ENGR 133 INDIVIDUAL PROJECT REPORT

LC1-15 Yolanda Chen, Nathan Budiman, Rachel Evrard, Collin Gernhardt

Chen, Tzu-Yu

Introduction:

This is a python program that would output the number of deaths, recovered, and confirmed cases globally from the coronavirus when the user inputs a date and country. I chose this topic because as an international student, my family and I have been closely monitoring these statistics to see if it's safe to go travel (US, specifically, since I'm a Purdue Student). I believe that coronavirus is a hot topic that everyone's interested in and it would be helpful if people could just search for the date/country they're curious in and have the answers, instead of searching through a list of dates/countries.

Overall overview of program, function descriptions, description of inputs and outputs:

There is a main program called *main.py* and two separate scripts called *date.py* and *country.py* for two user defined functions. When you run it, you run the main program. The program prompts user to input a country and date. Then it will output the number of cases. You repeated this stage for another two times, and it will output the number of deaths and recovered corresponding to the date and country you entered. The user can input different date and different country each time the programs asks.

Description of all user defined functions:

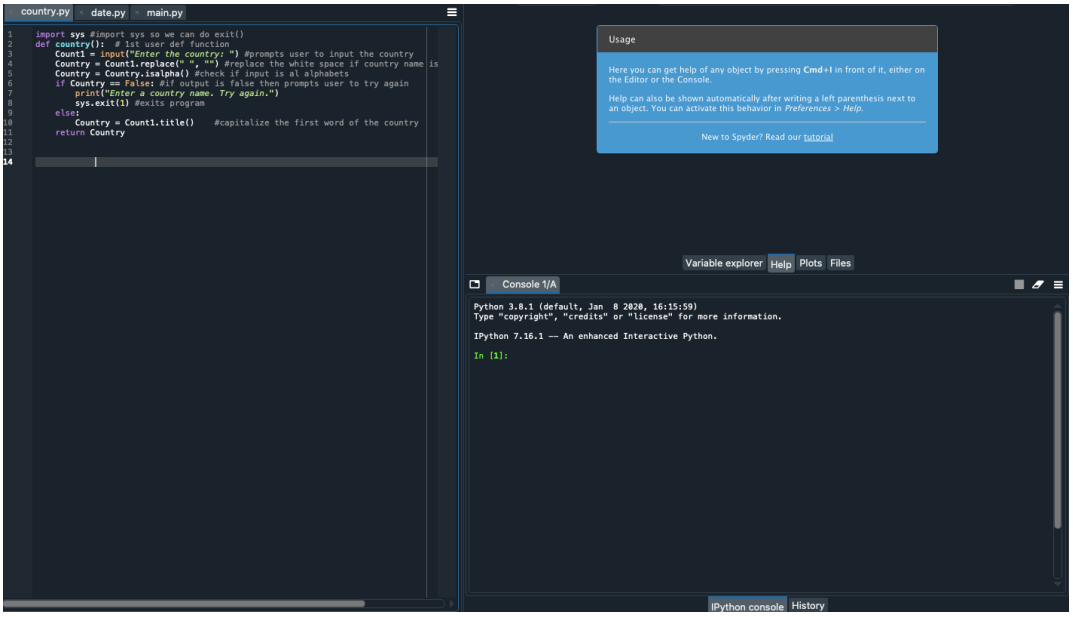
The first user defined function is in a file called *date.py*. The function date() imports datetime and turns the user input numbers into the correct format so the main program can scan it in the data file and find the corresponding column of data. This program also does error checking to make sure the user did not input the numbers in another format or enter any non-integer. It exits the system if error is found and asks the user to try again. The next user defined function is in a file called *country.py*. The function country() makes sure that the user inputs alphabetical characters and no integers or other characters. It then capitalizes the first letter of each word if there is no error. It capitalizes the first letter because there are countries that have two words. The third user defined function is in the main program called *main.py*. The function findrow() finds the row in the data with the header that corresponds with the country the user inputted and then returns it.

User manual:

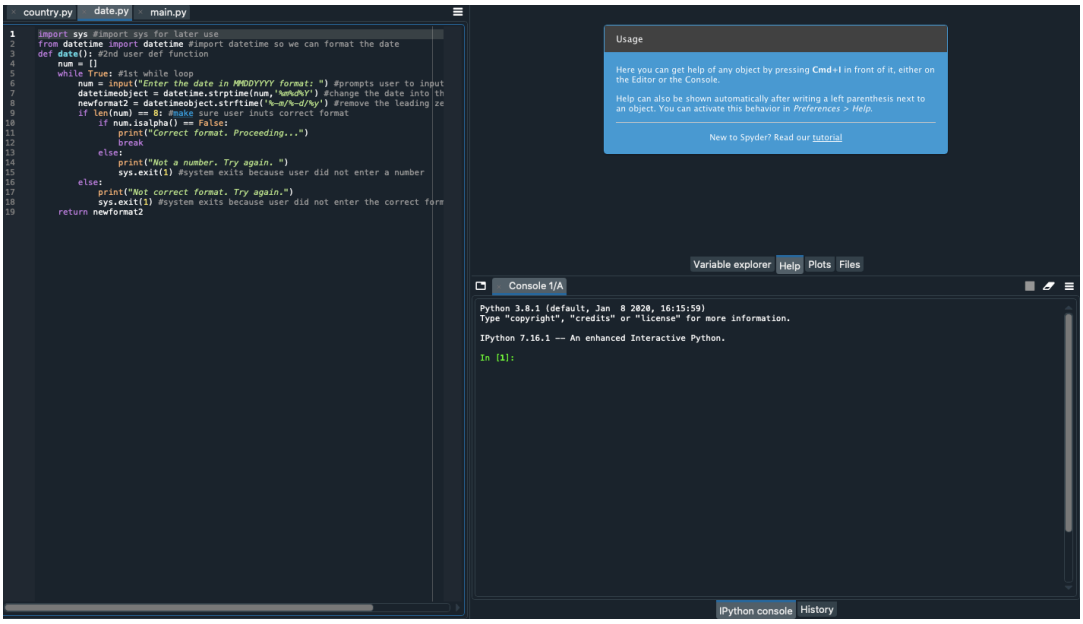
For the whole program to run, you will have these three files opened: *date.py*, *country.py*, *main.py*

