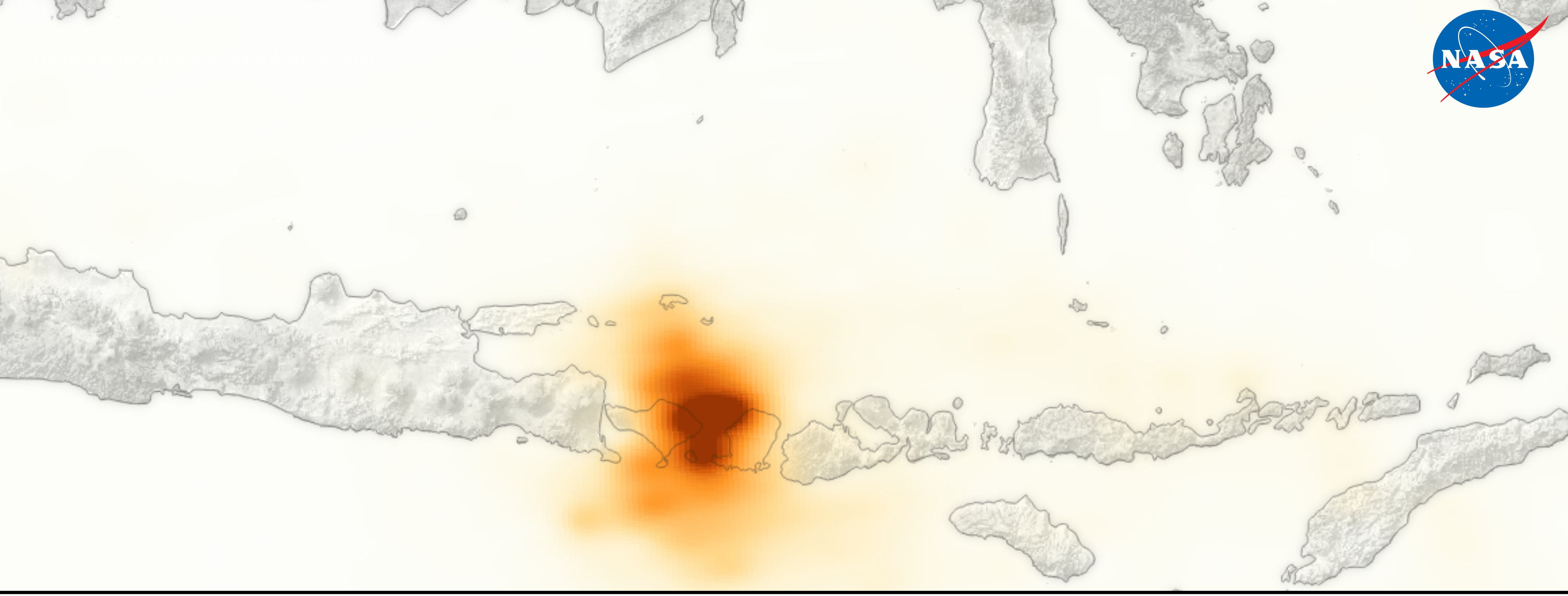
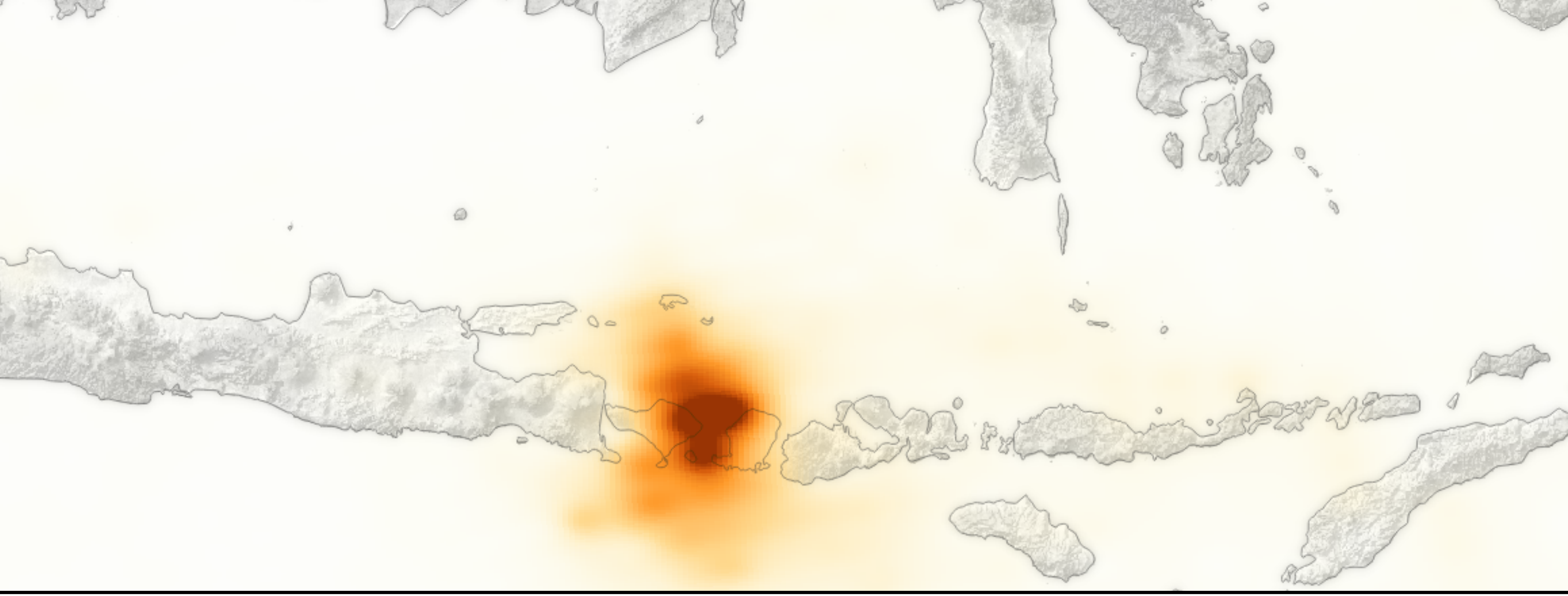
National Aeronautics and Space Administration



Read, Map and Extract MODIS Aerosol Data Using Python Scripts

Advanced Webinar: Data Analysis Tools for High Resolution Air Quality Satellite Datasets

Pawan Gupta & Melanie Follette-Cook, January 17-22, 2018



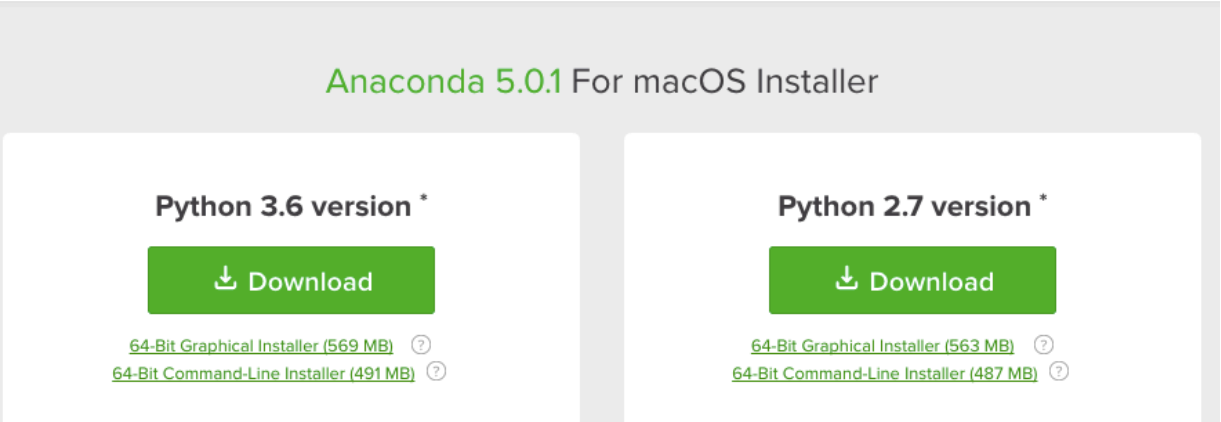
Computer and Python Requirements

Operating System

- Both session 2 & session 3 will cover python scripts to perform similar tasks.
- Session 2 is presented using a mac (apple) operating system.
- All the directory structure and python layout will be according to mac operating system.
- Session 3 is presented using a windows machine.
- Showing both operating systems will allow end-users to get a feeling of the layout of both operating systems.
- We expect all codes to run smoothly on mac, windows, and linux operating systems.

Computer Requirements

- Install Python 2.7 using Anaconda
- Install all required python packages
  - Package List (right)
- Test python and package installations using following python test code
  - test_python.py
- Download MODIS Data and Python Codes using following link
  - ARSET LINK ZIP FILE
- For more detail on the code, visit: https://arset.gsfc.nasa.gov/airquality/python-scripts-aerosol-data-sets-merra-modis-and-omi



Anaconda 5.0.1 For macOS Installer

Python 3.6 version *

64-Bit Graphical Installer (569 MB)

64-Bit Command-Line Installer (491 MB)

Python 2.7 version *

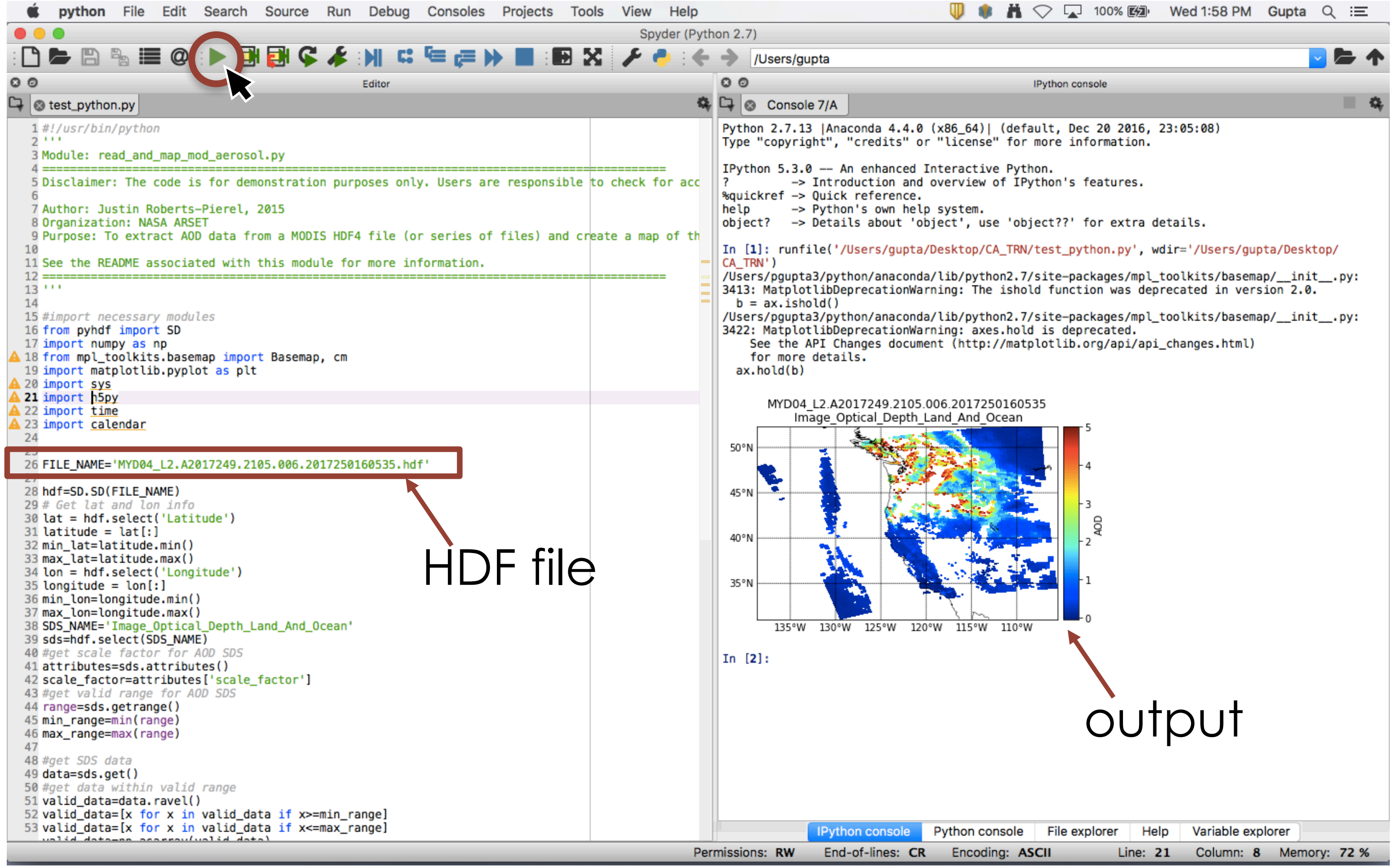
64-Bit Graphical Installer (563 MB)

64-Bit Command-Line Installer (487 MB)

- Python package list:
  - pyhdf
  - numpy
  - sys
  - mpl_toolkits.basemap
  - matplotlib
  - linearSegmented Colormap
  - h5py
  - time
  - calendar

Python Test

- Open the spyder editor inside Anaconda
- Open **test_python.py**
- Make sure the directory has the python code and HDF file
- Open the **ipython** console in the spyder
- Run the code using the **green arrow** on the top
- Output should be an image as shown



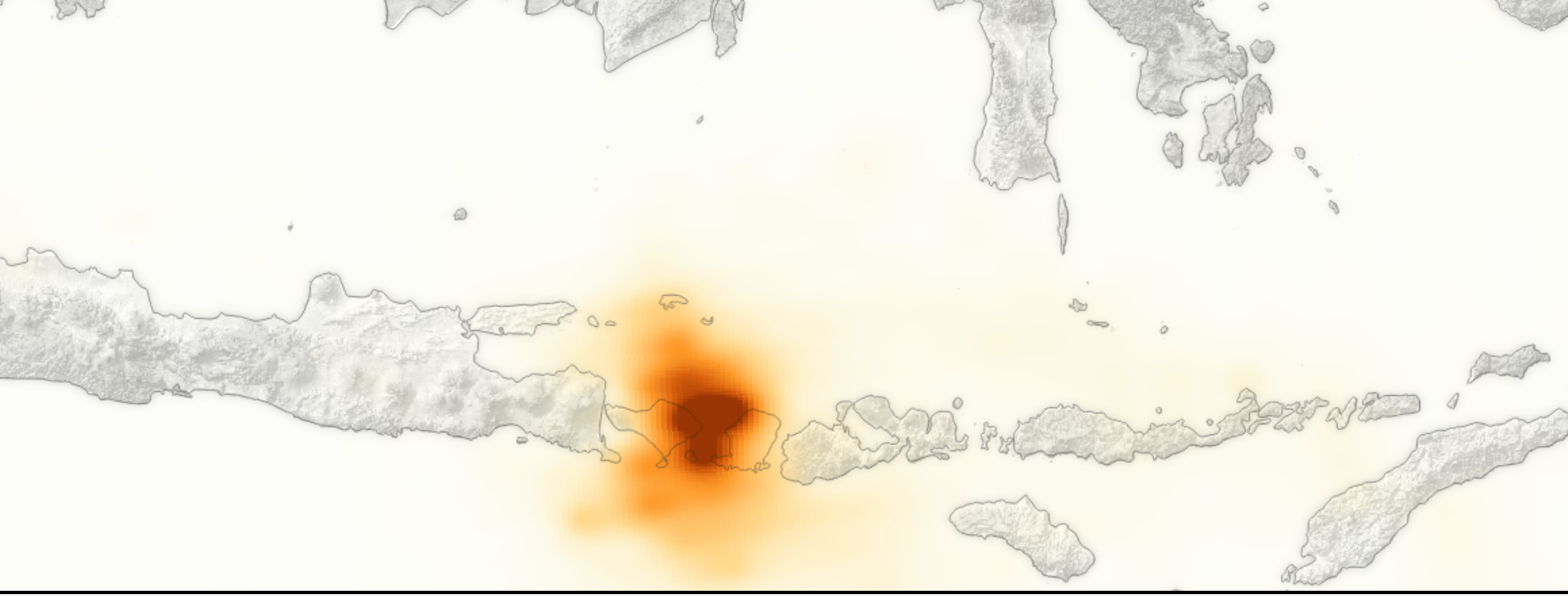
```
#!/usr/bin/python
'''
Module: read_and_map_mod_aerosol.py
==========================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for acc

Author: Justin Roberts-Pierel, 2015
Organization: NASA ARSET
Purpose: To extract AOD data from a MODIS HDF4 file (or series of files) and create a map of th

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

#import necessary modules
from pyhdf import SD
import numpy as np
from mpl_toolkits.basemap import Basemap, cm
import matplotlib.pyplot as plt
import sys
import h5py
import time
import calendar


FILE_NAME='MYD04_L2.A2017249.2105.006.2017250160535.hdf'

hdf=SD.SD(FILE_NAME)
# Get lat and lon info
lat = hdf.select('Latitude')
latitude = lat[:]
min_lat=latitude.min()
max_lat=latitude.max()
lon = hdf.select('Longitude')
longitude = lon[:]
min_lon=longitude.min()
max_lon=longitude.max()
SDS_NAME='Image_Optical_Depth_Land_And_Ocean'
sds=hdf.select(SDS_NAME)
#get scale factor for AOD SDS
attributes=sds.attributes()
scale_factor=attributes['scale_factor']
#get valid range for AOD SDS
range=sds.getrange()
min_range=min(range)
max_range=max(range)

#get SDS data
data=sds.get()
#get data within valid range
valid_data=data.ravel()
valid_data=[x for x in valid_data if x>=min_range]
valid_data=[x for x in valid_data if x<=max_range]
valid_data=np.asarray(valid_data)
```

HDF file

```
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/test_python.py', wdir='/Users/gupta/Desktop/
CA_TRN')
/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)

In [2]:
```

output

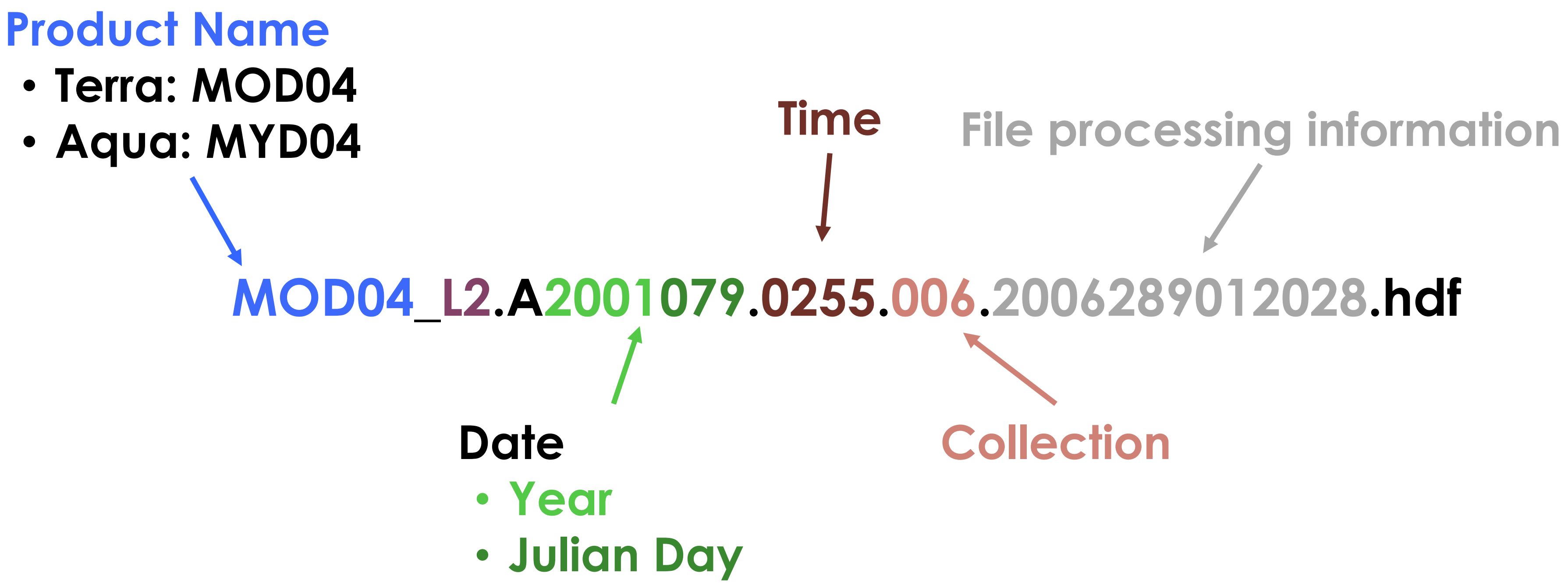
Permissions: RW   End-of-lines: CR   Encoding: ASCII   Line: 21   Column: 8   Memory: 72 %



Know your Data

Understanding a MODIS File Name

Level 2, 10 km, Aerosol Product



Product Name

- Terra: MOD04
- Aqua: MYD04

Time

File processing information

MOD04_L2.A2001079.0255.006.2006289012028.hdf

Date

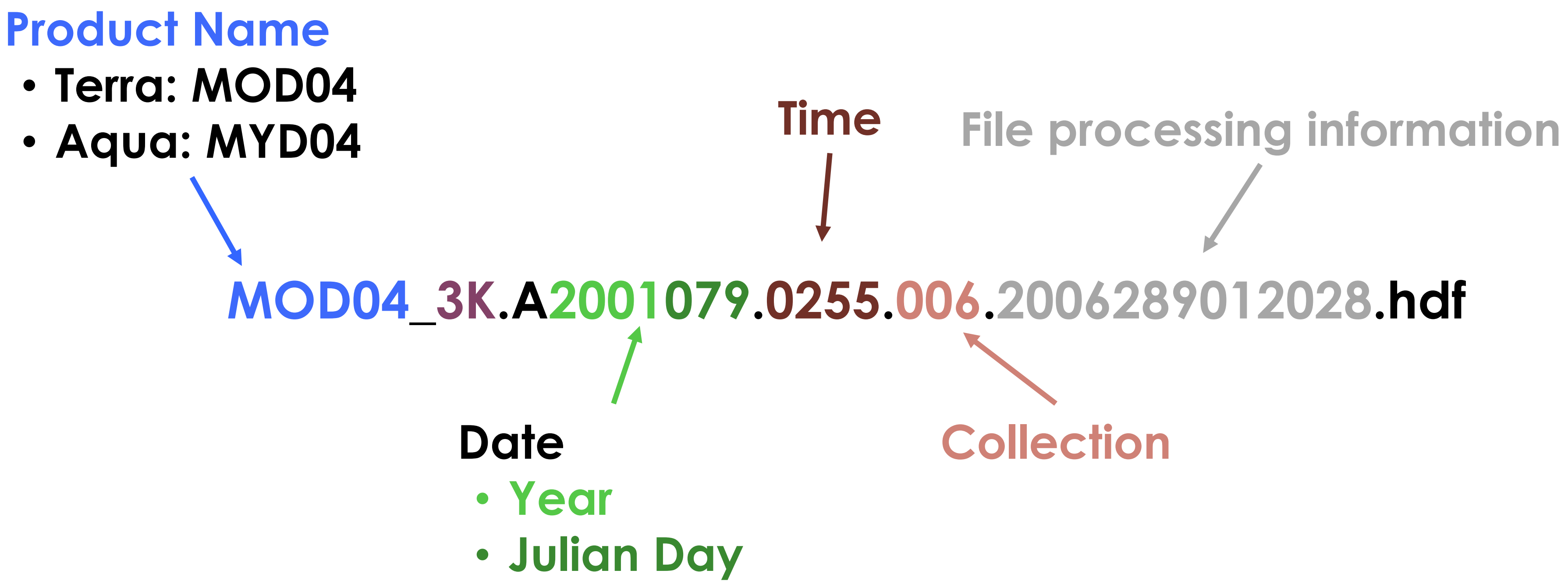
- Year
- Julian Day

Collection

HDFLook, Panoply, IDL, Python, Fortran, MatLab, and more can be used to read the data

Understanding a MODIS File Name

Level 2, 3 km, Aerosol Product



Product Name

- Terra: MOD04
- Aqua: MYD04

Time

File processing information

MOD04_3K.A2001079.0255.006.2006289012028.hdf

Date

- Year
- Julian Day

Collection

HDFLook, Panoply, IDL, Python, Fortran, MatLab, and more can be used to read the data

MODIS Aerosol Parameters (SDS)

- Optical_Depth_Land_and_Ocean
  - Retrieved using Dark Target Algorithm
  - Only high quality data
    - Over land QA = 3
    - Over ocean QA = 1, 2, 3
  - 10 km and 3km
- Dark_Target_Deep_Blue_Optical_Depth_550_Combined
  - Deep Blue & Dark Target Algorithm Merged Product
  - 10 km only
- Quality_Assurance_Land
  - Quality flag associated with DD product

Quality Assurance is Extremely Important

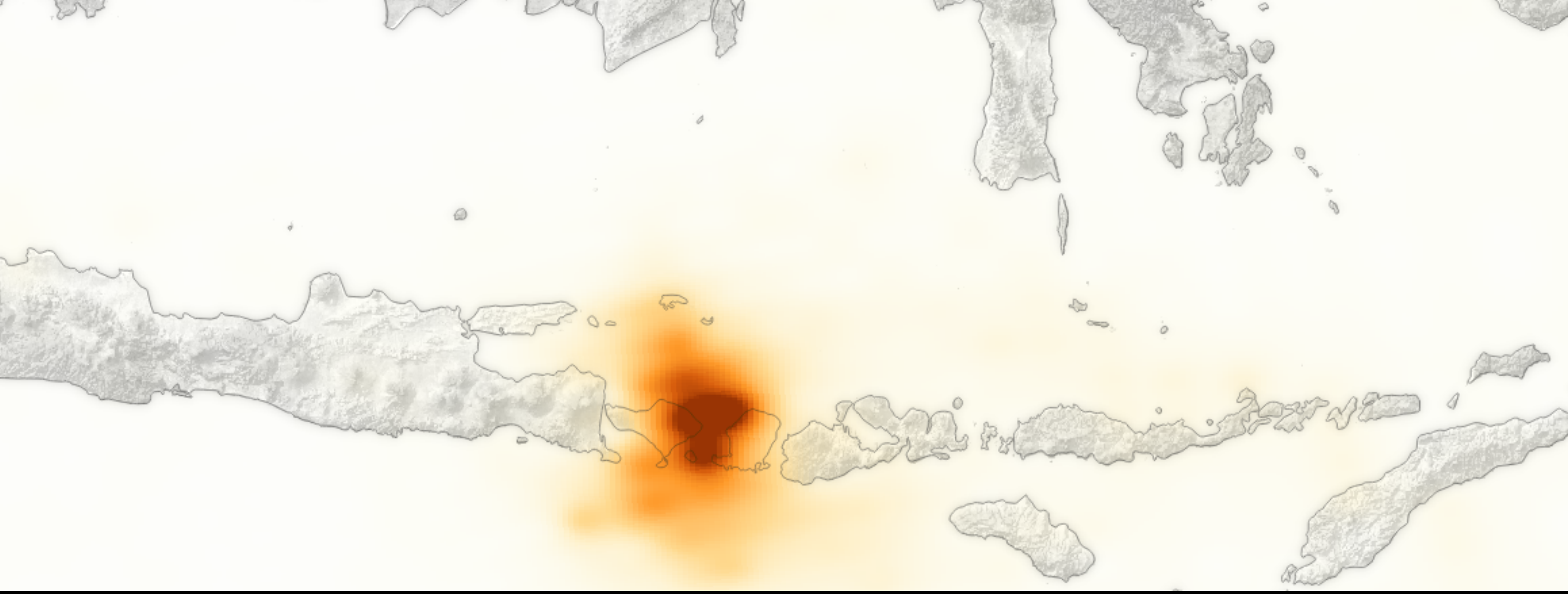
QA indicates confidence in the quality of the retrieval

Quality_Assurance_Ocean

- Scale is 0-3
- Recommended Ocean QA above 1, 2, 3
- Factors:
  - Number of pixels
  - Error fitting
  - **How close to glint**

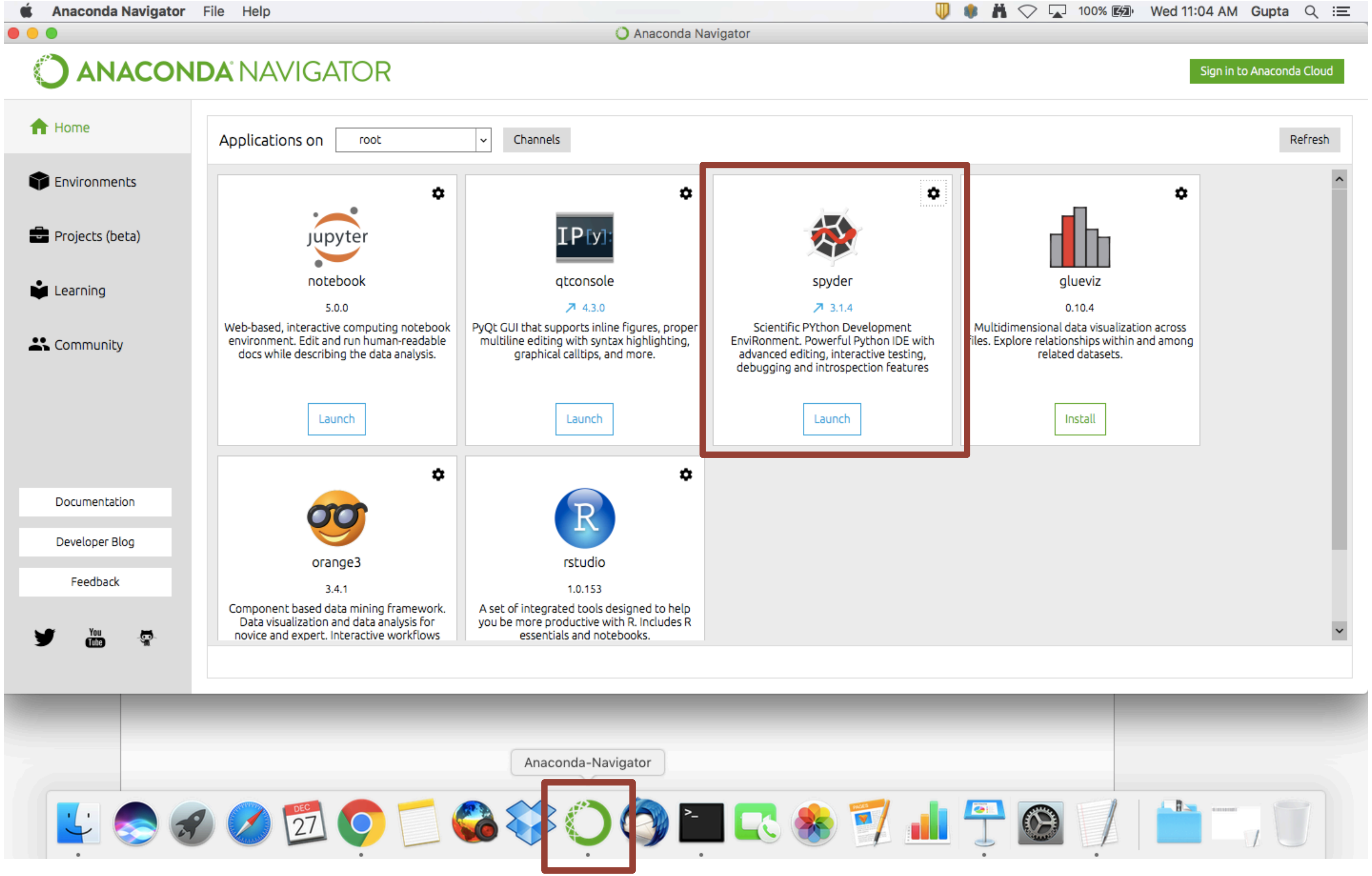
Quality_Assurance_Land

- Scale is 0-3
- Recommended Land QA of 3
- Factors:
  - Number of pixels
  - Error fitting
  - **Surface reflectance**



Getting Ready with Python

Anaconda & the Spyder Editor



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

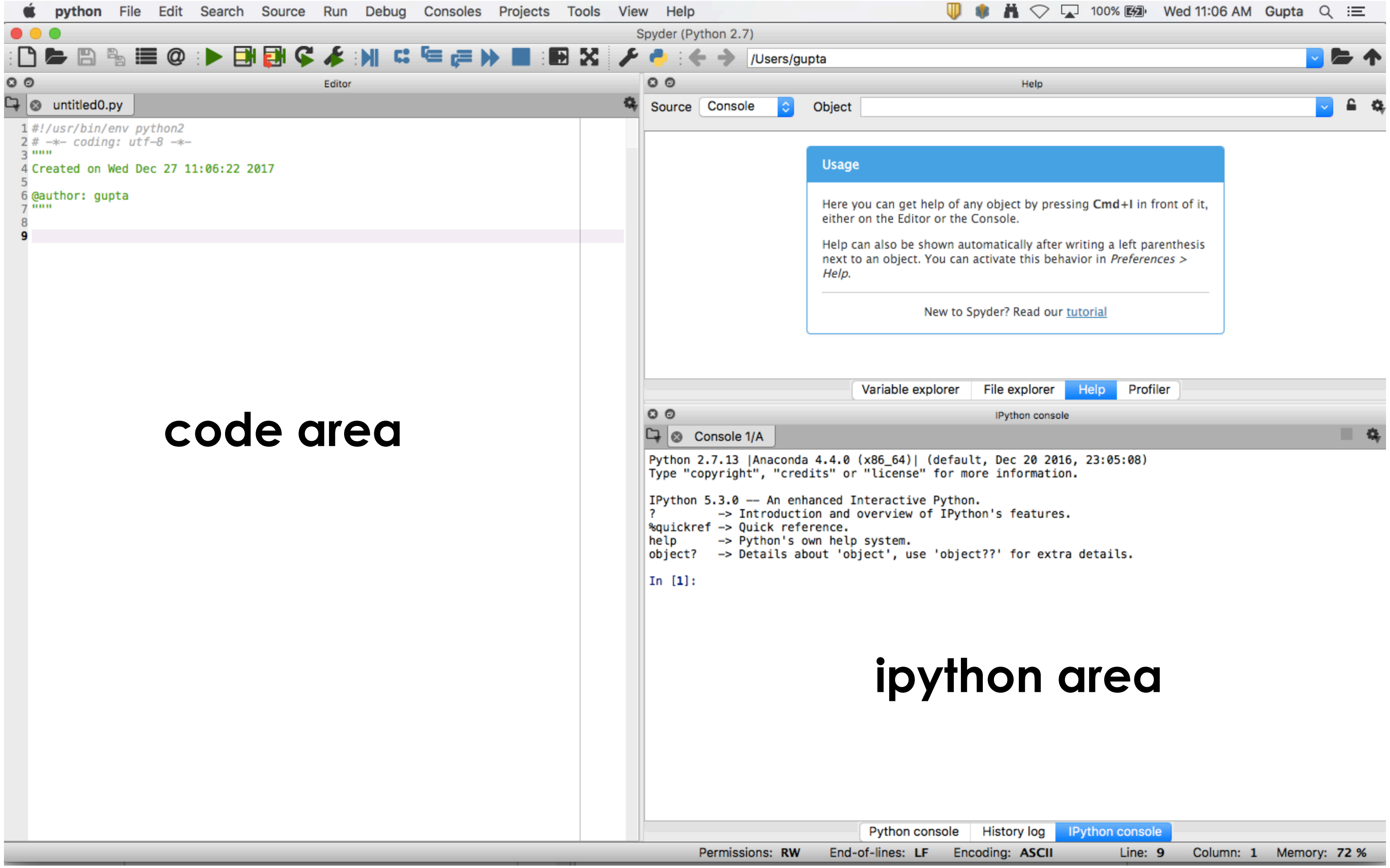
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Spyder View


python File Edit Search Source Run Debug Consoles Projects Tools View Help
100% Wed 11:06 AM Gupta
Spyder (Python 2.7)
/Users/gupta
Editor
untitled0.py
Help
Source Console Object
Variable explorer File explorer Help Profiler
IPython console
Console 1/A
Python console History log IPython console
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 9 Column: 1 Memory: 72 %
code area
ipython area

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 27 11:06:22 2017

@author: gupta
"""

```

Usage

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

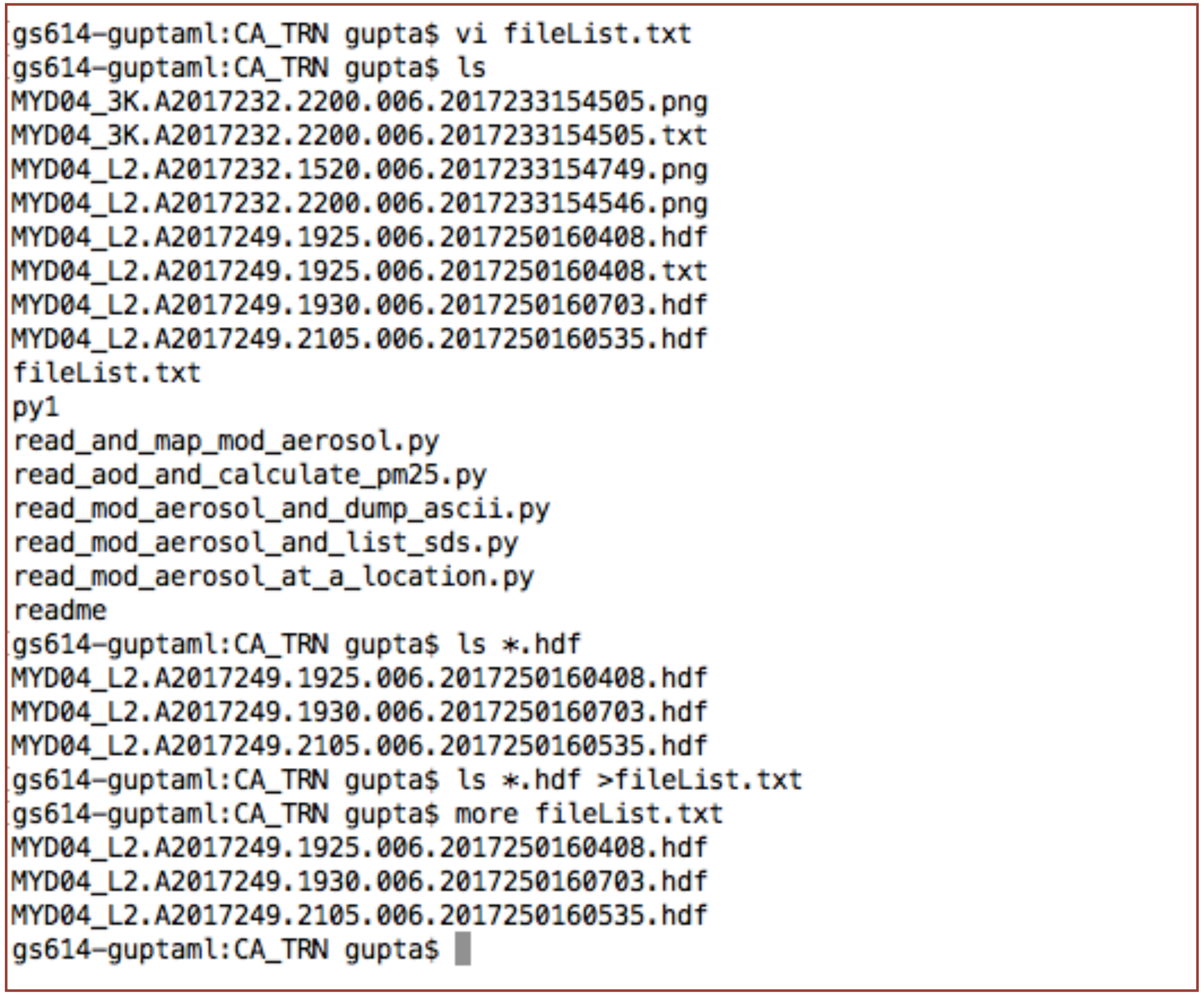
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

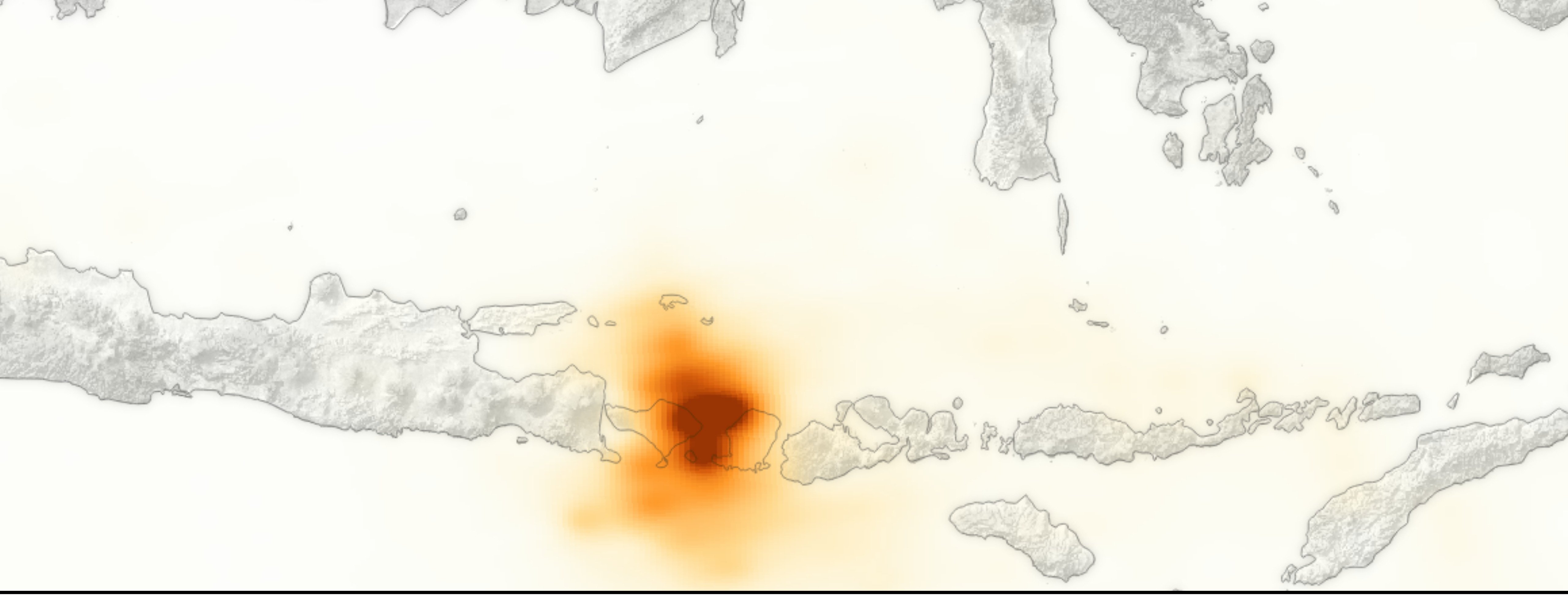
```
Python 2.7.13 |Anaconda 4.4.0 (x86_64)| (default, Dec 20 2016, 23:05:08)
Type "copyright", "credits" or "license" for more information.

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]:
```

Current Directory View & File List

- Create a list of HDF files ‘fileList.txt’
- The directory should have
  – All the python codes
  – All the HDF data files
  – A list of HDF files named as ‘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$
```

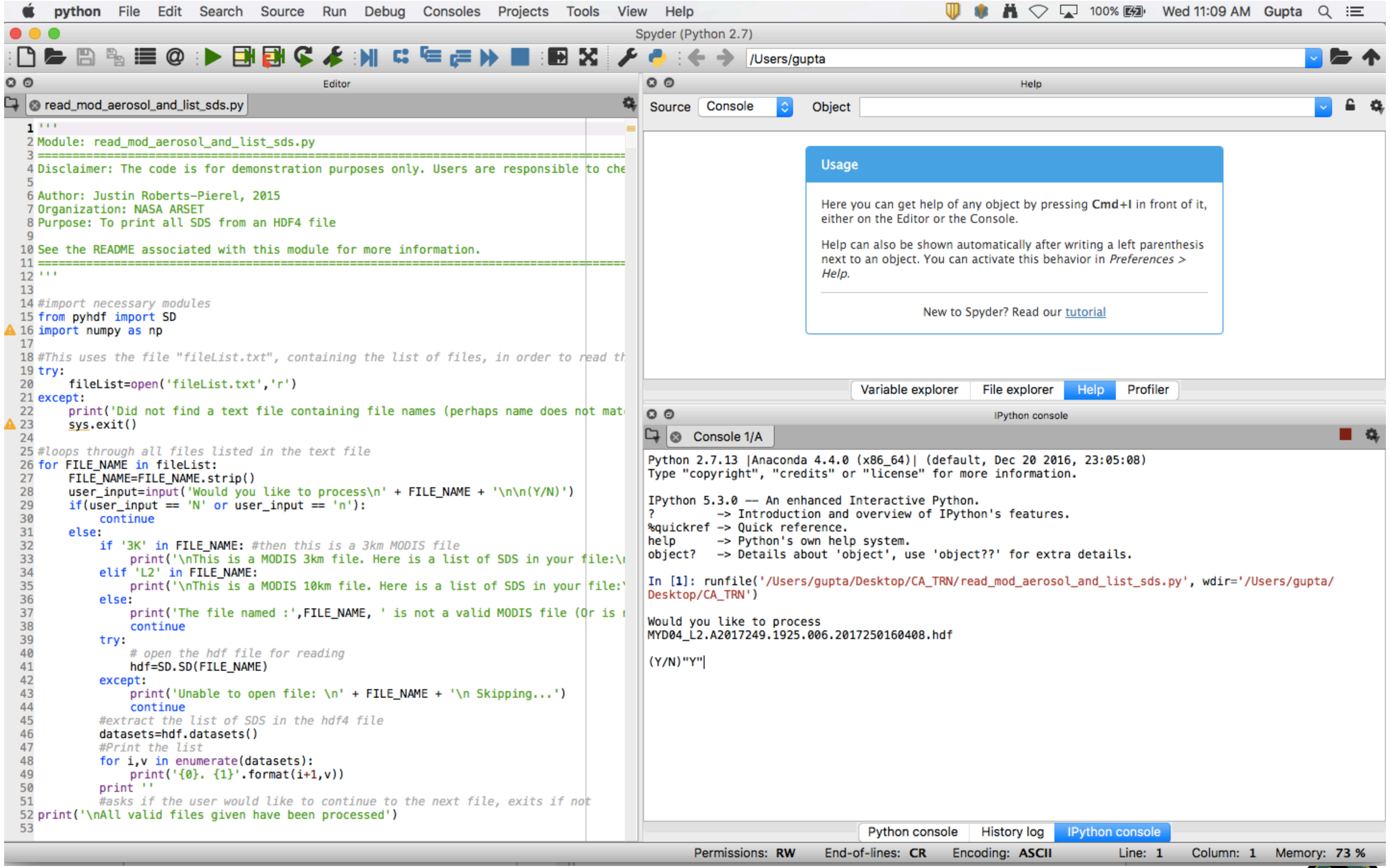


Read a MODIS Aerosol File (HDF) and Print SDS List

Print Scientific Data Sets (SDSs)

read_mod_aerosol_and_list_sds.py

- **Purpose**: read a MODIS aerosol level 2 data file in HDF format and print all the **Scientific Data Sets** (SDS)
- The code works for both 10 km and 3 km products


python File Edit Search Source Run Debug Consoles Projects Tools View Help
100% Wed 11:09 AM Gupta
Spyder (Python 2.7)
/Users/gupta
Editor
read_mod_aerosol_and_list_sds.py
Help
Source Console Object
Variable explorer File explorer Help Profiler
IPython console
Console 1/A
Python console History log IPython console
Permissions: RW End-of-lines: CR Encoding: ASCII Line: 1 Column: 1 Memory: 73 %

```python
'''
Module: read_mod_aerosol_and_list_sds.py
=================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to che

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 th
try:
    fileList=open('fileList.txt','r')
except:
    print('Did not find a text file containing file names (perhaps name does not mat
    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:\r
        elif 'L2' in FILE_NAME:
            print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:'
        else:
            print('The file named :',FILE_NAME, ' is not a valid MODIS file (Or is r
            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')
```

Usage

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

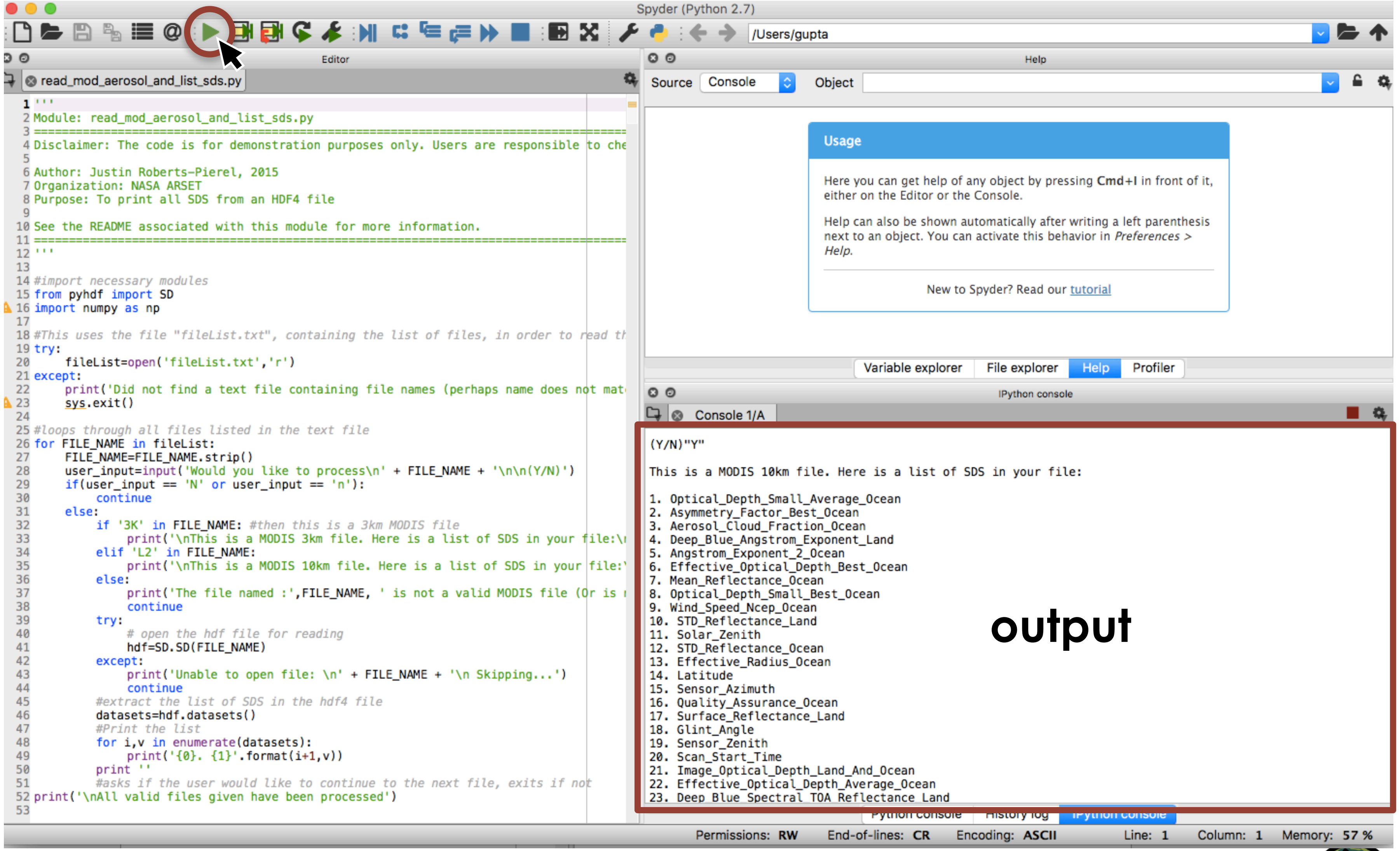
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
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_list_sds.py', wdir='/Users/gupta/
Desktop/CA_TRN')

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

(Y/N)"Y"
```

Running and Output

- Click the green arrow to run the code
- The code will process all the files in the **fileList.txt** one-by-one
- Follow the instructions in the **ipython** terminal (i.e. enter ‘Y’ or ‘N’ when prompted and hit enter)



```
'''
Module: read_mod_aerosol_and_list_sds.py
==========================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to che

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 th
try:
    fileList=open('fileList.txt','r')
except:
    print('Did not find a text file containing file names (perhaps name does not mat
    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:\
        elif 'L2' in FILE_NAME:
            print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:'
        else:
            print('The file named :',FILE_NAME, ' is not a valid MODIS file (Or is 
            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')
```

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*.

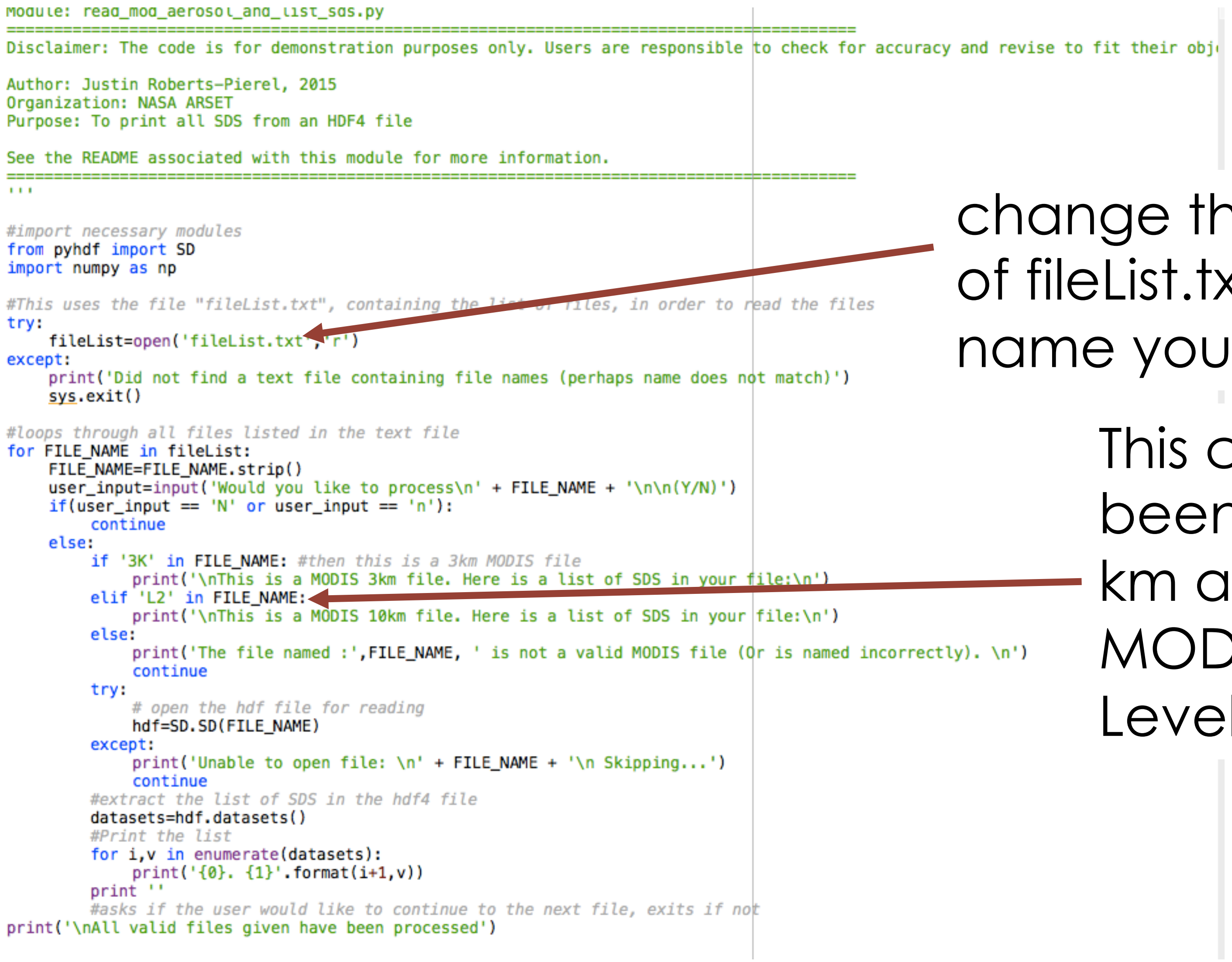
New to Spyder? Read our tutorial

```
(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
```

output

Permissions: **RW** End-of-lines: **CR** Encoding: **ASCII** Line: **1** Column: **1** Memory: **57 %**

Editing the Code



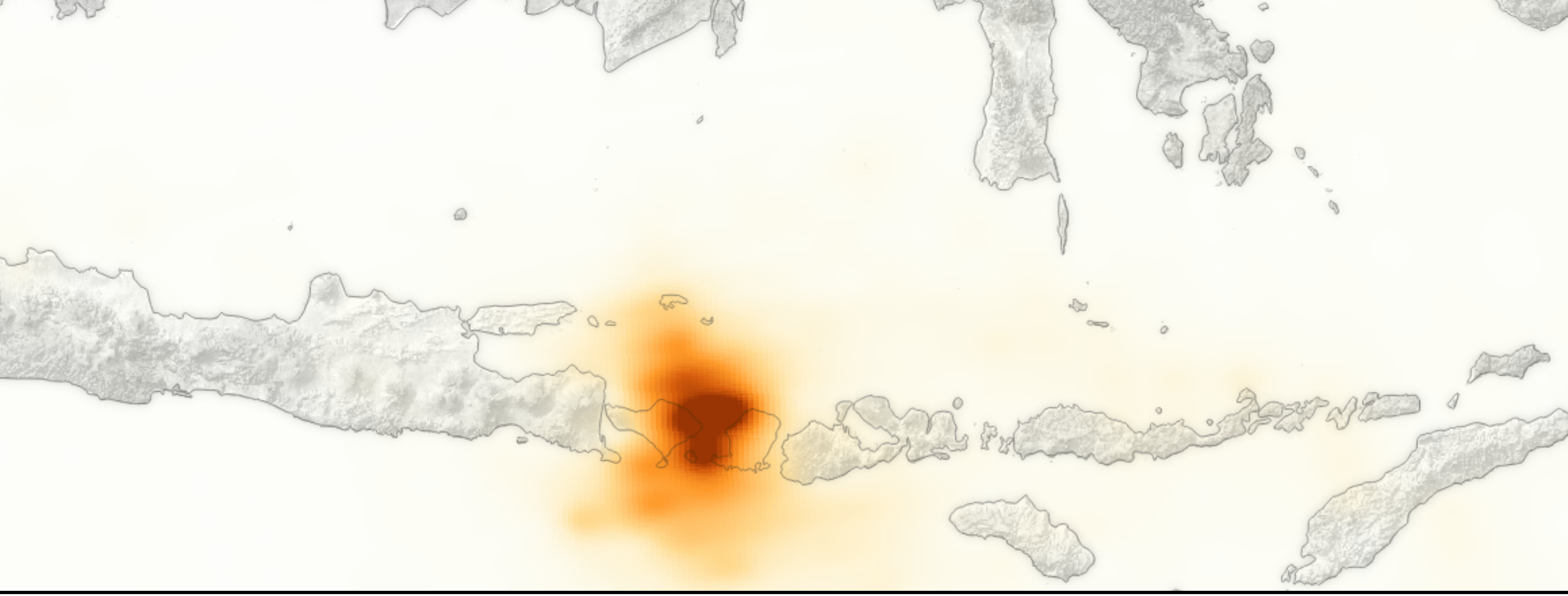
```
Module: read_mod_aerosol_and_list_sds.py
=====================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their obj

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')
```

change the name of fileList.txt to any name you'd like

This code has been tested for 3 km and 10 km MODIS aerosol Level 2 data files

Applications

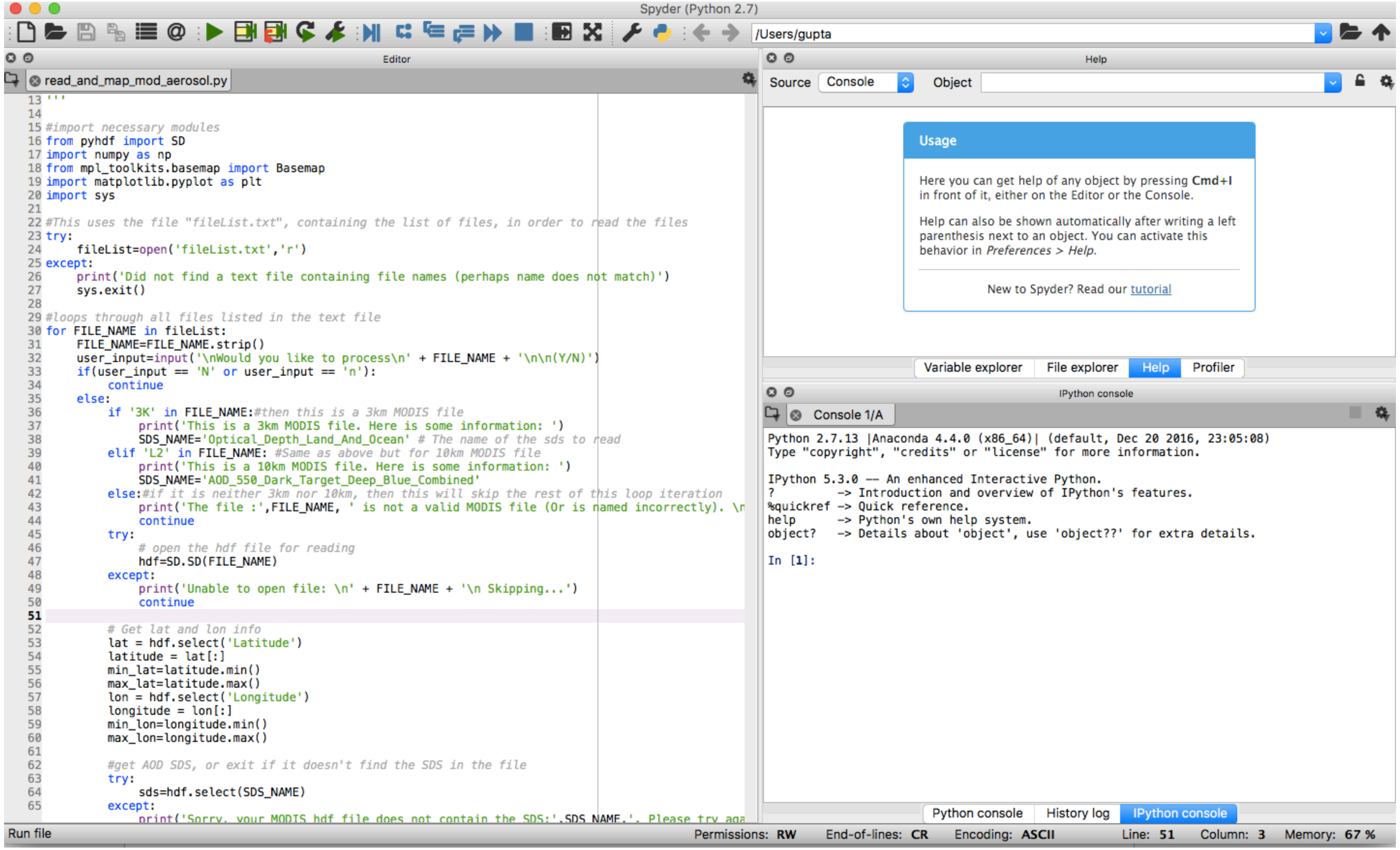
- MODIS Level 2 aerosol data are provided in HDF files
- Each HDF file contains several geophysical parameters
- Special codes and tools are required to open HDF files
- This code helps users see the name of the available SDSs inside an HDF file for further analysis



Map Aerosol Optical Depth

Plot and save a map of MODIS AOD

read_and_map_mod_aerosol.py



```
'''

#import necessary modules
from pyhdf import SD
import numpy as np
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import sys

#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('\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

        # Get lat and lon info
        lat = hdf.select('Latitude')
        latitude = lat[:]
        min_lat=latitude.min()
        max_lat=latitude.max()
        lon = hdf.select('Longitude')
        longitude = lon[:]
        min_lon=longitude.min()
        max_lon=longitude.max()

        #get AOD SDS, or exit if it doesn't find the SDS in the file
        try:
            sds=hdf.select(SDS_NAME)
        except:
            print('Sorry, your MODIS hdf file does not contain the SDS:',SDS_NAME,'. Please try aga
```

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*.

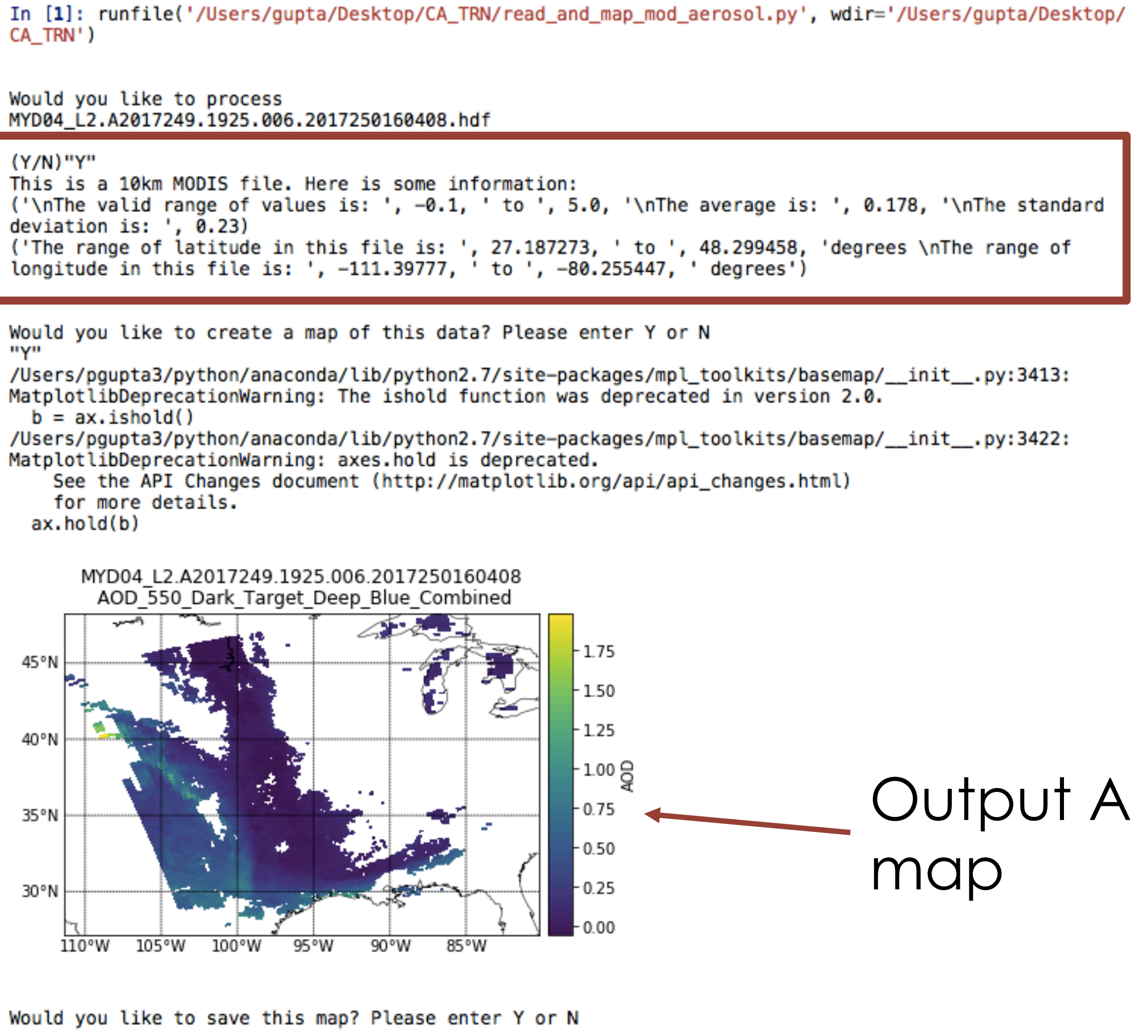
New to Spyder? Read our tutorial

```
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]:
```

Permissions: RW   End-of-lines: CR   Encoding: ASCII   Line: 51   Column: 3   Memory: 67 %

Running and Output

AOD statistics



```
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:
('\nThe 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 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)
```

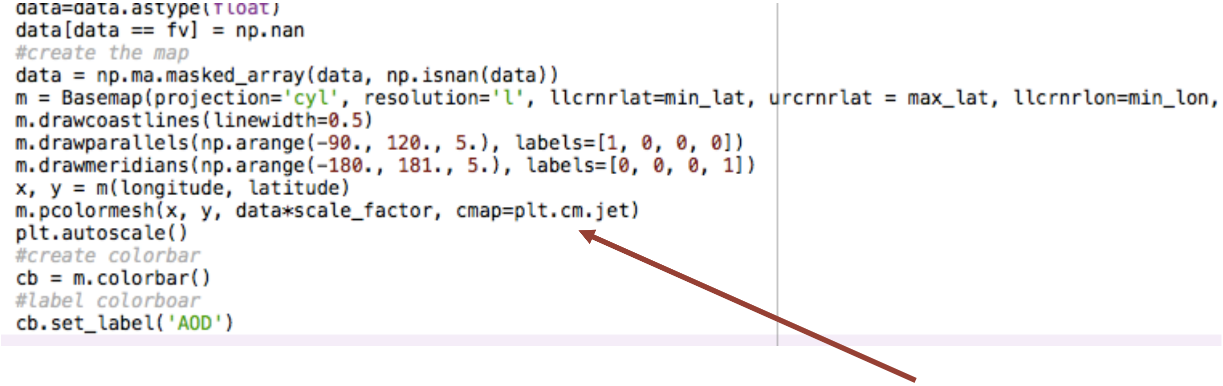
MYD04_L2.A2017249.1925.006.2017250160408
AOD_550_Dark_Target_Deep_Blue_Combined
45°N
40°N
35°N
30°N
110°W
105°W
100°W
95°W
90°W
85°W
AOD
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

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

Output AOD map

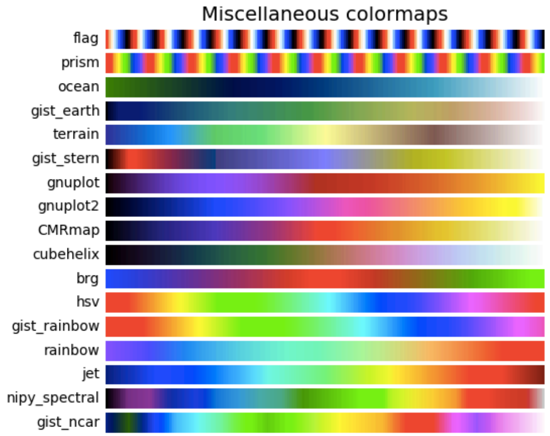
Editing the Code

Change the Color Scale



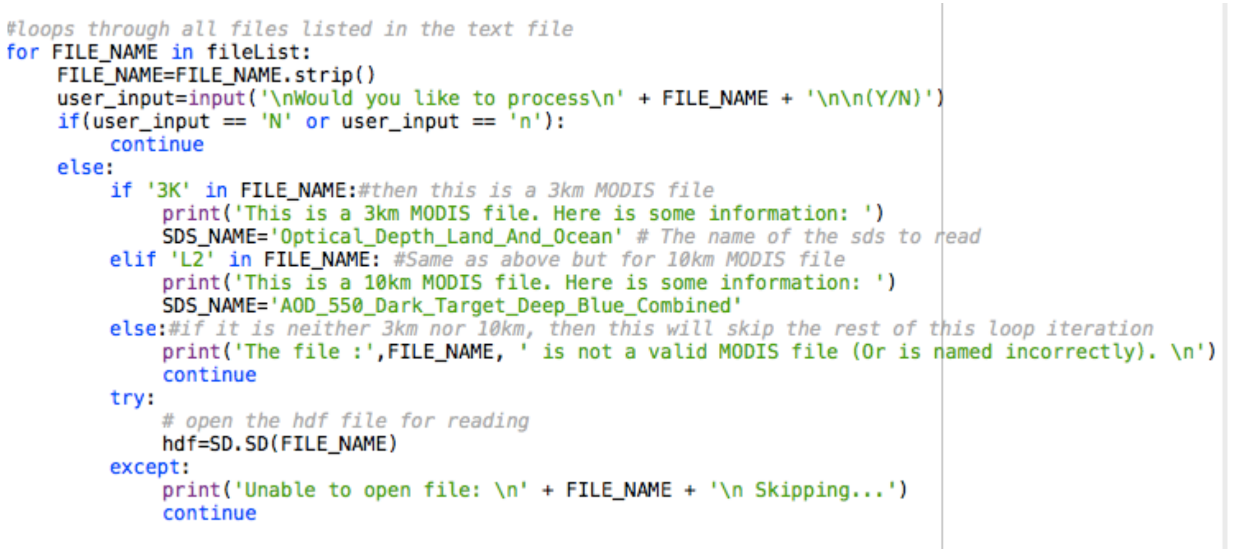
```
data=data.astype(float)
data[data == fv] = np.nan
#create the map
data = np.ma.masked_array(data, np.isnan(data))
m = Basemap(projection='cyl', resolution='l', llcrnrlat=min_lat, urcrnrlat = max_lat, llcrnrlon=min_lon,
m.drawcoastlines(linewidth=0.5)
m.drawparallels(np.arange(-90., 120., 5.), labels=[1, 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 colorboar
cb.set_label('AOD')
```

(png, pdf)



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

Change the SDS

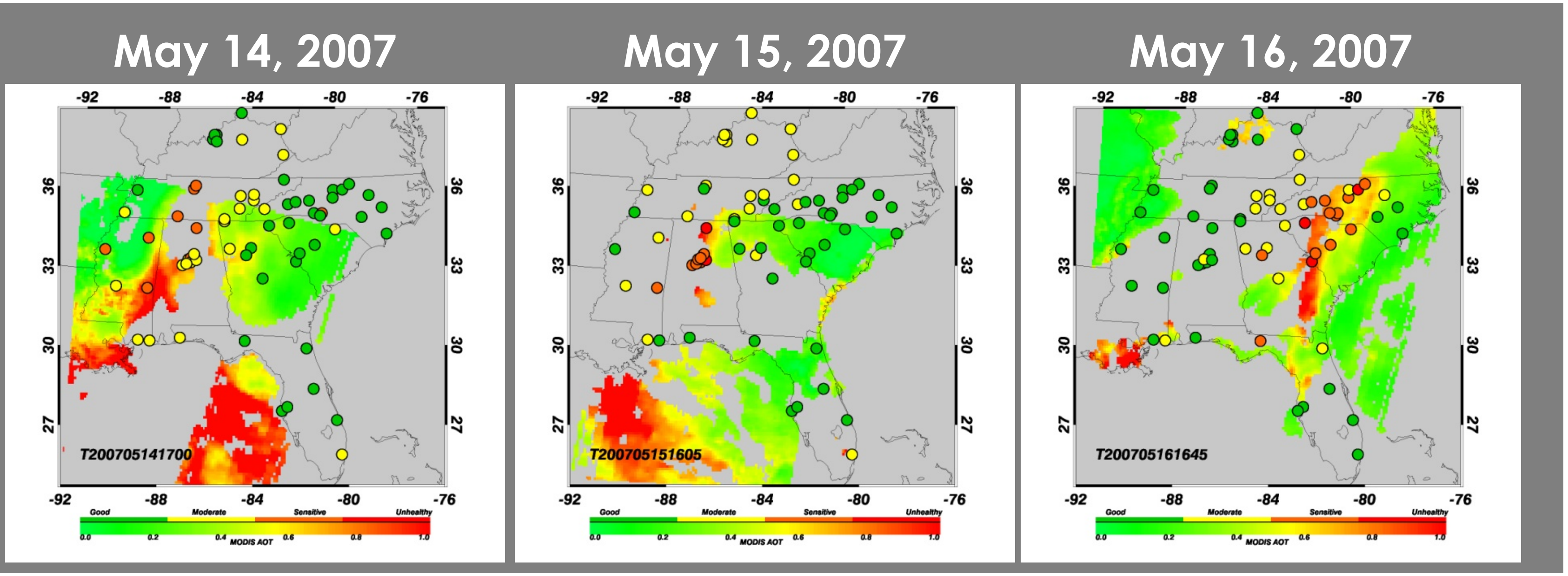


```
#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if(user_input == 'N' or user_input == 'n'):
        continue
    else:
        if '3K' in FILE_NAME:#then 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
```

Applications

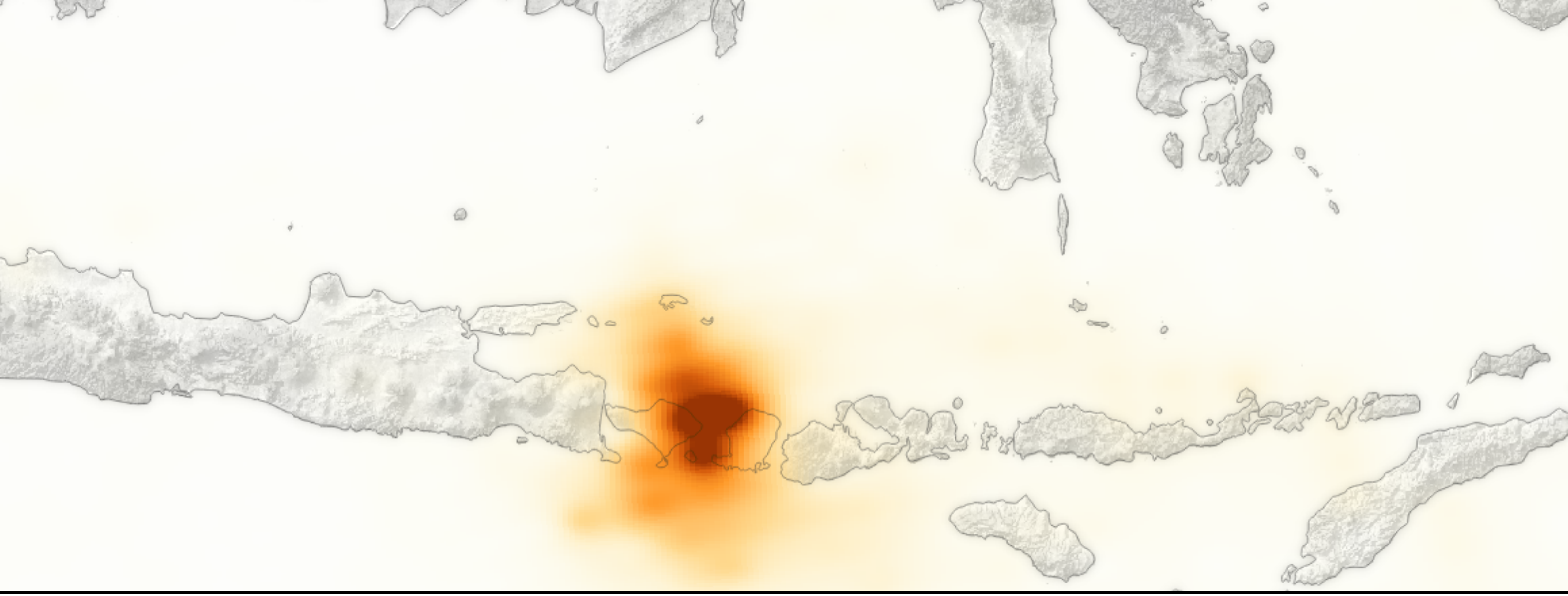
- This is a sample code to read and map the MODIS Level 2 aerosol data
- The code can be modified to address different mapping needs
- Users can create daily maps of AOD over certain regions and start analyzing changes over time
- AOD maps can also help identify regions with high pollution levels

Example:
High AOD values from smoke show good agreement with surface monitors (circles).


May 14, 2007

May 15, 2007

May 16, 2007

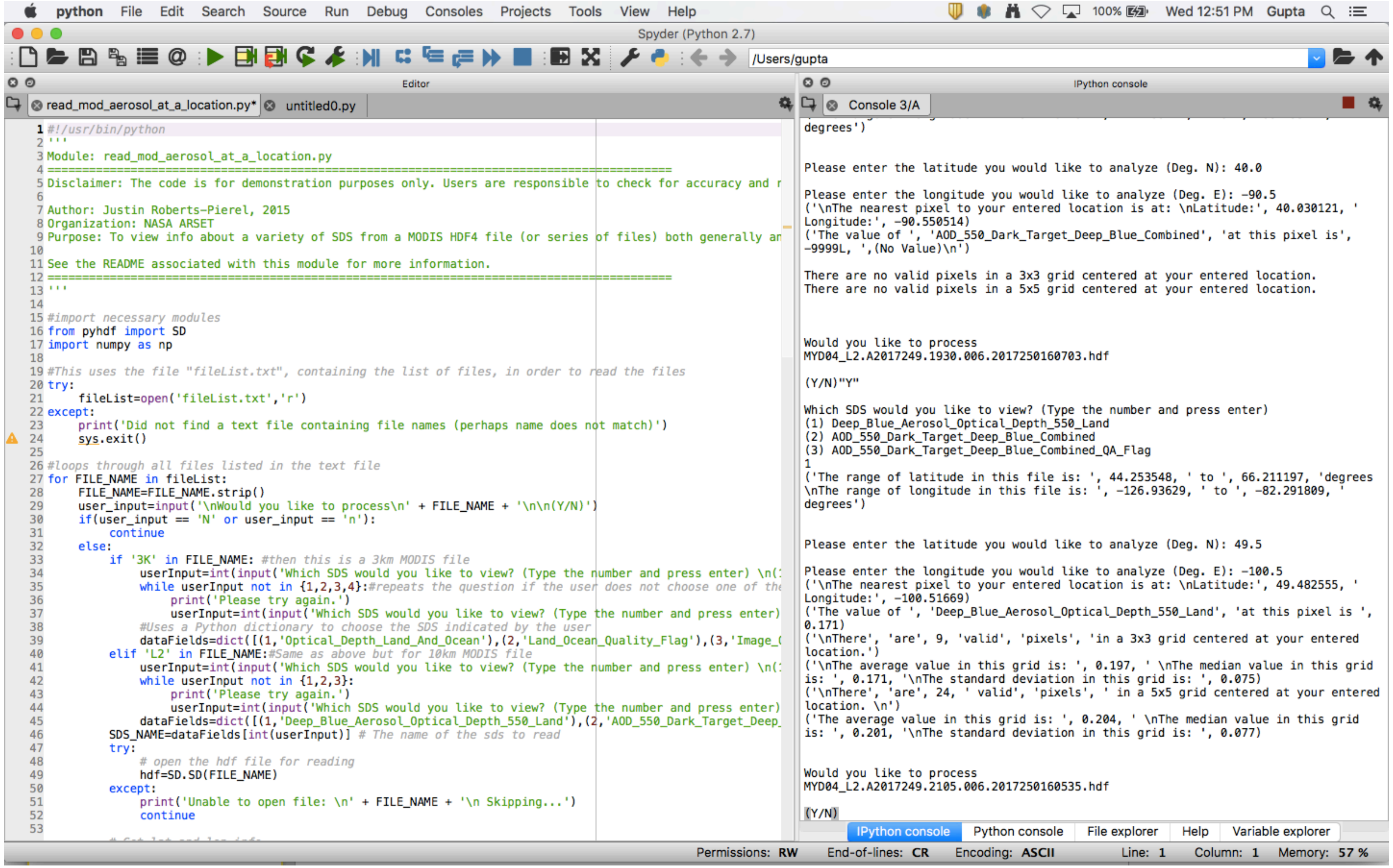


Extract AOD at a Surface Station

Extract AOD Values at a given location

read_mod_aerosol_at_a_location.py

- **Purpose**: read a MODIS aerosol level 2 data file in HDF format and extract AOD values at a given ground location
- The code works for both 10 km and 3 km products



```python
#!/usr/bin/python
'''
Module: read_mod_aerosol_at_a_location.py
==========================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and r

Author: Justin Roberts-Pierel, 2015
Organization: NASA ARSET
Purpose: To view info about a variety of SDS from a MODIS HDF4 file (or series of files) both generally an

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('\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
            while userInput not in {1,2,3,4}:#repeats the question if the user does not choose one of the
                print('Please try again.')
                userInput=int(input('Which SDS would you like to view? (Type the number and press enter)
            #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_(
        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
            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)
            dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Dark_Target_Deep_
        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
```

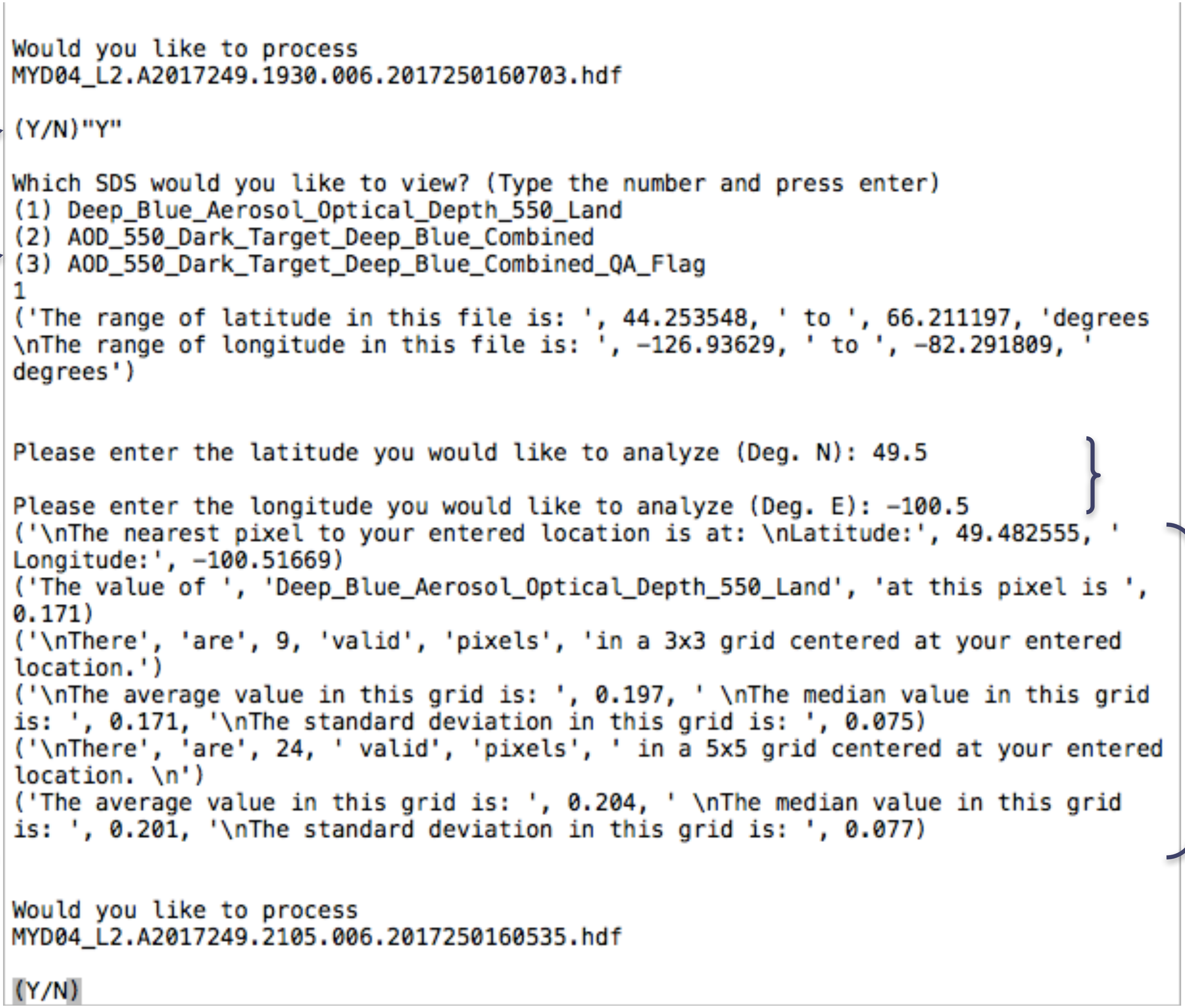
```
degrees')


Please enter the latitude you would like to analyze (Deg. N): 40.0

Please enter the longitude you would like to analyze (Deg. E): -90.5
('\nThe nearest pixel to your entered location is at: \nLatitude:', 40.030121, '
Longitude:', -90.550514)
('The value of ', 'AOD_550_Dark_Target_Deep_Blue_Combined', 'at this pixel is',
-9999L, ',(No Value)\n')

There are no valid pixels in a 3x3 grid centered at your entered location.
There are no valid pixels in a 5x5 grid centered at your entered location.




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
('\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)
('\nThere', 'are', 9, 'valid', 'pixels', 'in a 3x3 grid centered at your entered
location.')
('\nThe average value in this grid is: ', 0.197, ' \nThe median value in this grid
is: ', 0.171, '\nThe standard deviation in this grid is: ', 0.075)
('\nThere', 'are', 24, ' valid', 'pixels', ' in a 5x5 grid centered at your entered
location. \n')
('The average value in this grid is: ', 0.204, ' \nThe median value in this grid
is: ', 0.201, '\nThe standard deviation in this grid is: ', 0.077)


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

(Y/N)
```

Running and Output

Type “Y” to process file, “N” to skip

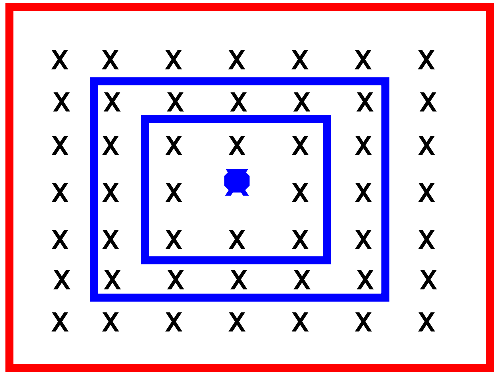
Select SDS

Lat & Lon of station

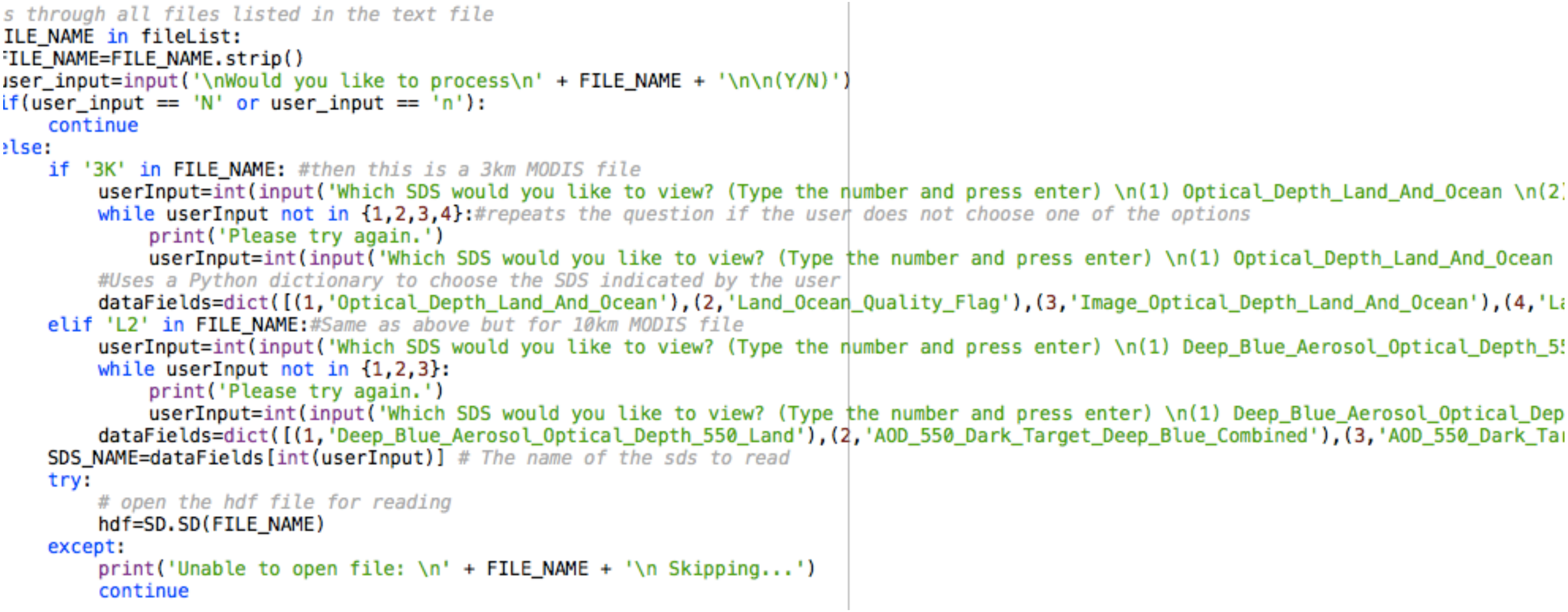


```
Would you like to process
MYD04_L2.A2017249.1930.006.2017250160703.hdf

(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
('\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)
('\nThere', 'are', 9, 'valid', 'pixels', 'in a 3x3 grid centered at your entered
location.')
('\nThe average value in this grid is: ', 0.197, ' \nThe median value in this grid
is: ', 0.171, '\nThe standard deviation in this grid is: ', 0.075)
('\nThere', 'are', 24, ' valid', 'pixels', ' in a 5x5 grid centered at your entered
location. \n')
('The average value in this grid is: ', 0.204, ' \nThe median value in this grid
is: ', 0.201, '\nThe standard deviation in this grid is: ', 0.077)


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

(Y/N)
```

Outputs

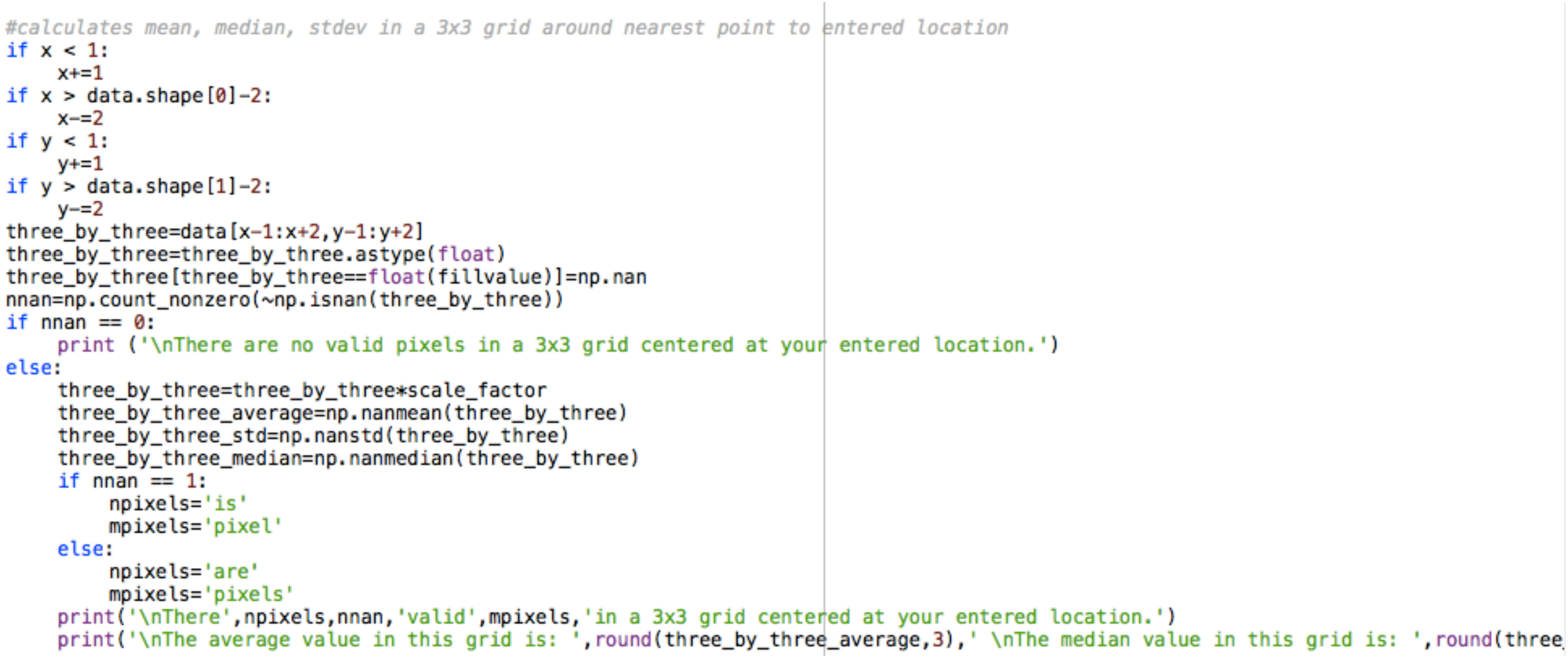


Editing the Code – Change the SDS



