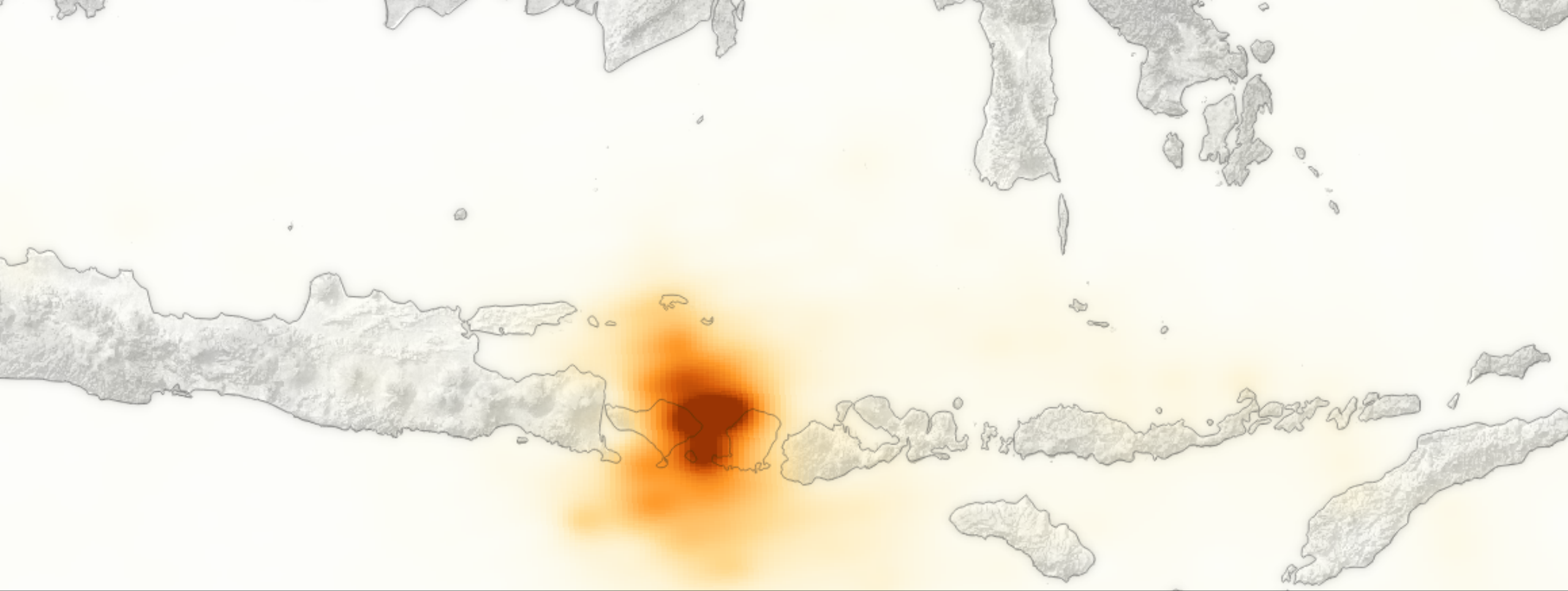


Lea, Mapee y Extraiga Datos de Aerosoles de MODIS Usando Scripts de Python

Herramientas para el Análisis de Conjuntos de Datos Satelitales de la Calidad del Aire de Alta Resolución

Pawan Gupta y Melanie Follette-Cook, 17 al 22 de enero de 2018





Requisitos para su Computadora y Python

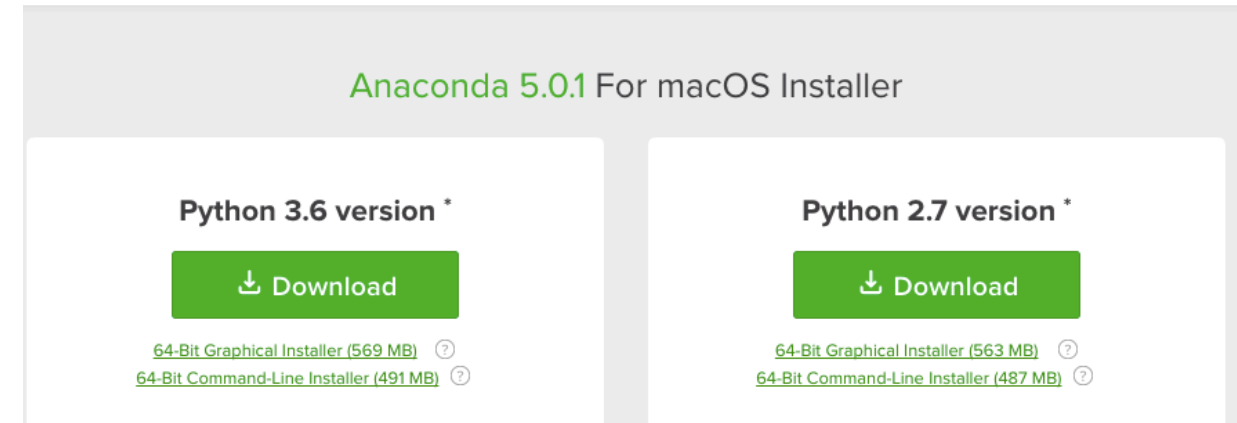
Sistema Operativo

- Tanto la Sesión 2 como la Sesión 3 cubrirán scripts de Python para realizar tareas similares.
- La Sesión 2 se presentará usando un sistema operativo Mac (Apple).
- Toda la estructura de los directorios y el diseño de Python serán según el sistema operativo para Mac.
- La Sesión 3 se presentará usando una máquina Windows.
- El mostrar usando ambos sistemas operativos permitirá a los usuarios experimentar el diseño de cada uno.
- Anticipamos que los códigos se ejecuten exitosamente en los sistemas operativos de Mac, Windows y Linux.



Requisitos Computacionales

- Instalar Python 2.7 usando [Anaconda](#)
- Instalar todos los paquetes (packages) de Python requeridos
 - Lista de paquetes (derecha)
- Probar la instalación de Python y los paquetes usando el siguiente código de prueba para Python
 - test_python.py
- Descargar los datos de MODIS Data y códigos de Python de la página en línea de la capacitación
 - <https://arset.gsfc.nasa.gov/airquality/webinars/2018-hiresdatasets>
- Para más detalles sobre el código, visite: <https://arset.gsfc.nasa.gov/airquality/python-scripts-aerosol-data-sets-merra-modis-and-omi>



- Lista de paquetes de Python:
 - pyhdf
 - numpy
 - sys
 - mpl_toolkits.
basemap
 - matplotlib
 - linearSegmented
Colormap
 - h5py
 - time
 - calendar



Prueba de Python

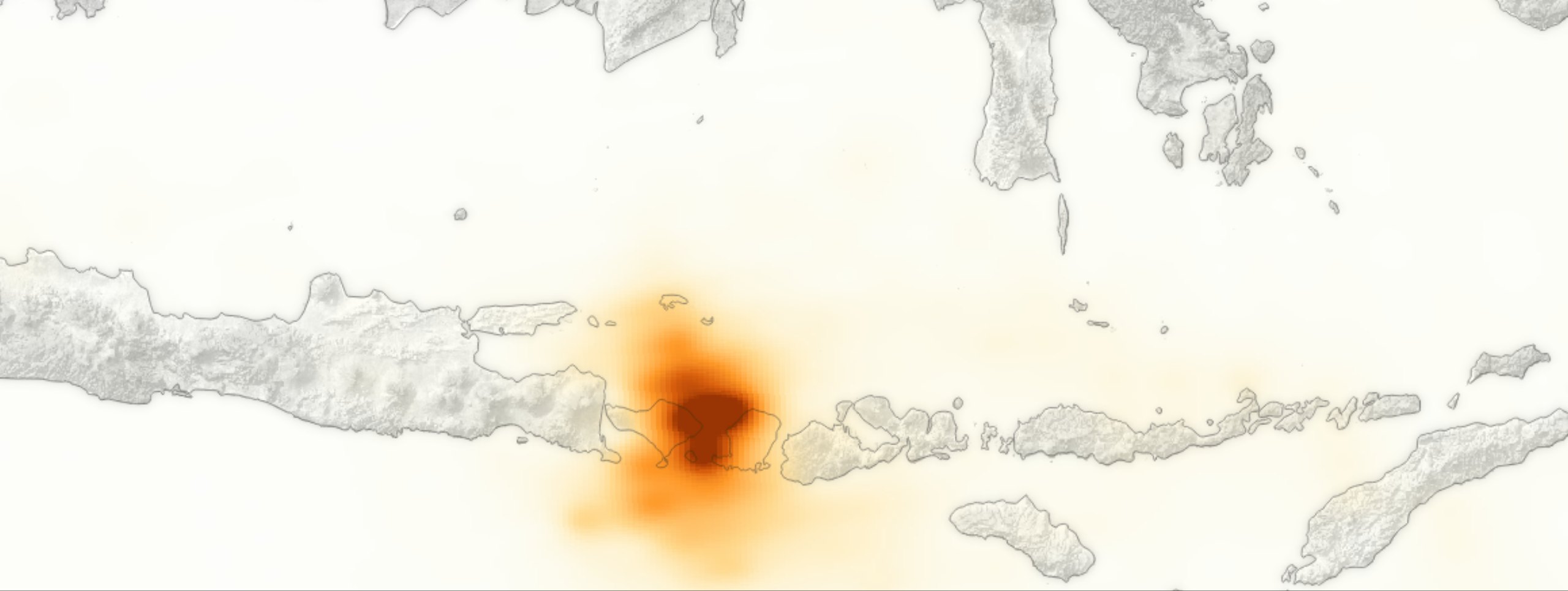
- Abra el editor “spyder” dentro de Anaconda
- Abra **test_python.py**
- Asegúrese que el directorio tenga el código Python y el archivo HDF
- Abra la consola **ipython** en el spider
- Ejecute el código usando la **flecha verde** en la parte superior
- Debe producirse una imagen como la que se muestra aquí

The screenshot displays the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The main editor window shows a Python script named `test_python.py` with the following content:

```
1#!/usr/bin/python
2'''
3Module: read_and_map_mod_aerosol.py
4=====
5Disclaimer: The code is for demonstration purposes only. Users are responsible to check for acc
6
7Author: Justin Roberts-Pierel, 2015
8Organization: NASA ARSET
9Purpose: To extract AOD data from a MODIS HDF4 file (or series of files) and create a map of th
10
11See the README associated with this module for more information.
12=====
13'''
14
15#Import necessary modules
16from pyhdf import SD
17import numpy as np
18from mpl_toolkits.basemap import Basemap, cm
19import matplotlib.pyplot as plt
20import sys
21import h5py
22import time
23import calendar
24
25
26 FILE_NAME='MYD04_L2.A2017249.2105.006.2017250160535.hdf'
27
28 hdf=SD(FILE_NAME)
29 # Get lat and lon info
30 lat = hdf.select('Latitude')
31 latitude = lat[:]
32 min_lat=latitude.min()
33 max_lat=latitude.max()
34 lon = hdf.select('Longitude')
35 longitude = lon[:]
36 min_lon=longitude.min()
37 max_lon=longitude.max()
38 SDS_NAME='Image_Optical_Depth_Land_And_Ocean'
39 sds=hdf.select(SDS_NAME)
40 #get scale factor for AOD SDS
41 attributes=sds.attributes()
42 scale_factor=attributes['scale_factor']
43 #get valid range for AOD SDS
44 range=sds.get_range()
45 min_range=min(range)
46 max_range=max(range)
47
48 #get SDS data
49 data=sds.get()
50 #get data within valid range
51 valid_data=data.ravel()
52 valid_data=[x for x in valid_data if x>=min_range]
53 valid_data=[x for x in valid_data if x<=max_range]
```

A red circle highlights the green play button in the toolbar, and a red box highlights line 26 of the code. A red arrow points from the text "archivo HDF" to the highlighted line. The IPython console on the right shows the execution output, including a map of the Pacific Northwest region. The map is titled "MYD04_L2.A2017249.2105.006.2017250160535 Image_Optical_Depth_Land_And_Ocean" and displays AOD data with a color scale from 0 to 5. A red arrow points from the text "imagen producida" to the map.





Conozca Sus Datos

Cómo Entender el Nombre de un Archivo de MODIS

Producto de Aerosoles, Nivel 2, 10 km

Nombre del Producto

- Terra: MOD04
- Aqua: MYD04



Se puede usar HDFLook, Panoply, IDL, Python, Fortran, MatLab y otros para leer los datos

Cómo Entender el Nombre de un Archivo de MODIS

Producto de Aerosoles, Nivel 2, 3 km

Nombre del Producto

- Terra: MOD04
- Aqua: MYD04

MOD04_3K.A2001079.0255.006.2006289012028.hdf

Hora

Información de
procesamiento del archivo

Fecha

- Año
- Día Juliano

Colección

Se puede usar HDFLook, Panoply, IDL, Python, Fortran, MatLab y otros para leer los datos



Parámetros de Aerosoles de MODIS (SDS)

- Optical_Depth_Land_and_Ocean
 - Se recupera usando el algoritmo de objetivo oscuro (Dark Target Algorithm)
 - Sólo datos de alta calidad
 - QA (garantía de calidad) = 3
 - QA = 1, 2, 3
 - 10 km y 3km
- Dark_Target_Deep_Blue_Optical_Depth_550_Combined
 - Producto combinado de los algoritmos Deep Blue (azul profundo) y Dark Target (objetivo oscuro)
 - Sólo de 10 km
- Quality_Assurance_Land
 - Indicación de calidad asociada con producto DD



La Garantía de Calidad (Quality Assurance) es Sumamente Importante

La Garantía de Calidad (QA) indica la confianza en la calidad de la imagen recuperada

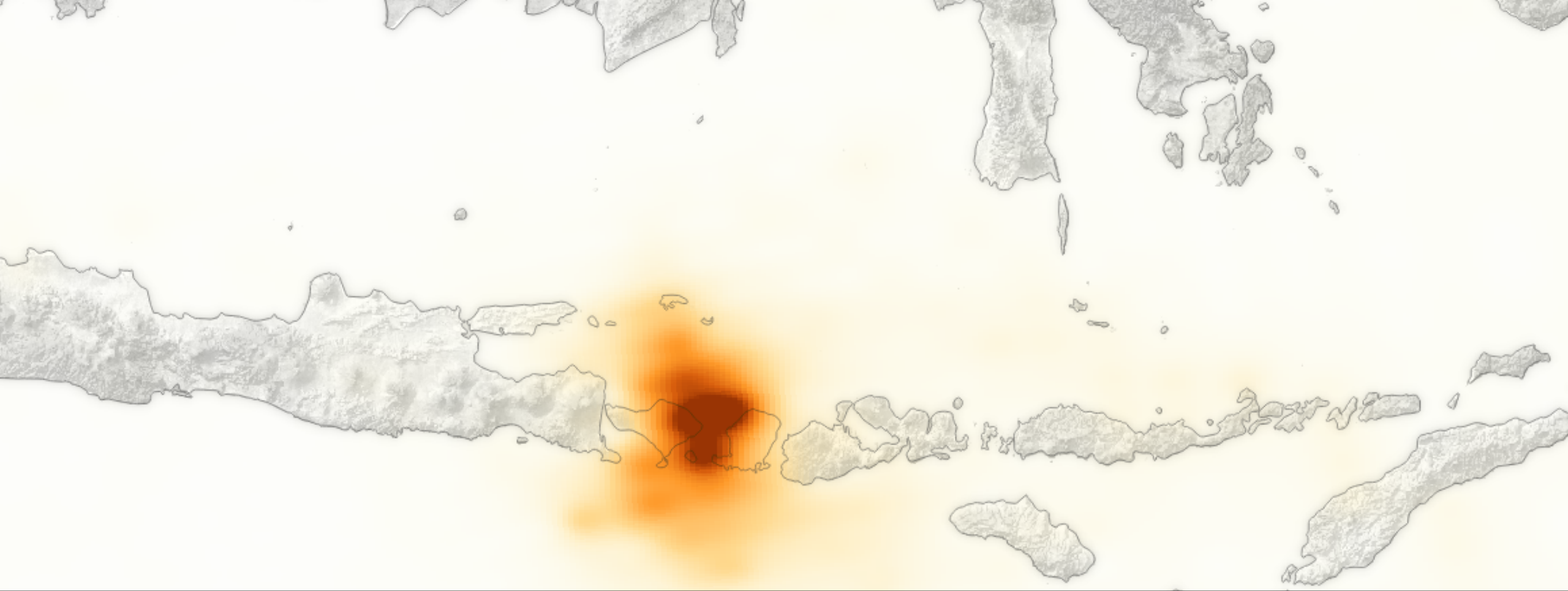
Quality_Assurance_Ocean

- Escala de 0 a 3
- Se Recomienda QA por encima de 1,2,3 para Océano
- Factores:
 - Número de píxeles
 - Adecuación de errores
 - **Proximidad al destello**

Quality_Assurance_Land

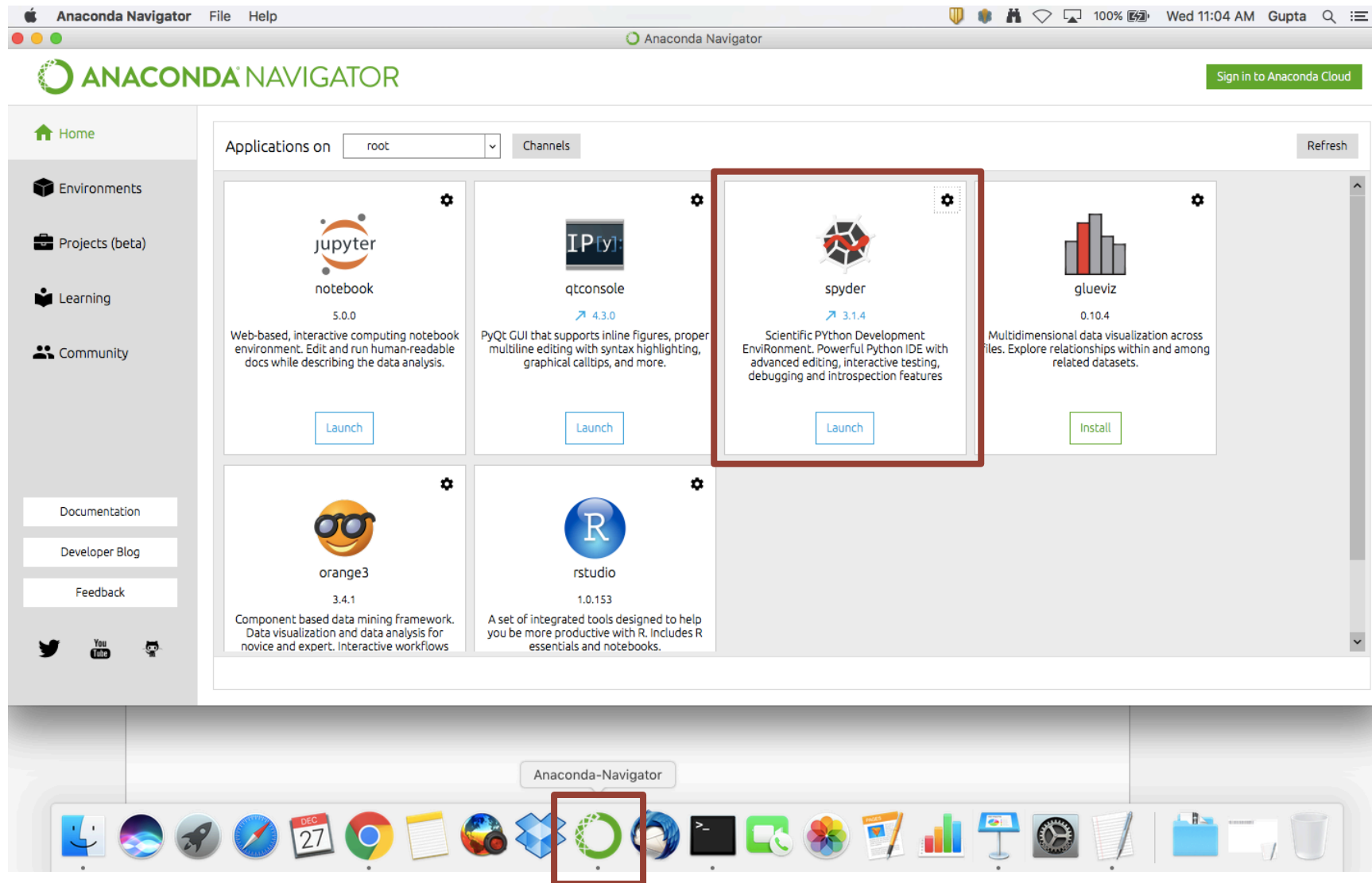
- Escala de 0 a 3
- Se Recomienda QA de 3 para Tierra
- Factores:
 - Número de píxeles
 - Adecuación de errores
 - **Reflectancia superficial**



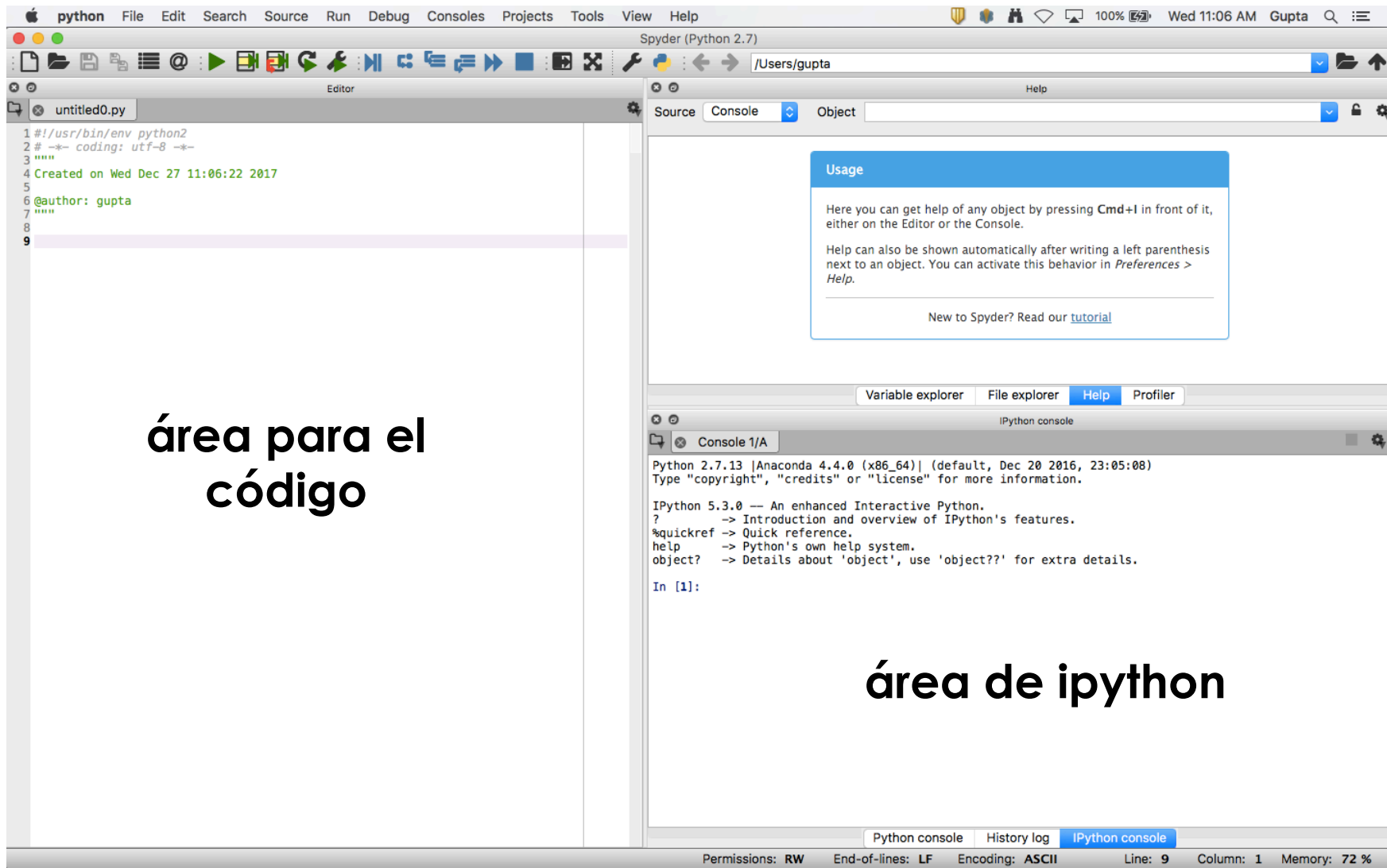


Alistándose con Python

Anaconda y el Editor Spyder



Vista de Spyder



área para el
código

área de ipython

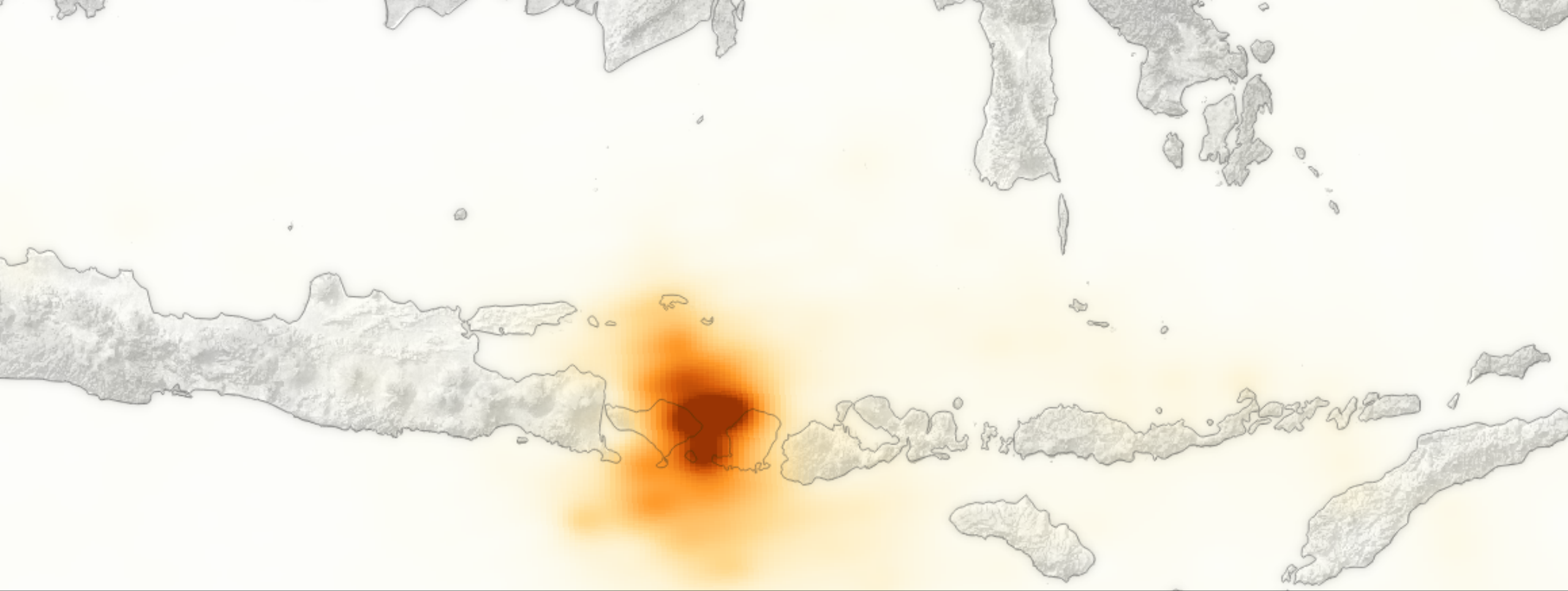


Vista del Directorio Actual y Lista de Archivos

- Cree una lista de archivos HDF 'fileList.txt'
- El directorio debe tener
 - Todos los códigos Python
 - Todos los archivos HDF
 - Una lista de los archivos HDF titulado 'fileList.txt'

```
gs614-guptaml:CA_TRN gupta$ vi fileList.txt
gs614-guptaml:CA_TRN gupta$ ls
MYD04_3K.A2017232.2200.006.2017233154505.png
MYD04_3K.A2017232.2200.006.2017233154505.txt
MYD04_L2.A2017232.1520.006.2017233154749.png
MYD04_L2.A2017232.2200.006.2017233154546.png
MYD04_L2.A2017249.1925.006.2017250160408.hdf
MYD04_L2.A2017249.1925.006.2017250160408.txt
MYD04_L2.A2017249.1930.006.2017250160703.hdf
MYD04_L2.A2017249.2105.006.2017250160535.hdf
fileList.txt
py1
read_and_map_mod_aerosol.py
read_aod_and_calculate_pm25.py
read_mod_aerosol_and_dump_ascii.py
read_mod_aerosol_and_list_sds.py
read_mod_aerosol_at_a_location.py
readme
gs614-guptaml:CA_TRN gupta$ ls *.hdf
MYD04_L2.A2017249.1925.006.2017250160408.hdf
MYD04_L2.A2017249.1930.006.2017250160703.hdf
MYD04_L2.A2017249.2105.006.2017250160535.hdf
gs614-guptaml:CA_TRN gupta$ ls *.hdf >fileList.txt
gs614-guptaml:CA_TRN gupta$ more fileList.txt
MYD04_L2.A2017249.1925.006.2017250160408.hdf
MYD04_L2.A2017249.1930.006.2017250160703.hdf
MYD04_L2.A2017249.2105.006.2017250160535.hdf
gs614-guptaml:CA_TRN gupta$ █
```



