p4

May 7, 2022

```
[19]: import folium
      import requests
      import pandas as pd
      import re
      from decimal import *
```

Read data from uploaded csv and create a dataframe. Create two empty columns to record latitude and longitude.

```
[20]: df = pd.read_csv("Maryland_Road_Closures.csv")
      df['latitude'] = 0
      df['longitude'] = 0
```

Then calculate the total number of rows.

I create a regular expression to extrack the latitude and longitude and assign them to the columns.

```
[21]: total_number = df[pd.notnull(df["county"])]["county"].count()
      string_pattern = r"\d{2}.\d+"
      regex_pattern = re.compile(string_pattern)
      for i in range(0,total_number):
          result = regex_pattern.findall(df['lat'][i])
          df.iat[i,11] = 0 - float(result[0])
          df.iat[i,10] = float(result[1])
```

Then I drop the original 'lat' and 'long' column to prevent misunderstanding.

```
[22]: df.drop('lat', axis=1,inplace = True)
      df.drop('long', axis=1,inplace = True)
```

Calculate to mean of latitude and longitude as the center of our map.

```
[23]: avg_latitude = Decimal("{:.2f}".format(df['latitude'].mean()))
      avg_longitude = Decimal("{:.2f}".format(df['longitude'].mean()))
      map_osm = folium.Map(location=[avg_latitude,avg_longitude], zoom_start=8)
      map_osm
```

```
[23]: <folium.folium.Map at 0xffff6ad74f40>
```

Then we differentiate the data by directions of closure: South-red, North-blue,East-green, West-purple and for more than one single direction-black.

```python
[24]: for i in range(0,total_number):
          if df.iat[i,2] == "South":
              folium.Marker(location=[df.iat[i,8], df.iat[i,9]], tooltip = 'South␣
          ↪direction of road is closed', popup = df['lanes'][i],
                            icon=folium.Icon(color='red')).add_to(map_osm)
          elif df.iat[i,2] == "North":
              folium.Marker(location=[df.iat[i,8], df.iat[i,9]], tooltip = 'North␣
          ↪direction of road is closed', popup = df['lanes'][i],
                            icon=folium.Icon(color='blue')).add_to(map_osm)
          elif df.iat[i,2] == "East":
              folium.Marker(location=[df.iat[i,8], df.iat[i,9]], tooltip = 'East␣
          ↪direction of road is closed', popup = df['lanes'][i],
                            icon=folium.Icon(color='green')).add_to(map_osm)
          elif df.iat[i,2] == "West":
              folium.Marker(location=[df.iat[i,8], df.iat[i,9]], tooltip = 'West␣
          ↪direction of road is closed', popup = df['lanes'][i],
                            icon=folium.Icon(color='purple')).add_to(map_osm)
          else :folium.Marker(location=[df.iat[i,8], df.iat[i,9]], tooltip = 'More␣
          ↪than one direction of road is closed', popup = df['lanes'][i],
                            icon=folium.Icon(color='black')).add_to(map_osm)

      map_osm
```

[24]: <folium.folium.Map at 0xffff6ad74f40>

In the map, Red Icon means South direction of road is closed, Purper Icon means West direction of road is closed, Blue Icon means North direction of road is closed, Green Icon means East direction of road is closed. Black icon means morethan one direction is closed. It can show the closed direction when moving mouse to the icon and show the detailed infomation when clicking the icon.

WriteUp:

Detailed code with explanation and output are shown above.

An explanation of your chosen dataset The dataset I choose is Maryland Road Closures Dataset. It is updated and provided by Maryland Department of Transportation Coordinated Highways Action Response Teamon May 2, 2022. It shows the closure of highways in Maryland state.

In this map, it shows the location of each read closure. Each different color represents different direction of each road closure. In this map, to prevent showing large amounts of output, it mainly focus on one single direction. From there, we can easily conclude that which highway has the most number of closures and which area has the most number of closures. Then we can foresee the traffic condiction of that place and make plans early.

[ ]:

[ ]: