In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
import os
import folium
import json
```

In [2]:
```
# This command propts matplotlib visuals to appear in the notebook

%matplotlib inline
```

In [39]:
```
#Immport ".json" file for the US
country_geo = r'C:\Users\msyeu\Hate Crime Analysis\Data\Original Data\us-states.json'
```

In [40]:
```
country_geo
```

Out[40]:
```
'C:\\Users\\msyeu\\Hate Crime Analysis\\Data\\Original Data\\us-states.json'
```

In [41]:
```
#Defining path
path = r'C:\Users\msyeu\Hate Crime Analysis'
```

In [42]:
```
#Import Data
df = pd.read_csv(os.path.join(path, 'Data', 'Prepared Data', 'hate_crime_wrangled.csv'))
```

In [6]:
```
df.head()
```

Out[6]:

	Unnamed: 0	INCIDENT_ID	DATA_YEAR	ORI	PUB_AGENCY_NAME	AGENCY_TYPE_NAME	STATE_ABBR	STATE_NAME	DI
0	0	3015	1991	AR0040200	Rogers	City	AR	Arkansas	
1	1	3016	1991	AR0290100	Hope	City	AR	Arkansas	
2	2	43	1991	AR0350100	Pine Bluff	City	AR	Arkansas	
3	3	44	1991	AR0350100	Pine Bluff	City	AR	Arkansas	
4	4	3017	1991	AR0350100	Pine Bluff	City	AR	Arkansas	

5 rows × 23 columns

In [8]:
```
df=df.drop('Unnamed: 0', axis=1)
```

In [9]:
```
df.head()
```

Out[9]:

	INCIDENT_ID	DATA_YEAR	ORI	PUB_AGENCY_NAME	AGENCY_TYPE_NAME	STATE_ABBR	STATE_NAME	DIVISION_NAI
0	3015	1991	AR0040200	Rogers	City	AR	Arkansas	West Sou Cent
1	3016	1991	AR0290100	Hope	City	AR	Arkansas	West Sou Cent
2	43	1991	AR0350100	Pine Bluff	City	AR	Arkansas	West Sou Cent
3	44	1991	AR0350100	Pine Bluff	City	AR	Arkansas	West Sou Cent
4	3017	1991	AR0350100	Pine Bluff	City	AR	Arkansas	West Sou Cent

5 rows × 22 columns

In [10]:
```
df.shape
```

Out[10]:
```
(199797, 22)
```

2. Data Wrangling

In [13]:
```
#Select only the necessary columns and put them in a list called columns
columns = ['DATA_YEAR', 'STATE_ABBR', 'STATE_NAME', 'REGION_NAME', 'TOTAL_OFFENDER_COUNT', 'OFFENDER_RACE', 'VICTIM_C
```

In [14]:
```
#Create a subset
hate_crime_visual = df[columns]
```

In [15]:
```
hate_crime_visual.head()
```

Out[15]:

	DATA_YEAR	STATE_ABBR	STATE_NAME	REGION_NAME	TOTAL_OFFENDER_COUNT	OFFENDER_RACE	VICTIM_COUNT	
0	1991	AR	Arkansas	South	1	White	1	
1	1991	AR	Arkansas	South	1	Black or African American	1	
2	1991	AR	Arkansas	South	1	Black or African American	1	
3	1991	AR	Arkansas	South	1	Black or African American	2	Assault
4	1991	AR	Arkansas	South	1	Black or African American	1	

3. Conduct consistency checks.

In [16]:
```
# Check for missing values
hate_crime_visual.isnull().sum()
```

Out[16]:
```
DATA_YEAR                  0
STATE_ABBR                 0
STATE_NAME                 0
REGION_NAME                0
TOTAL_OFFENDER_COUNT       0
OFFENDER_RACE             19
VICTIM_COUNT               0
OFFENSE_NAME               0
TOTAL_INDIVIDUAL_VICTIMS   0
BIAS_DESC                  0
dtype: int64
```

In [17]:
```
hate_crime_visual['OFFENDER_RACE'].fillna('Unknown',inplace=True)
```

```
C:\Users\msyeu\AppData\Local\Temp\ipykernel_5888\958338699.py:1: FutureWarning: A value is trying to be set on a copy
of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which
we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or d
f[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  hate_crime_visual['OFFENDER_RACE'].fillna('Unknown',inplace=True)
C:\Users\msyeu\AppData\Local\Temp\ipykernel_5888\958338699.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  hate_crime_visual['OFFENDER_RACE'].fillna('Unknown',inplace=True)
```

In [18]:
```
# checking again for missing values
hate_crime_visual.isnull().sum()
```

Out[18]:
```
DATA_YEAR                   0
STATE_ABBR                  0
STATE_NAME                  0
REGION_NAME                 0
TOTAL_OFFENDER_COUNT        0
OFFENDER_RACE               0
VICTIM_COUNT                0
OFFENSE_NAME                0
TOTAL_INDIVIDUAL_VICTIMS    0
BIAS_DESC                   0
dtype: int64
```

In [19]:
```
# Checking for duplicates
dups = hate_crime_visual.duplicated()
```

In [20]:
```
dups.shape
```

Out[20]:
```
(199797,)
```

In [21]:
```
hate_crime_visuals = hate_crime_visual[(hate_crime_visual['BIAS_DESC'] == 'Anti-Black or African American')
         | (hate_crime_visual['BIAS_DESC'] == 'Anti-Jewish')
```