```
s through all files listed in the text file
ILE_NAME in fileList:
ILE_NAME=FILE_NAME.strip()
ser_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
if(user_input == 'N' or user_input == 'n'):
    continue
lse:
    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,'La
    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_Ta
    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
```

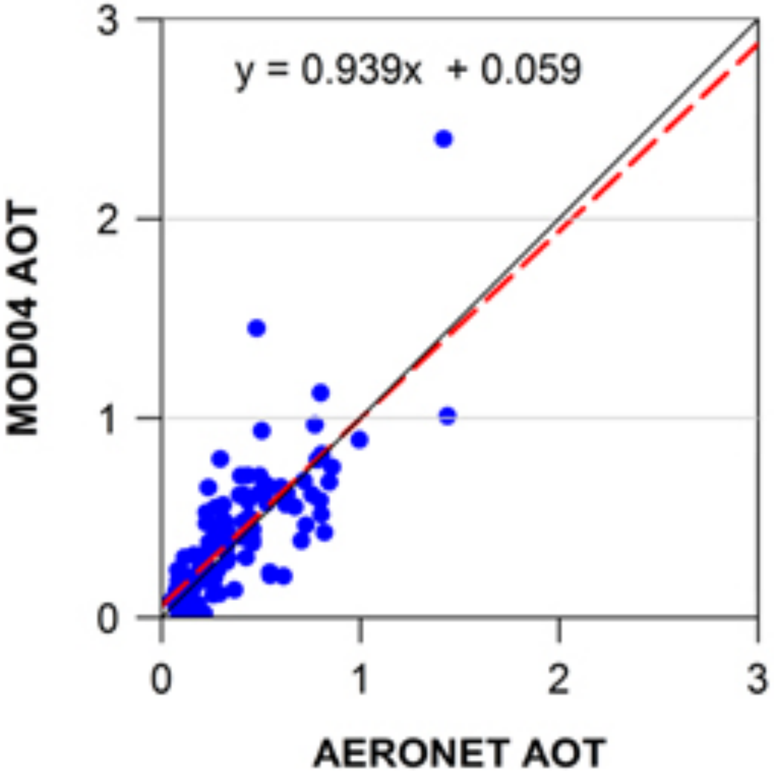
Editing the Code – Change the AOD Calculations



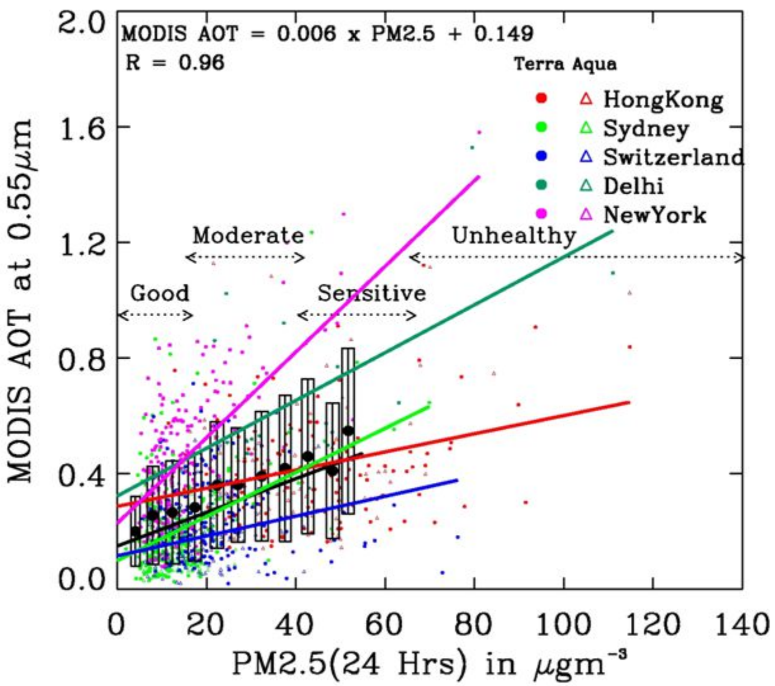
```
#calculates mean, median, stdev in a 3x3 grid around nearest point to entered location
if x < 1:
    x+=1
if x > data.shape[0]-2:
    x-=2
if y < 1:
    y+=1
if y > data.shape[1]-2:
    y-=2
three_by_three=data[x-1:x+2,y-1:y+2]
three_by_three=three_by_three.astype(float)
three_by_three[three_by_three==float(fillvalue)]=np.nan
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
```

Applications

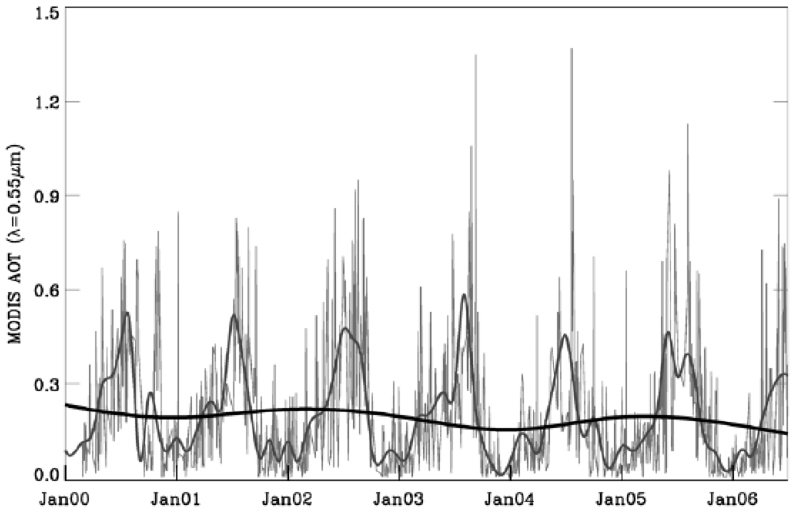
Satellite AOD Validation

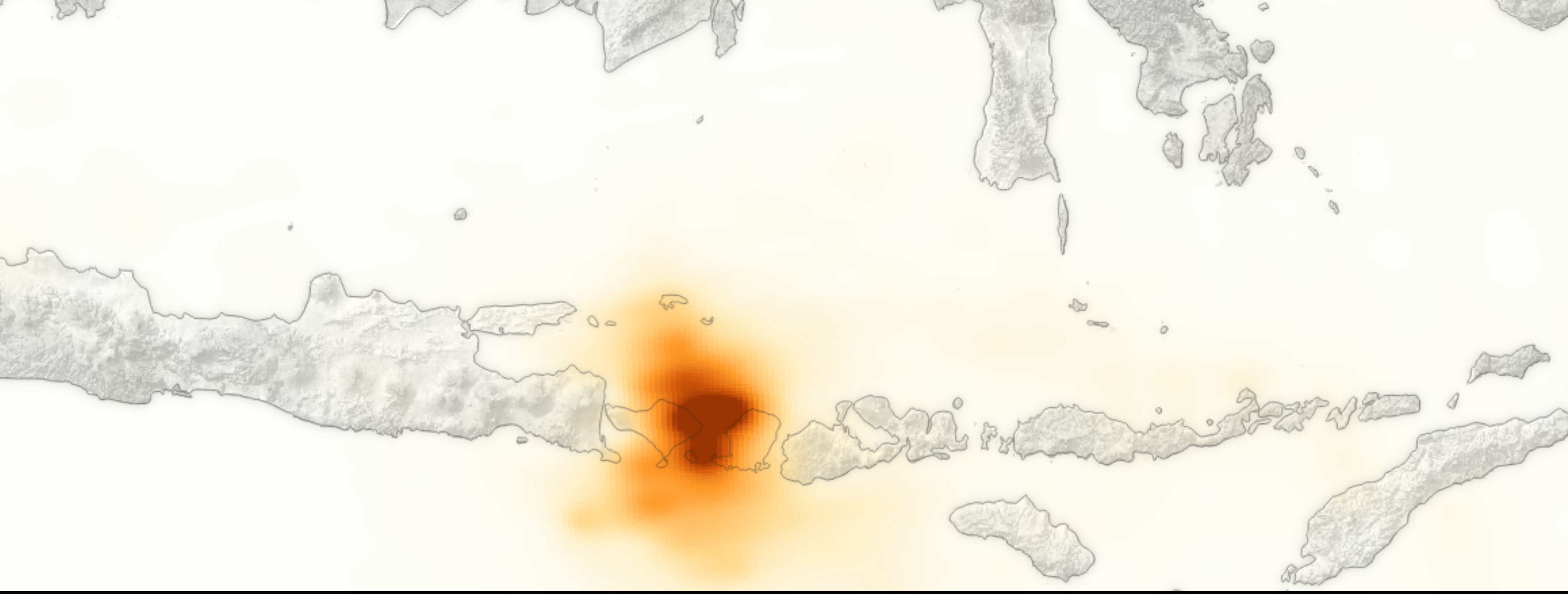


AOD-PM$_{2.5}$ Relationship



Time Series Analysis



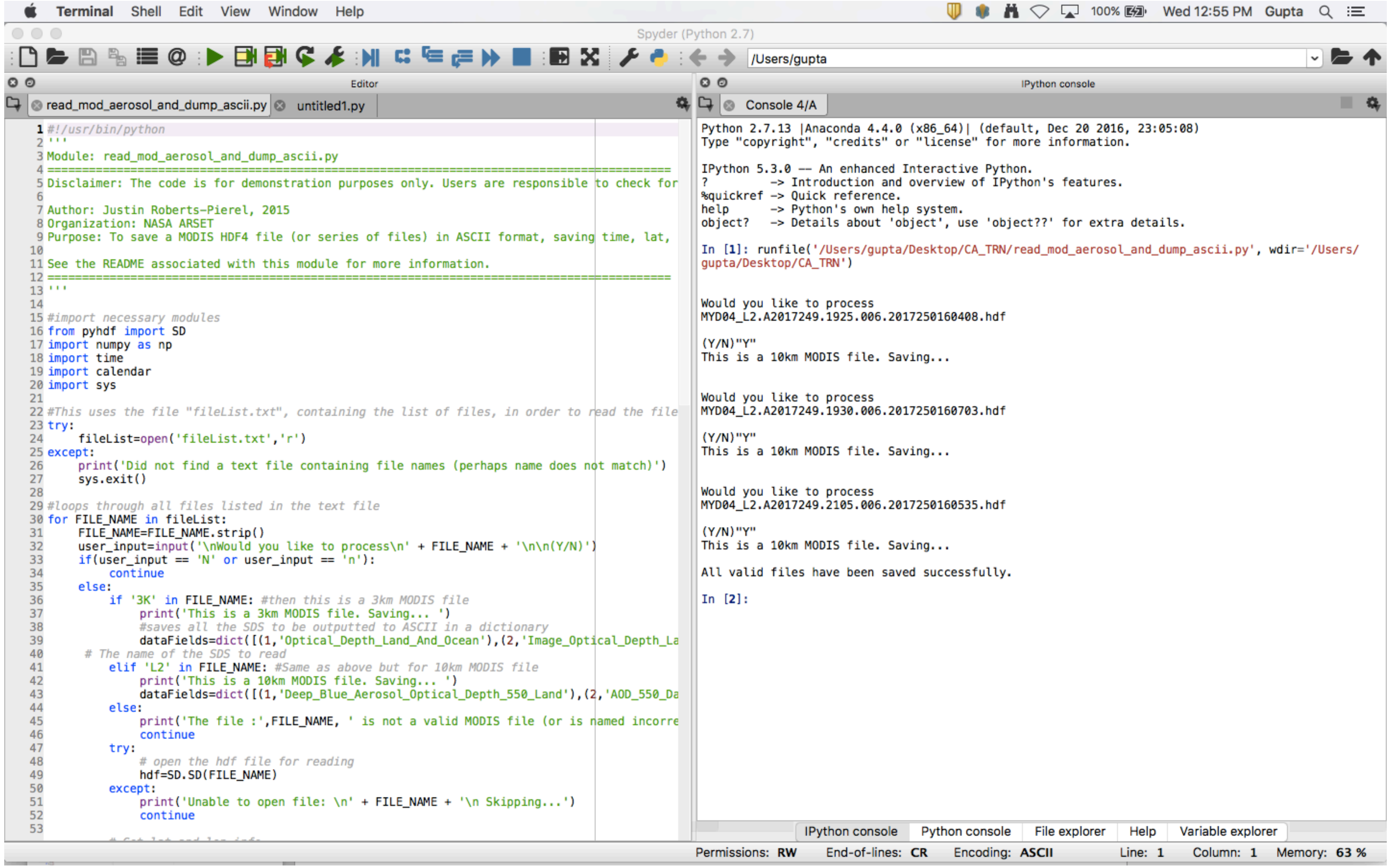


Output HDF variables to CSV

Output MODIS Aerosol Level 2 HDF variables to a CSV file

read_mod_aerosol_and_dump_ascii.py

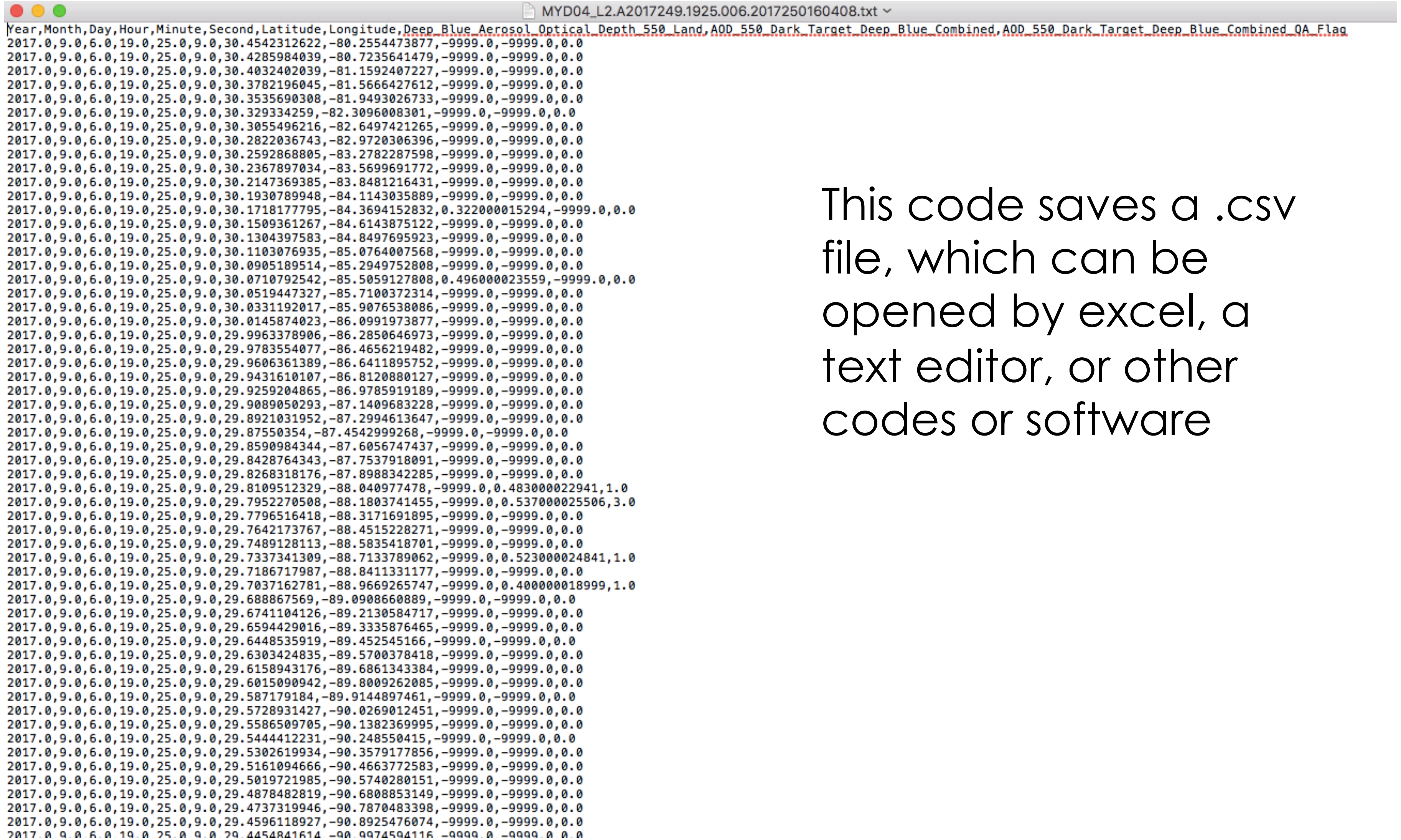
- **Purpose**: read a MODIS aerosol level 2 data file in HDF format and write certain SDSs into a csv (text) file
- The code works for both 10 km and 3 km products



```python
#!/usr/bin/python
'''
Module: read_mod_aerosol_and_dump_ascii.py
========================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for

Author: Justin Roberts-Pierel, 2015
Organization: NASA ARSET
Purpose: To save a MODIS HDF4 file (or series of files) in ASCII format, saving time, lat,

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

#import necessary modules
from pyhdf import SD
import numpy as np
import time
import calendar
import sys

#This uses the file "fileList.txt", containing the list of files, in order to read the file
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('\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. Saving... ')
            #saves all the SDS to be outputted to ASCII in a dictionary
            dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Image_Optical_Depth_La
     # 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. Saving... ')
            dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Da
        else:
            print('The file :',FILE_NAME, ' is not a valid MODIS file (or is named incorre
            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

```

```
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]:
```

Output

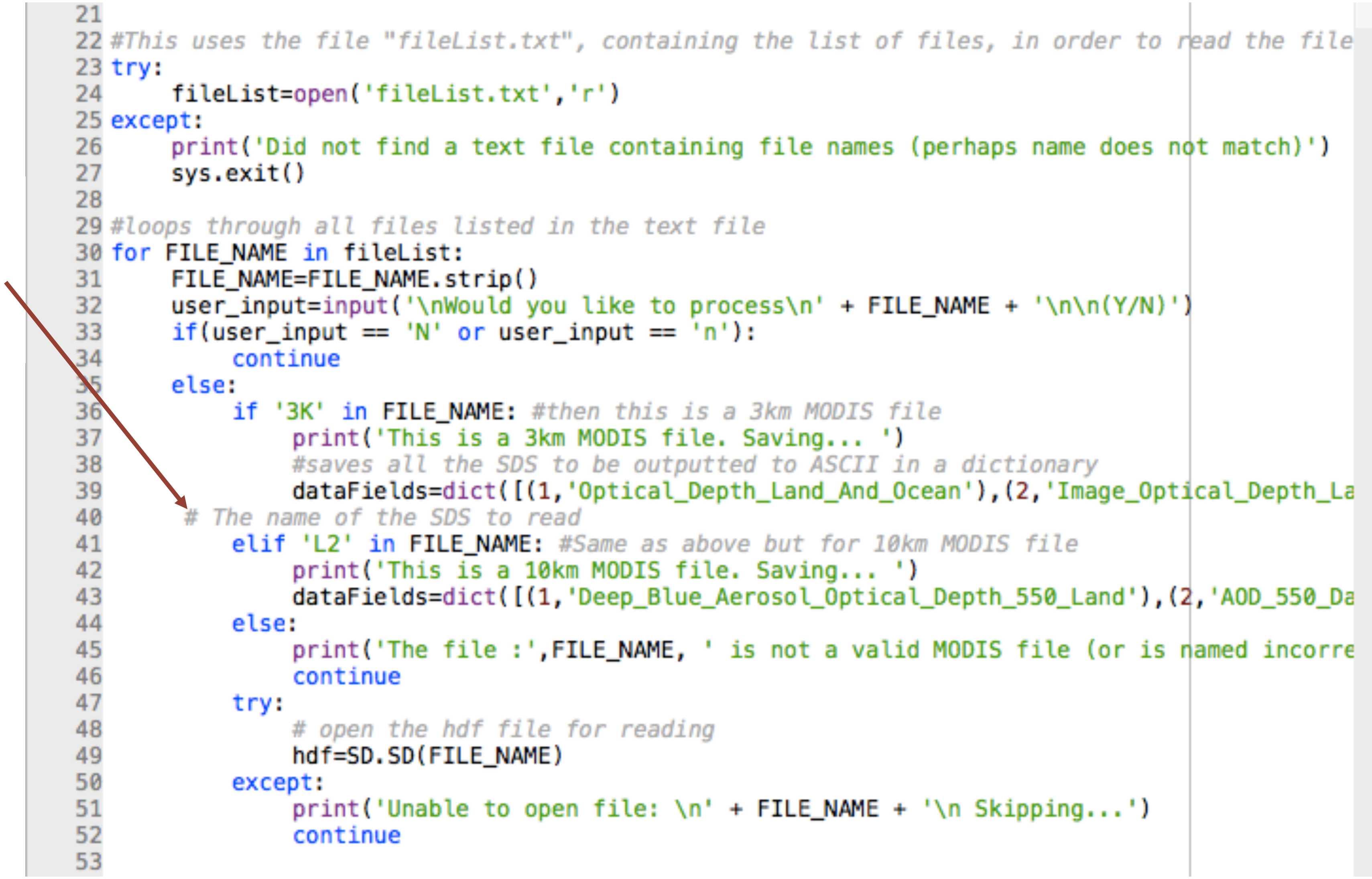