Sample input: Australia and 09202020

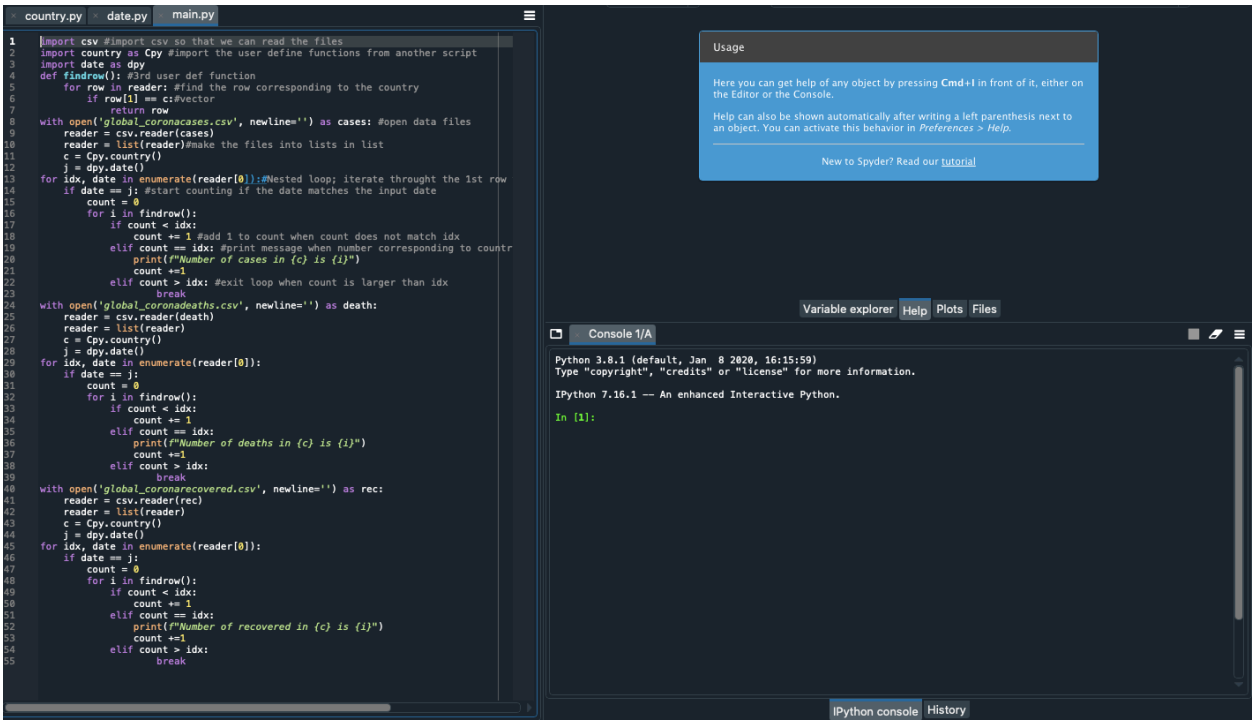
Country.py



Date.py



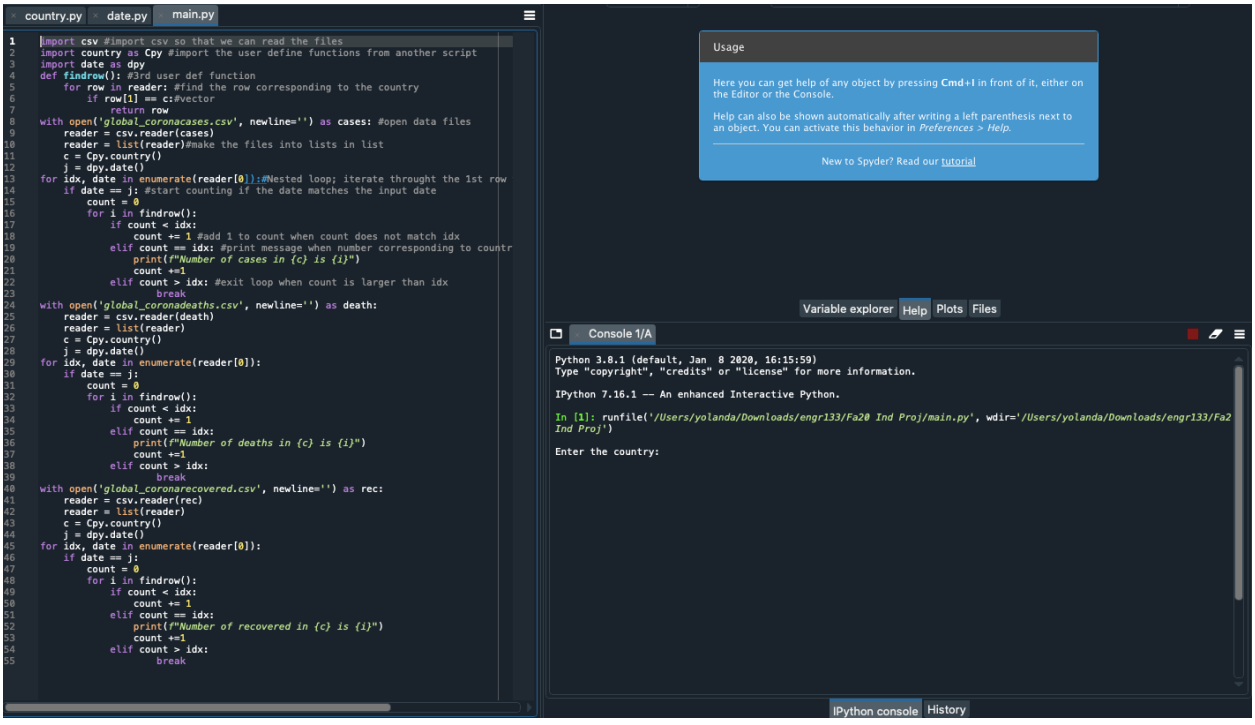
Main.py



```
import csv #import csv so that we can read the files
import country as Cpy #import the user define functions from another script
import date as dpy
def findrow(): #3rd user def function
    for row in reader: #find the row corresponding to the country
        if row[1] == c:#vector
            return row
with open('global_coronacases.csv', newline='') as cases: #open data files
    reader = csv.reader(cases)
    reader = list(reader)#make the files into lists in list
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):#Nested loop; iterate throught the 1st row
    if date == j: #start counting if the date matches the input date
        count = 0
        for i in findrow():
            if count < idx:
                count += 1 #add 1 to count when count does not match idx
            elif count == idx: #print message when number corresponding to countr
                print(f"Number of cases in {c} is {i}")
                count +=1
            elif count > idx: #exit loop when count is larger than idx
                    break
with open('global_coronadeaths.csv', newline='') as death:
    reader = csv.reader(death)
    reader = list(reader)
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):
    if date == j:
        count = 0
        for i in findrow():
            if count < idx:
                count += 1
            elif count == idx:
                print(f"Number of deaths in {c} is {i}")
                count +=1
            elif count > idx:
                    break
with open('global_coronarecovered.csv', newline='') as rec:
    reader = csv.reader(rec)
    reader = list(reader)
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):
    if date == j:
        count = 0
        for i in findrow():
            if count < idx:
                count += 1
            elif count == idx:
                print(f"Number of recovered in {c} is {i}")
                count +=1
            elif count > idx:
                    break
```

```
Usage

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

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

New to Spyder? Read our tutorial

Variable explorer  Help  Plots  Files

Console 1/A

Python 3.8.1 (default, Jan  8 2020, 16:15:59)
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]:

IPython console  History
```

