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.

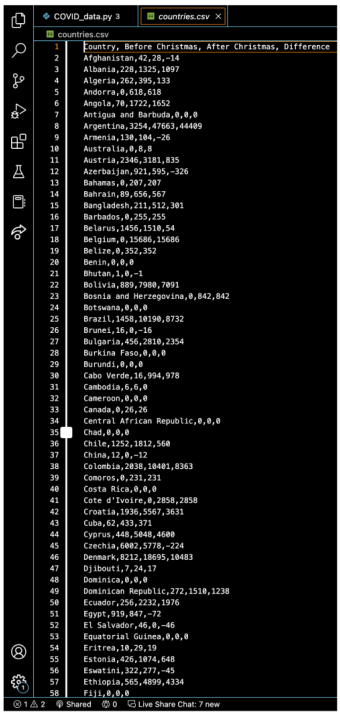
When trying to execute the SQL max function and adding it to a number, we got the error *"int() argument must be a string, a bytes-like object or a number, not 'sqlite3.Cursor'"* because we were supposed to use cur.fetchone() to get a value to assign to a variable. Doing this instead solved the issue.

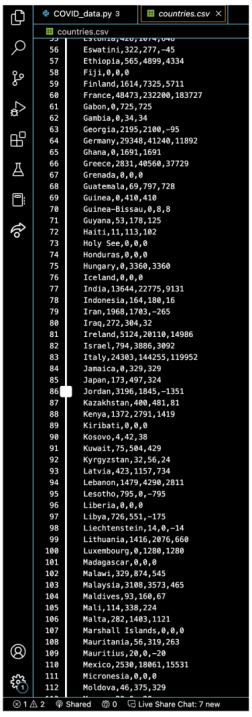
An issue we faced when performing our calculations was that we needed to join three different tables (the country ids, region ids, and covid data tables) in order to collect information for our calculations, but did not know how to do so. We then surfed the web and discovered a resource which demonstrated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

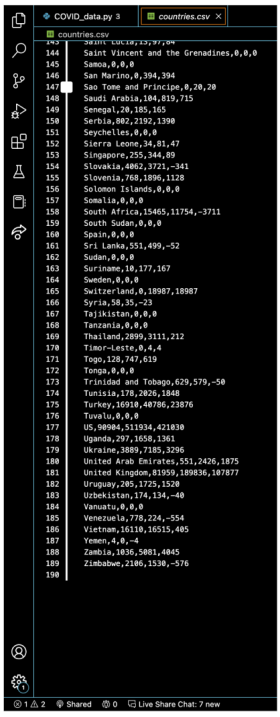
When trying to write our calculations into a CSV file, we looped through each item in our tuple list parameter and used *if, else* branches to add commas after each item with the exception of the last tuple item. This last item would be followed by a newline character instead. In looping through and 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:





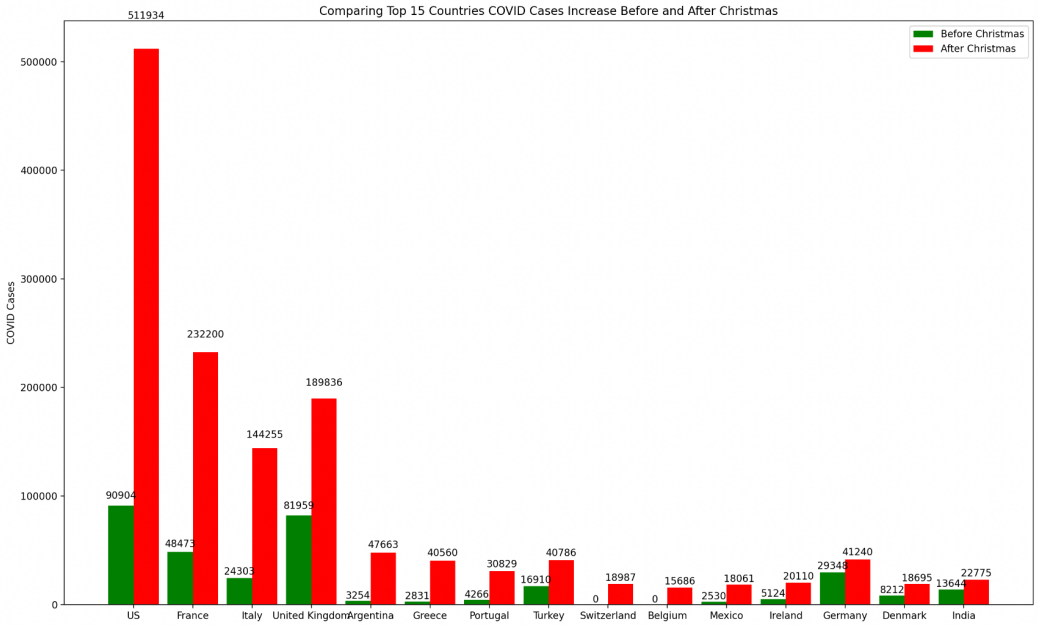


Regions.csv:

