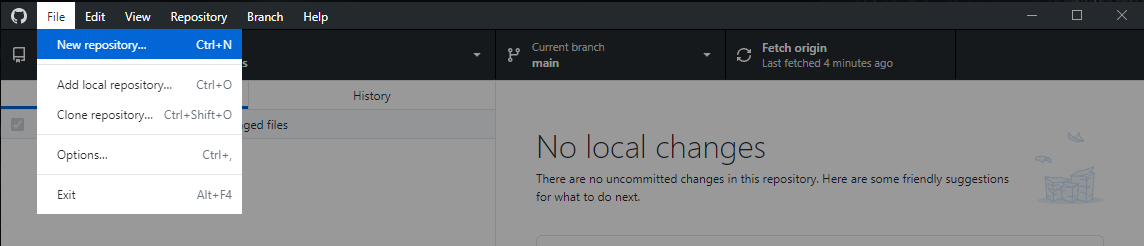
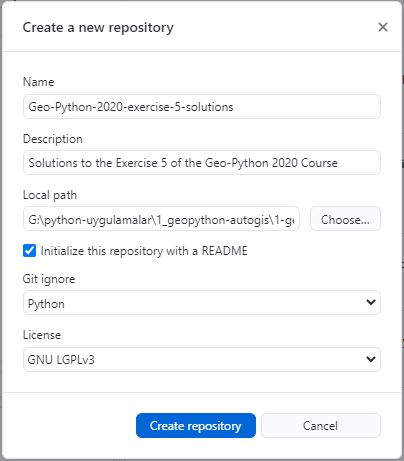
**How to do?**

Dr. Hayati TAŞTAN

September 3, 2021, Ankara

**Download GitHub Desktop for Windows (GitHubDesktopSetup-x64.exe ) from** [**https://desktop.github.com/**](https://desktop.github.com/) **into G:\python-uygulamalar\0\_GitHub\_Desktop\) and install.**

**Create a local repository within GitHub desktop (File / New repository):**

**New repository:**

**Name:** Geo-Python-2020-exercise-5-solutions

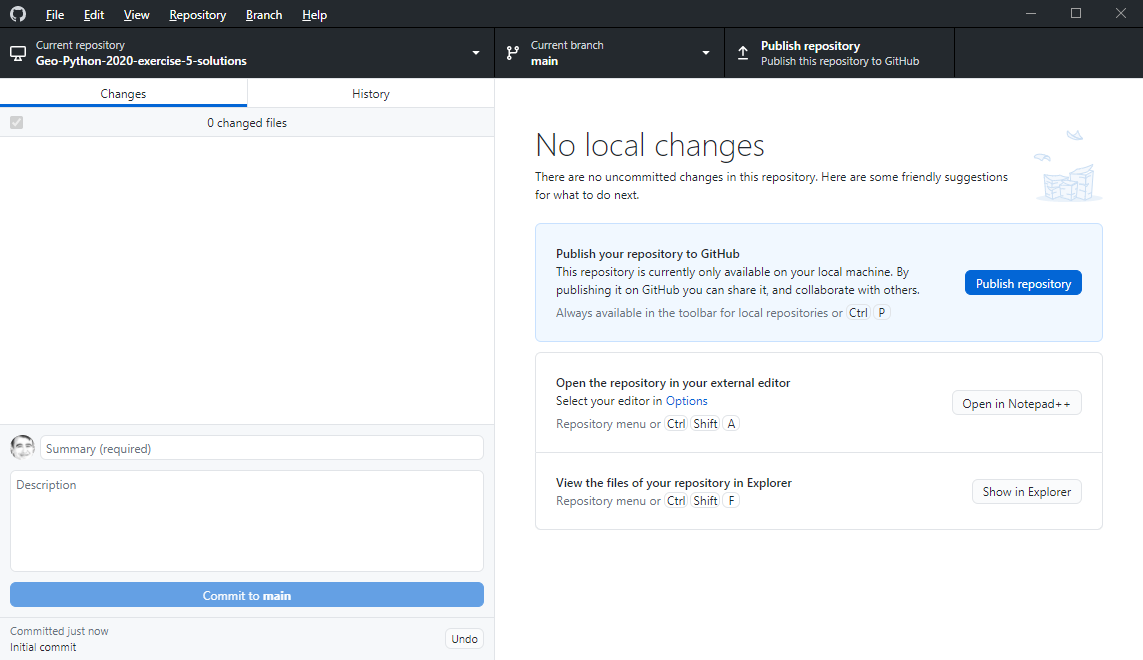
**Description:** Solutions to the Exercise 5 of the Geo-Python 2020 Course

**Local path:** G:\python-uygulamalar\1\_geopython-autogis\1-geo-python-2020\ornekler

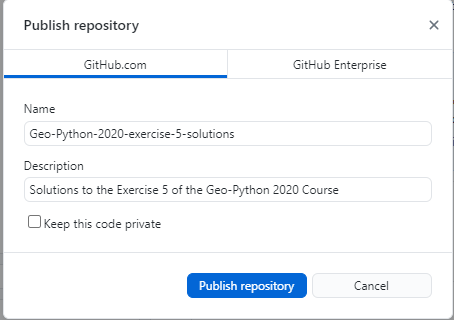
Initialize the reposiyory with a README

**Git ignore:** Python

**License:** GNU LGPLv3

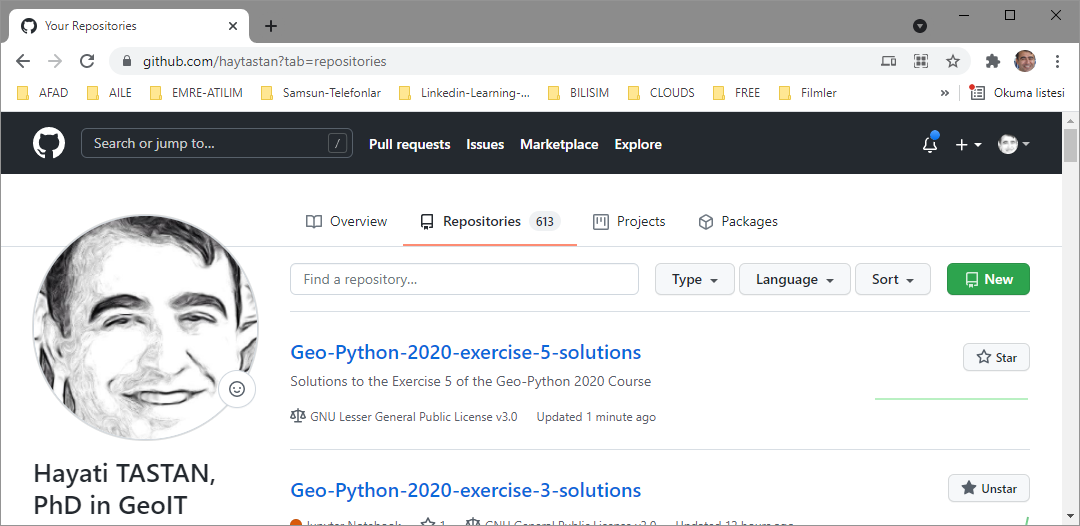
**Create repository:**

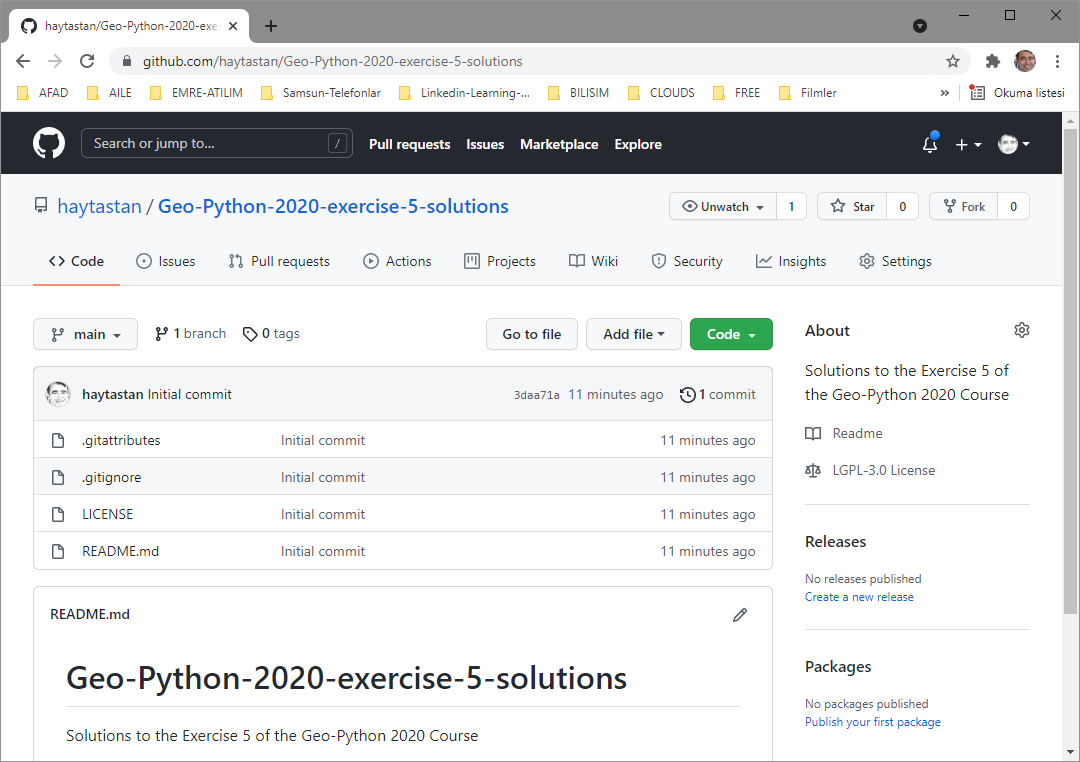
**Publish reposiyory:**

****

**Uncheck Keep this code private**

**Publish repository**

**Sign in to** [**https://github.com/haytastan**](https://github.com/haytastan)

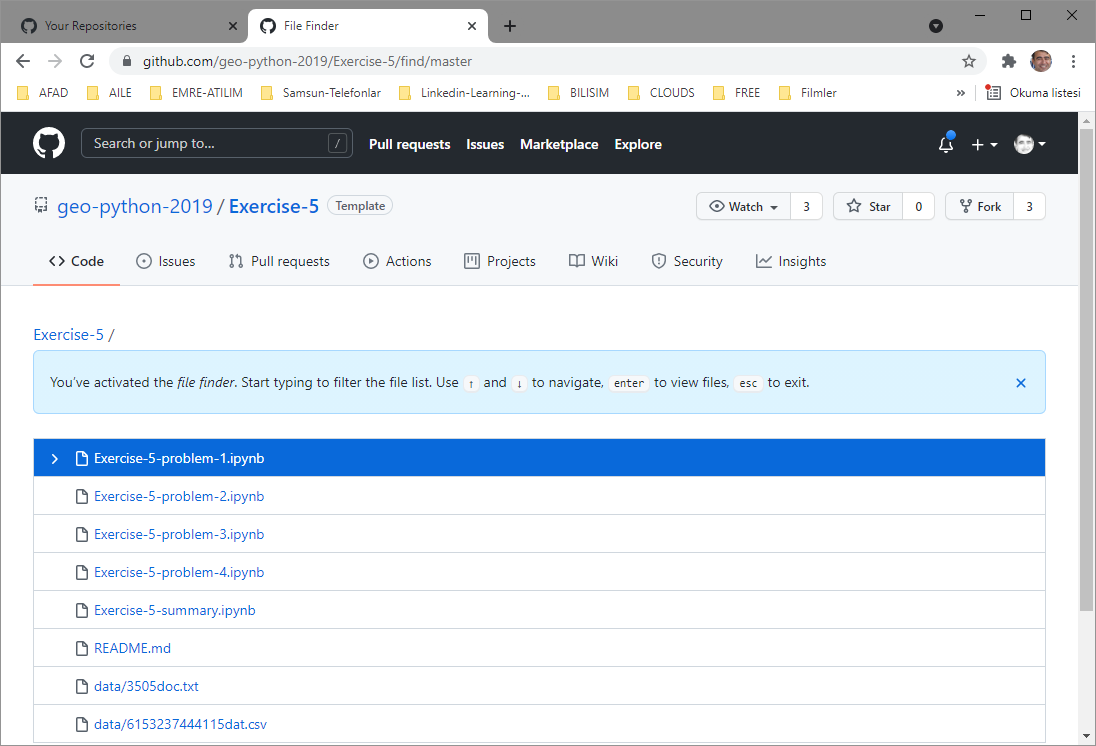
 **Enter the repo** Geo-Python-2020-exercise-5-solutions:

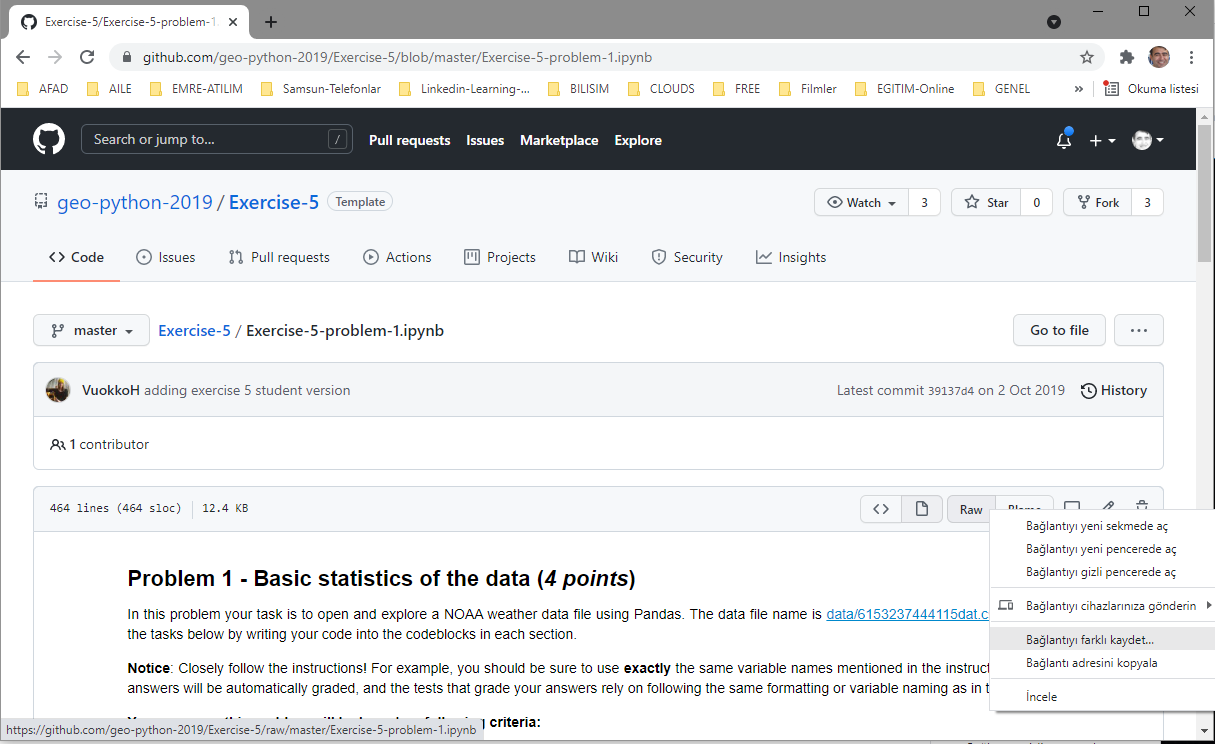
**Download jupyter projects (\*.ipynb) and README.md files from:**

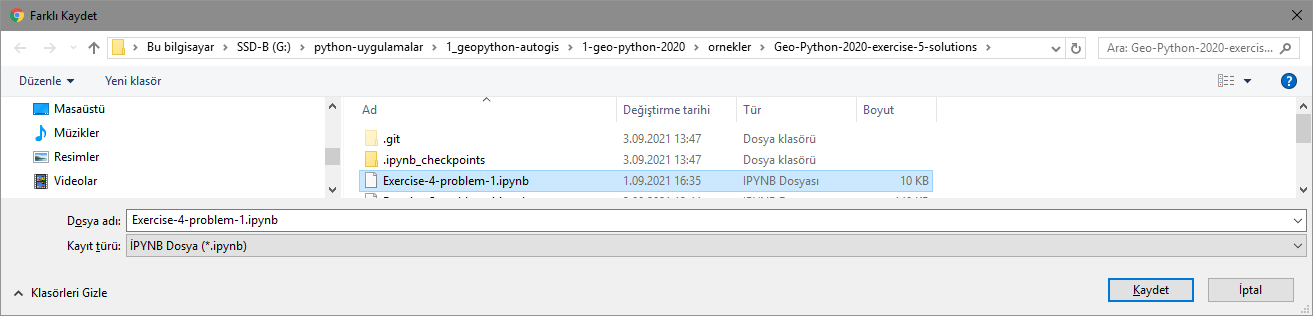
<https://github.com/geo-python-2019/Exercise-5>

**into the local repo:**

G:\python-uygulamalar\1\_geopython-autogis\1-geo-python-2020\ornekler

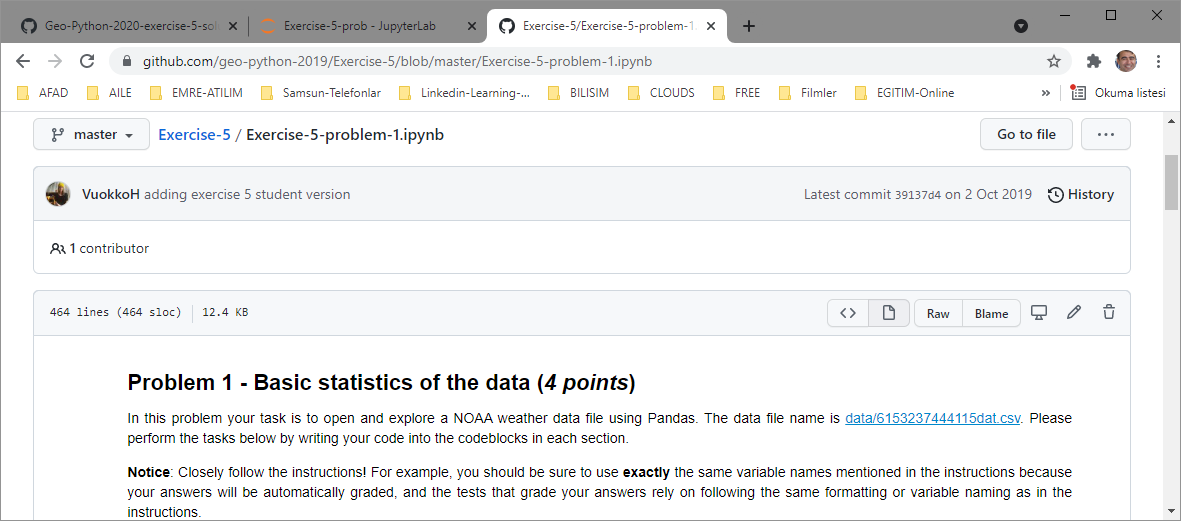
**Go to file:**

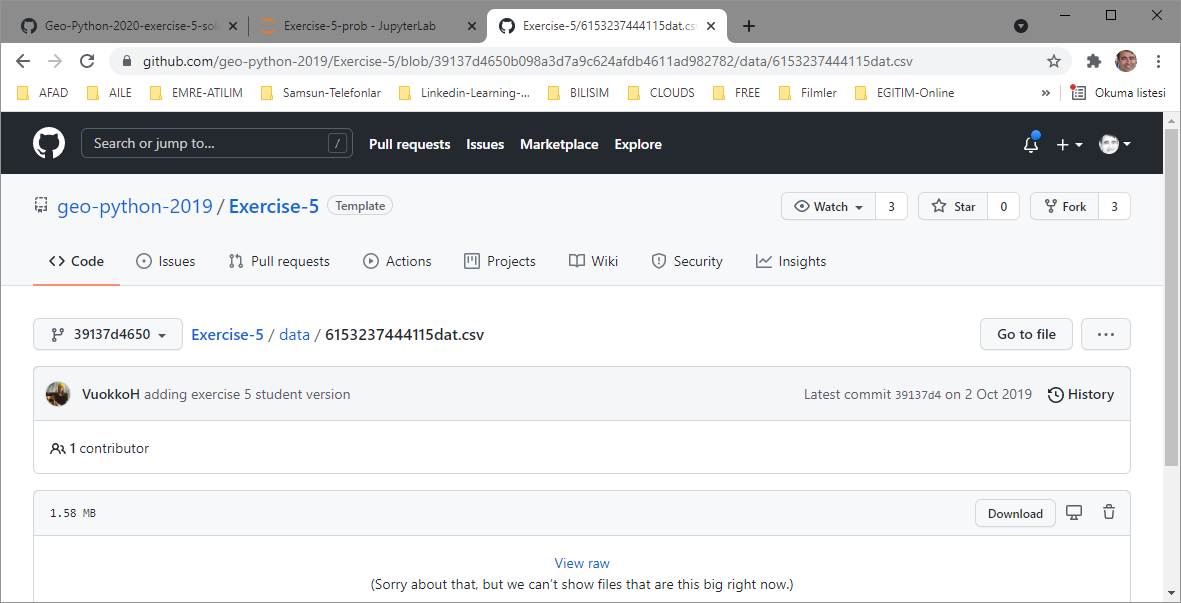
**Select the file to download, double click, goto to the ikon raw, mouse right click and save as:**

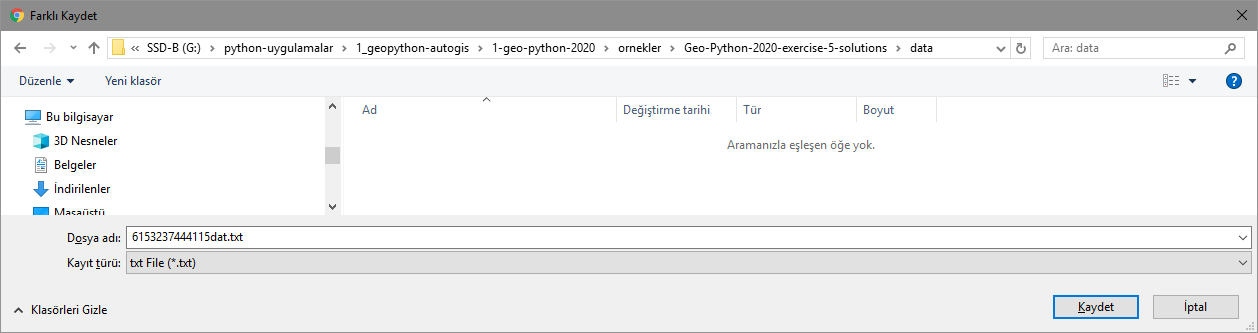
**Save as with the same name (i.e: Exercise-4-problem-1.ipynb)**

**Goto** <https://github.com/geo-python-2019/Exercise-5/find/master>

and repeat this copy procedure for all **\*.ipnyb** and **README.md** files.

Click Exercise-5problem-1.ipynb:

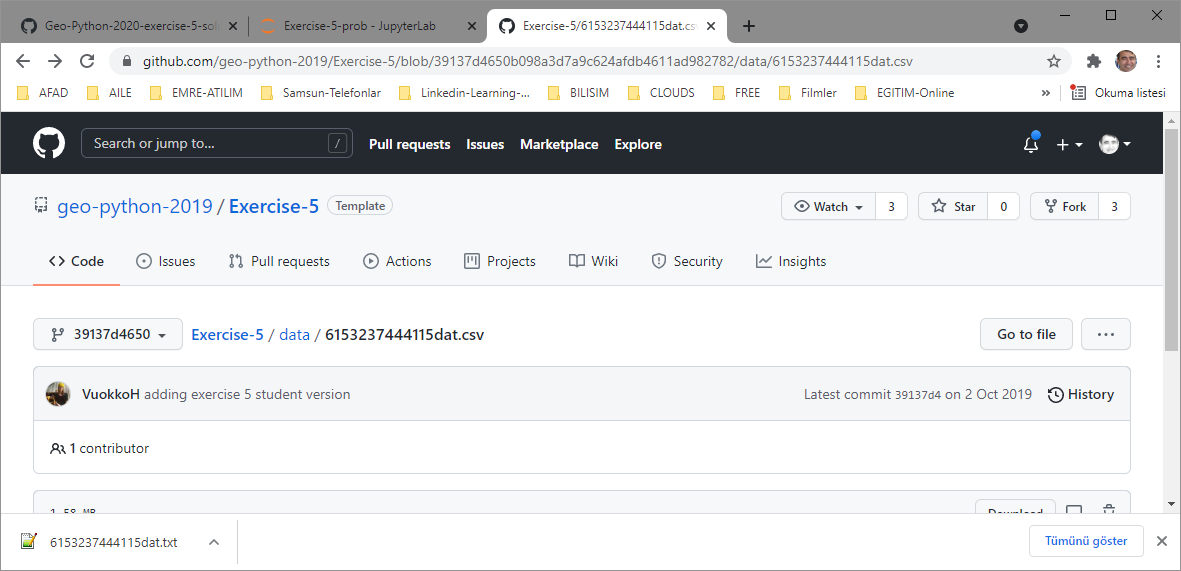
Click csv file:

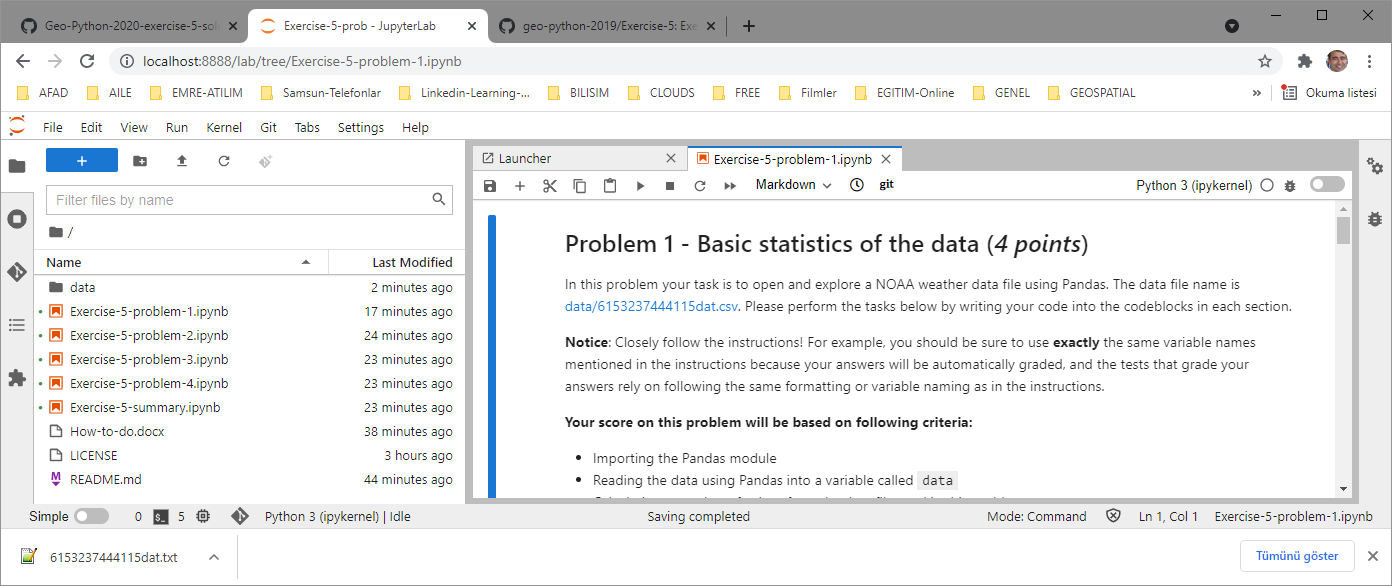
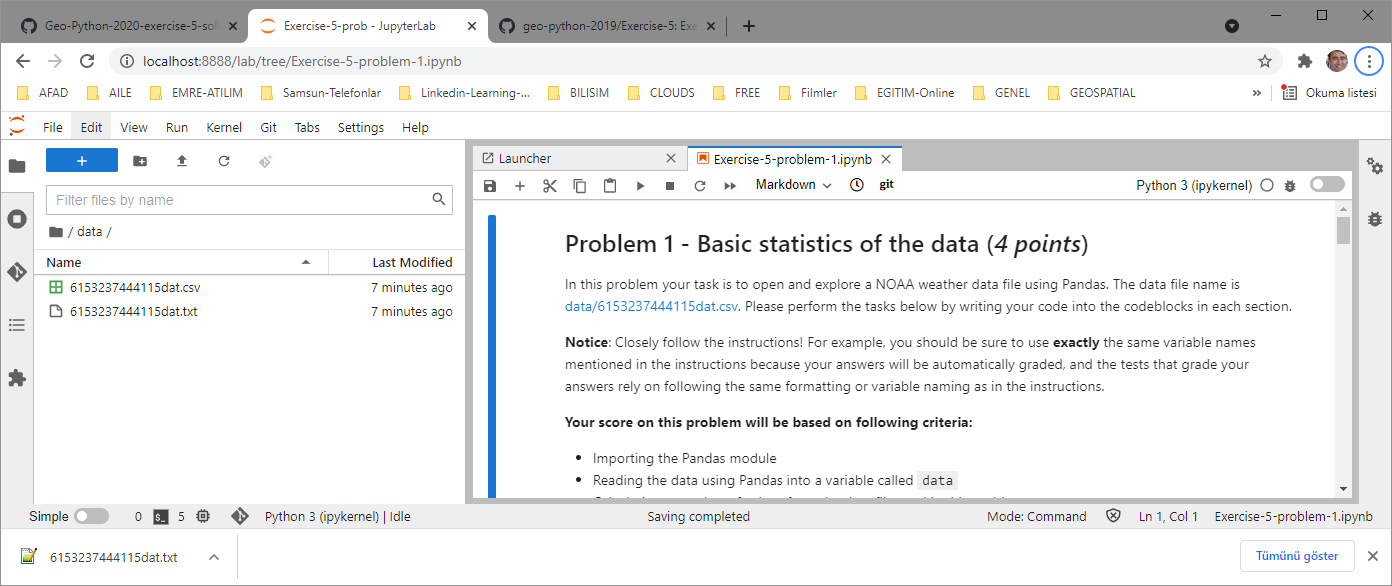
**Goto Download icon and right click and save as** into in the data directory in the local repo:

Rename the extention of the file from .txt to .csv

Goto GitHub desktop, Commit new & changed files in local repo and push to remote repo (github.com)

Verify that the new file is uploaded to the remote repo:

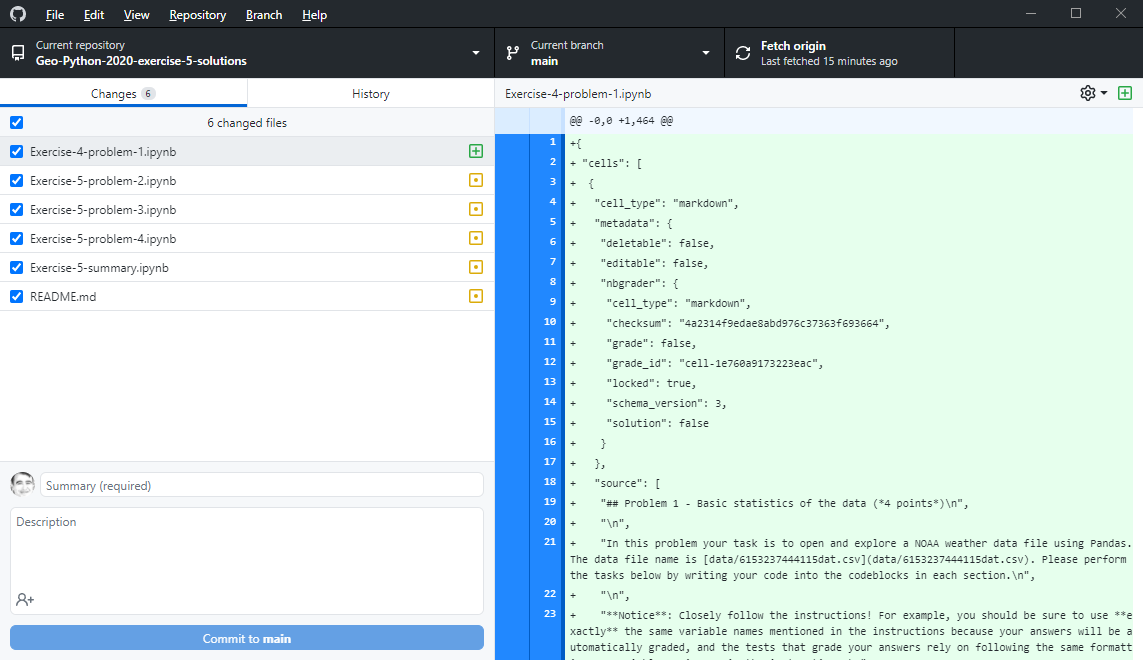


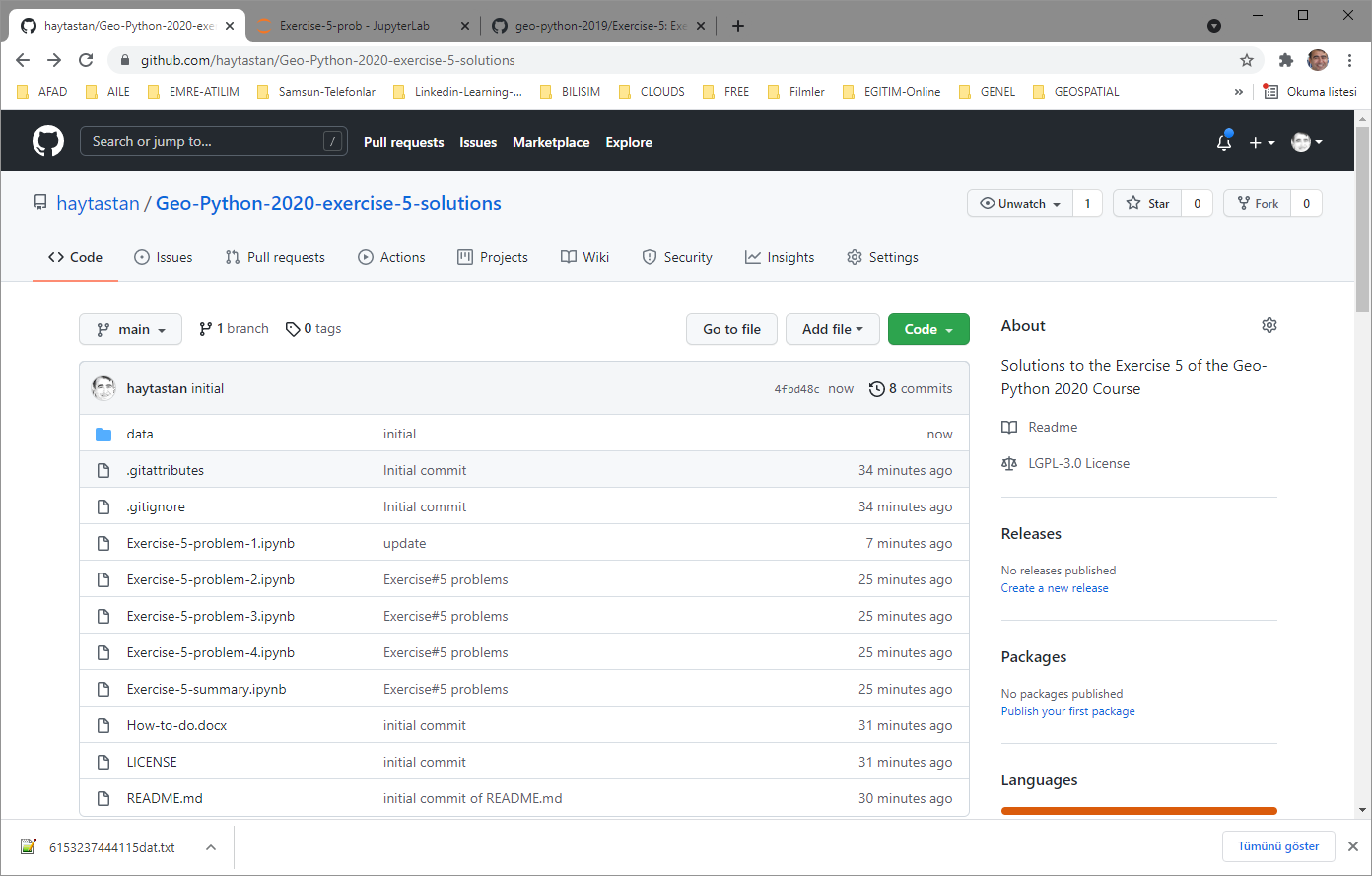
Files downloaded are:

**Goto GitHub Desktop (which is already running):**

**Write something into Summary and Description boxes as below:**

**Summary:** Exercise#5 problems

**Description:** Downloaded from <https://github.com/geo-python-2019/Exercise-5/find/master>



**Let’s start to solve the problems:**

**Run (OSGeo4W Shell if you installed QGIS 3.201) or Run cmd (if you didn’t install QGIS but installed Python 3.9.7 seperately) or Run the shortcut to the OSGeo4W Shell (**i.e. for my case python-qgis-cmd-shotcut which points to python-qgis-cmd.bat (i.e to C:\programs\qgis320\bin\python-qgis-cmd.bat).

The content of python-qgis-cmd.bat is as below:

@**echo** off

**call** "**%~dp0\o4w\_env.bat"**

@**echo** off

**path** **%OSGEO4W\_ROOT%**\apps\qgis\bin;**%PATH%**

**path** **%PATH%**;C:\programs\nodejs

**path** **%PATH%**;C:\programs\Git\cmd

**set** QGIS\_PREFIX\_PATH**=%OSGEO4W\_ROOT:\=/%**/apps/qgis

**set** GDAL\_FILENAME\_IS\_UTF8**=**YES

rem Set VSI cache to be used as buffer, see #6448

**set** VSI\_CACHE**=**TRUE

**set** VSI\_CACHE\_SIZE**=**1000000

**set** QT\_PLUGIN\_PATH**=%OSGEO4W\_ROOT%**\apps\qgis\qtplugins;**%OSGEO4W\_ROOT%**\apps\qt5\plugins

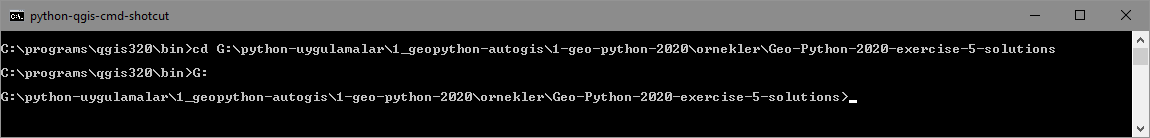
**set** PYTHONPATH**=%OSGEO4W\_ROOT%**\apps\qgis\python;**%PYTHONPATH%**

rem "%PYTHONHOME%\python" %\*

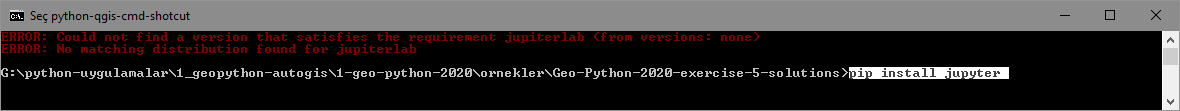
cmd

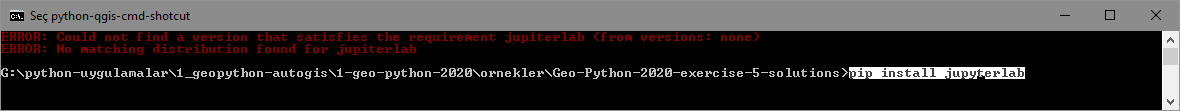
**Change directory to the local repo:**

cd G:\python-uygulamalar\1\_geopython-autogis\1-geo-python-2020\ornekler\Geo-Python-2020-exercise-5-solutions

G:

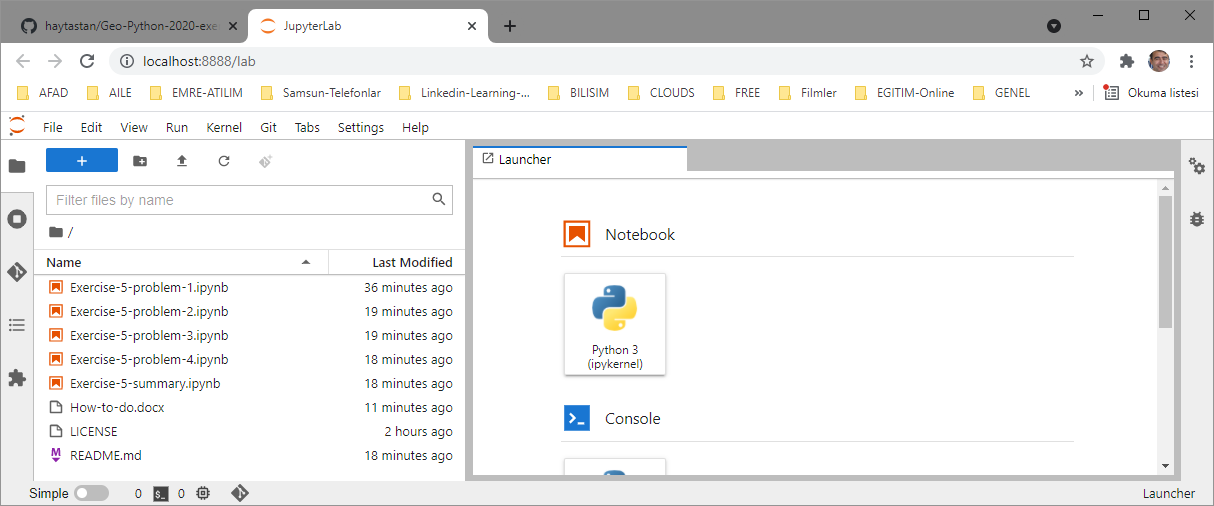
**Install Jupyter Lab (if not installed):**

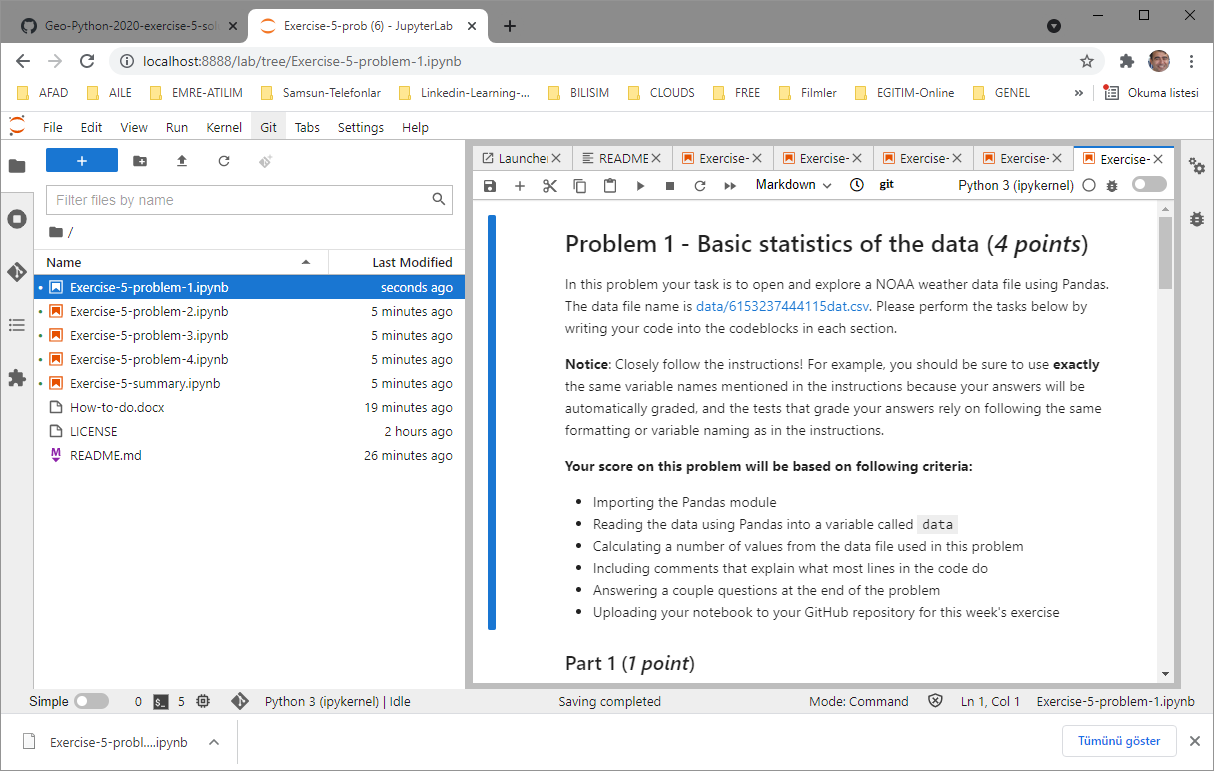
pip install jupiter

pip install jupyterlab

**Run Jupyter Lab:**

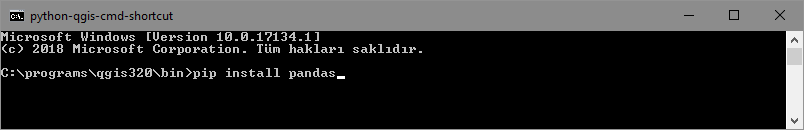
jupyter lab

**A tab for Jupyter lab is created at the browser:**

**Goto to the Jubyter Lab tab on the browser and double click one of the Jupiter projects (e.g. Exercise-5-problem-1.ipnyb) in the local repo:**

You may solve the problem step by step and **write your solution commands in the box** right after the question, then run the command in the box by **SHIFT+ENTER**:

**But first of all install the module pandas:**

**Run OSGeo4W Shell and pip install pandas:**

**Part 1:**

* Import the Pandas module (*0.5 points*)
* Read the data into a variable called data using Pandas (*0.5 points*)

**Important**: When reading the data, you need to tell Pandas that no-data values are specified with varying number of \* characters. You can do this by specifying a following parameter in the Pandas read\_csv() function:

na\_values=['\*', '\*\*', '\*\*\*', '\*\*\*\*', '\*\*\*\*\*', '\*\*\*\*\*\*']

**Commands:**

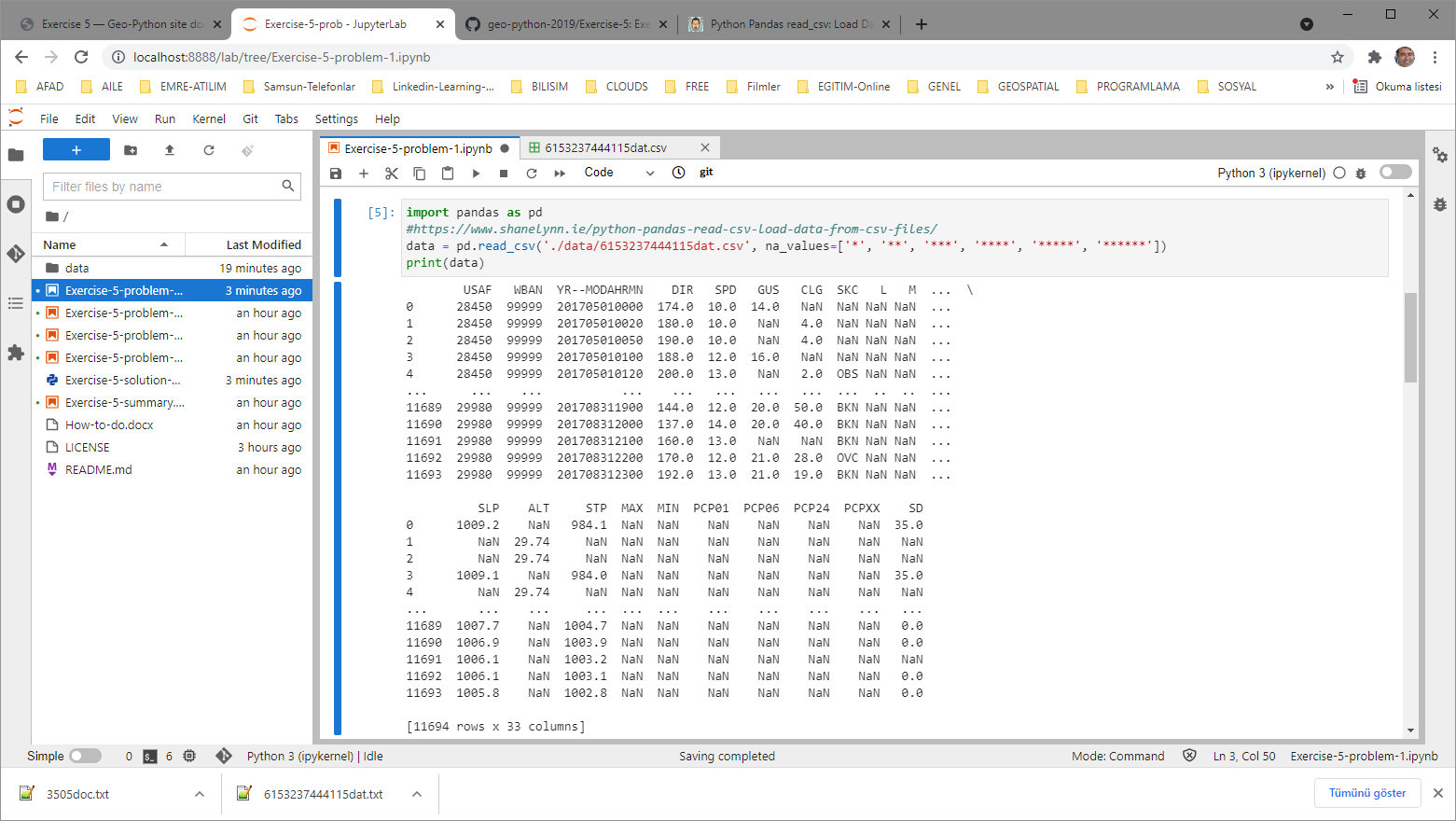
#Part 1

**import** pandas **as** pd

#https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files/

data **=** pd**.**read\_csv**(**'./data/6153237444115dat.csv'**,** na\_values**=[**'\*'**,** '\*\*'**,** '\*\*\*'**,** '\*\*\*\*'**,** '\*\*\*\*\*'**,** '\*\*\*\*\*\*'**])**

**print(**data**)**

**Output:**

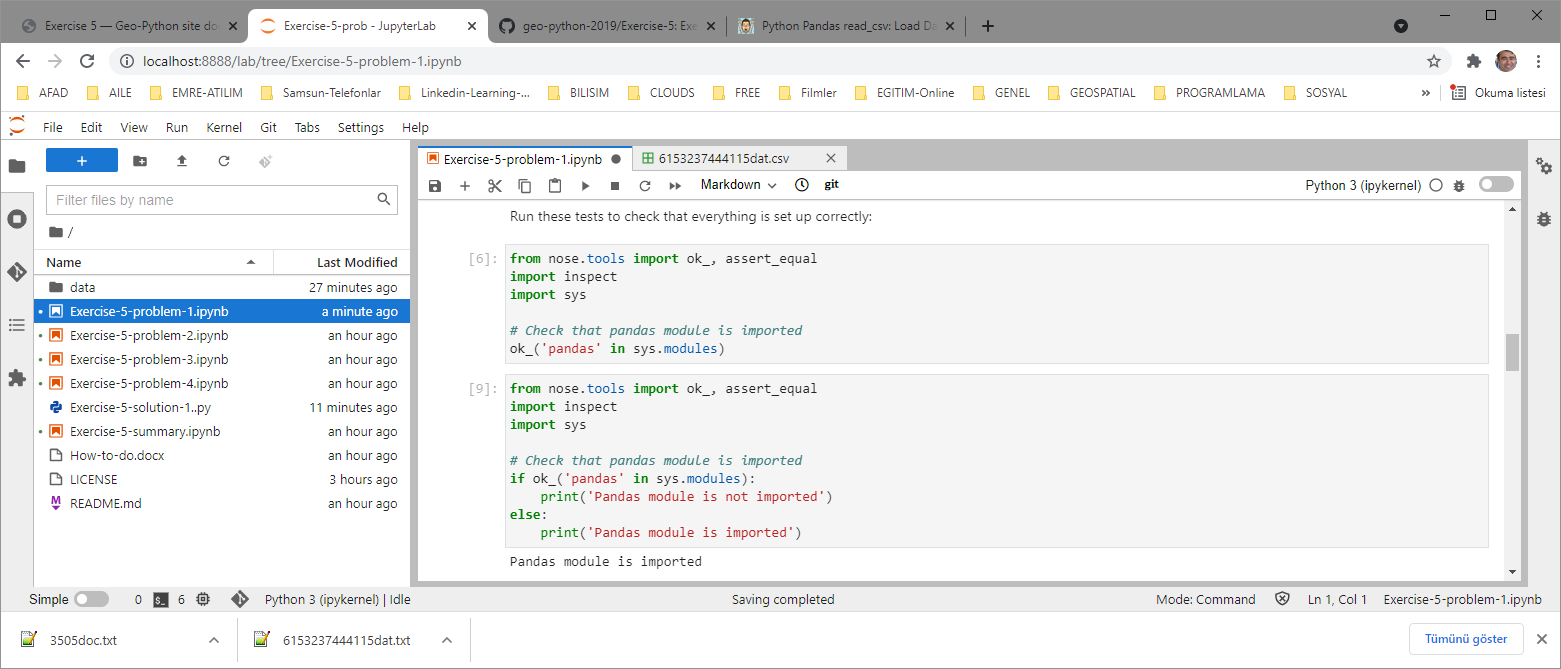
Add a new command box via plus icon in the menu and fill in the box with the code below:

# Check that pandas module is imported

**if** ok\_**(**'pandas' **in** sys**.**modules**):**

**print(**'Pandas module is not imported'**)**

**else:**

 **print(**'Pandas module is imported'**)**

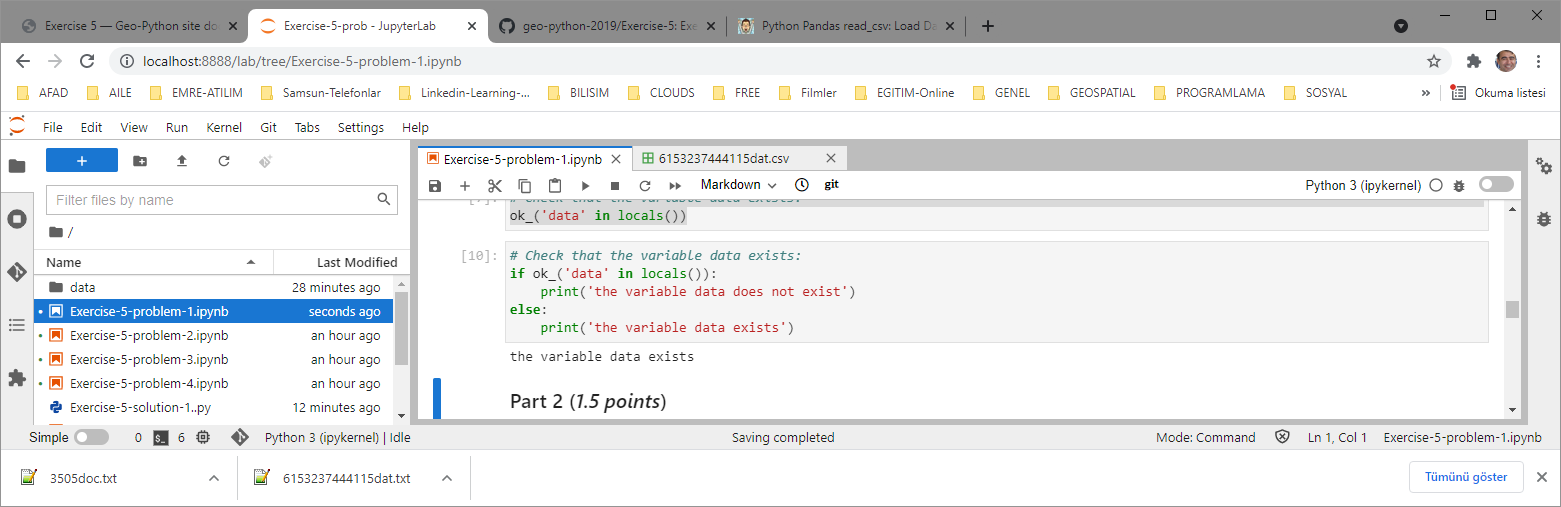
Add a new command box via plus icon in the menu and fill in the box with the code below:

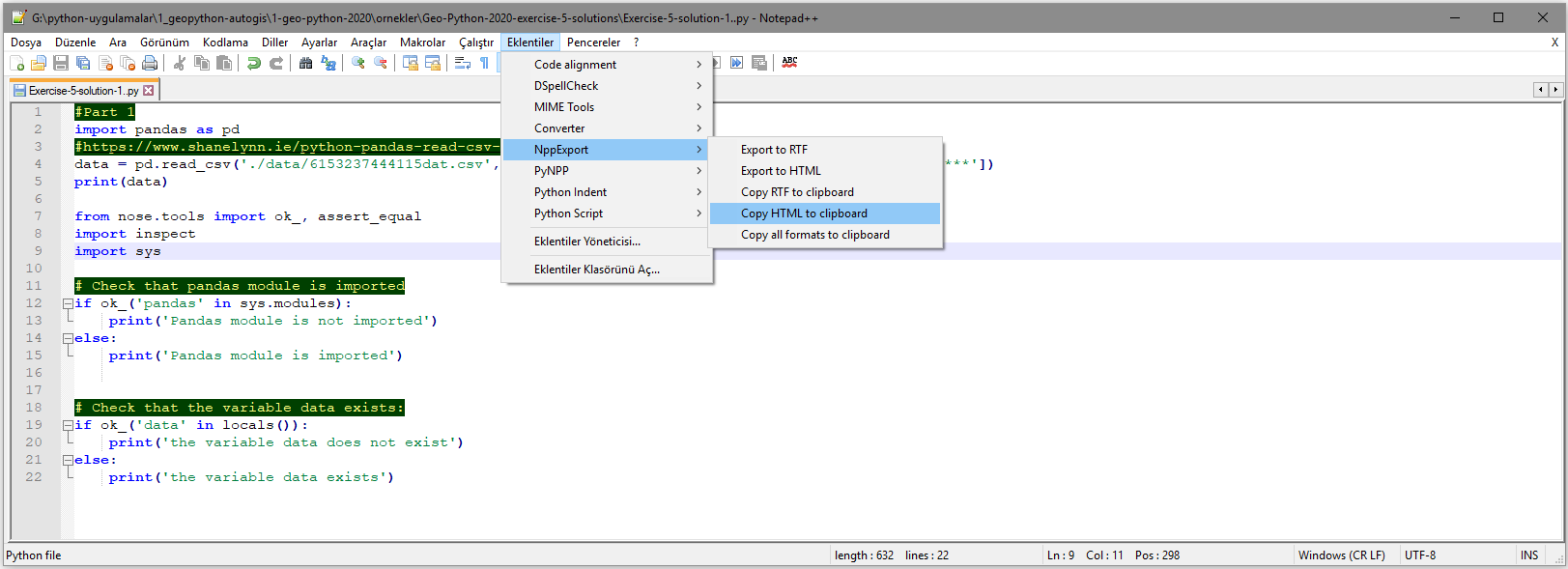
# Check that the variable data exists:

**if** ok\_**(**'data' **in** locals**()):**

**print(**'the variable data does not exist'**)**

**else:**

 **print(**'the variable data exists'**)**

**P.A.:** Using the NppExport extention of Notepad++ you may copy codes in color into clipboard (Copy HTML to Clipboard) and then paste into the word document as I did above:

**Whole code of Exercise-5-solution-1-part-1.py:**

#Part 1:

**import** pandas **as** pd

#https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files/

data **=** pd**.**read\_csv**(**'./data/6153237444115dat.csv'**,** na\_values**=[**'\*'**,** '\*\*'**,** '\*\*\*'**,** '\*\*\*\*'**,** '\*\*\*\*\*'**,** '\*\*\*\*\*\*'**])**

**print(**data**)**

**from** nose**.**tools **import** ok\_**,** assert\_equal

**import** inspect

**import** sys

# Check that pandas module is imported

**if** ok\_**(**'pandas' **in** sys**.**modules**):**

**print(**'Pandas module is not imported'**)**

**else:**

**print(**'Pandas module is imported'**)**

# Check that the variable data exists:

**if** ok\_**(**'data' **in** locals**()):**

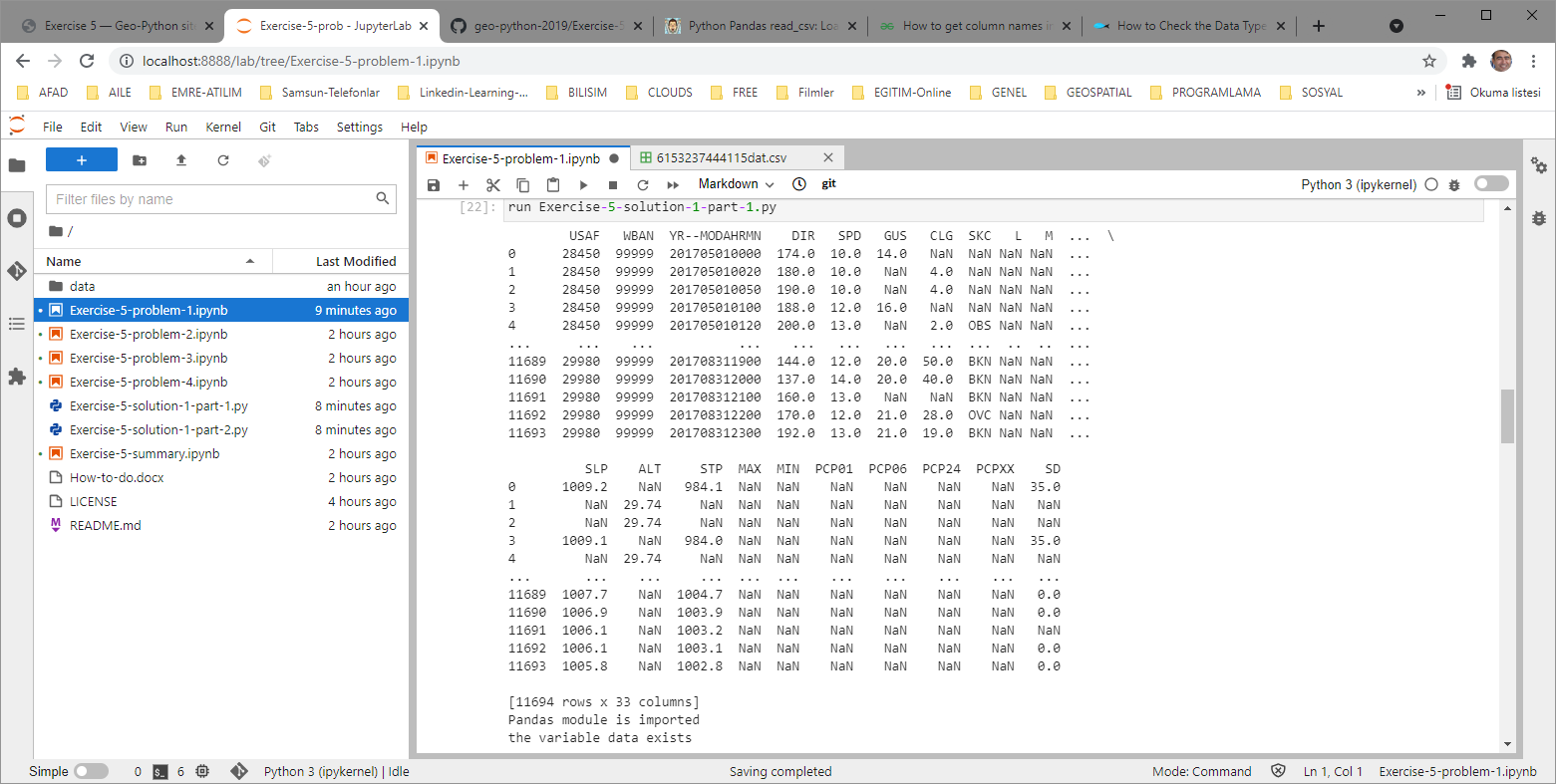
**print(**'the variable data does not exist'**)**

**else:**

**print(**'the variable data exists'**)**

**Running the whole code at once:**

Add a new command box and write the following code in the new box

**run Exercise-5-solution-1-part-1.py**

**Part 2:**

* How many rows is there in the data (variable rows)?
* What are the column names (variable column\_names)?
* What are the datatypes of the columns (variable column\_datatypes)?

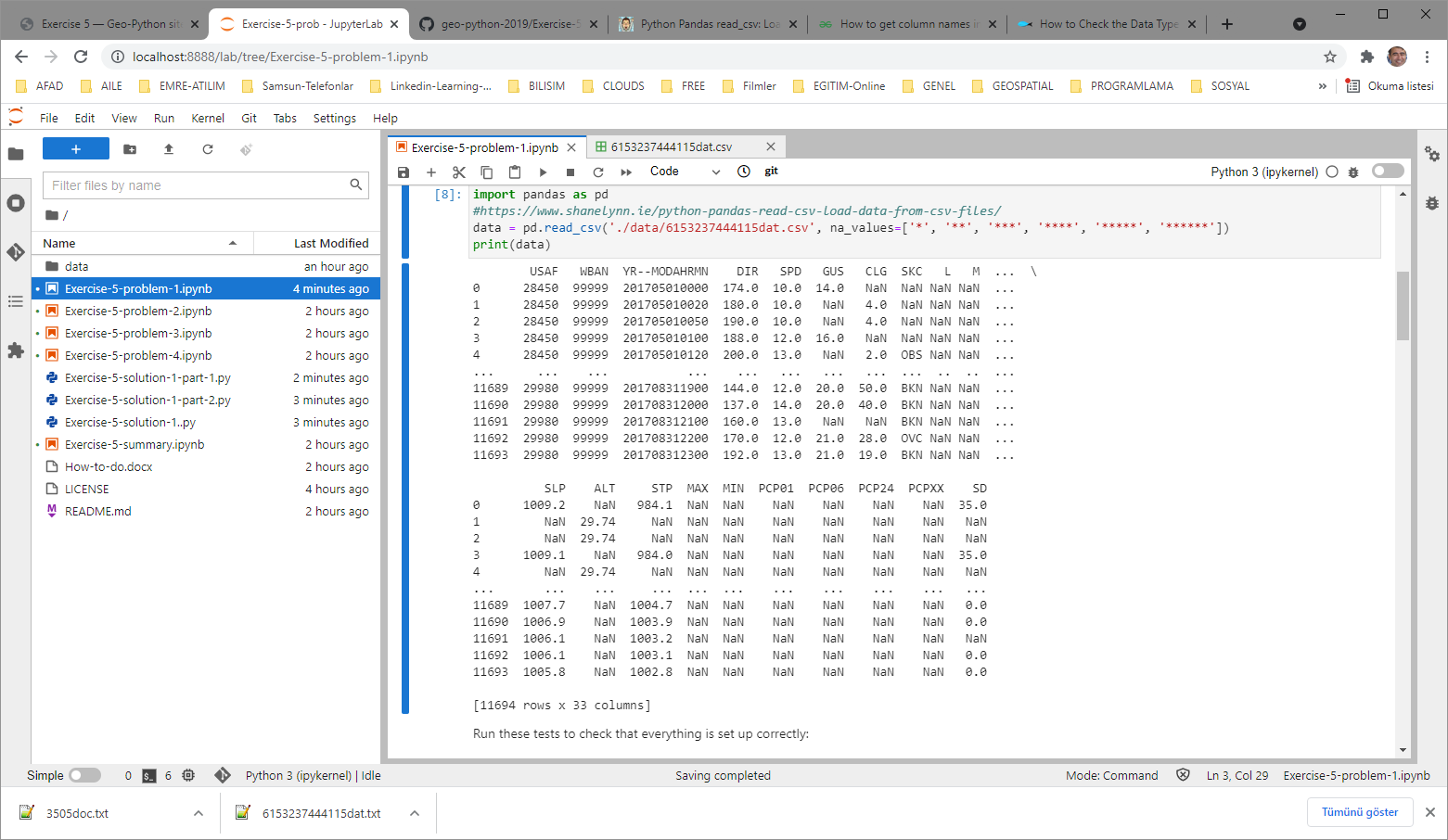
**Commands:**

#Part 2:

**import** pandas **as** pd

#https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files/

data **=** pd**.**read\_csv**(**'./data/6153237444115dat.csv'**,** na\_values**=[**'\*'**,** '\*\*'**,** '\*\*\*'**,** '\*\*\*\*'**,** '\*\*\*\*\*'**,** '\*\*\*\*\*\*'**])**

**print(**data**)**

**#** Run these tests to check that everything is set up correctly:

**from** nose**.**tools **import** ok\_**,** assert\_equal

**import** inspect

**import** sys

# Check that pandas module is imported

**if** ok\_**(**'pandas' **in** sys**.**modules**):**

**print(**'Pandas module is not imported'**)**

**else:**

**print(**'Pandas module is imported'**)**

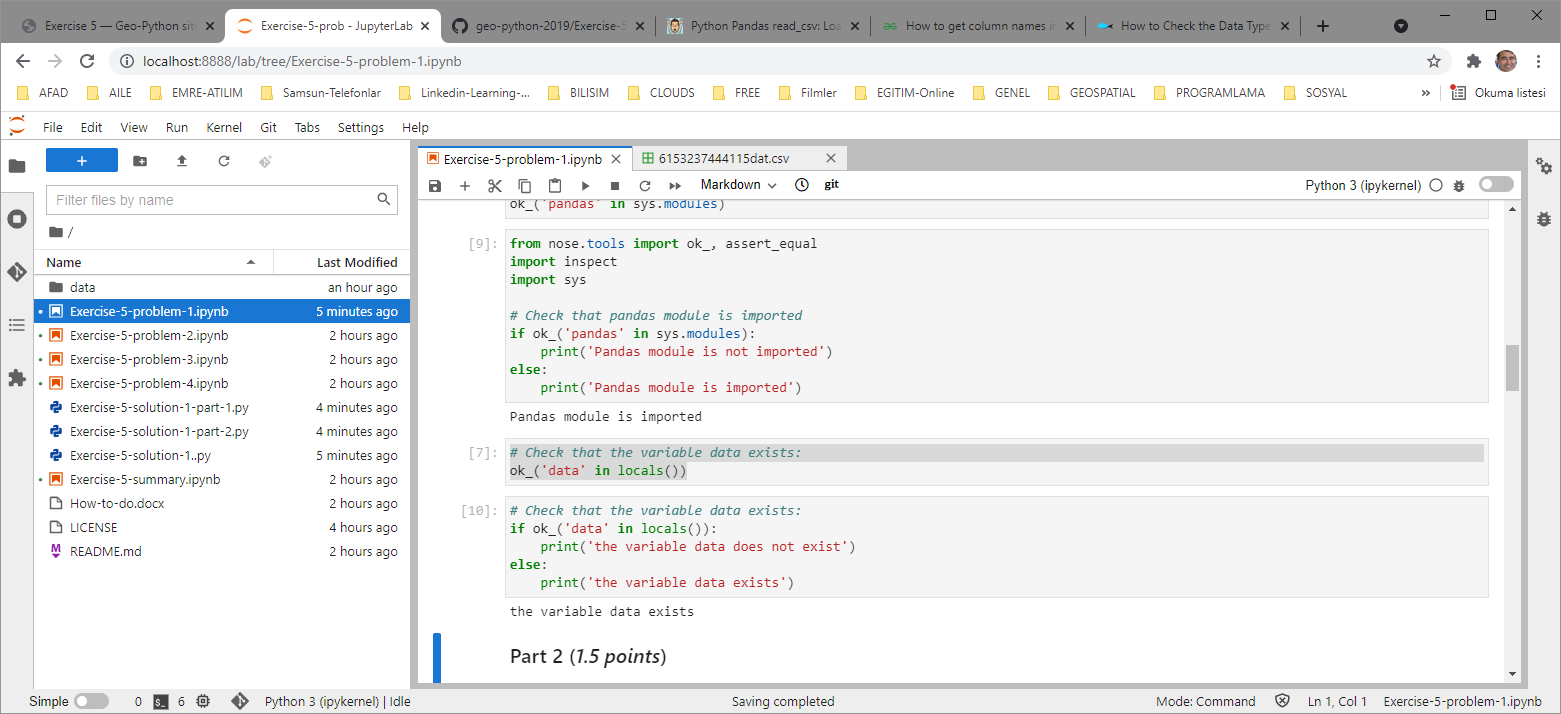
# Check that the variable data exists:

**if** ok\_**(**'data' **in** locals**()):**

**print(**'the variable data does not exist'**)**

**else:**

**print(**'the variable data exists'**)**



**Whole code of Exercise-5-solution-1-part-2.py:**

#Part 2:

**import** pandas **as** pd

#https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files/

data **=** pd**.**read\_csv**(**'./data/6153237444115dat.csv'**,** na\_values**=[**'\*'**,** '\*\*'**,** '\*\*\*'**,** '\*\*\*\*'**,** '\*\*\*\*\*'**,** '\*\*\*\*\*\*'**])**

**print(**data**)**

**from** nose**.**tools **import** ok\_**,** assert\_equal

**import** inspect

**import** sys

# Number of rows in the dataframe:

df **=** pd**.**DataFrame**(**data**)**

rows **=** df**.**shape**[0]** # Gives number of rows

cols **=** df**.**shape**[1]** # Gives number of columns

**print(**df**)**

# Column names:

# https://www.geeksforgeeks.org/how-to-get-column-names-in-pandas-dataframe/

# https://www.geeksforgeeks.org/python-pandas-dataframe/

column\_names **=** **[]**

**for** col **in** df**.**columns**:**

column\_names**.**append**(**col**)**

# Column datatypes:

# https://datatofish.com/data-type-pandas-dataframe/

column\_datatypes **=** **[]**

**for** col **in** df**.**columns**:**

column\_datatypes**.**append**(**df**[**col**].**dtypes**)**

# Print the number of rows in the dataframe:

**print(**'There are ' **+** str**(**rows**)** **+** ' rows'**)**

# Print the column names:

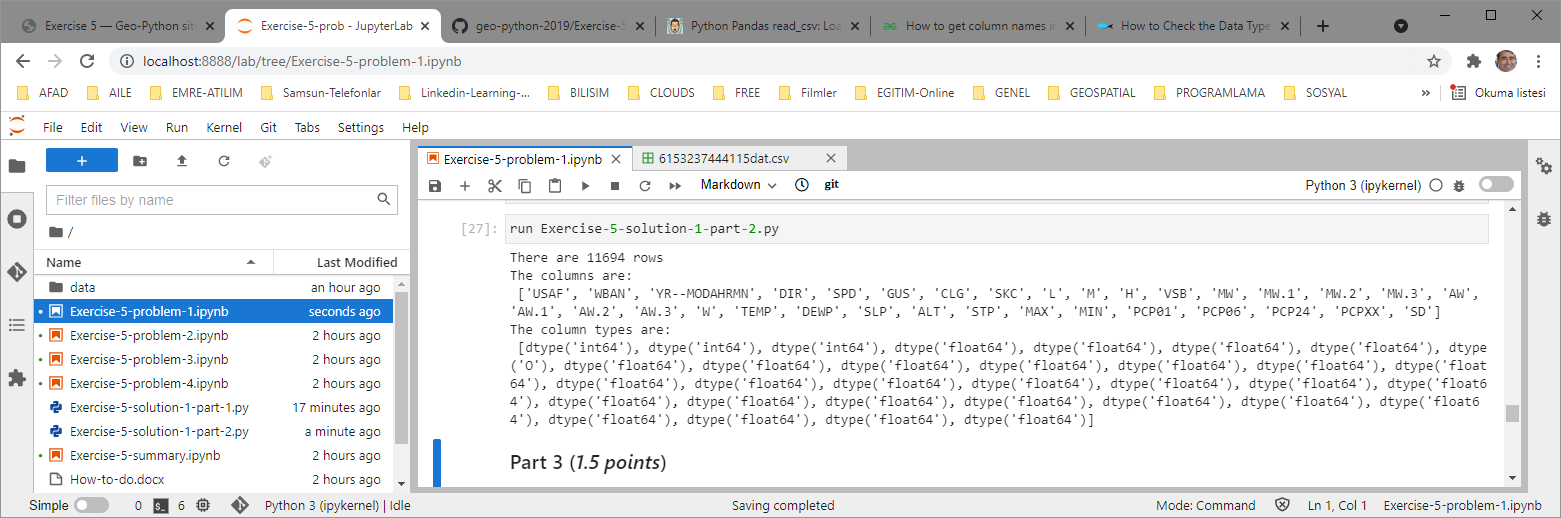
**print(**'The columns are: \n'**,** column\_names**)**

# Print the column datatypes:

**print(**'The column types are: \n'**,** column\_datatypes**)**

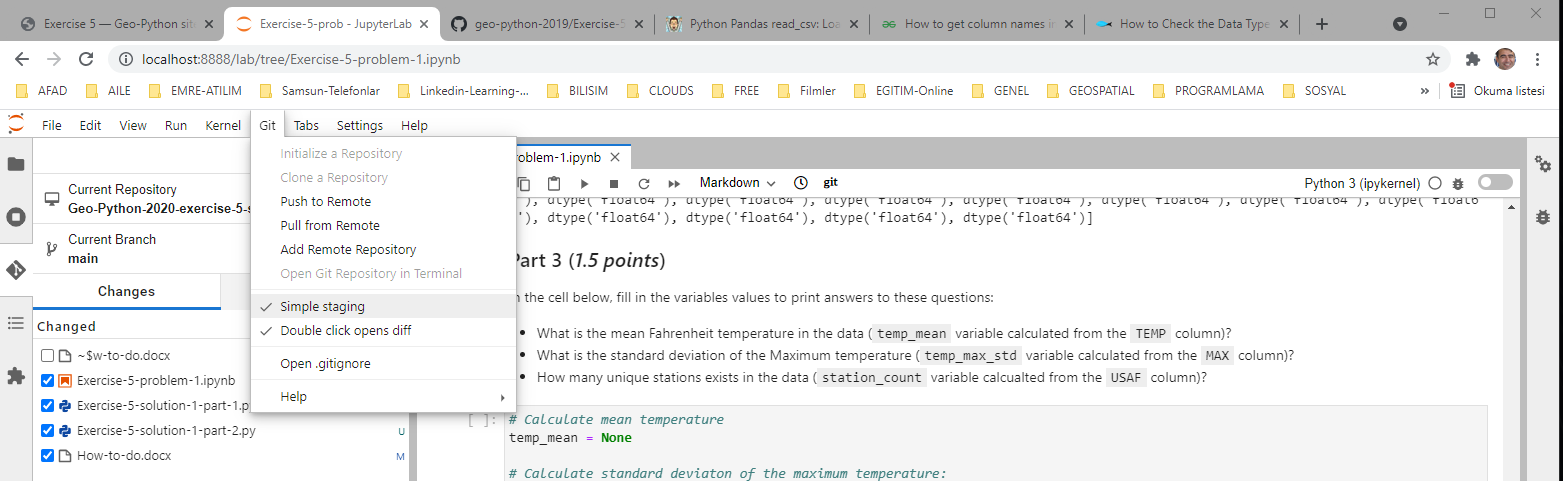
**Running the whole code at once:**

Add a new command box and write the following code in the new box

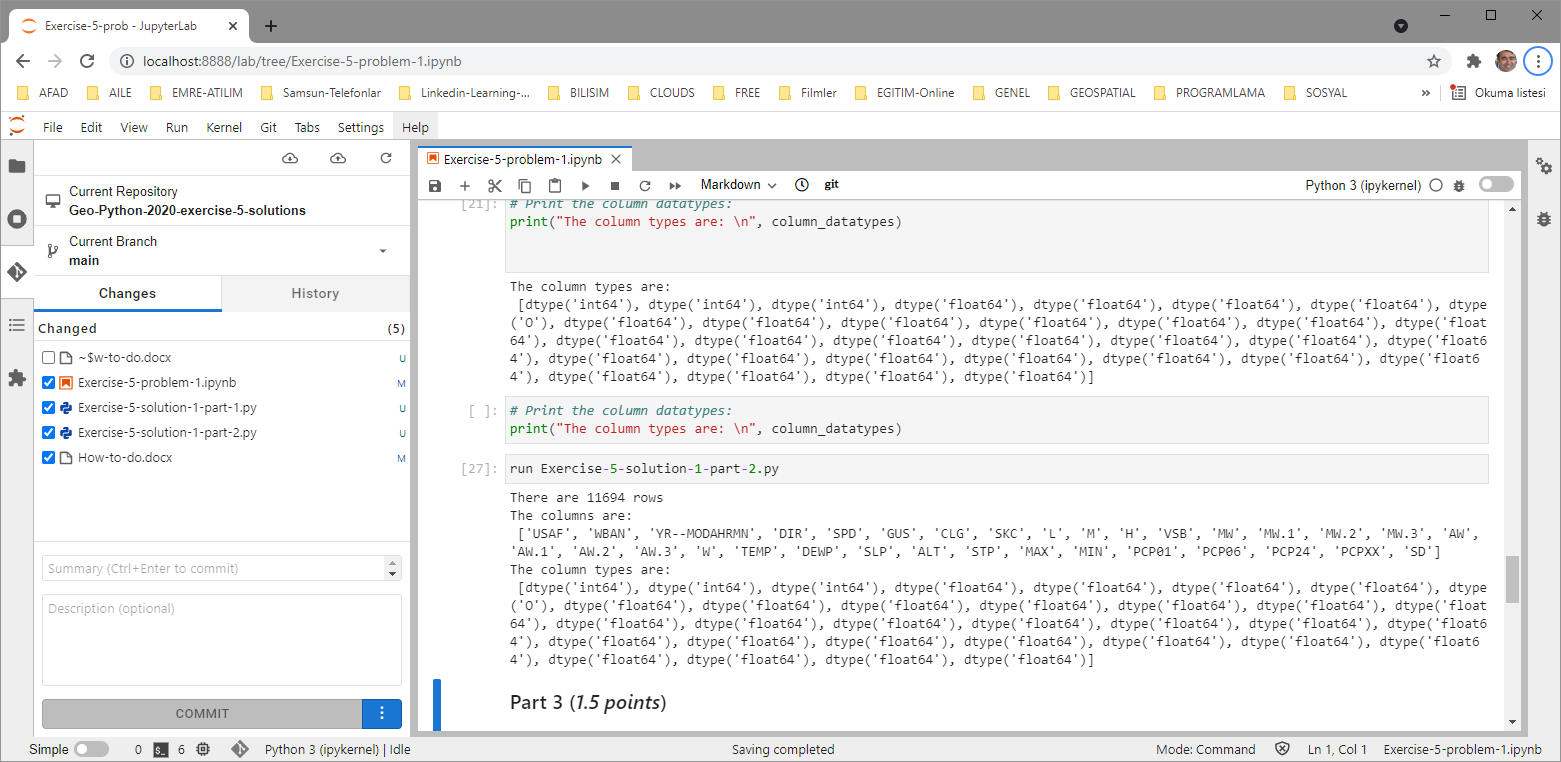
**run Exercise-5-solution-1-part-2.py**

**Updating remote repo (GitHub):**

You may update repote repo using GitHub Desktop as explained before or you may send changes to the Github (remote desktop) in the Jupyter Lab environment as explained below:

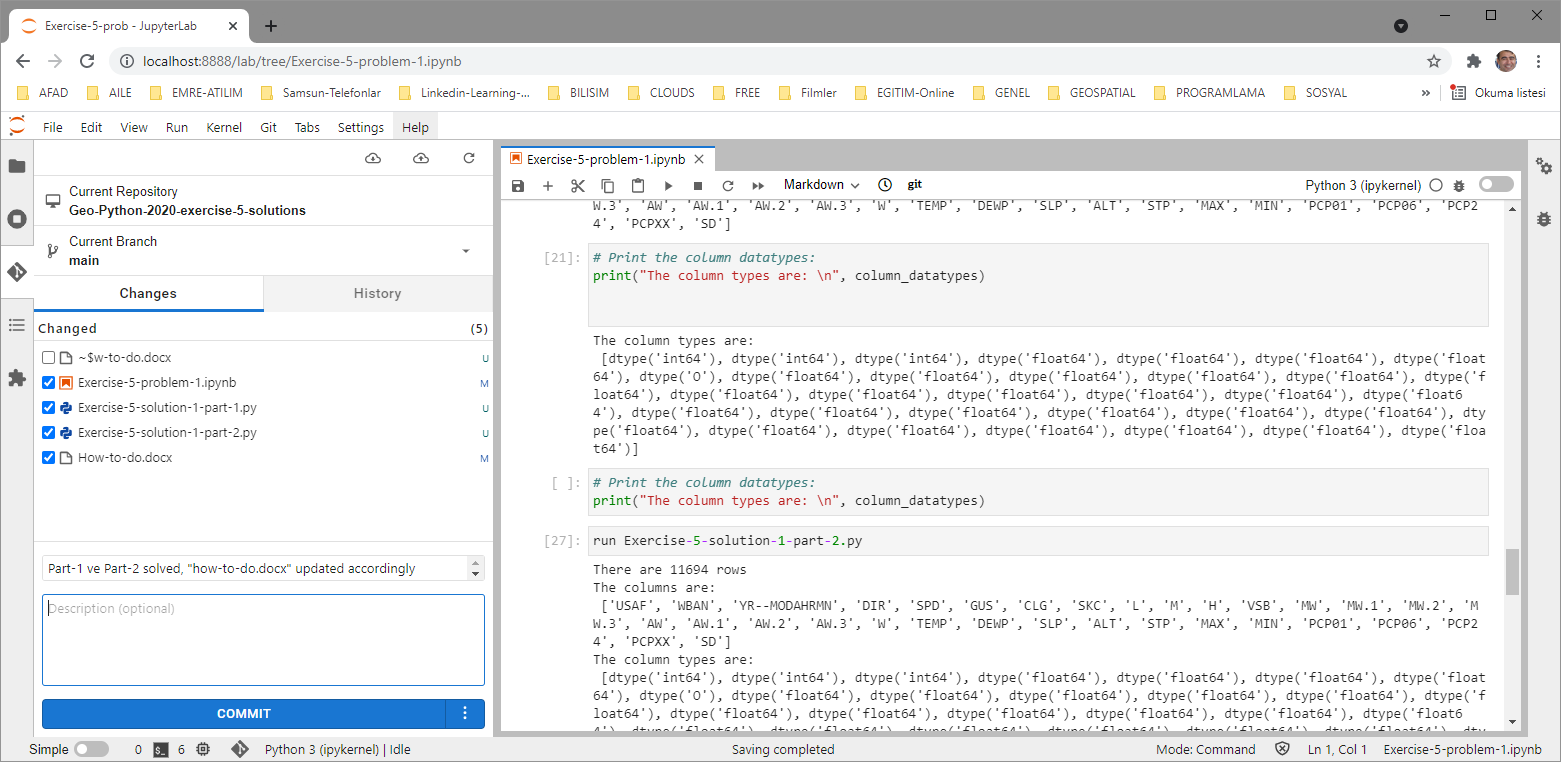
Select “Git/Simple Staging” from the menu:

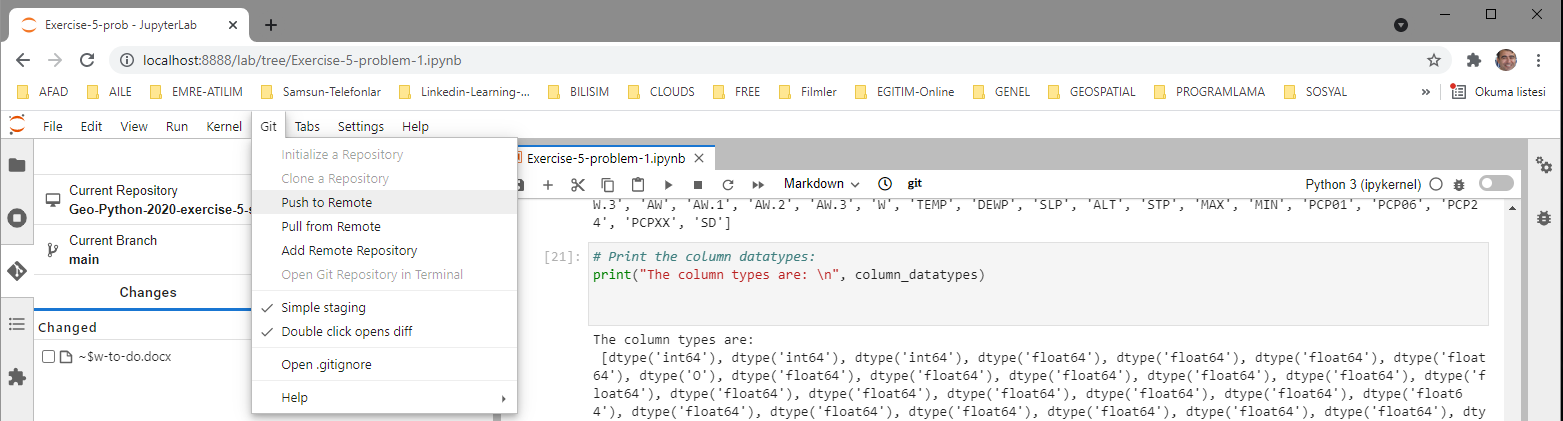
Select the Git icon on the left bar and check all changed files:



Write Summary and descriptions for the commit:

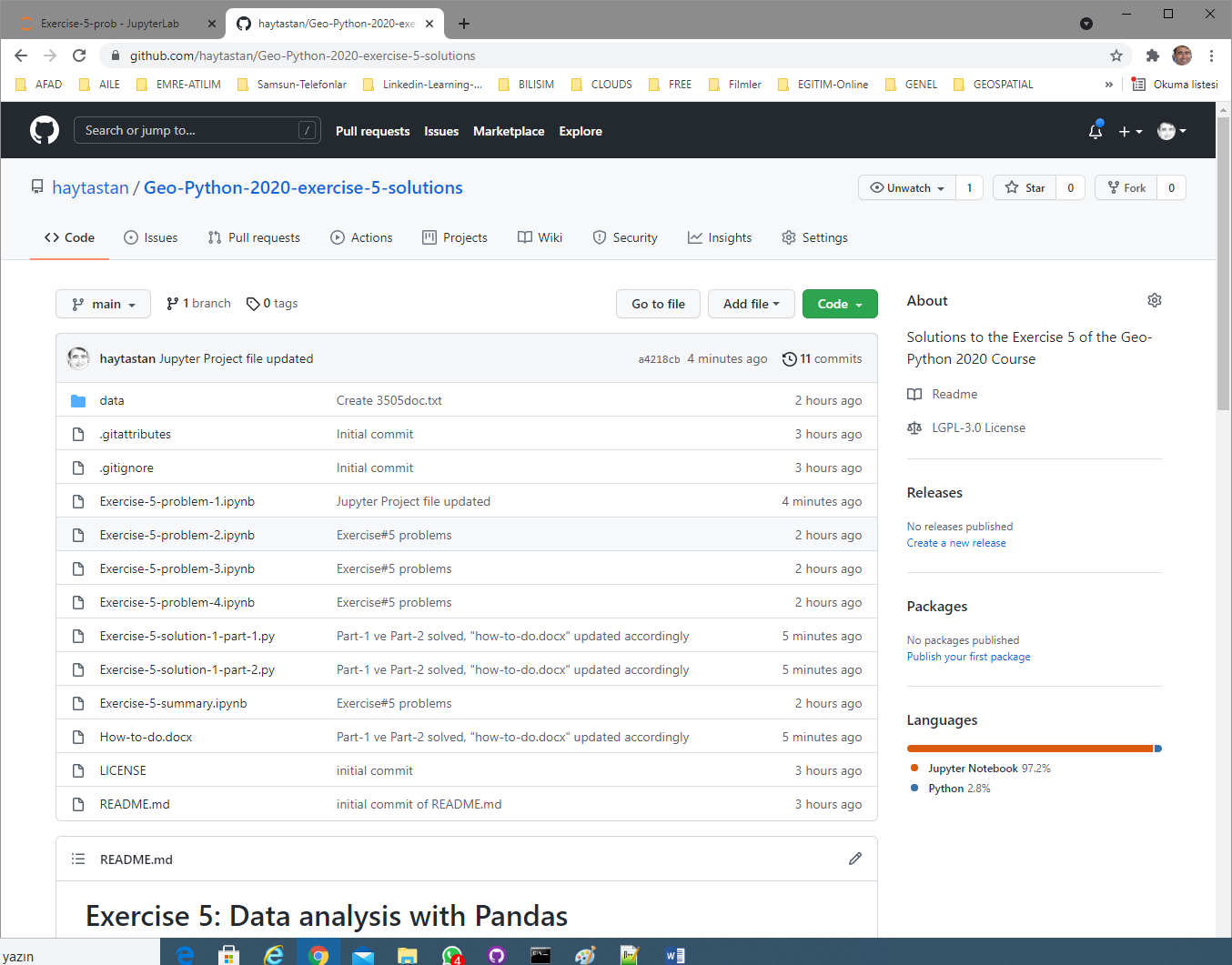
**Summary**: Part-1 ve Part-2 solved, "how-to-do.docx" updated accordingly

**Description:**

COMMIT

Git/Push to remote:

**P.A:** Be sure that “Successfully pushed” message is displayed at the right-buttom corner of the browser.

**Verify that the new files are uploaded to the remote repo:**