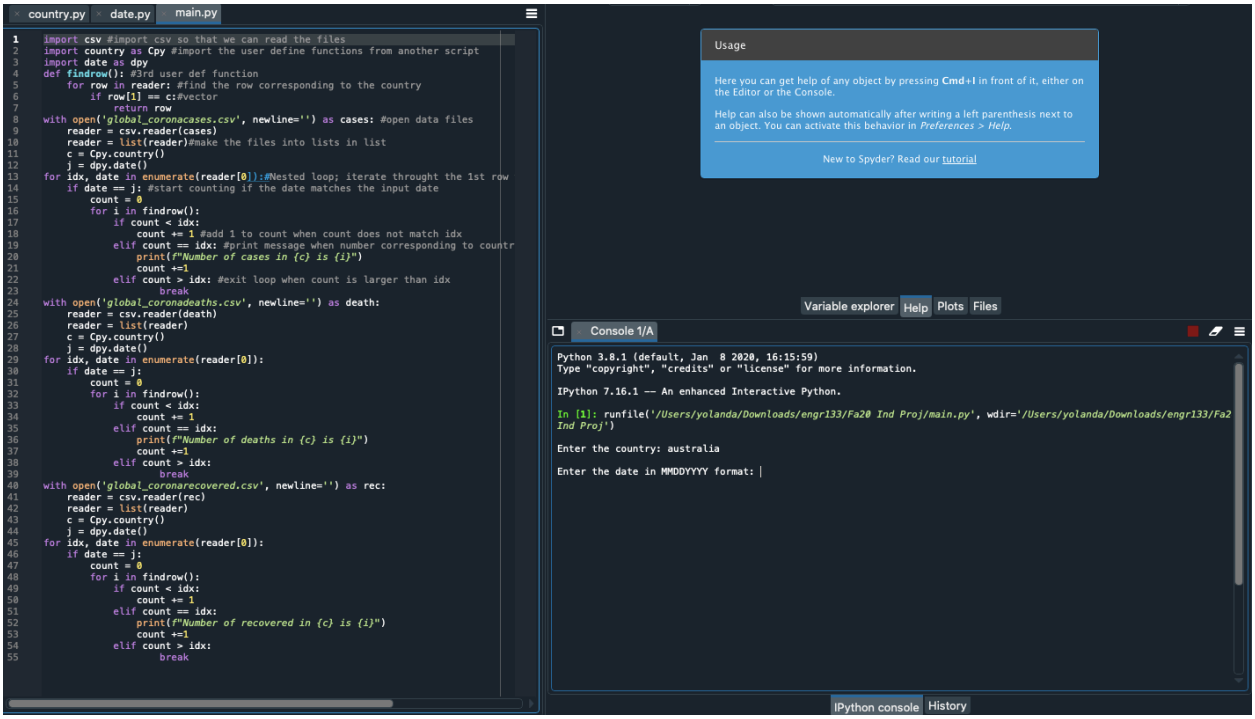
When you first run the main program, it will prompt the user to enter the country. See below.



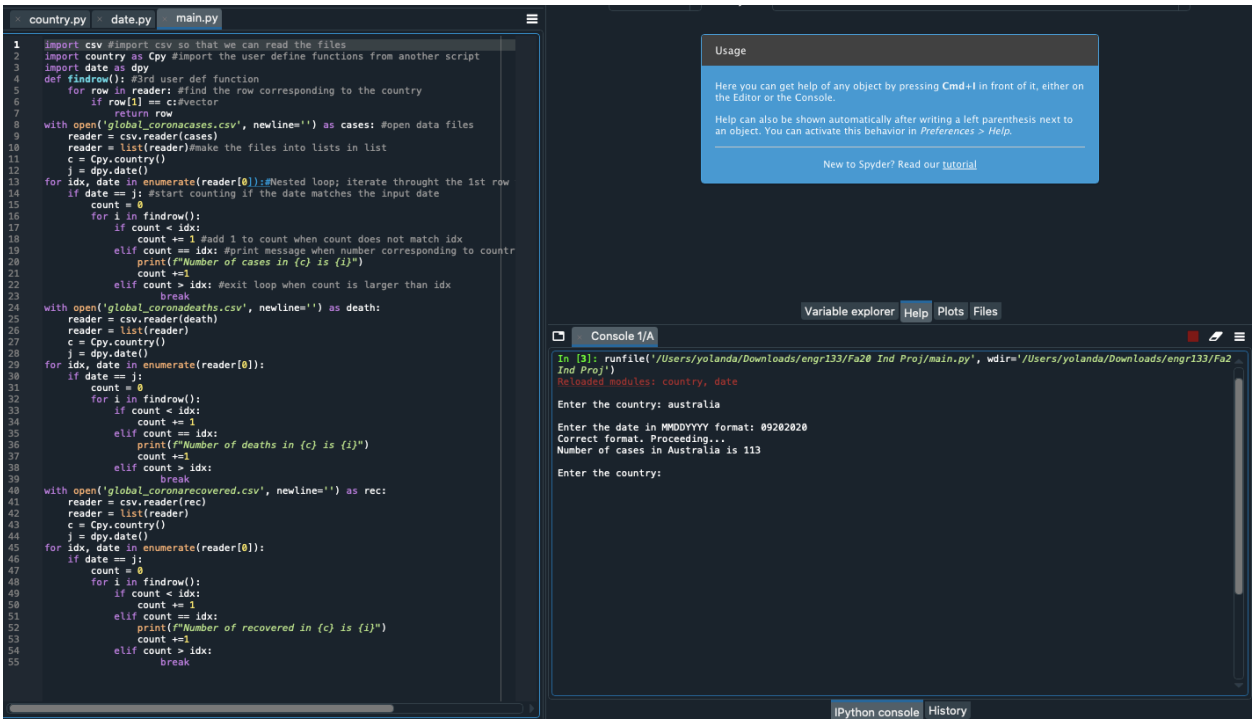
```
import csv #import csv so that we can read the files
import country as Cpy #import the user define functions from another script
import date as dpy
def findrow(): #3rd user def function
    for row in reader: #find the row corresponding to the country
        if row[1] == c:#vector
            return row
with open('global_coronacases.csv', newline='') as cases: #open data files
    reader = csv.reader(cases)
    reader = list(reader)#make the files into lists in list
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):#Nested loop; iterate throught the 1st row
    if date == j: #start counting if the date matches the input date
        count = 0
        for i in findrow():
            if count < idx:
                count += 1 #add 1 to count when count does not match idx
            elif count == idx: #print message when number corresponding to countr
                print(f"Number of cases in {c} is {i}")
                count +=1
            elif count > idx: #exit loop when count is larger than idx
                    break
with open('global_coronadeaths.csv', newline='') as death:
    reader = csv.reader(death)
    reader = list(reader)
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):
    if date == j:
        count = 0
        for i in findrow():
            if count < idx:
                count += 1
            elif count == idx:
                print(f"Number of deaths in {c} is {i}")
                count +=1
            elif count > idx:
                    break
with open('global_coronarecovered.csv', newline='') as rec:
    reader = csv.reader(rec)
    reader = list(reader)
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):
    if date == j:
        count = 0
        for i in findrow():
            if count < idx:
                count += 1
            elif count == idx:
                print(f"Number of recovered in {c} is {i}")
                count +=1
            elif count > idx:
                    break
```

```
Usage

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

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

New to Spyder? Read our tutorial

Variable explorer  Help  Plots  Files

Console 1/A

Python 3.8.1 (default, Jan  8 2020, 16:15:59)
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: runfile('/Users/yolanda/Downloads/engr133/Fa20 Ind Proj/main.py', wdir='/Users/yolanda/Downloads/engr133/Fa2
Ind Proj')

Enter the country:

IPython console  History
```

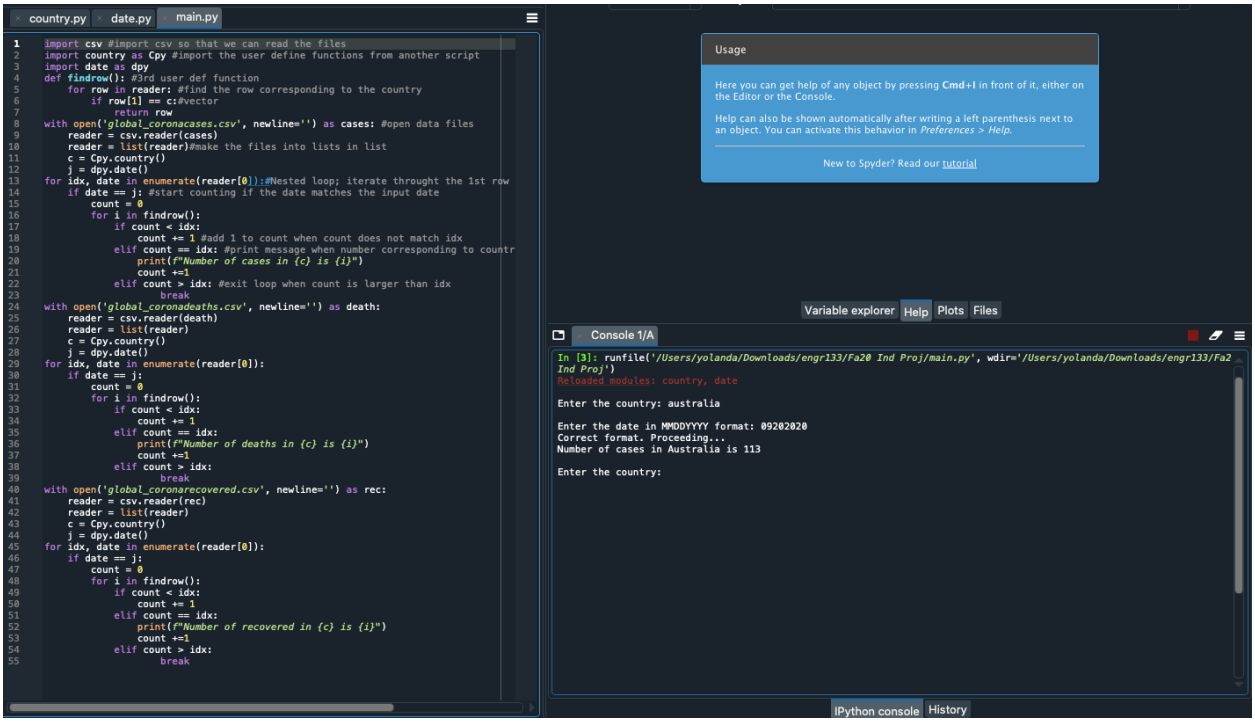
For demonstration, I'm using Australia. Then it will prompt the user to input a date in MMDDYYYY format. See below.