```
Year,Month,Day,Hour,Minute,Second,Latitude,Longitude,Deep_Blue_Aerosol_Optical_Depth_550_Land,AOD_550_Dark_Target_Deep_Blue_Combined,AOD_550_Dark_Target_Deep_Blue_Combined_QA_Flag
2017.0,9.0,6.0,19.0,25.0,9.0,30.4542312622,-80.2554473877,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.4285984039,-80.7235641479,-9999.0,-9999.0,0.0
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
```

This code saves a .csv file, which can be opened by excel, a text editor, or other codes or software

Editing the Code

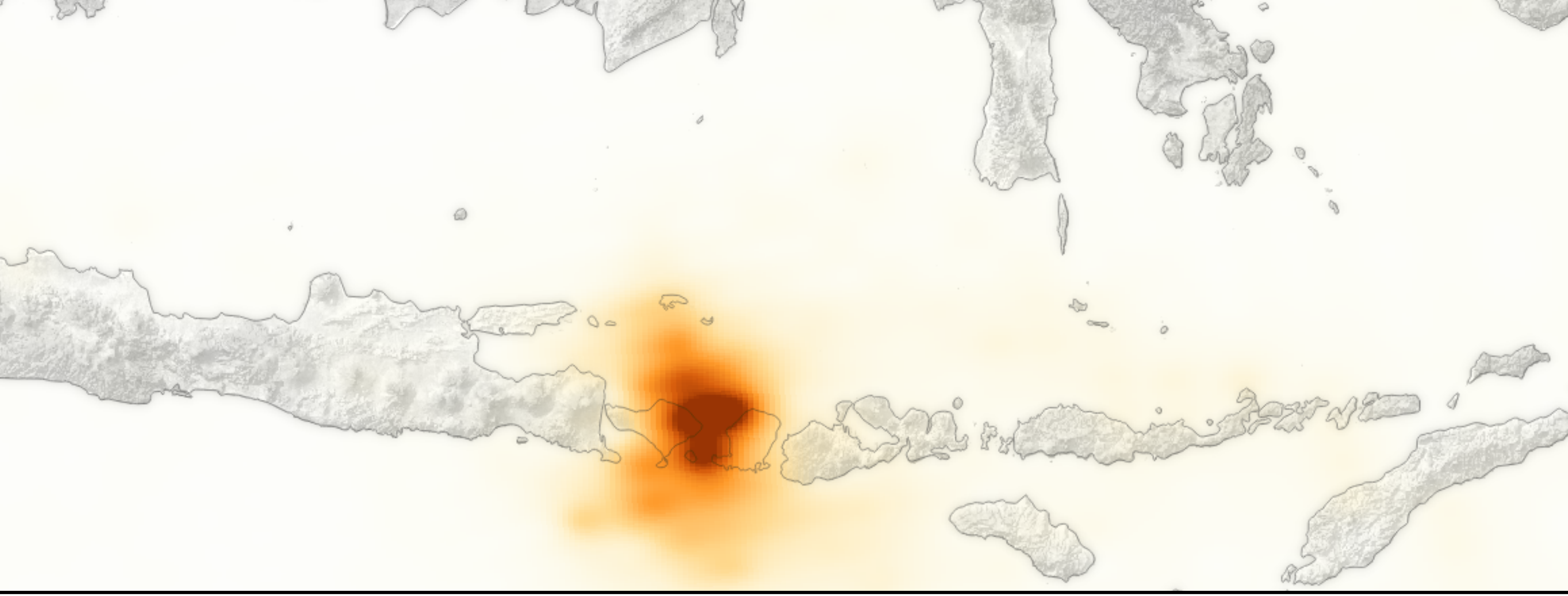
Change the list SDS to be written as an output



```

#This uses the file "fileList.txt", containing the list of files, in order to read the file
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('\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. Saving... ')
            #saves all the SDS to be outputted to ASCII in a dictionary
            dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Image_Optical_Depth_La
     # 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. Saving... ')
            dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Da
        else:
            print('The file :',FILE_NAME, ' is not a valid MODIS file (or is named incorre
            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

```

Applications

- This is a sample code to read and extract the MODIS Level 2 aerosol data
- The code can be modified to extract multiple SDSs into a single .csv file
- The code be easily modified to extract data over a certain region
- The output file can be opened in excel, or any other data analysis tool

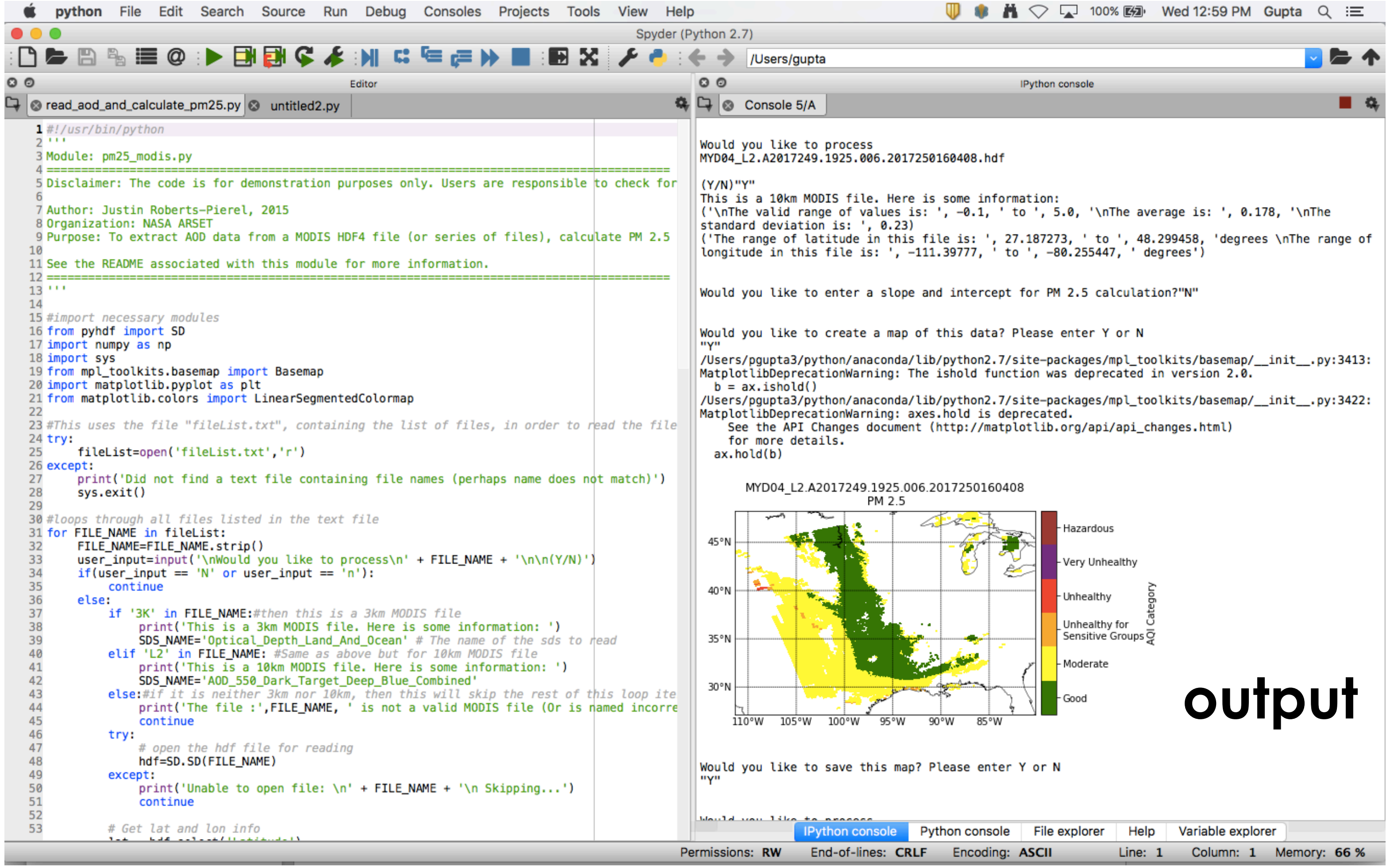


Create Air Quality Maps

Create an Air Quality Map

read_aod_and_calculate_pm25.py

- Purpose: read a MODIS aerosol level 2 data file in HDF format and create a PM2.5 air quality category map using the relationship between AOD and PM2.5
- The code works for both 10 km and 3 km products



```python
#!/usr/bin/python
'''
Module: pm25_modis.py
==========================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for

Author: Justin Roberts-Pierel, 2015
Organization: NASA ARSET
Purpose: To extract AOD data from a MODIS HDF4 file (or series of files), calculate PM 2.5

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

#import necessary modules
from pyhdf import SD
import numpy as np
import sys
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

#This uses the file "fileList.txt", containing the list of files, in order to read the file
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('\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 ite
            print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorre
            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

        # 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:
('\nThe 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"
/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)
```

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

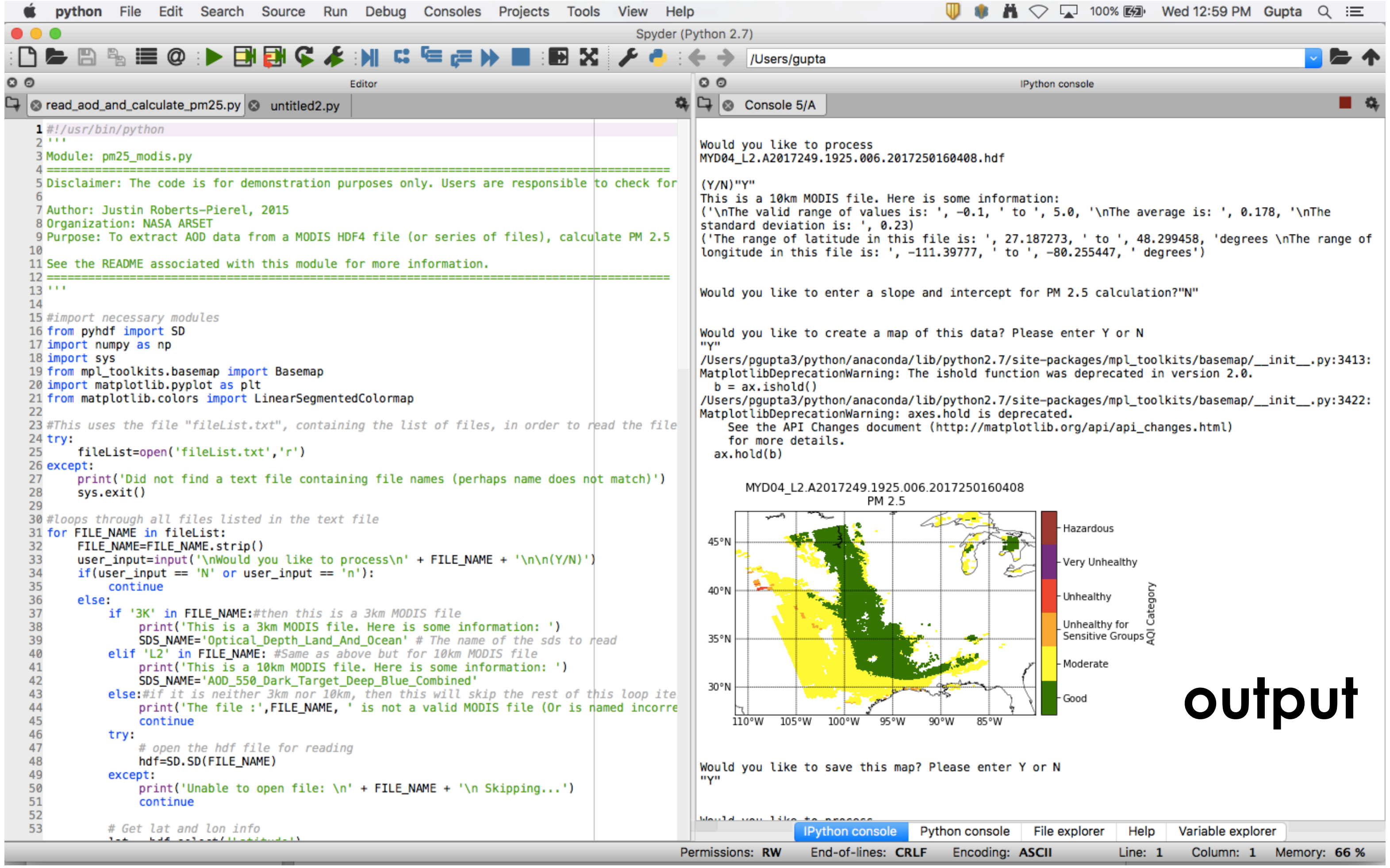
output

Create an Air Quality Map

read_aod_and_calculate_pm25.py

Disclaimer: This is just an example code. The default AOD-PM2.5 relationship used here is assumed relationship over USA.

The users of this code are responsible for checking the validity of this relationship and encouraged to use local relationships for visualizing AQ in different parts of the world.



```python
#!/usr/bin/python
'''
Module: pm25_modis.py
==========================================================================================
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for

Author: Justin Roberts-Pierel, 2015
Organization: NASA ARSET
Purpose: To extract AOD data from a MODIS HDF4 file (or series of files), calculate PM 2.5

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

#import necessary modules
from pyhdf import SD
import numpy as np
import sys
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

#This uses the file "fileList.txt", containing the list of files, in order to read the file
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('\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 ite
            print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorre
            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

        # Get lat and lon info
```

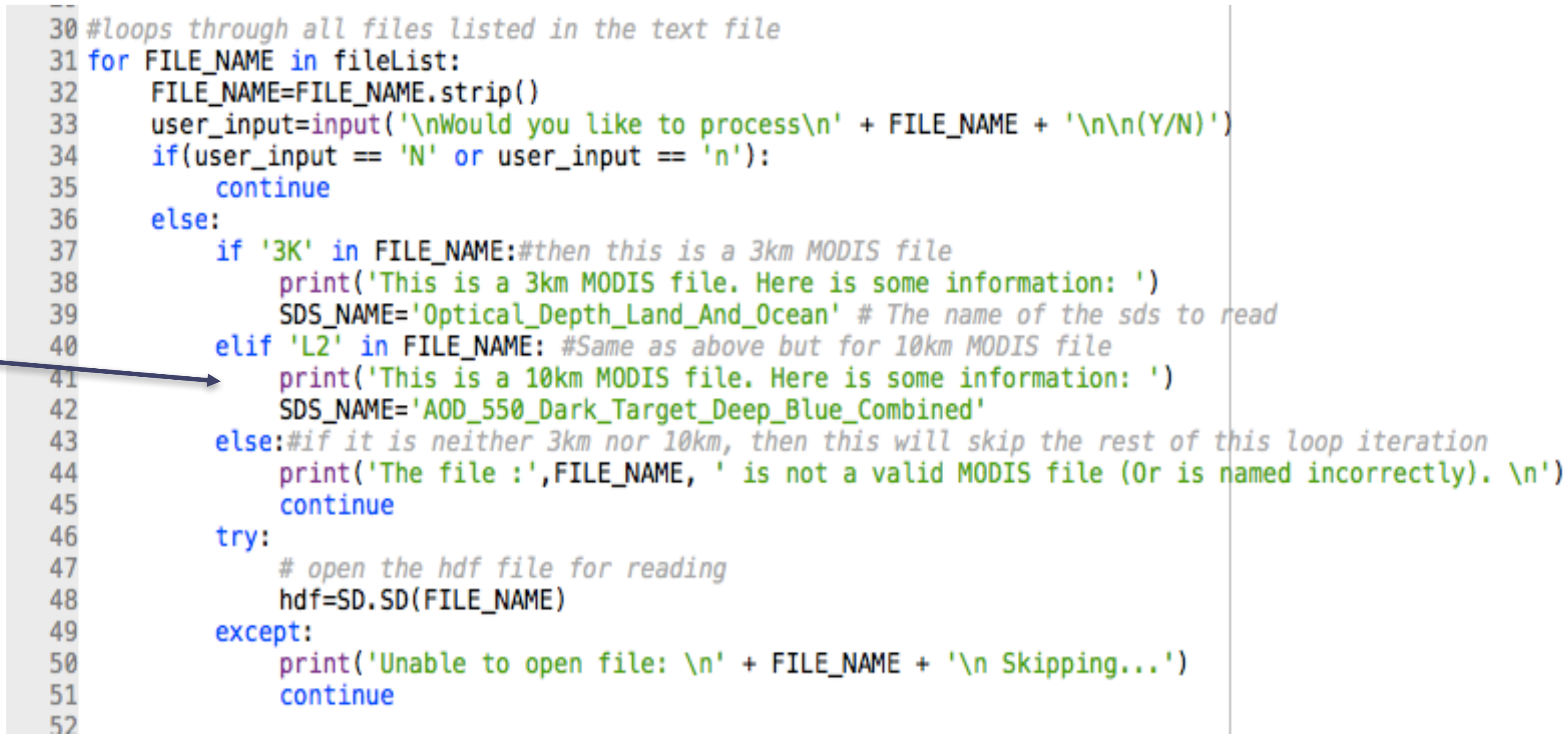
```
Would you like to process
MYD04_L2.A2017249.1925.006.2017250160408.hdf

