

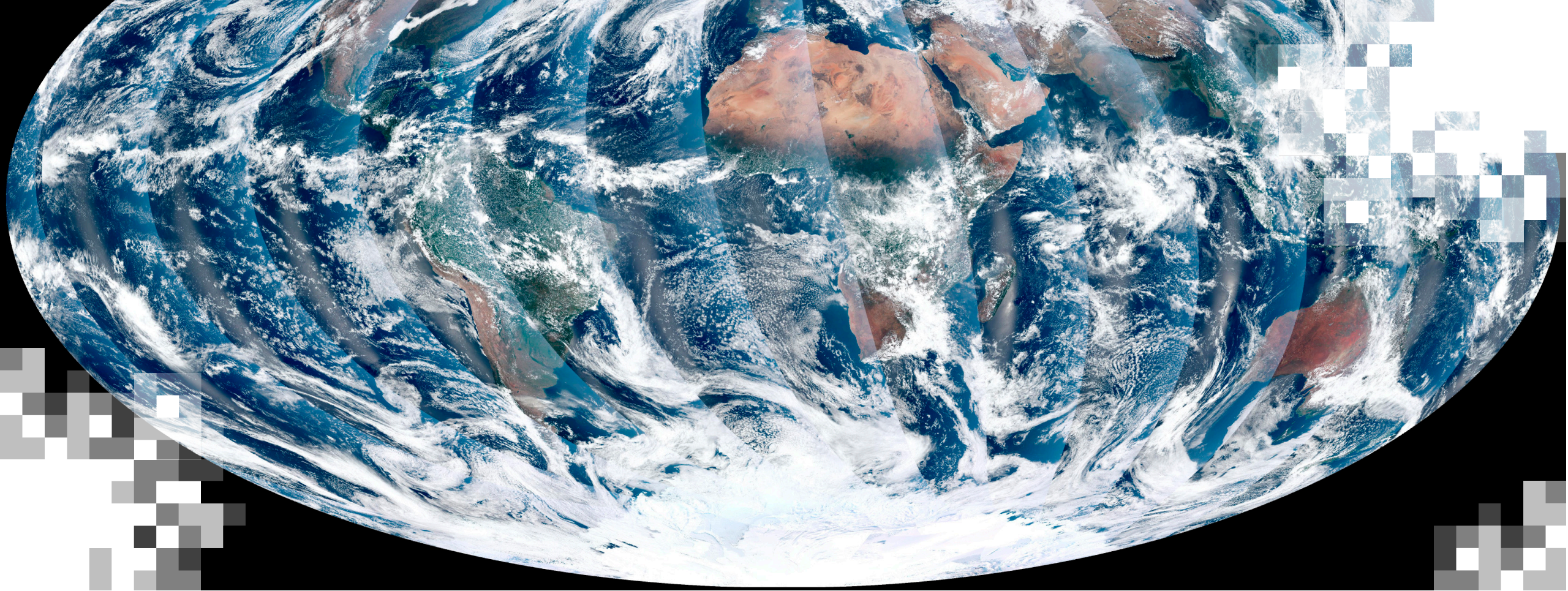
# Things to do before you attend the webinar...

1. Create an Earthdata login if you don't have one (directions below).
2. Create a Google account if you would like to use the Jupyter notebooks as demonstrated during the webinar (optional Notebook walkthrough below if you want to view what we will cover).
3. Watch Session 2 of our Introductory Webinar, "An Inside Look at how NASA Measures Air Pollution." This session covers:
  - What are Aerosols?
  - Interpreting Aerosol Imagery: Dos and Don'ts
  - A Tour of NASA Resources for Generating Your Own Visualizations

<https://appliedsciences.nasa.gov/join-mission/training/english/inside-look-how-nasa-measures-air-pollution> (English)

<https://appliedsciences.nasa.gov/join-mission/training/spanish/un-vistazo-como-la-nasa-mide-la-contaminacion-del-aire-0> (Spanish)

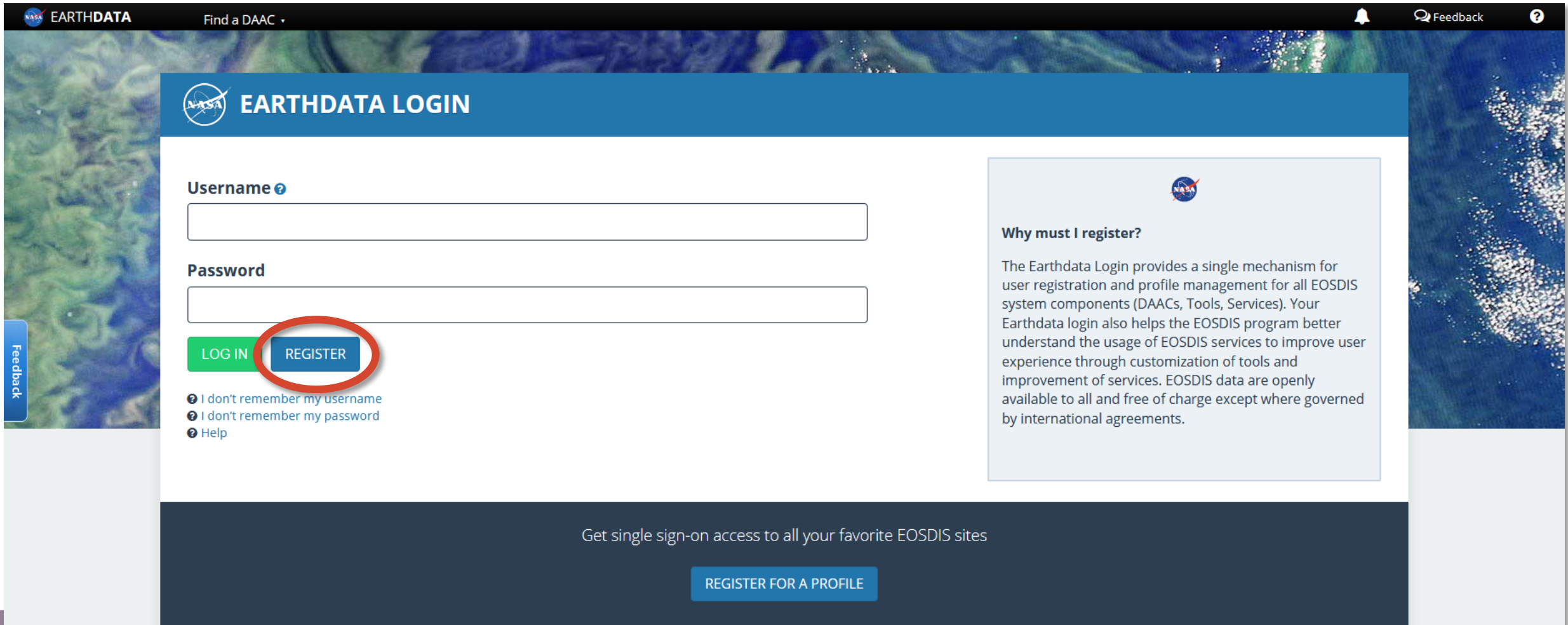




Create an Earthdata Account

# Create an Earthdata Account

Step 1: Go to <https://urs.earthdata.nasa.gov/> and click “Register.”



The screenshot shows the Earthdata Login page. At the top, there is a navigation bar with the NASA Earthdata logo, a search bar for DAACs, and links for Feedback and Help. The main content area has a blue header with the NASA logo and the text "EARTHDATA LOGIN". Below this, there are two input fields: "Username" and "Password". To the right of these fields is a text box titled "Why must I register?" which explains the benefits of the Earthdata Login system. Below the input fields, there are two buttons: "LOG IN" (green) and "REGISTER" (blue). The "REGISTER" button is circled in red. Below the buttons, there are three links: "I don't remember my username", "I don't remember my password", and "Help". At the bottom of the page, there is a dark blue banner with the text "Get single sign-on access to all your favorite EOSDIS sites" and a "REGISTER FOR A PROFILE" button.



# Create an Earthdata Account

## Step 2: Fill out registration form and click “Register for Earthdata Login”

**EARTHDATA LOGIN**

### Register for an Earthdata Login Profile

Profile Information

**Username:**

**Password:**

**Password Confirmation:**

**Required field**

**Username must:**

- Be a Minimum of 4 characters
- Be a Maximum of 30 characters
- Use letters, numbers, periods and underscores
- Not contain any blank spaces
- Not begin, end or contain two consecutive special characters( . \_ )

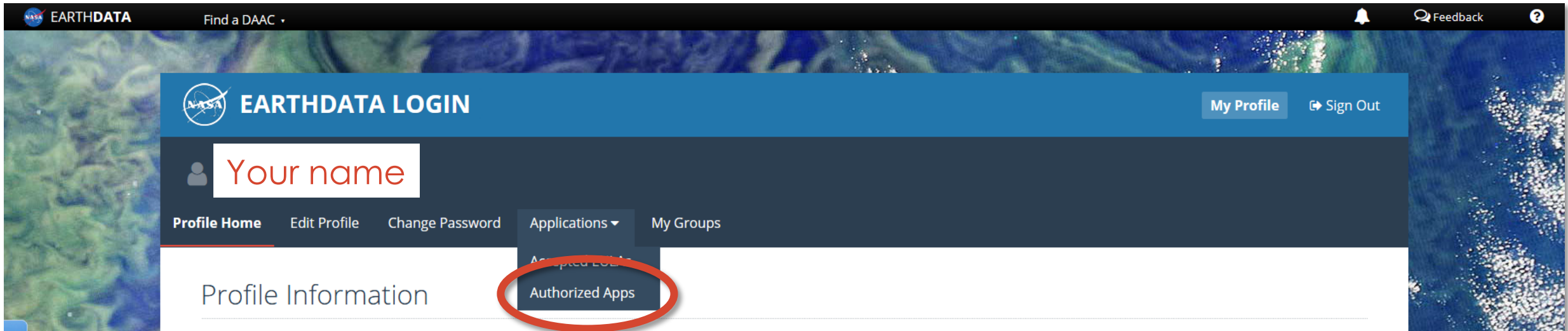
**Password must contain:**

- Minimum of 8 characters
- One Uppercase letter
- One Lowercase letter
- One Number














# Add Applications

Step 1: After you log in, you'll see your profile screen, click Applications → Authorized Apps.



# Add Applications

Step 2: Scroll down to the bottom and click “Approve More Applications.”

GES DISC	 
Earthdata Search Prod (new)	 
CMR SSO APP for EDL in PROD	
Earthdata Search PROD (Serverless)	 
SEDAC Website	 
ORNL DAAC apache module	 

APPROVE MORE APPLICATIONS



# Add Applications

Step 3: Search for “LAADS Web” and the click “Search.”

## Approve Applications

LAADS Web

SEARCH

These applications have a EULA, and must be authorized before you can use them

+ AESICS	AUTHORIZE
+ Alaska Satellite Facility Data Access	AUTHORIZE
+ Alaska Satellite Facility Data Access (DEV/TEST)	AUTHORIZE
+ Alaska Satellite Facility Data Access Egress Control	AUTHORIZE
+ ASTER Free Data	AUTHORIZE
+ Contingency app	AUTHORIZE
+ DB Direct	AUTHORIZE

Feedback



# Add Applications

Step 4: Check 'Show Applications that can be auto-authorized' and click "Authorize."

The screenshot displays the EarthData Login interface. At the top, the NASA logo and 'EARTHDATA LOGIN' are visible, along with 'My Profile' and 'Sign Out' links. Below this, the user's name 'Your name' is shown. A navigation bar includes 'Profile Home', 'Edit Profile', 'Change Password', 'Applications' (which is underlined), and 'My Groups'. The main content area is titled 'Approve Applications' and features a search bar with a 'SEARCH' button. Underneath, the 'Application Results' section contains a checked checkbox for 'Show applications that can be auto-authorized'. Below this checkbox, a list of applications is shown, with 'LAADS Web' as the first entry. To the right of 'LAADS Web' is a blue 'AUTHORIZE' button, which is circled in red. A 'Feedback' button is visible on the left side of the page.

For questions regarding the EOSDIS Earthdata Login, please contact Earthdata Support.





# Add Applications

## Step 5: Authorize any additional user attributes, and click “Authorize”

### Authorize Application

Additional user attributes required by the application.  
Please provide the following information to be able to authorize and use the application.

Sentinel3: ⓘ •

1

Meris: ⓘ •

1

Application Administrators may send out occasional emails notifying users about application updates or alerts.  Yes, I would like to be notified.

AUTHORIZE

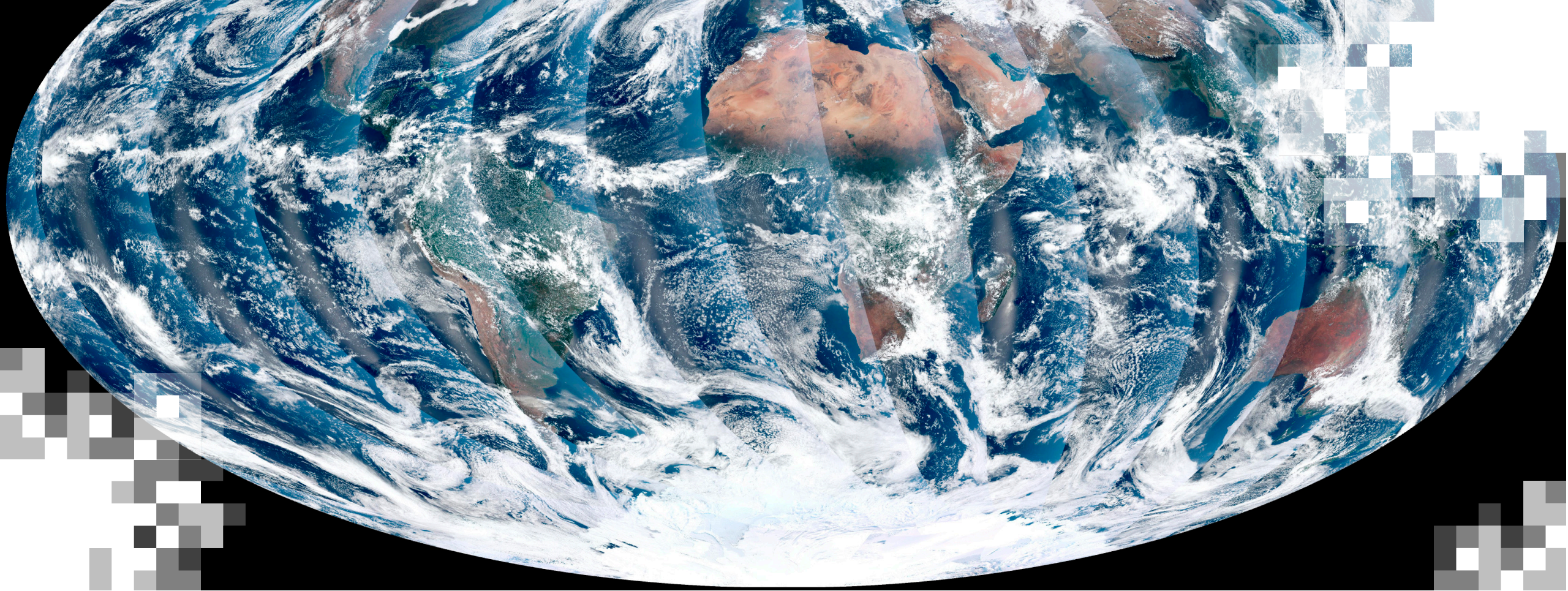
NO THANKS

For questions regarding the EOSDIS Earthdata Login, please contact [Earthdata Support](#)

V 4.123 [Home](#) [Register](#) [Documentation](#) [NASA](#)

Feedback





Download Jupyter Notebooks and Save to  
Your Computer

# Download Jupyter Notebooks and Save to Your Computer

Step 1: Go to <https://github.com/NASAARSET/>, click on VIIRS\_NASA

The screenshot shows the GitHub profile page for NASAARSET. The profile name is NASAARSET, and the profile picture is the Applied Remote Sensing Training Program logo. A red circle highlights the 'VIIRS\_NASA' repository in the 'Popular repositories' section. The repository is categorized as a 'Jupyter Notebook'. Below this, there is a contribution calendar showing 18 contributions in the last year, and a 'Contribution activity' section for October 2020.



# Download Jupyter Notebooks and Save to Your Computer

## Step 2: Click on read\_and\_map\_viirs.ipynb

The screenshot shows the GitHub interface for the repository NASAARSET/VIIRS\_NASA. The repository is currently on the master branch with 1 branch and 0 tags. The file list includes:

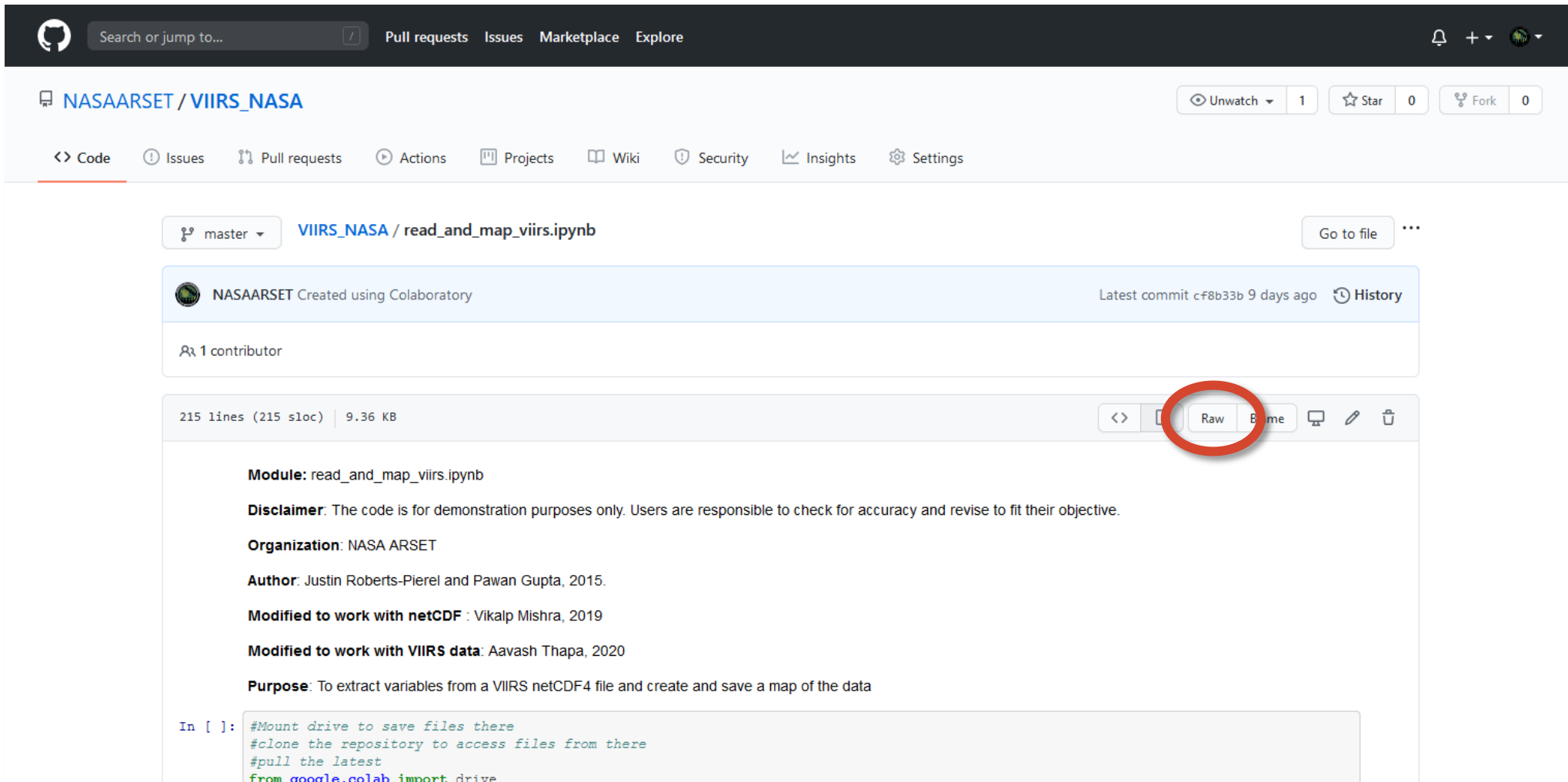
File Name	Action	Time
NASAARSET Created using Colaboratory		d49baf3 23 hours ago 14 commits
AERDB_L2_VIIRS_SNPP.A2020056.19...	Add files via upload	9 days ago
README.md	Update README.md	9 days ago
fileList.txt	Add files via upload	9 days ago
<b>read_and_map_viirs.ipynb</b>	Created using Colaboratory	9 days ago
read_aod_and_calculate_pm25.ipynb	Created using Colaboratory	23 hours ago
read_viirs_and_list_sds.ipynb	Created using Colaboratory	23 hours ago
read_viirs_at_a_location.ipynb	Created using Colaboratory	23 hours ago
viirs_export_csv.ipynb	Created using Colaboratory	23 hours ago

The file **read\_and\_map\_viirs.ipynb** is circled in red. The right sidebar shows repository metadata: 1 Unwatch, 0 Stars, 0 Forks, and no releases or packages published. The language statistics show 100.0% Jupyter Notebook.



# Download Jupyter Notebooks and Save to Your Computer

Step 3: Above the code, click 'Raw'. This will display the raw code.



The screenshot shows the GitHub interface for the repository NASAARSET/VIIRS\_NASA. The file 'read\_and\_map\_viirs.ipynb' is selected, and the 'Raw' button is circled in red. The file information shows it was created using Colaboratory and has a latest commit of c8b33b 9 days ago. The file size is 9.36 KB and it contains 215 lines of code. The code content includes a disclaimer, organization information, author details, and a purpose statement.

215 lines (215 sloc) | 9.36 KB

**Module:** read\_and\_map\_viirs.ipynb

**Disclaimer:** The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objective.

**Organization:** NASA ARSET

**Author:** Justin Roberts-Pierel and Pawan Gupta, 2015.

**Modified to work with netCDF :** Vikalp Mishra, 2019

**Modified to work with VIIRS data:** Aavash Thapa, 2020

**Purpose:** To extract variables from a VIIRS netCDF4 file and create and save a map of the data

```
In [ ]: #Mount drive to save files there
#clone the repository to access files from there
#pull the latest
from google.colab import drive
```



# Download Jupyter Notebooks and Save to Your Computer

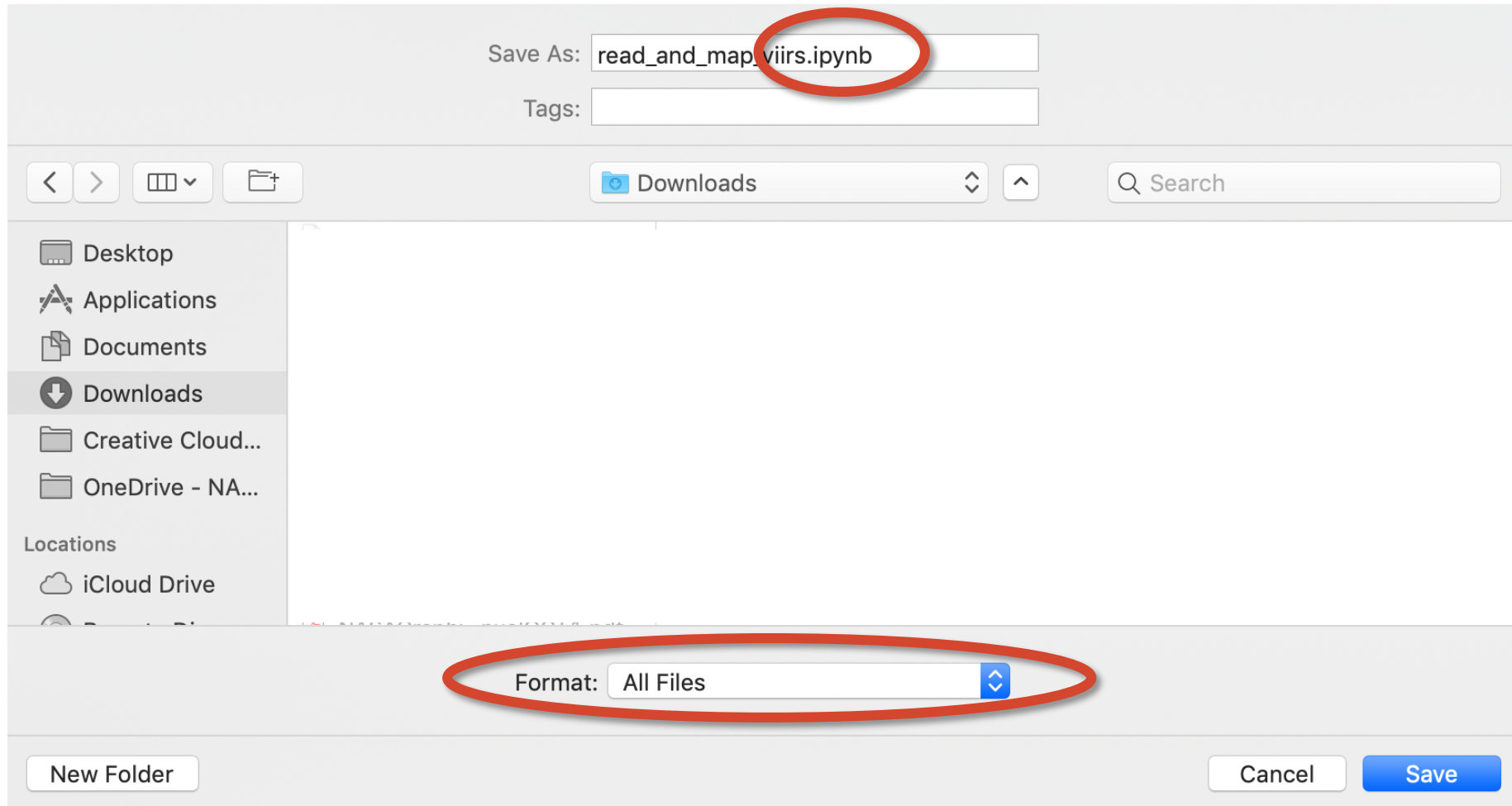
## Step 4: Click Ctrl+S to save to your computer as a .ipynb.

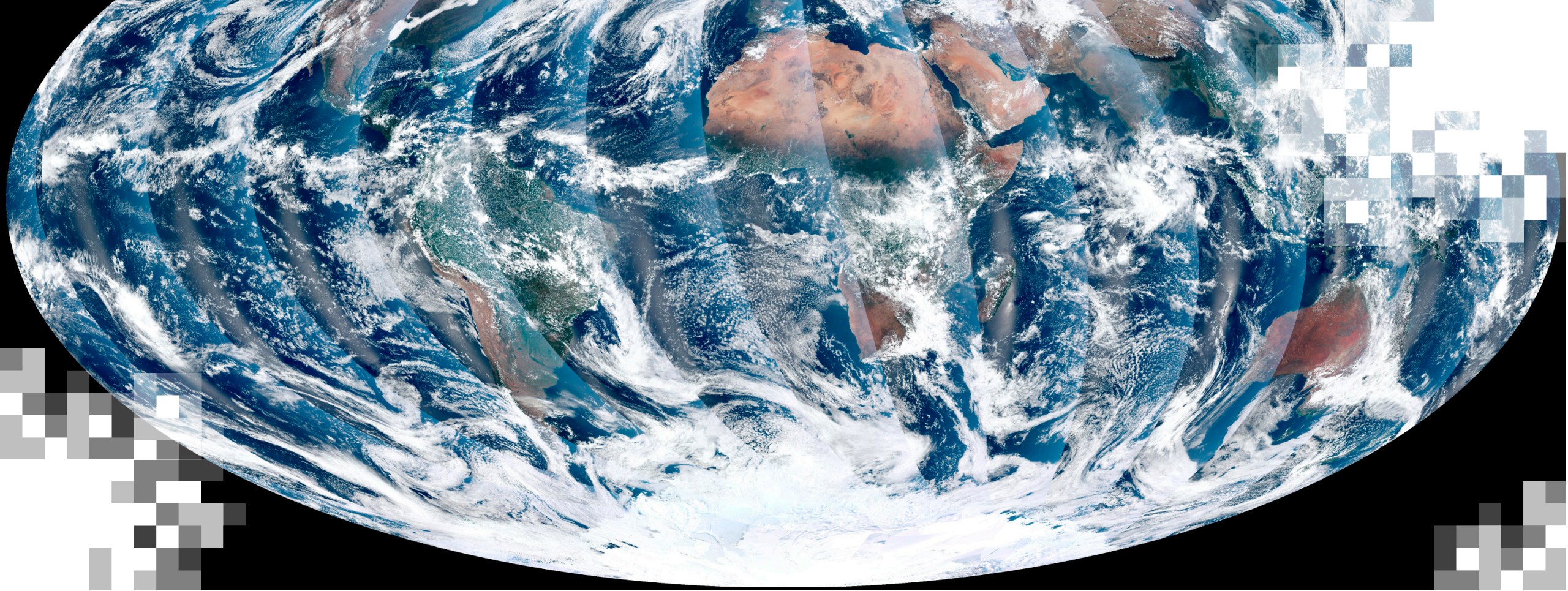
```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "name": "read_and_map_viirs.ipynb",
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "im8zRbMsiQLL"
      },
      "source": [
        "***Module** read_and_map_viirs.ipynb\n",
        "\n",
        "***Disclaimer**": The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objective.\n",
        "\n",
        "***Organization**": NASA ARSET\n",
        "\n",
        "***Author**": Justin Roberts-Pierel and Pawan Gupta, 2015.\n",
        "\n",
        "***Modified to work with netCDF** : Vikalp Mishra, 2019 \n",
        "\n",
        "***Modified to work with VIIRS data**": Aavash Thapa, 2020\n",
        "\n",
        "***Purpose**": To extract variables from a VIIRS netCDF4 file and create and save a map of the data"
      ]
    },
    {
      "cell_type": "code",
      "metadata": {
        "id": "Y7U02gkECz6c"
      },
      "source": [
        "#Mount drive to save files there\n",
        "#clone the repository to access files from there\n",
        "#pull the latest\n",
        "from google.colab import drive\n",
        "drive.mount('/content/drive', force_remount=True)\n",
        "! git clone https://github.com/NASAARSET/VIIRS_NASA.git\n",
        "! git -C VIIRS_NASA/ pull"
      ]
    }
  ]
}
```



# Download Jupyter Notebooks and Save to Your Computer

Make sure you save the file with the extension “.ipynb”!



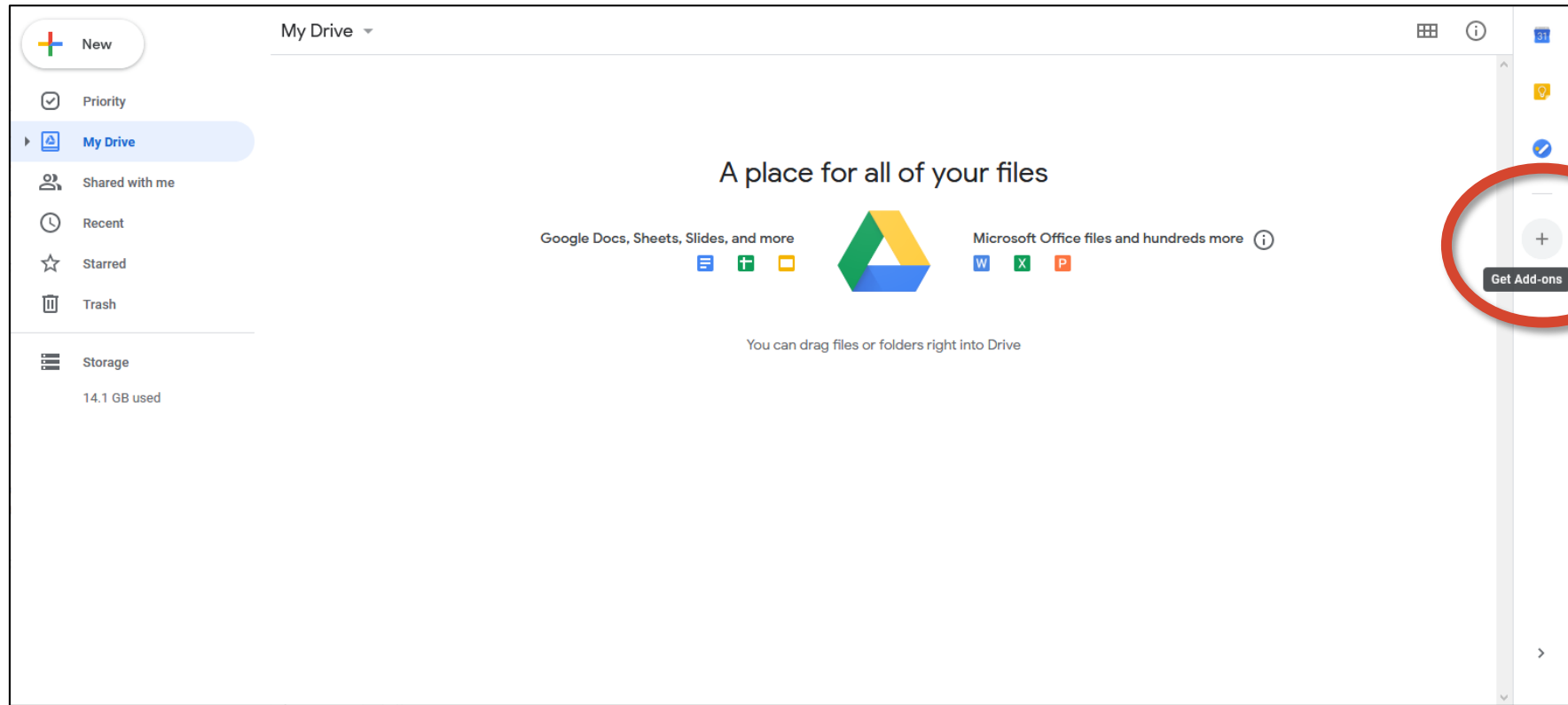


Install Google Colaboratory Add-on and Add Notebooks  
to 'Colab Notebooks' folder on Google Drive



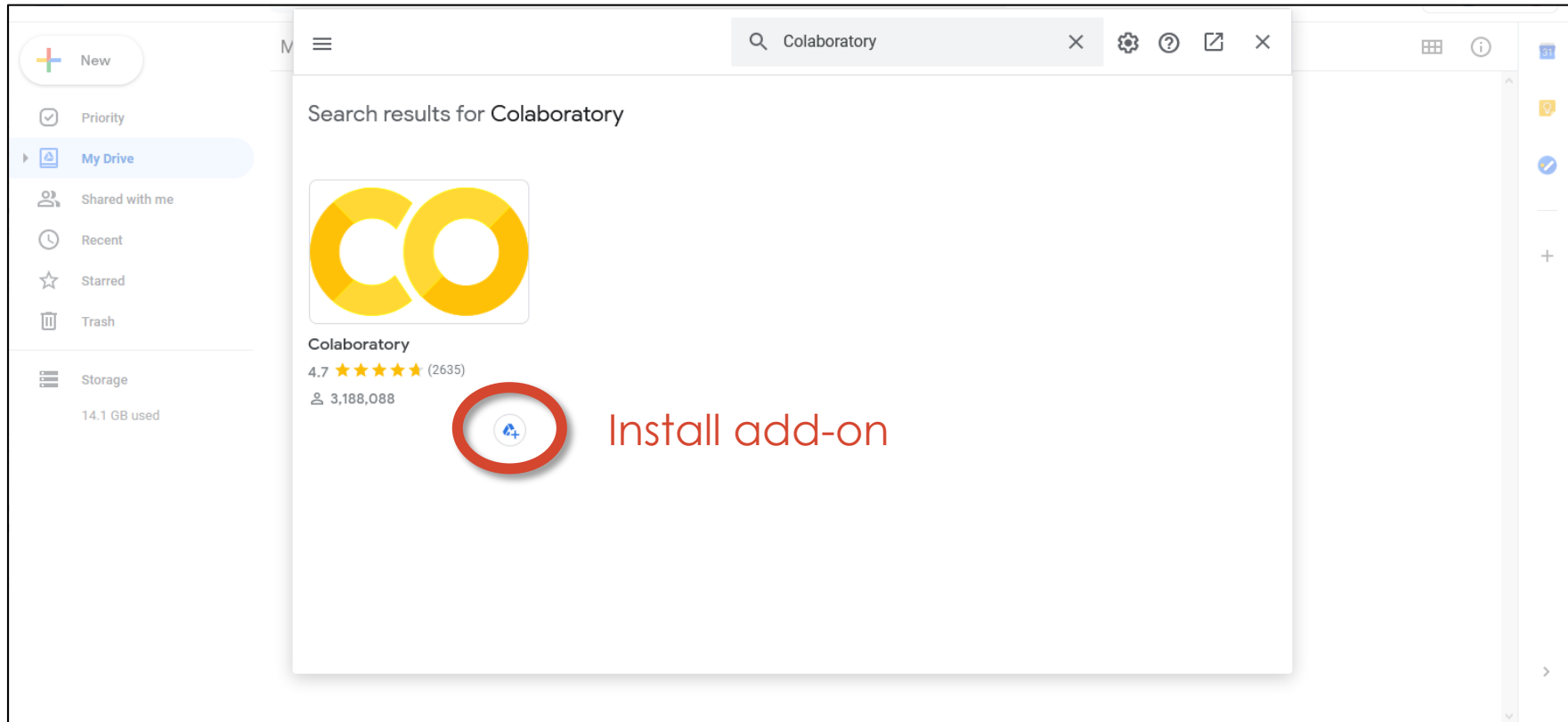
# Add Your Notebooks to Google Drive

Step 1: Go to [drive.google.com](https://drive.google.com) and click the + on the right to add add-ons.



# Add Your Notebooks to Google Drive

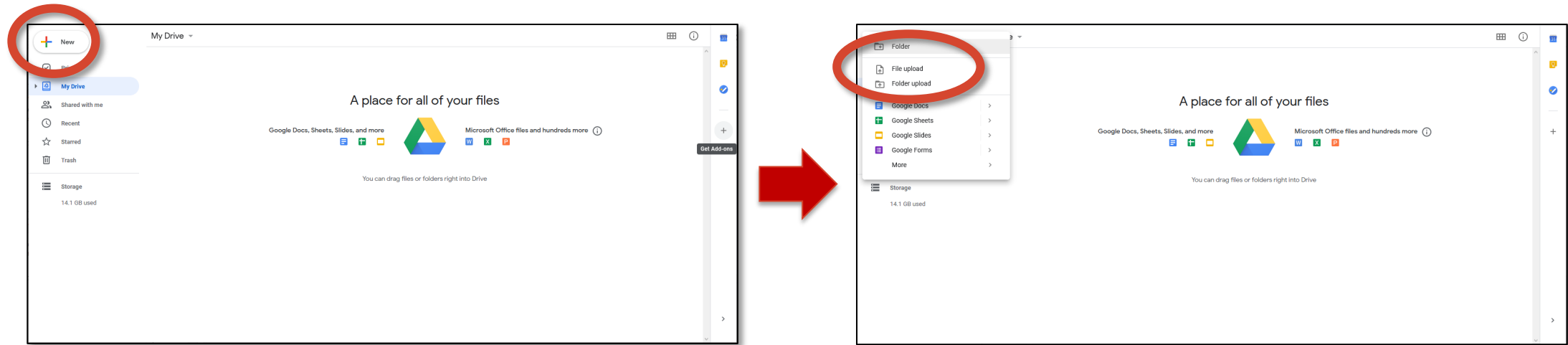
## Step 2: Search for “Colaboratory” and install.



# Add Your Notebooks to Google Drive

**Step 3: Add Notebook to Google Drive by dragging over files, or clicking New → File Upload.**

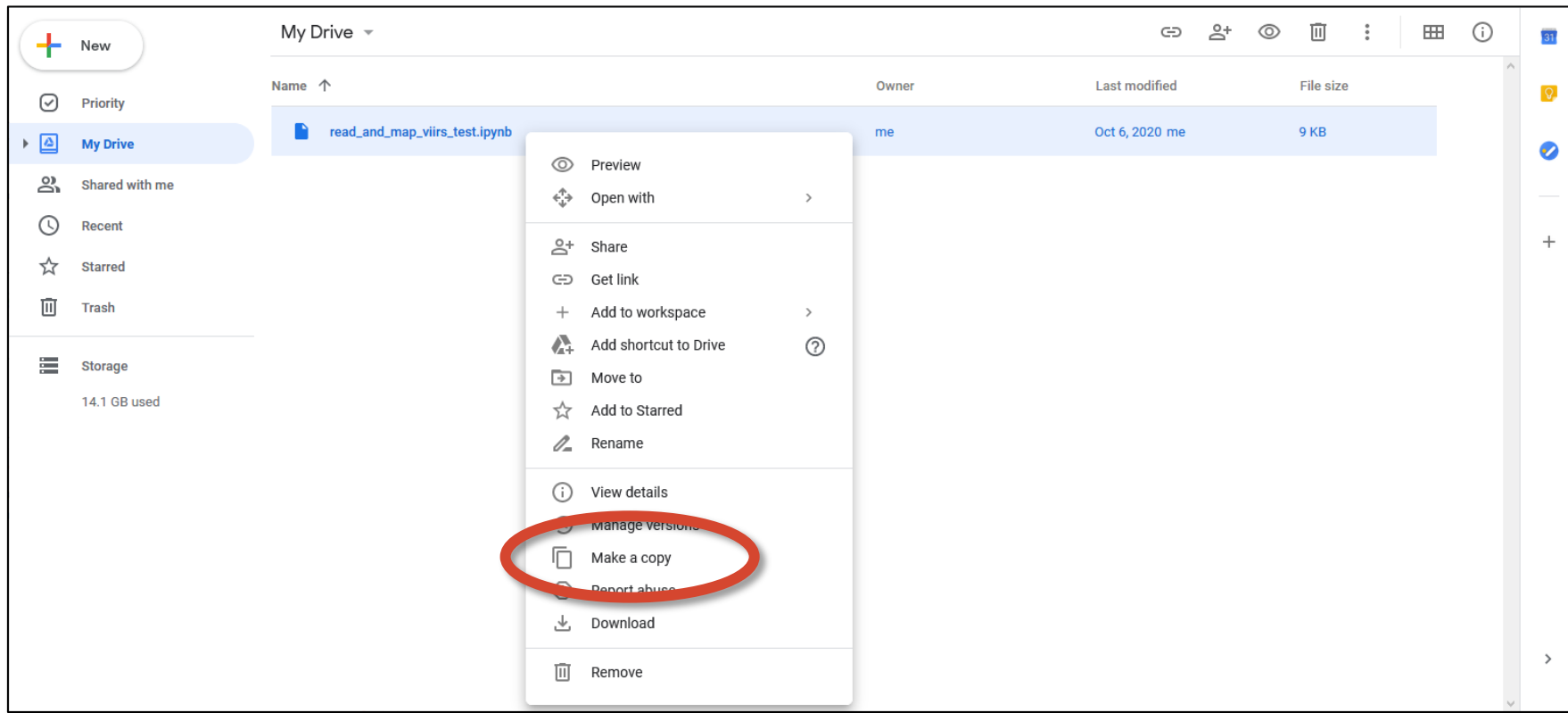
\* If you already had Colaboratory installed, add the file to your Colab Notebooks folder. \*