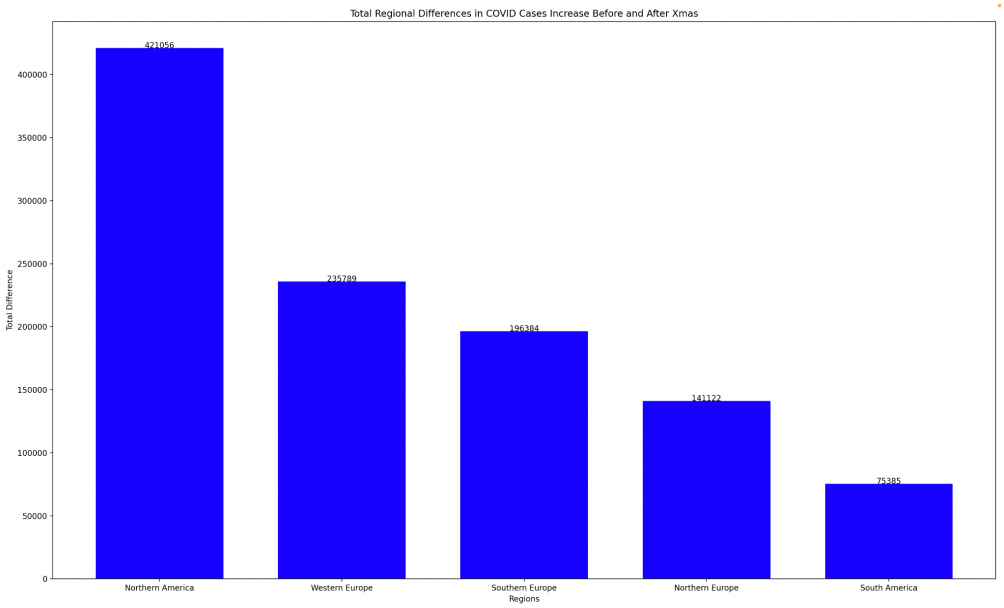

COVID_data.py 3
regions.csv
regions.csv
Region, Total COVID Difference, Celebrat
Northern America,421056,50
Western Europe,235789,100
Southern Europe,196384,88
Northern Europe,141122,100
South America,75385,100
Western Asia,36683,88
Central America,21056,100
Eastern Africa,15502,89
Southern Asia,9456,78
Western Africa,6543,75
South-Eastern Asia,3872,89
Northern Africa,3582,67
Caribbean,2617,69
Middle Africa,2397,86
Eastern Europe,430,90
Eastern Asia,282,67
Central Asia,65,50
Melanesia,0,100
Micronesia,0,100
Polynesia,0,100
Oceania,-62,100
Southern Africa,-4560,100

E. Visualizations

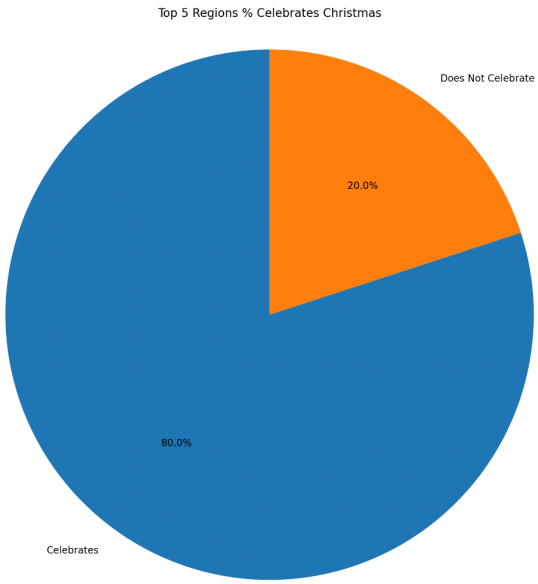
Comparing Top 15 Country COVID Cases Increase Before & After Christmas



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.