Unit 40.

Global Corona Pandemic Analysis Mini Project

Mission

Global Corona Pandemic Analysis Mini Project

- In the history of mankind, there have been several pandemic situations.
- Every time a new epidemic appears, mankind has been trying to find new vaccines and treatments. In each case, the most important role was to conduct reliable epidemiological investigations and transparently disclose the results.
- Mankind created new vaccines and treatments based on publicly released data and tried to find a way until an alternative is available.
- In the pandemic situation, various data analysis, such as how many people are currently infected? where are the regions where infected people are currently decreasing? what is the age and living environment where the number of infected people is high? and whether the number of infected people is decreasing due to the effectiveness of the vaccine? and data visualization for people to understand easier have made a lot of contributions.
- Through this mini-project, we are also trying to get descriptive statistics on which countries have the highest number of infections based on the corona-related data available to the public.
- The result will display the cumulative infection level on a map of the world so that anyone can check it at a glance.
- We use the data from the [Coronavirus (COVID-19) Vaccinations] database collected in real time by https://ourworldindata.org/
- We can use the data required for practice in **"./data//covid/covid-vaccination-doses-per-capita.csv".**
- For reference, we need to understand that there are some data are excluded because some countries do not disclose the data to the world, but we can still practice with the data provided.

Let’s code

Step 1

Let's prepare the data.

```
1  import numpy as np
2  import pandas as pd
3  import datetime
4  import matplotlib.pyplot as plt
5  from datetime import date, datetime, time, timezone
6
7  df = pd.read_csv("./data//covid/covid-vaccination-doses-per-capita.csv")
```

Line 7

- Enter the path of the downloaded file as a relative path.

Step 1

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34684 entries, 0 to 34683
Data columns (total 4 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Entity                          34684 non-null  object
 1   Code                            29711 non-null  object
 2   Day                             34684 non-null  object
 3   total_vaccinations_per_hundred  34684 non-null  float64
dtypes: float64(1), object(3)
memory usage: 1.1+ MB
```

Line 1

- Check the technical summary of the data.

Step 1

```
1 df['Date'] = pd.to_datetime(df['Day'])
2 df.set_index('Date', inplace= True)
3 df.drop(['Day'], axis=1, inplace=True)
4
5 df.head()
```

Date	Entity	Code	total_vaccinations_per_hundred
2021-02-22	Afghanistan	AFG	0.00
2021-02-28	Afghanistan	AFG	0.02
2021-03-16	Afghanistan	AFG	0.14
2021-04-07	Afghanistan	AFG	0.30
2021-04-22	Afghanistan	AFG	0.60

Line 1, 2, 3, 5

- 1: Since Day is an Object data type, there is no need to change it to a character type, just convert it to datetime.
- 2: Change the index to the newly created column.
- 3: Delete unnecessary columns.
- 5: Check the data frame.

Step 1

```
1 len(df['Entity'].unique())
```

```
236
```

Line 1

- You can check the data from a total of 236 countries.

Step 2

Let's create a group around the entity column and create a new data frame to store accumulated data by country.

```
1 covid_c = df.groupby(['Entity'])
```

Step 2

If you see the result of the code below, you can see that the number of rows (data frame length) of the data frame for each key (which will be the country name in this case) is different. This means that the actual data provision status is different for each country. Remember that this is representative of one of the real-world situations.

```
# Separate the information of the created group by key and print them.
for key,group in covid_c:
    print('+key:', key)
    print('+number:', len(group))
    print(group.head())
    print('\n')
```

```
+key: Afghanistan
+number: 28
                Entity Code  total_vaccinations_per_hundred
Date
2021-02-22  Afghanistan  AFG                            0.00
2021-02-28  Afghanistan  AFG                            0.02
2021-03-16  Afghanistan  AFG                            0.14
2021-04-07  Afghanistan  AFG                            0.30
2021-04-22  Afghanistan  AFG                            0.60


+key: Africa
```

Step 2

If you see the result of the code below, you can see that the number of rows (data frame length) of the data frame for each key (which will be the country name in this case) is different. This means that the actual data provision status is different for each country. Remember that this is representative of one of the real-world situations.

