Fatal Police Shootings

The following notebook visualizes the Washington Post database of fatal police shootings. Beginning in 2015, the database has logged every fatal shooting by an on-duty police officer in the United States (https://www.washingtonpost.com/graphics/investigations/police-shootings-database/). The database contains information regarding the victim's age, race, gender and location, along with whether the victim was armed, and if they could be considered a threat to the officers or to the public at large.

This analysis is not meant to offer an opinion on the actions of the officers or the circumstances surrounding the fatal shooting, but rather to identify and investigate any trends or divergences. While much more detailed information and testimony is required to understand the situation surrounding each shooting, the following visualizations aim to summarize the dataset and inform the audience.

Importing relevent libraries & data

In [1]:
```python
# Packages / libraries
import numpy as np
import pandas as pd
from datetime import datetime
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```python
# load the washington post dataset via github
data_url = 'https://raw.githubusercontent.com/washingtonpost/data-police-shootings/master/fatal-police-shootings-data.csv'
raw_df = pd.read_csv(data_url)
raw_df.head()
```

Out[2]:

	id	name	date	manner_of_death	armed	age	gender	race	city	state	signs_of_mental_illness	threat_level	flee	bo
0	3	Tim Elliot	2015-01-02	shot	gun	53.0	M	A	Shelton	WA	True	attack	Not fleeing	
1	4	Lewis Lee Lembke	2015-01-02	shot	gun	47.0	M	W	Aloha	OR	False	attack	Not fleeing	
2	5	John Paul Quintero	2015-01-03	shot and Tasered	unarmed	23.0	M	H	Wichita	KS	False	other	Not fleeing	
3	8	Matthew Hoffman	2015-01-04	shot	toy weapon	32.0	M	W	San Francisco	CA	True	attack	Not fleeing	
4	9	Michael Rodriguez	2015-01-04	shot	nail gun	39.0	M	H	Evans	CO	False	attack	Not fleeing	

Data Wrangling

In [3]:
```python
# inspect shape of dataframe, columns, datatypes
print(raw_df.info())

# check null values
print('\n{}'.format(raw_df.isnull().sum()))

# inspect columns with missing data and check for data error
# raw_df[raw_df.race.isnull() == True]
# raw_df[raw_df.name.isnull() == True]
# raw_df.describe()
# etc...
```

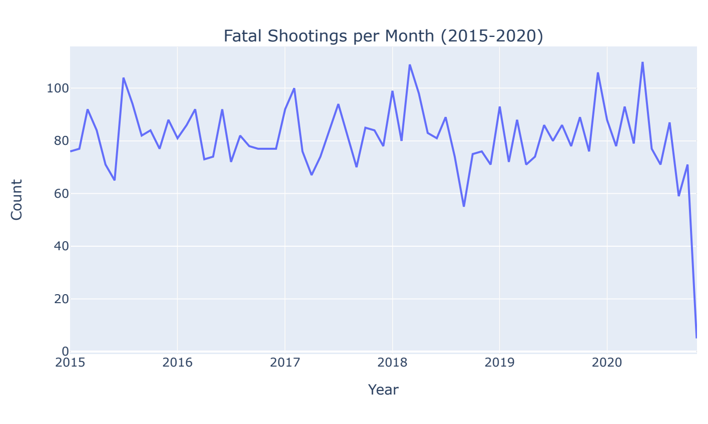
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5748 entries, 0 to 5747
Data columns (total 17 columns):
id                         5748 non-null int64
name                       5529 non-null object
date                       5748 non-null object
manner_of_death            5748 non-null object
armed                      5535 non-null object
age                        5484 non-null float64
gender                     5747 non-null object
race                       5125 non-null object
city                       5748 non-null object
state                      5748 non-null object
signs_of_mental_illness    5748 non-null bool
threat_level               5748 non-null object
flee                       5452 non-null object
body_camera                5748 non-null bool
longitude                  5466 non-null float64
latitude                   5466 non-null float64
is_geocoding_exact         5748 non-null bool
dtypes: bool(3), float64(3), int64(1), object(10)
memory usage: 645.7+ KB
None

id                           0
name                       219
date                         0
manner_of_death              0
armed                      213
age                        264
gender                       1
race                       623
city                         0
state                        0
signs_of_mental_illness      0
threat_level                 0
flee                       296
body_camera                  0
longitude                  282
latitude                   282
is_geocoding_exact           0
dtype: int64
```

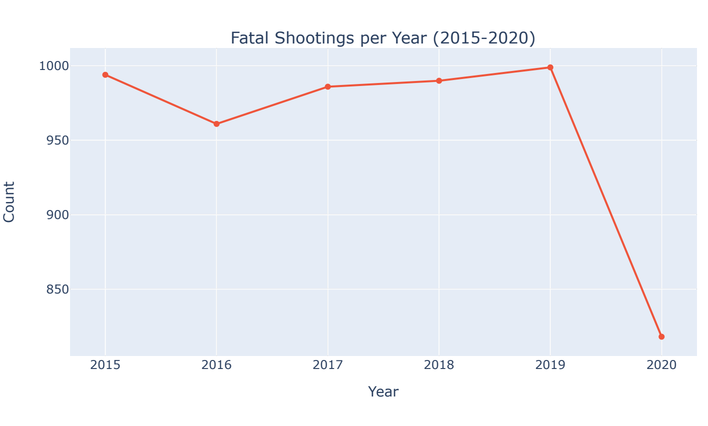
In [4]:
```python
# drop unnecessary columns from df
df = raw_df.copy()
df.drop(['id', 'name','is_geocoding_exact'], axis=1, inplace=True)
# df.head()

# Feature Engineering
# convert date column to datetime object and add additional datetime columns
df['date'] = pd.to_datetime(df['date'])
df['year'] = pd.to_datetime(df['date']).dt.year
df['month'] = pd.to_datetime(df['date']).dt.month
df['year_month'] = df['date'].dt.to_period('M')

# bin ages into decade age ranges
df['age_range'] = np.where(df['age'] < 18, '<18',
                           np.where((df['age'] > 17) & (df['age'] <= 30), '18-30',
                                    np.where((df['age'] > 30) & (df['age'] <= 40), '31-40',
                                             np.where((df['age'] > 40) & (df['age'] <= 50), '41-50',
                                                      np.where((df['age'] > 50), '50+', 'Not Specified'
)))))
# provide full race name
df['race'] = np.where(df['race'] == 'W', 'White',
                      np.where(df['race'] == 'B', 'Black',
                               np.where(df['race'] == 'N', 'Native American',
                                        np.where(df['race'] == 'H', 'Hispanic',
                                                 np.where(df['race'] == 'A', 'Asian',
                                                          np.where(df['race'] == 'O', 'Others', 'Not Sp
ecified'))))))

# create true/false is_armed column
df['is_armed'] = np.where(df['armed'] == 'unarmed', 'unarmed', 'armed')
```

Distribution by Month & Year

In [5]:
```python
# create monthly & yearly_shooting dfs with count of deaths
monthly_shootings = df.date.groupby(df['year_month']).agg(
    'count').to_frame(name="count").reset_index()
yearly_shootings = df.date.groupby(df['year']).agg(
    'count').to_frame(name="count").reset_index()
# monthly_shootings.head()
# yearly_shootings.head()
```

In [6]:
```python
# convert year_month into str series for plotly scatter plot
month_year = []
for i in monthly_shootings['year_month']:
    month_year.append(str(i))

# create scatter plots for monthly_shootings and yearly_shootings dfs
fig = make_subplots(rows=2, cols=1, subplot_titles=('Fatal Shootings per Month (2015-2020)',
                                                    'Fatal Shootings per Year (2015-2020)'))

fig.add_trace(go.Scatter(x=month_year,
                         y=monthly_shootings['count'],
                         mode='lines'), row=1, col=1)
fig.add_trace(go.Scatter(x=yearly_shootings['year'],
                         y=yearly_shootings['count'],
                         mode='lines+markers'), row=2, col=1)

fig.update_xaxes(title_text='Year')
fig.update_yaxes(title_text='Count')
fig.update_layout(showlegend=False, height=1000)
fig.show()

print('There have been {} fatal shootings since the start of 2015.'.format(raw_df.shape[0]))
```






```
There have been 5748 fatal shootings since the start of 2015.
```

Hover over scatter plot for numbers each month or year

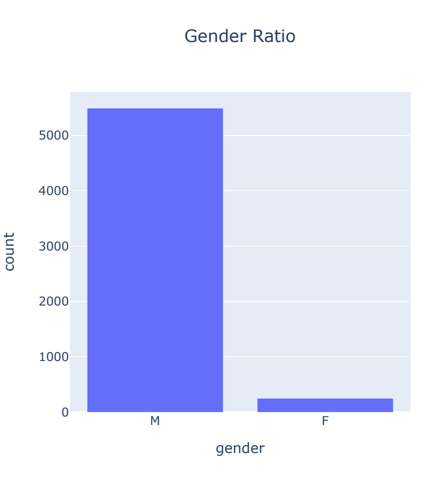
Observations:

Since January 2015, the average number of police shootings sits at ~80 deaths per month and the middle 50% of the data (25-75th percentile) lies between 76-88 deaths. The highest monthly total was 110 this past May and the low of 55 occured in September of 2018.

Looking at the yearly scatter plot, the previous 5 years had between 999 (2019) and 962 (2016) fatal shootings. Currently we are on pace for approximately 960 deaths in 2020.

Distribution by Gender

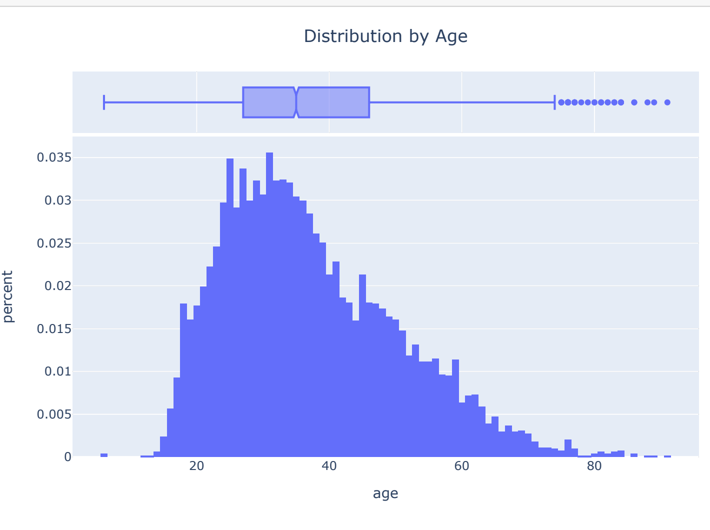
In [7]:
```python
# create df for victim counts by gender
df_gender = df['gender'].value_counts().reset_index().rename(
    columns={'index': 'gender', 'gender': 'count'})

# bar chart with count by gender
fig = go.Figure(go.Bar(x=df_gender['gender'], y=df_gender['count']))
fig.update_layout(title_text='Gender Ratio', title_x=0.5,
                  xaxis_title="gender", yaxis_title="count", height=500, width=500)
fig.show()

# create dfs for yearly shootings by each gender
by_gender = df.groupby(['year', 'gender']).agg(
    'count')['date'].to_frame(name='count').reset_index()
by_gender_male = by_gender[by_gender['gender'] == 'M']
by_gender_female = by_gender[by_gender['gender'] == 'F']

# print the percentages of shootings by gender
print('Percentage of shooting victims by gender: \n{}'.format(
    df['gender'].value_counts(normalize=True)))
print('\nMen are approximately {}x more likely to be killed than women.'.format(
    int(round(by_gender_male['count'].sum() / by_gender_female['count'].sum()))))
```




```
Percentage of shooting victims by gender:
M    0.956151
F    0.043849
Name: gender, dtype: float64

Men are approximately 22x more likely to be killed than women.
```

Distribution by Age

In [8]:
```python
# plot the distribution of victims by age

fig = px.histogram(df, x='age', histnorm='probability density', marginal='box')
fig.update_layout(title="Distribution by Age",
                  title_x=0.5, yaxis_title_text='percent')
fig.show()
```



Hover over histogram for percent of victims by age

Observations:

- The median age of victims is 35 years with a slightly higher average of ~37
- The ages of the victims range from 6 to 91 years and the middle 50% of victims range from 27-46 years old
- The distribution is right-skewed with numerous outliers greater than the upper fence of 74 years

Was The Victim a Threat?

In [9]:
```python
# df with the threat the victim posed to the police
df_threat = df['threat_level'].value_counts(normalize=True).to_frame(
    name='percent').reset_index().rename(columns={'index': 'threat'})
df_threat['percent'] = (df_threat['percent']*100).round(2)
df_threat['count'] = df['threat_level'].value_counts(
).reset_index().iloc[:, -1:]

# bar chart with counts and percentage of each threat_level
fig = px.bar(df_threat, x='threat', y='count', color='threat',
             text='count', hover_data=['percent', 'count'])
fig.update_traces(textposition='outside')
fig.update_layout(title='Threat Level', title_x=0.5,
                  height=500, width=600, showlegend=False)
fig.show()
```

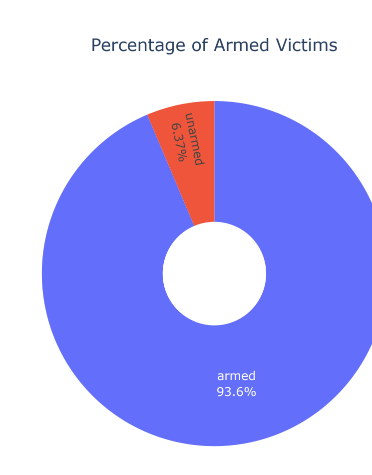
Hover over bars for percentages

Observations:

- Approximately 65% of victims were attacking the police when they were shot
- The 'other' and 'undetermined' groups above will be considered non-attacking and represent the remainin 35%

Was The Victim Armed?

In [10]:
```python
# df with count of armed and unarmed victims
df_armed = df['is_armed'].value_counts().to_frame(
    name='count').reset_index().rename(columns={'index': 'armed'})

# Pie chart with counts and percentages for armed vs unarmed victims
fig = go.Figure(data=[go.Pie(labels=df_armed['armed'],
                             values=df_armed['count'], hole=0.3, textinfo='label+percent')])
fig.update_layout(title='Percentage of Armed Victims',
                  title_x=0.5, showlegend=False)
fig.show()
```



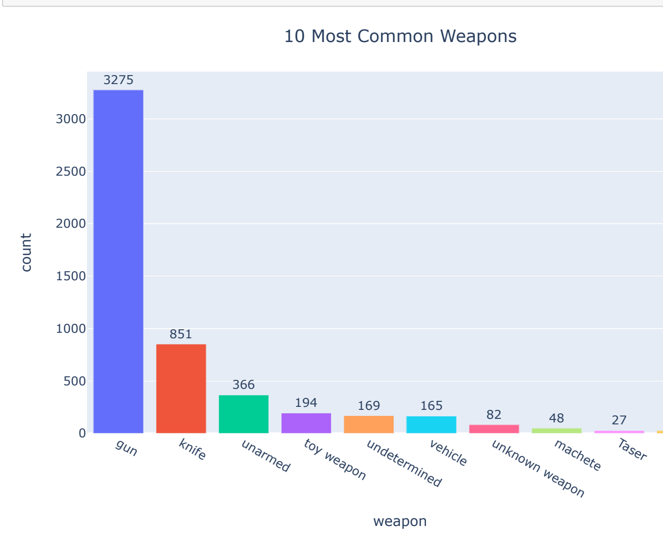
Hover for count

Observations:

- Almost 94% of victims were armed with a weapon at the time of the shooting

The Most Frequently Used Weapons

In [11]:
```python
# df with 10 most frequent weapons possessed by victim
df_weapons = df['armed'].value_counts(normalize=True).to_frame(name='percent').reset_index().rename(
    columns={'index': 'weapon'})[:10]
df_weapons['percent'] = (df_weapons['percent']*100).round(2)
df_weapons['count'] = df['armed'].value_counts().reset_index().iloc[:10, -1:]

# bar chart with counts and percentages of weapons
fig = px.bar(df_weapons, x='weapon', y='count',
             color='weapon', text='count',
             hover_data=['percent', 'count'])
fig.update_traces(textposition='outside')
fig.update_layout(showlegend=False,
                  title='10 Most Common Weapons', title_x=0.5)
fig.show()
```



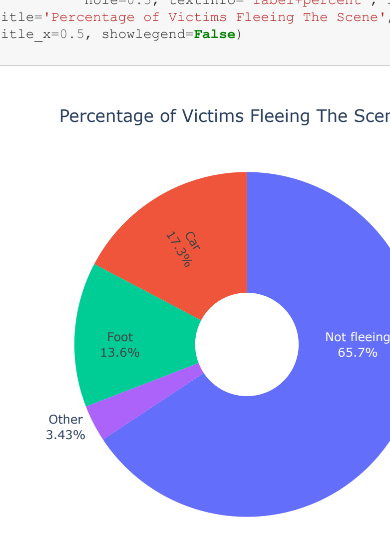
Hover over bars for the percentage of victims armed with each weapon

Observations:

- Approximately 75% of victims possessed a gun (60%) or knife (15%)
- Less than 7% of victims were unarmed at the time of the shooting

Was The Victim Fleeing?

In [12]:
```python
# counts of if and how the victim fled police
df_flee = df['flee'].value_counts().to_frame(
    name='count').reset_index().rename(columns={'index': 'flee'})

# pie chart with counts and percentages of how the victim fled
fig = go.Figure(data=[go.Pie(labels=df_flee['flee'], values=df_flee['count'],
                             hole=0.3, textinfo='label+percent', insidetextorientation='radial')])
fig.update_layout(title='Percentage of Victims Fleeing The Scene',
                  title_x=0.5, showlegend=False)
fig.show()
```



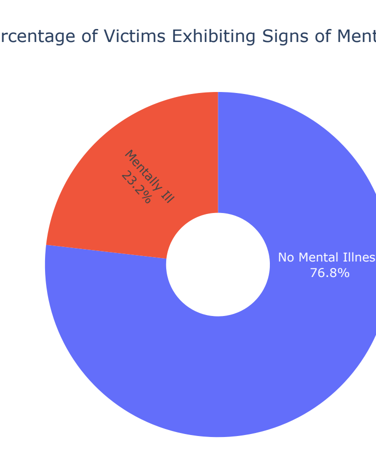
Hover for counts

Observations:

- 66% of victims were not fleeing the police compared to 34% fleeing via various methods

Did The Victim Exhibit Signs of Mental Illness?

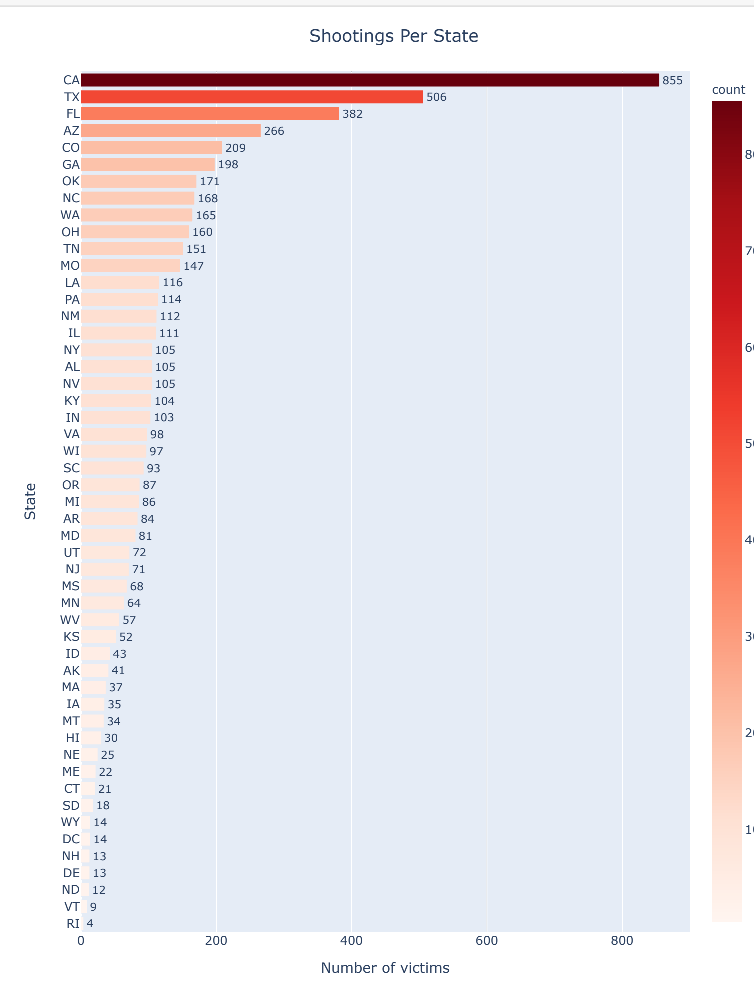
In [13]:
```python
# did the victims exhibit signs of mental illness
df_illness = df['signs_of_mental_illness'].value_counts().to_frame(name='count').reset_index().rename(
    columns={'index': 'mental_illness'})
df_illness['mental_illness'] = np.where(
    df_illness['mental_illness'] == True, 'Mentally Ill', 'No Mental Illness')

# pie chart with counts and percentages of mental illness in victims
fig = go.Figure(data=[go.Pie(labels=df_illness['mental_illness'], values=df_illness['count'],
                             hole=0.3, textinfo='label+percent', insidetextorientation='radial')])
fig.update_layout(title='Percentage of Victims Exhibiting Signs of Mental Illness',
                  title_x=0.5, showlegend=False)
fig.show()
```



Hover for counts

Distribution by State

In [14]:
```python
# victim counts per state
df_state = df['state'].value_counts().to_frame(name='count').reset_index().rename(
    columns={'index': 'state'}).sort_values('count')
# calculate percentage of total victims by state
df_state['percent'] = round(
    (df_state['count']/df_state['count'].sum())*100, 2).astype(str)

# horizontal bar chart of victims per state in descending order
fig = px.bar(data_frame=df_state, x='count', y='state', orientation='h',
             color='count', color_continuous_scale='reds', text='count',
             hover_data=['percent'])
fig.update_layout(title_text='Shootings Per State', yaxis_title='State',
                  xaxis_title='Number of victims', title_x=0.5, height=1000)
fig.update_traces(textposition='outside')
fig.show()

print('The average number of victims per state is {}'.format(
    round(df_state['count'].mean())))
print('The median number of victims per state is {}'.format(
    df_state['count'].median()))
```




```
The average number of victims per state is 113
The median number of victims per state is 86.0
```

Hover for percentages

Observations:

- California, Texas & Florida account for 30% of all fatal shootings by police
- The southwest is home to several states with the highest victim counts, while the northeast is witness to far fewer fatalities

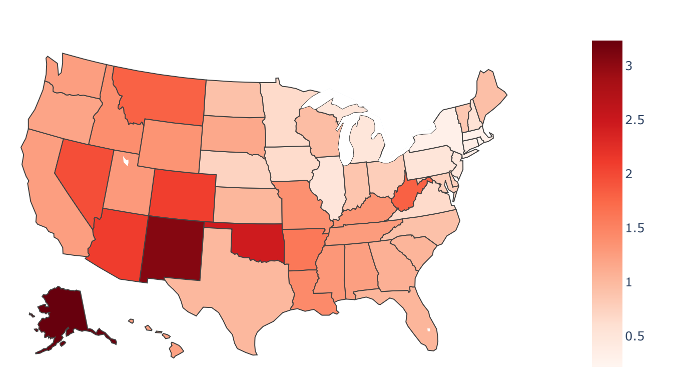
Mapping of Victims by State

In [15]:
```python
# load population data from census.gov
# percent of US population by state
df_map = pd.read_csv('C:/Users/garcr/Desktop/state_populations.csv')

# grab percent of US shootings by state
df_map['shot_pct'] = df_state.sort_values(
    'state').reset_index(drop=True)['percent']
df_map['shot_pct'] = pd.to_numeric(df_map['shot_pct'])

# calculate the relative frequency ratio of shootings to population for each state
df_map['diff'] = round(df_map['shot_pct']/df_map['pop_pct'], 2)
df_map.sort_values('diff', ascending=False).head() # sorted by states with highest relative frequency o
f shootings
```

Out[15]:

	state	pop_pct	shot_pct	diff
0	AK	0.22	0.71	3.23
32	NM	0.64	1.95	3.05
36	OK	1.21	2.97	2.45
3	AZ	2.22	4.63	2.09
5	CO	1.75	3.64	2.08

In [16]:
```python
# choropleth map of victims per state
fig = go.Figure(data=go.Choropleth(locations=df_map['state'], z=df_map['diff'],
                                   locationmode='USA-states', colorscale='reds'))
fig.update_layout(geo_scope='usa')
fig.show()
```



Hover for the relative frequency ratio of shootings to population for each state

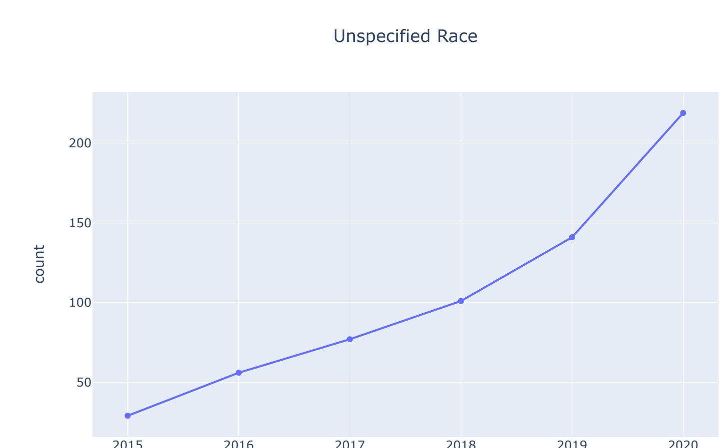
Observations:

- A relative frequency ratio of 1 indicates that the state has experienced an expected amount of shootings given its population
- Alaska and New Mexico have ratios over 3, indicating that they experience 3x more shootings than expected given their populations (>1)
- A majority of midwestern and northeastern states witness fewer shootings with respect to their populations (<1)
- Numerous southwestern states are experiencing 2x as many fatal shootings as expected (AZ,CO,NM,NV,OK)

Distribution by Race

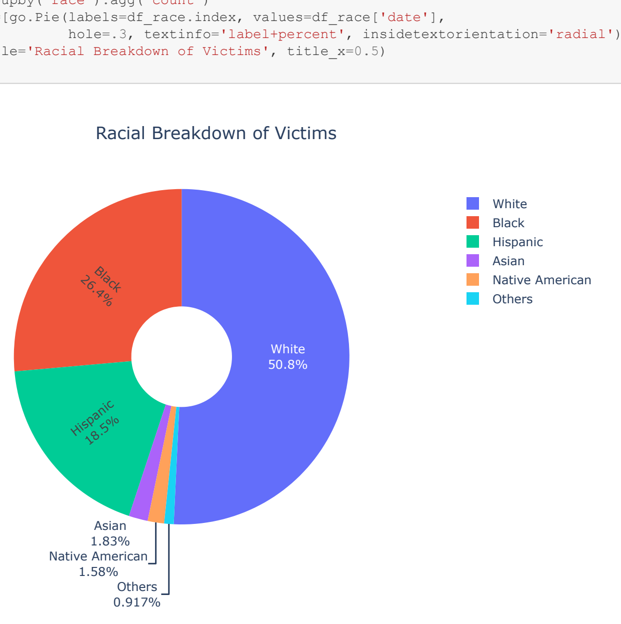
Before proceeding with the racial breakdown of the dataset, we need to address the 617 victims with an unspecified race.

```
In [17]: unspecified = df[df['race']=='Not Specified'].groupby('year').agg('count')['date'].to_frame(name='count').reset_index()

fig = go.Figure(go.Scatter(x=unspecified['year'],
                y=unspecified['count'],
                mode='lines+markers'))
fig.update_layout(title='Unspecified Race', title_x=0.5,
                  xaxis_title_text='year', yaxis_title_text='count')
fig.show()
```



Victims with an unspecified race have increased year over year since 2015. Potential explainations for the yearly increase are ongoing investigations where personal information is not released to the public and police departments retroactively enter the victim's personal information at a later date. Whatever the reasoning, unspecified victims will not be included in the visualizations to follow.

```
In [18]: # remove null race values (previously replaced with 'Not Specified')
df_race = df[df['race'] != 'Not Specified']

# groupby race and create a pie chart for racial distribution of victims
df_race = df_race.groupby('race').agg('count')
fig = go.Figure(data=[go.Pie(labels=df_race.index, values=df_race['date'],
                             hole=.3, textinfo='label+percent', insidetextorientation='radial')])
fig.update_layout(title='Racial Breakdown of Victims', title_x=0.5)
fig.show()
```



Hover for the number of victims per race

Observations:

The US racial demographic is approximately 60% White, 18.5% Hispanic, 13.5% Black, 6% Asian, 1.5% Native American, and 0.5% Other (https://www.census.gov/quickfacts/fact/table/US/PST045219). The purpose of these numbers is to compare the racial makeup of the victims to the racial makeup of the US. We will assume the probability of interacting with the police occurs at the same rate as the racial breakdown written above. Therefore we would assume, all things being equal, that white people would make up ~60% of all police encounters and ~60% of fatal shootings by police.

- Black Americans and 'Others' were killed at nearly twice their expected rates
- White and Asian Americans were killed significantly less than expected- approximately -10% and -4% respectively

Distribution by Age Range & Race

```
In [19]: # drop any victims with null values for age and race
df_age = df[(df['age_range'] != 'Not Specified')
            & (df['race'] != 'Not Specified')]

# create stacked bar chart of victims by race for each age_range
fig = px.histogram(df_age, x='age_range', color='race',
                   title='Racial Breakdown By Age Bracket')
fig.update_layout(title_x=0.5)
fig.show()
```

Hover for racial victim count per age bracket

Observations:

- Black Americans are killed at the highest rate over expectation in the <18, 18-30, 31-40, and 41-50 age brackets at +25%, +22%, +12%, and +6% respectively
- In the 50+ age bracket, White Americans were killed at nearly 15% over expectation

Pivot Table of Age vs Race

```
In [20]: # create df filtering out the formerly null values ('Not Specified') in race and age_range
race_age = df[(df['race'] != "Not Specified") & (
    df['age_range'] != "Not Specified")][['race', 'age_range', 'date']]

# count the number of victims grouped by age_range and race
race_age = race_age.groupby(['age_range', 'race']).agg('count').reset_index()

# pivot the data
race_age = race_age.pivot(index="age_range", columns="race")

# flatten and fix pivot table column names
race_age = pd.DataFrame(race_age.to_records()).rename(
    columns={"age_range": "Age", "('date', 'Asian')": "Asian", "('date', 'Black')": "Black",
             "('date', 'Hispanic')": "Hispanic", "('date', 'Native American')": "Native American",
             "('date', 'Others')": "Others", "('date', 'White')": "White"})

# sum victims by race and age
race_age['Total'] = race_age.sum(axis=1)
race_age = race_age.append(race_age.agg(['sum']))
race_age['Age'] = ['18-30', '31-40', '41-50', '50+', '<18', 'Total']

# create plotly pivot table
fig = go.Figure(data=[go.Table(
    header=dict(values=list(race_age.columns),
                fill_color=['slateblue', 'crimson', 'mediumblue',
                            'lightgreen', 'orange', 'plum', 'deepskyblue', 'slateblue'],
                line_color='black', align='left', font=dict(color='white', size=14)),
    cells=dict(values=[race_age.Age, race_age['Black'], race_age.Asian, race_age.Hispanic,
                       race_age['Native American'], race_age.Others, race_age.White,
                       race_age.Total], line_color='black', font=dict(color='black', size=13), align='left'))])

fig.update_layout(title_text='Age vs Race', title_x=0.5)
fig.show()
```

Age vs Race

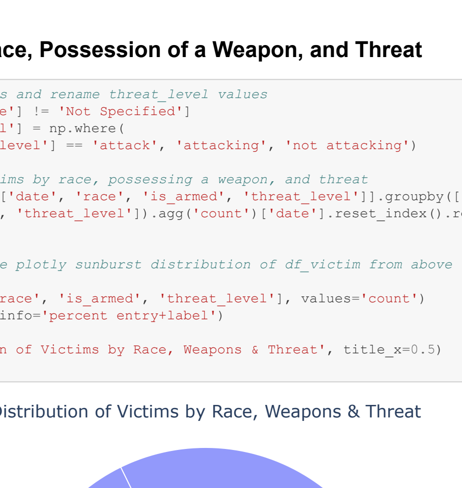
Age	Asian	Black	Hispanic	Native American	Others	White	Total
18-30	626	26	382	37	26	668	1765
31-40	382	30	307	29	10	732	1490
41-50	177	17	145	10	6	556	911
50+	106	14	67	3	4	574	768
<18	38	3	24	1	null	33	99
Total	1329	90	925	80	46	2563	5033

Observations:

- The 18-30 and 31-40 age brackets are the most common among victims of all races

Distribution by Race, Possession of a Weapon, and Threat

```
In [21]: # drop null race values and rename threat_level values
df_victim = df[df['race'] != 'Not Specified']
df_victim['threat_level'] = np.where(
    df_victim['threat_level'] == 'attack', 'attacking', 'not attacking')

# count and group victims by race, possessing a weapon, and threat
df_victim = df_victim[['date', 'race', 'is_armed', 'threat_level']].groupby([
    'race', 'is_armed', 'threat_level']).agg('count')['date'].reset_index().rename(columns={'date': 'count'})

# create an interactive plotly sunburst distribution of df_victim from above
fig = px.sunburst(
    df_victim, path=['race', 'is_armed', 'threat_level'], values='count')
fig.update_traces(textinfo='percent entry+label')
fig.update_layout(
    title='Distribution of Victims by Race, Weapons & Threat', title_x=0.5)
fig.show()
```



Click pie chart to slice by variable and hover to view label and count

Observations:

- At least 89% of victims from each race were armed
- Between 3-6% of victims were unarmed and not attacking- depending on race
- Black Americans and 'Other' victims were 3-5% more likely to be unarmed at the time of shooting
- Black, White, and 'Other' Americans were considered to be attacking officers at the highest rate (67%, 66%, 66%)

Yearly Distribution by Race

```
In [22]: # drop null values and group victims by year and race
year_race = df[df['race'] != 'Not Specified']
year_race = year_race.groupby(['year', 'race']).agg(
    'count')['date'].to_frame(name='count').reset_index()

# filter df by each race for bar chart below
year_race_asian = year_race.loc[year_race['race'] == 'Asian']
year_race_black = year_race.loc[year_race['race'] == 'Black']
year_race_hispanic = year_race.loc[year_race['race'] == 'Hispanic']
year_race_native_american = year_race.loc[year_race['race'] == 'Native American']
year_race_others = year_race.loc[year_race['race'] == 'Others']
year_race_white = year_race.loc[year_race['race'] == 'White']

# create plotly bars by year and race
asian = go.Bar(x=year_race_asian['year'],
               y=year_race_asian['count'], name='asian')
black = go.Bar(x=year_race_black['year'],
               y=year_race_black['count'], name='black')
hispanic = go.Bar(x=year_race_hispanic['year'],
                  y=year_race_hispanic['count'], name='hispanic')
native_american = go.Bar(x=year_race_native_american['year'],
                         y=year_race_native_american['count'], name='native_american')
others = go.Bar(x=year_race_others['year'],
                y=year_race_others['count'], name='others')
white = go.Bar(x=year_race_white['year'],
               y=year_race_white['count'], name='white')
races = [asian, black, hispanic, native_american, others, white]

# bar chart of victims by race per year
fig = go.Figure(races)
fig.update_layout(title='Yearly Distribution by Race', title_x=0.5,
                  xaxis_title_text='year', yaxis_title_text='count')
fig.show()


# create a pivot table with the year_race df from above
pivot = year_race.pivot(index='year',columns='race',values='count')
cols = pivot.columns
# divide each races yearly count by the yearly total to find the yearly percentages by race
pivot = pivot[cols].div(pivot[cols].sum(axis=1),axis=0).multiply(100).round(2)
pivot = pd.DataFrame(pivot.to_records()).set_index('year') # flatten and set index
pivot
```

Out[22]:

year	Asian	Black	Hispanic	Native American	Others	White
2015	1.45	26.74	17.82	0.93	1.45	51.61
2016	1.66	25.86	17.68	1.77	1.22	51.82
2017	1.76	24.64	19.80	2.42	0.66	50.72
2018	2.25	25.76	18.56	1.69	0.45	51.29
2019	2.21	29.14	19.00	1.52	1.05	47.09
2020	1.67	26.38	18.20	1.00	0.50	52.25

Hover to view yearly counts by race

Observations:

- 2020 is expected to have the fewest fatal shootings among each race
- 2020 victims by race: White 52%, Black 26%, Hispanic 18%, Asian 1.7%, Native American 1%, and Others 0.5%
  - In comparison to the US racial demographic detailed above, White Americans fell 8% below expectation, Black Americans were killed 13% more often than expected, and Hispanic, Asian, Native, and Other Americans fell at or near expectation
- Since 2015, shootings have decreased by approximately: 26% for Black, 7% for Asian, 24% for Hispanic, 27% for Native, 71% for Others, 21% for White Americans (based on 2020 projections)
  - However, the racial makeup of victims has remained stable throughout the years
- Prior to 2020 White American victims had decreased by nearly 20% from a height of 498 in 2015, while no other race witnessed sustained improvement
- The rate of non-white victims has hovered around 50% since the database's inception, although it has improved roughly 5% from 53%

Unarmed Yearly Distribution by Race

```
In [23]: # drop null values and group victims by year and race
unarmed = df[(df['race'] != 'Not Specified') & (df['is_armed'] == 'unarmed')]
unarmed = unarmed.groupby(['year', 'race']).agg(
    'count')['date'].to_frame(name='count').reset_index()

# filter df by each race for bar chart below
unarmed_asian = unarmed.loc[unarmed['race'] == 'Asian']
unarmed_black = unarmed.loc[unarmed['race'] == 'Black']
unarmed_hispanic = unarmed.loc[unarmed['race'] == 'Hispanic']
unarmed_native_american = unarmed.loc[unarmed['race'] == 'Native American']
unarmed_others = unarmed.loc[unarmed['race'] == 'Others']
unarmed_white = unarmed.loc[unarmed['race'] == 'White']

# create plotly bars by year and race
asian = go.Bar(x=unarmed_asian['year'],
               y=unarmed_asian['count'], name='asian')
black = go.Bar(x=unarmed_black['year'],
               y=unarmed_black['count'], name='black')
hispanic = go.Bar(x=unarmed_hispanic['year'],
                  y=unarmed_hispanic['count'], name='hispanic')
native_american = go.Bar(x=unarmed_native_american['year'],
                         y=unarmed_native_american['count'], name='native_american')
others = go.Bar(x=unarmed_others['year'],
                y=unarmed_others['count'], name='others')
white = go.Bar(x=unarmed_white['year'],
               y=unarmed_white['count'], name='white')
races = [asian, black, hispanic, native_american, others, white]

# bar chart of victims by race per year
fig = go.Figure(races)
fig.update_layout(title='Unarmed Yearly Distribution by Race', title_x=0.5,
                  xaxis_title_text='year', yaxis_title_text='count')
fig.show()


# create a pivot table with the year_race df from above
pivot = unarmed.pivot(index='year',columns='race',values='count')
cols = pivot.columns
# divide each races yearly count by the yearly total to find the yearly percentages by race
pivot = pivot[cols].div(pivot[cols].sum(axis=1),axis=0).multiply(100).round(2)
pivot = pd.DataFrame(pivot.to_records()).set_index('year') # flatten and set index
pivot.fillna(0)
```

Out[23]:

year	Asian	Black	Hispanic	Native American	Others	White
2015	0.00	40.43	20.21	1.06	4.26	34.04
2016	1.75	33.33	21.05	1.75	0.00	42.11
2017	1.43	31.43	18.57	2.86	1.43	44.29
2018	1.75	40.35	14.04	0.00	0.00	43.86
2019	7.27	25.45	20.00	1.82	0.00	45.45
2020	3.57	35.71	14.29	0.00	0.00	46.43

Hover to view yearly counts by race

Observations:

- There has been a 60% decrease in unarmed victims from 94 in 2015, to an estimated 37 unarmed victims EOY 2020 (including unspecified race victims)
- Since 2015 unarmed victim counts have decreased by approximately: 68% for Black, 79% for Hispanic, and 50% for White Americans (based on 2020 projections)
  - Over that same period, the yearly rates of unarmed victims have slightly improved for Black and Hispanic Americans down ~5% and ~6% respectively, while White Americans increased from 34% of victims in 2015 to 46% in 2020
- Since 2015, unarmed Black Americans have been killed at a substantially higher rate than expected at ~35% of the total
  - Meanwhile unarmed White victims make up ~42% of the total- roughly 18% below demographic since 2015

Unarmed & Non-violent Yearly Distribution by Race

```
In [24]: # drop null values and group victims by year and race
non_violent = df[(df['race'] != 'Not Specified') & (
    df['is_armed'] == 'unarmed') & (df['threat_level'] != 'attack')]
non_violent = non_violent.groupby(['year', 'race']).agg(
    'count')['date'].to_frame(name='count').reset_index()

# filter df by each race for bar chart below
non_violent_asian = non_violent.loc[non_violent['race'] == 'Asian']
non_violent_black = non_violent.loc[non_violent['race'] == 'Black']
non_violent_hispanic = non_violent.loc[non_violent['race'] == 'Hispanic']
non_violent_native_american = non_violent.loc[non_violent['race'] == 'Native American']
non_violent_others = non_violent.loc[non_violent['race'] == 'Others']
non_violent_white = non_violent.loc[non_violent['race'] == 'White']

# create plotly bars by year and race
asian = go.Bar(x=non_violent_asian['year'],
               y=non_violent_asian['count'], name='asian')
black = go.Bar(x=non_violent_black['year'],
               y=non_violent_black['count'], name='black')
hispanic = go.Bar(x=non_violent_hispanic['year'],
                  y=non_violent_hispanic['count'], name='hispanic')
native_american = go.Bar(x=non_violent_native_american['year'],
                         y=non_violent_native_american['count'], name='native_american')
others = go.Bar(x=non_violent_others['year'],
                y=non_violent_others['count'], name='others')
white = go.Bar(x=non_violent_white['year'],
               y=non_violent_white['count'], name='white')
races = [asian, black, hispanic, native_american, others, white]

# bar chart of victims by race per year
fig = go.Figure(races)
fig.update_layout(title='Unarmed & Non-violent Yearly Distribution by Race', title_x=0.5,
                  xaxis_title_text='year', yaxis_title_text='count')
fig.show()


# create a pivot table with the year_race df from above
pivot = non_violent.pivot(index='year',columns='race',values='count')
cols = pivot.columns
# divide each races yearly count by the yearly total to find the yearly percentages by race
pivot = pivot[cols].div(pivot[cols].sum(axis=1),axis=0).multiply(100).round(2)
pivot = pd.DataFrame(pivot.to_records()).set_index('year') # flatten and set index
pivot.fillna(0)
```


Out[24]:

year	Asian	Black	Hispanic	Native American	Others	White
2015	0.00	38.98	23.73	1.69	5.08	30.51
2016	2.86	31.43	22.86	2.86	0.00	40.00
2017	2.38	26.19	26.19	2.38	0.00	42.86
2018	2.94	41.18	14.71	0.00	0.00	41.18
2019	8.33	27.78	25.00	2.78	0.00	36.11
2020	6.25	25.00	25.00	0.00	0.00	43.75

Hover to view yearly counts by race

Observations:

- Of the 5700+ victims of police shootings, less than 4% were unarmed and non-violent
- The number of unarmed and non-violent victims is projected to decrease from a high of 59 in 2015 to 40 individuals(~68%) by EOY 2020 (including unspecified race victims)
- Non-violent unarmed victims are expected to decrease by approximately 80% for Black Americans, 70% for Hispanic Americans, and 55% for White Americans since their highs of 2015
- Since the start of 2015 33% of non-violent unarmed victims were Black, 23% Hispanic, and 38% White
  - Respectively +20%, +5%, and -22% versus their demographic
- Presently, unarmed non-violent victims consist of 6.25% Asian Americans, 25% Black Americans, 25% Hispanic Americans, and 43.75% White Americans

Summary

Throughout the analysis above, we've discovered some interesting trends including some progress as it pertains to the decreasing number of unarmed victims and non-violent victims. Although the yearly number of fatal shootings has held steady since 2015, the number of unarmed victims and unarmed non-violent victims has fallen considerably since 2015. We've also identified that the overwhelming majority of victims were either armed (93%) or attacking the officer (65%) at the time of the shooting.

It is important to re-emphasize the disparity between the Black American demographic (13.5%) and the rate at which they are killed (26%). While the racial demographics provide us with a ballpark figure for expectation and comparison, it should be noted that we can only speculate that each race interacts with police at the same rate as their respective demographic. Hopefully we continue to see the number of yearly victims decline and that the rate of victims for each race more accurately reflect the country's demographic.