```
!pip install earthpy
!pip install rasterstats
```

```
# mouting to Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Bibliotecas

```
import earthpy as et
import earthpy.spatial as es
import earthpy.plot as ep

import os
from glob import glob

import rasterio as rio
from rasterio.plot import show
from rasterio.plot import plotting_extent

import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import matplotlib.patches as mpatches
from matplotlib import pyplot

import numpy as np

import pandas as pd
import geopandas as gpd

from rasterstats import point_query
```

Visualizando a imagem Landsat 8

1ª etapa. Importar as bibliotecas:

- `import earthpy.spatial as es`
- `import earthpy.plot as ep`
- `from glob import glob`
- `import matplotlib.pyplot as plt`

2ª etapa. Obter lista de bandas e classificar por número crescente de bandas:

- inserir o diretório das imagens e, entre colchetes [ ], digitar o intervalo de bandas que irá trabalhar.

```
landsat_bands_data_path =
"C:/Users/eltes/Desktop/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T1_B*[1-
7]*.tif"
```

- O módulo `glob` encontra todos os nomes de caminho que correspondem a um padrão especificado de acordo com as regras usadas pelo shell Unix

```
stack_band_paths = glob(landsat_bands_data_path)
```

- ordernar por número crescente

```
stack_band_paths.sort()
```

- Criar uma pilha de imagens e apliquar o valor nodata para o Landsat

```
arr_st, meta = es.stack(stack_band_paths, nodata=-9999)
```

```
# Obter lista de bandas e classificar por número crescente de bandas

#Lembre-se de utilizar .TIF (maísculo)
landsat_bands_data_path = "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_
stack_band_paths = glob(landsat_bands_data_path)
stack_band_paths.sort()

#imprima a saída dos arquivos
stack_band_paths
```

```
['/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T

'/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T1

'/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T1

'/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T1

'/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T1

'/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T1

'/content/drive/MyDrive/Artigo_Nuvens/LandSat8/LC08_L1TP_227062_20180721_20180731_01_T1
```

```
# Criar uma pilha de imagens e apliquar o valor NODATA para o Landsat
arr_st, meta = es.stack(stack_band_paths, nodata=-9999)
```

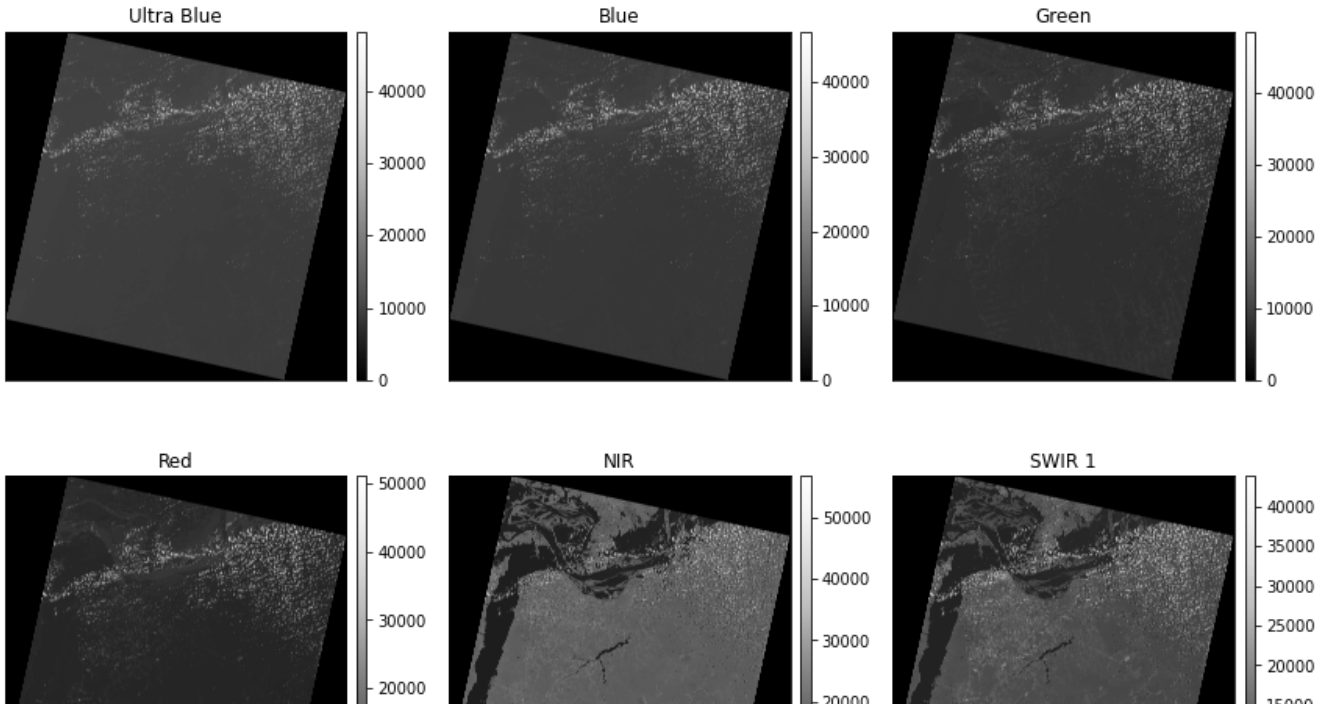
3ª etapa - Opicional. Plotar todas as bandas em um "Grid" sequencial:

- Inserir o títulos de cada banda. NOTE que o número de títulos precisa ser igual ao número de bandas que foi obtidos na 2ª etapa (no nosso caso, estamos trabalhando com 7 bandas (1-7)

  ```
  titles = ["Ultra Blue", "Blue", "Green", "Red", "NIR", "SWIR 1", "SWIR 2"]
  ```

- #sphinx_gallery_thumbnail_number = 1 (opicional)
- Plotar as bandas com `earthpy`

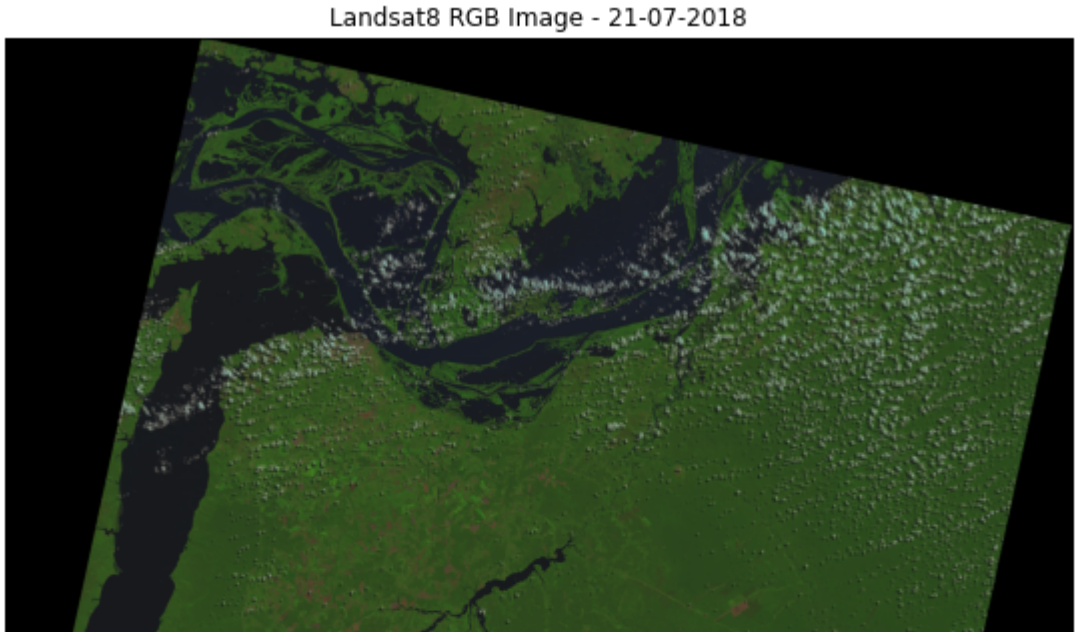
```
# Opicional. Plotar todas as bandas em um "Grid" sequencial:
titles = ["Ultra Blue", "Blue", "Green", "Red", "NIR", "SWIR 1", "SWIR 2"]
sphinx_gallery_thumbnail_number = 1
ep.plot_bands(arr_st, title=titles)
plt.show()

```



4ª etapa. Plotar a imagem do LandSat 8:

- Criar figura com um gráfico
- Gerar imagem RGB com o `earthpy`. NOTE que a Banda 1 (Ultra Blue) será o index 0 e a Banda 7 (SWIR 2) será o index 6

