Tutoriel 5

May 6, 2021

1 Importation des packages

[1]:
```
#**geemap**  : package pour la cartographie interactive avec Google Earth Engine␣
↪(GEE).
try:
    import geemap, ee
except ModuleNotFoundError:
    if 'google.colab' in str(get_ipython()):
        print("package not found, installing w/ pip in Google Colab...")
        !pip install geemap
    else:
        print("package not found, installing w/ conda...")
        !conda install mamba -c conda-forge -y
        !mamba install geemap -c conda-forge -y
    import geemap, ee
```

[ ]:
```
!pip install geopandas
```

2 Authentification

[2]:
```
try:
        ee.Initialize()
except Exception as e:
        ee.Authenticate()
        ee.Initialize()
```

[5]:
```
# **GeoPandas**  facilite le travail avec des données géospatiales en python. et␣
↪ permet des opérations spatiales sur les types géométriques.
import geopandas as gpd
import pandas as pd
from geopandas import GeoDataFrame
from shapely.geometry import Point,Polygon
import numpy as np
from functools import reduce
import shapefile
import geopandas as gpd
```

3 Fonction pour lire les shapefiles

[6]:
```
#shapefile to FeatureCollection
def upload_shapefile_to_gee1(shp_file):
    """
    Upload a shapefile to Google Earth Engine as an asset.

    Args:
        user (django.contrib.auth.User): the request user.
        shp_file (shapefile.Reader): A shapefile reader object.
    """
    features = []
    fields = shp_file.fields[1:]
    field_names = [field[0] for field in fields]

    # Convert Shapefile to ee.Features
    for record in shp_file.shapeRecords():
        # First convert to geojson
        attributes = dict(zip(field_names, record.record))
        geojson_geom = record.shape.__geo_interface__
        geojson_feature = {
            'type': 'Feature',
            'geometry': geojson_geom,
            'properties': attributes
        }

        # Create ee.Feature from geojson (this is the Upload, b/c ee.Feature is␣
 ↪a server object)
        features.append(ee.Feature(geojson_feature))

    feature_collection = ee.FeatureCollection(features)
    return feature_collection
```

[7]:
```
#print a shape file (geojson)
def upload_shapefile_to_gee(shp_file):
    """
    Upload a shapefile to Google Earth Engine as an asset.

    Args:
        user (django.contrib.auth.User): the request user.
        shp_file (shapefile.Reader): A shapefile reader object.
    """
    features = []
    fields = shp_file.fields[1:]
    field_names = [field[0] for field in fields]

    # Convert Shapefile to ee.Features
```

```
    for record in shp_file.shapeRecords():
        # First convert to geojson
        attributes = dict(zip(field_names, record.record))
        geojson_geom = record.shape.__geo_interface__
        geojson_feature = {
            'type': 'Feature',
            'geometry': geojson_geom,
            'properties': attributes
        }

        print(geojson_feature)
```

[ ]:
```
sf = shapefile.Reader("regions_updated.shp")
shapeRecs = sf.shapeRecords()
shapeRecs.__geo_interface__['type']
upload_shapefile_to_gee(sf)
shapes = sf.shapes()
shapes
```

[ ]:
```
shapefile = gpd.read_file("regions_updated.shp")
print(shapefile)
shapefile.head(12)
print(shapeRecs.__geo_interface__)
```

4 Shapefile Visualization

[11]:
```
ee_object = geemap.shp_to_ee("regions_updated.shp")
```

[13]:
```
#Exemple
viirs2020_12 = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG").
 ↪filterDate("2020-12-01","2020-12-31").select('avg_rad').median()
# initialize our map
map1 = geemap.Map()
map1.add_basemap('SATELLITE')
map1.addLayer(viirs2020_12, {}, "VIIRS-DNB Dec 2020")
map1.addLayer(ee_object, {}, 'Morocco subregions')
map1.addLayerControl()
map1
```

```
Map(center=[40, -100], controls=(WidgetControl(options=['position',␣
 ↪'transparent_bg'], widget=HBox(children=(T...
```

5 Extraction de SOL

[10]:
```
# revise our reducer function to be to get SOL for morocco
def get_morocco_sol(img):
    sol = img.reduceRegion(reducer=ee.Reducer.sum(), geometry=morocco00,␣
 ↪scale=500, maxPixels=1e12).get('avg_rad')
    return img.set('date', img.date().format()).set('SOL',sol)
```

[11]:
```
def Sol(s,Date1,Date2):
    global morocco00

    viirs = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG").
 ↪filterDate(Date1,Date2)
    morocco00 = ee.FeatureCollection(upload_shapefile_to_gee1(sf)).filter(ee.
 ↪Filter.eq('name', s)).first().geometry()
    get_morocco_sol
    morocco00_sol = viirs.map(get_morocco_sol)

    # get lists
    nested_list = morocco00_sol.reduceColumns(ee.Reducer.toList(2),␣
 ↪['date','SOL']).values().get(0)

    # convert to dataframe
    soldf = pd.DataFrame(nested_list.getInfo(), columns=['date','SOL'])
    soldf = soldf.rename(columns={'SOL': s})
    return soldf
```

[17]:
```
Date1='2014-01-01'
Date2='2021-03-01'
```

[13]:
```
Regions = ['Tanger-Tétouan-Al Hoceima', 'Fès-Meknès', 'Beni Mellal-Khénifra',␣
 ↪'Rabat-Salé-Kénitra', 'Casablanca-Settat', 'Marrakech-Safi', 'Draa-Tafilalet',␣
 ↪'Souss-Massa', 'Guelmim-Oued Noun', 'Laayoune-Sakia-El-Hamra', 'Dakhla-Oued␣
 ↪Ed-Dahab']
```

[14]:
```
len(Regions)
```

[14]:
```
11
```

[18]:
```
D1=Sol('Oriental', Date1, Date2)
D1
```

[18]:

	date	Oriental
0	2014-01-01T00:00:00	131843.684245
1	2014-02-01T00:00:00	129066.644328
2	2014-03-01T00:00:00	125839.063910
3	2014-04-01T00:00:00	114641.407431

```
4   2014-05-01T00:00:00  125517.813221
..                  ...            ...
79  2020-08-01T00:00:00  182982.674992
80  2020-09-01T00:00:00  192693.403151
81  2020-10-01T00:00:00  201118.837390
82  2020-11-01T00:00:00  234669.928200
83  2020-12-01T00:00:00  235208.519730

[84 rows x 2 columns]
```

[19]:
```
def qwe(regions):
    D1=Sol('Oriental',Date1,Date2)
    D=D1
    for i in regions :
        D=D.join(Sol(i,Date1,Date2)[i], on=None, how='left', lsuffix='',␣
 ↪rsuffix='', sort=False)
    return D
```

[20]:
```
Data=qwe(Regions)
Data
```

[20]:
```
                   date  ...  Dakhla-Oued Ed-Dahab
0   2014-01-01T00:00:00  ...          32942.090562
1   2014-02-01T00:00:00  ...          48540.585423
2   2014-03-01T00:00:00  ...         119392.083914
3   2014-04-01T00:00:00  ...         109252.289501
4   2014-05-01T00:00:00  ...          44178.139672
..                  ...  ...                   ...
79  2020-08-01T00:00:00  ...         145599.580255
80  2020-09-01T00:00:00  ...         207152.904109
81  2020-10-01T00:00:00  ...         189088.816849
82  2020-11-01T00:00:00  ...         242521.028459
83  2020-12-01T00:00:00  ...         180940.780257

[84 rows x 13 columns]
```

[21]:
```
Data['date'] = pd.to_datetime(Data['date'])
```

[22]:
```
Data
```

[22]:
```
         date       Oriental  ...  Laayoune-Sakia-El-Hamra  Dakhla-Oued Ed-Dahab
0  2014-01-01  131843.684245  ...             71744.245857          32942.090562
1  2014-02-01  129066.644328  ...             86537.865948          48540.585423
2  2014-03-01  125839.063910  ...            159659.931900         119392.083914
3  2014-04-01  114641.407431  ...            134870.148482         109252.289501
4  2014-05-01  125517.813221  ...             74541.111365          44178.139672
..        ...            ...  ...                      ...                   ...
```

```
79 2020-08-01  182982.674992  ...          182427.896096         145599.580255
80 2020-09-01  192693.403151  ...          261474.966889         207152.904109
81 2020-10-01  201118.837390  ...          216067.795730         189088.816849
82 2020-11-01  234669.928200  ...          295499.072240         242521.028459
83 2020-12-01  235208.519730  ...          256522.026617         180940.780257

[84 rows x 13 columns]
```

[23]:
```
Data.to_csv(r'/content/MyData.csv')
```