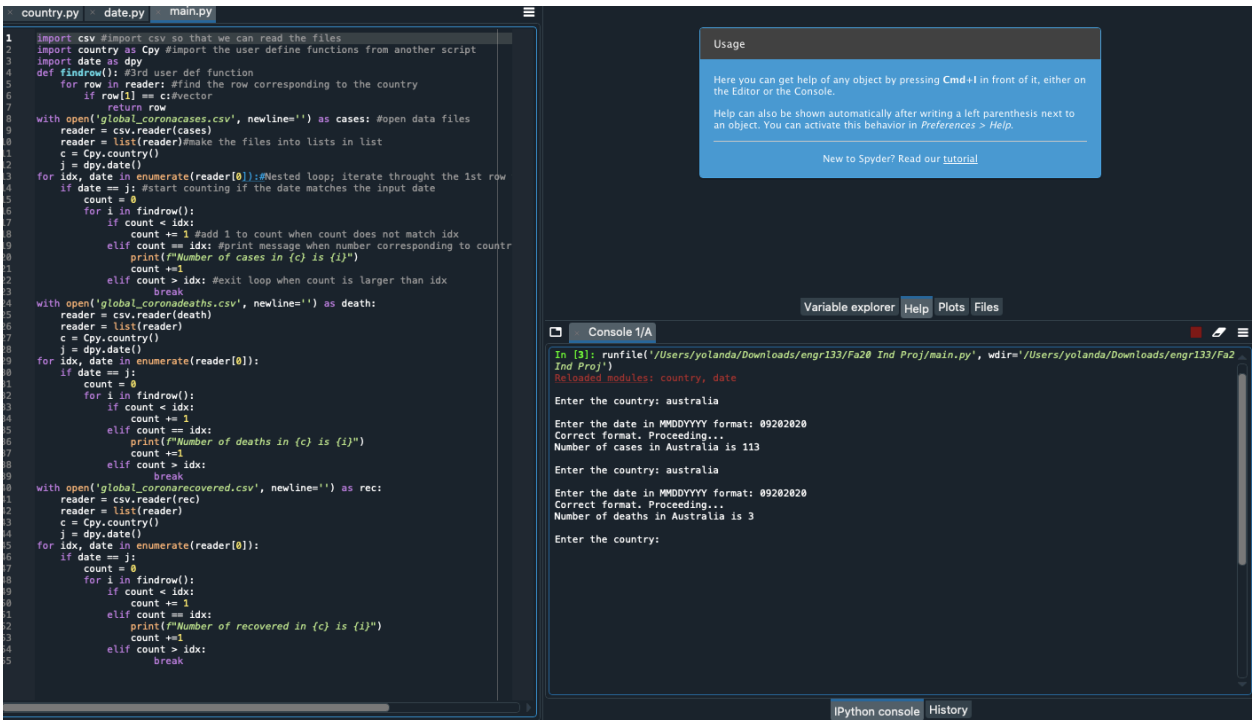
Then it will output the corresponding cases with the country name and date.



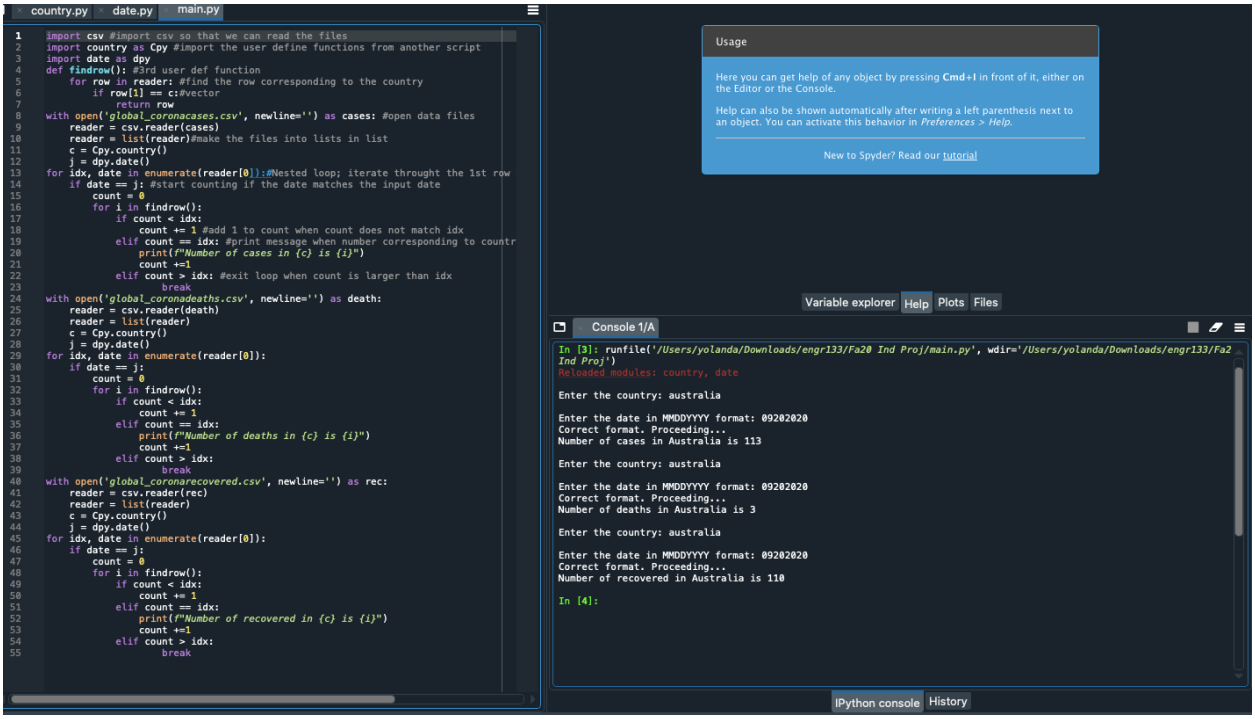
After that, the program will prompt the user to input another country name and date. See below.