```
# Criar uma figura com um único plot (Create figure with one plot)
fig, ax = plt.subplots(figsize=(12, 10))

# Plotar as bandas do Vermelho, Verde e Azul (Plot red, green, and blue bands, respectivel
ep.plot_rgb(arr_st, rgb=(5, 4, 3), ax=ax, title="Landsat8 RGB Image - 21-07-2018")
#plt.show()

# Salvar imagem
plt.savefig('Imagem_RGB.png',dpi=300,format='png',orientation='landscape')
```



Esquema inicial - fmask

1ª etapa. Preparar o ambiente conda:

1. Acessar o site *Anaconda Cloud* para sabermos em que *canal* que o pacote está disponível (https://anaconda.org/conda-forge/python-fmask)
2. No prompt:

```
conda create --name geo_ame

conda activate geo_ame

conda search fmask # verificar se a biblioteca está no ambiente

conda install -c conda-forge python-fmask

conda config --add channels conda-forge
```

2ª etapa. Preparar as imagens:

1. Conferir os arquivos MTL.txt

- O fmask utiliza imagens de refletância de topo da atmosfera (TOA) e por isso precisa do arquivo MLT.txt
- O pacote fmask tem um script submodulo *(landsatTOA)* que converte ND em TOA

2. Colocar as bandas em um unico diretório

3ª etapa. Executar o algoritmo fmask no prompt

input landsat:

- fmask_usgsLandsatStacked.py -o cloud.img --scenedir **diretório que estão as imagens**

```
fmask_usgsLandsatStacked.py -o cloud.img --scenedir
C:\dados_traba_ser\landsat_nd\LC08_L1TP_227062_20180721_20180731_01_T1
```

- Foram usadas todas as bandas

Obs: **cloud.img** é nome do arquivo de saída

Entendendo o arquivo de saída

- cloud.img vai aparecer lá no **user**

Descrição:

Arquivo saí em raster com valores de 0 à 5:

- 0 = Null
- 1 = Área sem nuvem
- 2 = Nuvem
- 3 = Sombra de nuvem
- 4 = Neve
- 5 = Água

Visualização Imagem do *fmask*

1ª etapa. Bibliotecas necessárias `rasterio` e `earthpy`. Caso precisem ser instaladas, executar no prompt os seguintes comandos:

- Rasterio

  ```
  conda install -c conda-forge rasterio
  ```

- EarthPy

  ```
  conda install -c conda-forge earthpy
  ```

2ª etapa. Importar bibliotecas `matplotlib`, `rasterio` e `show` do rasterio

```
# import matplotlib.pyplot as plt
# import rasterio as rio
# from rasterio.plot import show
```

3ª etapa. Definindo o caminho da imagem. Lembre-se que precisa alterar o caminho que a imagem está salva!

```
# Definir o caminho da imagem (define path to the image)
img2018 = "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/cloud.img"
```

4ª etapa. Abrir a imagem no `rasterio`

```
img2018_rasterio = rio.open(img2018)
```

5ª etapa. Opicional - Visualizar a extensão espacial

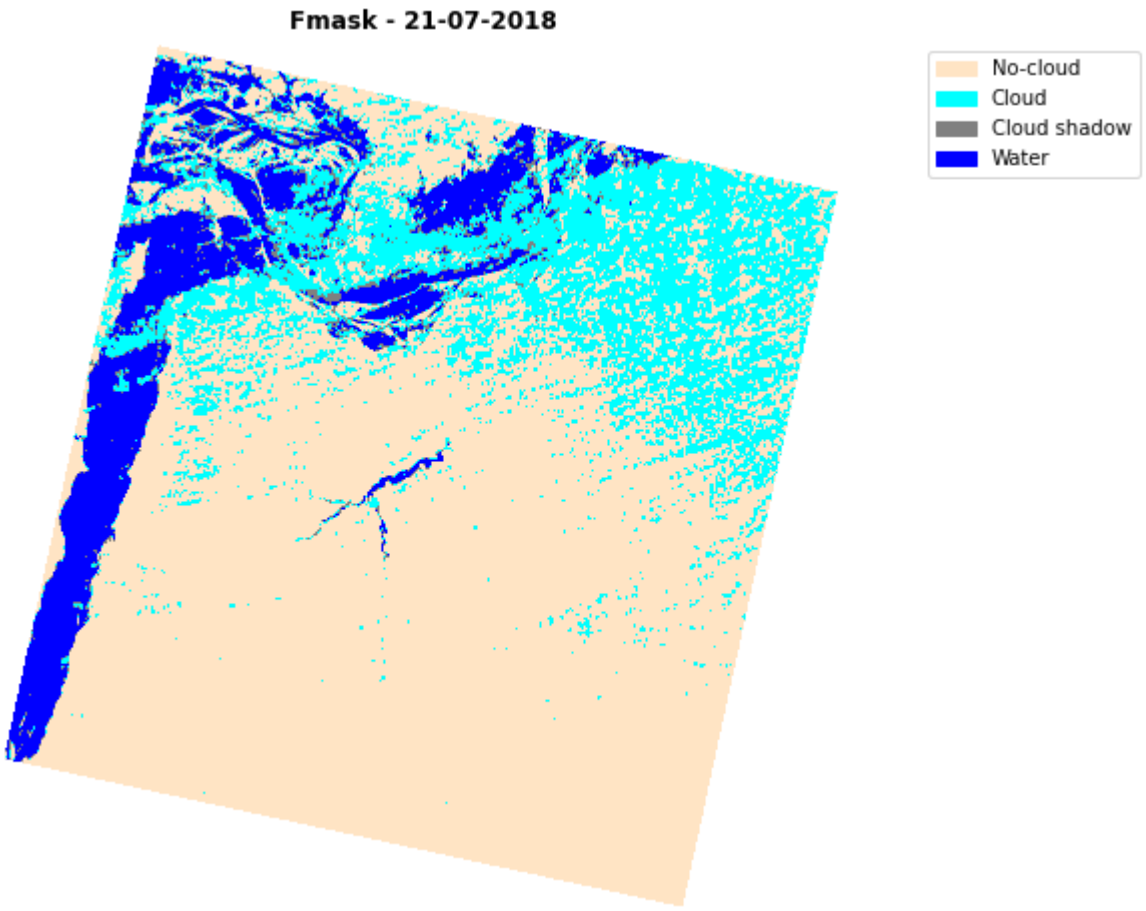
```
# Opicional - Visualizar a extensão espacial da imagem (optional - view spatial extent)
img2018_rasterio.bounds
```

```
BoundingBox(left=678885.0, bottom=-436815.0, right=907515.0, top=-203985.0)
```

6ª etapa. Plotar a imagem usando `rasterio`

```
# definindo as cores com ListedColormap
cmap=ListedColormap(['bisque', 'aqua', 'grey', 'blue'])

# Plotar a imagem gerada pelo fsmak utilizando rasterio (plot fmask using rasterio)
fig, ax = plt.subplots(figsize = (10,8))
show(img2018_rasterio, title="Fmask - 21-07-2018",ax=ax,cmap=cmap)
ax.set_axis_off()

# Inserindo legenda personalizada
leg1 = mpatches.Patch(color='bisque', label='No-cloud')
leg2 = mpatches.Patch(color='aqua', label='Cloud')
leg3 = mpatches.Patch(color='grey', label='Cloud shadow')
leg4 = mpatches.Patch(color='blue', label='Water')