(Y/N)"Y"
This is a 10km MODIS file. Here is some information:
('\nThe 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"
/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)
```

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

output

Editing the Code – Change the SDS

The user can change the AOD SDS to be used in PM2.5 calculation



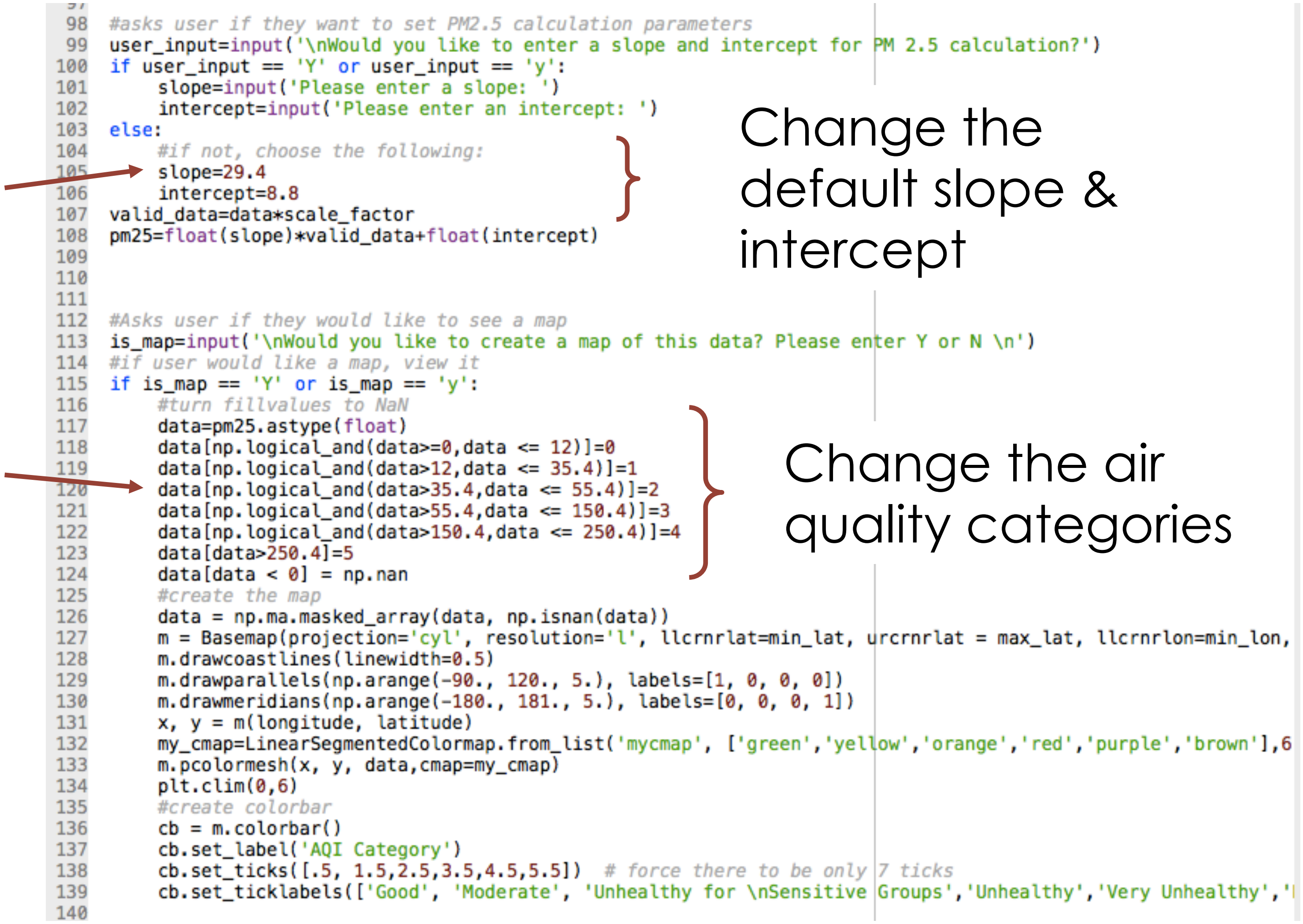
```
#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if(user_input == 'N' or user_input == 'n'):
        continue
    else:
        if '3K' in FILE_NAME:#then 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
```

Editing the Code: Change the AOD-$PM_{2.5}$ Relationship and AQI

The code uses
$PM_{2.5}$ = Slope * AOD + Intercept
as the linear regression equation to calculate $PM_{2.5}$ from AOD

The code uses the U.S. EPA definition of air quality categories based on $PM_{2.5}$

AQI Calculator:
https://airnow.gov/index.cfm?action=airnow.calculator

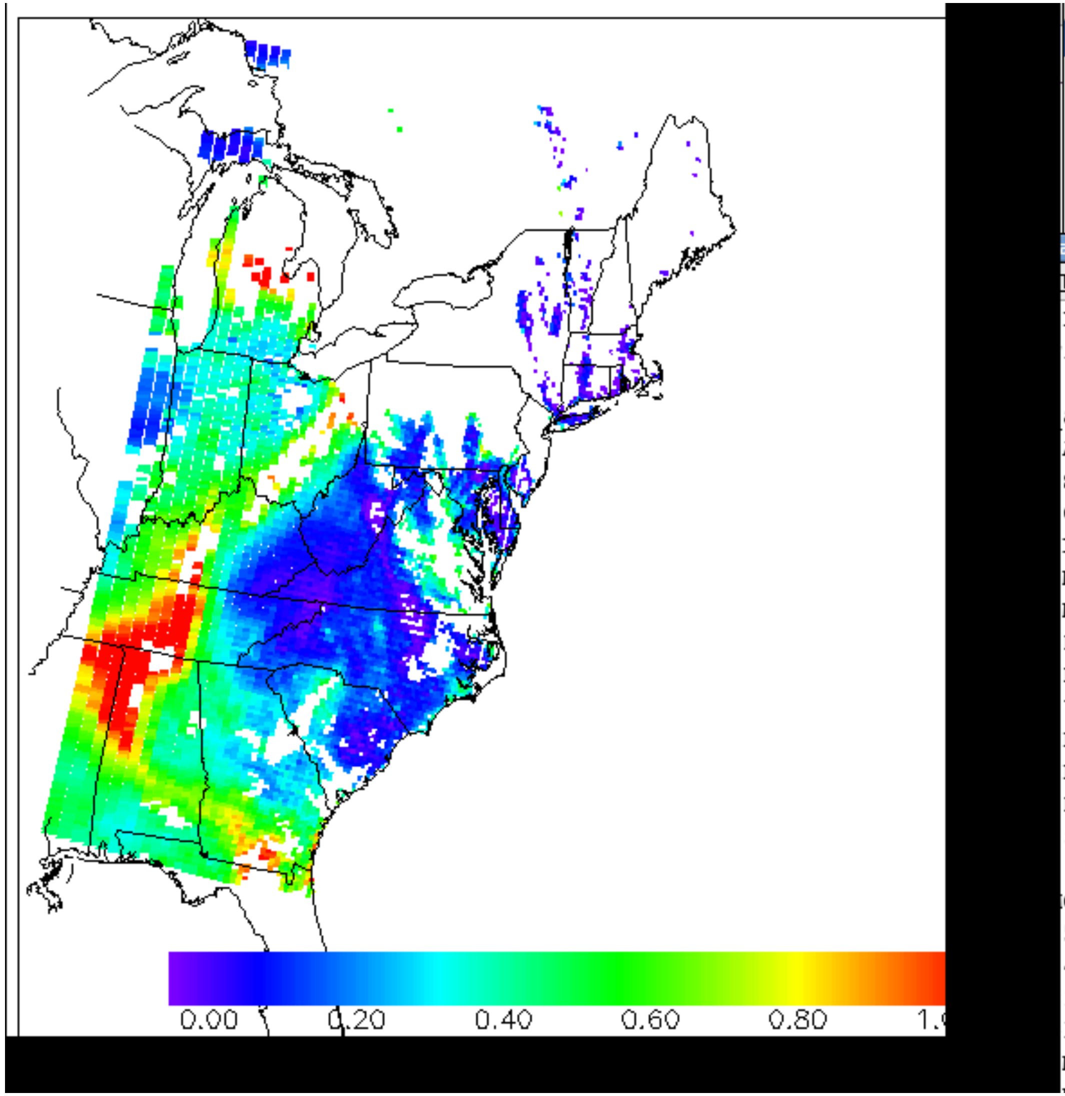


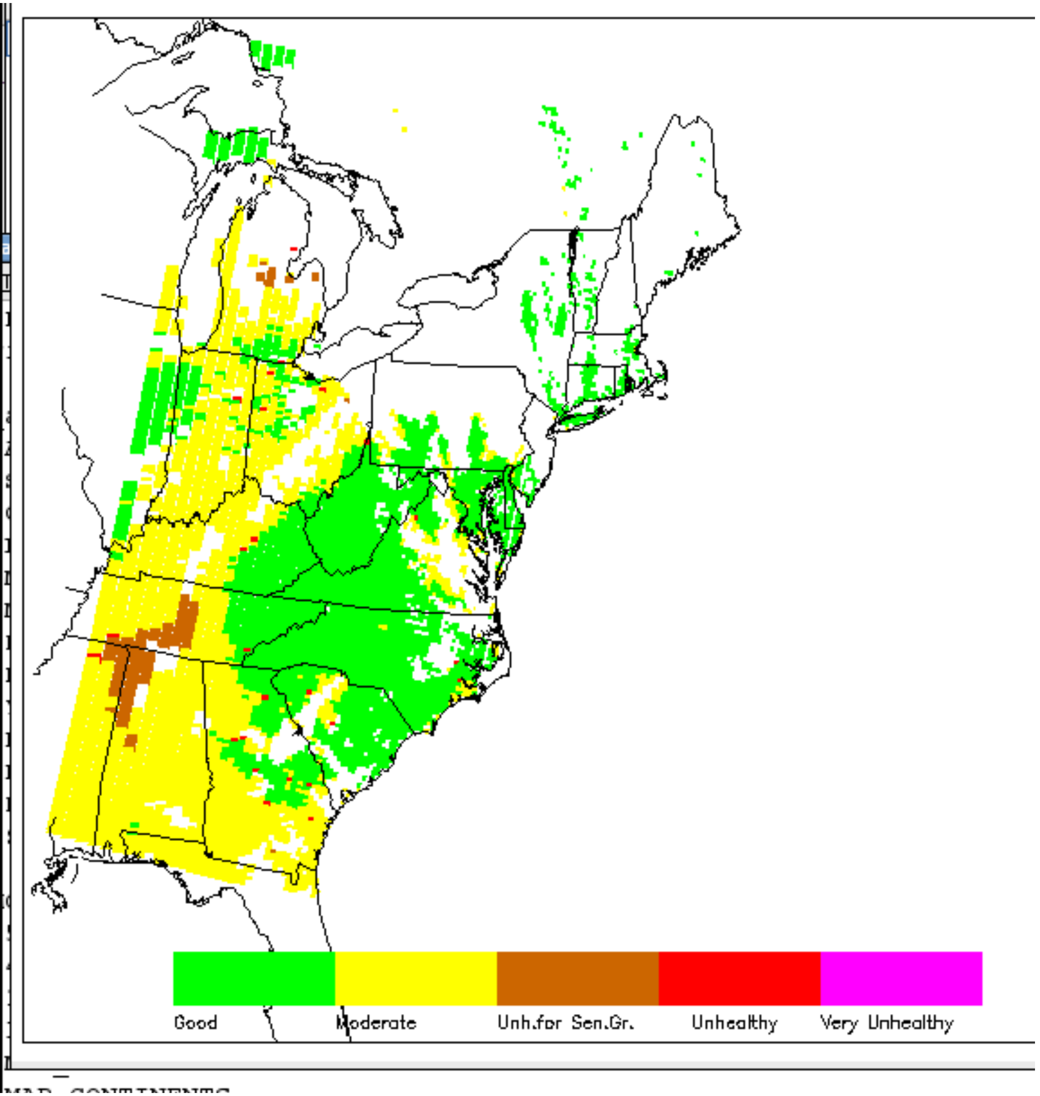
```
#asks user if they want to set PM2.5 calculation parameters
user_input=input('\nWould you like to enter a slope and intercept for PM 2.5 calculation?')
if user_input == 'Y' or user_input == 'y':
    slope=input('Please enter a slope: ')
    intercept=input('Please enter an intercept: ')
else:
    #if not, choose the following:
    slope=29.4
    intercept=8.8
valid_data=data*scale_factor
pm25=float(slope)*valid_data+float(intercept)



#Asks user if they would like to see a map
is_map=input('\nWould you like to create a map of this data? Please enter Y or N \n')
#if user would like a map, view it
if is_map == 'Y' or is_map == 'y':
    #turn fillvalues to NaN
    data=pm25.astype(float)
    data[np.logical_and(data>=0,data <= 12)]=0
    data[np.logical_and(data>12,data <= 35.4)]=1
    data[np.logical_and(data>35.4,data <= 55.4)]=2
    data[np.logical_and(data>55.4,data <= 150.4)]=3
    data[np.logical_and(data>150.4,data <= 250.4)]=4
    data[data>250.4]=5
    data[data < 0] = 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)
    my_cmap=LinearSegmentedColormap.from_list('mycmap', ['green','yellow','orange','red','purple','brown'],6
    m.pcolormesh(x, y, data,cmap=my_cmap)
    plt.clim(0,6)
    #create colorbar
    cb = m.colorbar()
    cb.set_label('AQI Category')
    cb.set_ticks([.5, 1.5,2.5,3.5,4.5,5.5])  # force there to be only 7 ticks
    cb.set_ticklabels(['Good', 'Moderate', 'Unhealthy for \nSensitive Groups','Unhealthy','Very Unhealthy','
```

Change the default slope & intercept

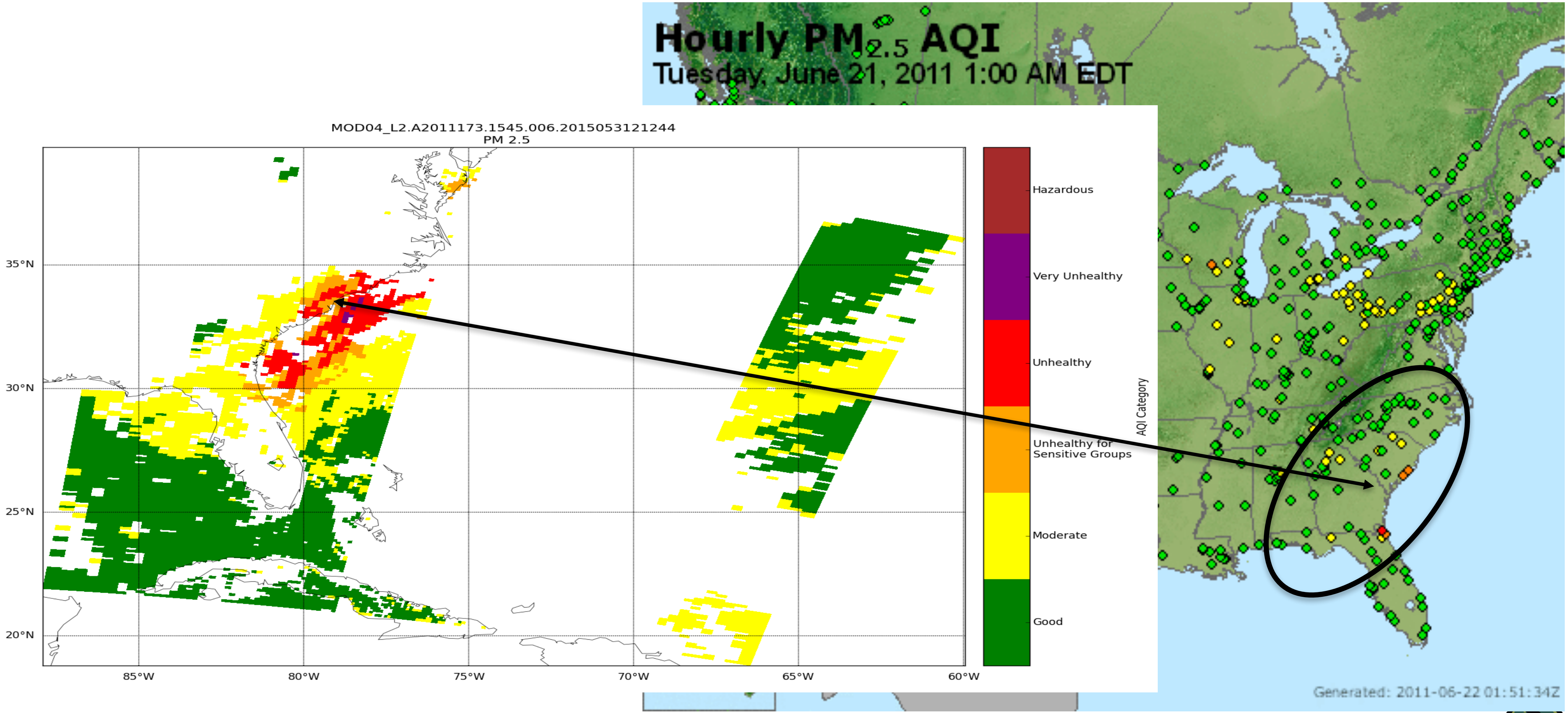
Change the air quality categories

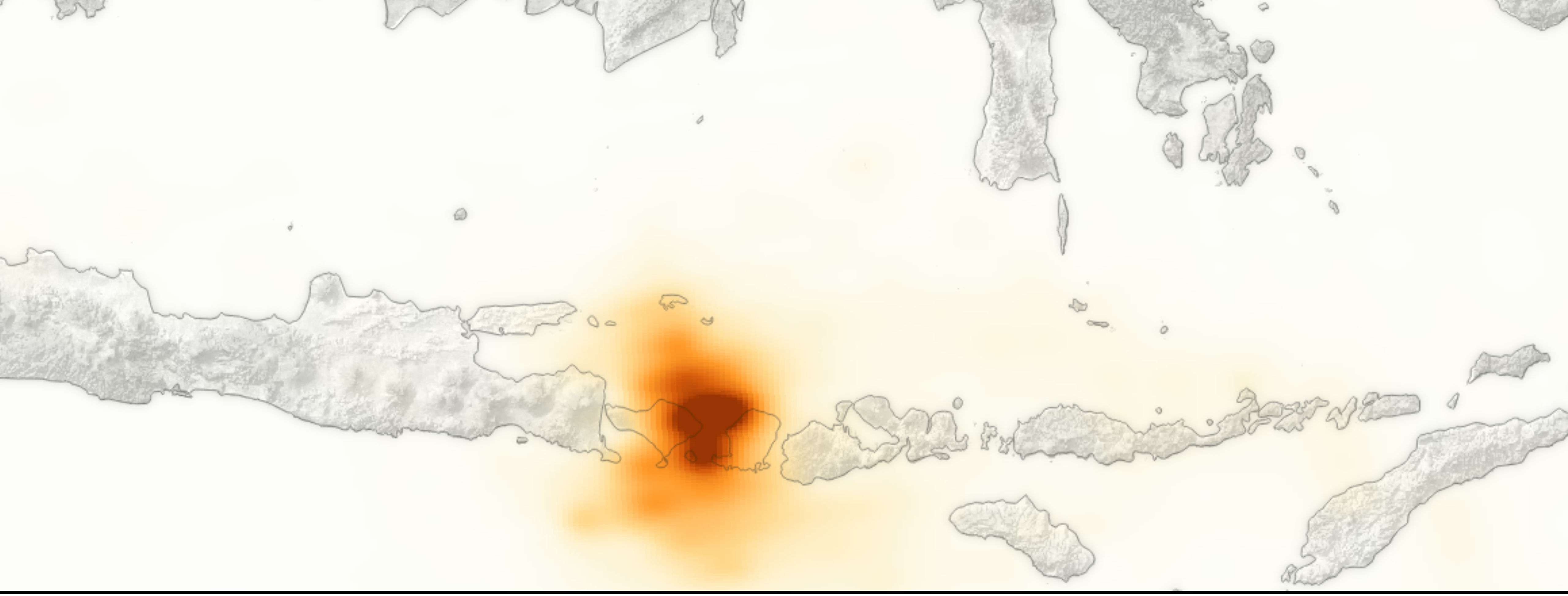
Application: Convert AOD into $PM_{2.5}$ & Air Quality Maps





Application – Compare Satellite with Surface Maps


Hourly PM$_{2.5}$ AQI
Tuesday, June 21, 2011 1:00 AM EDT
MOD04_L2.A2011173.1545.006.2015053121244
PM 2.5
35°N
30°N
25°N
20°N
85°W
80°W
75°W
70°W
65°W
60°W
Hazardous
Very Unhealthy
Unhealthy
Unhealthy for Sensitive Groups
Moderate
Good
AQI Category
Generated: 2011-06-22 01:51:34Z



Questions & Answers