Name: Joseph Medina

This program solves a problem of being able to compare data. More in particular, the dataset I used was California's COVID19 history where the user is able to obtain data from two columns of the csv file and obtain a graph comparing two items. The goal of this program is for the user to be able to select the predetermined data they want to compare and have a visualization of that data. By setting up a loop-driven text menu, the user will be able to see what they would like to select. Concurrent.futures is implemented when downloading the csv files from the data.ca.gov website using the copy link address from the download link. First I defined the function of download_csv and use the requests module in order to retrieve the content of the list of urls. Afterwards, the name of the dataset is parsed to show only the name of the file. Afterwards I create the thread pool executor and use the map method to pass through the list of urls and download those individuals. This makes the program run asyncronously so that one thread is downloading one while the second thread is downloading the second url. After downloading the csv files using the link address, variables are set to that they can used and read by using the pandas module. I chose the panda module in order to best read and compare the data with specific columns. Along side matplotlib, plot is used to compare to different columns in the panda dataframe. One small limitation with using the pandas dataframe is that if you wanted to compare individual lines, it would be harder to do. The pandas module is more suited to handle columns of data and compare them rather then look at data on individual rows. One of the biggest limitations of this program is that it is not able to append to the list of csv urls and obtain data from user inputted urls. If there were more urls to download, the concurrent.futures implemented in the program would serve better in optimizing program speed. I would improve this program by allowing the user to input their own url addresses of data sets the user wishes to read. Alongside I would have to implement a way so that the user can choose what data they would like to compare instead of having predetermined datasets to look at.

IMPORT pandas as pd from pandas module
IMPORT time from time module
IMPORT concurrent.futures from concurrent futures module
IMPORT requests from requests module
IMPORT matplotlib.pyplot as plot from matplotlib module

SET list of csv urls = [

'https://data.chhs.ca.gov/dataset/e39edc8e-9db1-40a7-9e87-89169401c3f5/resource/c5978614-6a23-450b-b637-171252052214/download/covid19postvaxstatewidestats.csv',

'https://data.chhs.ca.gov/dataset/e39edc8e-9db1-40a7-9e87-89169401c3f5/resource/de27ce58-edc8-45fb-bebc-08c4b29c5efe/download/covid19postvaxstatewidestats_07172022.csv'

]

FUNCTION main():

SET t1 equal to number of seconds counted at program start

FUNCTION download_csv with parameter csv_url

SET csv_bytes equal to the content retrieved from the url

SET csv_names equal to the name at the end of the csv file

WITH concurrent futures module as threadpoolexecutor as executor

Download the contents using the download_csv functions in the list concurrently

SET t2 equal to number of seconds after program start

PRINT Finishing downloading in (time elapsed) seconds

SET Post_Vax_Without_Boosted equal to pandas read_csv of the first csv file
Set the index of csv file to date

SET Post_Vax_With_Boosted equal to pandas read_csv of the second csv file
Set the index of csv file to date

FUNCTION menu()

print [1] Load Unboosted CSV File
print [2] Load Boosted CSV File
print [3] Vaccinated Cases vs. Unvaccinated Cases
print [4] Vaccinated Deaths vs. Unvaccinated Deaths
print [5] Population Vaccinated vs. Population Unvaccinated
print [6] Population Vaccinated vs. Population Boosted

```
        print [0] Exit the Program

FUNCTION unboosted_CSV():
        print Post_Vax_Without_Boosted

FUNCTION boosted_CSV():
        print Post_Vax_With_Boosted

FUNCTION VC_vs_UC():
        SET data equal Post_Vax_Without_Boosted with columns vaccinated cases and
        unvaccinated cases
        SET df equal to panda dataframe of columns of vaccinated cases and
        unvaccinated cases
        SET plot title to ‘Covid Cases: Vaccinated vs Unvaccinated’ and plot parameters
        Display plot

FUNCTION VD_vs_UD():
        SET data equal Post_Vax_Without_Boosted with columns vaccinated deaths and
        unvaccinated deaths
        SET df equal to panda dataframe of columns of vaccinated deaths and
        unvaccinated deaths
        SET plot title to ‘Covid Cases: Vaccinated Deaths vs Unvaccinated Deaths’ and
        plot parameters
        Display Plot

FUNCTION PV_vs_PuV():
        SET data equal Post_Vax_Without_Boosted with columns population
        unvaccinated  and population vaccinated
        SET df equal to panda dataframe of columns of population vaccinated
         and population unvaccinated
        SET plot title to ‘Covid Population: Population Vaccinated vs Population
        Unvaccinated’ and plot parameters
        Display plot

FUNCTION PV_vs_PB():
        SET data equal Post_Vax_With_Boosted with columns population vaccinated
         and population boosted
        SET df equal to panda dataframe of columns of population vaccinated and
        population boosted
        SET plot title to ‘Covid Population: Population Vaccinated vs Population Boosted’
        and plot parameters
        Display plot

DISPLAY menu
```

```
    SET option equal to the integer of user input option

    WHILE option is not equal to zero
        If user inputs 1
            print option 1 has been called
            DISPLAY unboosted_CSV()
        If user inputs 2
            print option 2 has been called
            DISPLAY boosted_CSV()
        If user inputs 3
            print option 3 has been called
            DISPLAY VC_vs_UC()
        If user inputs 4
            print option 4 has been called
            DISPLAY VD_vs_UD()
        If user inputs 5
            print option 5 has been called
            DISPLAY PV_vs_PuV()
        If user inputs 6
            print option 6 has been called
            DISPLAY PV_vs_PB()
        ELSE
            print Invalid option. Try again
        print space
        DISPLAY menu()
        SET option equal to the integer of user input option

CALL main()
```