# Legenda
plt.legend(handles=[leg1, leg2, leg3,leg4], loc='upper right', bbox_to_anchor=(1.35, 1))

#plt.show();

```

```
20 # Salvar imagem
21 fig.savefig('fmask-julho2018.png',dpi=300,format='png',orientation='landscape')
```

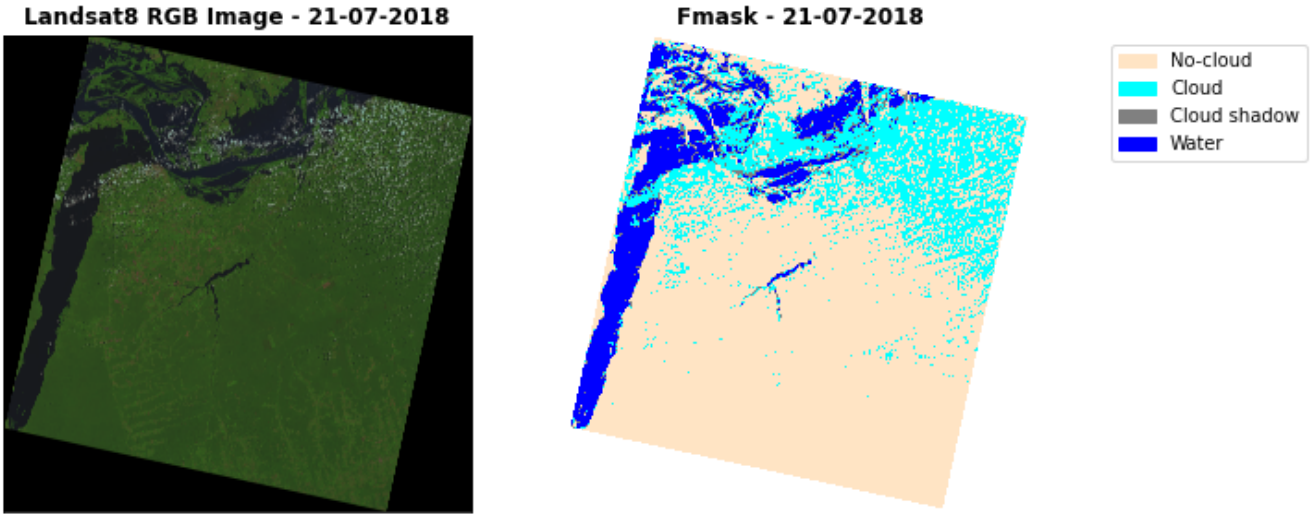


Criar figura com vários eixos ou subplots

```
 1 # Bibliotecas utilizadas
 2 # import matplotlib.pyplot as plt
 3 # import rasterio as rio
 4 # from rasterio.plot import show
 5
 6 # Criar uma figura com 2 plots (Create figure with two plots)
 7 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 8))
 8 ax.set_axis_off()
 9
10 # definindo as cores com ListedColormap
11 cmap=ListedColormap(['bisque', 'aqua', 'grey', 'blue'])
12
13 # Plotar 2 Imagens (Plot 2 Images)
14 ax1 = ep.plot_rgb(arr_st, rgb=(5, 4, 3), ax=ax1,)
15
16 ax2 = show(img2018_rasterio,ax=ax2,cmap=cmap)
17 plt.axis('off')
18
19 # Adicionar títulos (Add titles)
```

```
# (Está separado, pois ax1 usa a biblioteca earthpy enquanto que ax2 usa rasterio - assim
ax1.set_title("Landsat8 RGB Image - 21-07-2018",fontname="Times New Roman",fontweight="bol
ax2.set_title("Fmask - 21-07-2018",fontname="Times New Roman",fontweight="bold")

# Inserindo legenda personalizada
leg1 = mpatches.Patch(color='bisque', label='No-cloud')
leg2 = mpatches.Patch(color='aqua', label='Cloud')
leg3 = mpatches.Patch(color='grey', label='Cloud shadow')
leg4 = mpatches.Patch(color='blue', label='Water')

# Legenda
plt.legend(handles=[leg1, leg2, leg3,leg4], loc='upper right', bbox_to_anchor=(1.6, 1))

plt.show()

# Salvar imagem
fig.savefig('RGB+fmask.png',dpi=300,format='png',orientation='landscape',bbox_inches = 'ti
```

```
findfont: Font family ['Times New Roman'] not found. Falling back to DejaVu Sans.
```



Abrir o arquivo shapefile do centroide do pixel com o geopandas

```
# Arquivo com o centróide do PIXEL analisado
centroide=gpd.read_file('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide
```

```
# Criando a extensão do plot
plot_extent
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-17-b3c9febe9530> in <module>()
      1 # Criando a extensão do plot
----> 2 plot_extent

NameError: name 'plot_extent' is not defined
```

```
# ESCREVER O CODIGO

# Criar uma figura com 2 plots (Create figure with two plots)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 8))
ax.set_axis_off()

# definindo as cores com ListedColormap
cmap=ListedColormap(['bisque', 'aqua', 'grey', 'blue'])

# Plotar 2 Imagens (Plot 2 Images)
ep.plot_rgb(arr_st, rgb=(5, 4, 3), ax=ax1, extent=plotting_extent(img2018_rasterio))
centroide.plot(ax=ax1, marker='s', markersize=45, color='none', edgecolor='red')

ax2 = show(img2018_rasterio,ax=ax2,cmap=cmap)
centroide.plot(ax=ax2, marker='s', markersize=45, color='none', edgecolor='red')
plt.axis('off')

# Adicionar títulos (Add titles)
# (Está separado, pois ax1 usa a biblioteca earthpy enquanto que ax2 usa rasterio - assim
ax1.set_title("Landsat8 RGB Image - 21-07-2018",fontname="Times New Roman",fontweight="bol
ax2.set_title("Fmask - 21-07-2018",fontname="Times New Roman",fontweight="bold")

# Inserindo legenda personalizada
leg1 = mpatches.Patch(color='bisque', label='No-cloud')
leg2 = mpatches.Patch(color='aqua', label='Cloud')
leg3 = mpatches.Patch(color='grey', label='Cloud shadow')
leg4 = mpatches.Patch(color='blue', label='Water')

# Legenda
plt.legend(handles=[leg1, leg2, leg3,leg4], loc='upper right', bbox_to_anchor=(1.6, 1))

plt.show()


# Salvar imagem
fig.savefig('RGB+fmask+centroide.png',dpi=300,format='png',orientation='landscape',bbox_ir

```

Visualização do fmask de todos os anos

Bibliotecas necessárias

- import matplotlib.pyplot as plt
- from matplotlib.colors import ListedColormap

1ª etapa. Caminho do diretório 2017 e 2018

2017

```
# January 2017
cloud_1_17 = "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud01_17.img"

# February 2017
cloud_2_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud02_17.img"

# March 2017
cloud_3_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud03_17.img"

# April 2017
cloud_4_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud04_17.img"

# May 2017
cloud_5_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud05_17.img"

# June 2017
cloud_6_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud06_17.img"

# July 2017
cloud_7_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud07_17.img"

# August 2017
cloud_8_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud08_17.img"

# September 2017
cloud_9_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud09_17.img"

# October 2017
cloud_10_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud10_17.img"

# November 2017
cloud_11_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud11_17.img"

# December 2017
cloud_12_17= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2017/cloud12_17.img"
```

2018

```
# January 2018
cloud_1_18 = "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud01_18.img"

# February 2018
cloud_2_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud02_18.img"

# March 2018
cloud_3_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud03_18.img"

# April 2018
cloud_4_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud04_18.img"

# May 2018
cloud_5_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud05_18.img"

# June 2018
cloud_6_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud06_18.img"

# July 2018
cloud_7_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud07_18.img"

# August 2018
cloud_8_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud08_18.img"

# September 2018
cloud_9_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud09_18.img"

# October 2018
cloud_10_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud10_18.img"

# November 2018
cloud_11_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud11_18.img"

# December 2018
cloud_12_18= "/content/drive/MyDrive/Artigo_Nuvens/LandSat8/fmask_2018/cloud12_18.img"
```

2ª etapa. Abrir os aquivos:

- Abrir os arquivos pela `rasterio`
- Usar *try/except* para abrir os arquivos

2017

```
try:
    with rio.open(cloud_1_17) as cloud_1_17_rasterio:
     cloud_1_17_rasterio_data = cloud_1_17_rasterio.read(1)

    with rio.open(cloud_2_17) as cloud_2_17_rasterio:
```

```
     cloud_2_17_rasterio_data = cloud_2_17_rasterio.read(1)

    with rio.open(cloud_3_17) as cloud_3_17_rasterio:
     cloud_3_17_rasterio_data = cloud_3_17_rasterio.read(1)

    with rio.open(cloud_4_17) as cloud_4_17_rasterio:
     cloud_4_17_rasterio_data = cloud_4_17_rasterio.read(1)

    with rio.open(cloud_5_17) as cloud_5_17_rasterio:
     cloud_5_17_rasterio_data = cloud_5_17_rasterio.read(1)

    with rio.open(cloud_6_17) as cloud_6_17_rasterio:
     cloud_6_17_rasterio_data = cloud_6_17_rasterio.read(1)

    with rio.open(cloud_7_17) as cloud_7_17_rasterio:
     cloud_7_17_rasterio_data = cloud_7_17_rasterio.read(1)

    with rio.open(cloud_8_17) as cloud_8_17_rasterio:
     cloud_8_17_rasterio_data = cloud_8_17_rasterio.read(1)

    with rio.open(cloud_9_17) as cloud_9_17_rasterio:
     cloud_9_17_rasterio_data = cloud_9_17_rasterio.read(1)

    with rio.open(cloud_10_17) as cloud_10_17_rasterio:
     cloud_10_17_rasterio_data = cloud_10_17_rasterio.read(1)

    with rio.open(cloud_11_17) as cloud_11_17_rasterio:
     cloud_11_17_rasterio_data = cloud_11_17_rasterio.read(1)

    with rio.open(cloud_12_17) as cloud_12_17_rasterio:
     cloud_12_17_rasterio_data = cloud_12_17_rasterio.read(1)

except:
    print ("Erro na abertura de algum arquivo!")

```

2018

```
try:
    with rio.open(cloud_1_18) as cloud_1_18_rasterio:
     cloud_1_18_rasterio_data = cloud_1_18_rasterio.read(1)

    with rio.open(cloud_2_18) as cloud_2_18_rasterio:
     cloud_2_18_rasterio_data = cloud_2_18_rasterio.read(1)

    with rio.open(cloud_3_18) as cloud_3_18_rasterio:
     cloud_3_18_rasterio_data = cloud_3_18_rasterio.read(1)

    with rio.open(cloud_4_18) as cloud_4_18_rasterio:
```

```
    cloud_4_18_rasterio_data = cloud_4_18_rasterio.read(1)

  with rio.open(cloud_5_18) as cloud_5_18_rasterio:
    cloud_5_18_rasterio_data = cloud_5_18_rasterio.read(1)

  with rio.open(cloud_6_18) as cloud_6_18_rasterio:
    cloud_6_18_rasterio_data = cloud_6_18_rasterio.read(1)

  with rio.open(cloud_7_18) as cloud_7_18_rasterio:
    cloud_7_18_rasterio_data = cloud_7_18_rasterio.read(1)

  with rio.open(cloud_8_18) as cloud_8_18_rasterio:
    cloud_8_18_rasterio_data = cloud_8_18_rasterio.read(1)

  with rio.open(cloud_9_18) as cloud_9_18_rasterio:
    cloud_9_18_rasterio_data = cloud_9_18_rasterio.read(1)

  with rio.open(cloud_10_18) as cloud_10_18_rasterio:
    cloud_10_18_rasterio_data = cloud_10_18_rasterio.read(1)

  with rio.open(cloud_11_18) as cloud_11_18_rasterio:
    cloud_11_18_rasterio_data = cloud_11_18_rasterio.read(1)

  with rio.open(cloud_12_18) as cloud_12_18_rasterio:
    cloud_12_18_rasterio_data = cloud_12_18_rasterio.read(1)

except:
  print ("Erro na abertura de algum arquivo!")
```

Plotando os resultados do `fmask`

Bibliotecas necessárias

- import matplotlib.pyplot as plt
- from matplotlib.colors import ListedColormap
- import matplotlib.patches as mpatches
- from matplotlib import pyplot

2017

```
# definindo as cores com ListedColormap
cmap=ListedColormap(['white','bisque', 'aqua', 'grey', 'blue'])

# plot
fig, ((ax1, ax2, ax3), (ax4, ax5, ax6), (ax7, ax8, ax9), (ax10, ax11, ax12)) = pyplot.subp
```

```
 6
 7 fmask1 = ax1.imshow(cloud_1_17_rasterio_data, cmap=cmap)
 8 ax1.set_title('Jan/2017',fontname="Times New Roman",fontweight="bold", size=18)
 9 ax1.axis('off')
10
11 fmask2  = ax2.imshow(cloud_2_17_rasterio_data,cmap=cmap)
12 ax2.set_title('Fev/2017',fontname="Times New Roman",fontweight="bold", size=18)
13 ax2.axis('off')
14
15 fmask3 = ax3.imshow(cloud_3_17_rasterio_data, cmap=cmap)
16 ax3.set_title('Mar/2017',fontname="Times New Roman", fontweight="bold", size=18)
17 ax3.axis('off')
18
19 fmask4  = ax4.imshow(cloud_4_17_rasterio_data, cmap=cmap)
20 ax4.set_title('Abr/2017',fontname="Times New Roman",fontweight="bold", size=18)
21 ax4.axis('off')
22
23 fmask5  = ax5.imshow( cloud_5_17_rasterio_data, cmap=cmap)
24 ax5.set_title('Mai/2017',fontname="Times New Roman",fontweight="bold", size=18)
25 ax5.axis('off')
26
27 fmask6 = ax6.imshow(cloud_6_17_rasterio_data, cmap=cmap)
28 ax6.set_title('Jun/2017',fontname="Times New Roman",fontweight="bold", size=18)
29 ax6.axis('off')
30
31 fmask7  = ax7.imshow( cloud_7_17_rasterio_data, cmap=cmap)
32 ax7.set_title('Jul/2017',fontname="Times New Roman",fontweight="bold", size=18)
33 ax7.axis('off')
34
35 fmask8  = ax8.imshow(cloud_8_17_rasterio_data, cmap=cmap)
36 ax8.set_title('Ago/2017',fontname="Times New Roman",fontweight="bold", size=18)
37 ax8.axis('off')
38
39 fmask9 = ax9.imshow(  cloud_9_17_rasterio_data, cmap=cmap)
40 ax9.set_title('Set/2017',fontname="Times New Roman",fontweight="bold", size=18)
41 ax9.axis('off')
42
43 fmask10  = ax10.imshow( cloud_10_17_rasterio_data, cmap=cmap)
44 ax10.set_title('Out/2017',fontname="Times New Roman",fontweight="bold", size=18)
45 ax10.axis('off')
46
47 fmask11  = ax11.imshow(cloud_11_17_rasterio_data, cmap=cmap)
48 ax11.set_title('Nov/2017',fontname="Times New Roman",fontweight="bold", size=18)
49 ax11.axis('off')
50
51 fmask12 = ax12.imshow(cloud_12_17_rasterio_data, cmap=cmap)
52 ax12.set_title('Dez/2017',fontname="Times New Roman",fontweight="bold", size=18)
53 ax12.axis('off')
54
55 # Inserindo legenda personalizada
56 leg1 = mpatches.Patch(color='bisque', label='Área sem nuvem')
```

```
leg2 = mpatches.Patch(color='aqua', label='Nuvem')
leg3 = mpatches.Patch(color='grey', label='Sombra de nuvem')
leg4 = mpatches.Patch(color='blue', label='Água')

plt.legend(handles=[leg1, leg2, leg3,leg4], loc='upper right', bbox_to_anchor=(1.8, 1))

plt.show();

# Salvar imagem
# fig.savefig('fmask2017.png',dpi=300,format='png',orientation='landscape',bbox_inches = '

```

2018

```
# definindo as cores com ListedColormap
cmap=ListedColormap(['white','bisque', 'aqua', 'grey', 'blue'])

# plot
fig, ((ax1, ax2, ax3), (ax4, ax5, ax6), (ax7, ax8, ax9), (ax10, ax11, ax12)) = pyplot.subp

fmask1 = ax1.imshow(cloud_1_18_rasterio_data, cmap=cmap)
ax1.set_title('Jan/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax1.axis('off')

fmask2  = ax2.imshow(cloud_2_18_rasterio_data,cmap=cmap)
ax2.set_title('Fev/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax2.axis('off')

fmask3 = ax3.imshow(cloud_3_18_rasterio_data, cmap=cmap)
ax3.set_title('Mar/2018',fontname="Times New Roman", fontweight="bold", size=18)
ax3.axis('off')

fmask4  = ax4.imshow(cloud_4_18_rasterio_data, cmap=cmap)
ax4.set_title('Abr/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax4.axis('off')

fmask5  = ax5.imshow( cloud_5_18_rasterio_data, cmap=cmap)
ax5.set_title('Mai/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax5.axis('off')

fmask6 = ax6.imshow(cloud_6_18_rasterio_data, cmap=cmap)
ax6.set_title('Jun/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax6.axis('off')

fmask7  = ax7.imshow( cloud_7_18_rasterio_data, cmap=cmap)
ax7.set_title('Jul/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax7.axis('off')

fmask8  = ax8.imshow(cloud_8_18_rasterio_data, cmap=cmap)
```

```
ax8.set_title('Ago/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax8.axis('off')

fmask9 = ax9.imshow(  cloud_9_18_rasterio_data, cmap=cmap)
ax9.set_title('Set/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax9.axis('off')

fmask10  = ax10.imshow( cloud_10_18_rasterio_data, cmap=cmap)
ax10.set_title('Out/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax10.axis('off')

fmask11  = ax11.imshow(cloud_11_18_rasterio_data, cmap=cmap)
ax11.set_title('Nov/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax11.axis('off')

fmask12 = ax12.imshow(cloud_12_18_rasterio_data, cmap=cmap)
ax12.set_title('Dez/2018',fontname="Times New Roman",fontweight="bold", size=18)
ax12.axis('off')


# Inserindo legenda personalizada usando mpatches
leg1 = mpatches.Patch(color='bisque', label='Área sem nuvem')
leg2 = mpatches.Patch(color='aqua', label='Nuvem')
leg3 = mpatches.Patch(color='grey', label='Sombra de nuvem')
leg4 = mpatches.Patch(color='blue', label='Água')

plt.legend(handles=[leg1, leg2, leg3,leg4], loc='upper right', bbox_to_anchor=(1.8, 1))

plt.show();

# Salvar imagem
# fig.savefig('fmask2018.png',dpi=300,format='png',orientation='landscape',bbox_inches = '
```

Extrair valores de um raster

1ª etapa. Bibliotecas necessárias

- import numpy as np
- import rasterio as rio
- from rasterio.plot import plotting_extent
- import earthpy as et
- import earthpy.plot as ep

2ª etapa. Abrir a imagem que deseja trabalhar com `Rasterio`

```
# Nesse caso ainda estamos trabalhando com o arquivo cloud_7_18

with rio.open(cloud_7_18) as cloud_7_18_src:
    # Masked = True sets no data values to np.nan if they are in the metadata
    cloud_7_18_data = cloud_7_18_src.read(1, masked=True)
    cloud_7_18_meta = cloud_7_18_src.profile
```

3ª etapa. Contar os valores de pixels de cada classe

```
# get pixel numbers of each class
classe0 = np.count_nonzero(cloud_7_18_data == 0)
classe1 = np.count_nonzero(cloud_7_18_data == 1)
classe2 = np.count_nonzero(cloud_7_18_data == 2)
classe3 = np.count_nonzero(cloud_7_18_data == 3)
classe4 = np.count_nonzero(cloud_7_18_data == 4)
classe5 = np.count_nonzero(cloud_7_18_data == 5)

# get total pixel numbers of the image - NONZERO
total_nonzero = np.count_nonzero(cloud_7_18_data)

# Exibir o número de pixels de cada classe (Print pixel numbers of each class)
print("Class 0:", classe0)
print("Class 1:", classe1)
print("Class 2:", classe2)
print("Class 3:", classe3)
print("Class 4:", classe4)
print("Class 5:", classe5)

# Exibir o valor total de pixels sem zero (Print Total pixel values without NONZERO)
print("Total(nonzero):", total_nonzero)
```

4ª etapa. Plotar o histograma com o número absoluto de Pixels de cada classe

```
fig = plt.figure()

# Histograma
ax=ep.hist(cloud_7_18_data, figsize=(8,4), colors="blue")

# Título
ax[1].set_title('Pixel number per class - Jul 2018',fontname="Times New Roman",fontweight=

# Personalizando os eixos
plt.xlabel('Classes',fontsize = 15,fontweight="bold")
plt.xticks([1.1,2.1,3.1,4.1,4.9],["No-cloud", "Cloud","Cloud shadow","Snow","Water"],fonts
plt.ylabel('Total of Pixels',fontsize = 15,fontweight="bold")
plt.yticks([5000000,10000000,15000000,20000000,25000000],["5mi", "10mi","15mi","20mi","25m

```

```
15 # Salvar imagem
16 plt.savefig('Pixel_classe.png',dpi=300,format='png',orientation='landscape',bbox_inches =
17
```

Extrair porcentagem de cada classe

Ainda para a imagem *cloud_7_18= "C:/Users/eltes/Desktop/LandSat8/fmask2018/cloud07_18.img"*

Obter os valores de porcetagem

```
 1 class_value = []
 2 class_value.append(classe1)
 3 class_value.append(classe2)
 4 class_value.append(classe3)
 5 class_value.append(classe4)
 6 class_value.append(classe5)
 7
 8 class_value = tuple(class_value)
 9 class_value_per = []
10
11 for i in class_value:
12     class_value_per.append((i/total_nonzero)*100)
13
14 print("Valores por classe:", class_value)
15 print("Porcentagem por classe:", class_value_per)
```

Plotar os valores

```
 1 classes = ["No-cloud", "Cloud","Cloud shadow","Snow","Water"]
 2
 3 fig, ax = plt.subplots(figsize=(10,4))
 4
 5 ax.bar(classes, class_value_per,  width=0.4, color='blue', align='center')
 6
 7 plt.title('Percentage of each class - Jul 2018', fontsize = 20, fontname="Times New Roman'
 8 plt.xlabel('Classes', fontsize = 20, fontname="Times New Roman")
 9
10 plt.yticks( fontsize = 15, fontname="Times New Roman")
11 plt.xticks(fontsize = 15, fontname="Times New Roman")
12 plt.yticks([0,10,20,30,40,50,60],["0","10%","20%","30%","40%","50%","60%"],fontsize = 15)
13
14 plt.show()
15
16 # Salvar imagem
17 fig.savefig('%-Classes.png',dpi=300,format='png',orientation='landscape',bbox_inches = 'ti
```

Lozalizando o pixel no raster (*fmask*)

Bibliotecas necessárias

- `rasterio`
- `earthpy`
- `matplotlib`
- `geopandas`

Abrir o arquivo *shapefile* do centroide do pixel com o `geopandas`

```
centroide=gpd.read_file('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide
```

Abrir o *fmask* com o `rasterio`

```
# importante para colocar no mesmo plano (sobrepor pixel e raster)
from rasterio.plot import plotting_extent
```

Plotando 2017 com o 'pixel'

```
# definindo as cores com ListedColormap

cmap=ListedColormap(['white','bisque', 'aqua', 'grey', 'blue'])

# plot
fig, ((ax1, ax2, ax3), (ax4, ax5, ax6), (ax7, ax8, ax9), (ax10, ax11, ax12)) = pyplot.subp

fmask1 = ax1.imshow(cloud_1_17_rasterio_data, extent=plotting_extent(cloud_1_17_rasterio),
ax1.set_title('Jan/2017',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax1, marker='s', markersize=45, color='none', edgecolor='red')
ax1.axis('off')

fmask2  = ax2.imshow(cloud_2_17_rasterio_data, extent=plotting_extent(cloud_2_17_rasterio)
ax2.set_title('Fev/2017',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax2, marker='s', markersize=45, color='none', edgecolor='red')
ax2.axis('off')

fmask3 = ax3.imshow(cloud_3_17_rasterio_data, extent=plotting_extent(cloud_3_17_rasterio),
ax3.set_title('Mar/2017',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax3, marker='s', markersize=45, color='none', edgecolor='red')
ax3.axis('off')

```

