

```
In [1]: import pandas as pd
import numpy as np
import os, shutil
import seaborn as sns
import matplotlib.pyplot as plt
import geopandas, geopy

import warnings
warnings.filterwarnings("ignore")

In [2]: # read the dataset to be used for the visualization
df = pd.read_csv("vehicle_fatality.csv")

In [3]: # check the data sample
df.sample(5)

Out[3]:
```

	NAMELSAD10	Fatality_alcoh	Fatality_rate_alcoh	Fatality_bike	Fatality_rate_bike	Fatality	Fatality_rate	Fatality_ped	Fatality_rate_ped
124	Seminole County	1	12.03	0	0.00	6	72.16	0	0.00
88	Liberty County	6	9.76	0	0.00	7	11.38	1	1.63
15	Bulloch County	1	1.29	1	1.29	8	10.35	1	1.29
80	Jefferson County	0	0.00	0	0.00	1	6.48	0	0.00
77	Jackson County	4	5.68	3	4.26	14	19.88	2	2.84

```
In [4]: # dimension
df.shape

Out[4]:
(159, 11)

In [5]: # check whether there are nulls
df.isna().sum()

Out[5]:
NAMELSAD10      0
Fatality_alcoh  0
Fatality_rate_alcoh  0
Fatality_bike    0
Fatality_rate_bike  0
Fatality         0
Fatality_rate    0
Fatality_ped     0
Fatality_rate_ped  0
Fatality_speed   0
Fatality_rate_speed  0
dtype: int64

In [6]: # check data information
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159 entries, 0 to 158
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   NAMELSAD10            159 non-null    object
 1   Fatality_alcoh         159 non-null    int64
 2   Fatality_rate_alcoh    159 non-null    float64
 3   Fatality_bike          159 non-null    int64
 4   Fatality_rate_bike     159 non-null    float64
 5   Fatality               159 non-null    int64
 6   Fatality_rate          159 non-null    float64
 7   Fatality_ped          159 non-null    int64
 8   Fatality_rate_ped      159 non-null    float64
 9   Fatality_speed         159 non-null    int64
10  Fatality_rate_speed    159 non-null    float64
dtypes: float64(5), int64(5), object(1)
memory usage: 13.8+ KB

In [7]: # check summary
df.describe().T

Out[7]:
```

	count	mean	std	min	25%	50%	75%	max
Fatality_alcoh	159.0	2.358491	4.378143	0.0	0.00	1.00	3.000	36.00
Fatality_rate_alcoh	159.0	6.031887	8.062437	0.0	0.00	3.83	7.385	62.19
Fatality_bike	159.0	0.968553	2.385137	0.0	0.00	0.00	1.000	21.00
Fatality_rate_bike	159.0	1.245157	2.208593	0.0	0.00	0.00	1.600	9.29
Fatality	159.0	9.459119	15.544034	0.0	3.00	5.00	10.000	130.00
Fatality_rate	159.0	23.954969	27.325604	0.0	11.82	19.88	30.240	310.95
Fatality_ped	159.0	1.641509	4.504966	0.0	0.00	0.00	1.000	36.00
Fatality_rate_ped	159.0	2.203208	3.681075	0.0	0.00	0.00	3.435	18.65
Fatality_speed	159.0	1.679245	3.471987	0.0	0.00	1.00	2.000	26.00
Fatality_rate_speed	159.0	3.459245	6.900631	0.0	0.00	0.85	4.165	62.19

```
In [ ]:
```

CHOREPLETH MAP VISUALIZATION

- In order to plot chorepleth, I will use `Counties_Georgia.shp` file that contains all Georgia's Counties information about the location.
- The file will help us in drawing the location's map.
- Once the file data is loaded, It will then Be merged with the data with other information on NamesAD10 as the key.
- This will give the final data for plotting

```
In [8]: # read counties location information
counties = geopandas.read_file("Counties_Georgia.shp")

In [9]: # merge the data on NAMELSAD10 of counties information and Fatalities data
data = counties.merge(df, left_on = 'NAMELSAD10', right_on = 'NAMELSAD10')

In [10]: # check head
data.head()

Out[10]:
```

	OBJECTID	STATEFP10	COUNTYFP10	GEOID10	NAME10	NAMELSAD10	totpop10	WFD	RDC_AAA	MNGWPD	...	Fatality_alcoh	Fatal
0	1	13	173	13173	Lanier	Lanier County	10078	N	N	N	...	0	
1	2	13	029	13029	Bryan	Bryan County	30233	N	N	N	...	1	
2	3	13	001	13001	Appling	Appling County	18236	N	N	N	...	3	
3	4	13	241	13241	Rabun	Rabun County	16276	N	N	N	...	1	
4	5	13	023	13023	Bleckley	Bleckley County	13063	N	N	N	...	0	

5 rows × 31 columns

```
In [11]: # check new columns
data.columns

Out[11]:
Index(['OBJECTID', 'STATEFP10', 'COUNTYFP10', 'GEOID10', 'NAME10', 'NAMELSAD10', 'totpop10', 'WFD', 'RDC_AAA', 'MNGWPD', 'HPO', 'MSA', 'F1HR_NA', 'F8HR_NA', 'Reg_Comm', 'Acres', 'Sq_Miles', 'Label', 'GlobalID', 'last_edit', 'geometry', 'Fatality_alcoh', 'Fatality_rate_alcoh', 'Fatality_bike', 'Fatality_rate_bike', 'Fatality', 'Fatality_rate', 'Fatality_ped', 'Fatality_rate_ped', 'Fatality_speed', 'Fatality_rate_speed'],
      dtype='object')

In [12]: # check coordinate information
data.crs

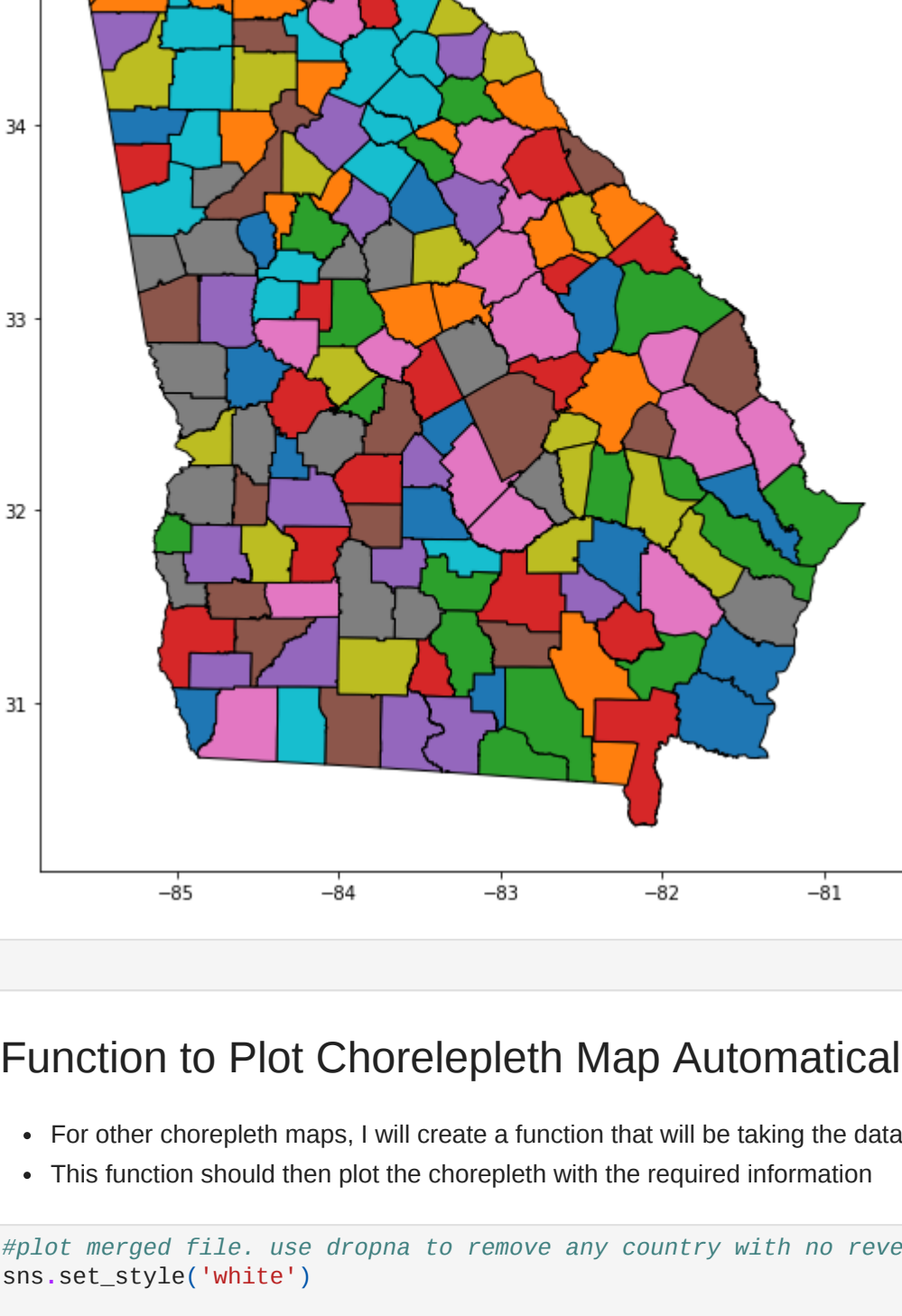
Out[12]:
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich

In [13]: # To make the map look a little more familiar lets reproject it's coordinates to Mercator.
# data = data.to_crs("EPSG:3395")
```

a. Create a map to visualize the counties with different colors, choropleth map

```
In [14]: ax= data.plot(cmap='tab10', figsize=(12, 10), edgecolor='black')
ax.set_title(f'Georgia Counties Map with different Colors', fontdict= {'fontsize':20, "fontweight":'bold'})

Out[14]: Text(0.5, 1.0, 'Georgia Counties Map with different Colors')
```



```
In [ ]:
```

Function to Plot Chorelepleth Map Automatical on a certain column Value

- For other chorepleth maps, I will create a function that will be taking the dataset and the column to be plotted.
- This function should then plot the chorepleth with the required information

```
In [15]: #plot merged file. use dropna to remove any country with no revenue value
sns.set_style('white')

# the function to plot choropleth map from dataset column provided
def draw_selected_col_choropleth(data, column , title, cmap = "OrRd", fontsize=25, scheme='quantiles'):
    """
    This function is used to plot chorepleth map and output it as a graph.
    The function receives the data and columns to be used.
    It also has other informations of the graph to be draw like color map, title, fontsize and scheme to be use
    """
    #if column passed in not in data, ignore drawing the graph.
    if column not in data.columns:
        print("No such Column in the dataset, Please check the column and plot again")
        return
    #plot the map with the column passed as the value
    ax = data.plot(column=f'{column}',
                    cmap =f'{cmap}',
                    figsize=(10,10),
                    legend=True,
                    k=10, #we divide the data into 10 groups and color them differently depending on their value
                    scheme=f'{scheme}',
                    edgecolor='black')

    #add title to the map
    ax.set_title(f'{title}', fontdict= {'fontsize':fontsize, "fontweight":'bold'})
    #move legend to an empty space and remove axis
    ax.set_axis_off()
    # set the legend at the left bottom
    ax.get_legend().set_bbox_to_anchor((-12,12))
    # ax.get_figure()
    return ax

In [ ]:
```

b. Create a choropleth map indicating the fatalities rate with different colors

```
In [16]: # plot the map for fatalities rates
draw_selected_col_choropleth(data, "Fatality_rate", "Fatality Rates of Georgia County")

Out[16]: <AxesSubplot:title={'center':'Fatality Rates of Georgia County'}>
```



```
In [ ]:
```

c. Create a choropleth map indicating alcohols related fatality in counties

```
In [17]: # plot the map for fatalities of alcohol
draw_selected_col_choropleth(data, "Fatality_alcoh", "Fatality of Alcohol of Georgia County", cmap="twilight_r")

Out[17]: <AxesSubplot:title={'center':'Fatality of Alcohol of Georgia County'}>
```



```
In [ ]:
```

d. Create a choropleth map indicating pedestrians' fatality in counties

```
In [18]: # plot the map for fatalities of pedestrian
draw_selected_col_choropleth(data, "Fatality_ped", "Pedestrian Fatality of Georgia County", cmap="cubehelix_r")

Out[18]: <AxesSubplot:title={'center':'Pedestrian Fatality of Georgia County'}>
```



```
In [ ]:
```

e. Create a choropleth map indicating bike fatality rate in counties

```
In [19]: # plot the map for fatalities rate for the bike
draw_selected_col_choropleth(data, "Fatality_rate_bike", "Bike's Fatality Rate of Georgia County", cmap="crest")

Out[19]: <AxesSubplot:title={'center':'Bike's Fatality Rate of Georgia County'}>
```



```
In [ ]:
```

f. Create a choropleth map indicating speeding fatality in counties

```
In [20]: # plot the map for fatalities rate for the bike
draw_selected_col_choropleth(data, "Fatality_speed", "Speeding Fatality of Georgia County", cmap="tab10")

Out[20]: <AxesSubplot:title={'center':'Speeding Fatality of Georgia County'}>
```



```
In [ ]:
```

```
In [21]: df.columns
```

```
Out[21]: Index(['NAMELSAD10', 'Fatality_alcoh', 'Fatality_rate_alcoh', 'Fatality_bike', 'Fatality_rate_bike', 'Fatality', 'Fatality_rate', 'Fatality_ped', 'Fatality_rate_ped', 'Fatality_speed', 'Fatality_rate_speed'],
      dtype='object')
```

```
In [ ]:
```