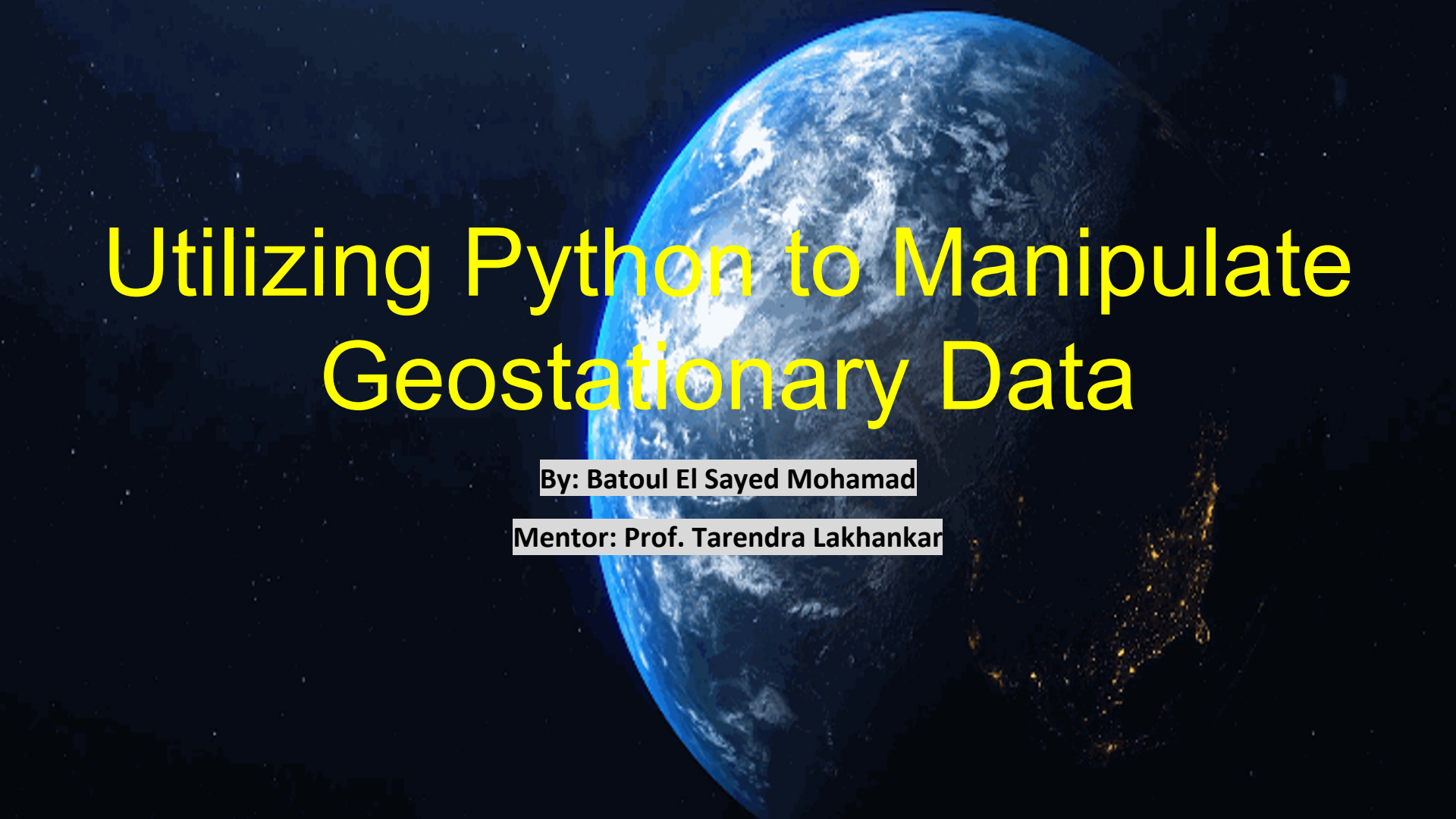