```
fmask4 = ax4.imshow(cloud_4_17_rasterio_data, extent=plotting_extent(cloud_4_17_rasterio),
ax4.set_title('Abr/2017',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax4, marker='s', markersize=45, color='none', edgecolor='red')
ax4.axis('off')

fmask5  = ax5.imshow( cloud_5_17_rasterio_data, extent=plotting_extent(cloud_5_17_rasteric
centroide.plot(ax=ax5, marker='s', markersize=45, color='none', edgecolor='red')
ax5.set_title('Mai/2017',fontname="Times New Roman",fontweight="bold", size=18)
ax5.axis('off')

fmask6 = ax6.imshow(cloud_6_17_rasterio_data, extent=plotting_extent(cloud_6_17_rasterio),
centroide.plot(ax=ax6, marker='s', markersize=45, color='none', edgecolor='red')
ax6.set_title('Jun/2017',fontname="Times New Roman",fontweight="bold", size=18)
ax6.axis('off')

fmask7  = ax7.imshow( cloud_7_17_rasterio_data,extent=plotting_extent(cloud_7_17_rasterio)
centroide.plot(ax=ax7, marker='s', markersize=45, color='none', edgecolor='red')
ax7.set_title('Jul/2017',fontname="Times New Roman",fontweight="bold", size=18)
ax7.axis('off')

fmask8  = ax8.imshow(cloud_8_17_rasterio_data, extent=plotting_extent(cloud_8_17_rasterio)
centroide.plot(ax=ax8, marker='s', markersize=45, color='none', edgecolor='red')
ax8.set_title('Ago/2017',fontname="Times New Roman",fontweight="bold", size=18)
ax8.axis('off')

fmask9 = ax9.imshow(  cloud_9_17_rasterio_data, extent=plotting_extent(cloud_9_17_rasteric
centroide.plot(ax=ax9, marker='s', markersize=45, color='none', edgecolor='red')
ax9.set_title('Set/2017',fontname="Times New Roman",fontweight="bold", size=18)
ax9.axis('off')

fmask10  = ax10.imshow( cloud_10_17_rasterio_data, extent=plotting_extent(cloud_10_17_rast
ax10.set_title('Out/2017',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax10, marker='s', markersize=45, color='none', edgecolor='red')
ax10.axis('off')

fmask11  = ax11.imshow(cloud_11_17_rasterio_data, extent=plotting_extent(cloud_11_17_raste
centroide.plot(ax=ax11, marker='s', markersize=45, color='none', edgecolor='red')
ax11.set_title('Nov/2017',fontname="Times New Roman",fontweight="bold", size=18)
ax11.axis('off')

fmask12 = ax12.imshow(cloud_12_17_rasterio_data, extent=plotting_extent(cloud_12_17_raster
centroide.plot(ax=ax12, marker='s', markersize=45, color='none', edgecolor='red')
ax12.set_title('Dez/2017',fontname="Times New Roman",fontweight="bold", size=18)
ax12.axis('off')


# Inserindo legenda personalizada usando mpatches
leg1 = mpatches.Patch(color='bisque', label='Área sem nuvem')
leg2 = mpatches.Patch(color='aqua', label='Nuvem')
leg3 = mpatches.Patch(color='grey', label='Sombra de nuvem')
leg4 = mpatches.Patch(color='blue', label='Água')
```

```
74
75 plt.legend(handles=[leg1, leg2, leg3,leg4], loc='upper right', bbox_to_anchor=(1.8, 1))
76
77 plt.show();
78
79 # Salvar imagem
80 # fig.savefig('fmask2017_centroide.png',dpi=300,format='png',orientation='landscape',bbox_
```

Plotando 2018 com o 'pixel'

```
 1 # definindo as cores com ListedColormap
 2 cmap=ListedColormap(['white','bisque', 'aqua', 'grey', 'blue'])
 3
 4 # plot
 5 fig, ((ax1, ax2, ax3), (ax4, ax5, ax6), (ax7, ax8, ax9), (ax10, ax11, ax12)) = pyplot.subp
 6
 7 fmask1 = ax1.imshow(cloud_1_18_rasterio_data,extent=plotting_extent(cloud_1_18_rasterio),
 8 ax1.set_title('Jan/2018',fontname="Times New Roman",fontweight="bold", size=18)
 9 centroide.plot(ax=ax1, marker='s', markersize=45, color='none', edgecolor='red')
10 ax1.axis('off')
11
12 fmask2  = ax2.imshow(cloud_2_18_rasterio_data,extent=plotting_extent(cloud_2_18_rasterio),
13 ax2.set_title('Fev/2018',fontname="Times New Roman",fontweight="bold", size=18)
14 centroide.plot(ax=ax2, marker='s', markersize=45, color='none', edgecolor='red')
15 ax2.axis('off')
16
17 fmask3 = ax3.imshow(cloud_3_18_rasterio_data,extent=plotting_extent(cloud_3_18_rasterio),
18 ax3.set_title('Mar/2018',fontname="Times New Roman", fontweight="bold", size=18)
19 centroide.plot(ax=ax3, marker='s', markersize=45, color='none', edgecolor='red')
20 ax3.axis('off')
21
22 fmask4  = ax4.imshow(cloud_4_18_rasterio_data,  extent=plotting_extent(cloud_4_18_rasteric
23 ax4.set_title('Abr/2018',fontname="Times New Roman",fontweight="bold", size=18)
24 centroide.plot(ax=ax4, marker='s', markersize=45, color='none', edgecolor='red')
25 ax4.axis('off')
26
27 fmask5  = ax5.imshow( cloud_5_18_rasterio_data, extent=plotting_extent(cloud_5_18_rasteric
28 ax5.set_title('Mai/2018',fontname="Times New Roman",fontweight="bold", size=18)
29 centroide.plot(ax=ax5, marker='s', markersize=45, color='none', edgecolor='red')
30 ax5.axis('off')
31
32 fmask6 = ax6.imshow(cloud_6_18_rasterio_data, extent=plotting_extent(cloud_6_18_rasterio),
33 ax6.set_title('Jun/2018',fontname="Times New Roman",fontweight="bold", size=18)
34 centroide.plot(ax=ax6, marker='s', markersize=45, color='none', edgecolor='red')
35 ax6.axis('off')
36
37 fmask7  = ax7.imshow( cloud_7_18_rasterio_data, extent=plotting_extent(cloud_7_18_rasteric
38 ax7.set_title('Jul/2018',fontname="Times New Roman",fontweight="bold", size=18)
39 centroide.plot(ax=ax7, marker='s', markersize=45, color='none', edgecolor='red')
```

```
ax7.axis('off')

fmask8  = ax8.imshow(cloud_8_18_rasterio_data, extent=plotting_extent(cloud_8_18_rasterio)
ax8.set_title('Ago/2018',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax8, marker='s', markersize=45, color='none', edgecolor='red')
ax8.axis('off')

fmask9 = ax9.imshow(cloud_9_18_rasterio_data,extent=plotting_extent(cloud_9_18_rasterio),
ax9.set_title('Set/2018',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax9, marker='s', markersize=45, color='none', edgecolor='red')
ax9.axis('off')

fmask10  = ax10.imshow( cloud_10_18_rasterio_data, extent=plotting_extent(cloud_10_18_rast
ax10.set_title('Out/2018',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax10, marker='s', markersize=45, color='none', edgecolor='red')
ax10.axis('off')

fmask11  = ax11.imshow(cloud_11_18_rasterio_data, extent=plotting_extent(cloud_11_18_raste
ax11.set_title('Nov/2018',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax11, marker='s', markersize=45, color='none', edgecolor='red')
ax11.axis('off')

fmask12 = ax12.imshow(cloud_12_18_rasterio_data,extent=plotting_extent(cloud_12_18_rasteri
ax12.set_title('Dez/2018',fontname="Times New Roman",fontweight="bold", size=18)
centroide.plot(ax=ax12, marker='s', markersize=45, color='none', edgecolor='red')
ax12.axis('off')

# Inserindo legenda personalizada usando mpatches
leg1 = mpatches.Patch(color='bisque', label='Área sem nuvem')
leg2 = mpatches.Patch(color='aqua', label='Nuvem')
leg3 = mpatches.Patch(color='grey', label='Sombra de nuvem')
leg4 = mpatches.Patch(color='blue', label='Água')

plt.legend(handles=[leg1, leg2, leg3,leg4], loc='upper right', bbox_to_anchor=(1.8, 1))

plt.show();

# Salvar imagem
fig.savefig('fmask2018_centroide.png',dpi=300,format='png',orientation='landscape',bbox_ir
```

Extraindo o valor (classe) **do mesmo pixel** ao longo do tempo

Bibliotecas necessárias

- `rasterstats`
- `matplotlib`

Extraindo o valor do pixel (*fmask*) para o ponto (centroide)

OBSERÇÃO: a função `point_query` do `rasterstats` pede um arquivo `.shp` e uma `imagem`, respectivamente, nessa ordem!

