

SI 206 Final Project Report

Jenny Do, Imani Ahmad

GitHub Repository:

https://github.com/jenn1753/206_final_project/blob/main/Final_project/MerryCovid.db

A. Project Goals

Our project goals were to investigate the change in globally reported COVID-19 rates across regions from 6 days before and 6 days after Christmas Day in 2021, using data from the API Ninjas COVID-19 API as well as the API Ninjas Country API that provided regional information. Using the data, we wished to identify which regions of the world experienced the highest spikes in reported COVID-19 cases after Christmas depending on how many countries in certain regions celebrate the holiday. Our reason for choosing Christmas is a time of the year which we hypothesized that the majority of the world would come together and celebrate with friends and family, seeing as it is the most widely recognized and celebrated holiday in the world.

B. Achieved Goals

The goals we were able to achieve in this project were calculating the difference in reported COVID cases for 188 countries around the world before and after Christmas and identifying the top 15 countries that experienced spikes in this time frame. We were also able to identify the region that each country belonged to, and calculate which regions had the highest total COVID case increase in reported COVID cases before and after Christmas by identifying the top 5 regions. Finally, we were able to identify the percentage of each region that celebrates Christmas, the most celebrated holiday in the world, in order to investigate the correlation between Christmas time and COVID spikes.

C. Problems Faced

Some problems that we encountered while working on this project were that not all of the countries that the COVID-19 API was reporting data for were supported by the Countries API, and so there was a slight mismatch of data. In order to account for this issue, we made sure to return None if the country was not supported by the Country API and created a condition that only added supported countries to the data.

Another problem we faced was that when we were adding data into our country_data table, we were getting the error *“sqlite3.ProgrammingError: Incorrect number of bindings supplied. The current statement uses 1, and there are 13 supplied”*. This was due to the fact that our argument was viewing the entry as a list of 13 characters rather

than its own singular entity, so we were able to solve the problem by making the entry an item in a list so it was taken in as a single item.

When trying to execute the SQL max function and adding it to a number, we got the error *“int() argument must be a string, a bytes-like object or a number, not 'sqlite3.Cursor'”* because we were supposed to use `cur.fetchone()` to get a value to assign to a variable. Doing this instead solved the issue.

An issue we faced when performing our calculations was that we needed to join three different tables (the country ids, region ids, and covid data tables) in order to collect information for our calculations, but did not know how to do so. We then surfed the web and discovered a resource which demonstrated how to perform a multiple join operation, so we were able to join three tables simultaneously and retrieve the appropriate data.

When trying to write our calculations into a CSV file, we looped through each item in our tuple list parameter and used *if, else* branches to add commas after each item with the exception of the last tuple item. This last item would be followed by a newline character instead. In looping through and using *item != tup[-1]* as our if condition initially, every tuple item that was identical to the last tuple item would be put on a newline instead of on the same csv file line. We instead shifted to looping through each item in our tuple list parameter through indexing. The condition for adding a comma after a tuple item became *i < (len(tup))*, allowing us to avoid having duplicate values put on a new line.

D. Calculations File

Countries.csv:

Country, Before Christmas, After Christmas, Difference
Algeria, 228, 1325, 1097
Algeria, 262, 399, 133
Andorra, 0, 618, 618
Angola, 19, 1222, 1202
Antigua and Barbuda, 0, 0, 0
Argentina, 3254, 47663, 44409
Armenia, 138, 104, -34
Australia, 0, 0, 0
Austria, 2346, 3181, 835
Azerbaijan, 921, 395, -526
Bahamas, 0, 207, 207
Bahrain, 89, 656, 567
Bangladesh, 211, 512, 301
Barbados, 0, 255, 255
Belarus, 1456, 1510, 54
Belgium, 0, 15688, 15688
Belize, 0, 352, 352
Benin, 0, 0, 0
Bhutan, 1, 0, -1
Bolivia, 889, 7988, 7099
Bosnia and Herzegovina, 0, 842, 842
Botswana, 0, 0, 0
Brazil, 1458, 10190, 8732
Brunai, 16, 0, -16
Bulgaria, 456, 2810, 2354
Burkina Faso, 0, 0, 0
Burundi, 0, 0, 0
Cabo Verde, 16, 394, 378
Cambodia, 0, 0, 0
Cameroon, 0, 0, 0
Canada, 0, 26, 26
Central African Republic, 0, 0, 0
Chad, 0, 0, 0
Chile, 1252, 1012, 560
China, 12, 0, -12
Colombia, 2838, 18481, 8363
Comoros, 0, 231, 231
Costa Rica, 0, 0, 0
Cote d'Ivoire, 0, 2858, 2858
Croatia, 1936, 5567, 3631
Cuba, 62, 433, 371
Cyprus, 448, 9488, 4000
Czechia, 6882, 5778, -224
Denmark, 8212, 18695, 10483
Djibouti, 7, 24, 17
Dominica, 0, 0, 0
Dominican Republic, 272, 1518, 1238
Ecuador, 256, 2232, 1976
Egypt, 910, 947, -37
El Salvador, 46, 0, -46
Equatorial Guinea, 0, 0, 0
Eritrea, 19, 29, 10
Estonia, 426, 1074, 648
Eswatini, 322, 277, -45
Ethiopia, 565, 4899, 4334
Fiji, 0, 0, 0

Eswatini, 322, 277, -45
Ethiopia, 565, 4899, 4334
Fiji, 0, 0, 0
Finland, 1614, 7325, 5711
France, 48473, 232280, 183727
Gabon, 0, 725, 725
Gambia, 0, 24, 24
Georgia, 2195, 2190, -5
Germany, 20348, 41240, 11892
Ghana, 0, 1691, 1691
Greece, 2831, 4056, 3779
Grenada, 0, 0, 0
Guatemala, 69, 797, 728
Guinea, 0, 419, 419
Guinea-Bissau, 0, 0, 0
Guyana, 53, 178, 125
Haiti, 11, 113, 102
Holy See, 0, 0, 0
Honduras, 0, 0, 0
Hungary, 0, 3360, 3360
Iceland, 0, 0, 0
India, 13644, 22775, 9131
Indonesia, 164, 180, 16
Iran, 1968, 1765, -205
Iraq, 272, 384, 32
Ireland, 5124, 28110, 14986
Israel, 794, 3865, 3092
Italy, 24383, 144255, 119952
Jamaica, 0, 329, 329
Japan, 173, 497, 324
Jordan, 3166, 1845, -1351
Kazakhstan, 488, 481, 81
Kenya, 1372, 2791, 1419
Kiribati, 0, 0, 0
Kosovo, 4, 42, 38
Kuwait, 75, 584, 429
Kyrgyzstan, 32, 56, 24
Latvia, 421, 1157, 734
Lebanon, 1479, 4290, 2811
Lesotho, 785, 0, -785
Liberia, 0, 0, 0
Libya, 725, 551, -175
Liechtenstein, 14, 0, -14
Lithuania, 1416, 2076, 668
Luxembourg, 0, 1288, 1288
Madagascar, 0, 0, 0
Malawi, 329, 874, 545
Malaysia, 3188, 3573, 465
Maldives, 93, 160, 67
Mali, 114, 338, 224
Malta, 282, 1483, 1121
Marshall Islands, 0, 0, 0
Mauritania, 56, 319, 263
Mauritius, 28, 0, -28
Mexico, 2538, 18861, 15323
Moldova, 46, 375, 329

Mexico, 2538, 18861, 15323
Micronesia, 0, 0, 0
Moldova, 46, 375, 329
Monaco, 28, 0, -28
Mongolia, 623, 393, -230
Montenegro, 17, 1786, 1529
Morocco, 186, 2834, 1848
Mozambique, 1793, 4861, 3868
Namibia, 1263, 1234, -29
Nauru, 0, 0, 0
Nepal, 133, 224, 91
Netherlands, 12154, 15578, 3416
New Zealand, 28, 0, -28
Nicaragua, 0, 0, 0
Niger, 0, 74, 74
Nigeria, 828, 0, -828
North Macedonia, 190, 542, 352
Norway, 3822, 3845, 23
Oman, 69, 0, -69
Pakistan, 359, 557, 198
Palau, 0, 0, 0
Panama, 386, 4877, 4491
Papua New Guinea, 0, 0, 0
Paraguay, 0, 0, 0
Peru, 1581, 4577, 2996
Philippines, 22, 2219, 2097
Poland, 15976, 15987, -239
Portugal, 4266, 38829, 26563
Qatar, 176, 743, 571
Romania, 469, 1688, 1169
Russia, 27688, 28482, -7198
Rwanda, 169, 1228, 1059
Saint Kitts and Nevis, 0, 81, 81
Saint Lucia, 11, 97, 84
Saint Vincent and the Grenadines, 0, 0, 0
Samoa, 0, 0, 0
San Marino, 0, 394, 394
Sao Tome and Principe, 0, 28, 28
Saudi Arabia, 184, 819, 715
Senegal, 28, 145, 115
Serbia, 882, 2182, 1398
Seychelles, 0, 0, 0
Sierra Leone, 34, 81, 47
Singapore, 255, 344, 89
Slovakia, 4862, 3721, -341
Slovenia, 768, 1896, 1128
Solomon Islands, 0, 0, 0
Somalia, 0, 0, 0
South Africa, 15465, 11754, -3711
South Sudan, 0, 0, 0
Spain, 0, 0, 0
Sri Lanka, 551, 499, -52
Sudan, 0, 0, 0
Suriname, 18, 177, 167
Sweden, 0, 0, 0
Switzerland, 0, 18987, 18987
Syria, 58, 35, -23
Tajikistan, 0, 0, 0
Tanzania, 0, 0, 0
Thailand, 2899, 3111, 212
Timor-Leste, 0, 4, 4
Togo, 128, 147, 619
Tonga, 0, 0, 0
Trinidad and Tobago, 628, 570, -50
Tunisia, 278, 2878, 1848
Turkey, 16918, 40786, 23876
Tuvalu, 0, 0, 0
US, 99844, 511836, 421898
Uganda, 267, 1658, 1361
Ukraine, 3889, 7185, 3296
United Arab Emirates, 551, 2426, 1875
United Kingdom, 81959, 168836, 167877
Uruguay, 285, 1725, 1528
Uzbekistan, 174, 134, -40
Venezuela, 0, 0, 0
Venezuela, 778, 224, -554
Vietnam, 16118, 16515, 495
Yemen, 4, 0, -4
Zambia, 1836, 5881, 4045
Zimbabwe, 2186, 1538, -578

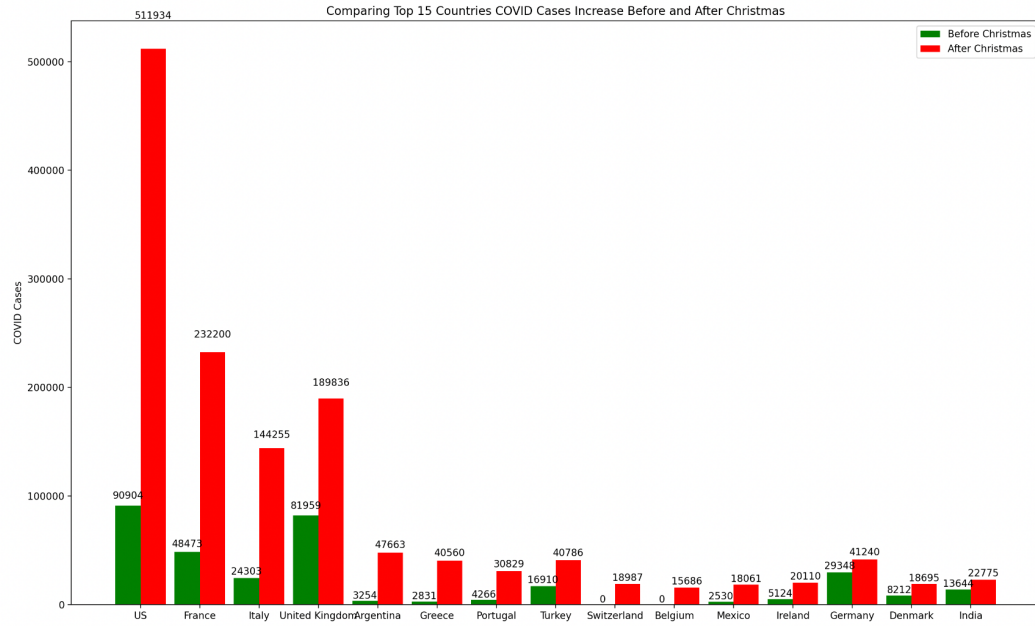
Yemen, 4, 0, -4
Zambia, 1836, 5881, 4045
Zimbabwe, 2186, 1538, -578

[Regions.csv:](#)

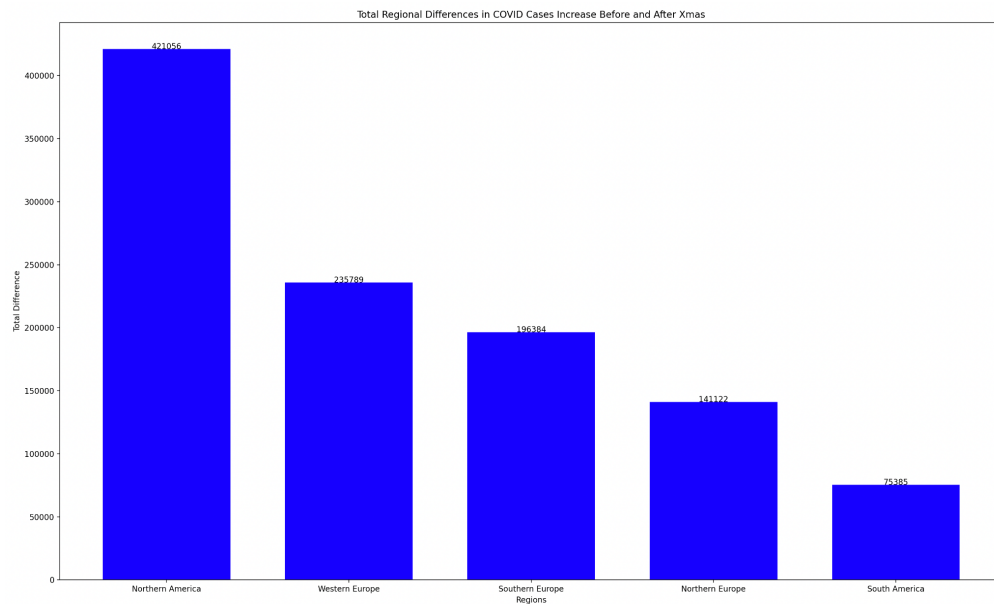
COVID_data.py 3		regions.csv X
regions.csv		
1	Region, Total COVID Difference, Celebrat	
2	Northern America,421056,50	
3	Western Europe,235789,100	
4	Southern Europe,196384,88	
5	Northern Europe,141122,100	
6	South America,75385,100	
7	Western Asia,36683,88	
8	Central America,21056,100	
9	Eastern Africa,15502,89	
10	Southern Asia,9456,78	
11	Western Africa,6543,75	
12	South-Eastern Asia,3872,89	
13	Northern Africa,3582,67	
14	Caribbean,2617,69	
15	Middle Africa,2397,86	
16	Eastern Europe,430,90	
17	Eastern Asia,282,67	
18	Central Asia,65,50	
19	Melanesia,0,100	
20	Micronesia,0,100	
21	Polynesia,0,100	
22	Oceania,-62,100	
23	Southern Africa,-4560,100	
24		

E. Visualizations

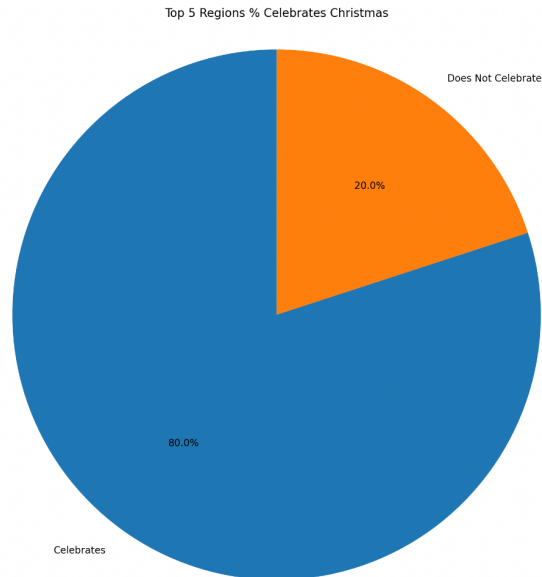
Comparing Top 15 Country COVID Cases Increase Before & After Christmas



Total Regional Differences in COVID Cases Increase Before & After Christmas



Top 5 Regions (With Highest Covid Spike Differences) Percentage that Celebrates Christmas



F. Instructions for Running Code

Run the file COVID_data.py and the database should automatically populate and visualizations should appear.

G. Function Documentation

COVID Data API

- `get_covid_data(api_k, date)`

```
"""
    Input:
    API key and date string in format 'YYYY-MM-DD'

    Output:
    Covid data in form of list of dictionaries containing global countries
    covid information for that date

    Format:
```

```

        [{'country': 'Afghanistan', 'region': '', 'cases': {'total':
157787, 'new': 42}}, {'country': 'Albania', 'region': '', 'cases':
{'total': 205777, 'new': 228}}]
    """

```

- **create_country_before_after_data_dict(before_data, after_data)**

```

    """
    Input:
        Two list of dictionaries, one for global covid data from the date
before Christmas and one for global data from the date after Christmas
    Format:
        [{'country': 'Afghanistan', 'region': '', 'cases': {'total':
157787, 'new': 42}}, {'country': 'Albania', 'region': '', 'cases':
{'total': 205777, 'new': 228}}]

    Output:
        Dictionary that contains country names as the keys and a dictionary
containing that countries' increase in number of covid cases before
Christmas and after Christmas as the values
    Format:
        {'country1': {'before':123, 'after':123}, 'country2':
{'before':123, 'after':123}}
    """

```

Country Data API

- **get_country_data(api_key, country)**

```

    """
    Input:
        API key and country name string

    Output:
        A list of one dictionary containing the entered country's information
    Format:
        [{"name": "United States", "pop_growth": "0.6", "region":
"Northern America", "pop_density": "36.2", "internet_users": "87.3",
"gdp_per_capita": "62917.9", "fertility": "1.8"}]
    """

```

- **get_regions(country_list)**

```

    """

```

```

Input:
List of countries

Output:
Dictionary containing country name as the keys and region category as
the value

Format:
{'country': 'region', 'country2': 'region'}

"""

```

- **celebrated_pct_per_region(sorted_reg_covid_diff_data, region_dict, filename)**

```

"""
Input:
A dictionary containing region names as keys and the total region
differences of all covid increase data before christmas and after
christmas

A dictionary containing the country name as keys and the region
category as the keys

The file name of the json file holding a list of dictionaries
containing country name and whether they celebrated Christmas
(unofficially, yes, or no)

Format:
[{"country": "Afghanistan", "celebrated": "No", "date": "-",
"notes": "Christmas and Christianity are actively—and at times
violently—discouraged by the current rulers of Afghanistan"}]

Output:
A dictionary containing, in descending order, the region name as the
key and the percentage of all countries in region that celebrates
Christmas (Yes and Unofficially) as values

Format:
{'region_name': }

"""

```

- **sorted_regional_covid_increase_difference(filtered_before_after_data, regions_dict)**

```

"""

Input:

```



```

    A list of dictionaries containing countries that are confirmed to be
in both the COVID and countries API as the keys and a dictionary
containing their covid cases increase before christmas and covid cases
increase after christmas as the values

    Format:

    {'country':{'before': 123, 'after':123}}

    A dictionary containing the country name as keys and the region
category as the values

    Output:

    A dictionary containing, in descending order, region names as keys and
the total region differences of all covid increase data before christmas
and after christmas

    """

```

Database

- **setUpDatabase(db_name)**

```

    """

    Input:

    Database name (string)

    Output:

    Creates cursor and connection and sets up empty database

    """

```

- **create_region_table(cur, conn)**

```

    """

    Input:

    Cursor and connection variables

    Output:

    Creates table regions_data with columns: region_id, region_name, and
celebrate_pct (percent of region that celebrates Xmas)

    """

```

- **create_country_id_table(cur, conn)**

```

    """

    Input:

    Cursor and connection variables

```

```
Output:
Creates table country_ids with columns: country_id and country_name
"""
```

- **create_country_table(cur, conn)**

```
"""
Input:
Cursor and connection variables

Output:
Creates table country_data with columns: country_id and region_id
"""
```

- **create_country_before_after_data_table(cur, conn)**

```
"""
Input:
Cursor and connection variables

Output:
Creates table country_covid_data with columns: country_id,
before_covid_data (cases reported before Xmas), after_covid_data (cases
reported after Xmas)
"""
```

- **add_region(cur, conn, sorted_covid_reg_diff_dict, celebrated_pct_per_region)**

```
"""
Input:
Cursor and connection variables

A dictionary containing region names as keys and the total region
differences of all covid increase data before christmas and after
christmas

A dictionary containing, in descending order, the region name as the
key and the percentage of all countries in region that celebrates
Christmas (Yes and Unofficially) as values

Format:
{'region_name': 90, 'region_name':88}

Output:
```

```

    Populates region_data table with country_id, before_covid_data (cases
reported before Xmas), and after_covid_data (cases reported after Xmas)
for each country, limiting to 25 entries at a time

    """

```

- **add_country_ids(cur, conn, regions_dict)**

```

    """

    Input:

    Cursor and connection variables

    A dictionary containing the country name as keys and the region
category as the values

    Output:

    Populates country_ids table with country_id and country_name for each
country, limiting to 25 entries at a time

    """

```

- **add_country(cur, conn, regions_dict)**

```

    """

    Input:

    Cursor and connection variables

    A dictionary containing the country name as keys and the region
category as the values

    Output:

    Populates region_data table with country_id and region_id for each
country, limiting to 25 entries at a time

    """

```

- **add_covid_info(cur, conn, filtered_covid_before_after_dict)**

```

    """

    Input:

    Cursor and connection variables

    A list of dictionaries containing countries that are confirmed to be
in both the COVID and countries API as the keys and a dictionary
containng their covid cases increase before christmas and covid cases
increase after christmas as the values

    Format:

    {'country':{'before': 123, 'after':123}}

```

```

Output:
Populates country_covid_data table with country_id, before xmas covid
data, and after xmas covid data for each country, limiting to 25 entries
at a time
"""

```

Calculations

- **select_db_covid_data(cur, conn)**

```

"""
Input:
Cursor and connection variables

Output:
Joins country_ids table and country_covid_data table on country_id
column to create a list of tuples containing country name, number of
COVID case increase before Christmas and the number of COVID case
increase after Christmas

Format:
[(country, 123, 123), (country, 123, 123)]

"""

```

- **calc_db_covid_diff(tup_list_covid_data)**

```

"""
Input:
A list of tuples containing country name, number of COVID case
increase before Christmas and the number of COVID case increase after
Christmas

Format:
[(country, 123, 123), (country, 123, 123)]

Output:
A list of tuples containing country name and the difference in number
of COVID case increase before Christmas and the number of COVID case
increase after Christmas

"""

```

- **region_country(cur, conn)**

```

"""
Input:

```

```
Cursor and connection variables
```

```
Output:
```

```
Joins country_data table, region_data table, and country_ids table on  
country_id column and region_id column to create a list of tuples  
containing region name and country name
```

```
Format:
```

```
[(region, country), (region, country)]
```

```
"""
```

- `region_diff_dict(country_diff_tup_list, country_region_tup_list)`

```
"""
```

```
Input:
```

```
A list of tuples containing country name and the difference in number  
of COVID case increase before Christmas and the number of COVID case  
increase after Christmas
```

```
A list of tuples containing region name and country name
```

```
Format:
```

```
[(region, country), (region, country)]
```

```
Output:
```

```
A dictionary containing the region name as keys and the sum of the  
difference of COVID cases increase before Christmas and COVID cases  
increase after Christmas for the countries in the region as values
```

```
"""
```

- `get_region_percent(cur, conn)`

```
"""
```

```
Input:
```

```
Cursor and connection variables
```

```
Output:
```

```
A list of tuples containing region name and the percentage of the  
region that celebrates Christmas officially or unofficially
```

```
Format:
```

```
[(region, %), (region, %)]
```

```
"""
```

- `prep_country_csv(tup_list_covid_data, country_covid_diff)`

```

"""
    Input:
        A list of tuples containing country name, number of COVID case
        increase before Christmas and the number of COVID case increase after
        Christmas

        Format:
            [(country, 123, 123), (country, 123, 123)]

        A list of tuples containing country name and difference between before
        and after Christmas Covid cases

        Format:
            [(country, 123), (country, 123)]

    Output:
        A list of tuples containing country name, number of COVID case
        increase before Christmas, number of COVID case increase after Christmas,
        and the difference between the two values

        Format:
            [(country, 123, 123, 123), (country, 123, 123, 123)]
"""

```

- **prep_region_csv(region_xmas_pct, region_diff_d)**

```

"""
    Input:
        A list of tuples containing the region name and the percentage of
        countries within that region that either unofficially or does celebrate
        Christmas

        A dictionary containing the region names as the keys and the sum of
        the difference of COVID cases increase before Christmas and COVID cases
        increase after Christmas for the countries in the region as the values

    Output:
        A list of tuples containing the region name, the sum of the difference
        of COVID cases increase before Christmas and COVID cases increase after
        Christmas for the countries in the region, and the percentage of
        countries within that region that either unofficially or does celebrate
        Christmas

    """

```

- **write_country_csv(file_name, prep_lst)**

```

"""

```

Input:

The name of the file that will be created and written to

A list of tuples containing the region name, the sum of the difference of COVID cases increase before Christmas and COVID cases increase after Christmas for the countries in the region, and the percentage of countries within that region that either unofficially or does celebrate Christmas

Output:

None, function writes country name, before christmas COVID cases increase, and after Christmas COVID cases increase, and the difference between before and after data into a csv file

"""

- **write_region_csv(file_name, prep_lst)**

"""

Input:

File name (string)

A list of tuples containing the region name, the sum of the difference of COVID cases increase before Christmas and COVID cases increase after Christmas for the countries in the region, and the percentage of countries within that region that either unofficially or does celebrate Christmas

Output:

None, function writes region name, the difference between before and after data, and the percentage of countries within that region that either unofficially or does celebrate Christmas into a csv file

"""

Visualizations

- **sorted_covid_diff(country_covid_diff, tup_list_country_covid_ba)**

"""

Input:

A list of tuples containing country name and the difference in number of COVID case increase before Christmas and the number of COVID case increase after Christmas

```

    Joins country_ids table and country_covid_data table on country_id
    column to create a list of tuples containing country name, number of
    COVID case increase before Christmas and the number of COVID case
    increase after Christmas

    Format:

    [(country, 123, 123), (country, 123, 123)]

    Output:

    A list of tuples that contains the top 15 countries based on their
    difference in number of COVID case increase before Christmas and the
    number of COVID case increase after Christmas, their number of COVID case
    increase before Christmas, and their number of COVID case increase after
    Christmas
    """

```

- **country_double_bc(ba_15_countries)**

```

    """
    Input:

    A list of tuples that contains the top 15 countries based on their
    difference in number of COVID case increase before Christmas and the
    number of COVID case increase after Christmas, their number of COVID case
    increase before Christmas, and their number of COVID case increase after
    Christmas

    Output:

    A double bar chart Matplotlib visualization depicting before and after
    Covid data for the top 15 countries
    """

```

- **sorted_region_diff(reg_diff_d)**

```

    """
    Input:

    A dictionary containing the region name as keys and the sum of the
    difference of COVID cases increase before Christmas and COVID cases
    increase after Christmas for the countries in the region as values

    Output:

    A list of tuples that contains the top 5 regions based on their
    difference in number of COVID case increase before Christmas and the
    number of COVID case increase after Christmas

    """

```


- `region_diff_single_bc(sorted_5_regions_lst)`

```
"""
    Input:
        A list of tuples that contains the top 5 regions based on their
        difference in number of COVID case increase before Christmas and the
        number of COVID case increase after Christmas

    Output:
        A bar chart Matplotlib visualization depicting the top 5 regions with
        the highest difference in covid cases before and after Christmas
    """
```

- `sorted_5_region_diff(reg_diff_d, reg_pct_tup_lst)`

```
"""
    Input:
        A dictionary containing the region name as keys and the sum of the
        difference of COVID cases increase before Christmas and COVID cases
        increase after Christmas for the countries in the region as values

        A list of tuples containing region name, sum of the difference of
        COVID cases increase before Christmas and COVID cases increase after
        Christmas for the countries in the region as values, percentage of
        countries in that region that celebrates Christmas

    Output:
        A list of tuples that contains the top 5 regions based on their
        difference in number of COVID case increase before Christmas, the number
        of COVID case increase after Christmas, and the percentage of countries
        in that region that celebrates Christmas
    """
```

- `celebrates_pie_chart(reg_5_lst)`

```
"""
    Input:
        A list of tuples that contains the top 5 regions based on their
        difference in number of COVID case increase before Christmas and the
        number of COVID case increase after Christmas and the percentage of each
        region that celebrates Christmas

    Output:
```

```
A pie chart Matplotlib visualization depicting the percentage of the
top 5 regions with the highest difference in covid cases before and after
Christmas that celebrate Christmas

''' '''
```

Main

- **main()** - runs all functions

H. Resource Documentation

Date	Issue Description	Location of Resource	Result (Did it solve the issue?)
December 9, 2022	We needed to narrow down the timeframe in which we compared our covid data. We needed a range in which COVID symptoms are proven to have manifested after exposure.	https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html#:~:text=Symptoms%20may%20appear%202%2D14,Fever%20or%20chills	We decided to take a data snapshot 6 days prior to Christmas day and 6 days after Christmas day to take into account the time it takes for symptoms to be noticeable.
December 9, 2022	We needed to know which countries and regions celebrate Christmas globally in order to understand the significance of our calculations.	https://worldpopulationreview.com/country-rankings/countries-that-celebrate-christmas	Yes, we were able to identify which regions contain a majority of countries that celebrate Christmas and compare that to the regions with the greatest COVID-19 spikes.
December 8, 2022	We needed a source of COVID data that would allow us to specify the dates in which to analyze the global data.	https://api-ninjas.com/api/covid19	We were able to use this API to gather COVID data exactly 6 days before and 6 days after Christmas globally.

December 8, 2022	We needed a resource that would automate the process of assigning each country in our data to a particular region for more efficiency in analysis.	https://api-ninjas.com/api/country	We were able to use this API to categorize each country within our data to a certain region name. This allowed for a broader overview of our COVID data compared to analyzing country by country.
December 11, 2022	We were getting an error when adding into our country_data table (sqlite3.ProgrammingError: Incorrect number of bindings supplied. The current statement uses 1, and there are 13 supplied.)	https://techoverflow.net/2019/10/14/how-to-fix-sqlite3-python-incorrect-number-of-bindings-supplied-the-current-statement-uses-1-supplied/	This resource explained that the argument was viewing the entry as a list of characters rather than its own singular entity, so we made it an item in a list instead which solved the issue.
December 11, 2022	We were getting an error when trying to execute the SQL max function and adding it to a number (int() argument must be a string, a bytes-like object or a number, not 'sqlite3.Cursor')	Discussion12.py	This resource allowed us to refresh on what the fetchone() does so we were able to use that to get the first item from the first tuple in the database.
December 11, 2022	We needed to join three different tables (the country ids, region ids, and covid data tables) in order to collect information for our calculations but did not know how to do so.	https://learnsql.com/blog/how-to-join-3-tables-or-more-in-sql/	This resource demonstrated how to perform a multiple join operation, so we were able to join three tables simultaneously to get the appropriate data.

December 11, 2022	We did not know how to create a double bar chart visualization using Matplotlib.	https://stackoverflow.com/questions/14270391/python-matplotlib-multiple-bars	This resource provided an example and documentation for a double bar chart that we were able to refer to.
December 12, 2022	We did not know how to create a pie chart in Matplotlib.	https://matplotlib.org/stable/gallery/pie_and_polar_charts/pie_features.html	This resource provided an example and documentation for a pie chart that we were able to refer to.