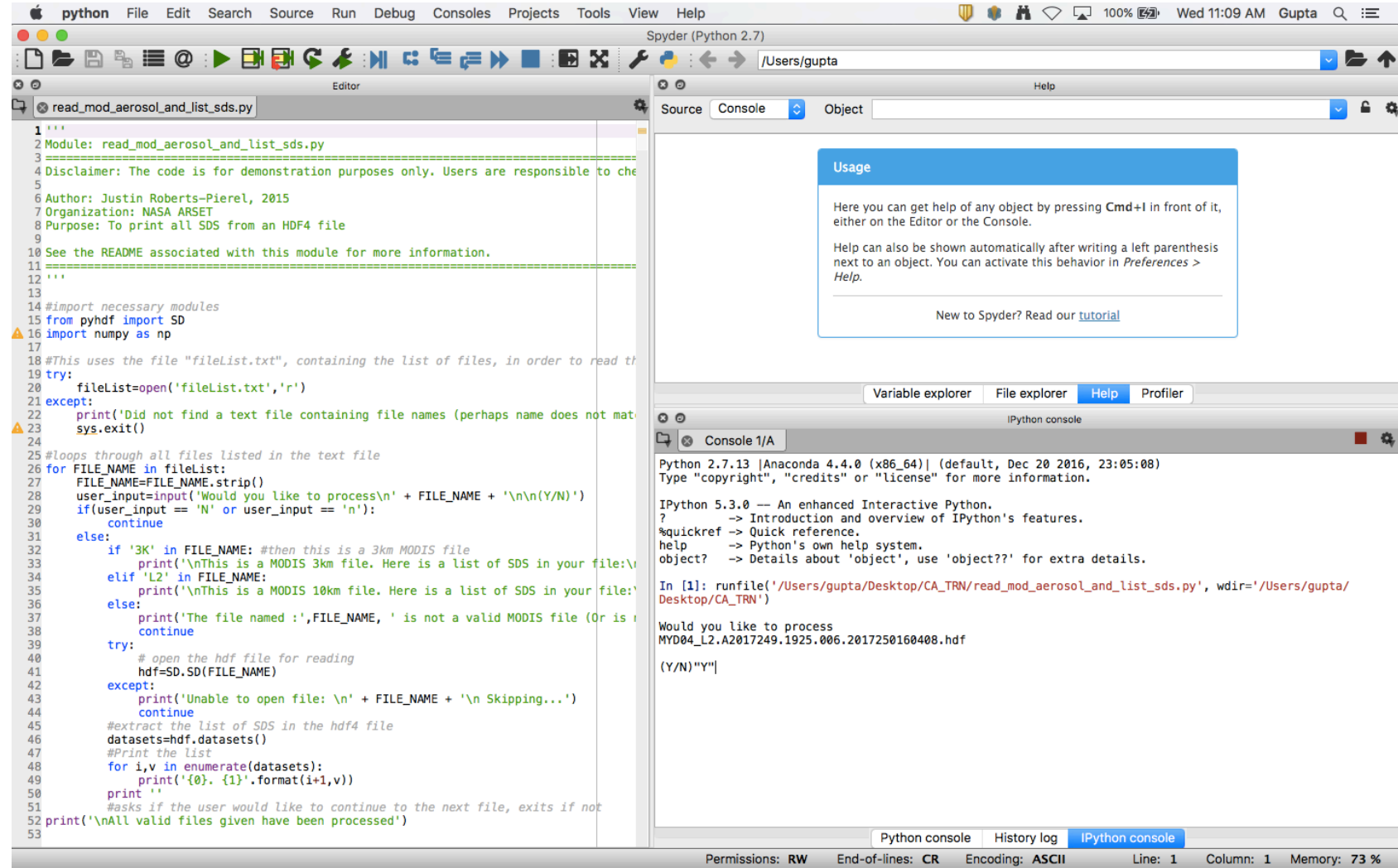


**Lea un Archivo de Aerosoles de MODIS (HDF)
e Imprima la Lista de SDS**

Imprima Conjuntos de Datos Científicos (Scientific Data Sets o SDSs)

read_mod_aerosol_and_list_sds.py

- **Propósito:** leer un archivo de aerosoles de MODIS de nivel 2 en formato HDF e imprimir todos los conjuntos de datos científicos (**Scientific Data Sets** o SDS)
- El código sirve para productos de 10 km y de 3 km



```
python File Edit Search Source Run Debug Consoles Projects Tools View Help
Spyder (Python 2.7)
/Users/gupta
read_mod_aerosol_and_list_sds.py
1 '''
2 Module: read_mod_aerosol_and_list_sds.py
3 =====
4 Disclaimer: The code is for demonstration purposes only. Users are responsible to che
5
6 Author: Justin Roberts-Pierel, 2015
7 Organization: NASA ARSET
8 Purpose: To print all SDS from an HDF4 file
9
10 See the README associated with this module for more information.
11 =====
12 '''
13
14 #import necessary modules
15 from pyhdf import SD
16 import numpy as np
17
18 #This uses the file "fileList.txt", containing the list of files, in order to read th
19 try:
20     fileList=open('fileList.txt','r')
21 except:
22     print('Did not find a text file containing file names (perhaps name does not mat
23     sys.exit()
24
25 #loops through all files listed in the text file
26 for FILE_NAME in fileList:
27     FILE_NAME=FILE_NAME.strip()
28     user_input=input('Would you like to process\n' + FILE_NAME + '\n\n(Y/N)')
29     if(user_input == 'N' or user_input == 'n'):
30         continue
31     else:
32         if '3K' in FILE_NAME: #then this is a 3km MODIS file
33             print('\nThis is a MODIS 3km file. Here is a list of SDS in your file:\n
34         elif 'L2' in FILE_NAME:
35             print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:\n
36         else:
37             print('The file named :',FILE_NAME, ' is not a valid MODIS file (Or is i
38             continue
39     try:
40         # open the hdf file for reading
41         hdf=SD(FILE_NAME)
42     except:
43         print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
44         continue
45     #extract the list of SDS in the hdf4 file
46     datasets=hdf.datasets()
47     #Print the list
48     for i,v in enumerate(datasets):
49         print('{0}. {1}'.format(i+1,v))
50     print ''
51     #asks if the user would like to continue to the next file, exits if not
52 print('\nAll valid files given have been processed')
53
```

Usage

Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Variable explorer | File explorer | Help | Profiler

IPython console

Console 1/A

Python 2.7.13 [Anaconda 4.4.0 (x86_64)] (default, Dec 20 2016, 23:05:08)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.

In [1]: runfile('/Users/gupta/Desktop/CA_TRW/read_mod_aerosol_and_list_sds.py', wdir='/Users/gupta/Desktop/CA_TRW')

Would you like to process
MYD04_L2.A2017249.1925.006.2017250160408.hdf
(Y/N)"Y"

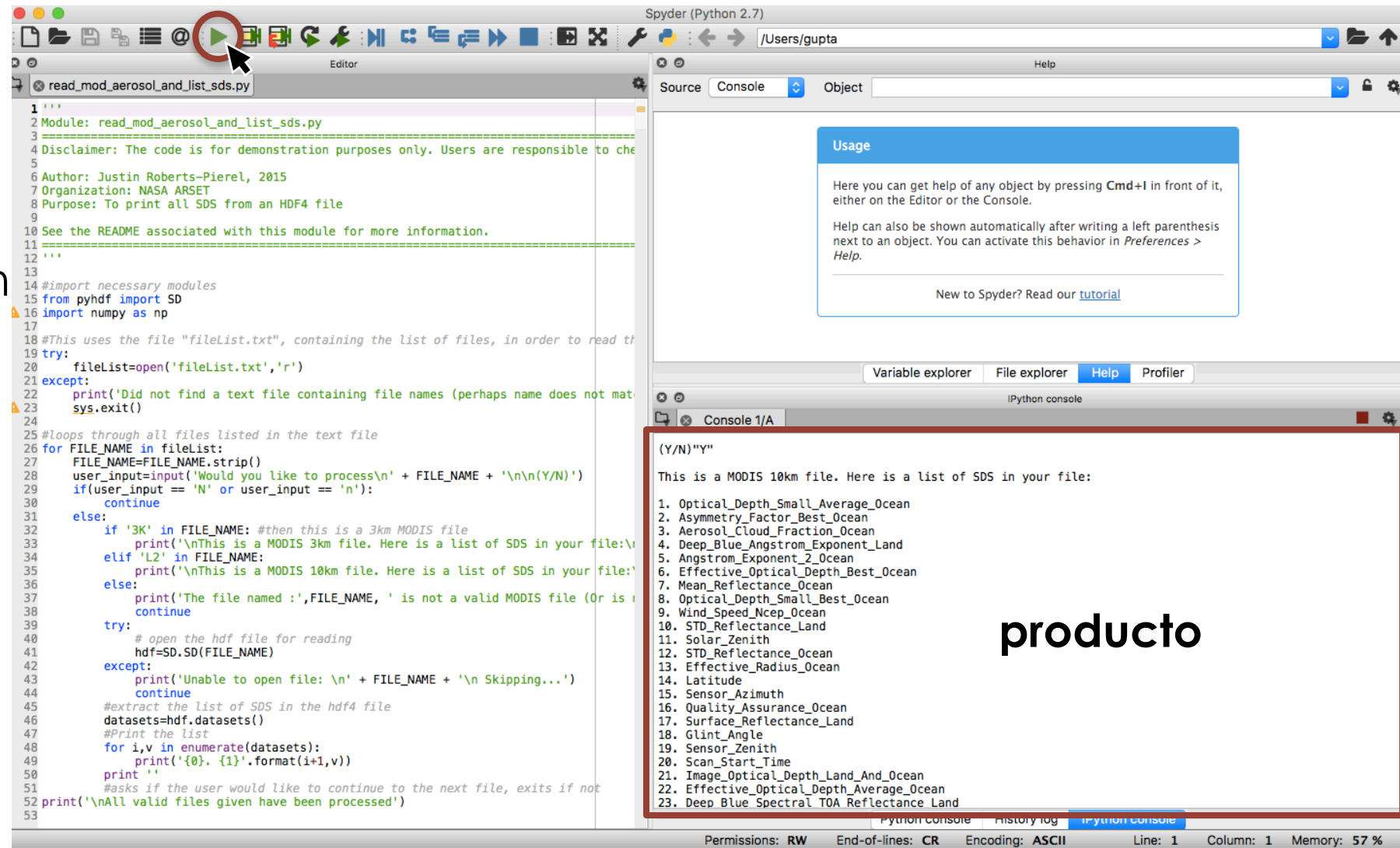
Python console | History log | IPython console

Permissions: RW End-of-lines: CR Encoding: ASCII Line: 1 Column: 1 Memory: 73 %



Ejecución y Producto

- Haga clic en la flecha verde para ejecutar el código
- El código procesará todos los archivos en la lista **fileList.txt** uno por uno
- Siga las instrucciones en la terminal de **ipython** (es decir, ingrese 'Y' o 'N' cuando éste se lo pida y teclee Enter)



The screenshot shows the Spyder Python IDE interface. The top toolbar contains a green play button (run icon) circled in red, with a mouse cursor pointing to it. The main editor window displays a Python script named `read_mod_aerosol_and_list_sds.py`. The script includes a docstring with a disclaimer and author information, followed by imports for `pyhdf` and `numpy`. The code then reads a file named `fileList.txt` and iterates through its contents. For each file, it prompts the user with a question: "Would you like to process\n" + FILE_NAME + "\n\n(Y/N)". The user has entered 'Y' in the console, and the script has printed a list of 23 SDS parameters for a MODIS 10km file. The console output is highlighted with a red box and labeled "producto".

```
1 '''
2 Module: read_mod_aerosol_and_list_sds.py
3
4 Disclaimer: The code is for demonstration purposes only. Users are responsible to che
5
6 Author: Justin Roberts-Pierel, 2015
7 Organization: NASA ARSET
8 Purpose: To print all SDS from an HDF4 file
9
10 See the README associated with this module for more information.
11 =====
12 '''
13
14 #import necessary modules
15 from pyhdf import SD
16 import numpy as np
17
18 #This uses the file "fileList.txt", containing the list of files, in order to read th
19 try:
20     fileList=open('fileList.txt','r')
21 except:
22     print('Did not find a text file containing file names (perhaps name does not mat
23     sys.exit()
24
25 #loops through all files listed in the text file
26 for FILE_NAME in fileList:
27     FILE_NAME=FILE_NAME.strip()
28     user_input=input('Would you like to process\n' + FILE_NAME + '\n\n(Y/N)')
29     if(user_input == 'N' or user_input == '\n'):
30         continue
31     else:
32         if '3K' in FILE_NAME: #then this is a 3km MODIS file
33             print('\nThis is a MODIS 3km file. Here is a list of SDS in your file:\n
34         elif 'L2' in FILE_NAME:
35             print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:'
36         else:
37             print('The file named :,FILE_NAME, ' is not a valid MODIS file (Or is i
38             continue
39         try:
40             # open the hdf file for reading
41             hdf=SD(FILE_NAME)
42         except:
43             print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
44             continue
45         #extract the list of SDS in the hdf4 file
46         datasets=hdf.datasets()
47         #Print the list
48         for i,v in enumerate(datasets):
49             print('{0}. {1}'.format(i+1,v))
50         print ''
51         #asks if the user would like to continue to the next file, exits if not
52     print('\nAll valid files given have been processed')
53
```

(Y/N)"Y"

This is a MODIS 10km file. Here is a list of SDS in your file:

1. Optical_Depth_Small_Average_Ocean
2. Asymmetry_Factor_Best_Ocean
3. Aerosol_Cloud_Fraction_Ocean
4. Deep_Blue_Angstrom_Exponent_Land
5. Angstrom_Exponent_2_Ocean
6. Effective_Optical_Depth_Best_Ocean
7. Mean_Reflectance_Ocean
8. Optical_Depth_Small_Best_Ocean
9. Wind_Speed_Ncep_Ocean
10. STD_Reflectance_Land
11. Solar_Zenith
12. STD_Reflectance_Ocean
13. Effective_Radius_Ocean
14. Latitude
15. Sensor_Azimuth
16. Quality_Assurance_Ocean
17. Surface_Reflectance_Land
18. Glint_Angle
19. Sensor_Zenith
20. Scan_Start_Time
21. Image_Optical_Depth_Land_And_Ocean
22. Effective_Optical_Depth_Average_Ocean
23. Deep_Blue_Spectral_TOA_Reflectance_Land

producto



Editando el Código

```
Module: read_mod_aerosol_and_lst_sds.py
=====
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objects.

Author: Justin Roberts-Pierel, 2015
Organization: NASA ARSET
Purpose: To print all SDS from an HDF4 file

See the README associated with this module for more information.
=====
'''

#import necessary modules
from pyhdf import SD
import numpy as np

#This uses the file "fileList.txt", containing the list of files, in order to read the files
try:
    fileList=open('fileList.txt','r')
except:
    print('Did not find a text file containing file names (perhaps name does not match)')
    sys.exit()

#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input('Would you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if(user_input == 'N' or user_input == 'n'):
        continue
    else:
        if '3K' in FILE_NAME: #then this is a 3km MODIS file
            print('\nThis is a MODIS 3km file. Here is a list of SDS in your file:\n')
        elif 'L2' in FILE_NAME:
            print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:\n')
        else:
            print('The file named :,FILE_NAME, ' is not a valid MODIS file (Or is named incorrectly). \n')
            continue
        try:
            # open the hdf file for reading
            hdf=SD.SD(FILE_NAME)
        except:
            print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
            continue
        #extract the list of SDS in the hdf4 file
        datasets=hdf.datasets()
        #Print the list
        for i,v in enumerate(datasets):
            print('{0}. {1}'.format(i+1,v))
        print ''
        #asks if the user would like to continue to the next file, exits if not
    print('\nAll valid files given have been processed')
```

cambie el nombre de
fileList.txt por cualquiera
que le guste

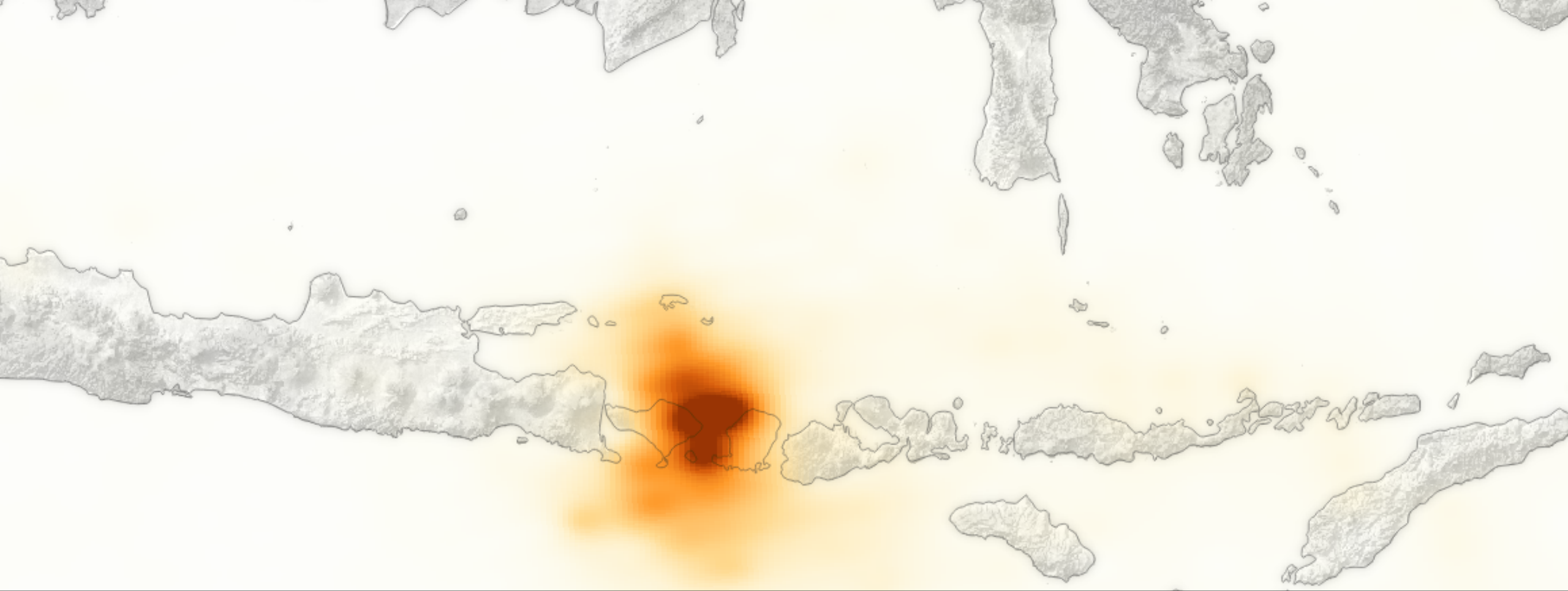
Este código ha
sido probado
para archivos de
aerosoles de
MODIS de Nivel 2
de 3 km y 10 km



Aplicaciones

- Los datos de aerosoles de MODIS de Nivel 2 están guardados en archivos HDF
- Cada archivo HDF contiene varios parámetros geofísicos
- Se requieren códigos y herramientas especiales para abrir archivos HDF
- Este código ayuda a los usuarios a ver los nombres de los SDSs dentro de un archivo HDF para análisis adicional





Mapee el Espesor Óptico de Aerosoles

Trace y Guarde un Mapa del AOD de MODIS

read_and_map_mod_aerosol.py

The image shows the Spyder Python IDE interface. The main editor window displays the code for `read_and_map_mod_aerosol.py`. The code includes imports for `pyhdf`, `numpy`, `Basemap`, and `matplotlib.pyplot`. It reads a list of files from `fileList.txt` and processes each file based on its MODIS resolution (3km, 10km, or AOD_550_Dark_Target_Deep_Blue_Combined). The script extracts latitude and longitude information and attempts to read the Aerosol Optical Depth (AOD) data from the HDF files.

```
13 '''
14
15 #import necessary modules
16 from pyhdf import SD
17 import numpy as np
18 from mpl_toolkits.basemap import Basemap
19 import matplotlib.pyplot as plt
20 import sys
21
22 #This uses the file "fileList.txt", containing the list of files, in order to read the files
23 try:
24     fileList=open('fileList.txt','r')
25 except:
26     print('Did not find a text file containing file names (perhaps name does not match)')
27     sys.exit()
28
29 #Loops through all files listed in the text file
30 for FILE_NAME in fileList:
31     FILE_NAME=FILE_NAME.strip()
32     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
33     if(user_input == 'N' or user_input == 'n'):
34         continue
35     else:
36         if '3K' in FILE_NAME:#then this is a 3km MODIS file
37             print('This is a 3km MODIS file. Here is some information: ')
38             SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
39         elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
40             print('This is a 10km MODIS file. Here is some information: ')
41             SDS_NAME='AOD_550_Dark_Target_Deep_Blue_Combined'
42         else:#if it is neither 3km nor 10km, then this will skip the rest of this loop iteration
43             print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorrectly).\n')
44             continue
45         try:
46             # open the hdf file for reading
47             hdf=SD(FILE_NAME)
48         except:
49             print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
50             continue
51
52         # Get lat and lon info
53         lat = hdf.select('Latitude')
54         latitude = lat[:]
55         min_lat=latitude.min()
56         max_lat=latitude.max()
57         lon = hdf.select('Longitude')
58         longitude = lon[:]
59         min_lon=longitude.min()
60         max_lon=longitude.max()
61
62         #get AOD SDS, or exit if it doesn't find the SDS in the file
63         try:
64             sds=hdf.select(SDS_NAME)
65         except:
66             print('Sorry, your MODIS hdf file does not contain the SDS:',SDS_NAME, '. Please try aga
```



Ejecución y Producto Output

```
In [1]: runfile('/Users/gupta/Desktop/CA_TRN/read_and_map_mod_aerosol.py', wdir='/Users/gupta/Desktop/CA_TRN')
```

```
Would you like to process  
MYD04_L2.A2017249.1925.006.2017250160408.hdf
```

```
(Y/N)"Y"  
This is a 10km MODIS file. Here is some information:  
('\n\nThe valid range of values is: ', -0.1, ' to ', 5.0, '\n\nThe average is: ', 0.178, '\n\nThe standard deviation is: ', 0.23)  
('\n\nThe range of latitude in this file is: ', 27.187273, ' to ', 48.299458, 'degrees \n\nThe range of longitude in this file is: ', -111.39777, ' to ', -80.255447, ' degrees')
```

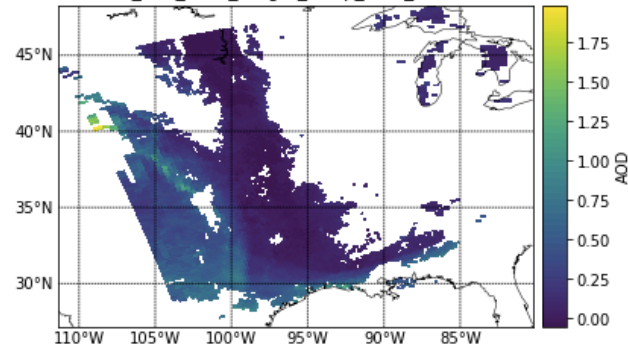
```
Would you like to create a map of this data? Please enter Y or N  
"Y"
```

```
/Users/pgupta3/python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/__init__.py:3413:  
MatplotlibDeprecationWarning: The ishold function was deprecated in version 2.0.
```

```
b = ax.ishold()  
/Users/pgupta3/python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/__init__.py:3422:  
MatplotlibDeprecationWarning: axes.hold is deprecated.
```

```
See the API Changes document (http://matplotlib.org/api/api\_changes.html)  
for more details.  
ax.hold(b)
```

```
MYD04_L2.A2017249.1925.006.2017250160408  
AOD_550_Dark_Target_Deep_Blue_Combined
```



Mapa del AOD
producido

```
Would you like to save this map? Please enter Y or N
```

Estadísticas
del AOD

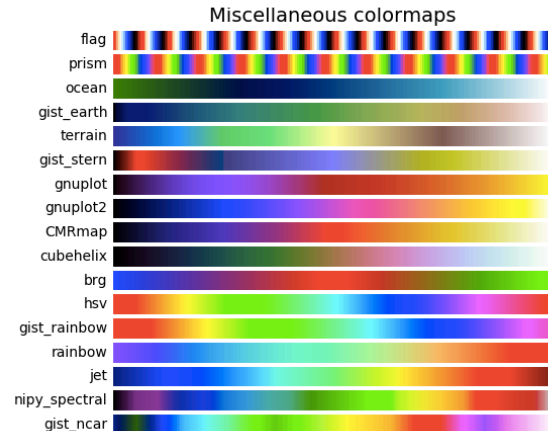


Editando el Código

Cambie de Escala Cromática

```
data=data.astype(float)
data[data == fv] = np.nan
#create the map
data = np.ma.masked_array(data, np.isnan(data))
m = Basemap(projection='cyl', resolution='l', llcrnrlat=min_lat, urcrnrlat = max_lat, llcrnrlon=min_lon,
m.drawcoastlines(linewidth=0.5)
m.drawparallels(np.arange(-90., 120., 5.), labels=[1, 0, 0, 0])
m.drawmeridians(np.arange(-180., 181., 5.), labels=[0, 0, 0, 1])
x, y = m(longitude, latitude)
m.pcolormesh(x, y, data*scale_factor, cmap=plt.cm.jet)
plt.autoscale()
#create colorbar
cb = m.colorbar()
#label colorbar
cb.set_label('AOD')
```

(png, pdf)



https://matplotlib.org/examples/color/colormaps_reference.html

Cambie de SDS

```
#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if(user_input == 'N' or user_input == 'n'):
        continue
    else:
        if '3K' in FILE_NAME:#then this is a 3km MODIS file
            print('This is a 3km MODIS file. Here is some information: ')
            SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
        elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
            print('This is a 10km MODIS file. Here is some information: ')
            SDS_NAME='AOD_550_Dark_Target_Deep_Blue_Combined'
        else:#if it is neither 3km nor 10km, then this will skip the rest of this loop iteration
            print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorrectly). \n')
            continue
    try:
        # open the hdf file for reading
        hdf=SD.SD(FILE_NAME)
    except:
        print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
        continue
```



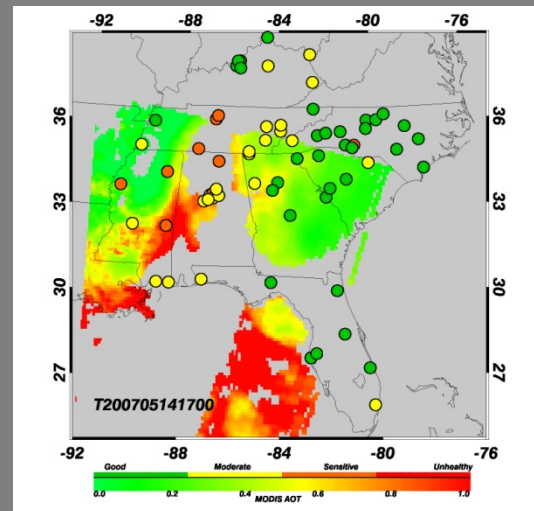
Aplicaciones

- Éste es un ejemplar de un código para leer y mapear los datos de aerosoles de MODIS de Nivel 2
- Se puede modificar el código para diferentes necesidades cartográficas
- Los usuarios pueden crear mapas diarios del AOD sobre ciertas regiones y comenzar a analizar cambios a través del tiempo
- Los mapas del AOD también pueden ayudar a identificar regiones con niveles de contaminación elevados

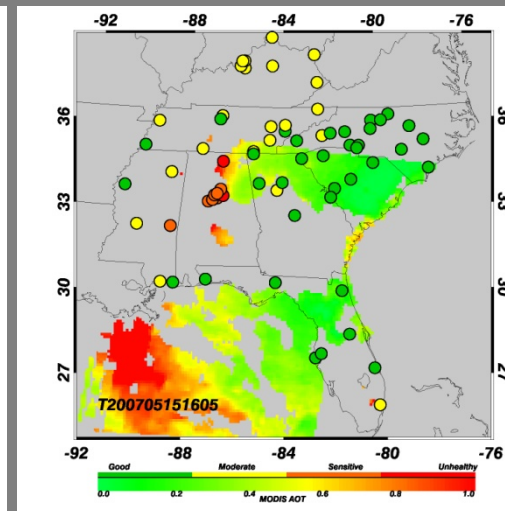
Ejemplo:

Valores elevados del AOD por causa del humo concuerdan estrechamente con monitores en la superficie (círculos).