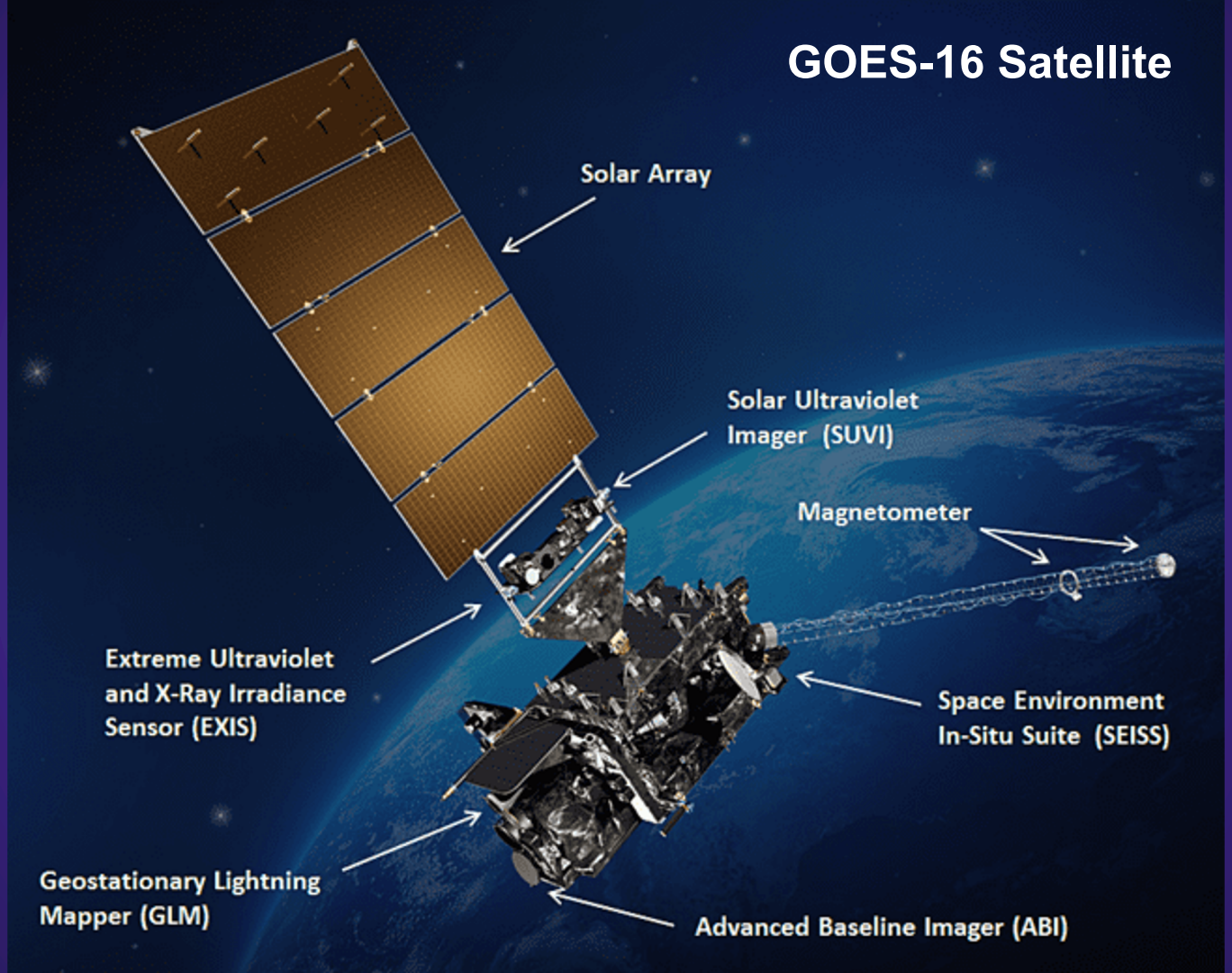
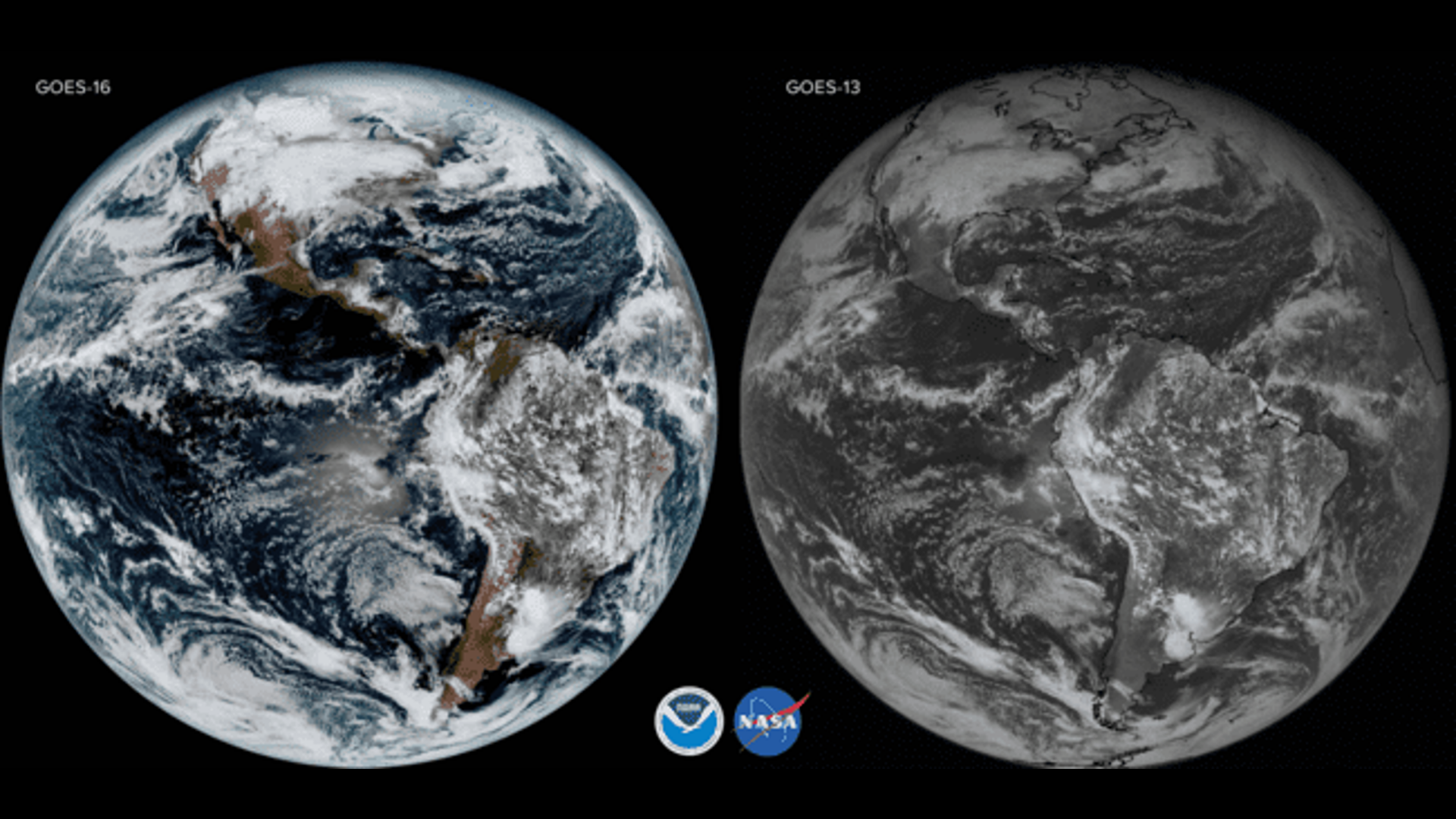


Utilizing Python to Manipulate Geostationary Data

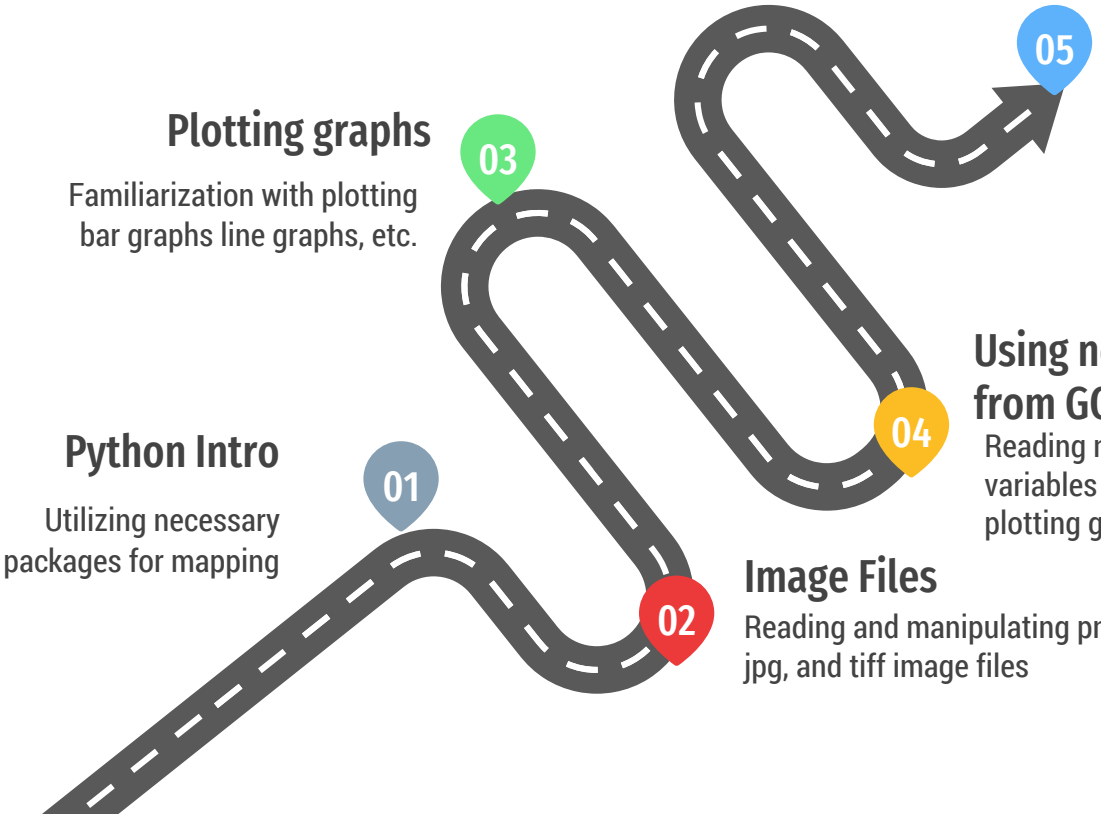
By: Batoul El Sayed Mohamad

Mentor: Prof. Tarendra Lakhankar


GOES-16 Satellite
Solar Array
Solar Ultraviolet Imager (SUVI)
Magnetometer
Extreme Ultraviolet and X-Ray Irradiance Sensor (EXIS)
Space Environment In-Situ Suite (SEISS)
Geostationary Lightning Mapper (GLM)
Advanced Baseline Imager (ABI)


GOES-16
GOES-13
NASA

Roadmap



Python Intro

Utilizing necessary packages for mapping

Image Files

Reading and manipulating png, jpg, and tiff image files

Plotting graphs

Familiarization with plotting bar graphs line graphs, etc.

Using netCDF files from GOES R-16

Reading netCDF files with variables like radiance and plotting geostationary map

Creating NYC map for Precipitation

Plotting Choropleth map as well as a bar graph

Reading Image Files, and Brightening them

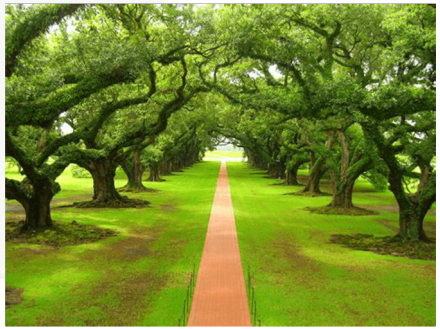
In [2]:
```
f = open ('earth.jpg', 'rb')

print (f.readlines())

for line in f:
    print (line)
```

```
[b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00\x00\x01\x00\x01\x00\x00\xff\xdb\x00C\x00\x05\x03\x04\x04\x0
4\x03\x05\x04\x04\x04\x05\x05\x05\x06\x07\x0c\x08\x07\x07\x07\x0f\x0b\x0b\t\x0c\x11\x0f\x12\x12\x11\x0f\
x11\x11\x13\x16\x1c\x17\x13\x14\x1a\x15\x11\x11\x18!\x18\x1a\x1d\x1d\x1f\x1f\x1f\x13\x17"$"\x1e$\x1c\x1e\x1
f\x1e\xff\xdb\x00C\x01\x05\x05\x05\x07\x06\x07\x0e\x08\x08\x0e\x1e\x14\x11\x14\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x
1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x
1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\x1e\xff\xc0\x00\x11\x08\x05\xdc\x05\xdc\x03\x01"\
x00\x02\x11\x01\x03\x11\x01\xff\xc4\x00\x1d\x00\x00\x01\x05\x01\x01\x01\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x02\x03\x04\x05\x06\x01\x07\x08\t\xff\xc4\x00B\x10\x00\x02\x01\x03\x03\x02\x04\x03\x05\x06\x05\x03\x
02\x02\x00\x17\x01\x02\x03\x00\x04\x11\x05\x12!1A\x06\x13Qa"q\x81\x07\x142\x91\xa1#B\xb1\xc1\xd1\xf0\x08\x15
R\xe1\xf1$3br\x82\x16C8\x92\xa24\xb2\xc2%Tcs\x83\x93\x17\x186Dd\xa3\xb3\xd2t\x94\xc3\xff\xc4\x00\x1b\x01\x0
1\x00\x03\x01\x01\x01\x01\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x02\x03\x04\x05\x06\x07\xff\xc4\x005\x
11\x00\x02\x02\x02\x02\x02\x01\x03\x03\x02\x04\x07\x01\x01\x01\x01\x00\x01\x02\x11\x03!\x121\x04AQ\x05\x13"2
aq\x06\x81\x14\x91\xa1\xb1#3B\xc1\xd1\xe1\xf0\xf1R\x15$\xff\xda\x00\x0c\x03\x01\x00\x02\x11\x03\x11\x00?\x0
0\xfb*\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x0
2\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a
(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\
n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b
'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa
2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa2\x80(\xa2\x8a\x00\xa2\x8a(\x02\x8a(\xa0\n', b'(\xa2\x80
```