```
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-White')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-Gay (Male)')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-Hispanic or Latino')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-Other Race/Ethnicity/Ancestry')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-Lesbian, Gay, Bisexual, or Transgender (Mixed Group)')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-Asian')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-Lesbian (Female)')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-Islamic (Muslim)')
        | (hate_crime_visual['BIAS_DESC'] == 'Anti-American Indian/Alaska Native')]
```

In [22]:
```
hate_crime_visuals.head()
```

Out[22]:

	DATA_YEAR	STATE_ABBR	STATE_NAME	REGION_NAME	TOTAL_OFFENDER_COUNT	OFFENDER_RACE	VICTIM_COUNT	
0	1991	AR	Arkansas	South	1	White	1	
1	1991	AR	Arkansas	South	1	Black or African American	1	
2	1991	AR	Arkansas	South	1	Black or African American	1	
3	1991	AR	Arkansas	South	1	Black or African American	2	Assault
4	1991	AR	Arkansas	South	1	Black or African American	1	

In [23]:
```
def nb_check(abbr):
    if abbr == 'NB':
        return 'NE'
    else:
        return abbr
```

In [24]:
```
by_state = hate_crime_visuals
by_state['STATE_ABBR'] = by_state['STATE_ABBR'].apply(nb_check)
```

```
C:\Users\msyeu\AppData\Local\Temp\ipykernel_5888\2590119508.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  by_state['STATE_ABBR'] = by_state['STATE_ABBR'].apply(nb_check)
```

In [25]:
```
by_state[by_state['STATE_ABBR']=='NE']
```

Out[25]:

	DATA_YEAR	STATE_ABBR	STATE_NAME	REGION_NAME	TOTAL_OFFENDER_COUNT	OFFENDER_RACE	VICTIM_COUNT
38113	1996	NE	Nebraska	Midwest	1	White	1
38114	1996	NE	Nebraska	Midwest	1	Black or African American	1
38115	1996	NE	Nebraska	Midwest	1	Black or African American	1
46476	1997	NE	Nebraska	Midwest	0	Unknown	1
46477	1997	NE	Nebraska	Midwest	4	White	4
...	...	...	...	...	...	...	...
196704	2018	NE	Nebraska	Midwest	0	Unknown	2
196705	2018	NE	Nebraska	Midwest	0	Unknown	1
196706	2018	NE	Nebraska	Midwest	0	Unknown	1
196707	2018	NE	Nebraska	Midwest	0	Unknown	2
196711	2018	NE	Nebraska	Midwest	0	Unknown	1

868 rows × 10 columns

In [26]:
```
by_state = hate_crime_visuals.groupby('STATE_NAME').count()
```

In [27]:
```
by_state.head()
```

Out[27]:

STATE_NAME	DATA_YEAR	STATE_ABBR	REGION_NAME	TOTAL_OFFENDER_COUNT	OFFENDER_RACE	VICTIM_COUNT	OFFENSE_
Alabama	182	182	182	182	182	182	
Alaska	156	156	156	156	156	156	
Arizona	5502	5502	5502	5502	5502	5502	
Arkansas	913	913	913	913	913	913	
California	30823	30823	30823	30823	30823	30823	

In [28]:
```
by_state = by_state.reset_index()
```

Choropleth Map

In [29]:
```
# Create a data frame with just the states and the values for rating we want plotted

data_to_plot = by_state[['STATE_NAME','TOTAL_INDIVIDUAL_VICTIMS']]
data_to_plot.head()
```

Out[29]:

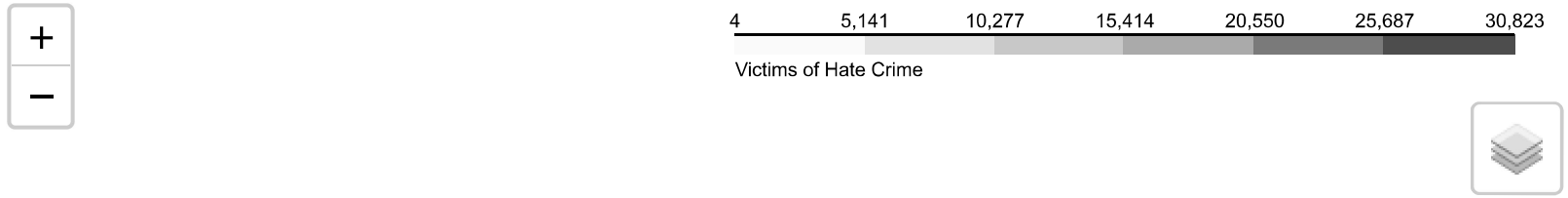
	STATE_NAME	TOTAL_INDIVIDUAL_VICTIMS
0	Alabama	182
1	Alaska	156
2	Arizona	5502
3	Arkansas	913
4	California	30823

In [43]:
```
# Setup a folium map at a high-level zoom
map = folium.Map(location = [100, 0], zoom_start = 1.5)

# Choropleth maps bind Pandas Data Frames and json geometries.This allows us to quickly visualize data combinations
folium.Choropleth(
    geo_data = country_geo,
    data = data_to_plot,
    columns = ['STATE_NAME', 'TOTAL_INDIVIDUAL_VICTIMS'],
    key_on = 'feature.properties.name',
    fill_color = 'YlOrBr', fill_opacity=0.6, line_opacity=0.1,
    legend_name = "Victims of Hate Crime").add_to(map)
folium.LayerControl().add_to(map)

map
```

Out[43]:



Above map answers one of my questions, which was 'Which states had the most hate crimes reported throughout 1991-2018?'

As we can see in the choropleth map above, CA has the highest number of hate crime victims. New York and New Jersey would be in the top 3 states with the highest number of hate crime victims.

In [44]:
```
hate_crime_visuals.to_csv(os.path.join(path, 'Data', 'Prepared Data', 'hate_crimes_wrangled.csv'))
```

In [ ]: