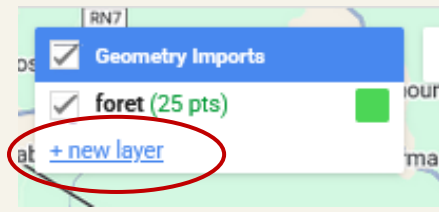


classe	code	nombre de points
Foret	0 	25
Bâti	1 	25
Eau	2 	25
Végétation basse	3 	25
Sol nu	4 	25

46

Création des points d'entraînement

classe	code		nombre de points
foret	0		25
bati	1		25
eau	2		25
vegetation basse	3		25
solnu	4		25



47

Extraction des signatures spectrales aux points d'entraînement

```
var entrainement = ee.FeatureCollection([foret, bati, eau, vegetationbasse,
solnu]).flatten();
print(entrainement);
```

transforme une FeatureCollection de
FeatureCollections en FeatureCollection

48

Extraction des signatures spectrales aux points d'entraînement

```
var entrainement = ee.FeatureCollection([foret, bati, eau, vegetationbasse,
so\lnu]).flatten();
print(entrainement);
```

transforme une FeatureCollection de
FeatureCollections en FeatureCollection

Sur une liste de listes, `flatten()` produira une liste
simple : `[['a', 'b'], ['c', 'd']]` deviant `['a', 'b', 'c', 'd']`

49

Extraction des signatures spectrales aux points d'entraînement

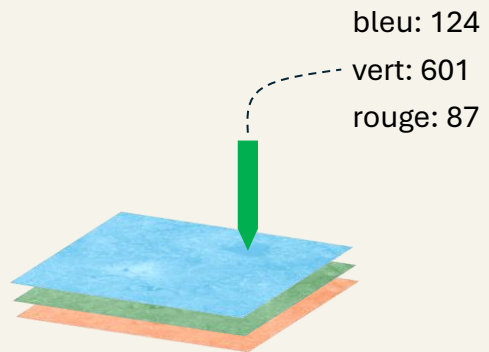
```
var entrainement = ee.FeatureCollection([foret, bati, eau,
vegetationbasse, so\lnu]).flatten();
print(entrainement);
```

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

50

Extraction des signatures spectrales aux points d'entraînement

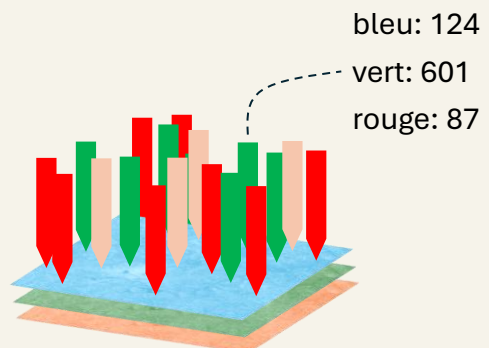
```
var echantillon = landsat.select(bandes)
  .sampleRegions({
    collection: entrainement,
    properties: ['classe'],
    scale: 30
  });
print(echantillon);
```



51

Extraction des signatures spectrales aux points d'entraînement

```
var echantillon = landsat.select(bandes)
  .sampleRegions({
    collection: entrainement,
    properties: ['classe'],
    scale: 30
  });
print(echantillon);
```



52

Extraction des signatures spectrales aux points d'entraînement

```
var classifieur = ee.Classifier.smileCart().train({
  features: echantillon,
  classProperty: 'classe',
  inputProperties: bandes
});
```

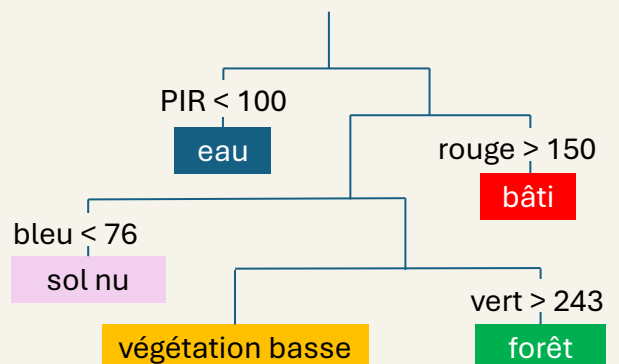
ee.Classifier

```
ee.Classifier.amnhMaxent(cat...
ee.Classifier.decisionTree(tree...
ee.Classifier.decisionTreeEnse...
ee.Classifier.libsvm(decisionPr...
ee.Classifier.load(id)
ee.Classifier.minimumDistan...
ee.Classifier.smileCart(maxNo...
ee.Classifier.smileGradientTre...
ee.Classifier.smileKNN(k, sear...
ee.Classifier.smileNaiveBayes...
ee.Classifier.smileRandomFor...
ee.Classifier.spectralRegion(c...
confusionMatrix()
explain()
```

53

Extraction des signatures spectrales aux points d'entraînement

```
var classifieur = ee.Classifier.smileCart().train({
  features: echantillon,
  classProperty: 'classe',
  inputProperties: bandes
});
```

**Arbre de décision**

54

Extraction des signatures spectrales aux points d'entraînement

```
var classifieur = ee.Classifier.smileCart().train({
  features: echantillon,
  classProperty: 'classe',
  inputProperties: bandes
});

var classification = landsat.select(bandes).classify(classifieur);
```

55

Extraction des signatures spectrales aux points d'entraînement

```
var classification = landsat.select(bandes).classify(classifieur);

var classificationVis = {
  min: 0,
  max: 4,
  palette: ["green", "red", "blue", "yellow", "pink"]
};
```

56

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Extraction des signatures spectrales aux points d'entraînement

```

var classification = landsat.select(bandes).classify(classifier);

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

Map.addLayer(classification, classificationVis, 'classificationsupervCART');

```

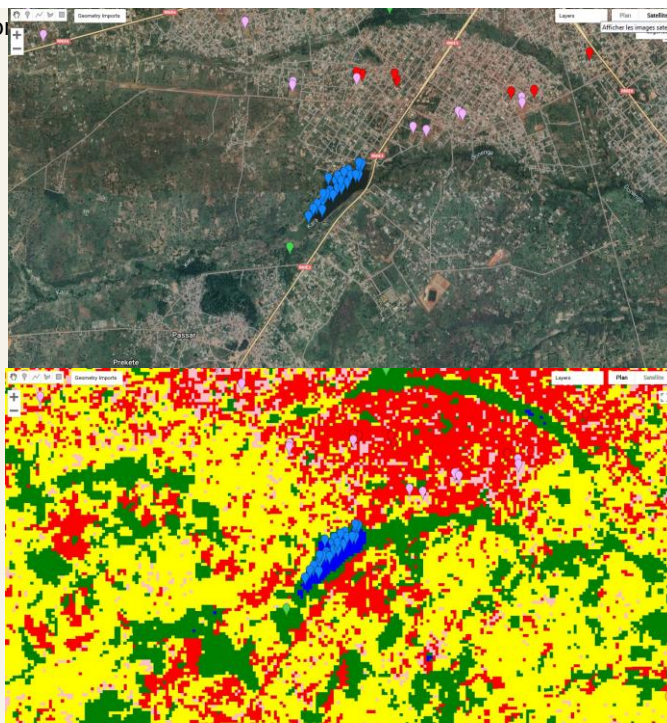
57

Utiliser Google Earth Engine

Cas 9 – Classification

Résultat

classe	code	nombre de points
Forêt	0	25
Bâti	1	25
Eau	2	25
Végétation basse	3	25
Sol nu	4	25



58

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Précision de la classification

Pour chaque classe (forêt, ...) :

Valeurs prédites par la classification**Valeurs réelles**

	Valeurs réelles	
	Positif	Négatif
Positif	Vrai positif	Faux positif
Négatif	Faux négatif	Vrai négatif

59

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Précision de la classification

Pour chaque classe (forêt, ...) :

Valeurs prédites par la classification**Valeurs réelles**

	Valeurs réelles	
	Positif	Négatif
Positif	Vrai positif	Faux positif
Négatif	Faux négatif	Vrai négatif

Valeurs prédites par la classification**Valeurs réelles**

	Valeurs réelles	
	Positif	Négatif
Positif	50	10
Négatif	5	35

60

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Précision de la classification

Pour chaque classe (forêt, ...) :

$$\text{Précision} = \frac{VP + VN}{\text{Total}} = \frac{50 + 35}{100}$$

$$\text{Précision (Producer)} = \frac{VP}{VP + FN} = \frac{50}{50 + 5}$$

$$\text{Précision (User)} = \frac{VP}{VP + FP} = \frac{50}{50 + 10}$$

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	Vrai positif	Faux positif
Négatif	Faux négatif	Vrai négatif

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	50	10
Négatif	5	35

61

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Précision de la classification

Pour chaque classe (forêt, ...) :

$$\text{Précision} = \frac{VP + VN}{\text{Total}} = \frac{50 + 35}{100}$$

$$\text{Précision (Producer)} = \frac{VP}{VP + FN} = \frac{50}{50 + 5}$$

$$\text{Précision (User)} = \frac{VP}{VP + FP} = \frac{50}{50 + 10}$$

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	Vrai positif	Faux positif
Négatif	Faux négatif	Vrai négatif

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	50	10
Négatif	5	35

62

Précision de la classification

Pour chaque classe (forêt, ...) :

$$\text{Précision} = \frac{VP + VN}{Total} = \frac{50 + 35}{100}$$

$$\text{Précision (Producer)} = \frac{VP}{VP + FN} = \frac{50}{50 + 5}$$

$$\text{Précision (User)} = \frac{VP}{VP + FP} = \frac{50}{50 + 10}$$

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	Vrai positif	Faux positif
Négatif	Faux négatif	Vrai négatif

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	50	10
Négatif	5	35

Erreur d'omission
= 100 – Producer

Erreur de commission
= 100 – User

63

Précision de la classification

Pour chaque classe (forêt, ...) :

Coefficient Kappa [-1; 1]

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	Vrai positif	Faux positif
Négatif	Faux négatif	Vrai négatif

Valeurs prédites par la classification

	Valeurs réelles	
	Positif	Négatif
Positif	50	10
Négatif	5	35

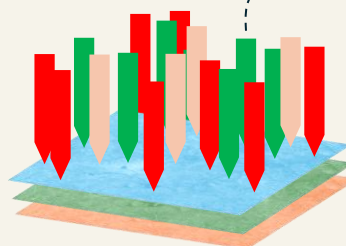
64

Précision de la classification

- Jeu de données d'entraînement
- Jeu de données de test

La classification produite avec les données d'entraînement est testée sur le second jeu de données

l'échantillonnage des
différentes couvertures
du sol



bleu: 124

vert: 601

rouge: 87

...

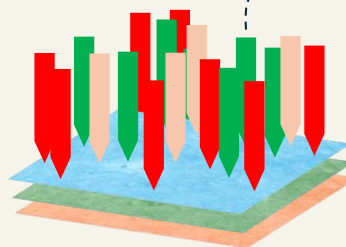
65

Précision de la classification

- Jeu de données d'entraînement (**80%**)
- Jeu de données de test (**20%**)

La classification produite avec les données d'entraînement est testée sur le second jeu de données

l'échantillonnage des
différentes couvertures
du sol



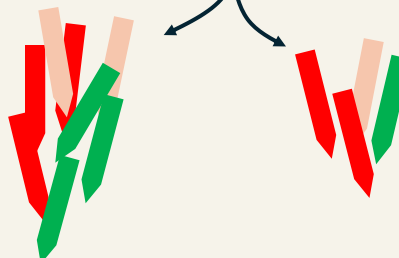
bleu: 124

vert: 601

rouge: 87

...

Sélection aléatoire des points



66

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Précision de la classification

```
var traintest =
echantillon.randomColumn();
```

'random' entre 0 et 1 (distribution uniforme)

l'échantillonnage des
différentes couvertures
du sol

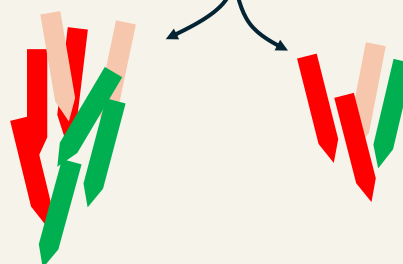
bleu: 124
vert: 601
rouge: 87

...

random: 0,5



Sélection aléatoire des points



67

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Précision de la classification

```
var traintest =
echantillon.randomColumn();
```

```
var groupeTrain = traintest
.filter(ee.Filter.lt('random', 0.8));
```

```
var groupeTest = traintest
.filter(ee.Filter.gte('random', 0.8));
```

```
print(groupeTrain);
print(groupeTest);
```

l'échantillonnage des
différentes couvertures
du sol

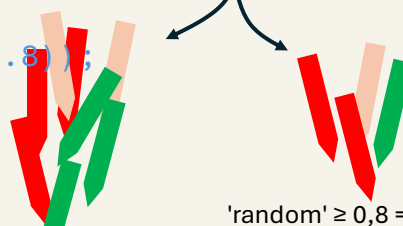
bleu: 124
vert: 601
rouge: 87

...

random: 0,5



Sélection aléatoire des points



'random' < 0,8 = groupeTrain

'random' ≥ 0,8 = groupeTest

68

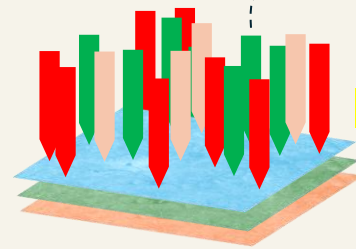
Précision de la classification

```
var classifieur2 =
ee.Classifier.smileCart().train({
  features: groupeTrain,
  classProperty: 'classe',
  inputProperties: bandes
});
```

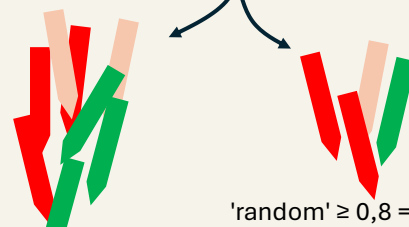
l'échantillonnage des
différentes couvertures
du sol

bleu: 124
vert: 601
rouge: 87
...

random: 0,5



Sélection aléatoire des points



'random' < 0,8 = groupeTrain

'random' ≥ 0,8 = groupeTest

69

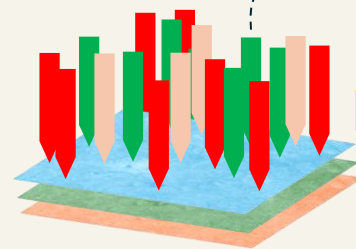
Précision de la classification

```
var matriceConfusion =
groupeTest.classify(classifieur2)
.errorMatrix({
  actual: 'classe',
  predicted: 'classification'
});
```

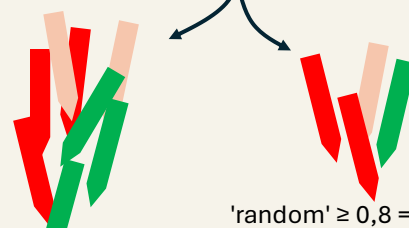
l'échantillonnage des
différentes couvertures
du sol

bleu: 124
vert: 601
rouge: 87
...

random: 0,5



Sélection aléatoire des points



'random' < 0,8 = groupeTrain

'random' ≥ 0,8 = groupeTest

70

Précision de la classification

```
print('Matrice de confusion :', matriceconfusion);
print('Précision:', matriceconfusion.accuracy());
print('Précision (prod) :', matriceconfusion.producersAccuracy());
print('Précision (user):', matriceconfusion.consumersAccuracy());
print('Kappa :', matriceconfusion.kappa());
```

```
ee.ConfusionMatrix
ee.ConfusionMatrix(array, ord...
accuracy()
array()
consumersAccuracy()
fscore(beta)
kappa()
order()
producersAccuracy()
```

```
Matrice de confusion :
List (5 elements)
▶ 0: [5,0,0,0,0]
▶ 1: [0,5,0,0,1]
▶ 2: [1,0,4,0,0]
▶ 3: [0,0,0,2,2]
▶ 4: [0,1,0,0,1]
0 1 2 3 4
```