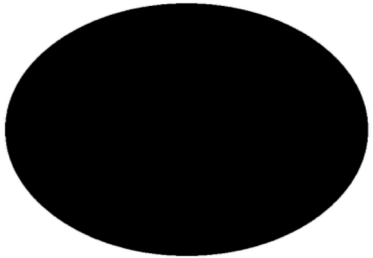
In [3]:
```
f = open ('earth.jpg', 'rb')

newfilew = open ('newfilew.jpg' , 'wb')

for line in f:
    newfilew.write(line)
```

In [3]:
```
from PIL import Image,ImageEnhance
img=Image.open("nature.jpg")
img_brightness_obj=ImageEnhance.Brightness(img)
factor=int(input())
enhanced_img=img_brightness_obj.enhance(factor)
enhanced_img.show()

enhanced_img.save("nature_bright.jpg")

#factor > 1  Brightness of Image increases according to given factor

#factor < 1  Brightness of Image decreases according to given factor
```

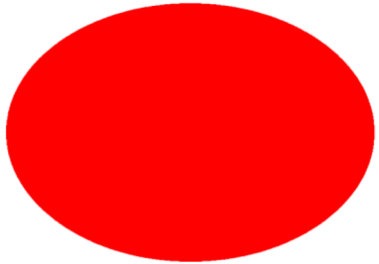
```
2
```



Creating a Color Detector

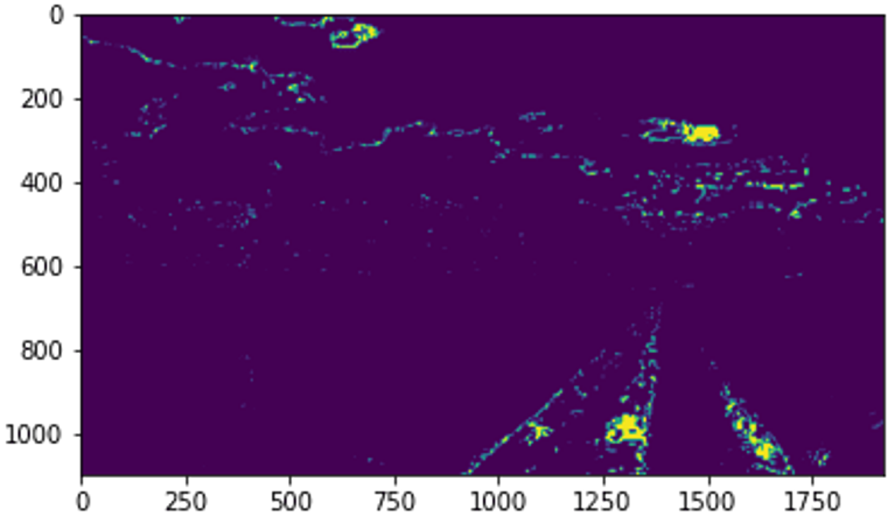


RGB color code: (0, 0, 0)



RGB color code: (255, 0, 0)

```
import cv2
import numpy as np
from PIL import Image

image=cv2.imread("oval.png")
image[np.where((image==[0, 0, 0]).all(axis=2))]=[0,0,255]
cv2.imwrite('oval2.png',image)
```



PALETTE:

Colors

HEX: #cbbdac

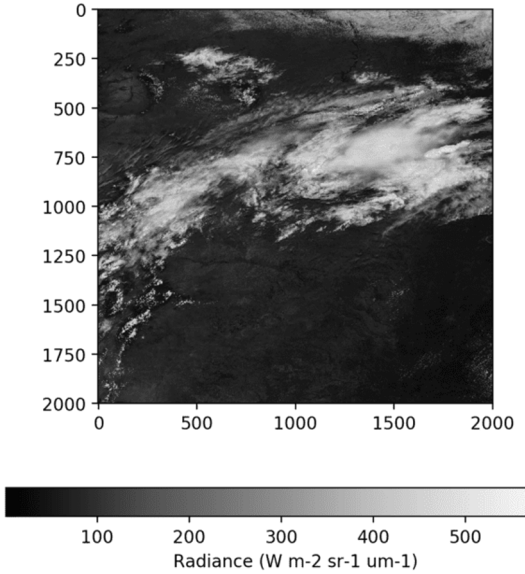
RGB: rgba(203,189,172

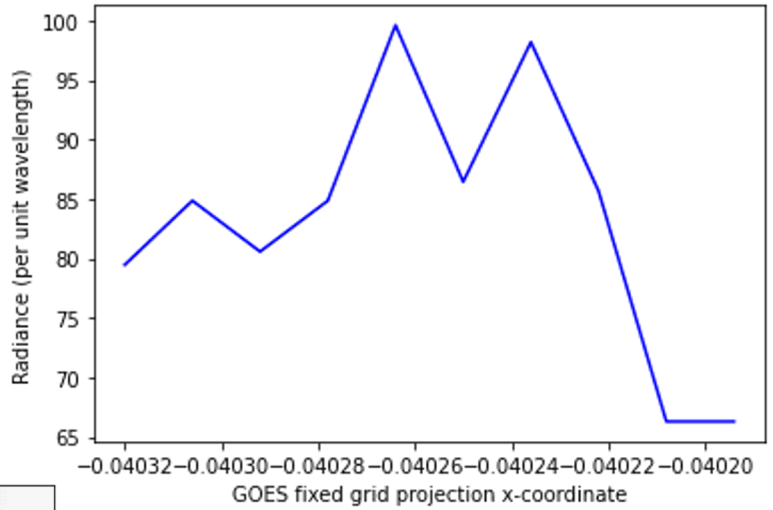
Show more

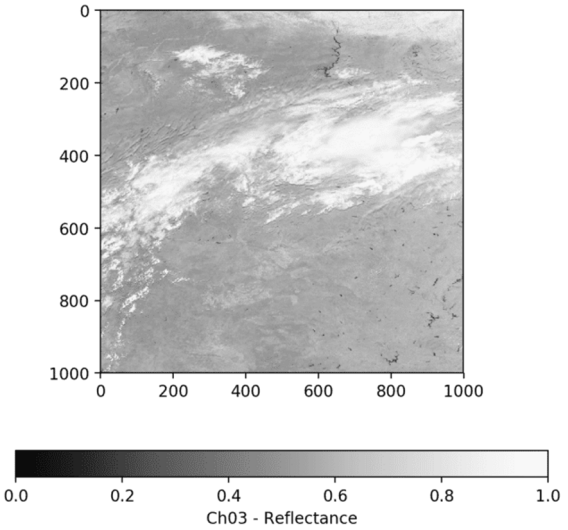
Use Your Image

We think data protection is important! `No data is sent.` The magic happens in your browser.

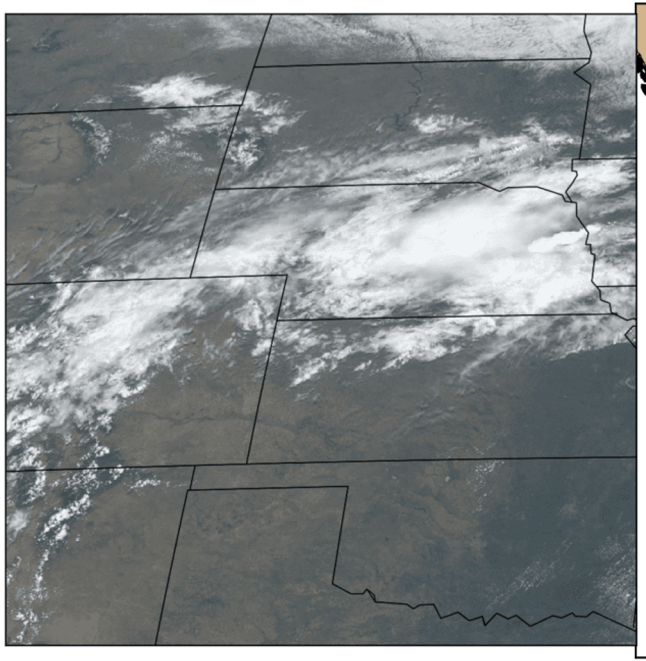
GOES-16 Maps

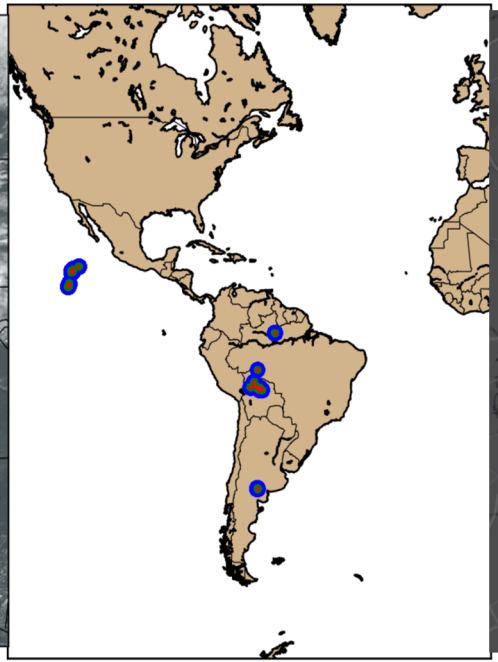

0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000
100
200
300
400
500
600
Radiance (W m-2 sr-1 um-1)

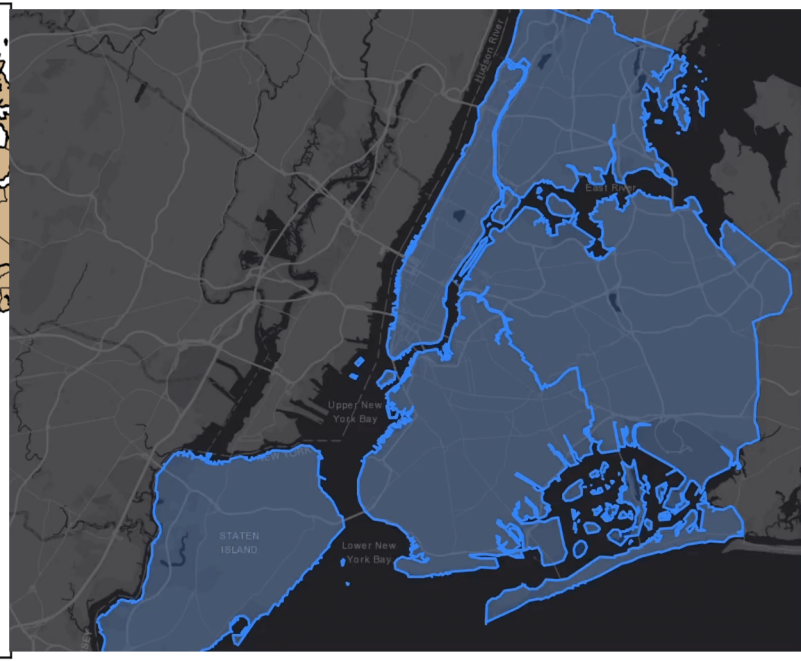

Radiance (per unit wavelength)
100
95
90
85
80
75
70
65
−0.04032
−0.04030
−0.04028
−0.04026
−0.04024
−0.04022
−0.04020
GOES fixed grid projection x-coordinate


0
200
400
600
800
1000
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0
Ch03 - Reflectance