```
1  # Separate the information of the created group by key and print them.
2  for key,group in covid_c:
3      print('+key:', key)
4      print('+number:', len(group))
5      print(group.head())
6      print('\n')
```

Line 1, 3, 4, 5

- 1: Separate the information of the created group by key and print them.
- 3: Print the group's key name.
- 4: The number of data for the key (The number of data by country)
- 5: Print 5 lines for each group.

Step 2

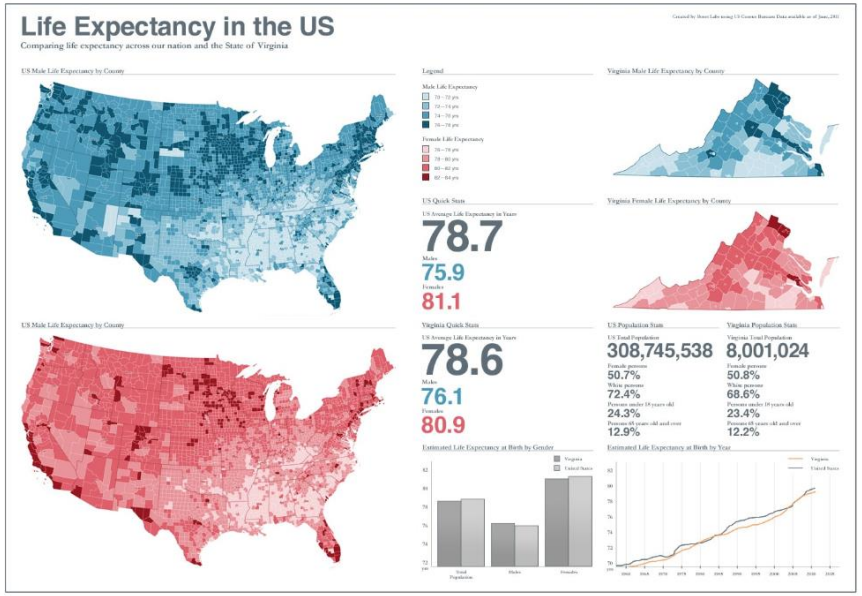
Let's save the total for each group in a new data frame. We create statistics about the cumulative number of cases by 100 people in each country from the time of the corona outbreak to the present time.

```
1 total_df = covid_c.sum()
2 total_df.head()
```

Entity	total_vaccinations_per_hundred
Afghanistan	66.26
Africa	1313.04
Albania	6456.90
Algeria	172.46
Andorra	1474.34

Step 3

- We have learned and practiced data visualization of charts for various descriptive statistics using matplotlib or seaborn library. However, in the field of data visualization, there are many cases of using map objects as shown in the image below. The result of this mini-project is to express the corona situation in the form of a data frame by changing the color according to the weight on the world map.



- Before solving the mission, we will learn how to express data on the map using the folium library.
- https://python-visualization.github.io/folium/

https://www.webdesignerdepot.com/2009/10/30-superb-examples-of-infographic-maps/

Step 3

1) Install folium library

- Move to the current virtual environment, and install the library through one of the two commands below.

  - **pip install folium**
  - **conda install folium -c conda-forge**: It is recommended to install using conda.

Step 3

2) Get coordinate information of the location you want

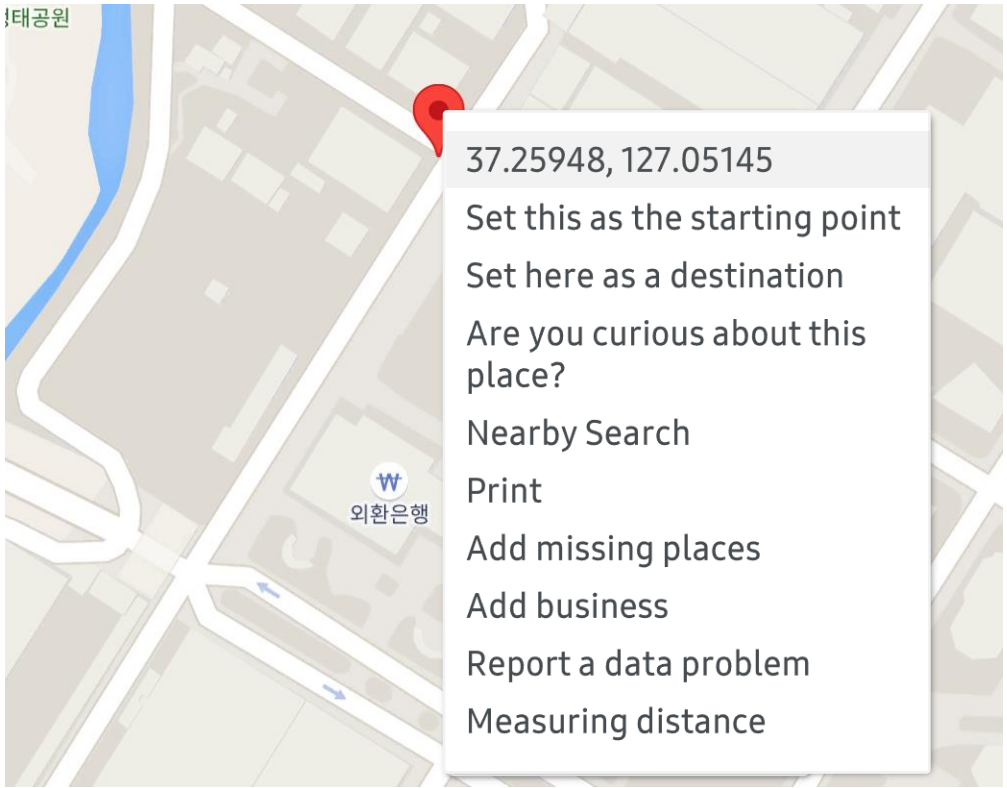
- You need to know the latitude and longitude information of the location to display a map of the location you want. The easiest way to get a latitude and longitude location is to use the Google Maps service. The method below is the method when using a normal PC.

  - Go to https://www.google.com/maps
  - Search the location you want on the map or move it by dragging the mouse.
  - Right-click the location where you want to accurately get latitude and longitude information.
  - When you click the coordinate information, the information is automatically copied to the clipboard.

Step 3

2) Get coordinate information of the location you want

- The image below is the result of searching for the location of Samsung Electronics' headquarters in Korea as a sample.



Step 3

3) Create a map

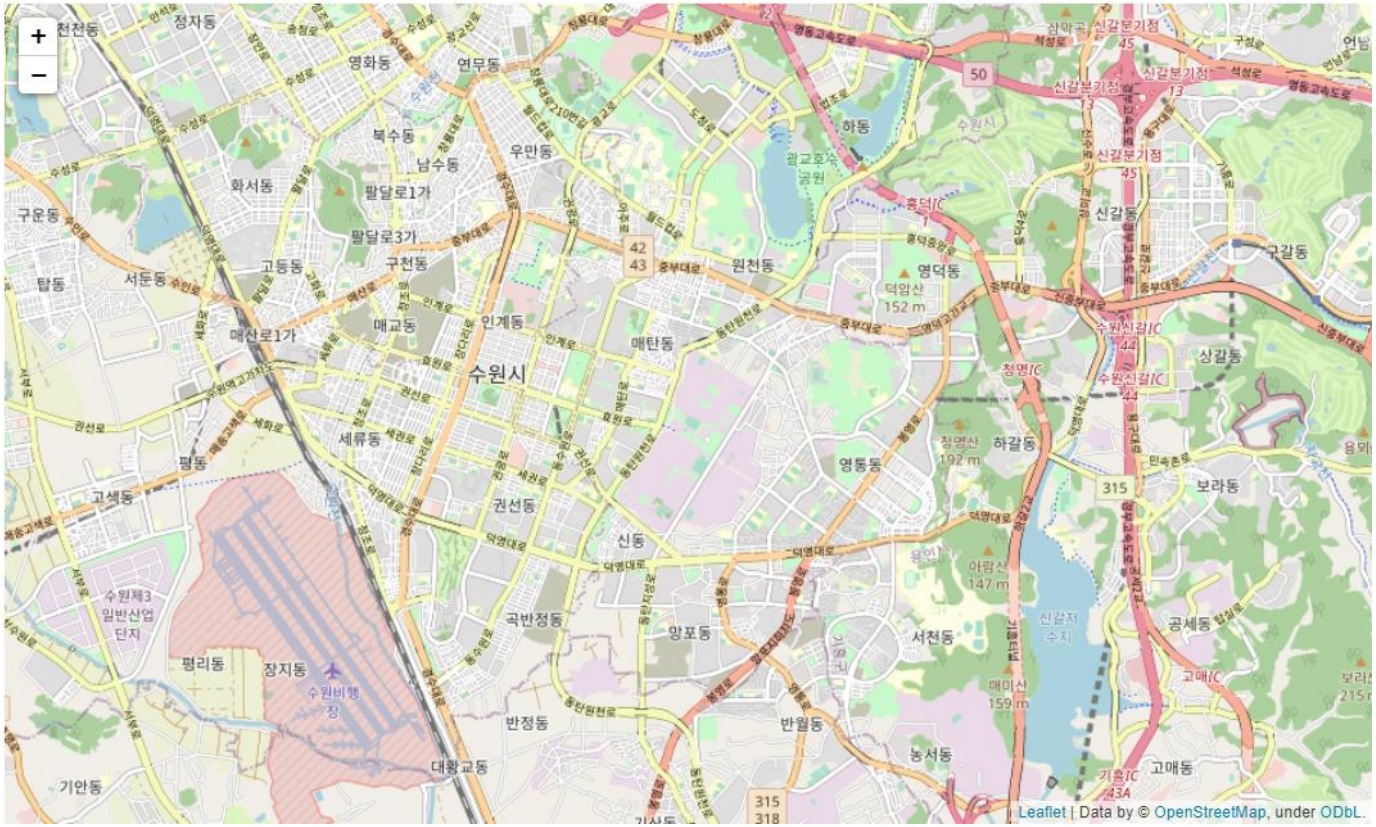
```
1  import folium
2
3  map = folium.Map(location = [37.2594750011864, 127.05145091394964],
4                   zoom_start=13,
5                   )
6
7  # m.save("index.html")
8
9  map
```

Line 3, 4, 6, 7

- 3: Enter the coordinates of the center of the map in the order of latitude and longitude.
- 4: Magnification factor for the initial rendering of the map
- 7: m.save("index.html")

Step 3

3) Create a map



Step 3

4) Apply a style to the map

- You can specify the graphic style of the map displayed through the tiles parameter input of the map() function. If no value is entered, the openstreetmap style is applied. Here are a few of the most used ones:

  - tamenterrain
  - stamentoner
  - stamenwatercolor
  - cartodbpositron
  - cartodbdark_matter
  - openstreetmap

- There are several styles, you can change them one by one and see them for yourself.

Step 3

4) Apply a style to the map

```
1  import folium
2
3  map = folium.Map(location = [37.2594750011864, 127.05145091394964],
4                   zoom_start=13,
5                   tiles="stamenwatercolor",
6                   )
7
8  map
```

Line 3, 4, 5

- 3: Center coordinates of the map
- 4: Magnification factor for the initial rendering of the map
- 5: If nothing is specified, the default value is openstreetmap.

Step 3

4) Apply a style to the map

Step 3

5) How to display marker icons and information at specific locations

```
marker_map = folium.Map(location=[45.372, -121.6972], zoom_start=12, tiles="Stamen Terrain")

folium.Marker(
    location=[45.3288, -121.6625], #마커가 표시 될 위도 경도 정보
    popup="Mt. Hood Meadows",
    icon=folium.Icon(icon="cloud"), #마커 아이콘 스타일
).add_to(marker_map)

folium.Marker(
    location=[45.3311, -121.7113],
    popup="Timberline Lodge",
    icon=folium.Icon(color="green"),
).add_to(marker_map)

folium.CircleMarker(
    location=[45.3800, -121.6000],
    radius=100,
    popup="circle",
    color="#3186cc",
    fill=True,
    fill_color="#3186cc", #원
).add_to(marker_map)


marker_map
```

Step 3

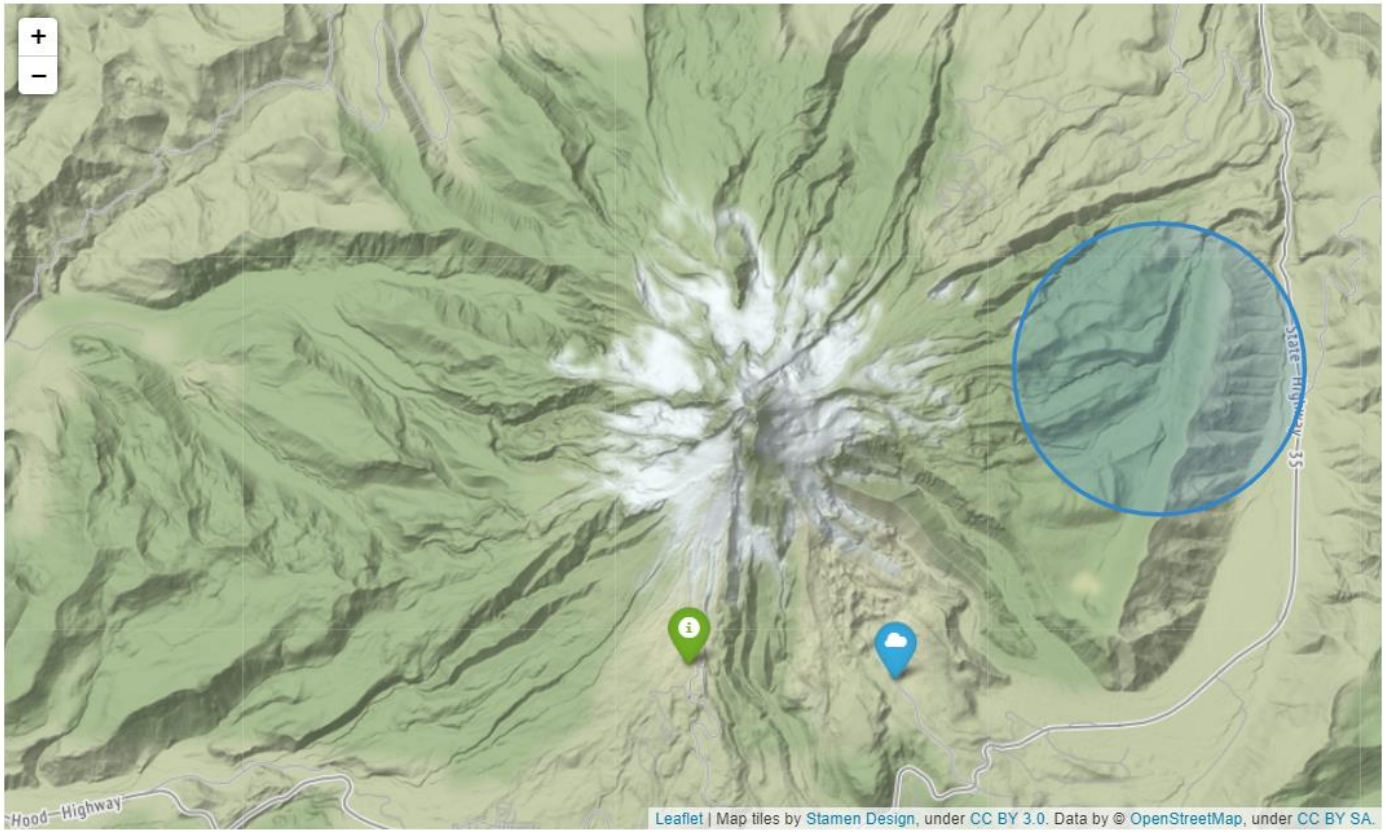
5) How to display marker icons and information at specific locations

Line 4, 5, 6, 15, 17, 18, 19, 20, 21

- 4: Latitude and longitude information where the marker will be displayed
- 5: Information to be displayed as a pop-up message when a marker is clicked
- 6: Marker icon style
- 15: A method to specify an area in a circle.
- 17: Specify the size of the circle
- 18: Information to be displayed as a pop-up message when the circle is clicked
- 19: Color information of the borderline of the circle
- 20: Decide whether to paint the inside of the circle
- 21: Color information to be painted in a circle

Step 3

5) How to display marker icons and information at specific locations



Step 3

6) How to display a step-by-step diagram in the map area

- It is a method to visualize data by coloring the area surrounded by a certain boundary on a map, such as a boundary line between specific countries or an administrative district.
- As the value of the information to be delivered increases, the color painted in the corresponding area becomes darker.

Step 3

6) How to display a step-by-step diagram in the map area

```
import pandas as pd

url = (
    "https://raw.githubusercontent.com/python-visualization/folium/master/examples/data"
)
state_geo = f"{url}/us-states.json"
state_unemployment = f"{url}/US_Unemployment_Oct2012.csv"
state_data = pd.read_csv(state_unemployment)

m = folium.Map(location=[48, -102], zoom_start=3)

folium.Choropleth(
    geo_data=state_geo,
    name="choropleth",
    data=state_data,
    columns=["State", "Unemployment"],
    key_on="feature.id",
    fill_color="YlGn",
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name="Unemployment Rate (%)",
).add_to(m)

folium.LayerControl().add_to(m)

m
```

Step 3

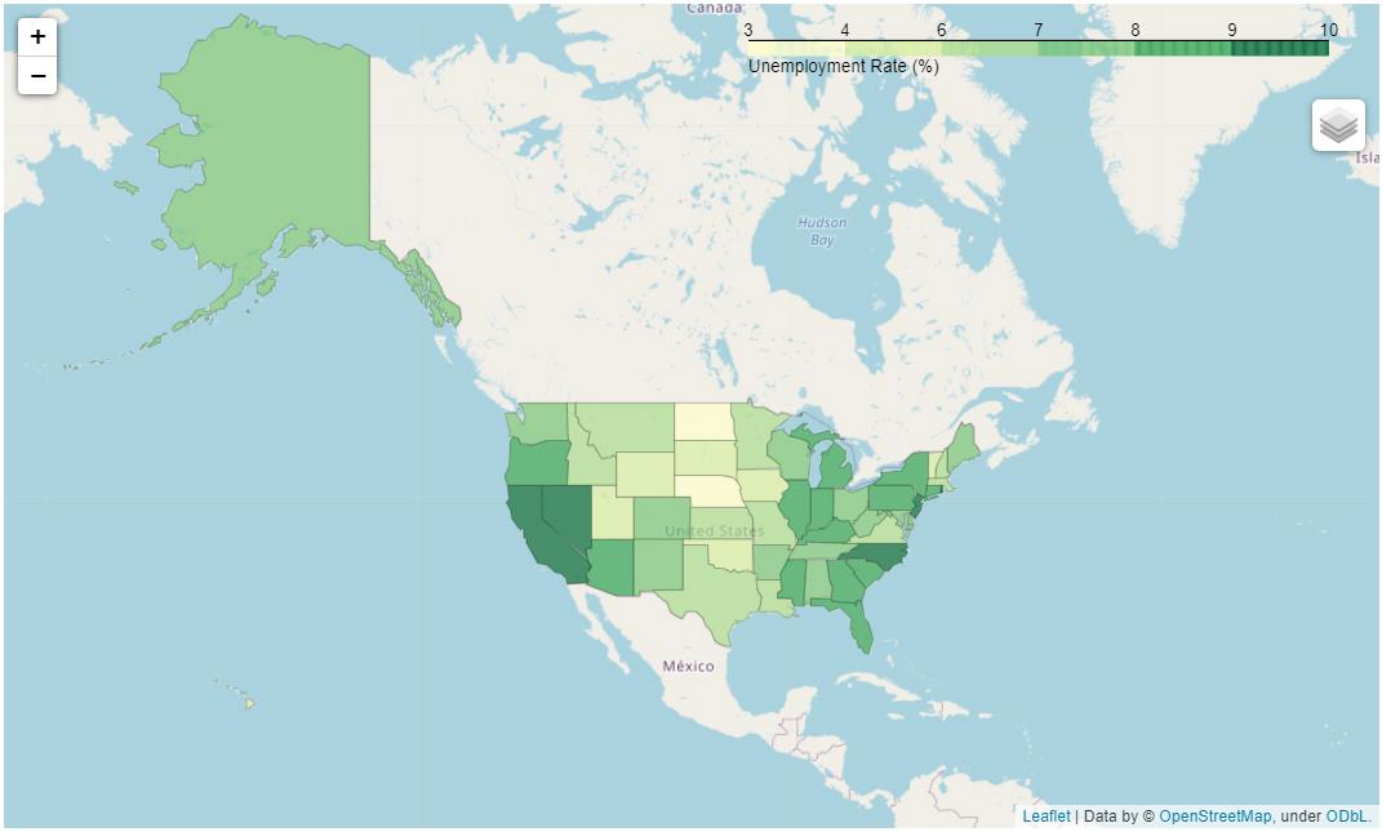
6) How to display a step-by-step diagram in the map area

Line 13, 15, 16, 17

- 13: geoJson data for the administrative district
- 15: Name of the data to be loaded into the administrative district
- 16: Column names and variables to use for this map in the dataframe
- 17: Matching between administrative districts in data and administrative districts in geojson

Step 3

6) How to display a step-by-step diagram in the map area



Step 4

- We can visualize the already processed data frame of the cumulative number of confirmed cases by 100 people in each country by applying the visualization method to the map learned in step 3.
- https://ourworldindata.org/covid-vaccination-global-projections

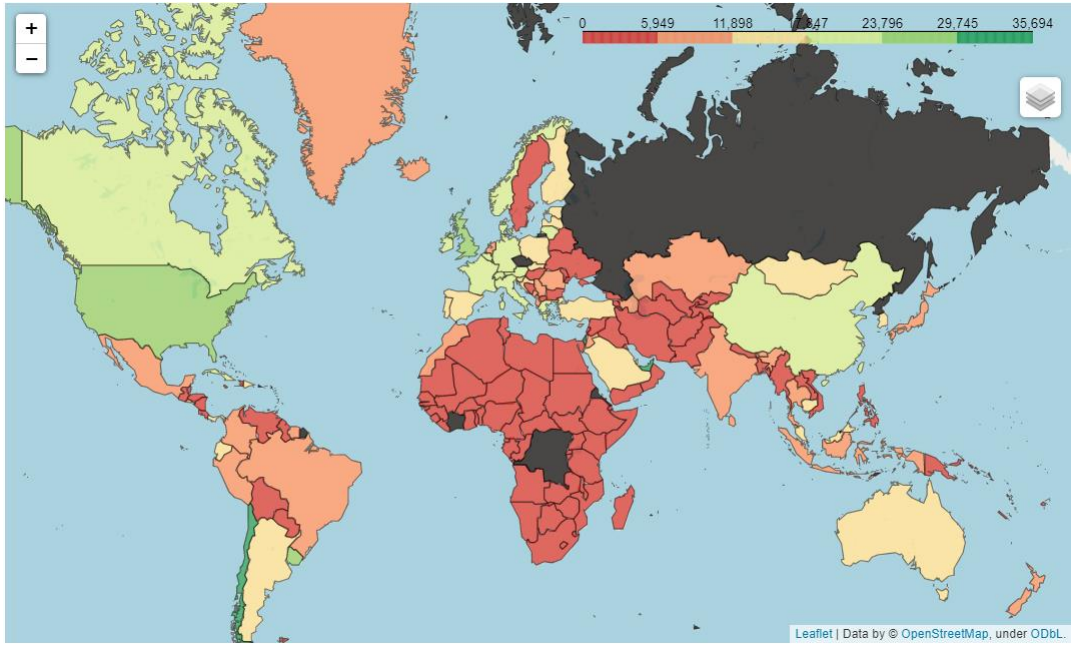
Step 4

```
import folium
import json

center = [35.762887375145795, 84.08313219586536]

m = folium.Map(location=center, zoom_start=2,
               max_bounds = True,
               min_zoom =1, min_lat = -84,
               max_lat = 84, min_lon =-175, max_lon = 187,
              )

geo_path = "./data//world_geojson/World_Countries__Generalized_.geojson"

json_data = json.load(open(geo_path), encoding='utf-8')

folium.Choropleth(geo_data = json_data,
                  data= total_df,
                  columns=(total_df.index,'total_vaccinations_per_hundred'),
                  key_on = 'properties.COUNTRY',
                  fill_color = 'RdYlGn',
                  fill_opacity = 0.7,
                  line_opacity = 0.5,
                 ).add_to(m)


folium.LayerControl().add_to(m)
```

```
<folium.map.LayerControl at 0x7fa3846d3d90>
```

Step 4

```
1 m
```



Line 1

- This is a viscoronaualization of the cumulative number of infected people by a group of 100 people from the time of the outbreak based on the current time.


End of
Document