```
# 2017
JAN17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
FEV17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
MAR17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
ABR17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
MAI17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
JUN17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
JUL17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
AGO17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
SET17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
OUT17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
NOV17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
DEZ17= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›

# 2018
JAN18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
FEV18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
MAR18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
ABR18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
MAI18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
JUN18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
JUL18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
AGO18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
SET18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
OUT18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
NOV18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›
DEZ18= point_query('/content/drive/MyDrive/Artigo_Nuvens/LandSat8/centroide/centroide1_pi›

print(JAN17,FEV17,MAR17,ABR17,MAI17,JUN17,JUL17,AGO17,SET17,OUT17,NOV17,DEZ17,JAN18,FEV18,
```

```
[2.0] [1.0] [1.0] [2.0] [2.0] [1.0] [2.0] [2.0] [2.0] [1.0] [1.0] [2.0] [2.0] [2.0] [2.
```

Unindo as listas (valores do pixel de cada ano) geradas com `point_query`

```
lista = JAN17 + FEV17 + MAR17 + ABR17 + MAI17 + JUN17 + JUL17 + AGO17 + SET17 + OUT17 + NC
print(lista)
```

```
[2.0, 1.0, 1.0, 2.0, 2.0, 1.0, 2.0, 2.0, 2.0, 1.0, 1.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2
```

Plotando o gráfico

```
# Elaborando uma estrutura de dados Dataframe
meses = ["Jan/2017", "Fev/2017","Mar/2017","Abr/2017","Mai/2017","Jun/2017","Jul/2017","Ag

pixel_value = lista

dados_pixel= {'meses':meses,
    'Value':pixel_value}

dados_pixel_df = pd.DataFrame(data=dados_pixel)
```

```
dados_pixel_df
```

	meses	Value
0	Jan/2017	2.0
1	Fev/2017	1.0
2	Mar/2017	1.0
3	Abr/2017	2.0

```
# definindo as cores para plotar cada valor(classe) com uma cor difente

colors = {1.0:'seagreen', 2.0:'royalblue'}
c = dados_pixel_df['Value'].apply(lambda x: colors[x]) # operando em cada linha


#Plotando o gráfico
fig, ax = plt.subplots(figsize=(50,15))

bars = ax.bar(dados_pixel_df['meses'], dados_pixel_df['Value'], align='center', width=0.4,
plt.title('Pixel variation between 2017 and 2018', fontsize = 50, fontname="Times New Roma


# Inserindo anotações no gráfico como as estações
seca1=ax.annotate("Dry Season",fontsize = 30, fontname="Times New Roman",
            xy=(0.20, 0.15), xycoords='axes fraction',
            xytext=(0.36, 0.15), textcoords='axes fraction',
            arrowprops=dict(arrowstyle="<|-|>, head_length=0.4, head_width=0.4", facecolor

seca2=ax.annotate("Dry Season",fontsize = 30, fontname="Times New Roman",
            xy=(2.17, 0.15), xycoords=seca1,
            xytext=(0.83, 0.14), textcoords='axes fraction',
            arrowprops=dict(arrowstyle="<|-|>, head_length=0.4,head_width=0.4", facecolor=


# colocando informações labels x e y
plt.xlabel('Temporal series', fontsize = 50, fontname="Times New Roman")

plt.ylabel('Classes', fontsize = 50, fontname="Times New Roman")


plt.yticks([])  # remove o eixo y
plt.xticks(rotation = (45), fontsize = 24, ha='right', fontname="Times New Roman")

# Inserindo legenda personalizada usando a biblioteca mpatches
leg1 = mpatches.Patch(color="seagreen", label='No-cloud')
leg2 = mpatches.Patch(color="royalblue", label='Cloud')


plt.legend(handles=[leg1, leg2], loc='upper right', bbox_to_anchor=(1.1, 0.5), fontsize=20
#plt.show();


```

```
# Salvar imagem
fig.savefig('mudança-pixel_2017-2018.png',dpi=300,format='png',orientation='landscape')
```