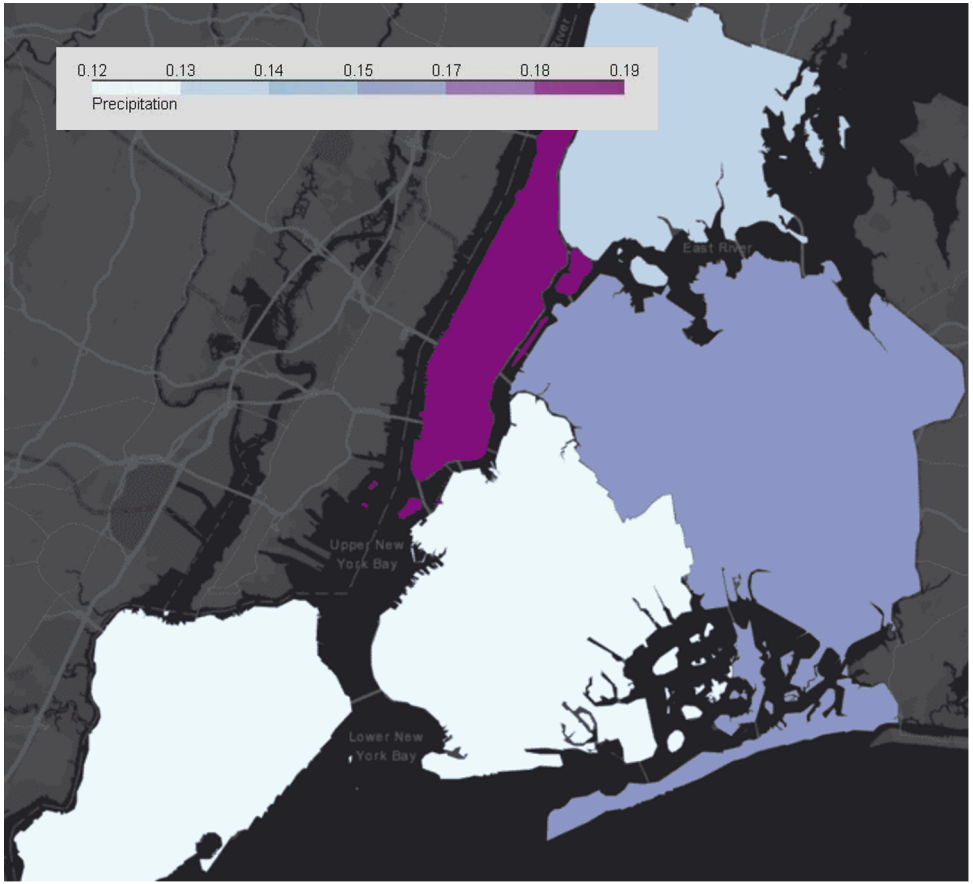
GOES-16 maps and NYC GeoJSON

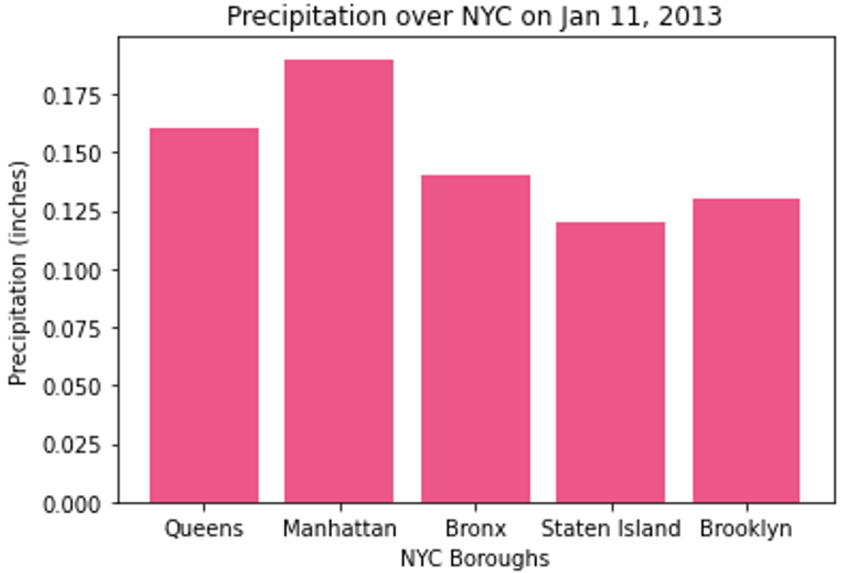






NYC Precipitation Map





Acknowledgments