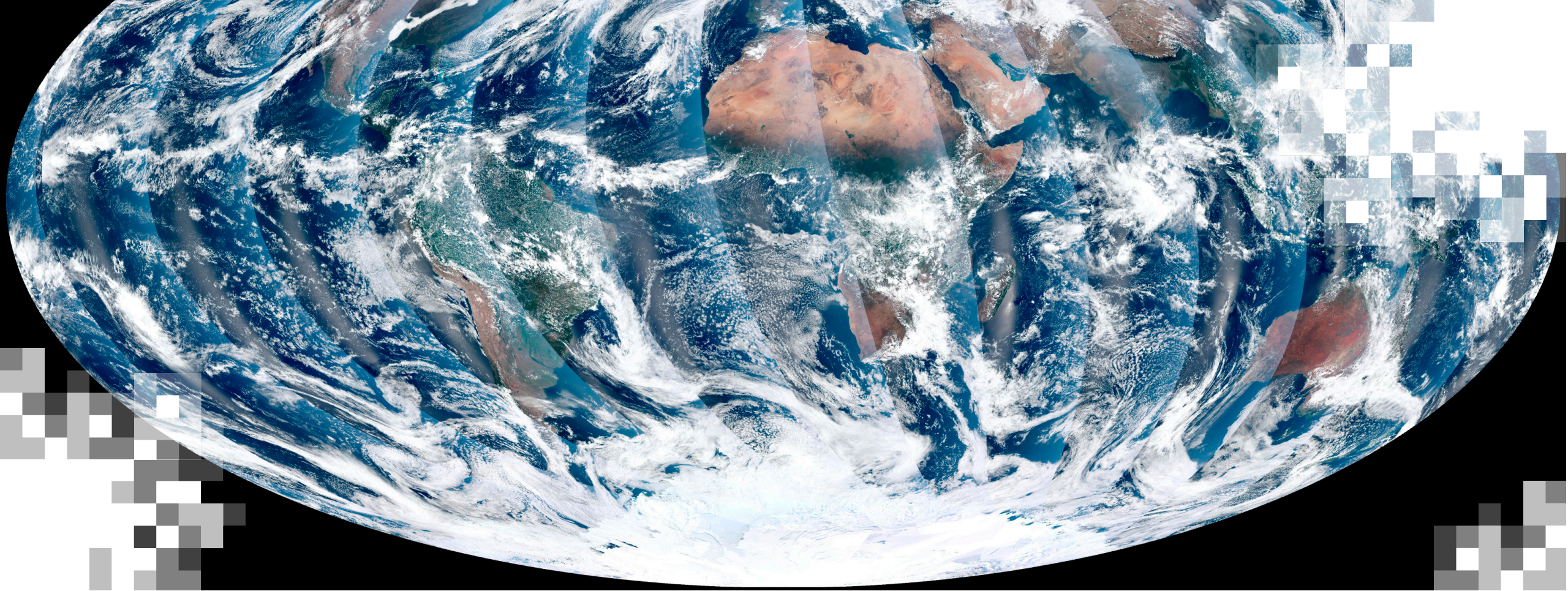


# Add Your Notebooks to Google Drive

**Step 3a: Right-click on your file and click “Make a copy”. This will create the Colab Notebooks folder in your Google Drive. The file copy will be inside this folder.**

\* This step is only necessary if you had to install Colaboratory. \*



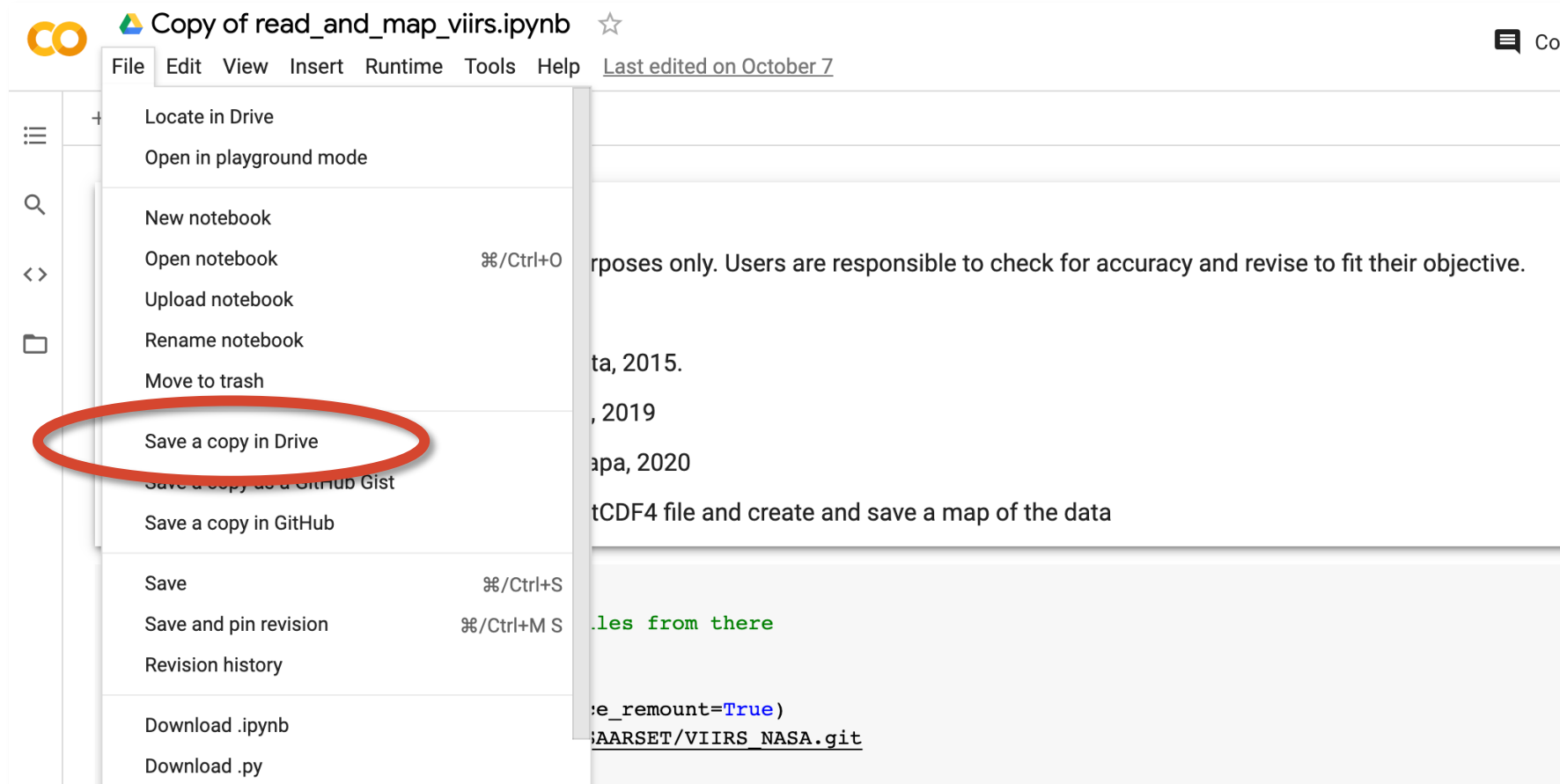


Run Jupyter Notebooks using Google  
Colaboratory



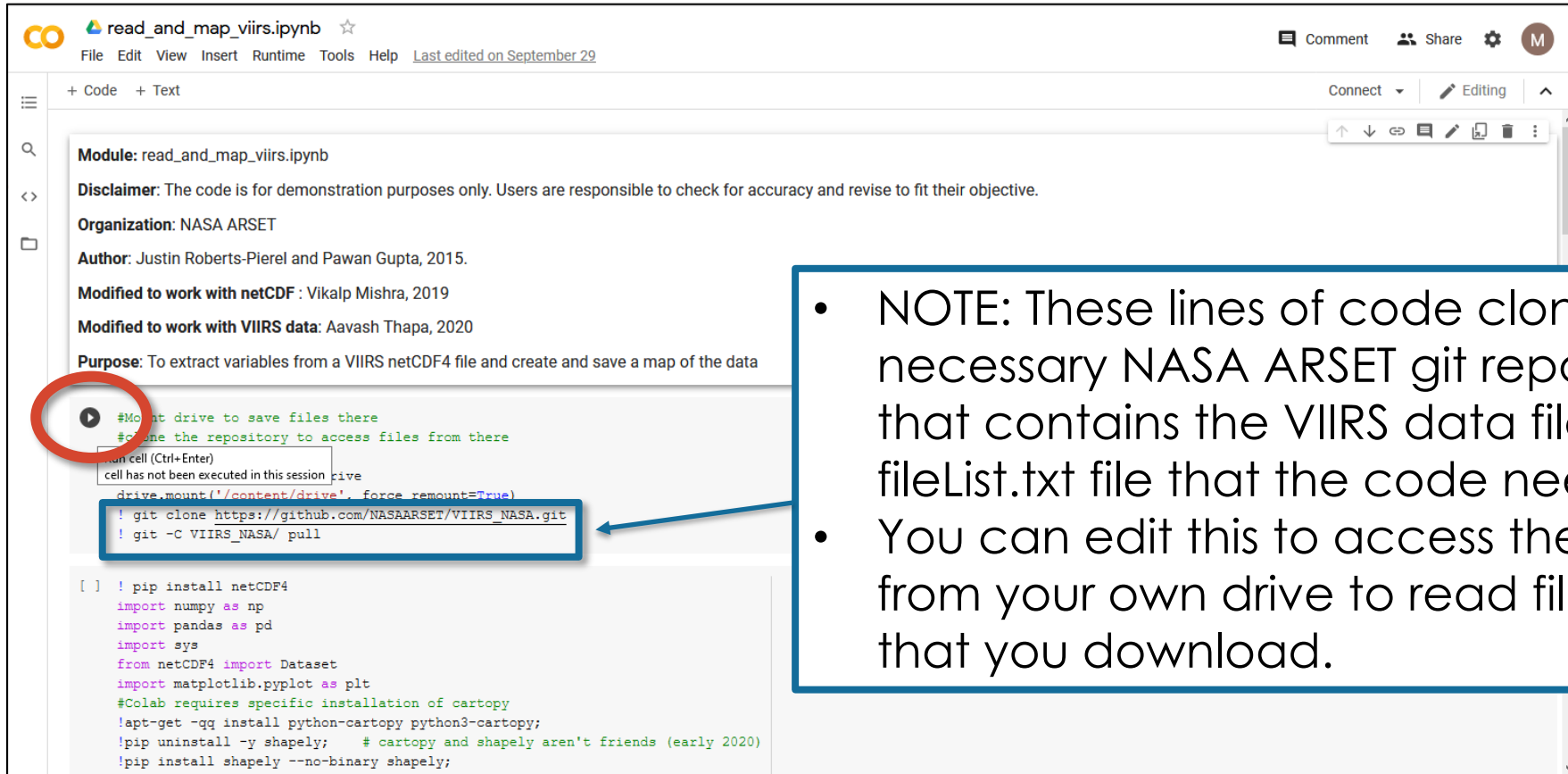
# Run your Jupyter Notebook

Step 1b: Once open, click File > Save a copy in Drive. This will automatically create a duplicate copy in a folder called “Colab Notebooks”.



# Run your Jupyter Notebook

## Step 2: Run each cell of your notebook in order.



The screenshot shows a Jupyter Notebook titled "read\_and\_map\_viirs.ipynb". The notebook contains several text cells with metadata and a code cell. The code cell contains the following text:

```
#Mount drive to save files there
#clone the repository to access files from there
! git clone https://github.com/NASAARSET/VIIRS_NASA.git
! git -C VIIRS_NASA/ pull
```

The play button icon for the code cell is circled in red. A blue box highlights the git clone command, with a blue arrow pointing from the text box on the right to it.