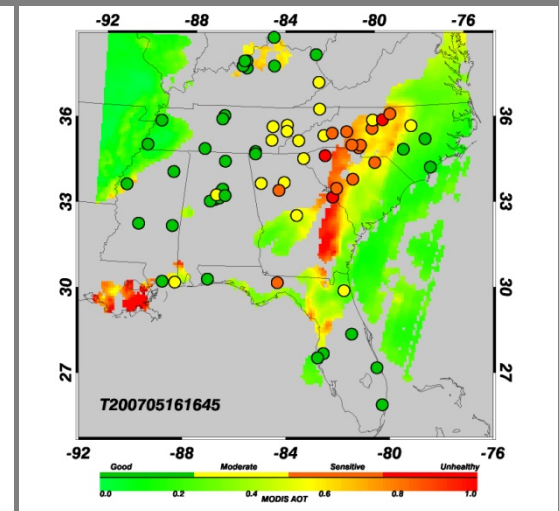
14 de mayo de 2007

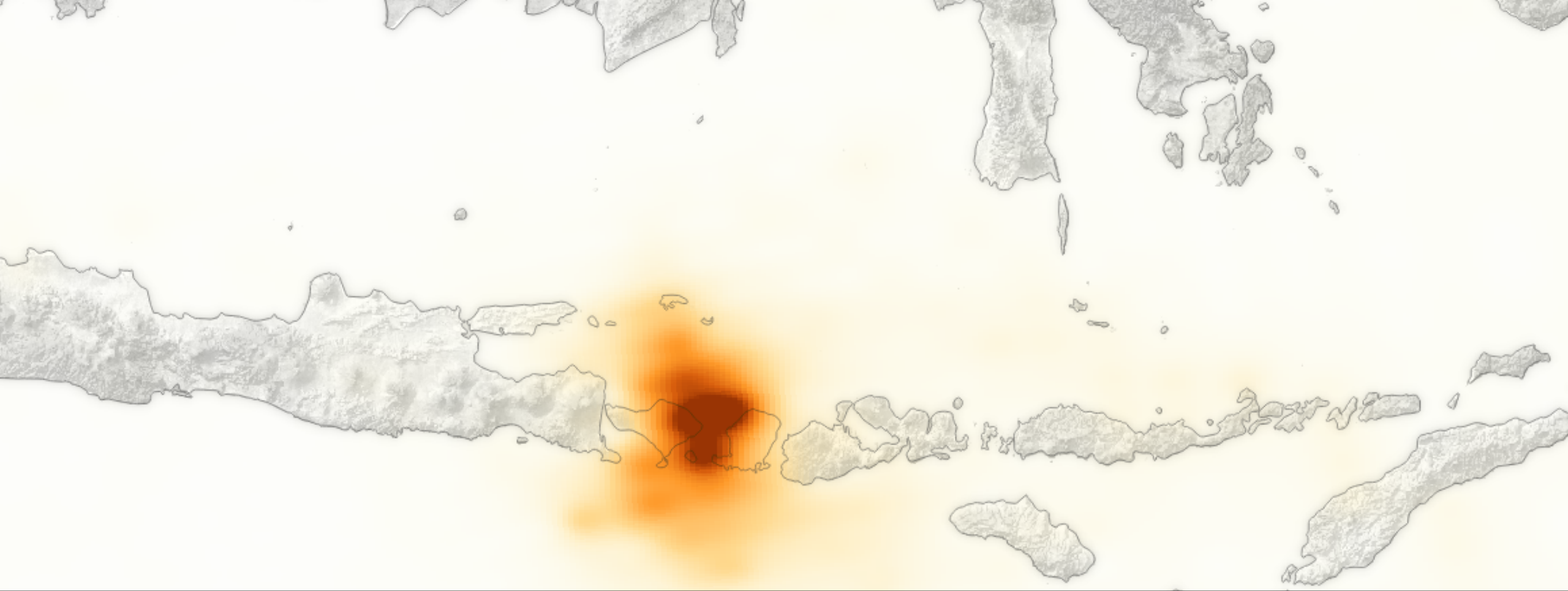


15 de mayo de 2007



16 de mayo de 2007





Extraiga el AOD de una Estación Superficial

Extraiga los Valores del AOD en un Lugar Indicado

read_mod_aerosol_at_a_location.py

- **Propósito:** leer un archivo de datos de MODIS de nivel 2 en formato HDF y extraer los valores del AOD en una ubicación en la superficie indicada
- El código sirve para productos de 10 km y 3 km

```
python File Edit Search Source Run Debug Consoles Projects Tools View Help
Spyder (Python 2.7)
/Users/gupta
read_mod_aerosol_at_a_location.py* untitled0.py
1 #!/usr/bin/python
2 '''
3 Module: read_mod_aerosol_at_a_location.py
4 =====
5 Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and
6
7 Author: Justin Roberts-Pierel, 2015
8 Organization: NASA ARSET
9 Purpose: To view info about a variety of SDS from a MODIS HDF4 file (or series of files) both generally and
10
11 See the README associated with this module for more information.
12 =====
13 '''
14
15 #import necessary modules
16 from pyhdf import SD
17 import numpy as np
18
19 #This uses the file "fileList.txt", containing the list of files, in order to read the files
20 try:
21     fileList=open('fileList.txt','r')
22 except:
23     print('Did not find a text file containing file names (perhaps name does not match)')
24     sys.exit()
25
26 #loops through all files listed in the text file
27 for FILE_NAME in fileList:
28     FILE_NAME=FILE_NAME.strip()
29     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
30     if(user_input == 'N' or user_input == 'n'):
31         continue
32     else:
33         if '3K' in FILE_NAME: #then this is a 3km MODIS file
34             userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(
35             while userInput not in {1,2,3,4}:#repeats the question if the user does not choose one of the
36                 print('Please try again.')
37             userInput=int(input('Which SDS would you like to view? (Type the number and press enter)
38             #Uses a Python dictionary to choose the SDS indicated by the user
39             dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Land_Ocean_Quality_Flag'),(3,'Image_
40             elif 'L2' in FILE_NAME:#Same as above but for 10km MODIS file
41             userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(
42             while userInput not in {1,2,3}:
43                 print('Please try again.')
44             userInput=int(input('Which SDS would you like to view? (Type the number and press enter)
45             dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Dark_Target_Deep
46             SDS_NAME=dataFields[int(userInput)] # The name of the sds to read
47             try:
48                 # open the hdf file for reading
49                 hdf=SD(FILE_NAME)
50             except:
51                 print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
52                 continue
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522

```

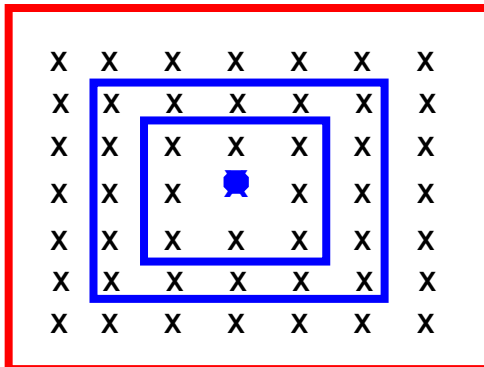

Ejecución y Producto

Teclee "Y" para procesar el archivo, "N" para saltar

Seleccione SDS

Lat. y Lon. de la estación

Productos



```
Would you like to process
MYD04_L2.A2017249.1930.006.2017250160703.hdf
```

```
(Y/N)"Y"
```

```
Which SDS would you like to view? (Type the number and press enter)
```

- (1) Deep_Blue_Aerosol_Optical_Depth_550_Land
- (2) AOD_550_Dark_Target_Deep_Blue_Combined
- (3) AOD_550_Dark_Target_Deep_Blue_Combined_QA_Flag

```
1
('The range of latitude in this file is: ', 44.253548, ' to ', 66.211197, 'degrees
\nThe range of longitude in this file is: ', -126.93629, ' to ', -82.291809, '
degrees')
```

```
Please enter the latitude you would like to analyze (Deg. N): 49.5
```

```
Please enter the longitude you would like to analyze (Deg. E): -100.5
```

```
('\n\nThe nearest pixel to your entered location is at: \nLatitude:', 49.482555, '
Longitude:', -100.51669)
('The value of ', 'Deep_Blue_Aerosol_Optical_Depth_550_Land', 'at this pixel is ',
0.171)
('\n\nThere', 'are', 9, 'valid', 'pixels', 'in a 3x3 grid centered at your entered
location.')
```

```
('\n\nThe average value in this grid is: ', 0.197, ' \n\nThe median value in this grid
is: ', 0.171, '\n\nThe standard deviation in this grid is: ', 0.075)
('\n\nThere', 'are', 24, 'valid', 'pixels', 'in a 5x5 grid centered at your entered
location. \n')
```

```
('The average value in this grid is: ', 0.204, ' \n\nThe median value in this grid
is: ', 0.201, '\n\nThe standard deviation in this grid is: ', 0.077)
```

```
Would you like to process
MYD04_L2.A2017249.2105.006.2017250160535.hdf
```

```
(Y/N)
```



Editando el Código – Cambie de SDS

```
s through all files listed in the text file
FILE_NAME in fileList:
FILE_NAME=FILE_NAME.strip()
user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
if(user_input == 'N' or user_input == 'n'):
    continue
else:
    if '3K' in FILE_NAME: #then this is a 3km MODIS file
        userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Optical_Depth_Land_And_Ocean \n(2)
        while userInput not in {1,2,3,4}:#repeats the question if the user does not choose one of the options
            print('Please try again.')
            userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Optical_Depth_Land_And_Ocean
            #Uses a Python dictionary to choose the SDS indicated by the user
            dataFields=dict([(1, 'Optical_Depth_Land_And_Ocean'), (2, 'Land_Ocean_Quality_Flag'), (3, 'Image_Optical_Depth_Land_And_Ocean'), (4, 'L
    elif 'L2' in FILE_NAME:#Same as above but for 10km MODIS file
        userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Deep_Blue_Aerosol_Optical_Depth_5
        while userInput not in {1,2,3}:
            print('Please try again.')
            userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Deep_Blue_Aerosol_Optical_Dep
            dataFields=dict([(1, 'Deep_Blue_Aerosol_Optical_Depth_550_Land'), (2, 'AOD_550_Dark_Target_Deep_Blue_Combined'), (3, 'AOD_550_Dark_Tai
        SDS_NAME=dataFields[int(userInput)] # The name of the sds to read
    try:
        # open the hdf file for reading
        hdf=SD.SD(FILE_NAME)
    except:
        print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
        continue
```



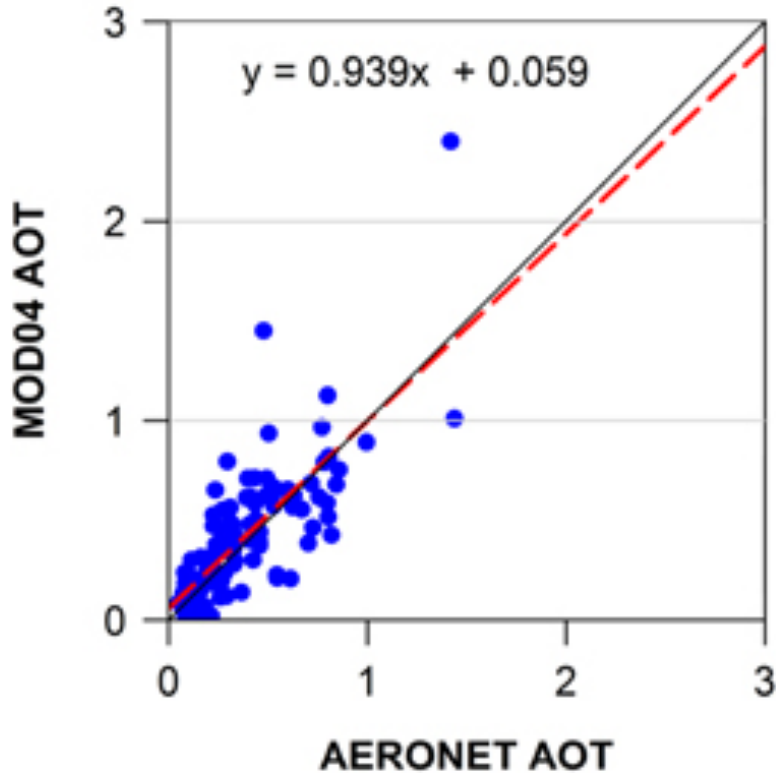
Editando el Código – Cambie los Cálculos del AOD

```
#calculates mean, median, stdev in a 3x3 grid around nearest point to entered location
if x < 1:
    x+=1
if x > data.shape[0]-2:
    x-=2
if y < 1:
    y+=1
if y > data.shape[1]-2:
    y-=2
three_by_three=data[x-1:x+2,y-1:y+2]
three_by_three=three_by_three.astype(float)
three_by_three[three_by_three==float(fillvalue)]=np.nan
nnan=np.count_nonzero(~np.isnan(three_by_three))
if nnan == 0:
    print ('\nThere are no valid pixels in a 3x3 grid centered at your entered location.')
else:
    three_by_three=three_by_three*scale_factor
    three_by_three_average=np.nanmean(three_by_three)
    three_by_three_std=np.nanstd(three_by_three)
    three_by_three_median=np.nanmedian(three_by_three)
    if nnan == 1:
        npixels='is'
        mpixels='pixel'
    else:
        npixels='are'
        mpixels='pixels'
    print ('\nThere', npixels, nnan, 'valid', mpixels, 'in a 3x3 grid centered at your entered location.')
    print ('\nThe average value in this grid is: ', round(three_by_three_average,3), ' \nThe median value in this grid is: ', round(three
```

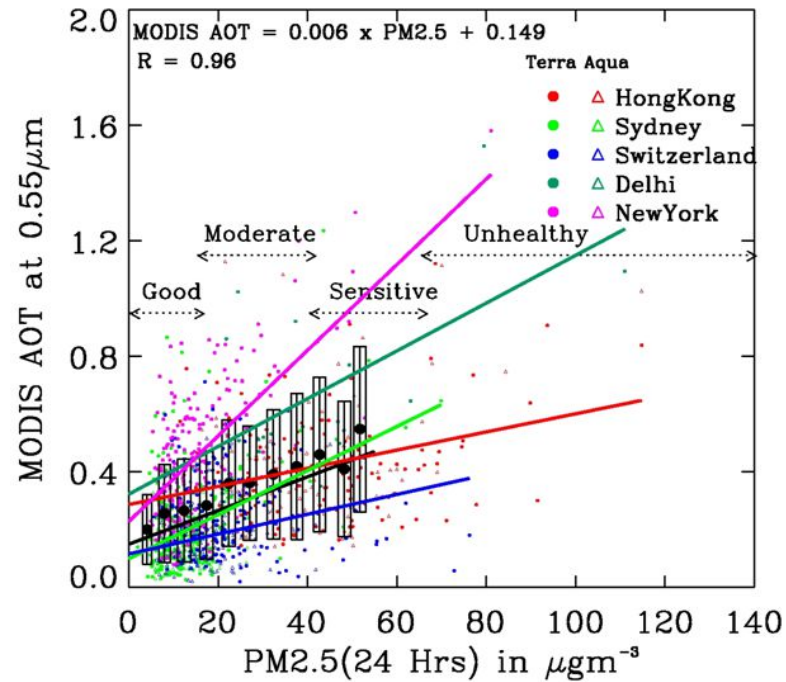


Aplicaciones

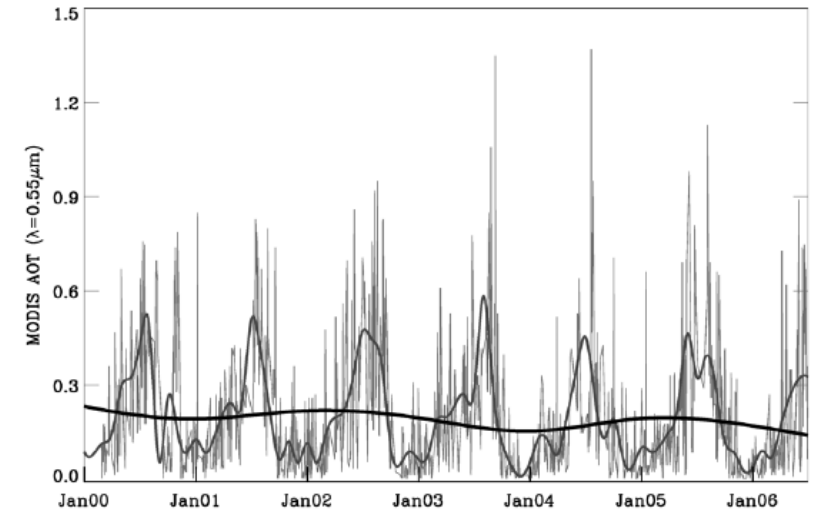
Validación del AOD por Satélite

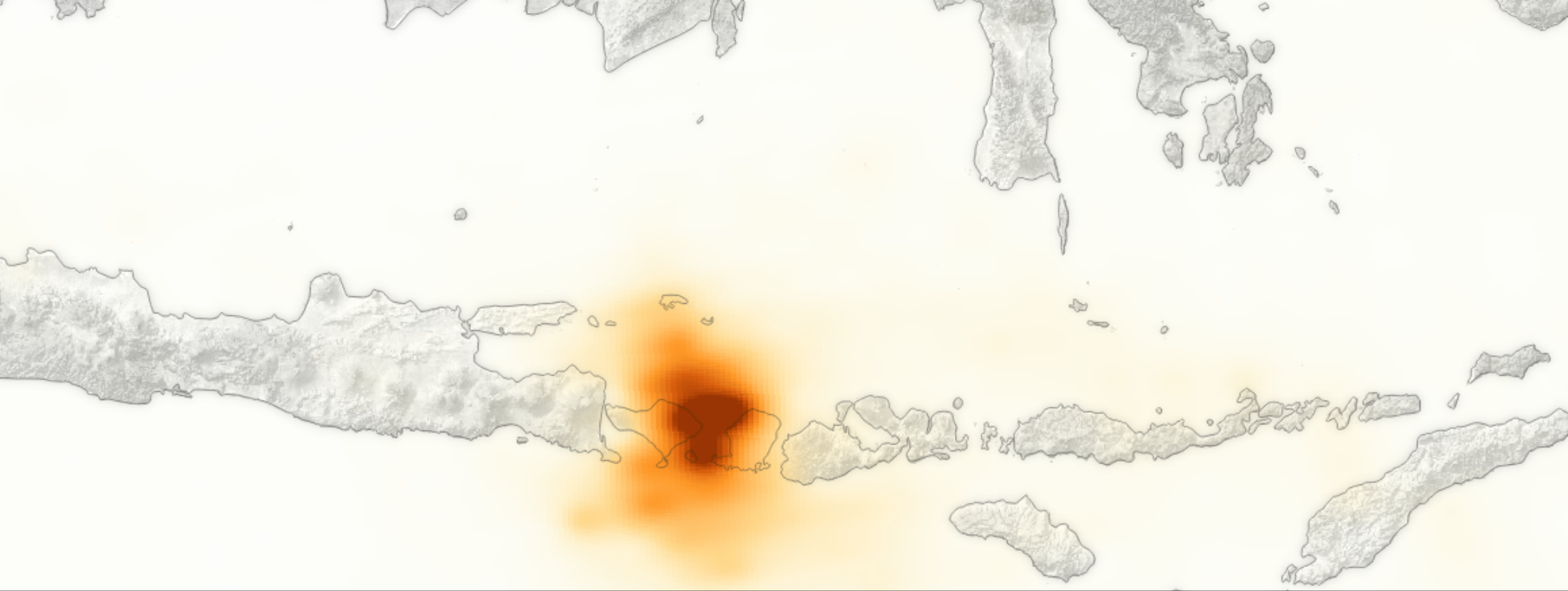


Relación AOD-PM_{2.5}



Análisis de Series Temporales



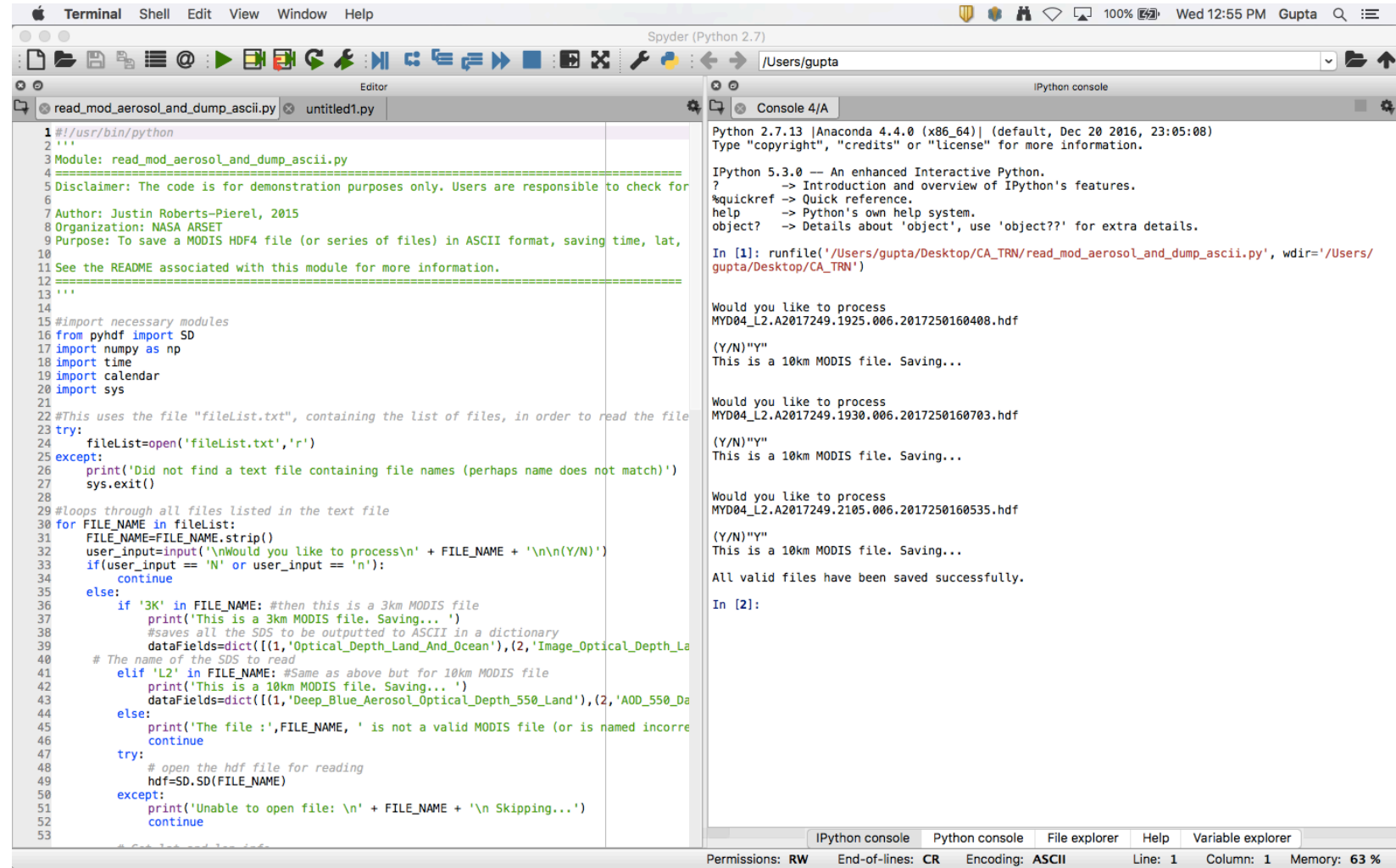


Convierta las Variables del HDF en CSV

Convierta variables de un HDF Nivel 2 de MODIS en un Archivo CSV

read_mod_aerosol_and_dump_ascii.py

- **Propósito:** leer un archivo de datos de MODIS de Nivel 2 en formato HDF y convertir ciertos SDSs en un archivo csv (texto)
- El código funciona para productos de 10 km y 3 km



```
1#!/usr/bin/python
2'''
3Module: read_mod_aerosol_and_dump_ascii.py
4
5Disclaimer: The code is for demonstration purposes only. Users are responsible to check for
6
7Author: Justin Roberts-Pierel, 2015
8Organization: NASA ARSET
9Purpose: To save a MODIS HDF4 file (or series of files) in ASCII format, saving time, lat,
10
11See the README associated with this module for more information.
12'''
13'''
14
15#Import necessary modules
16from pyhdf import SD
17import numpy as np
18import time
19import calendar
20import sys
21
22#This uses the file "fileList.txt", containing the list of files, in order to read the file
23try:
24    fileList=open('fileList.txt','r')
25except:
26    print('Did not find a text file containing file names (perhaps name does not match)')
27    sys.exit()
28
29#loops through all files listed in the text file
30for FILE_NAME in fileList:
31    FILE_NAME=FILE_NAME.strip()
32    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
33    if(user_input == 'N' or user_input == 'n'):
34        continue
35    else:
36        if '3K' in FILE_NAME: #then this is a 3km MODIS file
37            print('This is a 3km MODIS file. Saving... ')
38            #saves all the SDS to be outputted to ASCII in a dictionary
39            dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Image_Optical_Depth_La
40
41# The name of the SDS to read
42            elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
43                print('This is a 10km MODIS file. Saving... ')
44                dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Da
45
46            else:
47                print('The file :',FILE_NAME, ' is not a valid MODIS file (or is named incorre
48                continue
49
50            try:
51                # open the hdf file for reading
52                hdf=SD(FILE_NAME)
53            except:
54                print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
55                continue
```

Python 2.7.13 [Anaconda 4.4.0 (x86_64)] (default, Dec 20 2016, 23:05:08)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.

In [1]: runfile('/Users/gupta/Desktop/CA_TRN/read_mod_aerosol_and_dump_ascii.py', wdir='/Users/gupta/Desktop/CA_TRN')

Would you like to process
MYD04_L2.A2017249.1925.006.2017250160408.hdf
(Y/N)"Y"
This is a 10km MODIS file. Saving...

Would you like to process
MYD04_L2.A2017249.1930.006.2017250160703.hdf
(Y/N)"Y"
This is a 10km MODIS file. Saving...

Would you like to process
MYD04_L2.A2017249.2105.006.2017250160535.hdf
(Y/N)"Y"
This is a 10km MODIS file. Saving...

All valid files have been saved successfully.

In [2]:



Producto

```
MYD04_L2.A2017249.1925.006.2017250160408.txt
Year,Month,Day,Hour,Minute,Second,Longitude,Deep Blue Aerosol Optical Depth 550 Land,AOD 550 Dark Target Deep Blue Combined,AOD 550 Dark Target Deep Blue Combined QA Flag
2017.0.9.0.6.0.19.0.25.0.9.0.30.4542312622,-80.2554473877,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.4285984039,-80.7235641479,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.4032402039,-81.1592407227,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.3782196045,-81.5666427612,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.3535690308,-81.9493026733,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.329334259,-82.3096008301,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.3055496216,-82.6497421265,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.2822036743,-82.9720306396,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.2592868805,-83.2782287598,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.2367897034,-83.5699691772,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.2147369385,-83.8481216431,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.1930789948,-84.1143035889,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.1718177795,-84.3694152832,0.322000015294,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.1509361267,-84.6143875122,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.1304397583,-84.8497695923,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.1103076935,-85.0764007568,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.0905189514,-85.2949752808,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.0710792542,-85.5059127808,0.496000023559,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.0519447327,-85.7100372314,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.0331192017,-85.9076538086,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.30.0145874023,-86.0991973877,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.9963378906,-86.2850646973,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.9783554077,-86.4656219482,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.9606361389,-86.6411895752,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.9431610107,-86.8120880127,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.9259204865,-86.9785919189,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.9089050293,-87.1409683228,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.8921031952,-87.2994613647,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.87550354,-87.4542999268,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.8590984344,-87.6056747437,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.8428764343,-87.7537918091,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.8268318176,-87.8988342285,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.8109512329,-88.040977478,-9999.0,0.483000022941,1.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.7952270508,-88.1803741455,-9999.0,0.537000025506,3.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.7796516418,-88.3171691895,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.7642173767,-88.4515228271,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.7489128113,-88.5835418701,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.7337341309,-88.7133789062,-9999.0,0.523000024841,1.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.7186717987,-88.8411331177,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.7037162781,-88.9669265747,-9999.0,0.400000018999,1.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.688867569,-89.0908660889,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.6741104126,-89.2130584717,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.6594429016,-89.3335876465,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.6448535919,-89.452545166,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.6303424835,-89.5700378418,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.6158943176,-89.6861343384,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.6015090942,-89.8009262085,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.587179184,-89.9144897461,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.5728931427,-90.0269012451,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.5586509705,-90.1382369995,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.5444412231,-90.248550415,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.5302619934,-90.3579177856,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.5161094666,-90.4663772583,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.5019721985,-90.5740280151,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.4878482819,-90.6808853149,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.4737319946,-90.7870483398,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.4596118927,-90.8925476074,-9999.0,-9999.0,0.0
2017.0.9.0.6.0.19.0.25.0.9.0.29.4454841614,-90.9974594116,-9999.0,-9999.0,0.0
```

Este código crea un archivo .csv, el cual se puede abrir con Excel, un editor de texto u otros códigos o software



Editando el Código

Cambie el SDS de la lista a que sea escrito como un producto

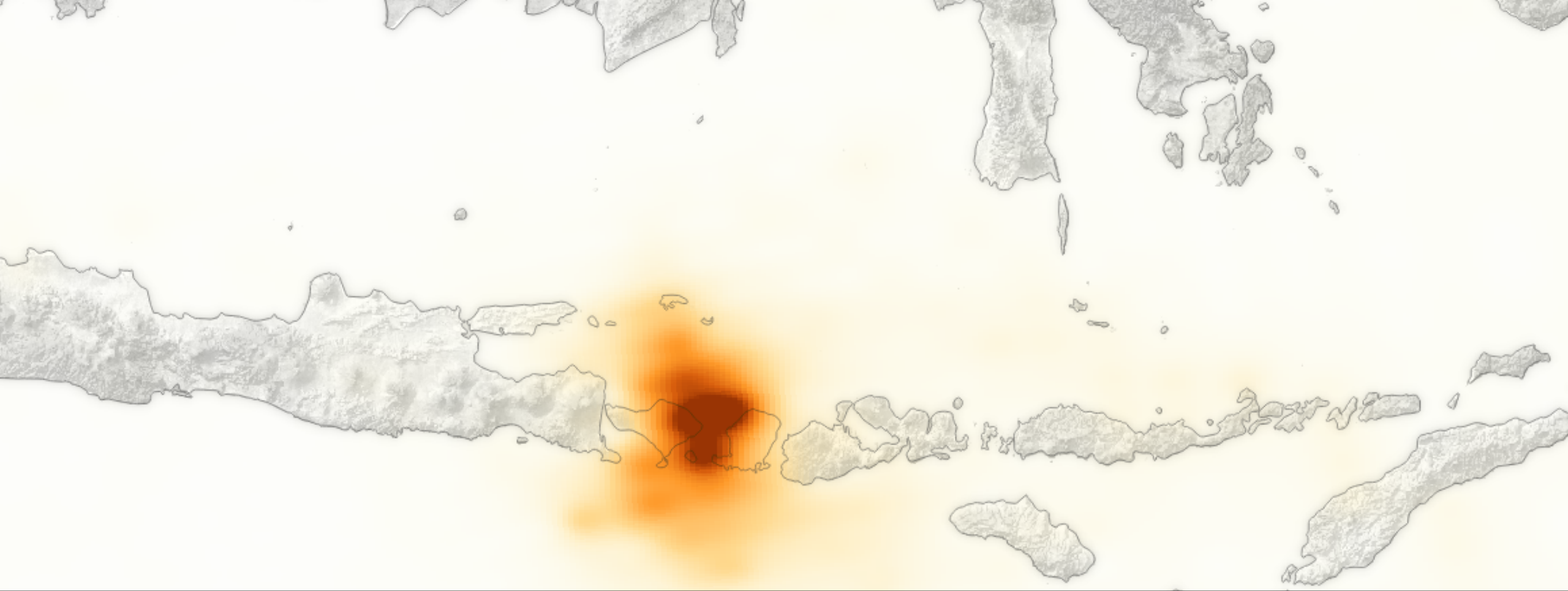
```
21
22 #This uses the file "fileList.txt", containing the list of files, in order to read the file
23 try:
24     fileList=open('fileList.txt','r')
25 except:
26     print('Did not find a text file containing file names (perhaps name does not match)')
27     sys.exit()
28
29 #loops through all files listed in the text file
30 for FILE_NAME in fileList:
31     FILE_NAME=FILE_NAME.strip()
32     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
33     if(user_input == 'N' or user_input == 'n'):
34         continue
35     else:
36         if '3K' in FILE_NAME: #then this is a 3km MODIS file
37             print('This is a 3km MODIS file. Saving... ')
38             #saves all the SDS to be outputted to ASCII in a dictionary
39             dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Image_Optical_Depth_La
40 # The name of the SDS to read
41             elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
42                 print('This is a 10km MODIS file. Saving... ')
43                 dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Da
44             else:
45                 print('The file :',FILE_NAME, ' is not a valid MODIS file (or is named incorre
46                 continue
47             try:
48                 # open the hdf file for reading
49                 hdf=SD.SD(FILE_NAME)
50             except:
51                 print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
52                 continue
53
```



Aplicaciones

- Éste es un ejemplar de un código para leer y extraer los datos de aerosoles de MODIS de Nivel 2
- Se puede modificar el código para extraer múltiples SDSs en un solo archivo .csv
- Se puede modificar el código fácilmente para extraer datos sobre una región en particular
- El archivo producido se puede abrir en Excel o cualquier otra herramienta para analizar datos





Cree Mapas de la Calidad del Aire

Cree un Mapa de la Calidad del Aire

read_aod_and_calculate_pm25.py

- **Propósito:** leer un archivo de datos de aerosoles de MODIS de nivel 2 en formato HDF y crear un mapa de categorías de calidad del aire usando la relación entre el AOD y PM2.5
- El código sirve para productos de 10 km y 3 km

The screenshot shows the Spyder Python IDE interface. The left pane displays the code for `read_aod_and_calculate_pm25.py`. The right pane shows the IPython console output, which includes a series of prompts and responses, followed by a map of PM2.5 concentrations. The map is titled "MYD04_L2_A2017249_1925_006_2017250160408" and "PM 2.5". The map shows a color-coded distribution of PM2.5 concentrations over a geographic area, with a legend on the right indicating categories: Hazardous (red), Very Unhealthy (orange), Unhealthy (yellow), Unhealthy for Sensitive Groups (light green), Moderate (green), and Good (dark green). The map axes show latitude from 30°N to 45°N and longitude from 110°W to 85°W.

```
1#!/usr/bin/python
2'''
3Module: pm25_modis.py
4
5Disclaimer: The code is for demonstration purposes only. Users are responsible to check for
6
7Author: Justin Roberts-Pierel, 2015
8Organization: NASA ARSET
9Purpose: To extract AOD data from a MODIS HDF4 file (or series of files), calculate PM 2.5
10
11See the README associated with this module for more information.
12'''
13'''
14
15#import necessary modules
16from pyhdf import SD
17import numpy as np
18import sys
19from mpl_toolkits.basemap import Basemap
20import matplotlib.pyplot as plt
21from matplotlib.colors import LinearSegmentedColormap
22
23#This uses the file "fileList.txt", containing the list of files, in order to read the file
24try:
25    fileList=open('fileList.txt','r')
26except:
27    print('Did not find a text file containing file names (perhaps name does not match)')
28    sys.exit()
29
30#loops through all files listed in the text file
31for FILE_NAME in fileList:
32    FILE_NAME=FILE_NAME.strip()
33    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
34    if(user_input == 'N' or user_input == 'n'):
35        continue
36    else:
37        if '3K' in FILE_NAME:#then this is a 3km MODIS file
38            print('This is a 3km MODIS file. Here is some information:')
39            SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
40        elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
41            print('This is a 10km MODIS file. Here is some information:')
42            SDS_NAME='AOD_550_Dark_Target_Deep_Blue_Combined'
43        else:#if it is neither 3km nor 10km, then this will skip the rest of this loop ite
44            print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorre
45            continue
46        try:
47            # open the hdf file for reading
48            hdf=SD.SD(FILE_NAME)
49        except:
50            print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
51            continue
52
53    # Get lat and lon info
```

Would you like to process MYD04_L2_A2017249_1925_006_2017250160408.hdf (Y/N) Y This is a 10km MODIS file. Here is some information: The valid range of values is: ', -0.1, ' to ', 5.0, '\nThe average is: ', 0.178, '\nThe standard deviation is: ', 0.23 ('The range of latitude in this file is: ', 27.187273, ' to ', 48.299458, 'degrees \n\nThe range of longitude in this file is: ', -111.39777, ' to ', -80.255447, ' degrees')

Would you like to enter a slope and intercept for PM 2.5 calculation? N

Would you like to create a map of this data? Please enter Y or N Y

/Users/pgupta3/python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/_init_.py:3413: MatplotlibDeprecationWarning: The ishold function was deprecated in version 2.0. b = ax.ishold() /Users/pgupta3/python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/_init_.py:3422: MatplotlibDeprecationWarning: axes.hold is deprecated. See the API Changes document (http://matplotlib.org/api/api_changes.html) for more details. ax.hold(b)

MYD04_L2_A2017249_1925_006_2017250160408 PM 2.5

Legend: Hazardous, Very Unhealthy, Unhealthy, Unhealthy for Sensitive Groups, Moderate, Good

Would you like to save this map? Please enter Y or N Y

output

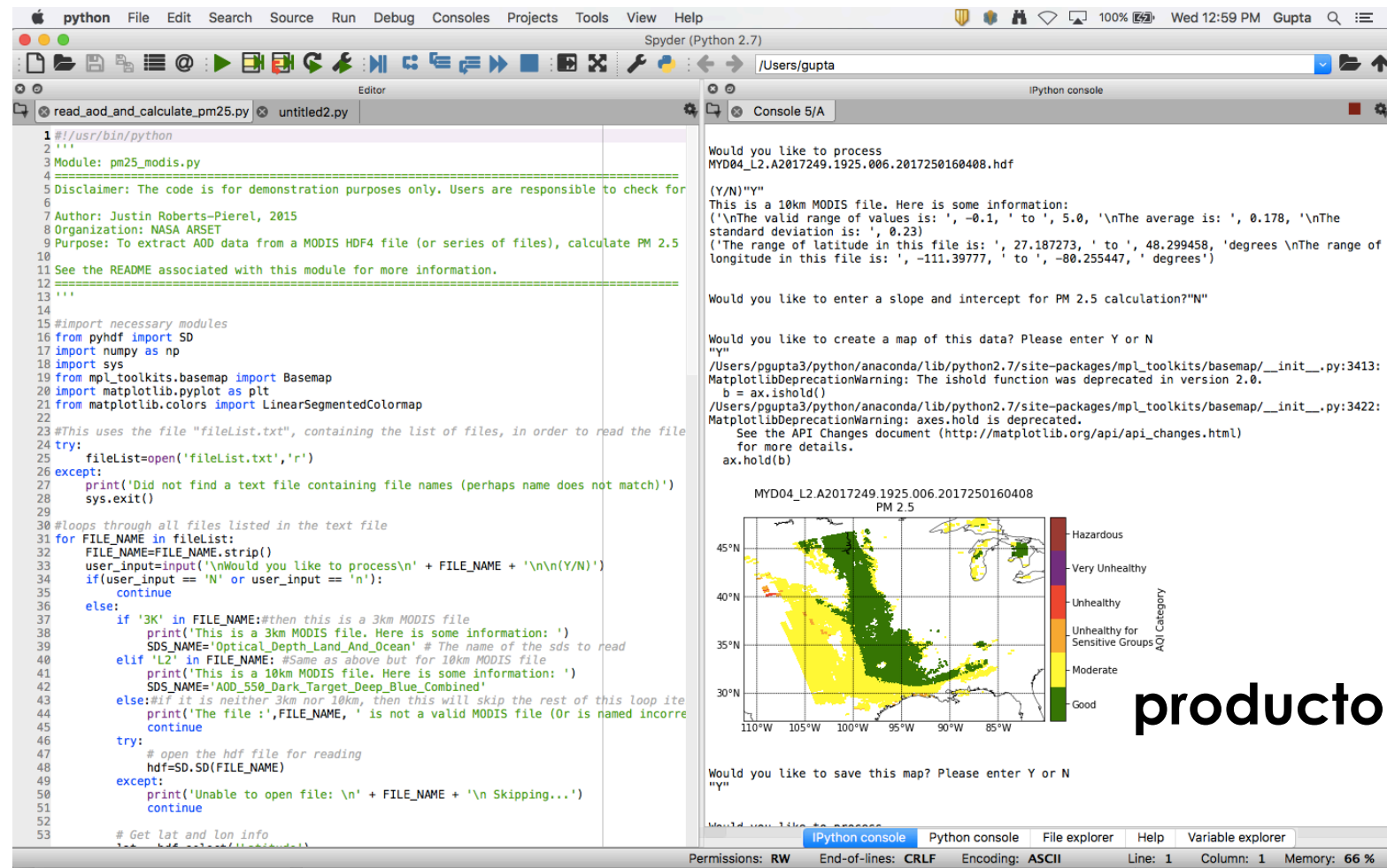


Cree un Mapa de la Calidad del Aire

read_aod_and_calculate_pm25.py

Advertencia: Éste es sólo un ejemplar de un código. La relación AOD-PM2.5 por defecto que se usa aquí es la relación supuesta sobre EEUU.

Los usuarios de este código tienen la responsabilidad de verificar la validez de esta relación y se les recomienda utilizar relaciones locales para visualizar la Calidad del Aire en diferentes partes del mundo.



The screenshot shows the Spyder Python IDE interface. The left pane displays the code for `read_aod_and_calculate_pm25.py`. The right pane shows the IPython console output, which includes a series of prompts and responses, and a map of the United States showing PM2.5 concentrations. The map is titled "MYD04_L2_A2017249.1925.006.2017250160408 PM 2.5" and features a color scale legend for AQI Category: Good (green), Moderate (yellow), Unhealthy for Sensitive Groups (orange), Unhealthy (red), Very Unhealthy (purple), and Hazardous (dark red). The map shows high concentrations (red and purple) in the eastern United States and lower concentrations (green and yellow) in the western and central regions.

```
1#!/usr/bin/python
2'''
3Module: pm25_modis.py
4
5Disclaimer: The code is for demonstration purposes only. Users are responsible to check for
6
7Author: Justin Roberts-Piereel, 2015
8Organization: NASA ARSET
9Purpose: To extract AOD data from a MODIS HDF4 file (or series of files), calculate PM 2.5
10
11See the README associated with this module for more information.
12'''
13
14
15#import necessary modules
16from pyhdf import SD
17import numpy as np
18import sys
19from mpl_toolkits.basemap import Basemap
20import matplotlib.pyplot as plt
21from matplotlib.colors import LinearSegmentedColormap
22
23#This uses the file "fileList.txt", containing the list of files, in order to read the file
24try:
25    fileList=open('fileList.txt','r')
26except:
27    print('Did not find a text file containing file names (perhaps name does not match)')
28    sys.exit()
29
30#loops through all files listed in the text file
31for FILE_NAME in fileList:
32    FILE_NAME=FILE_NAME.strip()
33    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
34    if(user_input == 'N' or user_input == 'n'):
35        continue
36    else:
37        if '3K' in FILE_NAME:#then this is a 3km MODIS file
38            print('This is a 3km MODIS file. Here is some information: ')
39            SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
40        elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
41            print('This is a 10km MODIS file. Here is some information: ')
42            SDS_NAME='AOD_550_Dark_Target_Deep_Blue_Combined'
43        else:#if it is neither 3km nor 10km, then this will skip the rest of this loop ite
44            print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorre
45            continue
46        try:
47            # open the hdf file for reading
48            hdf=SD(FILE_NAME)
49        except:
50            print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
51            continue
52
53        # Get lat and lon info
```

Would you like to process MYD04_L2_A2017249.1925.006.2017250160408.hdf (Y/N) Y This is a 10km MODIS file. Here is some information: ('The valid range of values is: ', -0.1, ' to ', 5.0, '\nThe average is: ', 0.178, '\nThe standard deviation is: ', 0.23) ('The range of latitude in this file is: ', 27.187273, ' to ', 48.299458, 'degrees \nThe range of longitude in this file is: ', -111.39777, ' to ', -80.255447, ' degrees')

Would you like to enter a slope and intercept for PM 2.5 calculation? N

Would you like to create a map of this data? Please enter Y or N Y

Would you like to save this map? Please enter Y or N Y



Editando el Código – Cambie el SDS

El usuario puede cambiar el SDS del AOD a ser usado en el cálculo del PM2.5

```
30 #loops through all files listed in the text file
31 for FILE_NAME in fileList:
32     FILE_NAME=FILE_NAME.strip()
33     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
34     if(user_input == 'N' or user_input == 'n'):
35         continue
36     else:
37         if '3K' in FILE_NAME:#then this is a 3km MODIS file
38             print('This is a 3km MODIS file. Here is some information: ')
39             SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
40         elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
41             print('This is a 10km MODIS file. Here is some information: ')
42             SDS_NAME='AOD_550_Dark_Target_Deep_Blue_Combined'
43         else:#if it is neither 3km nor 10km, then this will skip the rest of this loop iteration
44             print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorrectly). \n')
45             continue
46         try:
47             # open the hdf file for reading
48             hdf=SD.SD(FILE_NAME)
49         except:
50             print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
51             continue
52
```



Editando el Código: Cambie la Relación AOD-PM_{2.5} y el AQI

El código usa

$PM_{2.5} = \text{Slope} * \text{AOD} + \text{Intercept}$
como la ecuación de regresión lineal para calcular PM_{2.5} a partir del AOD

El código utiliza la definición de la Administración de Protección Ambiental de EEUU de categorías de calidad del aire a base de PM_{2.5}

Calculador del AQI:

<https://airnow.gov/index.cfm?action=airnow.calculator>

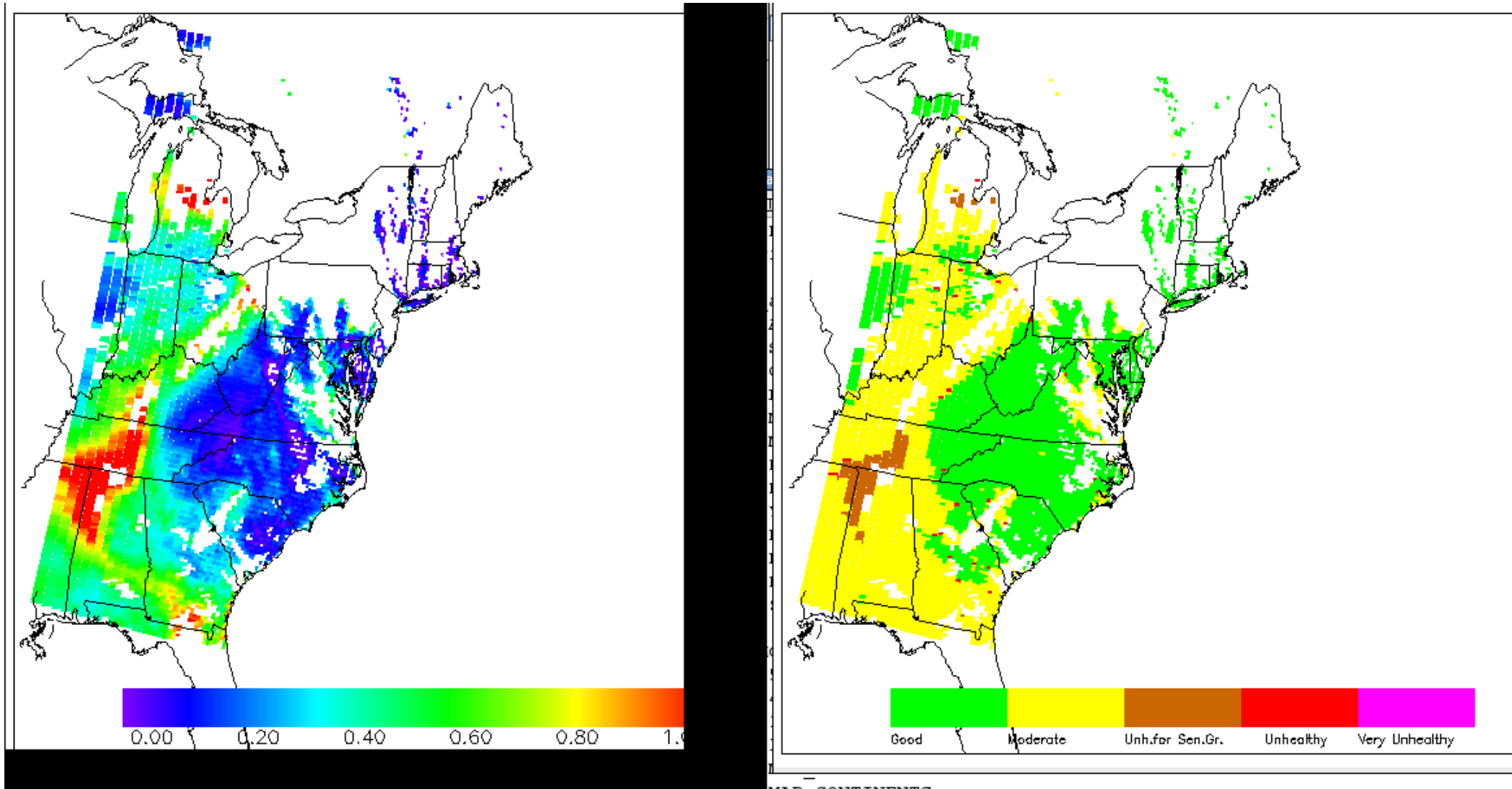
```
98 #asks user if they want to set PM2.5 calculation parameters
99 user_input=input('\nWould you like to enter a slope and intercept for PM 2.5 calculation?')
100 if user_input == 'Y' or user_input == 'y':
101     slope=input('Please enter a slope: ')
102     intercept=input('Please enter an intercept: ')
103 else:
104     #if not, choose the following:
105     slope=29.4
106     intercept=8.8
107 valid_data=data*scale_factor
108 pm25=float(slope)*valid_data+float(intercept)
109
110
111
112 #Asks user if they would like to see a map
113 is_map=input('\nWould you like to create a map of this d
114 #if user would like a map, view it
115 if is_map == 'Y' or is_map == 'y':
116     #turn fillvalues to NaN
117     data=pm25.astype(float)
118     data[np.logical_and(data>=0,data <= 12)]=0
119     data[np.logical_and(data>12,data <= 35.4)]=1
120     data[np.logical_and(data>35.4,data <= 55.4)]=2
121     data[np.logical_and(data>55.4,data <= 150.4)]=3
122     data[np.logical_and(data>150.4,data <= 250.4)]=4
123     data[data>250.4]=5
124     data[data < 0] = np.nan
125     #create the map
126     data = np.ma.masked_array(data, np.isnan(data))
127     m = Basemap(projection='cyl', resolution='l', llcrnrlat=min_lat, urcrnrlat = max_lat, llcrnrlon=min_lon,
128 m.drawcoastlines(linewidth=0.5)
129 m.drawparallels(np.arange(-90., 120., 5.), labels=[1, 0, 0, 0])
130 m.drawmeridians(np.arange(-180., 181., 5.), labels=[0, 0, 0, 1])
131 x, y = m(longitude, latitude)
132 my_cmap=LinearSegmentedColormap.from_list('mycmap', ['green','yellow','orange','red','purple','brown'],6
133 m.pcolormesh(x, y, data,cmap=my_cmap)
134 plt.clim(0,6)
135 #create colorbar
136 cb = m.colorbar()
137 cb.set_label('AQI Category')
138 cb.set_ticks([1.5, 1.5,2.5,3.5,4.5,5.5]) # force there to be only 7 ticks
139 cb.set_ticklabels(['Good', 'Moderate', 'Unhealthy for \nSensitive Groups','Unhealthy','Very Unhealthy','I
140
```

Cambie el “slope” (pendiente) y el “intercept” preprogramados

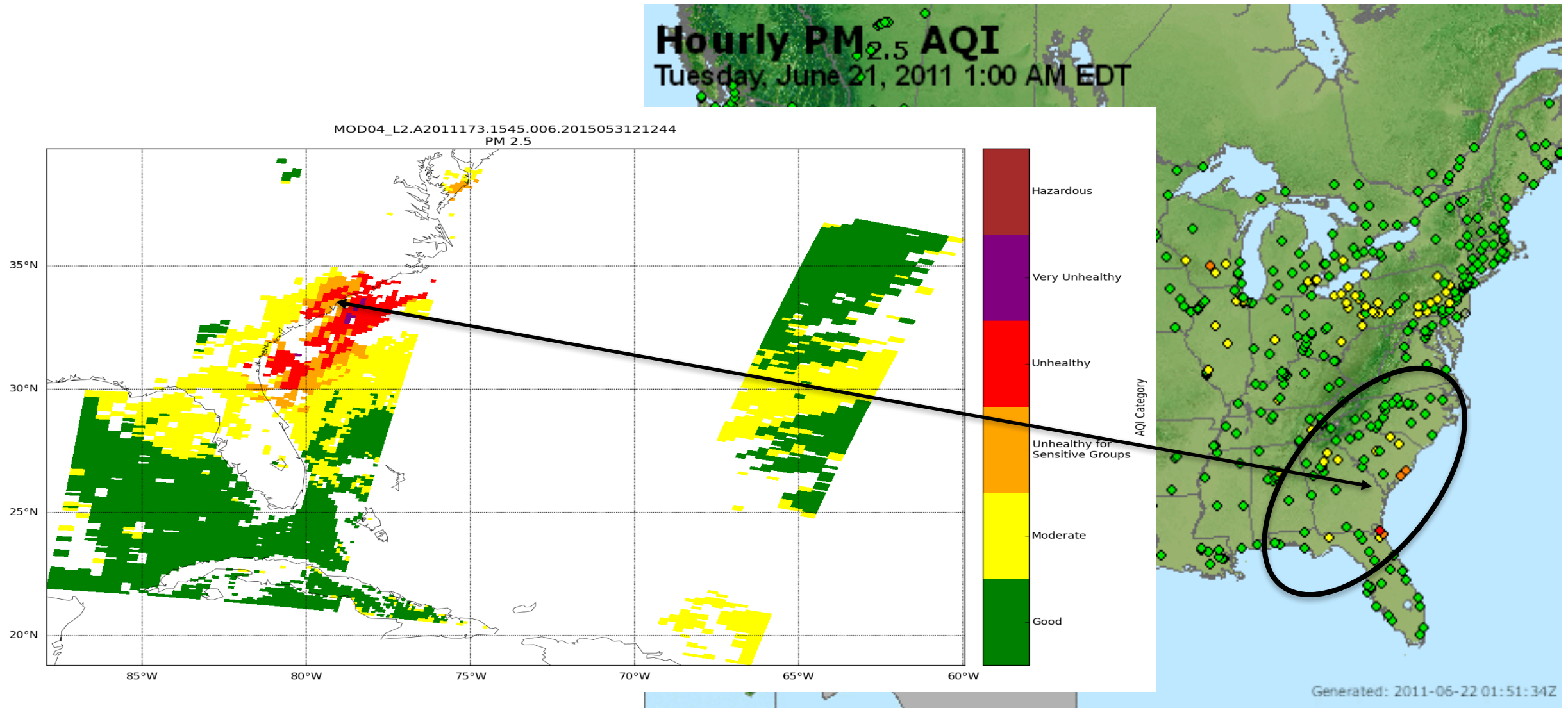
Cambie las categorías de calidad del aire

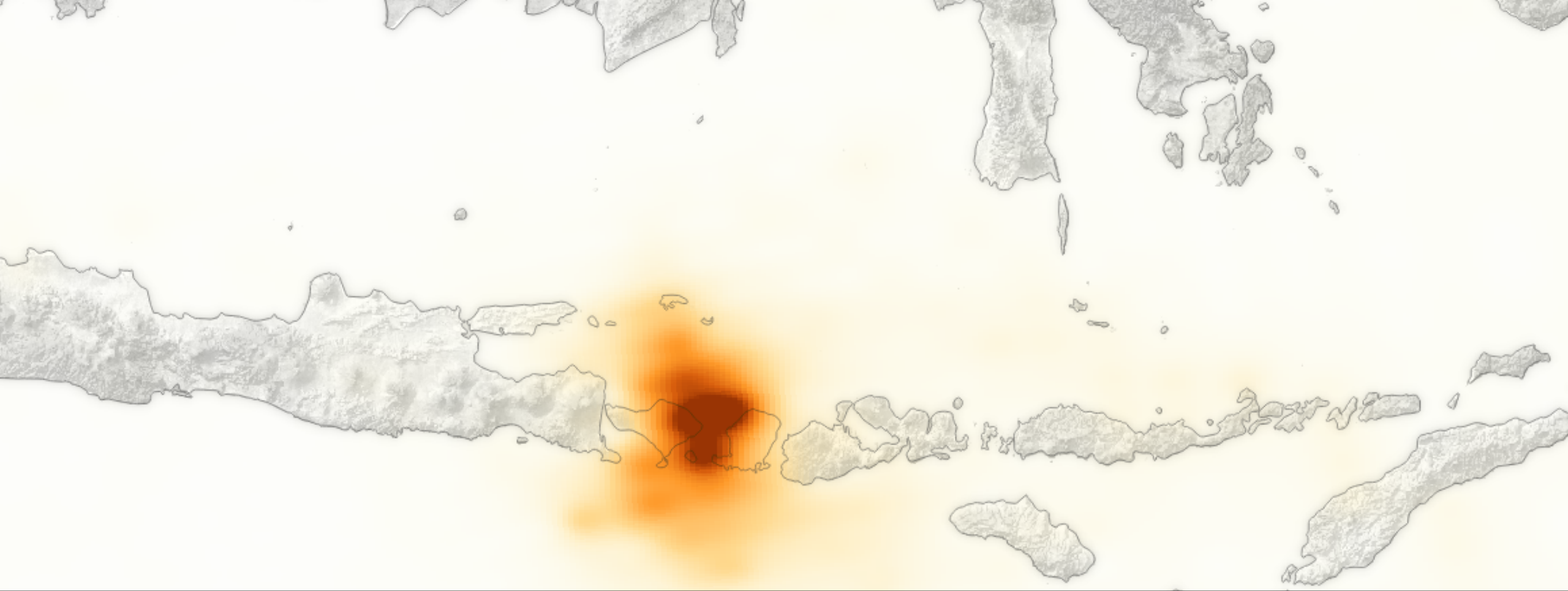


Aplicación: Convierta el AOD en $PM_{2.5}$ y Mapas de la Calidad del Aire



Aplicación – Compare Mapas Satelitales con Mapas Superficiaales





Preguntas