Lorena Vasquez
CIS 3120
Homework 3

Homework 3 Report

To begin I started with my API and to fully run through the program. I needed to install the pandas library as pd, import json, and import requests. First I found the website that I wanted to use. I first set the url variable to the url link for the website. Using the library requests I am then able to make a request.get call to the website. Luckily this website does not require an API key. Most API services will ask for an api account, an api token but this website only requires you to call for the website. I then want to print the status code to make sure I have a successful call. Next is to call for the response in json. I then formatted the response to the json.dumps with an indentation of 5, which is a normal indentation to use, to be able to read the data better. I then created several new lists so that I can then append the information to the according column. I then created a dictionary for the columns that I wanted. From there using the dictionary of the columns I zipped together all the lists into the dataframe, and in the following line I then changed the name for each column to my desire. From there using the pandas library I was able to create my dataframe, with a dictionary! I then checked the data frame to make sure all of the information was there with the command: Countries_df.head().

Next was to web scrape a website table. The website I chose was a Worldmeter population by country. I then made a request call for the URL. From that URL I was able to Beautifulsoup through the page content in a 'html.parser'. From there I then printed the soup.prettify() to be able to see the data easier. I then found the table by searching it by the id='example2'. Next I created an empty list so that I can insert my columns. From there I found all the 'th' tags that refer to the headings. I then created a for loop for each heading it will strip the('\n') at the end and then append it to the list. I then created another empty list to be able to call it all at once. I then created a for loop to reiterate through the loop of all the table tows. And from the rows and I then created another loop for it to reiterate through the cells to strip the remaining ending part of the text. From there I said that if the rows are greater than 0, to append the headings of 0 to the rows 0. Each one followed its coordinated row or column. I then created a new dataframe for the Population data. From there I then wanted to check the dataframe head, to make sure all of the information was there. After, I then decided to rename one of the columns to make it easier for me, when I want to merge. I renamed one of the columns 'Country (or dependency) to "Country", just like in the other dataset. From there I made sure to add the inpace=True line to permanently change the data frame. From there, I then wanted to merge the data frames, so I merged the dataframes on the same column name 'Country'. From there, I then checked to see the merged_df, and it worked! Both data frames are now into one. The API portion of this assignment references the world meter of how many cases there have been in total. The web scraping portion references the population of each country. Once merging we can see the number of cases, the population, the number of deaths all in one dataframe. In addition we have data on relevant day information on the pandemic. Lastly, the last step is to call the describe function. For my dataframe it only worked on some columns but not all.