71

GEE propose d'autres algorithmes de classification, Random Forest par exemple :

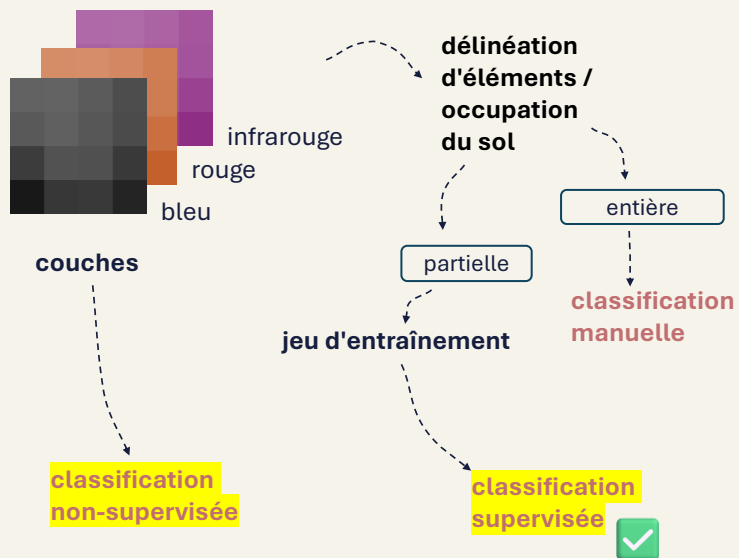
```
var classifieur = ee.Classifier.smileRandomForest(50).train({
  features: echantillon,
  classProperty: 'classe',
  inputProperties: bandes
});
```

72

Utiliser Google Earth Engine

Cas 9 – Classification d'image

Classification non-supervisée

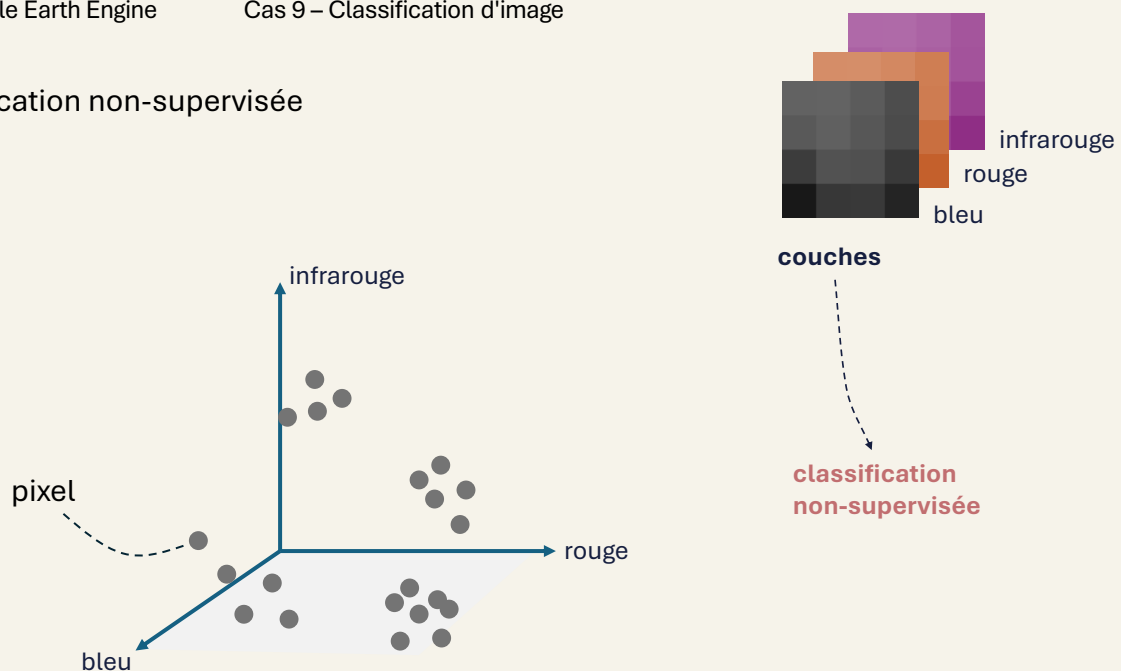


73

Utiliser Google Earth Engine

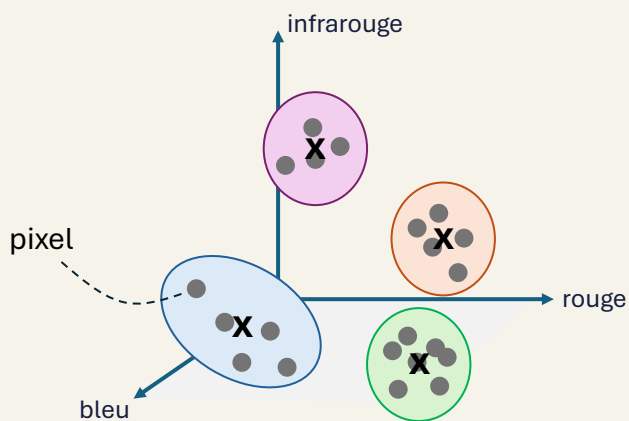
Cas 9 – Classification d'image

Classification non-supervisée



74

Classification non-supervisée



couches

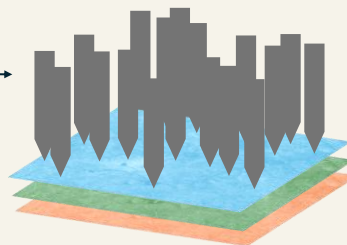
classification
non-supervisée

75

Classification non-supervisée

Sélection aléatoire de 1000 pixels sur l'image

```
var echantillon2 = landsat.sample({
  region: landsat.geometry(),
  scale: 30,
  numPixels: 1000,
  tileSize: 8
});
```

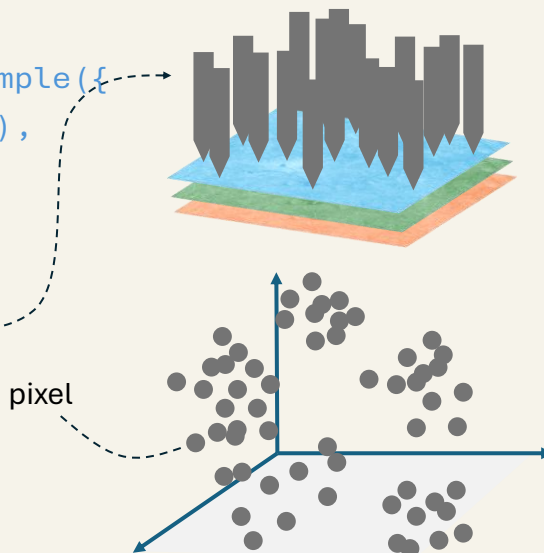


76

Classification non-supervisée

Sélection aléatoire de 1000 pixels sur l'image

```
var echantillon2 = landsat.sample({
  region: landsat.geometry(),
  scale: 30,
  numPixels: 1000,
  tileScale: 8
});
```



77

Classification non-supervisée

nombre de clusters (= classes)

```
var clusterer = ee.Clusterer.wekaKMeans(5).train(echantillon2);
```

Classification à k-moyennes

ee.Clusterer.

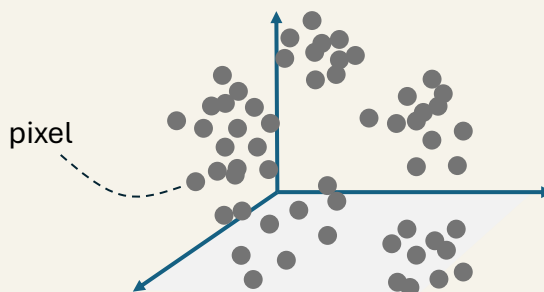
wekaKMeans

[Send feedback](#)

Cluster data using the k-means algorithm. Can use either the Euclidean distance (default) or the Manhattan distance. If the Manhattan distance is used, then centroids are computed as the component-wise median rather than mean. For more information see:

D. Arthur, S. Vassilvitskii: k-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 1027-1035, 2007.

Usage	Returns
<code>ee.Clusterer.wekaKMeans(nClusters, init, canopies, maxCandidates, periodicPruning, minDensity, t1, t2, distanceFunction, maxIterations, preserveOrder, fast, seed)</code>	Clusterer

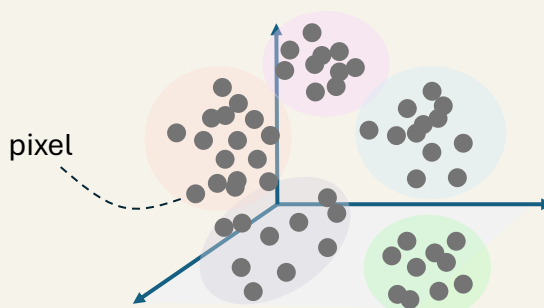


78

Classification non-supervisée

nombre de clusters (= classes)

```
var clusterer = ee.Clusterer.wekaKMeans(5).train(echantillon2);
```

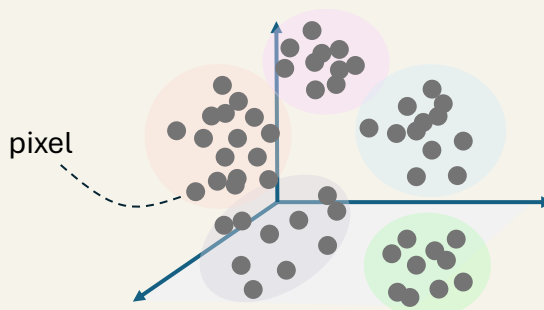


79

Classification non-supervisée

```
var clusterer = ee.Clusterer.wekaKMeans(5).train(echantillon2);
```

```
var Kclassifie = landsat.cluster(clusterer);
```



80

Classification non-supervisée

```
var clusterer = ee.Clusterer.wekaKMeans(5).train(echantillon2);  
  
var Kclassifie = landsat.cluster(clusterer);  
  
Map.addLayer(Kclassifie.randomVisualizer(), {},  
  'K-means classifié en 5 classes');
```

81

Classification non-supervisée

Questions :

Est-ce que les cartes produites sans et avec supervision sont comparables ?

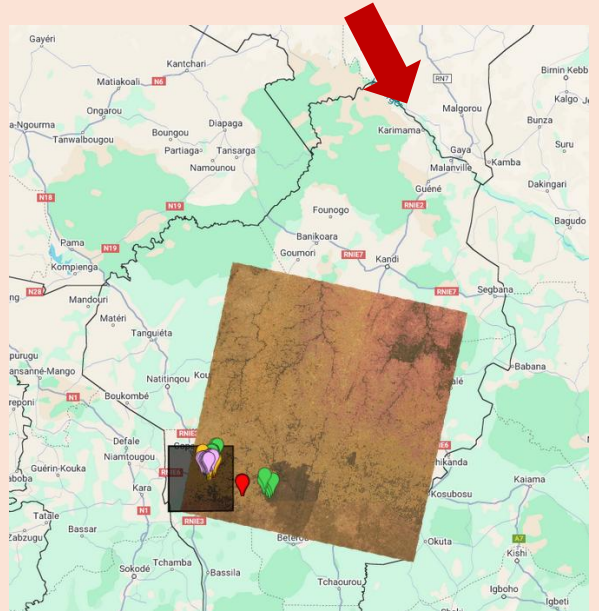
Visualiser ce qu'entraîne + ou – de groupes dans l'algorithme k-moyennes

82

Classification non-supervisée

Exercice :

Répéter le même type de classification supervisée dans la région de Karimama-Malanville



83

Conclusions

```

FeatureCollection.flatten()
Image.sampleRegions()
ee.Classifier.smileCart()
.train()
Image.classify()
FeatureCollection.randomColumn()
Image.sample()
ee.Clusterer.wekaKMeans()
Image.cluster()

```

84

Conclusions : Machine Learning

Nouvelles méthodes en développement (DeepMind + GEE) : **Embedding Fields**



<https://developers.google.com/machine-learning/crash-course/embeddings>

Emily Schechter & Valerie Pasquarella – Geo for Good 2024

85

Jointures & Jointures spatiales

- deux ImageCollection (a , b) ou FeatureCollection (a , b)
- une condition

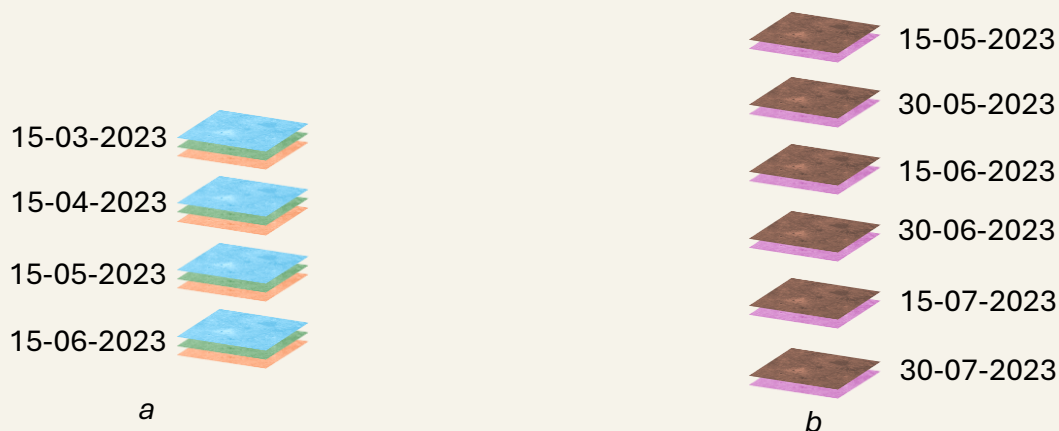
86

Utiliser Google Earth Engine

Cas 10 – Joint

Jointures & Jointures spatiales

- deux ImageCollection (a , b) ou FeatureCollection (a , b)
- une condition



87

Utiliser Google Earth Engine

Cas 10 – Joint

Jointures & Jointures spatiales

- deux ImageCollection (a , b) ou FeatureCollection (a , b)
- une condition

Produit une ImageCollection ou FeatureCollection contenant les éléments de a qui se trouvent aussi dans b .