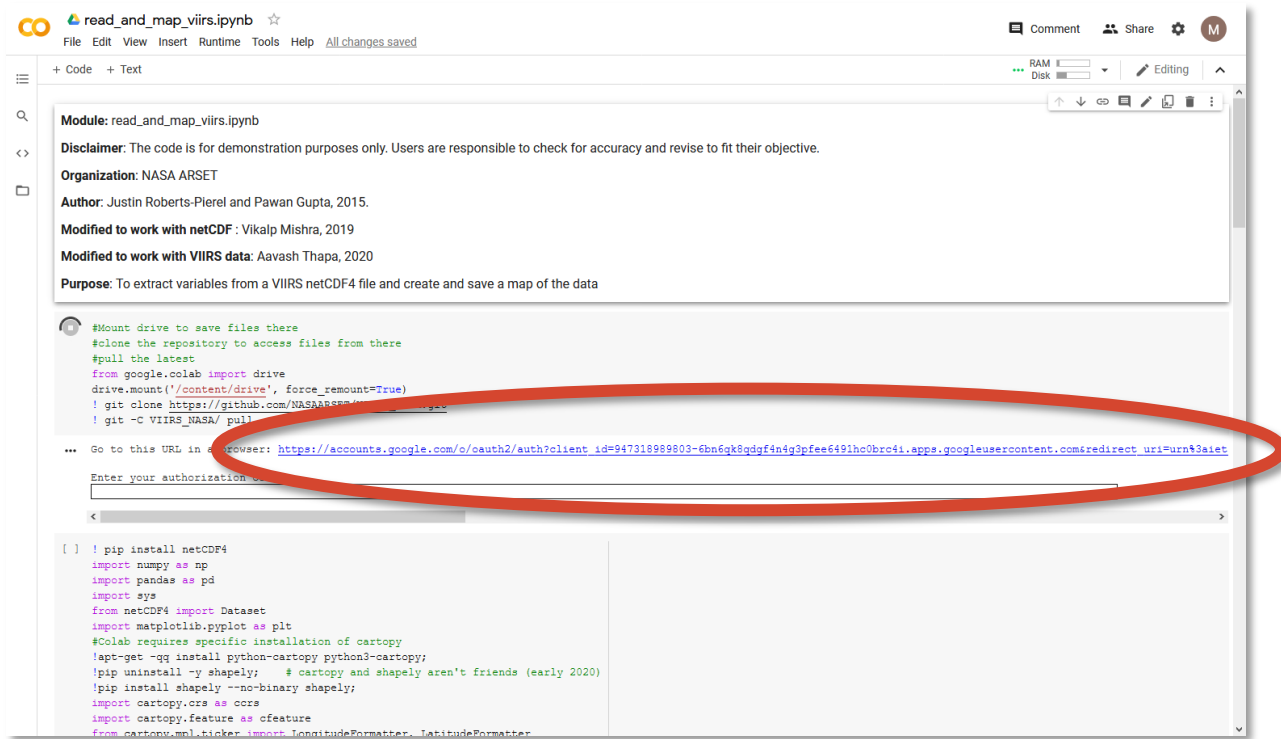
- NOTE: These lines of code clone the necessary NASA ARSET git repository that contains the VIIRS data file and fileList.txt file that the code needs.
- You can edit this to access these from your own drive to read files that you download.





# Run your Jupyter Notebook

Step 2a: The first time you run each code you will need to get an authorization code from Google File Stream. Click the link and choose your google drive account.

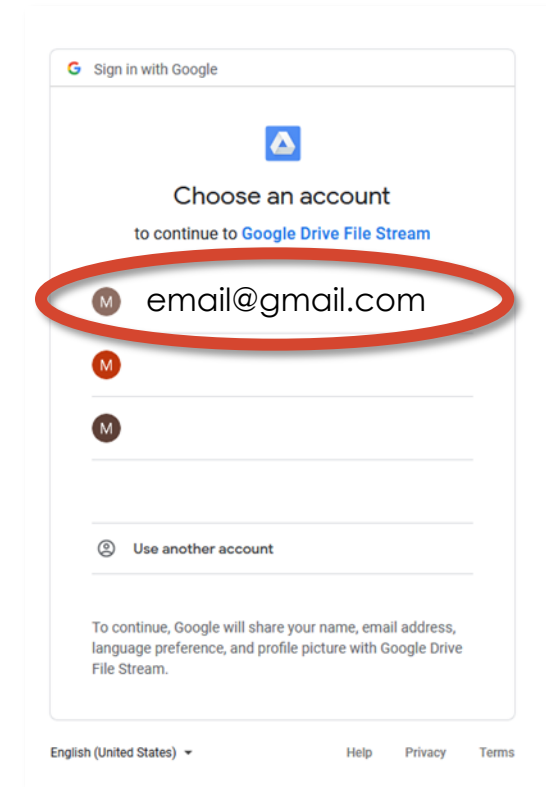


The screenshot shows a Jupyter Notebook titled "read\_and\_map\_viirs.ipynb". The code cell contains the following text:

```
#Mount drive to save files there
#clone the repository to access files from there
#pull the latest
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
! git clone https://github.com/NASAARSET/VIIRS_NASA.git
! git -C VIIRS_NASA/ pull
```

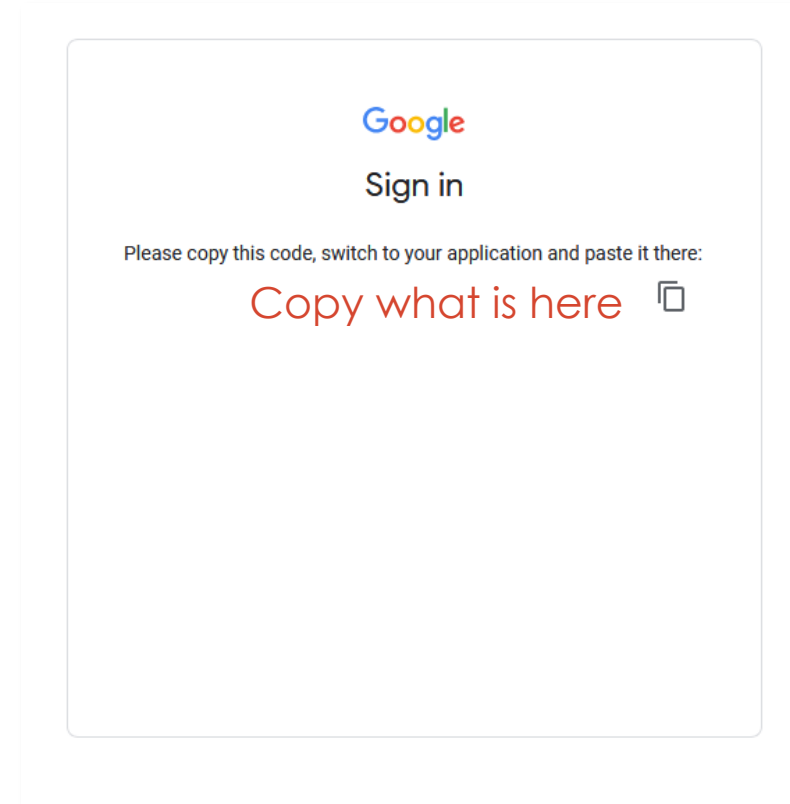
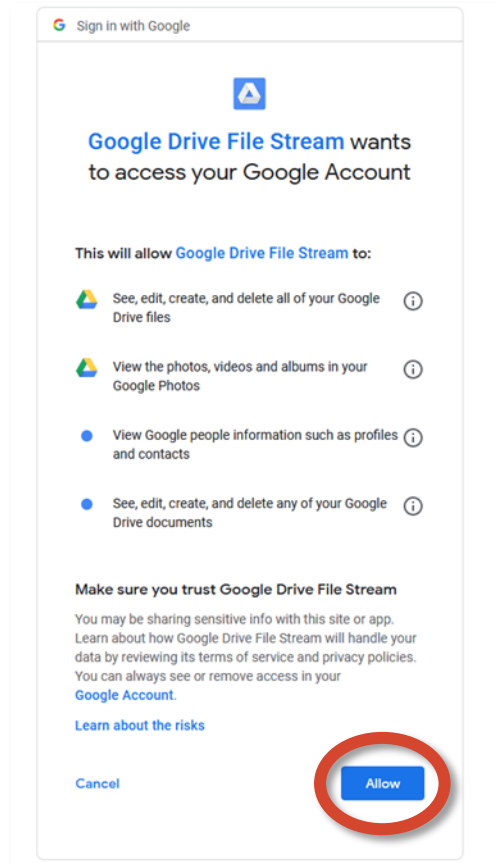
Below the code, there is a red circle around the following URL:

Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6gk8qdgf4nig3pfee6491hc0brc4i.apps.googleusercontent.com&redirect\\_uri=urn%3Aietf:ipynb](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6gk8qdgf4nig3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf:ipynb)



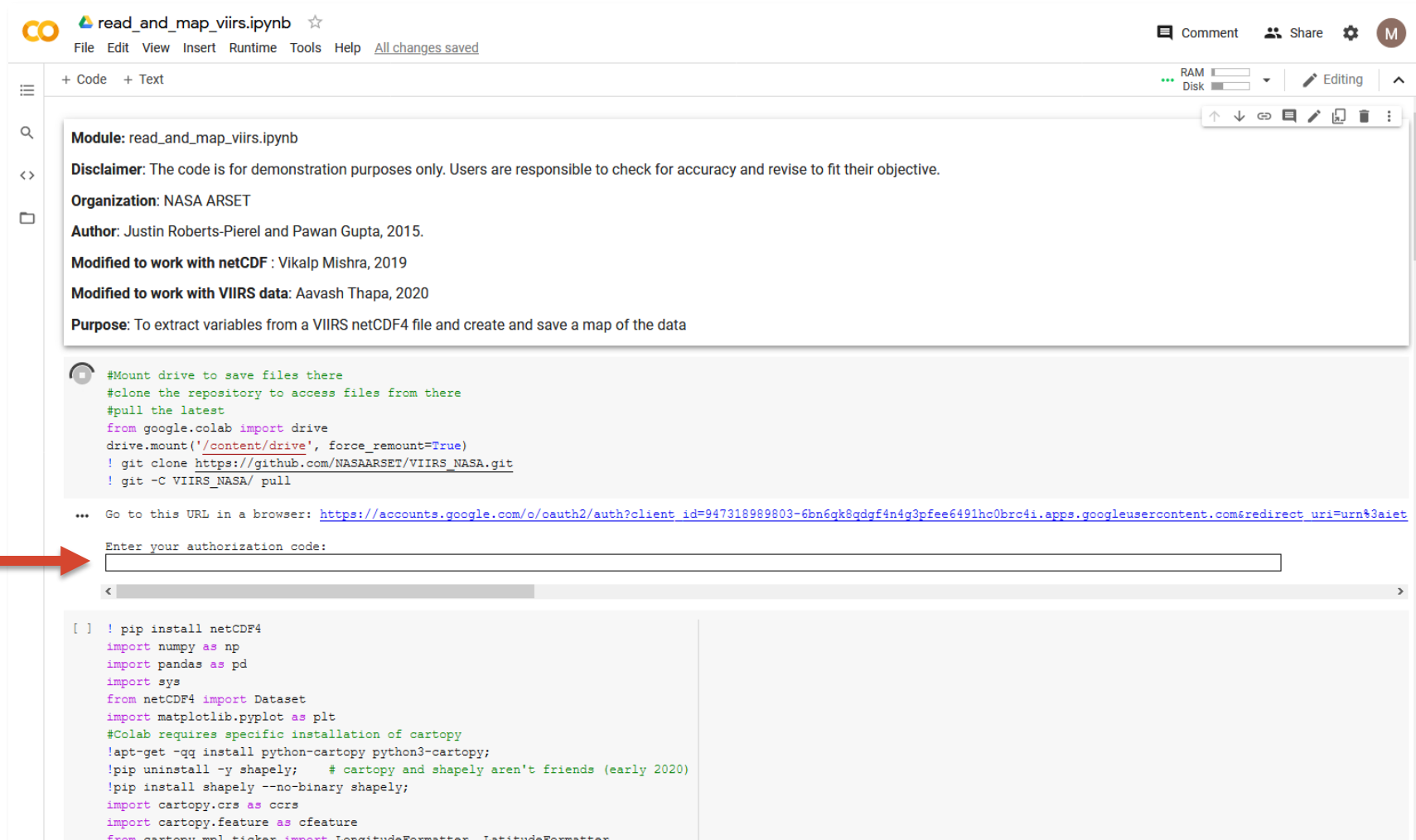
# Run your Jupyter Notebook

## Step 2b: Click Allow and copy the code.



# Run your Jupyter Notebook

## Step 2c: Paste the code into the notebook and hit Enter.



The screenshot shows a Jupyter Notebook titled "read\_and\_map\_viirs.ipynb". The notebook contains a code cell with the following content:

```
Module: read_and_map_viirs.ipynb
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objective.
Organization: NASA ARSET
Author: Justin Roberts-Pierel and Pawan Gupta, 2015.
Modified to work with netCDF: Vikalp Mishra, 2019
Modified to work with VIIRS data: Aavash Thapa, 2020
Purpose: To extract variables from a VIIRS netCDF4 file and create and save a map of the data

#Mount drive to save files there
#clone the repository to access files from there
#pull the latest
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
! git clone https://github.com/NASAARSET/VIIRS_NASA.git
! git -C VIIRS_NASA/ pull

... Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6gk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aie

Enter your authorization code:

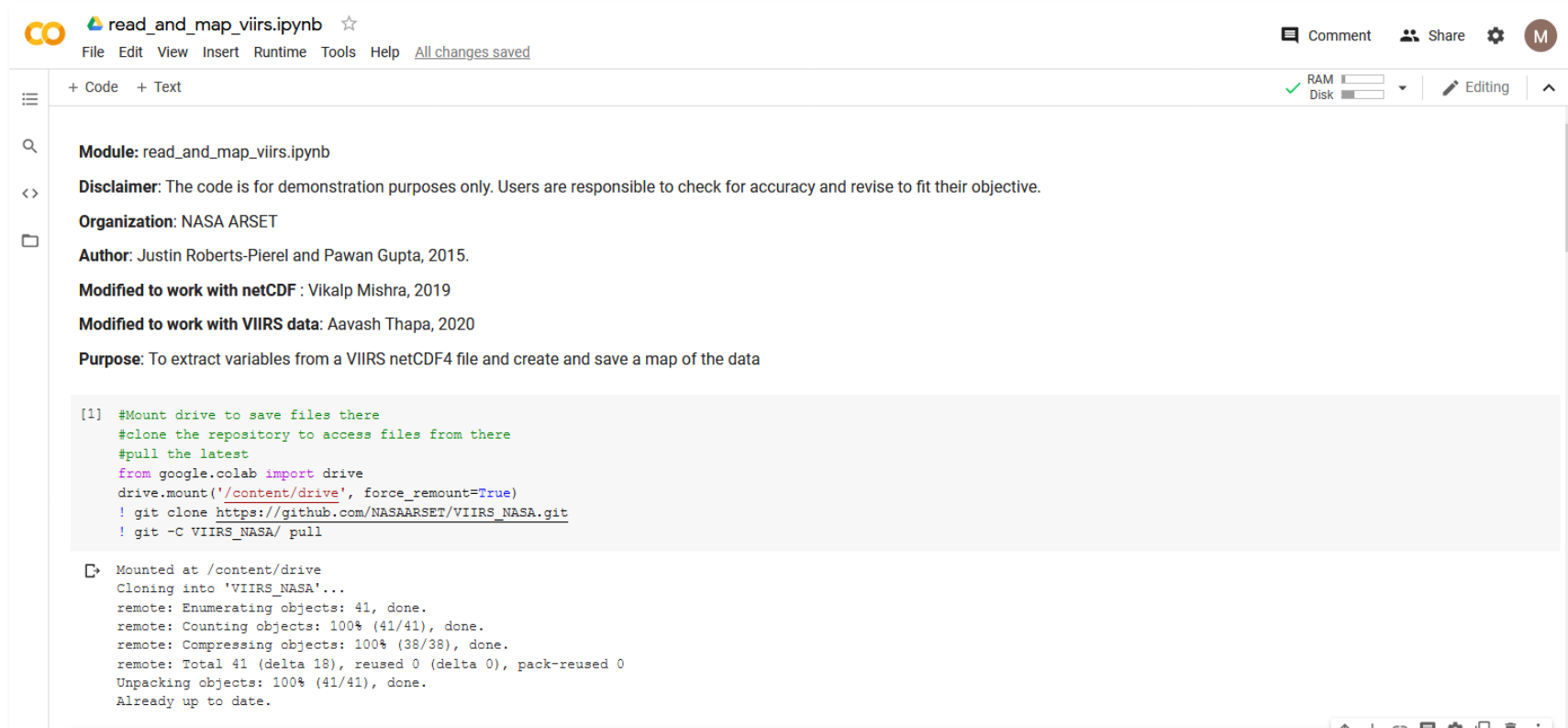

[ ] ! pip install netCDF4
import numpy as np
import pandas as pd
import sys
from netCDF4 import Dataset
import matplotlib.pyplot as plt
#Colab requires specific installation of cartopy
!apt-get -qq install python-cartopy python3-cartopy;
!pip uninstall -y shapely; # cartopy and shapely aren't friends (early 2020)
!pip install shapely --no-binary shapely;
import cartopy.crs as ccrs
import cartopy.feature as cfeature
from cartopy.mpl.ticker import LongitudeFormatter, LatitudeFormatter
```

Paste here



# Run your Jupyter Notebook



## Step 2c: Paste the code into the notebook and hit Enter.



The screenshot shows a Jupyter Notebook interface with the following content:

read\_and\_map\_viirs.ipynb ☆  
File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

RAM  Disk  Editing

**Module:** read\_and\_map\_viirs.ipynb

**Disclaimer:** The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objective.

**Organization:** NASA ARSET

**Author:** Justin Roberts-Pierel and Pawan Gupta, 2015.

**Modified to work with netCDF:** Vikalp Mishra, 2019

**Modified to work with VIIRS data:** Aavash Thapa, 2020

**Purpose:** To extract variables from a VIIRS netCDF4 file and create and save a map of the data

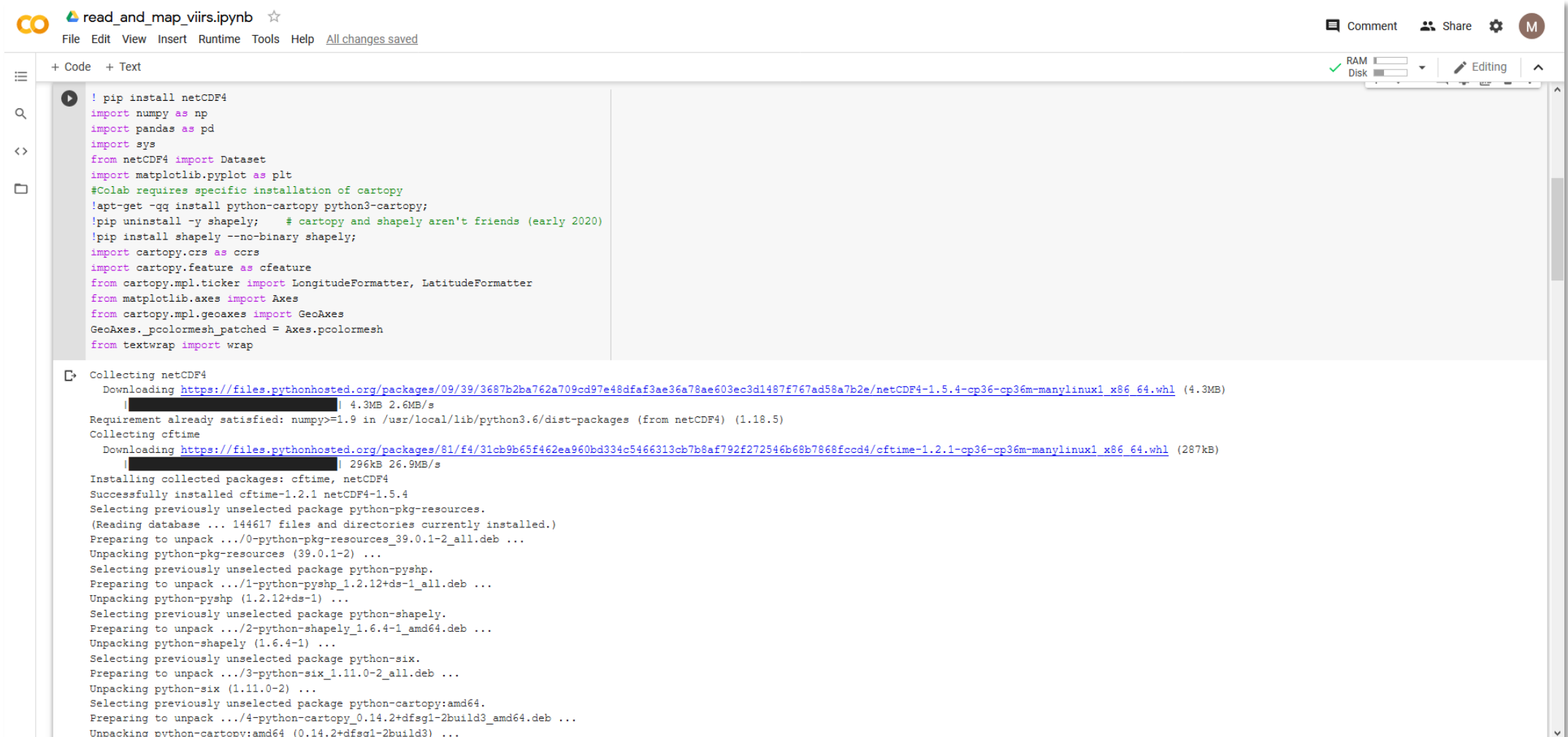
```
[1] #Mount drive to save files there
#clone the repository to access files from there
#pull the latest
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
! git clone https://github.com/NASAARSET/VIIRS_NASA.git
! git -C VIIRS_NASA/ pull
```

Mounted at /content/drive  
Cloning into 'VIIRS\_NASA'...  
remote: Enumerating objects: 41, done.  
remote: Counting objects: 100% (41/41), done.  
remote: Compressing objects: 100% (38/38), done.  
remote: Total 41 (delta 18), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (41/41), done.  
Already up to date.