```
findfont: Font family ['Times New Roman'] not found. Falling back to DejaVu Sans.
findfont: Font family ['Times New Roman'] not found. Falling back to DejaVu Sans.
findfont: Font family ['Times New Roman'] not found. Falling back to DejaVu Sans.
```

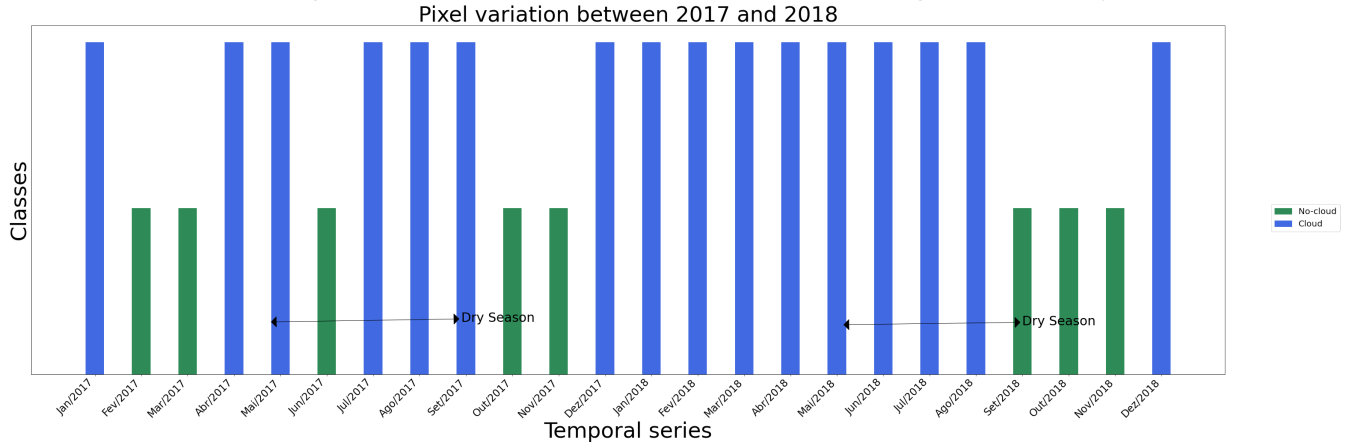
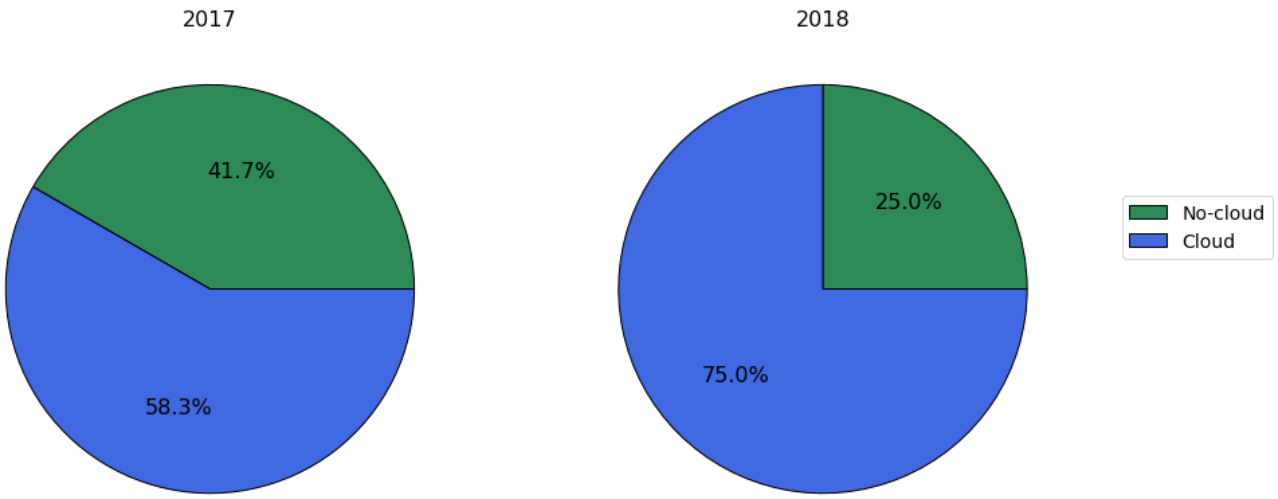


Gráfico de pizza - comportamento do pixel no tempo

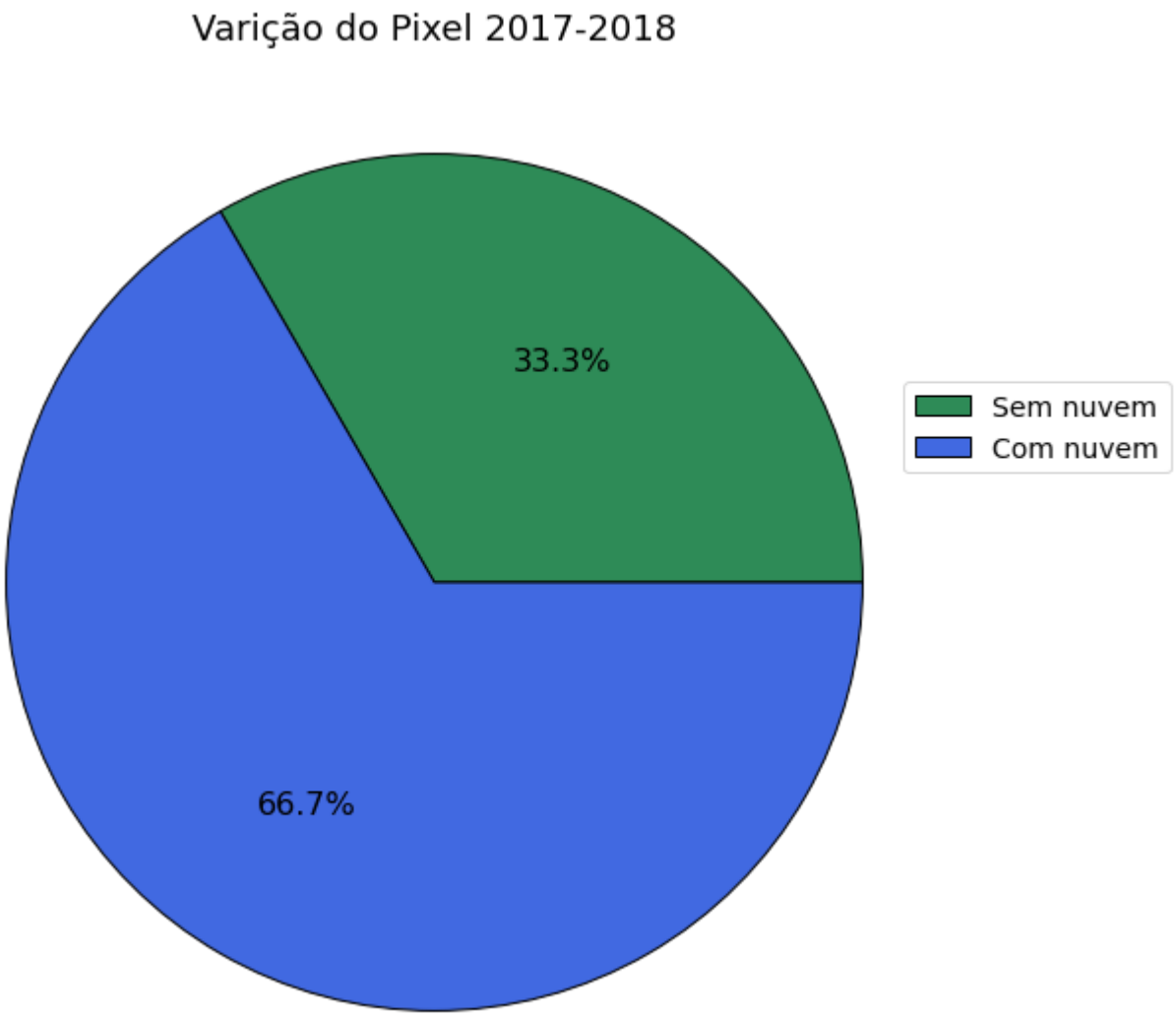
```
#2017 e 2018 separados
lista_17 = JAN17 + FEV17 + MAR17 + ABR17 + MAI17 + JUN17 + JUL17 + AGO17 + SET17 + OUT17 +

lista_18 = JAN18 + FEV18 + MAR18 + ABR18 + MAI18 + JUN18 + JUL18 + AGO18 + SET18 + OUT18 +

list_2017 = [(lista_17.count(1)/len(lista_17)*100)] + [(lista_17.count(2)/len(lista_17)*10
list_2018 = [(lista_18.count(1)/len(lista_18)*100)] + [(lista_18.count(2)/len(lista_18)*10
classe = "No-cloud", "Cloud"

fig = plt.figure( figsize=(15, 11))
plt.subplot(1, 2, 1)
plt.title(2017, fontsize = 16)
plt.pie(list_2017,
        autopct = '%1.1f%%',
        colors = [ "seagreen", "royalblue"],
        explode = [0,0],
        wedgeprops = {"ec": "k"},
        textprops = {"fontsize": 16},
        )
```

```
20
21 plt.subplot(1, 2, 2)
22 plt.title(2018, fontsize = 16)
23
24 plt.pie(list_2018,
25         autopct = '%1.1f%%',
26         colors = [ "seagreen", "royalblue"],
27         wedgeprops = {"ec": "k"},
28         textprops = {"fontsize": 16},
29         )
30 plt.legend(labels = classe, loc='upper right', bbox_to_anchor=(1.4, 0.7), fontsize = 14);
31
32
33 # Salvar imagem
34 fig.savefig('pixel_cada_ano.png',dpi=300,format='png',orientation='landscape',bbox_inches
```




```
 1 # para todo o período
 2 lista_geral = JAN17 + FEV17 + MAR17 + ABR17 + MAI17 + JUN17 + JUL17 + AGO17 + SET17 + OUT1
 3
 4 list_geral = [(lista_geral.count(1)/len(lista_geral)*100)] + [(lista_geral.count(2)/len(li
 5
 6 classe = "Sem nuvem", "Com nuvem"
 7
 8 plt.figure( figsize=(12, 10))
 9 plt.title("Varição do Pixel 2017-2018", fontsize = 18, va = "center_baseline")
10
11 plt.pie(list_geral,
12         autopct = '%1.1f%%',
```

```
13         colors = [ "seagreen", "royalblue"],
14         labeldistance = 1.05,
15         wedgeprops = {"ec": "k"},
16         textprops = {"fontsize": 16},
17         )
18
19 plt.legend(labels = classe, loc='upper right', bbox_to_anchor=(1.2, 0.7), fontsize = 14);
20
21 # Salvar imagem
22 # plt.savefig('pixel_2017-2018.png',dpi=300,format='png',orientation='landscape', bbox_inc
```



Considerações Finais

- `fmask` diminui o tamanho da imagem (número de linhas e colunas)
- Processamento do algoritmo relativamente simples

- Dificuldade de adquirir imagens sem nuvens na Amazônia