88

Jointures & Jointures spatiales

- deux ImageCollection (a , b) ou FeatureCollection (a , b)
- une condition

Produit une ImageCollection ou FeatureCollection contenant les éléments de a qui se trouvent aussi dans b .

```
ee.Joint.simple().apply(a, b, filtre)
```

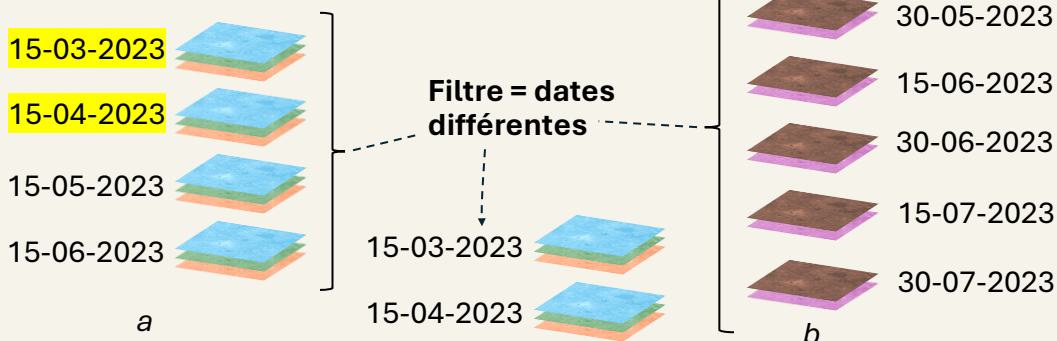


89

Jointures & Jointures spatiales

- deux ImageCollection (a , b) ou FeatureCollection (a , b)
- une condition

```
ee.Joint.inverted().apply(a, b, filtre)
```



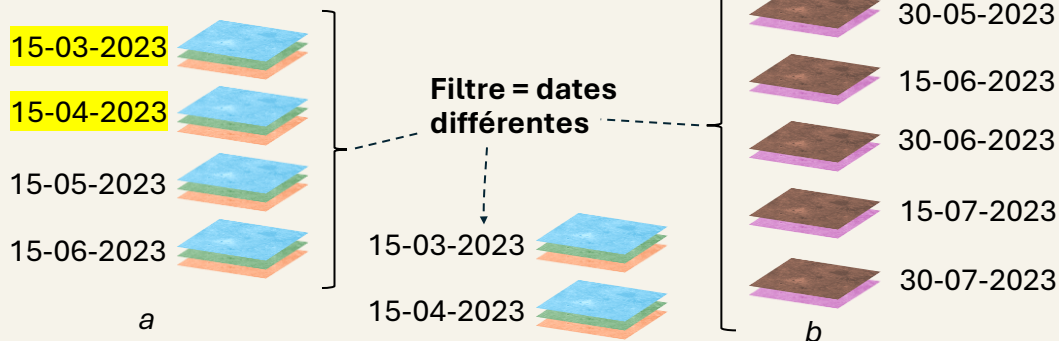
90

Jointures & Jointures spatiales

- deux ImageCollection (*a*, *b*) ou FeatureCollection (*a*, *b*)
- une condition

L'ordre dans apply() : **collection primaire**, puis **collection secondaire**

```
ee.Joint.inverted().apply(a, b, filtre)
```



91

```
var MODISEVI = ee.ImageCollection('MODIS/MCD43A4_006_EVI')
    .filter(ee.Filter.date('2014-01-01', '2014-02-01'));
var MODIS = ee.ImageCollection('MODIS/006/MCD43A2')
    .filter(ee.Filter.date('2014-01-27', '2014-02-15'));
```

92

```

var MODISEVI = ee.ImageCollection('MODIS/MCD43A4_006_EVI')
    .filter(ee.Filter.date('2014-01-01', '2014-02-01'));
var MODIS = ee.ImageCollection('MODIS/006/MCD43A2')
    .filter(ee.Filter.date('2014-01-27', '2014-02-15'));
var filtre = ee.Filter.equals({
  leftField: 'system:index',
  rightField: 'system:index'
});

```

Information sur la date dans GEE,
stockée dans les propriétés de
chaque Image

93

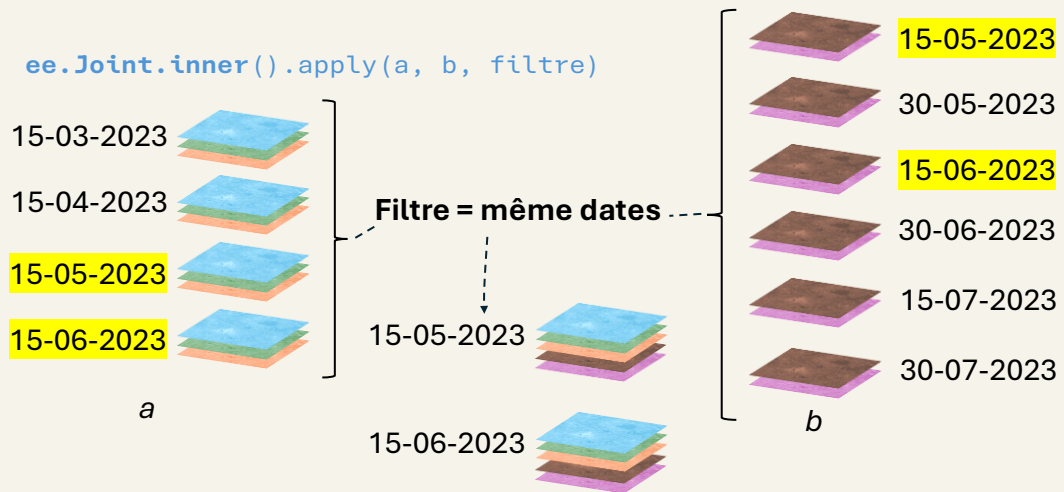
```

var MODISEVI = ee.ImageCollection('MODIS/MCD43A4_006_EVI')
    .filter(ee.Filter.date('2014-01-01', '2014-02-01'));
var MODIS = ee.ImageCollection('MODIS/006/MCD43A2')
    .filter(ee.Filter.date('2014-01-27', '2014-02-15'));
var filtre = ee.Filter.equals({
  leftField: 'system:index',
  rightField: 'system:index'
});
var joinsimple = ee.Join.simple().apply(MODISEVI, MODIS,
filtre);
print(joinsimple);

```

94

“Inner Joint”



95

```
var joininner = ee.Join.inner().apply(MODISEVI, MODIS, filtre);
print(joininner);

var joininnerMODIS = joininner.map(function(feature) {
  return ee.Image.cat(feature.get('primary'),
    feature.get('secondary'));
});
print(joininnerMODIS);
```

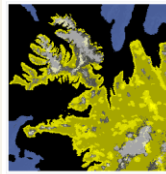
96

Utiliser Google Earth Engine

Cas 10 – Joint

“Inner Joint” : pratique quand des ImageCollection ou FeatureCollection sont complémentaires.

MCD43A2.061 MODIS BRDF-Albedo Quality Daily 500m



Dataset Availability
2000-02-24T00:00:00Z-
2025-01-14T00:00:00Z

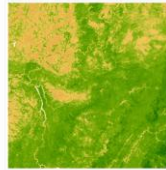
Dataset Provider
NASA LP DAAC at the
USGS EROS Center

Earth Engine Snippet
ee.ImageCollection("M
ODIS/061/MCD43A2")

Cadence
1 Day

Tags

albedo brdf daily global modis
nasa reflectance usgs quality

MODIS Combined 16-Day EVI 

Dataset Availability
2000-02-24T00:00:00Z-
2023-02-10T00:00:00Z

Dataset Provider
Google

Earth Engine Snippet
ee.ImageCollection("M
ODIS/
MCD43A4_006_EVI")

Cadence
1 Day

Tags

albedo brdf daily evi global
mcd43a4 modis nasa reflectance
usgs

97

Utiliser Google Earth Engine

Cas 10 – Joint

Jointures & Jointures spatiales

Différence : ici le filtre se base sur les caractéristiques spatiales

98

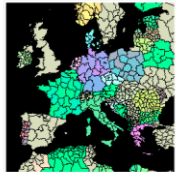
Utiliser Google Earth Engine

Cas 10 – Joint

Jointures & Jointures spatiales

Différence : ici le filtre se base sur les caractéristiques spatiales

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



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

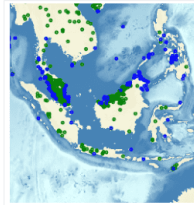
Dataset Provider
FAO UN

Earth Engine Snippet
FeatureCollection
ee.FeatureCollection("FAO/
GAUL/2015/level1")

Tags

borders departments fao
gaul provinces states table
un

WDPA: World Database on Protected Areas (points)



Dataset Availability

2017-07-01T00:00:00Z–
2030-01-01T00:00:00Z

Dataset Provider

UN Environment World Conservation
Monitoring Centre (UNEP-WCMC) /
Protected Planet

Earth Engine Snippet

FeatureCollection

Tags

boundaries iucn
mpa protected
wcmc wdpa



→ Le nombre d'aires protégées dans chaque département du Bénin

99

Utiliser Google Earth Engine

Cas 10 – Joint

Jointures & Jointures spatiales

Différence : ici le filtre se base sur les caractéristiques spatiales

```
var WDPApoints = ee.FeatureCollection('WCMC/WDPA/current/points');
var Benin = ee.FeatureCollection('FAO/GAUL/2015/level1')
.filter(ee.Filter.eq('ADM0_NAME', 'Benin'));
```

```
Map.addLayer(Benin);
Map.centerObject(Benin, 7);
```

100

Jointures & Jointures spatiales

Différence : ici le filtre se base sur les caractéristiques spatiales

```
var filtre = ee.Filter.intersects({
  leftField: '.geo',
  rightField: '.geo',
  maxError: 10
});
```

Indique qu'on utilise la géométrie des vecteurs pour filtrer ceux qui se touchent

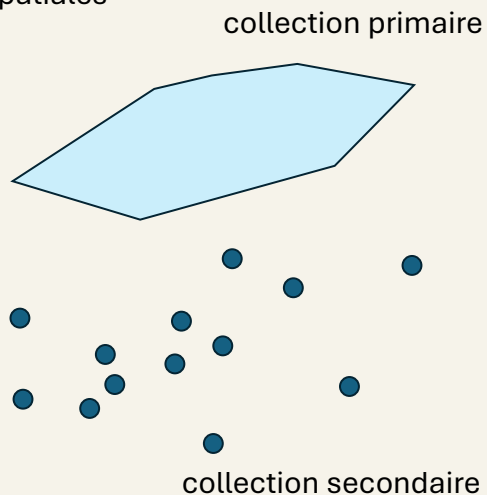
101

Jointures & Jointures spatiales

Différence : le filtre se base sur les caractéristiques spatiales

```
var saveAllJoin = ee.Join.saveAll({
  matchesKey: 'pointsWDPA',
});
```

envoie dans la collection primaire toutes les Feature de la collection secondaire qui touchent la Feature de la collection primaire sous le nom 'pointsWDPA'



102

Utiliser Google Earth Engine

Cas 10 – Joint

Jointures & Jointures spatiales

Différence : ici le filtre se base sur les caractéristiques spatiales

```
var WDPApointsauBenin = saveAllJoin.apply(Benin, WDPApoints, filtre);
print(WDPApointsauBenin);
```

103

Utiliser Google Earth Engine

Cas 10 – Joint

Jointures & Jointures spatiales

Différence : ici le filtre se base sur les caractéristiques spatiales

```
var WDPApointsauBenin = saveAllJoin.apply(Benin, WDPApoints, filtre);
print(WDPApointsauBenin);
```

```

▶6: Feature 00000000000000000996 (Polygon, 11 properties)
▶7: Feature 00000000000000000998 (MultiPolygon, 11 properties)
▼8: Feature 00000000000000000999 (MultiPolygon, 11 properties)
  type: Feature
  id: 00000000000000000999
  ▶geometry: MultiPolygon, 1391 vertices
  ▼properties: Object (11 properties)
    ADM0_CODE: 29
    ADM0_NAME: Benin
    ADM1_CODE: 618
    ADM1_NAME: Oueme
    DISP_AREA: NO
    EXPI_YEAR: 3000
    STATUS: Member State
    STR1_YEAR: 1999
    Shape_Area: 0.104274044486
    Shape_Leng: 2.17976755502
    ▼pointsWDPA: List (4 elements)
      ▶0: Feature 0000000000000000f8b (Point, 28 properties)
      ▶1: Feature 0000000000000000f8c (Point, 28 properties)
      ▶2: Feature 00000000000000002b45 (Point, 28 properties)
      ▼3: Feature 0000000000000000104d (Point, 28 properties)
        type: Feature
        id: 0000000000000000104d
        ▶geometry: Point (2.53, 6.65)
        ▼properties: Object (28 properties)
          CONS_OBJ: Not Applicable
          DESIG: Ramsar Site, Wetland of International Importance
          DESIG_ENG: Ramsar Site, Wetland of International Importance
          DESIG_TYPE: International
          GOV_TYPE: Not Reported
          INT_CRIT: (i)(ii)(iii)(iv)(vi)(vii)(viii)
          ISO3: BEN
          IUCN_CAT: Not Reported
          MANG_AUTH: Not Reported
          MANG_PLAN: Management plan is not implented and not available
          MARINE: 0
          METADATATD: 1856

```

104

Faire une figure

```
var WDPApointsauBenin_somme = WDPApointsauBenin.map(function(departement) {
  var npoints = ee.List(departement.get('pointsWDPA')).size();
  return departement.set('nb_points', npoints);
});
print(WDPApointsauBenin_somme);
```

105

Faire une figure

```
var figure = ui.Chart.feature.byFeature(WDPApointsauBenin_somme,
  'ADM1_NAME', 'nb_points')
print(figure);
```

`ui.Chart.feature.byFeature(features, xProperty, yProperties)`

Argument	Type	Details
<code>features</code>	Feature FeatureCollection List	The features to include in the chart
<code>xProperty</code>	String, optional	The property used as the value of x
<code>yProperties</code>	List, optional	Property or properties used on the y-axis (except xProperty).

106

Faire une figure

```
var figure = ui.Chart.feature.byFeature(WDPApointsauBenin_somme,
  'ADM1_NAME', 'nb_points')
  .setChartType('ColumnChart')
  .setOptions({
    title: 'Aires protégées par département',
    hAxis: {title: 'Département'},
    vAxis: {title: 'Nombre'}});
print(figure);
```

Chart types.

```
itaTable).setChartType('ScatterChart'));
`able).setChartType('LineChart'));
:aTable).setChartType('ColumnChart'));
ible).setChartType('BarChart'));
ible).setChartType('PieChart'));
`able).setChartType('AreaChart'));
.setChartType('Table');
```

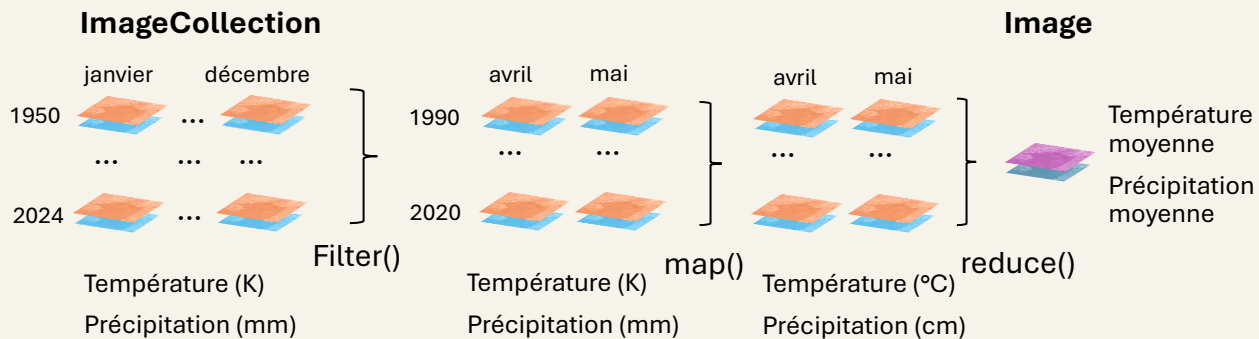
107

Jointures & Jointures spatiales : Conclusions

```
ee.Joint.simple()
  .apply()
ee.Joint.inverted()
ee.Filter.equals()
ee.Joint.inner()
ee.Filter.intersects()
ee.Join.saveAll()
list.size()
ui.Chart.feature.byFeature()
```

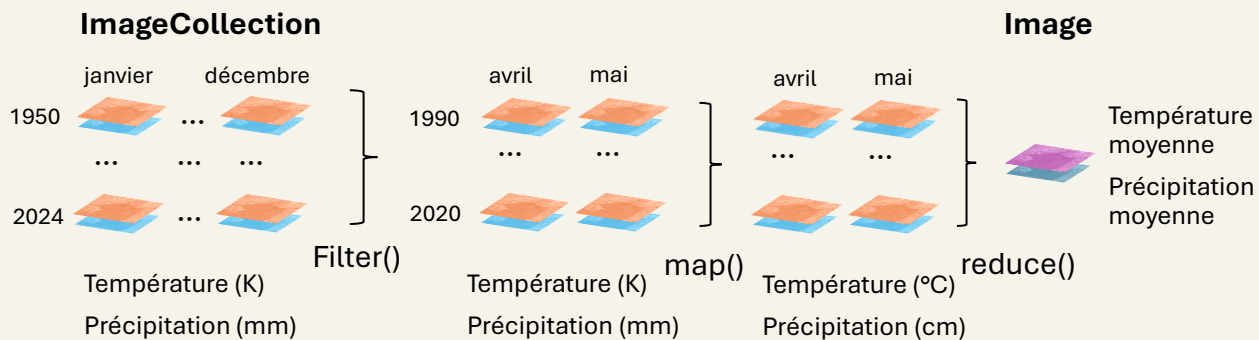
108

Rappels : Filter(), map(), et reduce()



109

Rappels : Filter(), map(), et reduce()



Filtrer (spatial et temporel) puis utiliser .map()

110

Utiliser Google Earth Engine

Cas 11 – Séries temporelles

Rappels : Filter(), map(), et reduce()

```
var parakoupoint = ee.Geometry.Point(2.6167, 9.338);
Map.addLayer(parakoupoint);
Map.centerObject(parakoupoint, 12);
```

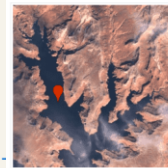
111

Utiliser Google Earth Engine

Cas 11 – Séries temporelle USGS Landsat 8 Collection 2 Tier 1 TOA Reflectance

Rappels : Filter(), map(), et reduce()

```
var parakoupoint = ee.Geometry.Point(2.6167, 9.338);
Map.addLayer(parakoupoint);
Map.centerObject(parakoupoint, 12);
```



Dataset Availability
2013-03-18T15:58:14Z–2025-01-23T14:35:38Z

Dataset Provider
USGS/Google

Earth Engine Snippet
ee.ImageCollection("LANDSAT/LC08/C02/T1_TOA")

Revisit Interval
16 Days

Tags
c2 global landsat

Description Bands Image Properties Terms of Use

Bands

Name	Pixel Size	Wavelength	Description
B1	30 meters	0.43 - 0.45 µm	Coastal aerosol
B2	30 meters	0.45 - 0.51 µm	Blue
B3	30 meters	0.53 - 0.59 µm	Green
B4	30 meters	0.64 - 0.67 µm	Red

```
var L8filtre = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA')
  .filterDate('2023-01-01', '2024-01-01')
  .filterBounds(parakoupoint)
  .select(['B4', 'B3', 'B2']);
```

112

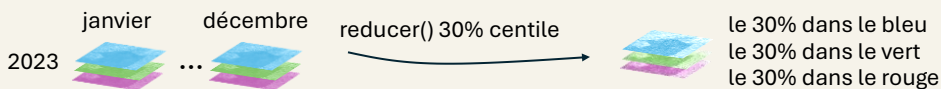
Rappels : Filter(), map(), et reduce()

```
var graph = ui.Chart.image.series({
  imageCollection: L8filtre,
  region: parakoupont,
  reducer: ee.Reducer.first(),
  scale: 30
});
print(graph);
```

113

Rappels : Filter(), map(), et reduce()

```
var p30 = L8filtre.reduce(ee.Reducer.percentile([30]))
print(p30);
```

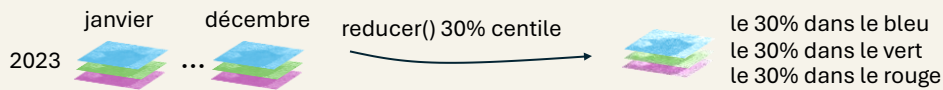


- ee.Reducer
- ee.Reducer.allNonZero()
- ee.Reducer.and()
- ee.Reducer.anyNonZero()
- ee.Reducer.autoHistogram(...)
- ee.Reducer.bitwiseAnd()
- ee.Reducer.bitwiseOr()
- ee.Reducer.centeredCovari...
- ee.Reducer.circularMean()
- ee.Reducer.circularStddev()
- ee.Reducer.circularVariance()
- ee.Reducer.count()
- ee.Reducer.countDistinct()
- ee.Reducer.countDistinctNo...
- ee.Reducer.countEvery()
- ee.Reducer.countRuns()
- ee.Reducer.covariance()
- ee.Reducer.first()
- ee.Reducer.firstNonNull()
- ee.Reducer.fixed2DHistogra...
- ee.Reducer.fixedHistogram(...)
- ee.Reducer.frequencyHistogr...
- ee.Reducer.geometricMedia...

114

Rappels : Filter(), map(), et reduce()

```
var p30 = L8filtre.reduce(ee.Reducer.percentile([30]));
print(p30);
```



```
Map.addLayer(p30, {
  min: 0.0,
  max: 0.35
}, 'Centile30%');
```

115

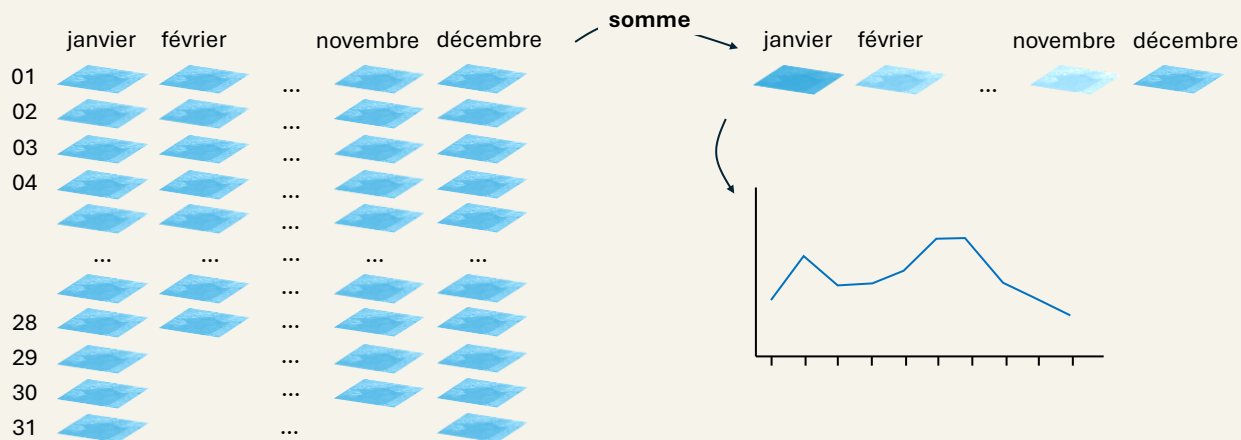
Rappels : Filter(), map(), et reduce()

```
var centiles = [0, 10, 20, 30, 40, 50, 60, 70, 80];
```

```
centiles.map(function(c) {
  var image = L8filtre.reduce(ee.Reducer.percentile([c]));
  Map.addLayer(image, {
    min: 0.05,
    max: 0.35
  }, c + '%');
});
```

116

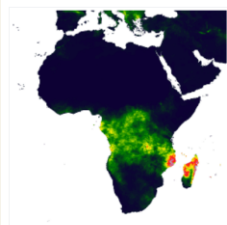
Aggrégation mensuelle à partir de données à plus fine resolution temporelle



117

Aggrégation mensuelle à partir de données à plus fine resolution temporelle

CHIRPS Pentad: Climate Hazards Center InfraRed Precipitation With Station Data (Version 2.0 Final)



Dataset Availability

1981-01-01T00:00:00Z-2024-12-26T00:00:00Z

Dataset Provider

UCSB/CHG

Earth Engine Snippet

```
ee.ImageCollection("UCSB-CHG/CHIRPS/PENTAD")
```

Cadence

1 Custom_time_unit

Tags

chg climate geophysical precipitation ucsb weather

Description **Bands** Image Properties Terms of Use Citations

Resolution

5566 meters

Bands

Name	Units	Min	Max	Description
precipitation	mm/pentad	0*	1072.43*	Precipitation

* estimated min or max value

118

Aggrégation mensuelle à partir de données à plus fine resolution temporelle

```
var chirps = ee.ImageCollection('UCSB-CHG/CHIRPS/PENTAD');  
var debut = '2023-01-01';  
var fin = '2024-01-01';  
var chirps2023 = chirps.filter(ee.Filter.date(debut, fin))  
print(chirps2023, 'chirps2023');
```

119

Aggrégation mensuelle à partir de données à plus fine resolution temporelle

GEE permet de manipuler des dates : ajout, retrait, filtrer certains mois...

120

Aggrégation mensuelle à partir de données à plus fine resolution temporelle

Manipuler des dates

```

var annee = 2023;
var debut = ee.Date.fromYMD(annee, 1, 1);
var fin = debut.advance(1, 'year');
var chirps2023_bis = chirps
    .filter(ee.Filter.date(debut, fin));
print(chirps2023_bis, 'chirps2023_bis');

```

YMD : Year Month Day pour le format entre parenthèses

A garder en anglais car il s'agit d'un paramètre de la fonction `.advance()`

121

Aggrégation mensuelle à partir de données à plus fine resolution temporelle

Manipuler des dates

La fonction `.millis()` convertit une date en millisecondes depuis 01/01/1970.

Les objets EarthEngine ont leurs propriétés `system:time_start` et `system:time_end` en format nombre et non date, ainsi `.millis()` permet de fournir une valeur de type nombre.

```

print(debut, 'date de debut');
print(fin, 'date de fin');

print('Start date as timestamp', debut.millis());
print('Start date as timestamp', fin.millis());

```

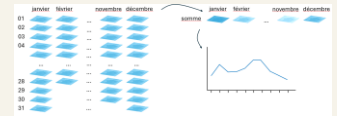
122

Agrégation mensuelle à partir de données à plus fine resolution temporelle

```

var mois_liste = ee.List.sequence(1, 12);
var total_mensuel_fun = function(mois) {
  var debut = ee.Date.fromYMD(annee, mois, 1);
  var fin = debut.advance(1, 'month');
  var mois_filtre = chirps2023_bis
    .filter(ee.Filter.date(debut, fin));
  var total = mois_filtre.reduce(ee.Reducer.sum());
  return total.set({
    'system:time_start': debut.millis(),
    'system:time_end': fin.millis(),
    'year': annee,
    'month': mois
  });
};

```



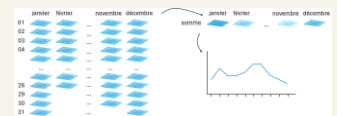
123

Agrégation mensuelle à partir de données à plus fine resolution temporelle

```

var images_mensuelles = mois_liste.map(total_mensuel_fun);
print(images_mensuelles);
var images_mensuelles_collection =
ee.ImageCollection.fromImages(images_mensuelles);
print(images_mensuelles_collection);

```



124

Agrégation mensuelle à partir de données à plus fine resolution temporelle

```

var images_mensuelles = mois_liste.map(total_mensuel);
print(images_mensuelles);

var images_mensuelles_collection =
ee.ImageCollection.fromImages(images_mensuelles);
print(images_mensuelles_collection);

```

```

ImageCollection (12 elements) JSC
  type: ImageCollection
  bands: []
  features: List (12 elements)
    ▶ 0: Image (1 band)
    ▶ 1: Image (1 band)
    ▶ 2: Image (1 band)
    ▶ 3: Image (1 band)
    ▶ 4: Image (1 band)
    ▶ 5: Image (1 band)
    ▶ 6: Image (1 band)
    ▶ 7: Image (1 band)
    ▶ 8: Image (1 band)
    ▶ 9: Image (1 band)
    ▶ 10: Image (1 band)
    ▼ 11: Image (1 band)
      type: Image
      bands: List (1 element)
        ▶ 0: "precipitation_sum", double, E...
      properties: Object (5 properties)
        month: 12
        system:index: 11
        system:time_end: 1704067200000
        system:time_start: 1701388800000
        year: 2023

```

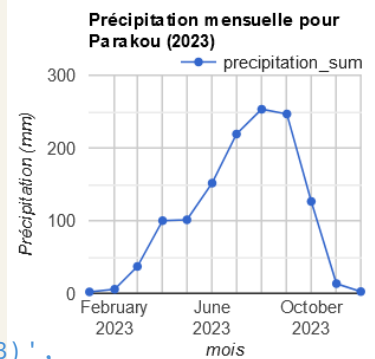
125

Agrégation mensuelle à partir de données à plus fine resolution temporelle

```

var graph = ui.Chart.image.series({
  imageCollection: images_mensuelles_collection,
  region: parakoupont,
  reducer: ee.Reducer.mean(),
  scale: 5566,
}).setOptions({
  lineWidth: 1,
  pointSize: 3,
  title: 'Précipitation mensuelle pour Parakou (2023)',
  vAxis: {title: 'Précipitation (mm)'},
  hAxis: {title: 'mois'}
});
print(graph);

```



126

Agrégation mensuelle à partir de données à plus fine résolution temporelle

Exercice 1 :

Répéter l'exercice en se focalisant sur la précipitation annuelle totale entre 1981 et 2024

127

Agrégation mensuelle à partir de données à plus fine résolution temporelle

Exercice 2 :

Répéter l'exercice avec :

- les données ERA5-Land Daily Aggregated,
ee.ImageCollection("ECMWF/ERA5_LAND/DAILY_AGGR")
(bande "total_precipitation_sum")
- à Kandi en 2021
- extraire un .csv avec la précipitation de chaque mois

128

Utiliser Google Earth Engine

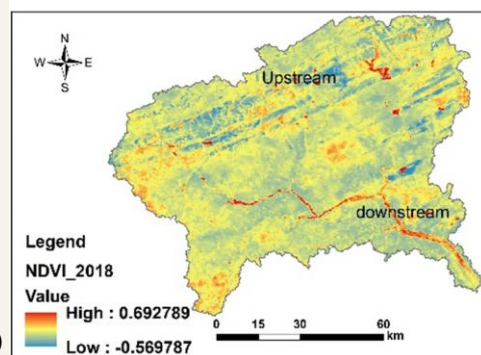
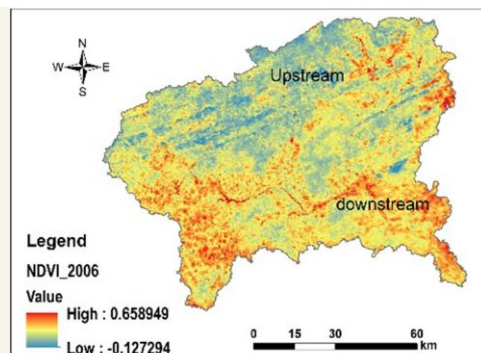
Cas 12 – Détection de changement

Détection de changement

Changement de reflectance : changement des caractéristiques de la surface

Changement de l'étendue de certaines classes d'occupation du sol

Sidi Almouctar (2021)



129

Utiliser Google Earth Engine

Cas 12 – Détection de changement

Détection de changement

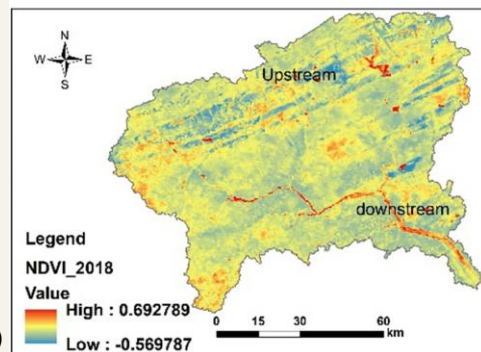
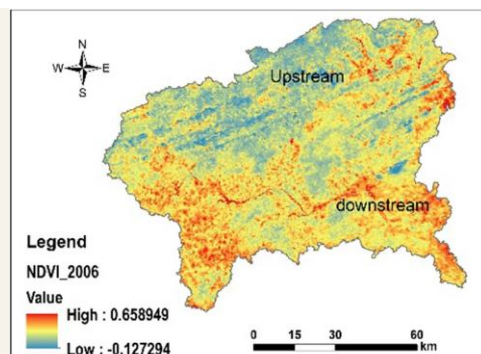
Changement de reflectance : changement des caractéristiques de la surface

Changement de l'étendue de certaines classes d'occupation du sol

Les séries temporelles permettent de suivre ou détecter les changements.

Cas le plus simple : deux images (pre- et post-).

Sidi Almouctar (2021)



130

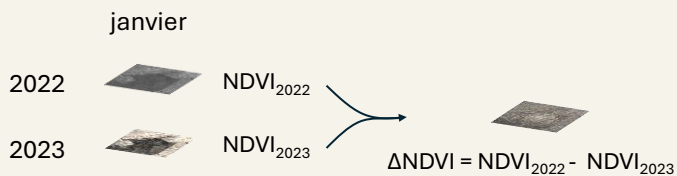
Utiliser Google Earth Engine

Cas 12 – Détection de changement

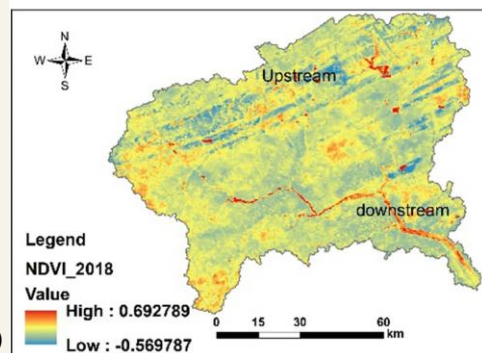
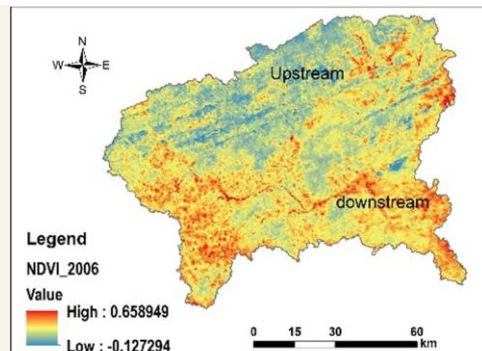
Utilisation d'indice lié aux caractéristiques de la surface suivie :

NBR, NDVI, NDWI...

Différence entre deux dates :



Sidi Almouctar (2021)



131

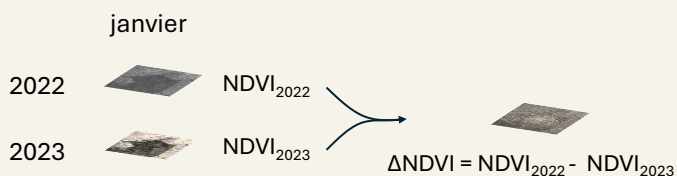
Utiliser Google Earth Engine

Cas 12 – Détection de changement

Utilisation d'indice lié aux caractéristiques de la surface suivie :

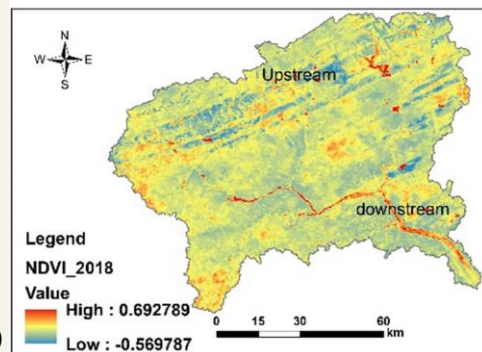
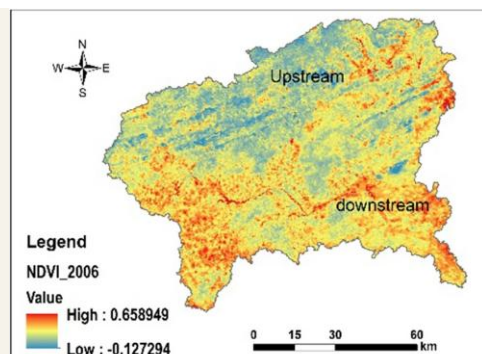
NBR, NDVI, NDWI...

Différence entre deux dates :



$\Delta\text{NDVI} < \text{seuil}$: perte de végétation ?

Sidi Almouctar (2021)



132

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```
var plantation =  
ee.FeatureCollection('WCMC/WDPA/current/polygons')  
  .filter(ee.Filter.eq('WDPA_PID', '33019'));  
Map.addLayer(plantation, {}, 'plantation');  
Map.centerObject(plantation, 14);
```

133

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```
var plantation =  
ee.FeatureCollection('WCMC/WDPA/current/polygons')  
  .filter(ee.Filter.eq('WDPA_PID', '33019'));  
Map.addLayer(plantation, {}, 'plantation');  
Map.centerObject(plantation, 14);  
  
var landsat8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')  
  .select(  
    ['SR_B2', 'SR_B3', 'SR_B4', 'SR_B5', 'SR_B6', 'SR_B7'],  
    ['blue', 'green', 'red', 'nir', 'swir1', 'swir2']  
  );
```

134

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```

var plantation =
ee.FeatureCollection('WCMC/WDPA/curr
  .filter(ee.Filter.eq('WDPA_PID', '
Map.addLayer(plantation, {}, 'planta
Map.centerObject(plantation, 14);

```

```

var landsat8 = ee.ImageCollection('L
  .select(
    ['SR_B2', 'SR_B3', 'SR_B4', 'SR_B5', 'SR_B6', 'SR_B7'],
    ['blue', 'green', 'red', 'nir', 'swir1', 'swir2']
  );

```

Usage		Returns
<code>ImageCollection.select(selectors, names)</code>		ImageCollection
Argument	Type	Details
<code>this: imagecollection</code>	ImageCollection	The ImageCollection instance.
<code>selectors</code>	List	A list of names, regexes or numeric indices specifying the bands to select.
<code>names</code>	List, optional	A list of new names for the output bands. Must match the number of bands selected.

Nouveaux noms

135

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```

var L82014 = landsat8
  .filterBounds(plantation)
  .filterDate('2014-02-01', '2014-03-01')
  .sort('CLOUD_COVER', true)
  .first();

var L82023 = landsat8
  .filterBounds(plantation)
  .filterDate('2023-02-01', '2023-03-01')
  .sort('CLOUD_COVER', true)
  .first();

```

136

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```
var visParam = {  
  'bands': ['swir2', 'nir', 'red'],  
  'min': 7750,  
  'max': 22200  
};  
Map.addLayer(L82014, visParam, '2014');  
Map.addLayer(L82023, visParam, '2023');
```

137

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```
var NDVI2014 = L82014.normalizedDifference(['nir', 'red'])  
  .rename('ndvi_2014');  
var NDVI2023 = L82023.normalizedDifference(['nir', 'red'])  
  .rename('ndvi_2023');  
  
var deltaNDVI = NDVI2023.subtract(NDVI2014)  
  .rename('deltaNDVI');
```

138

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```

var NDVI2014 = L82014.normalizedDifference(['nir', 'red'])
  .rename('ndvi_2014');
var NDVI2023 = L82023.normalizedDifference(['nir', 'red'])
  .rename('ndvi_2023');

var deltaNDVI = NDVI2023.subtract(NDVI2014)
  .rename('deltaNDVI');

var NDVIvis = {
  palette: ['red', 'beige', 'green'],
  min: -0.2,
  max: 0.2
};
Map.addLayer(deltaNDVI, NDVIvis, 'Changement en NDVI');

```

139

Utiliser Google Earth Engine

Cas 12 – Détection de changement

```

var seuilgain = 0.05;
var seuilperte = -0.05;

var changementclassifie = ee.Image(0);

var changementclassifie =
changementclassifie.where(deltaNDVI.lte(seuilperte), 2);

var changementclassifie =
changementclassifie.where(deltaNDVI.gte(seuilgain), 1);

```

Nouvelle Image vide (tous les pixels = 0)

L'image vide prend la valeur 2 où deltaNDVI est $\leq -0,05$

140

```

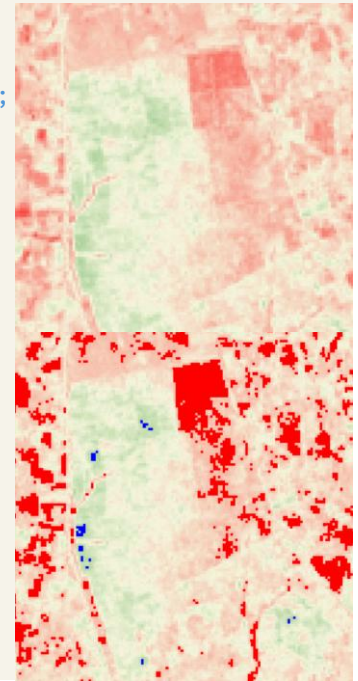
var changementclassifie =
changementclassifie.where(deltaNDVI.lte(seuilperte), 2);
var changementclassifie =
changementclassifie.where(deltaNDVI.gte(seuilgain), 1);

var changeVis = {
  palette: 'blue, red',
  min: 1,
  max: 2
};

Map.addLayer(changementclassifie.selfMask(),
  changeVis, 'changement lié aux seuils');

```

Masque tous les pixels = 0



141

```

var changementclassifie = ee.Image(0);

var changementclassifie =
changementclassifie.where(deltaNDVI.lte(seuilperte), 2);

var changementclassifie =
changementclassifie.where(deltaNDVI.gte(seuilgain), 1);

```

Équivaut à

```

var methodealternative = deltaNDVI.gte(seuilgain)
.where(deltaNDVI.lte(seuilperte), 2);

Map.addLayer(methodealternative.selfMask(),
  changeVis, 'changement lié aux seuils');

```

142

Détection de changement : Conclusions

```
.where()  
.selfMask()
```

143

Régressions

Déterminer la relation entre une variable dépendante ~ variables indépendante(s)

Par exemple : prédire la biomasse/biodiversité/taille en se basant sur des indices de végétation.

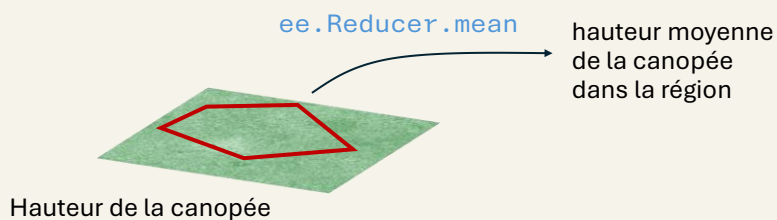
Beaucoup de types de regression

144

Régressions

Rappel :

Reducer dans reduceRegion

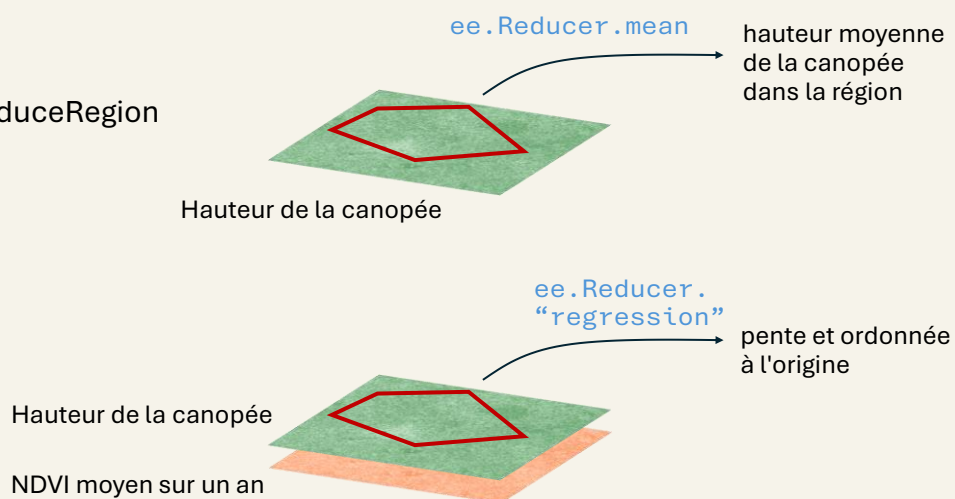


145

Régressions

Rappel :

Reducer dans reduceRegion



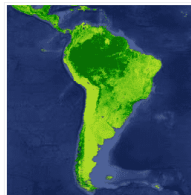
146

Utiliser Google Earth Engine

Cas 13 – Régressions

Produire un polygone autour de la forêt de N'Dali

```
Map.centerObject(polygone, 9);
var mod44b = ee.ImageCollection('MODIS/006/MOD44B');
```

MOD44B.006 Terra Vegetation Continuous Fields Yearly Global 250m □ ▾**Disponibilité des ensembles de données**

2000-03-05T00:00:00Z – 2020-03-05T00:00:00Z

Fournisseur d'ensembles de données
[DAAC LP de la NASA au centre EROS de l'USGS](#)
Extrait Earth Engine

```
ee.ImageCollection("MODIS/006/MOD44B")
```

Cadence

1 an

Tags
 annuel géophysique global modis nasa terra
 couverture arborée végétation mod44b

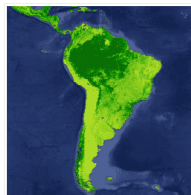
147

Utiliser Google Earth Engine

Cas 13 – Régressions

Produire un polygone autour de la forêt de N'Dali

```
Map.centerObject(polygone, 9);
var mod44b_PTC = ee.ImageCollection('MODIS/006/MOD44B')
  .filterDate('2019-01-01', '2020-01-01')
  .first()
  .clip(polygone)
  .select('Percent_Tree_Cover');
```

MOD44B.006 Terra Vegetation Continuous Fields Yearly Global 250m □ ▾**Disponibilité des ensembles de données**

2000-03-05T00:00:00Z – 2020-03-05T00:00:00Z

Fournisseur d'ensembles de données
[DAAC LP de la NASA au centre EROS de l'USGS](#)
Extrait Earth Engine

```
ee.ImageCollection("MODIS/006/MOD44B")
```

Cadence

1 an

Tags
 annuel géophysique global modis nasa terra
 couverture arborée végétation mod44b

148

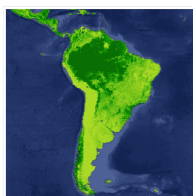
Utiliser Google Earth Engine

Cas 13 – Régressions

Produire un polygone autour de la forêt de N'Dali

```
print('Couverture 2020', mod44b_PTC);
```

```
Map.addLayer(mod44b_PTC, {max: 100}, 'Couverture 2020');
```

MOD44B.006 Terra Vegetation Continuous Fields Yearly Global 250m **Disponibilité des ensembles de données**

2000-03-05T00:00:00Z – 2020-03-05T00:00:00Z

Fournisseur d'ensembles de données
[DAAC LP de la NASA au centre EROS de l'USGS](#)
Extrait Earth Engine

```
ee.ImageCollection("MODIS/06/NO244B")
```

Cadence

1 an

Tags
 annuel géophysique global modis nasa terra
 couverture arborée végétation mod44b

149

Utiliser Google Earth Engine

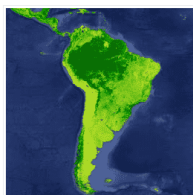
Cas 13 – Régressions

Produire un polygone autour de la forêt de N'Dali

```
print('Couverture 2020', mod44b_PTC);
```

```
Map.addLayer(mod44b_PTC, {max: 100}, 'Couverture 2020');
```

variable dépendante ←

MOD44B.006 Terra Vegetation Continuous Fields Yearly Global 250m **Disponibilité des ensembles de données**

2000-03-05T00:00:00Z – 2020-03-05T00:00:00Z

Fournisseur d'ensembles de données
[DAAC LP de la NASA au centre EROS de l'USGS](#)
Extrait Earth Engine

```
ee.ImageCollection("MODIS/06/NO244B")
```

Cadence

1 an

Tags
 annuel géophysique global modis nasa terra
 couverture arborée végétation mod44b

150

Ajout de la variable indépendante

```

var L8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_RT')
    .filterBounds(polygone)
    .filterDate('2019-02-01', '2019-03-01')
    .first();

var visL8 = {
  bands: ['B4', 'B3', 'B2'],
  max: 16000
};
Map.addLayer(L8, visL8, 'Image L8');

```

151

Ajout de la variable indépendante

```

var ndvi = L8.normalizedDifference(['B5',
  'B4']).rename('ndvi');

var ImageEntrainement = ndvi.addBands(mod44b_PTC);
print('Image dentrainement', ImageEntrainement);

```



Dans GEE, les bandes des variables indépendantes doivent être présentées d'abord, puis viennent celles des variables dépendantes

152

Régression lineaire

```
var linearFit = ImageEntrainement.reduceRegion({  
  reducer: ee.Reducer.linearFit(),  
  geometry: polygone,  
  scale: 30,  
  bestEffort: true  
});
```

153

Régression lineaire

```
var linearFit = ImageEntrainement.reduceRegion({  
  reducer: ee.Reducer.linearFit(),  
  geometry: polygone,  
  scale: 30,  
  bestEffort: true  
});  
  
print('OLS estimates:', linearFit);  
print('y-intercept:', linearFit.get('offset'));  
print('Pente:', linearFit.get('scale'));
```

154

Application des résultats à la region pour prédire la couverture végétale

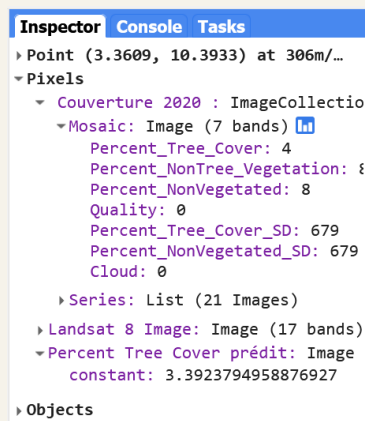
```
var predictionPTC = ndvi.expression(
  'intercept + slope * ndvi', {
    'ndvi': ndvi.select('ndvi'),
    'intercept': ee.Number(linearFit.get('offset')),
    'slope': ee.Number(linearFit.get('scale'))
  });
print('predictionPTC', predictionPTC);

Map.addLayer(predictionPTC, {max: 100}, 'Percent Tree Cover
prédit');
```

155

Application des résultats à la region pour prédire la couverture végétale

Utiliser l'inspecteur pour comparer les valeurs observées et prédites



156

Régression linéaire multiple

Au lieu d'utiliser le NDVI comme variable indépendante : utiliser les bandes de Landsat 8

```
var IndependantesBandes = ['B1', 'B2', 'B3', 'B4', 'B5',  
'B6', 'B7', 'B10', 'B11'];
```

157

Régression linéaire multiple

Au lieu d'utiliser le NDVI comme variables indépendantes : utiliser les bandes de Landsat 8

```
var IndependantesBandes = ['B1', 'B2', 'B3', 'B4', 'B5',  
'B6', 'B7', 'B10', 'B11'];
```

```
var ImageEntrainement = ee.Image(1)  
  .addBands(L8.select(IndependantesBandes))  
  .addBands(mod44b_PTC);
```

158

Utiliser Google Earth Engine

Cas 13 – Régressions

Régression linéaire multiple

```
print(ImageEntrainement);
```

159

Utiliser Google Earth Engine

Cas 13 – Régressions

Régression linéaire multiple

```
print(ImageEntrainement);
```

```
var RegressionLineaire = ImageEntrainement.reduceRegion({  
  reducer: ee.Reducer.linearRegression({  
    numX: 10,  
    numY: 1  
  }),  
  geometry: polygone,  
  scale: 30,  
  bestEffort: true  
});
```

Il ne s'agit plus de `linearFit()`

160

Utiliser Google Earth Engine

Cas 13 – Régressions

Régression linéaire multiple

```
print(ImageEntrainement);

var RegressionLineaire = ImageEntrainement.reduceRegion({
  reducer: ee.Reducer.linearRegression({
    numX: 10,-----nombre de variables indépendantes
    numY: 1-----nombre de variables dépendantes
  }),
  geometry: polygone,
  scale: 30,
  bestEffort: true
});
```

161

Utiliser Google Earth Engine

Cas 13 – Régressions

Régression linéaire multiple

```
print(RegressionLineaire, 'Résultat RegressionLineaire');
```

162

Régression linéaire multiple

```
print(RegressionLineaire, 'Résultat RegressionLineaire');

var coeff = ee.Array(RegressionLineaire.get('coefficients'))
  .project([0])
  .toList();
```

163

Application des résultats à la region pour prédire la couverture végétale

```
var predictionRLM = ee.Image(1)
  .addBands(L8.select(IndependantesBandes))
  .multiply(ee.Image.constant(coeff))
  .reduce(ee.Reducer.sum())
  .rename('predictedTreeLR')
  .clip(L8.geometry());
```

} Application des coefficients pour chaque bande et valeur de l'intercept

164

Application des résultats à la region pour prédire la couverture végétale

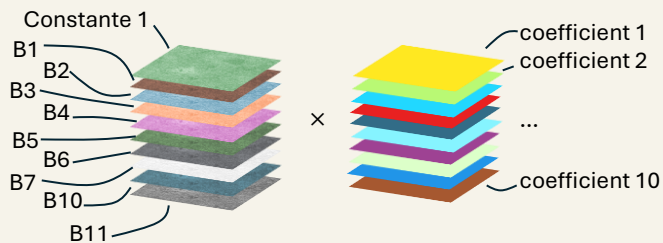
```

var predictionRLM = ee.Image(1)
  .addBands(L8.select(IndependantesBandes))
  .multiply(ee.Image.constant(coeff))
  .reduce(ee.Reducer.sum())
  .rename('predictedTreeLR')
  .clip(L8.geometry());

```

Application des coefficients
pour chaque bande et valeur
de l'intercept

$$PTC_{prédit} = 1 \times coeff1 + B01 \times coeff2 + B02 \times coeff3 \dots$$



165

Application des résultats à la region pour prédire la couverture végétale

```

var predictionRLM = ee.Image(1)
  .addBands(L8.select(IndependantesBandes))
  .multiply(ee.Image.constant(coeff))
  .reduce(ee.Reducer.sum())
  .rename('predictedTreeLR')
  .clip(L8.geometry());

```

Application des coefficients
pour chaque bande et valeur
de l'intercept

```

Map.addLayer(predictionRLM, {
  min: 0,
  max: 10
}, 'Percent Tree Cover prédit par RLM');

```

166

Utiliser Google Earth Engine

Cas 13 – Régressions

Régressions non-linéaires

Régressions linéaires : implémentées dans les reducers

Régressions non-linéaires : implémentées dans les classifieur

Par exemple *classification and regression tree* (CART)

167

Utiliser Google Earth Engine

Cas 13 – Régressions

Régressions non-linéaires : CART

Les données doivent être en format FeatureCollection

```
var CARTEntrainementImage = ee.Image(L8.select(IndependantesBandes))  
  .addBands(mod44b_PTC);
```

```
var CARTEntrainement = CARTEntrainementImage.sample({  
  region: polygone,  
  scale: 30,  
  numPixels: 1500,  
  seed: 5  
});  
print('CART Entrainement', CARTEntrainement);
```

168

Utiliser Google Earth Engine

Cas 13 – Régressions

Régressions non-linéaires : CART

Les données doivent être en format FeatureCollection

```
var cartRegression = ee.Classifier.smileCart()  
  .setOutputMode('REGRESSION')  
  .train({  
    features: CARTEntrainement,  
    classProperty: 'Percent_Tree_Cover',  
    inputProperties: IndependantesBandes  
  });
```

169

Utiliser Google Earth Engine

Cas 13 – Régressions

Régressions non-linéaires : CART

Les données doivent être en format FeatureCollection

```
var cartRegressionImage = L8.select(IndependantesBandes)  
  .classify(cartRegression, 'cartRegression');  
  
Map.addLayer(cartRegressionImage, {  
  min: 0,  
  max: 10  
}, 'Percent Tree Cover CART');
```

170

Evaluation des régressions

RMSE : comparaison des valeurs observée et des valeurs prédites

les différentes régressions

MOD44B

$$RMSE = \sqrt{\frac{\sum (Prédite_i - Observée_i)^2}{n}}$$

171

Evaluation des régressions

```
var obspred = ee.Image.cat(mod44b_PTC, predictionPTC,
  predictionRLM, cartRegressionImage)
  .rename(['TCpercent', 'LFprediction',
    'RLMprediction', 'CARTprediction'
  ]);
var echantillonRMSE = obspred.sample({
  region: polygone,
  scale: 30,
  numPixels: 500,
  seed: 5
});
```

172

Evaluation des régressions

```
print('Premier échantillon', echantillonRMSE.first());
```

173

Evaluation des régressions

```
print('Premier échantillon', echantillonRMSE.first());
```

Exercice : Créer une fonction pour calculer la partie surlignée pour chacun des trois modèles

$$RMSE = \sqrt{\frac{\sum (Prédite_i - Observée_i)^2}{n}}$$

Info :

la fonction `feature.get('p')` permet de prendre la valeur de la propriété p
`a.pow(2)` fait le carré de a

174

Evaluation des régressions

```
var calculDiff = function(feature) {  
  var TCpercent = ee.Number(feature.get('TCpercent'));  
  
  var diffLF2 = ee.Number(feature.get('LFprediction'))  
    .subtract(TCpercent).pow(2);  
  var diffRLM2 = ee.Number(feature.get('RLMprediction'))  
    .subtract(TCpercent).pow(2);  
  var diffCART2 = ee.Number(feature.get('CARTprediction'))  
    .subtract(TCpercent).pow(2);  
  
  return feature.set('diffLF2', diffLF2).set('diffRLM2', diffRLM2)  
    .set('diffCART2', diffCART2)  
};
```

175

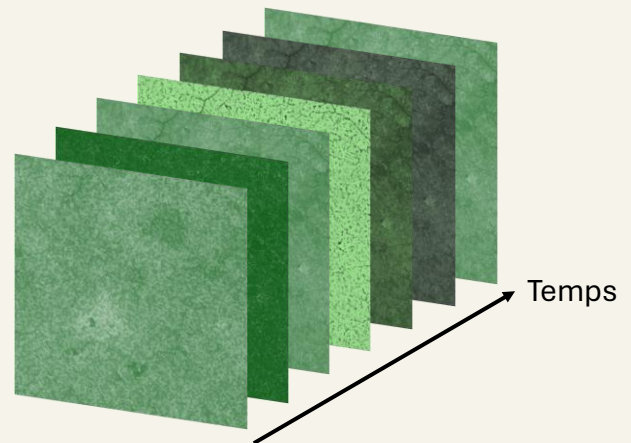
Evaluation des régressions

```
print(echantillonRMSE.map(calculDiff));  
  