# Run your Jupyter Notebook

## Step 3: Run the next cell to import the necessary libraries.



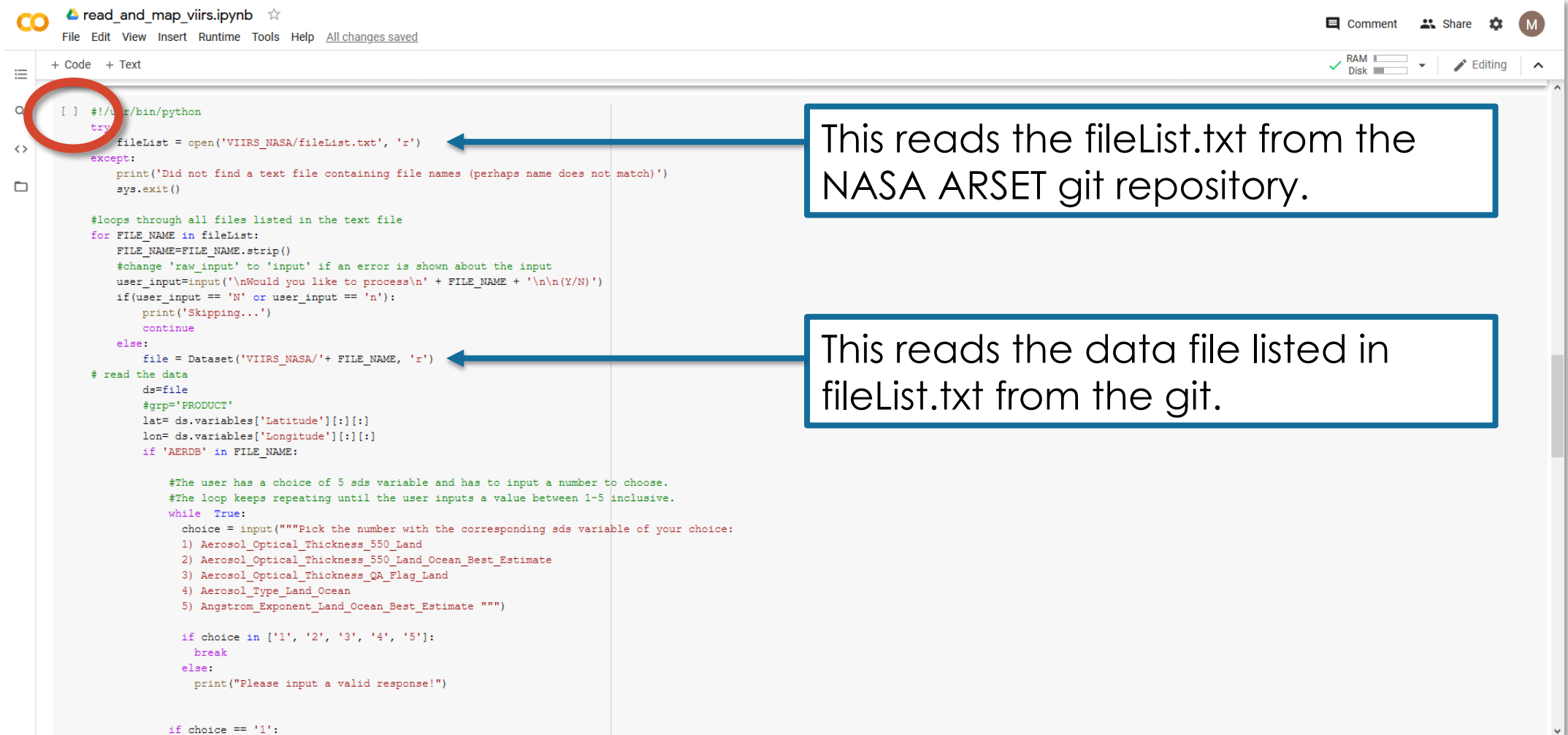
```
! pip install netCDF4
import numpy as np
import pandas as pd
import sys
from netCDF4 import Dataset
import matplotlib.pyplot as plt
#Colab requires specific installation of cartopy
!apt-get -qq install python-cartopy python3-cartopy;
!pip uninstall -y shapely; # cartopy and shapely aren't friends (early 2020)
!pip install shapely --no-binary shapely;
import cartopy.crs as ccrs
import cartopy.feature as cfeature
from cartopy.mpl.ticker import LongitudeFormatter, LatitudeFormatter
from matplotlib.axes import Axes
from cartopy.mpl.geoaxes import GeoAxes
GeoAxes._pcolormesh_patched = Axes.pcolormesh
from textwrap import wrap
```

Collecting netCDF4  
Downloading [https://files.pythonhosted.org/packages/09/39/3687b2ba709cd97e48dfaf3ae36a78ae603ec3d1487f767ad58a7b2e/netCDF4-1.5.4-cp36-cp36m-manylinux1\\_x86\\_64.whl](https://files.pythonhosted.org/packages/09/39/3687b2ba709cd97e48dfaf3ae36a78ae603ec3d1487f767ad58a7b2e/netCDF4-1.5.4-cp36-cp36m-manylinux1_x86_64.whl) (4.3MB)  
| ██████████ 4.3MB 2.6MB/s  
Requirement already satisfied: numpy>=1.9 in /usr/local/lib/python3.6/dist-packages (from netCDF4) (1.18.5)  
Collecting cftime  
Downloading [https://files.pythonhosted.org/packages/81/f4/31cb9b65f462ea960bd334c5466313cb7b8af792f272546b68b7868fced4/cftime-1.2.1-cp36-cp36m-manylinux1\\_x86\\_64.whl](https://files.pythonhosted.org/packages/81/f4/31cb9b65f462ea960bd334c5466313cb7b8af792f272546b68b7868fced4/cftime-1.2.1-cp36-cp36m-manylinux1_x86_64.whl) (287kB)  
| ██████████ 296kB 26.9MB/s  
Installing collected packages: cftime, netCDF4  
Successfully installed cftime-1.2.1 netCDF4-1.5.4  
Selecting previously unselected package python-pkg-resources.  
(Reading database ... 144617 files and directories currently installed.)  
Preparing to unpack .../0-python-pkg-resources\_39.0.1-2\_all.deb ...  
Unpacking python-pkg-resources (39.0.1-2) ...  
Selecting previously unselected package python-pyshp.  
Preparing to unpack .../1-python-pyshp\_1.2.12+ds-1\_all.deb ...  
Unpacking python-pyshp (1.2.12+ds-1) ...  
Selecting previously unselected package python-shapely.  
Preparing to unpack .../2-python-shapely\_1.6.4-1\_amd64.deb ...  
Unpacking python-shapely (1.6.4-1) ...  
Selecting previously unselected package python-six.  
Preparing to unpack .../3-python-six\_1.11.0-2\_all.deb ...  
Unpacking python-six (1.11.0-2) ...  
Selecting previously unselected package python-cartopy:amd64.  
Preparing to unpack .../4-python-cartopy\_0.14.2+dfsg1-2build3\_amd64.deb ...  
Unpacking python-cartopy:amd64 (0.14.2+dfsg1-2build3) ...



# Run your Jupyter Notebook

## Step 4: Run the last cell to run the code.



The screenshot shows a Jupyter Notebook window titled "read\_and\_map\_viirs.ipynb". The code in the cell is as follows:

```
[ ] #!/usr/bin/python
try:
    fileList = open('VIIRS_NASA/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()
    #change 'raw_input' to 'input' if an error is shown about the input
    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if(user_input == 'N' or user_input == 'n'):
        print('Skipping...')
        continue
    else:
        file = Dataset('VIIRS_NASA/'+ FILE_NAME, 'r')
# read the data
ds=file
#grp='PRODUCT'
lat= ds.variables['Latitude'][:,:]
lon= ds.variables['Longitude'][:,:]
if 'AERDB' in FILE_NAME:

#The user has a choice of 5 sds variable and has to input a number to choose.
#The loop keeps repeating until the user inputs a value between 1-5 inclusive.
while True:
    choice = input("""Pick the number with the corresponding sds variable of your choice:
1) Aerosol_Optical_Thickness_550_Land
2) Aerosol_Optical_Thickness_550_Land_Ocean_Best_Estimate
3) Aerosol_Optical_Thickness_QA_Flag_Land
4) Aerosol_Type_Land_Ocean
5) Angstrom_Exponent_Land_Ocean_Best_Estimate """)

    if choice in ['1', '2', '3', '4', '5']:
        break
    else:
        print("Please input a valid response!")

if choice == '1':
```

Two callout boxes with blue borders and arrows point to specific lines of code:

- The first callout box points to the line `fileList = open('VIIRS_NASA/fileList.txt', 'r')` and contains the text: "This reads the fileList.txt from the NASA ARSET git repository."
- The second callout box points to the line `file = Dataset('VIIRS_NASA/'+ FILE_NAME, 'r')` and contains the text: "This reads the data file listed in fileList.txt from the git."



# Run your Jupyter Notebook

## Step 4: Enter Y to process the file.



The screenshot shows a Jupyter Notebook titled "read\_and\_map\_viirs.ipynb". The code in the cell is as follows:

```
cb = plt.colorbar(shrink = 0.7)
cb.set_label(map_label, fontsize =9, wrap=True)
"""
    grd = m.gridlines(crs=ccrs.PlateCarree(), draw_labels=True, linewidth=2, color='gray', alpha=0.5, linestyle='--')
    grd.xlabels_top = None
    grd.ylabels_right = None
    grd.xformatter = LONGITUDE_FORMATTER
    grd.yformatter = LATITUDE_FORMATTER
    """
# Show the plot window.
plt.show()
#once you close the map it asks if you'd like to save it
#change 'raw_input' to 'input' if an error is shown about the input
is_save=str(input('\nWould you like to save this map? Please enter Y or N \n'))
if is_save == 'Y' or is_save == 'y':
    #saves as a png if the user would like
    pngfile = '{0}.png'.format(FILE_NAME[:-3])
    fig.savefig('/content/drive/My Drive/Colab Notebooks/' + pngfile, dpi = 300, bbox_inches='tight')
#close the hdf5 file
file.close()
```

Below the code, the notebook output shows a prompt: "Would you like to process" followed by a file path. Below the prompt is an input field containing "(Y/N)". A red circle highlights this input field. A blue box highlights the save path in the code: `fig.savefig('/content/drive/My Drive/Colab Notebooks/' + pngfile, dpi = 300, bbox_inches='tight')`. A blue arrow points from this box to the text "This is where your map will be saved."



# Run your Jupyter Notebook

**Step 5: Enter 2 (Aerosol\_Optical\_Thickness\_550\_Land\_Ocean\_Best\_Estimate) and then enter Y to create a map. Then enter Y to save it to your Google Drive/Colab Notebooks/ folder.**

```
read_and_map_viirs.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
#close the hdf5 file
file.close()

Would you like to process
AERDB_L2_VIIRS_SNPP.A2020056.1954.001.2020057113600.nc

(Y/N) Y
Pick the number with the appropriate description of your choice:
1) Aerosol_Optical_Thickness_550_Land
2) Aerosol_Optical_Thickness_550_Land_Ocean_Best_Estimate
3) Aerosol_Optical_Thickness_QA_Flag_Land
4) Aerosol_Type_Land_Ocean
5) Angstrom_Exponent_Land_Ocean_Best_Estimate 2

The average of this data is: 0.075
The standard deviation is: 0.073
The median is: 0.059
The range of latitude in this file is: 11.471696 to 36.39784 degrees
The range of longitude in this file is: -120.826866 to -85.44587 degrees

Would you like to create a map of this data? Please enter Y or N
Y
/usr/lib/python3/dist-packages/cartopy/io/_init__.py:264: DownloadWarning: Downloading: http://naciscdn.org/naturalearth/110m/physical/ne\_110m\_coastline.zip
warnings.warn("Downloading: {}".format(url), DownloadWarning)

AERDB_L2_VIIRS_SNPP.A2020056.1954.001.2020057113600.nc
Deep Blue/SOAR aerosol optical thickness at 550 nm over land and ocean, QA-filtered

Aerosol Optical Thickness at 550 nm over Land and Ocean Best Estimate

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