This time, it will output the number of deaths. See below.



After inputting country name and date for the third time, it will output the number of recovered. See below.



The output after everything is done should be:

```
Enter the country: australia

Enter the date in MMDDYYYY format: 09202020
Correct format. Proceeding...
Number of cases in Australia is 113

Enter the country: australia

Enter the date in MMDDYYYY format: 09202020
Correct format. Proceeding...
Number of deaths in Australia is 3

Enter the country: australia

Enter the date in MMDDYYYY format: 09202020
Correct format. Proceeding...
Number of recovered in Australia is 110
```

Appendix:

Country.py

```
import sys #import sys so we can do exit()
def country():  # 1st user def function
    Count1 = input("Enter the country: ") #prompts user to input the
country
    Country = Count1.replace(" ", "") #replace the white space if
country name is more than 1 word
    Country = Country.isalpha() #check if input is al alphabets
    if Country == False: #if output is false then prompts user to try
again
        print("Enter a country name. Try again.")
        sys.exit(1) #exits program
    else:
        Country = Count1.title()    #capitalize the first word of the
country
    return Country
```

date.py

```
import sys #import sys for later use
from datetime import datetime #import datetime so we can format the
date
def date(): #2nd user def function
    num = []
    while True: #1st while loop
        num = input("Enter the date in MMDDYYYY format: ") #prompts
user to input date
        datetimeobject = datetime.strptime(num,'%m%d%Y') #change the
date into the desired format
        newformat2 = datetimeobject.strftime('%-m/%-d/%y') #remove the
leading zeros
        if len(num) == 8: #make sure user inuts correct format
            if num.isalpha() == False:
                print("Correct format. Proceeding...")
                break
            else:
                print("Not a number. Try again. ")
                sys.exit(1) #system exits because user did not enter a
number
        else:
            print("Not correct format. Try again.")
            sys.exit(1) #system exits because user did not enter the
correct format
    return newformat2
```

main.py

```
import csv #import csv so that we can read the files
import country as Cpy #import the user define functions from another
script
import date as dpy
def findrow(): #3rd user def function
```

```
    for row in reader: #find the row corresponding to the country
        if row[1] == c:#vector
            return row
with open('global_coronacases.csv', newline='') as cases: #open data
files
    reader = csv.reader(cases)
    reader = list(reader)#make the files into lists in list
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):#Nested loop; iterate throught
the 1st row to find corresponding dates
    if date == j: #start counting if the date matches the input date
        count = 0
        for i in findrow():
            if count < idx:
                count += 1 #add 1 to count when count does not match
idx
            elif count == idx: #print message when number
corresponding to country and date is found
                print(f"Number of cases in {c} is {i}")
                count +=1
            elif count > idx: #exit loop when count is larger than idx
                    break
with open('global_coronadeaths.csv', newline='') as death:
    reader = csv.reader(death)
    reader = list(reader)
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):
    if date == j:
        count = 0
        for i in findrow():
            if count < idx:
                count += 1
            elif count == idx:
                print(f"Number of deaths in {c} is {i}")
                count +=1
            elif count > idx:
                    break
with open('global_coronarecovered.csv', newline='') as rec:
    reader = csv.reader(rec)
    reader = list(reader)
    c = Cpy.country()
    j = dpy.date()
for idx, date in enumerate(reader[0]):
    if date == j:
        count = 0
        for i in findrow():
            if count < idx:
                count += 1
            elif count == idx:
                print(f"Number of recovered in {c} is {i}")
```

```
    count +=1
elif count > idx:
        break
```