var RMSECART =  
echantillonRMSE.map(calculDiff).aggregate_mean('diffCART2').sqrt();  
print(RMSECART, 'RMSECART');
```

176

Tendance

- Séries temporelles souvent sous la forme d'ImageCollection
- Le nombre d'observation par pixel varie entre les pixels (nuages, ...)



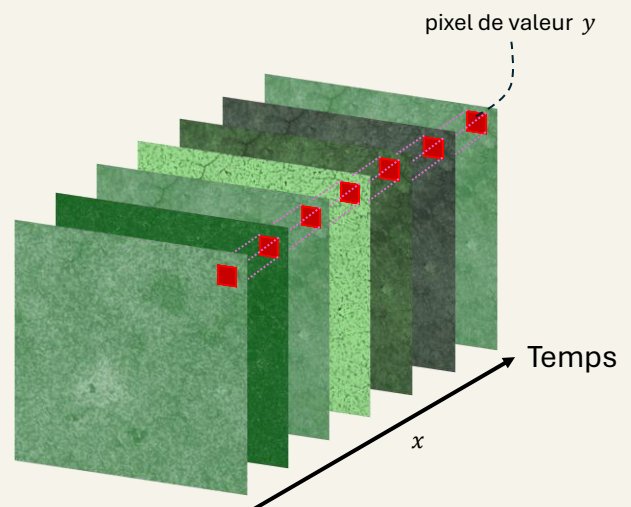
177

Tendance

- Séries temporelles souvent sous la forme d'ImageCollection
- Le nombre d'observation par pixel varie entre les pixels (nuages, ...)

Ajustement de courbe par pixel (pour une regression linéaire, idem *regress* dans terra sur R)

$$y = ax + b$$



178

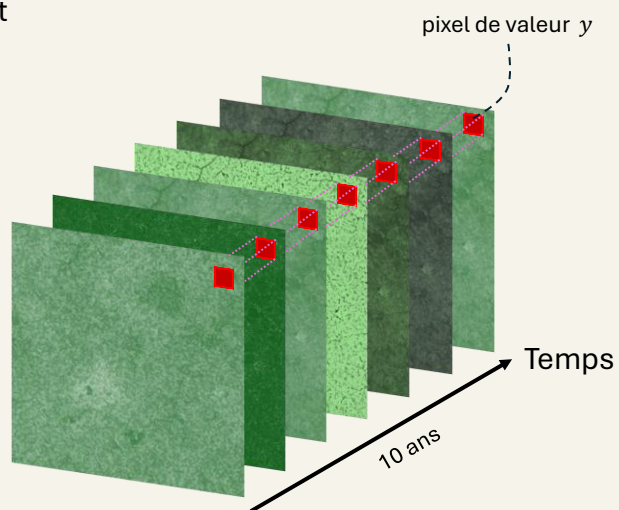
Utiliser Google Earth Engine

Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

$$NDVI = a \times date + b$$

Valeur continue comme
jours ou millisecons
(`.millis()`) depuis
1970



179

Utiliser Google Earth Engine

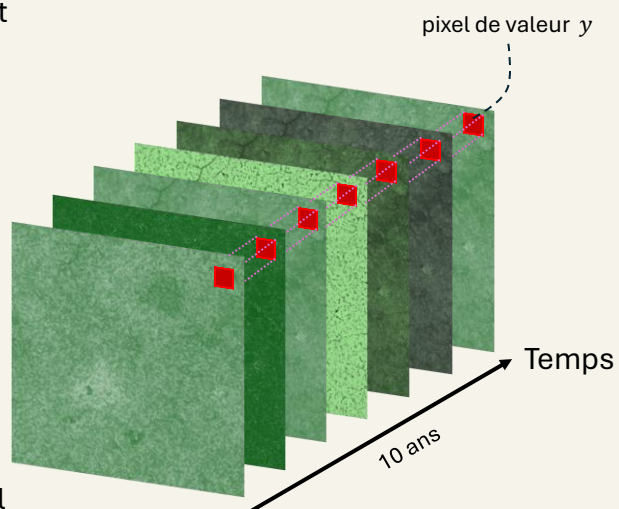
Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

$$NDVI = a \times date + b$$

Valeur continue comme
jours ou millisecons
(`.millis()`) depuis
1970

valeurs spécifiques pour chaque pixel



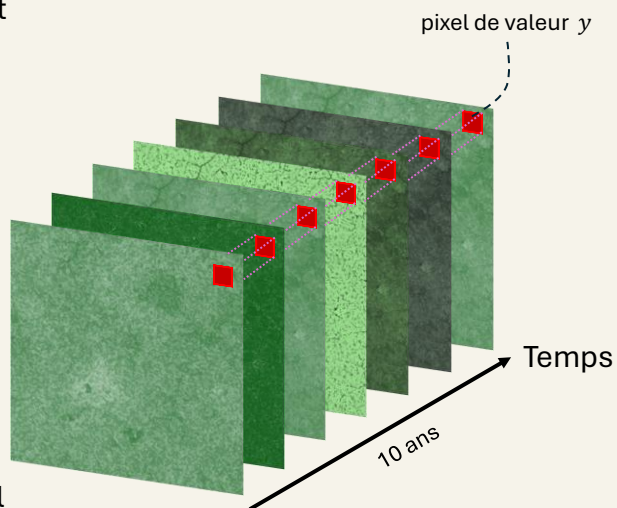
180

Tendance du NDVI sur 10 ans pour un point

$$NDVI = a \times date + b$$

Valeur continue comme
jours ou millisecondes
(`.millis()`) depuis
1970

valeurs spécifiques pour chaque pixel
→ *array image* : un pixel sur une bande peut
contenir ≥ 1 valeur (pente, intercept...)



181

Tendance du NDVI sur 10 ans pour un point

pixel de valeur y



ee.Image

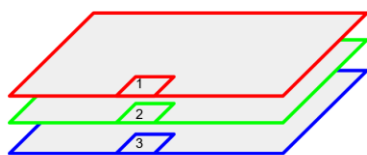
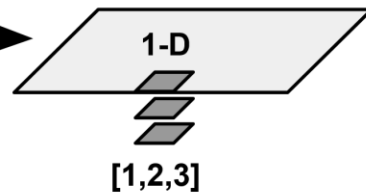


image.toArray()

ee.Image



valeurs spécifiques pour chaque pixel
→ *array image* : un pixel sur une bande peut
contenir ≥ 1 valeur (pente, intercept...)



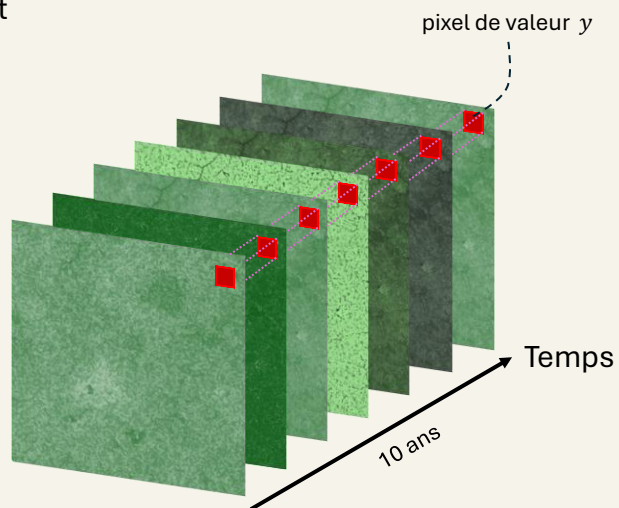
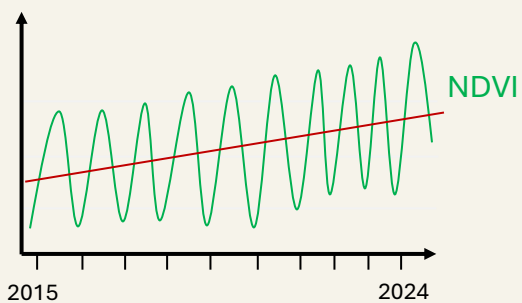
182

Utiliser Google Earth Engine

Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

$$NDVI = a \times date + b$$



183

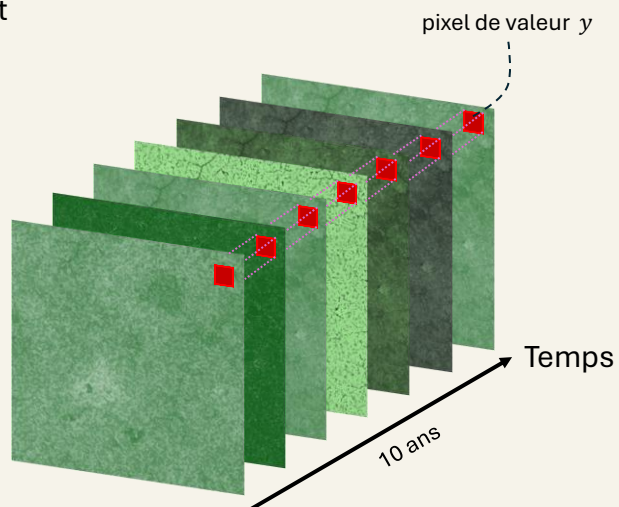
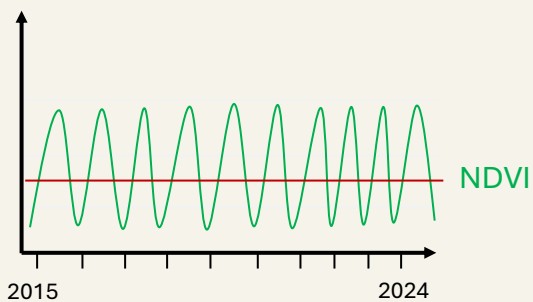
Utiliser Google Earth Engine

Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

$$NDVI = a \times date + b$$

Suppression de la tendance



184

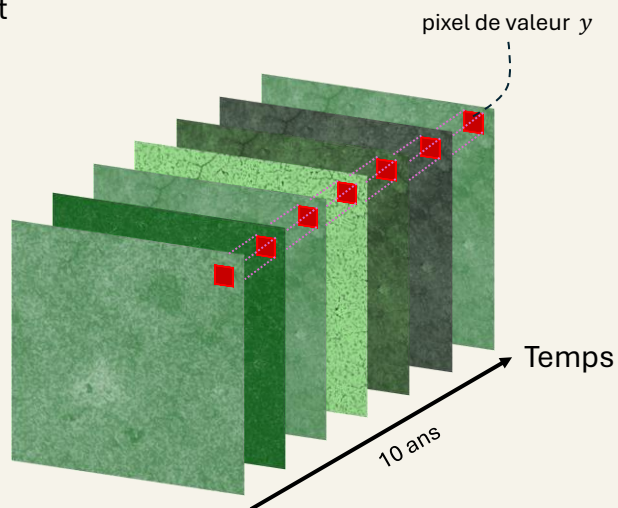
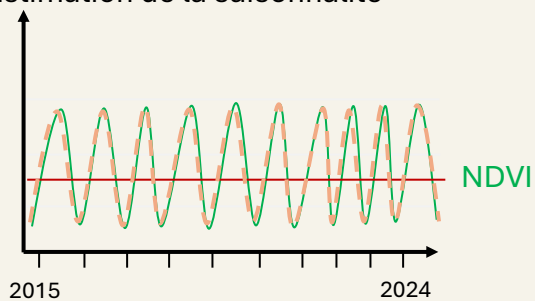
Utiliser Google Earth Engine

Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

$$NDVI = a \times date + b$$

Suppression de la tendance
Estimation de la saisonnalité



185

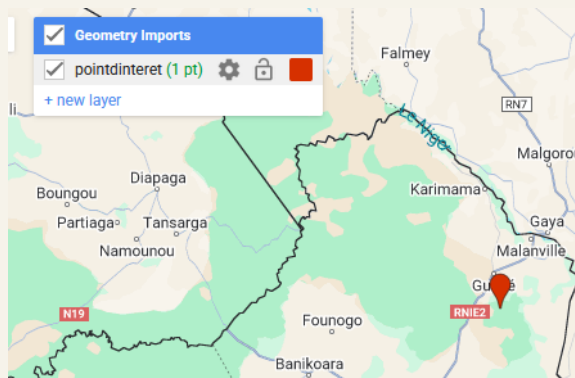
Utiliser Google Earth Engine

Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

- Créer un point d'intérêt

'pointdinteret'



186

Tendance du NDVI sur 10 ans pour un point

- Ajouter la fonction **maskS2clouds** ----->
- Modifier la dernière ligne pour que la date de chaque image soit conservée :

```
return image.updateMask(mask).divide(10000)
      .copyProperties(image, ['system:time_start']);
```

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



Dataset Availability
 2015-06-27T00:00:00Z -
 2025-01-29T03:15:36Z

Dataset Provider
[European Union/ESA/Copernicus](#)

Earth Engine Snippet

```
ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
```

Revisit Interval
 5 Days

Tags

copernicus esa eu esi
 radiance sentinel

187

Tendance du NDVI sur 10 ans pour un point

```
var S2_NDVI = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
  .filterDate('2015-01-01', '2025-01-01')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
  .filterBounds(pointdinteret)
  .map(maskS2clouds)
```

188

Tendance du NDVI sur 10 ans pour un point

```

var S2_NDVI = ee.ImageCollection('COPERNICUS/S2_HARMONIZED')
  .filterDate('2015-01-01', '2025-01-01')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
  .filterBounds(pointdinteret)
  .map(maskS2clouds)
  .map(function(image) {
    var date = ee.Date(image.get('system:time_start'));
    var years = date.difference(ee.Date('1970-01-01'), 'year');
    return image.normalizedDifference(['B8', 'B4']).rename('ndvi')
      .addBands(ee.Image(years).rename('t')).float()
      .addBands(ee.Image.constant(1))
      .copyProperties(image, ['system:time_start']);
  });

```

$$NDVI = a \times date + b$$

189

Tendance du NDVI sur 10 ans pour un point

```

print(S2_NDVI);

```

- ▼ ImageCollection COPERNICUS/S2_HARMONIZED (287 elements)
 - type: ImageCollection
 - id: COPERNICUS/S2_HARMONIZED
 - version: 1738137514264500
 - bands: []
 - ▼ features: List (287 elements)
 - ▼ 0: Image (3 bands)
 - type: Image
 - ▼ bands: List (3 elements)
 - ▶ 0: "ndvi", float, EPSG:32631, 10980x10980 px
 - ▶ 1: "t", float, EPSG:4326
 - ▶ 2: "constant", int ∈ [1, 1], EPSG:4326
 - ▼ properties: Object (3 properties)
 - ▶ system:footprint: LinearRing, 22 vertices
 - ▶ system:index: 20151012T102515_20160714T180520_T31PEN
 - ▶ system:time_start: 1444645515576
 - ▶ 1: Image (3 bands)
 - ▶ 2: Image (3 bands)

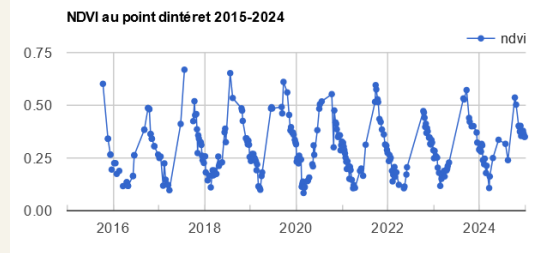
190

Utiliser Google Earth Engine

Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

```
var S2NDVIplot = ui.Chart.image.series({
  imageCollection: S2_NDVI.select('ndvi'),
  region: pointdinteret
})
.setChartType('ScatterChart')
.setOptions({
  title: 'NDVI au point dintéret 2015-2024',
  lineWidth: 1,
  pointSize: 3,
});
print(S2NDVIplot);
```



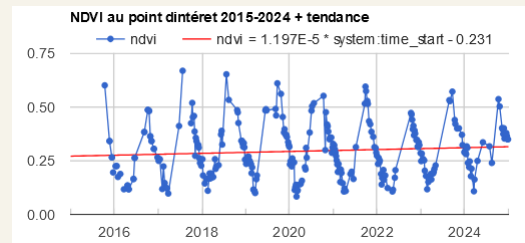
191

Utiliser Google Earth Engine

Cas 14 – Tendances

Tendance du NDVI sur 10 ans pour un point

```
var S2NDVIplot = ui.Chart.image.series({
  imageCollection: S2_NDVI.select('ndvi'),
  region: pointdinteret
})
.setChartType('ScatterChart')
.setOptions({
  title: 'NDVI au point dintéret 2015-2024 + tendance',
  trendlines: {0: {type: 'linear', color: 'red',
    visibleInLegend: true}},
  lineWidth: 1,
  pointSize: 3,
});
print(S2NDVIplot);
```



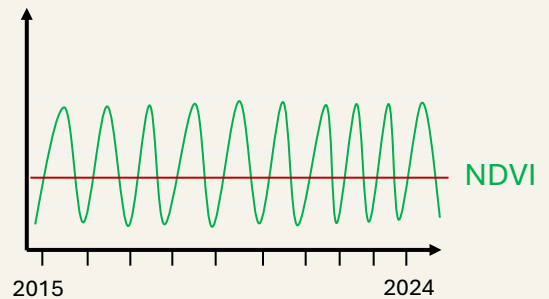
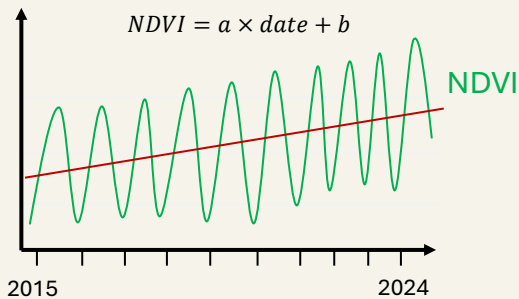
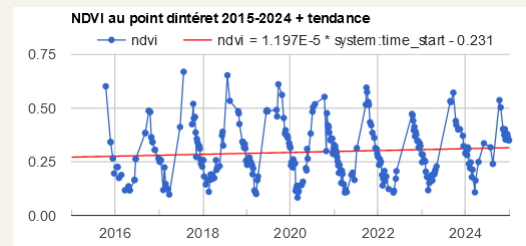
sinon 'polynomial' ou 'exponential'

pour la première série temporelle (0-indexed)

192

Correction de la tendance

- Applications
- Calcul de la valeur du NDVI sans l'effet de tendance



193

Correction de la tendance : à l'échelle de l'image

Production de nouvelles images avec les coefficients et résidus

```
var independantes = ee.List(['constant', 't']);
var dependante = ee.String('ndvi');

var tendance = S2_NDVI.select(independantes.add(dependante))
  .reduce(ee.Reducer.linearRegression(independantes.length(), 1));
```

ce Reducer fait une regression linéaire pour la série temporelle de chaque pixel avec deux variables indépendantes (2 = `independantes.length()`) et une variable dépendante (1)

194

Correction de la tendance : à l'échelle de l'image

Production de nouvelles images avec les coefficients et

```

var independantes = ee.List(['constant', 't']);
var dependante = ee.String('ndvi');

var tendance = S2_NDVI.select(independantes.add(dependante))
  .reduce(ee.Reducer.linearRegression(independantes.length(), 1));

Map.addLayer(tendance, {}, 'tendance en array image', false);
print(tendance);

```

```

Image (2 bands)
  type: Image
  bands: List (2 elements)
    0: "coefficients", double, 2 dimensions, EPSG:4326
      id: coefficients
      crs: EPSG:4326
      crs_transform: [1,0,0,0,1,0]
      data_type: double, 2 dimensions
        type: PixelType
        dimensions: 2
        precision: double
    1: "residuals", double, 1 dimension, EPSG:4326
      id: residuals
      crs: EPSG:4326
      crs_transform: [1,0,0,0,1,0]
      data_type: double, 1 dimension

```

195

Correction de la tendance : à l'échelle de l'image

Production de nouvelles images avec les coefficients et

```

var independantes = ee.List(['constant', 't']);
var dependante = ee.String('ndvi');

var tendance = S2_NDVI.select(independantes.add(dependante))
  .reduce(ee.Reducer.linearRegression(independantes.length(), 1));

Map.addLayer(tendance, {}, 'tendance en array image', false);
print(tendance);

var coefficients = tendance.select('coefficients')
  .arrayProject([0])
  .arrayFlatten([independantes]);

```

```

Image (2 bands)
  type: Image
  bands: List (2 elements)
    0: "coefficients", double, 2 dimensions, EPSG:4326
      id: coefficients
      crs: EPSG:4326
      crs_transform: [1,0,0,0,1,0]
      data_type: double, 2 dimensions
        type: PixelType
        dimensions: 2
        precision: double
    1: "residuals", double, 1 dimension, EPSG:4326
      id: residuals
      crs: EPSG:4326
      crs_transform: [1,0,0,0,1,0]
      data_type: double, 1 dimension

```

L'array image est convertie en image à deux bandes pour les coefficients (plus simple)

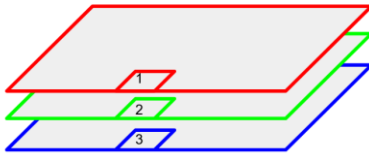
196

Correction de la tendance : à l'échelle de l'image

Production de nouvelles images avec les coefficients et

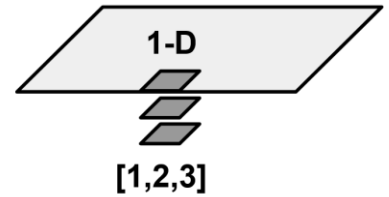
```
var independantes = ee.List(['constant', 't']);
var dependante = ee.String('ndvi');
```

ee.Image



```
Image (2 bands)
  type: Image
  bands: List (2 elements)
    0: "coefficients", double, 2 dimensions, EPSG:4326
      id: coefficients
      crs: EPSG:4326
      crs_transform: [1,0,0,0,1,0]
      data_type: double, 2 dimensions
      type: PixelType
      dimensions: 2
      precision: double
    1: "residuals", double, 1 dimension, EPSG:4326
      id: residuals
      crs: EPSG:4326
      crs_transform: [1,0,0,0,1,0]
```

ee.Image



```
var coefficients = tendance.select('coefficients')
  .arrayProject([0])
  .arrayFlatten([independantes]);
```

L'array image est convertie en image à deux bandes pour les coefficients (plus simple)

197

Correction de la tendance : à l'échelle de l'im

Production de nouvelles images avec les coe

```
var independantes = ee.List(['constant', 't']);
var dependante = ee.String('ndvi');
```

```
var tendance = S2_NDVI.select(independantes.add(dependante))
  .reduce(ee.Reducer.linearRegression(independantes.length(), 1));
```

```
Map.addLayer(tendance, {}, 'tendance en array image', false);
print(tendance);
```

```
var coefficients = tendance.select('coefficients')
  .arrayProject([0])
  .arrayFlatten([independantes]);
print(coefficients);
```

```
Image (2 bands)
  type: Image
  bands: List (2 elements)
    0: "constant", double, EPSG:4326
    1: "t", double, EPSG:4326
```

L'array image est convertie en image à deux bandes pour les coefficients (plus simple)

198

Correction de la tendance : à l'échelle de l'image

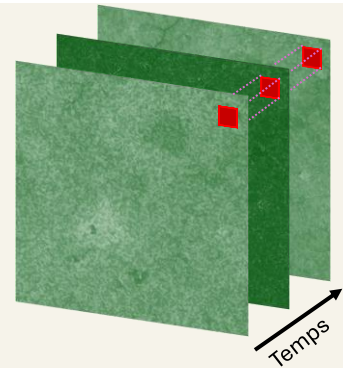
Production de nouvelles images avec les coefficients et résidus

Pour chaque pixel et image on a donc :

- La valeur de la variable dépendante = NDVI
- Les coefficients
- La date

→ NDVI sans tendance = NDVI observé – le NDVI prédit par la tendance

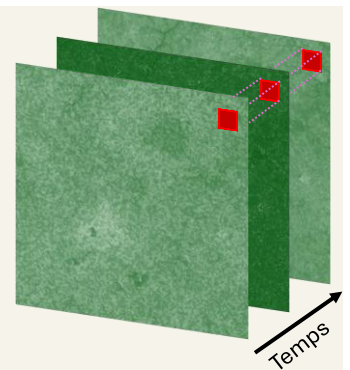
Où le NDVI prédit par la tendance = $a \times date + b = coefficient_1 \times date + coefficient_2$



199

Correction de la tendance : à l'échelle de l'image

```
var sanstendance = S2_NDVI.map(function(image) {
  return image.select(dependante)
    .subtract(image.select(independantes).multiply(coefficients)
      .reduce('sum'))
    .copyProperties(image, ['system:time_start']);
});
```



200

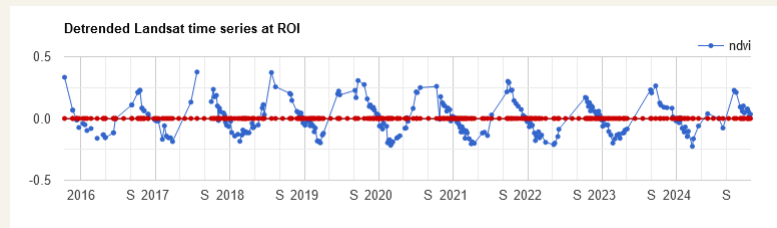
Correction de la tendance : à l'échelle du site

```

var sanstendanceplot = ui.Chart.image.series(sanstendance, pointdinteret,
null, 20)
    .setOptions({
        title: 'Série du NDVI sans tendance',
        lineWidth: 1,
        pointSize: 3,
        trendlines: {0: {color: 'red'}},
    });

print(sanstendanceplot);

```



201

Tendances : Conclusions (1)

```

.millis()
.float()
Image.copyProperties(Image, property)
Image.arrayProject()
Image.arrayFlatten()

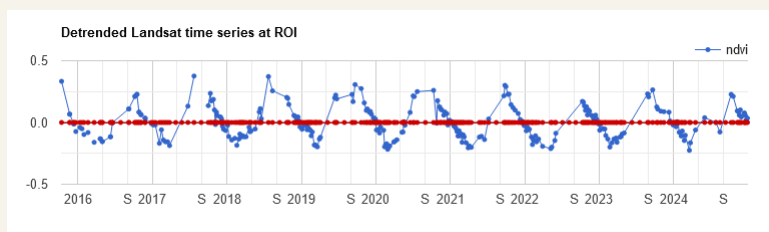
```

202

Périodicité

Autre possibilité : harmonique (série de Fourier)

Les valeurs du NDVI varient entre deux extrêmes (amplitude A) à une fréquence donnée (ω)



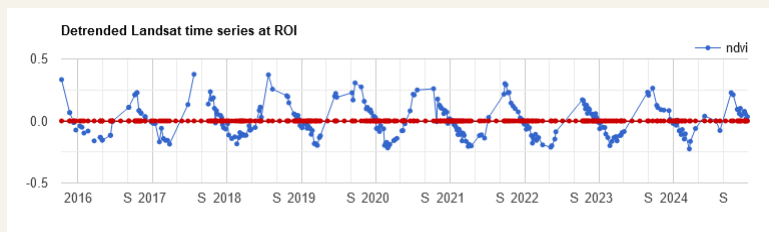
203

Périodicité

Autre possibilité : harmonique (série de Fourier)

Les valeurs du NDVI varient entre deux extrêmes (amplitude A) à une fréquence donnée (ω)

$$NDVI = a \times date + b + c \times \cos(date \times 2\pi\omega) + d \times \sin(date \times 2\pi\omega)$$



204

Périodicité

Autre possibilité : harmonique (série de Fourier)

Les valeurs du NDVI varient entre deux extrêmes (amplitude A) à une fréquence donnée (ω)

$$NDVI = a \times date + b + c \times \cos(date \times 2\pi\omega) + d \times \sin(date \times 2\pi\omega)$$



```
var independantesharmonique = ee.List(['constant', 't', 'cos', 'sin']);
var harmoniqueS2_NDVI = S2_NDVI.map(function(image) {
  var timeRadians = image.select('t').multiply(2 * Math.PI);
  return image
    .addBands(timeRadians.cos().rename('cos'))
    .addBands(timeRadians.sin().rename('sin'));
});
```

205

Périodicité

Autre possibilité : harmonique (série de Fourier)

```
var modelharmonique = harmoniqueS2_NDVI
  .select(independantesharmonique.add(dependante))
  .reduce(ee.Reducer.linearRegression(independantesharmonique.length(),
    1));

var coeffmodelharmonique = modelharmonique.select('coefficients')
  .arrayProject([0])
  .arrayFlatten([independantesharmonique]);
```

206

Utiliser Google Earth Engine

Cas 14 – Tendances

Périodicité

Autre possibilité : harmonique (série de Fourier)

```

var imageharmonique = harmoniqueS2_NDVI.map(function(image) {
  return image.addBands(
    image.select(independantesharmonique)
      .multiply(coeffmodelharmonique)
      .reduce('sum')
      .rename('ndvi_model'));
});

```

207

Utiliser Google Earth Engine

Cas 14 – Tendances

Périodicité

Autre possibilité : harmonique (série de Fourier)

```

var harmoniqueplot = ui.Chart.image.series(
  imageharmonique.select(['ndvi_model', 'ndvi']), pointdinteret,
  ee.Reducer.mean(), 30)
  .setSeriesNames(['ndvi', 'ndvi_model'])
  .setOptions({colors: ['3366CC', 'CB0000']});

print(harmoniqueplot);

```

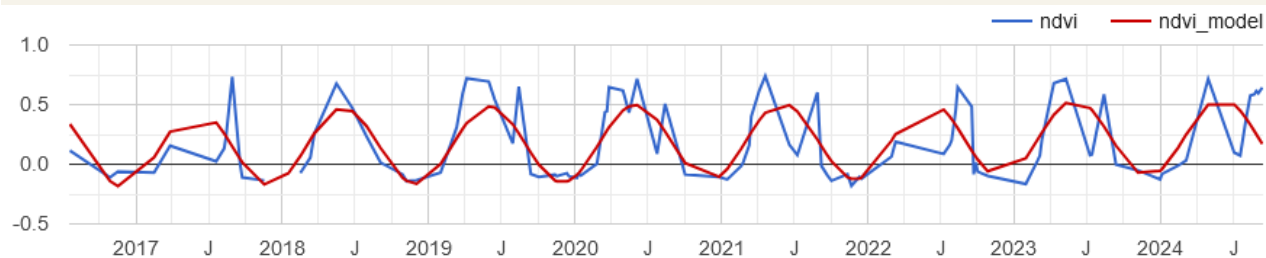
208

Utiliser Google Earth Engine

Cas 14 – Tendances

Périodicité

Autre possibilité : harmonique (série de Fourier)



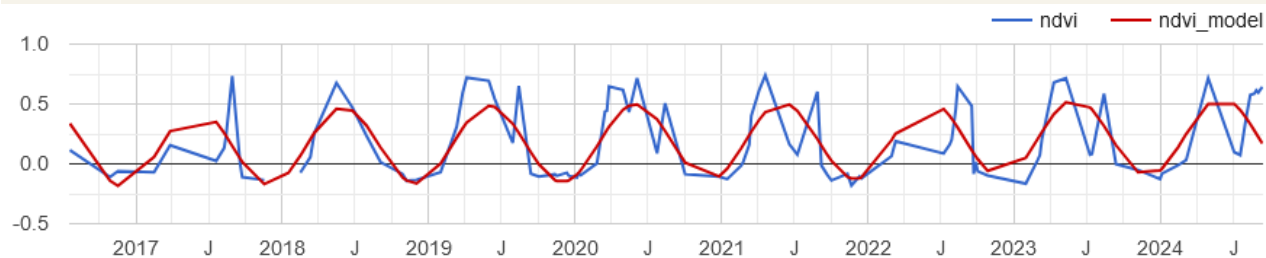
209

Utiliser Google Earth Engine

Cas 14 – Tendances

Périodicité

Autre possibilité : harmonique (série de Fourier)



Variation intra-annuelle. Par exemple: deux récoltes

210

Utiliser Google Earth Engine

Cas 14 – Tendances

Conclusions (2)

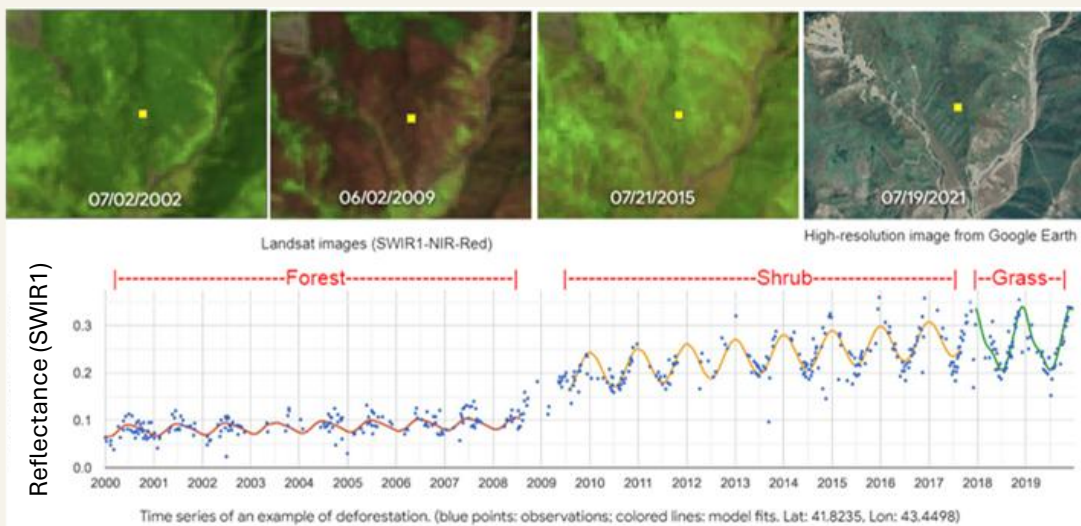
```
Math.PI
.cos()
.sin()
```

211

Utiliser Google Earth Engine

Cas 14 – Tendances

Conclusions (2)



Puzhao Zhang, DHI Water Environment – Geo For Good 2024

212

Utiliser Google Earth Engine

Recommandations

Recommandations

https://developers.google.com/earth-engine/guides/best_practices

213

Utiliser Google Earth Engine

Recommandations

Pour les FeatureCollection et ImageCollection

1) `Filter()` et `select()` puis 2) `map()`

214

Utiliser Google Earth Engine

Recommandations

Pour les FeatureCollection et ImageCollection

1) `Filter()` et `select()` puis 2) `map()`

La fonction `mask()` n'est pas recommandée : plutôt utiliser `updateMask()`

215

Utiliser Google Earth Engine

Recommandations

Pour les FeatureCollection et ImageCollection

1) `Filter()` et `select()` puis 2) `map()`

La fonction `mask()` n'est pas recommandée : plutôt utiliser `updateMask()`

Combiner les `reducer` quand plusieurs statistiques doivent être extraites dans `reduceRegion(s)`.

216

Utiliser Google Earth Engine

Recommandations

Pour les FeatureCollection et ImageCollection

1) `Filter()` et `select()` puis 2) `map()`

La fonction `mask()` n'est pas recommandée : plutôt utiliser `updateMask()`

Combiner les `reducer` quand plusieurs statistiques doivent être extraites dans `reduceRegion(s)`.

Exporter peut marcher même si l'affichage de l'image ne marche pas à cause de la quantité de calculs. Exporter des assets à des étapes du travail est plus efficace.

217

Utiliser Google Earth Engine

Recommandations

`clip()` ne doit être utilisé que lorsque c'est nécessaire (`clipToCollection()` pour une FeatureCollection)

218

`clip()` ne doit être utilisé que lorsque c'est nécessaire (`clipToCollection()` pour une `FeatureCollection`)

Eviter d'utiliser des fonctions demandant beaucoup de mémoire (`buffer`, `FeatureCollection.geometry()`, ...)

219

`clip()` ne doit être utilisé que lorsque c'est nécessaire (`clipToCollection()` pour une `FeatureCollection`)

Eviter d'utiliser des fonctions demandant beaucoup de mémoire (`buffer`, `FeatureCollection.geometry()`, ...)

`errorMargin` est un argument à utiliser pour simplifier certains calculs (quand la précision en mètres n'est pas parfaite)

220

`clip()` ne doit être utilisé que lorsque c'est nécessaire (`clipToCollection()` pour une `FeatureCollection`)

Eviter d'utiliser des fonctions demandant beaucoup de mémoire (`buffer`, `FeatureCollection.geometry()`, ...)

`errorMargin` est un argument à utiliser pour simplifier certains calculs (quand la précision en mètres n'est pas parfaite)

Dans certains cas la combinaison `map()` + `filter()` peut être plus rapide que `ee.Join`.