In [1]:
```
import pandas as pd
```

In [2]:
```
hatecrime = pd.read_csv("hate_crime.csv")
```

In [3]:
```
hatecrime.head()
```

Out[3]:

| | incident_id | data_year | ori | pug_agency_name | pub_agency_unit | agency_type_name |
|---|---|---|---|---|---|---|
| **0** | 43 | 1991 | AR0350100 | Pine Bluff | NaN | City |
| **1** | 44 | 1991 | AR0350100 | Pine Bluff | NaN | City |
| **2** | 45 | 1991 | AR0600300 | North Little Rock | NaN | City |
| **3** | 46 | 1991 | AR0600300 | North Little Rock | NaN | City |
| **4** | 47 | 1991 | AR0670000 | Sevier | NaN | County |

5 rows × 28 columns

In [4]:
```
hatecrime.shape
```

Out[4]:
```
(241663, 28)
```

In [37]:
```
hatecrime = hatecrime[hatecrime['data_year'] >= 2013]
```

In [38]:
```
hatecrime.shape
```

Out[38]:
```
(78609, 28)
```

In [39]:
```
agency = hatecrime['pug_agency_name'].unique()
```

In [104…
```
hatecrime_nyc = hatecrime[hatecrime['pug_agency_name']== 'New York']
```

In [105…
```
hatecrime_nyc.head()
```

Out[105]:

| | incident_id | data_year | ori | pug_agency_name | pub_agency_unit | agency_type |
|---|---|---|---|---|---|---|
| **167031** | 166896 | 2013 | NY0303000 | New York | NaN | |
| **167032** | 166897 | 2013 | NY0303000 | New York | NaN | |
| **167033** | 166898 | 2013 | NY0303000 | New York | NaN | |
| **167034** | 166899 | 2013 | NY0303000 | New York | NaN | |
| **167035** | 166900 | 2013 | NY0303000 | New York | NaN | |

5 rows × 28 columns

In [106…
```
hatecrime_nyc.shape
```

Out[106]:
```
(3756, 28)
```

In [107…
```
agg_hcnyc = hatecrime_nyc.groupby('data_year').count()
```

In [332…
```
agg_hcnyc.head(11)
```

Out[332]:

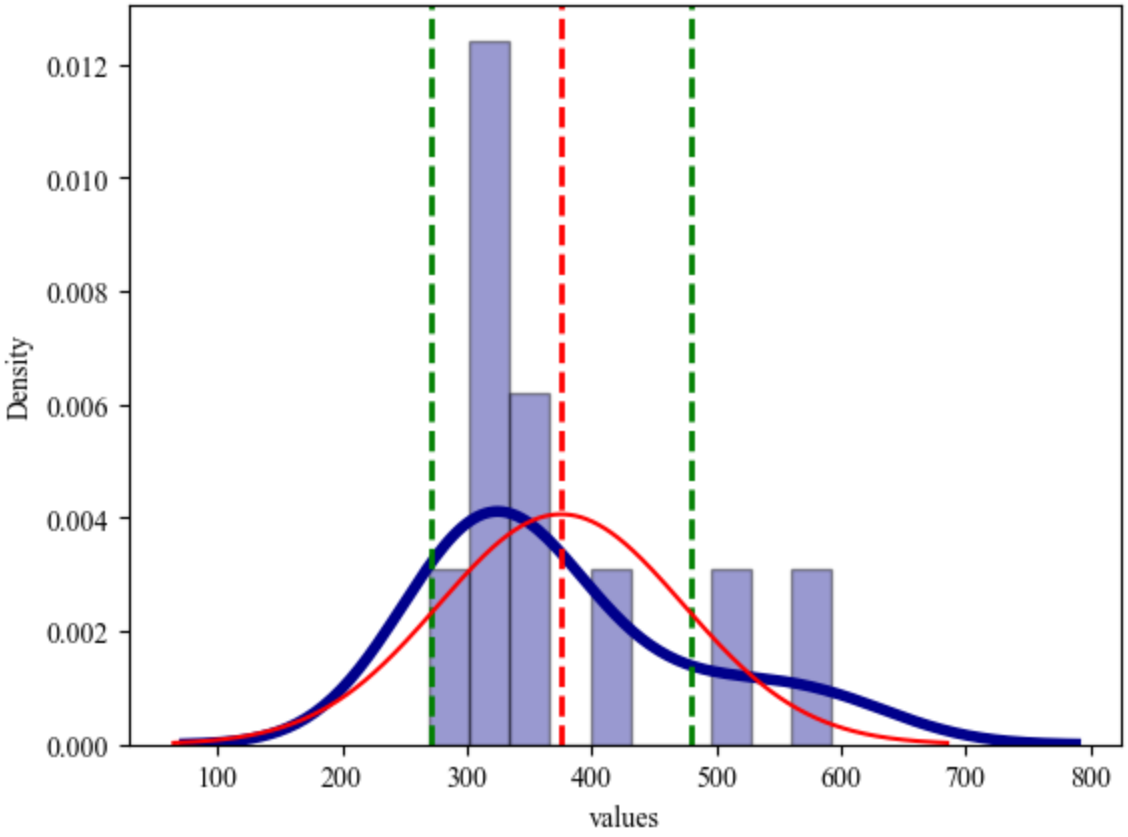
| data_year | incident_id | ori | pug_agency_name | pub_agency_unit | agency_type_name | state_ab |
|---|---|---|---|---|---|---|
| **2013** | 314 | 314 | 314 | 0 | 314 | 3 |
| **2014** | 307 | 307 | 307 | 0 | 307 | 3 |
| **2015** | 307 | 307 | 307 | 0 | 307 | 3 |
| **2016** | 361 | 361 | 361 | 0 | 361 | 3 |
| **2017** | 318 | 318 | 318 | 0 | 318 | 3 |
| **2018** | 351 | 351 | 351 | 0 | 351 | 3 |
| **2019** | 423 | 423 | 423 | 0 | 423 | 4 |
| **2020** | 270 | 270 | 270 | 0 | 270 | 2 |
| **2021** | 513 | 513 | 513 | 0 | 513 | 5 |
| **2022** | 592 | 592 | 592 | 0 | 592 | 5 |

10 rows × 27 columns

In [292…
```
distribution_analysis(agg_hcnyc.incident_id, fit_distribution='normal', bins=1(
```

```
Mean = 375.60
Standard deviation = 103.52
1 percentile = 273.33
5 percentile = 286.65
25 percentile = 308.75
50 percentile = 334.50
75 percentile = 407.50
95 percentile = 556.45
99 percentile = 584.89
```

```
/var/folders/tr/bl8c_0g517nfbgrdbn8f2b2w0000gn/T/ipykernel_25207/2988630648.py:29: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x1, hist=True, kde=vis_curve,
```




Out[292]: (375.6, 98.20814630161797)

In [111…
```
hcnyc_count = hatecrime_nyc.groupby('data_year')['incident_id'].count().reset_
```

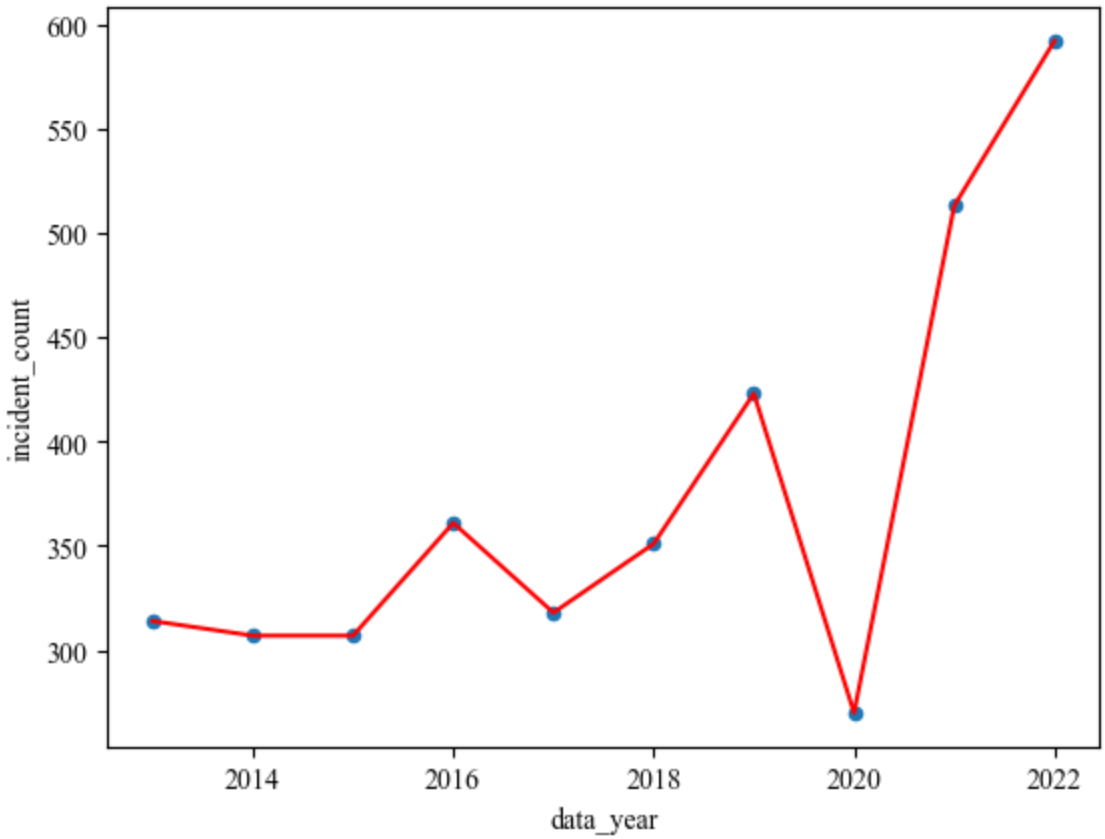
In [112…
```
hcnyc_count.head()
```

Out[112]:

| | data_year | incident_count |
|---|---|---|
| **0** | 2013 | 314 |
| **1** | 2014 | 307 |
| **2** | 2015 | 307 |
| **3** | 2016 | 361 |
| **4** | 2017 | 318 |

In [330…

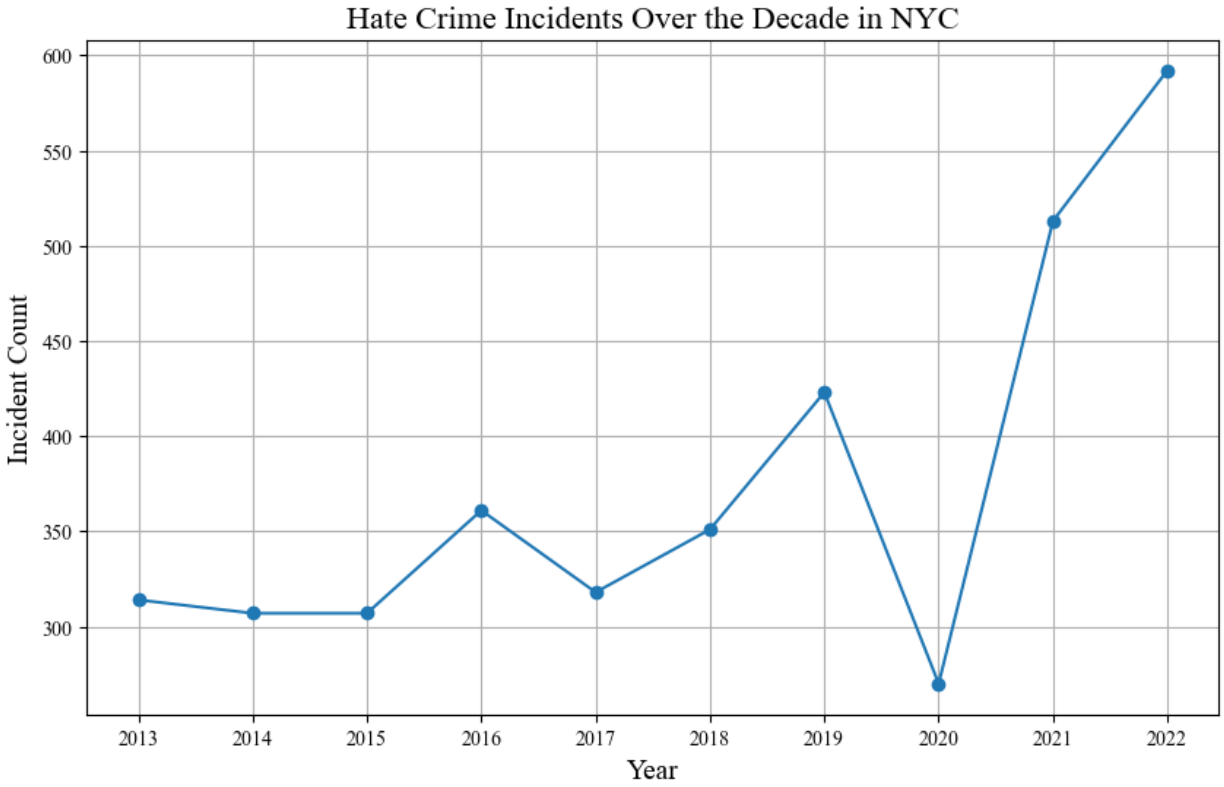
```
fig, ax = plt.subplots() #get axis to plot on
hcnyc_count.plot(ax=ax,kind='scatter',x='data_year', y='incident_count') #show
ax.plot(hcnyc_count['data_year'],hcnyc_count['incident_count'],'r-'); #show th
```




In [113…

```
import matplotlib.pyplot as plt
plt.rcParams['font.family'] = 'Times New Roman'

plt.figure(figsize=(10, 6))
plt.plot(hcnyc_count['data_year'], hcnyc_count['incident_count'], marker='o',
plt.title('Hate Crime Incidents Over the Decade in NYC', fontname='Times New R
plt.xlabel('Year', fontname='Times New Roman', fontsize=14)
plt.ylabel('Incident Count', fontname='Times New Roman', fontsize=14)
plt.grid(True)

# Set x-axis ticks to display each year
plt.xticks(hcnyc_count['data_year'])
```

Out[113]:
```
([<matplotlib.axis.XTick at 0x144ff3850>,
  <matplotlib.axis.XTick at 0x144ffe790>,
  <matplotlib.axis.XTick at 0x145015890>,
  <matplotlib.axis.XTick at 0x1461fac50>,
  <matplotlib.axis.XTick at 0x146204250>,
  <matplotlib.axis.XTick at 0x146207510>,
  <matplotlib.axis.XTick at 0x146209910>,
  <matplotlib.axis.XTick at 0x146204550>,
  <matplotlib.axis.XTick at 0x146210890>,
  <matplotlib.axis.XTick at 0x146212710>],
 [Text(2013, 0, '2013'),
  Text(2014, 0, '2014'),
  Text(2015, 0, '2015'),
  Text(2016, 0, '2016'),
  Text(2017, 0, '2017'),
  Text(2018, 0, '2018'),
  Text(2019, 0, '2019'),
  Text(2020, 0, '2020'),
  Text(2021, 0, '2021'),
  Text(2022, 0, '2022')])
```




In [98]:
```
hatecrime_sfo = hatecrime[hatecrime['pug_agency_name']== 'San Francisco']
hatecrime_sfo = hatecrime_sfo[hatecrime['state_abbr'] == 'CA']
```

```
/var/folders/tr/bl8c_0g517nfbgrdbn8f2b2w0000gn/T/ipykernel_25207/1066112706.p
y:2: UserWarning: Boolean Series key will be reindexed to match DataFrame inde
x.
  hatecrime_sfo = hatecrime_sfo[hatecrime['state_abbr'] == 'CA']
```

In [99]:
```
hatecrime_sfo.head()
```

Out[99]:

| | incident_id | data_year | ori | pug_agency_name | pub_agency_unit | agency_type_ |
|---|---|---|---|---|---|---|
| **163972** | 164006 | 2013 | CA0380100 | San Francisco | NaN | |
| **163973** | 164007 | 2013 | CA0380100 | San Francisco | NaN | |
| **163974** | 164008 | 2013 | CA0380100 | San Francisco | NaN | |
| **163975** | 164009 | 2013 | CA0380100 | San Francisco | NaN | |
| **163976** | 164010 | 2013 | CA0380100 | San Francisco | NaN | |

5 rows × 28 columns

In [100…
```
sf_agency = hatecrime_sfo['pug_agency_name'].unique()
sf_agency
```

Out[100]:
```
array(['San Francisco'], dtype=object)
```

In [167…
```
# Concatenate along rows (vertically)
concat_hc = pd.concat([hatecrime_nyc, hatecrime_sfo, hatecrime_px], ignore_ind
```

In [168…
```
concat_hc.head()
```

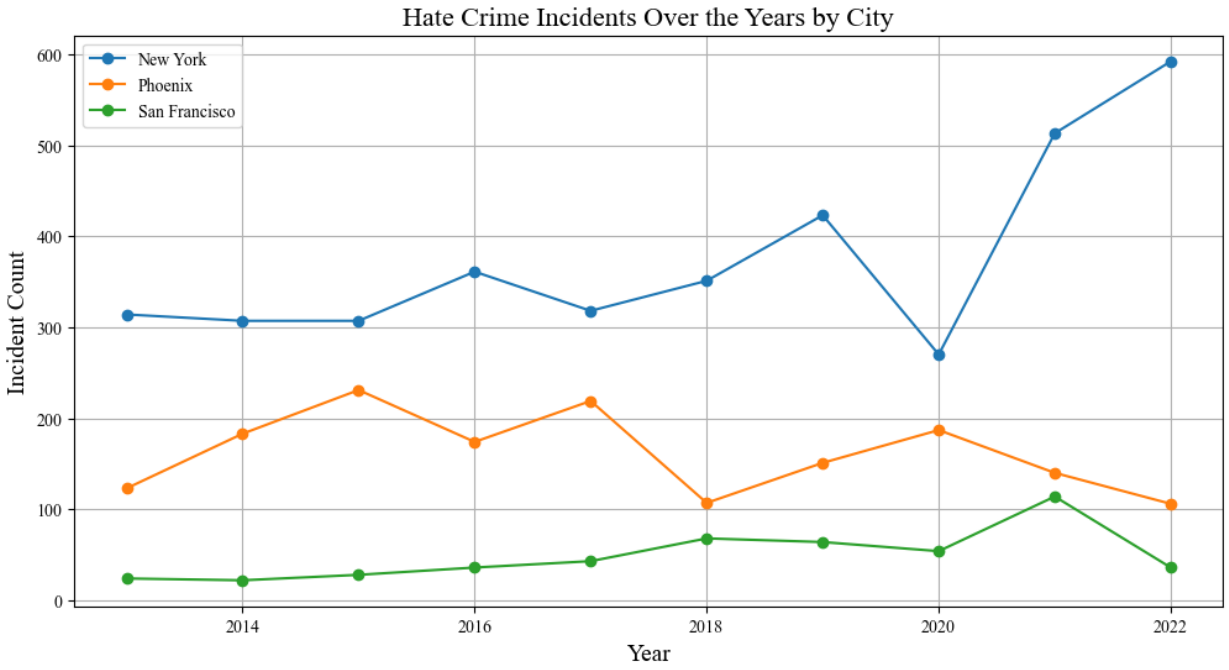
Out[168]:

| | incident_id | data_year | ori | pug_agency_name | pub_agency_unit | agency_type_nam |
|---|---|---|---|---|---|---|
| **0** | 166896 | 2013 | NY0303000 | New York | NaN | Cit |
| **1** | 166897 | 2013 | NY0303000 | New York | NaN | Cit |
| **2** | 166898 | 2013 | NY0303000 | New York | NaN | Cit |
| **3** | 166899 | 2013 | NY0303000 | New York | NaN | Cit |
| **4** | 166900 | 2013 | NY0303000 | New York | NaN | Cit |

5 rows × 28 columns

In [169…
```
hc_count = concat_hc.groupby(['data_year', 'pug_agency_name'])['incident_id'].
```

In [170…
```
plt.figure(figsize=(12, 6))
for agency in hc_count['pug_agency_name'].unique():
    agency_data = hc_count[hc_count['pug_agency_name'] == agency]
    plt.plot(agency_data['data_year'], agency_data['incident_count'], marker='
plt.title('Hate Crime Incidents Over the Years by City', fontname='Times New R
plt.xlabel('Year', fontname='Times New Roman', fontsize=14)
plt.ylabel('Incident Count', fontname='Times New Roman', fontsize=14)
plt.legend()
plt.grid(True)
plt.show()
```




In [336…
```
concat_target = concat_hc.groupby(['bias_desc'])['incident_id'].count().reset_
concat_target
```

Out[336]:

| | bias_desc | incident_count |
|---|---|---|
| 0 | Anti-American Indian or Alaska Native | 24 |
| 1 | Anti-American Indian or Alaska Native;Anti-Bla... | 1 |
| 2 | Anti-Arab | 33 |
| 3 | Anti-Asian | 405 |
| 4 | Anti-Asian;Anti-Female | 1 |
| ... | ... | ... |
| 64 | Anti-Protestant | 3 |
| 65 | Anti-Sikh | 3 |
| 66 | Anti-Transgender | 133 |
| 67 | Anti-Transgender;Anti-White | 2 |
| 68 | Anti-White | 300 |

69 rows × 2 columns

In [172…

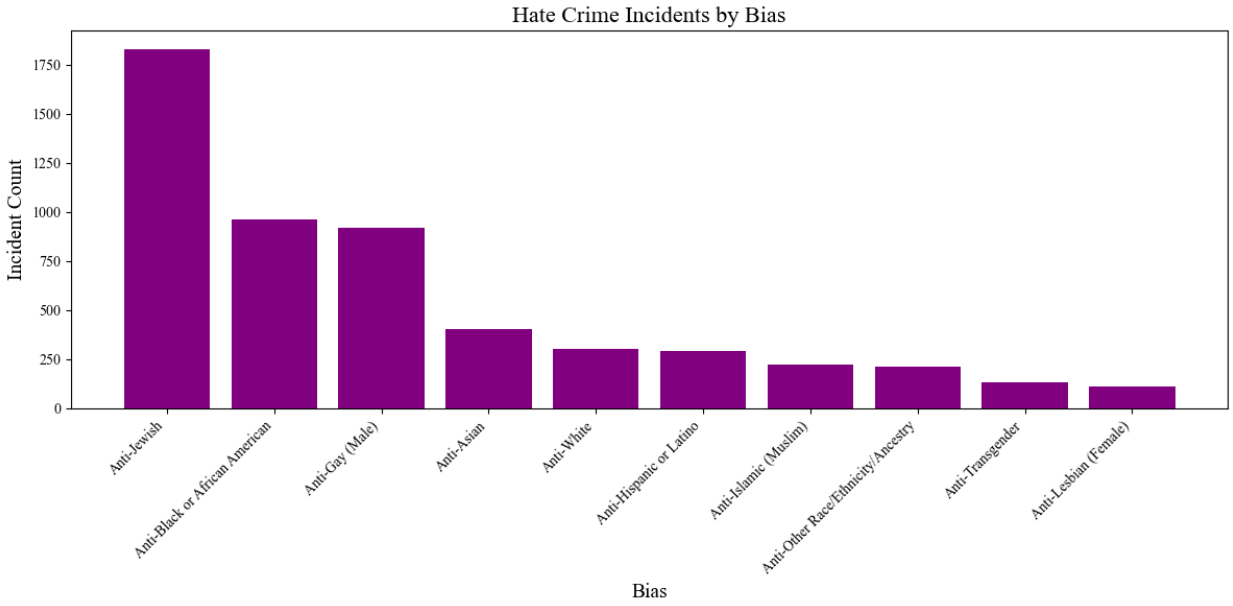
```
concat_target_sorted = concat_target.sort_values(by='incident_count', ascending
top_10_bias = concat_target_sorted.head(10)
top_10_bias
```

Out[172]:

| | bias_desc | incident_count |
|---|---|---|
| 42 | Anti-Jewish | 1832 |
| 7 | Anti-Black or African American | 961 |
| 26 | Anti-Gay (Male) | 917 |
| 3 | Anti-Asian | 405 |
| 68 | Anti-White | 300 |
| 34 | Anti-Hispanic or Latino | 293 |
| 39 | Anti-Islamic (Muslim) | 222 |
| 59 | Anti-Other Race/Ethnicity/Ancestry | 211 |
| 66 | Anti-Transgender | 133 |
| 48 | Anti-Lesbian (Female) | 111 |

In [173…

```
plt.figure(figsize=(12, 6))
plt.bar(top_10_bias['bias_desc'], top_10_bias['incident_count'], color='purple

# Angle x-axis labels
plt.xticks(rotation=45, ha='right')

plt.title('Hate Crime Incidents by Bias', fontname='Times New Roman', fontsize=
plt.xlabel('Bias', fontname='Times New Roman', fontsize=14)
plt.ylabel('Incident Count', fontname='Times New Roman', fontsize=14)
plt.tight_layout()  # Ensures the plot layout is adjusted to prevent clipping o
plt.show()
```




In [346…
```
concat_target_2 = concat_hc.groupby(['bias_desc','pug_agency_name'])['incident_
concat_target_sorted_2 = concat_target_2.sort_values(by='incident_count', ascen
top_10_bias_2 = concat_target_sorted_2.head(15)
top_10_bias_2
```

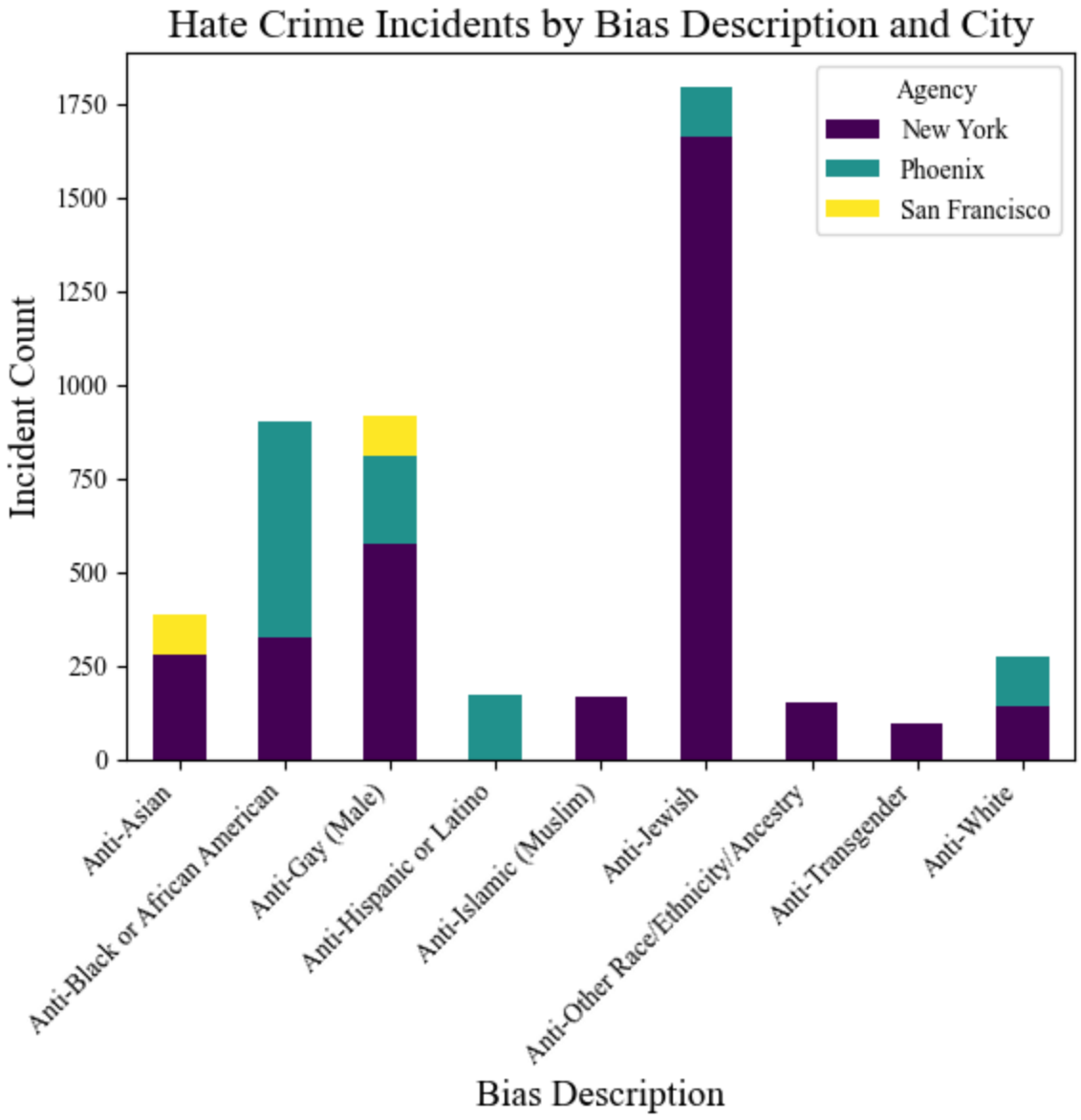
Out[346]:

| | bias_desc | pug_agency_name | incident_count |
|---|---|---|---|
| **62** | Anti-Jewish | New York | 1659 |
| **38** | Anti-Gay (Male) | New York | 576 |
| **14** | Anti-Black or African American | Phoenix | 572 |
| **13** | Anti-Black or African American | New York | 327 |
| **5** | Anti-Asian | New York | 281 |
| **39** | Anti-Gay (Male) | Phoenix | 234 |
| **51** | Anti-Hispanic or Latino | Phoenix | 174 |
| **57** | Anti-Islamic (Muslim) | New York | 168 |
| **90** | Anti-Other Race/Ethnicity/Ancestry | New York | 153 |
| **107** | Anti-White | New York | 142 |
| **63** | Anti-Jewish | Phoenix | 136 |
| **108** | Anti-White | Phoenix | 132 |
| **40** | Anti-Gay (Male) | San Francisco | 107 |
| **7** | Anti-Asian | San Francisco | 104 |
| **103** | Anti-Transgender | New York | 95 |

In [347…
```
pivot_df = top_10_bias_2.pivot(index='bias_desc', columns='pug_agency_name', va

# Set font to Times New Roman
plt.rcParams['font.family'] = 'Times New Roman'

# Plotting the stacked column chart
plt.figure(figsize=(12, 6))
```

```
pivot_df.plot(kind='bar', stacked=True, colormap='viridis')

plt.title('Hate Crime Incidents by Bias Description and City', fontname='Times
plt.xlabel('Bias Description', fontname='Times New Roman', fontsize=14)
plt.ylabel('Incident Count', fontname='Times New Roman', fontsize=14)
plt.legend(title='Agency', loc='upper right')
plt.xticks(rotation=45, ha='right')
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```




In [150…

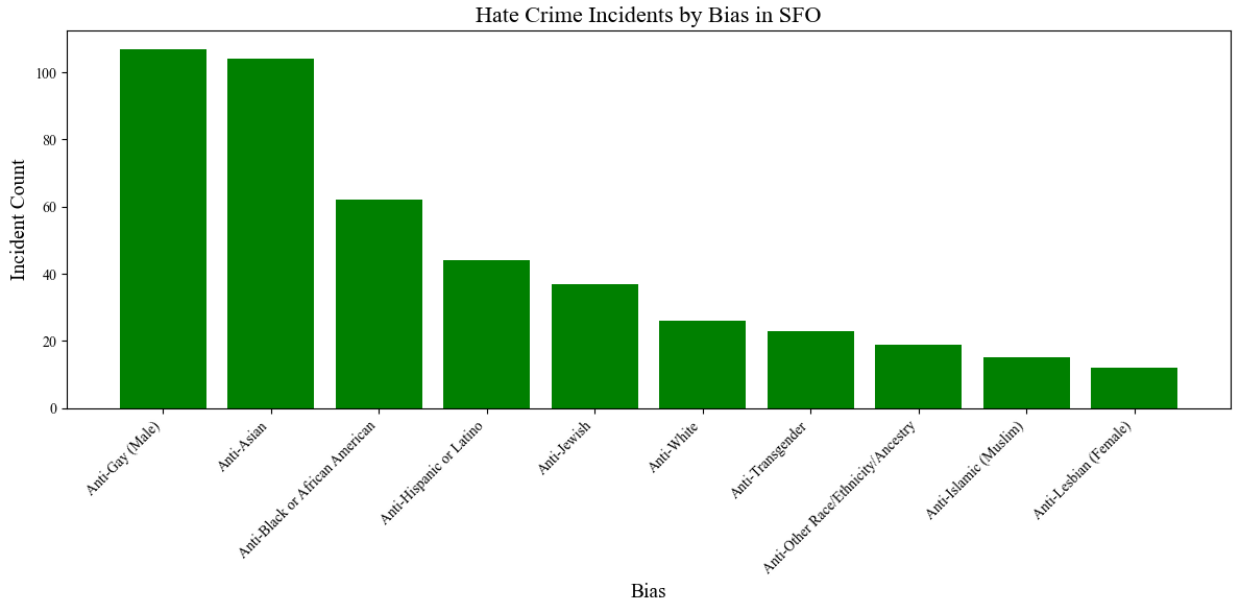
```
hatecrime_sfo.shape
sfo_target = hatecrime_sfo.groupby(['bias_desc'])['incident_id'].count().reset_
sfo_target_sorted = sfo_target.sort_values(by='incident_count', ascending=Fals
top_10_bias_sfo = sfo_target_sorted.head(10)
top_10_bias_sfo
```

Out[150]:

| | bias_desc | incident_count |
|---|---|---|
| 7 | Anti-Gay (Male) | 107 |
| 1 | Anti-Asian | 104 |
| 4 | Anti-Black or African American | 62 |
| 9 | Anti-Hispanic or Latino | 44 |
| 11 | Anti-Jewish | 37 |
| 21 | Anti-White | 26 |
| 20 | Anti-Transgender | 23 |
| 17 | Anti-Other Race/Ethnicity/Ancestry | 19 |
| 10 | Anti-Islamic (Muslim) | 15 |
| 12 | Anti-Lesbian (Female) | 12 |

In [174…

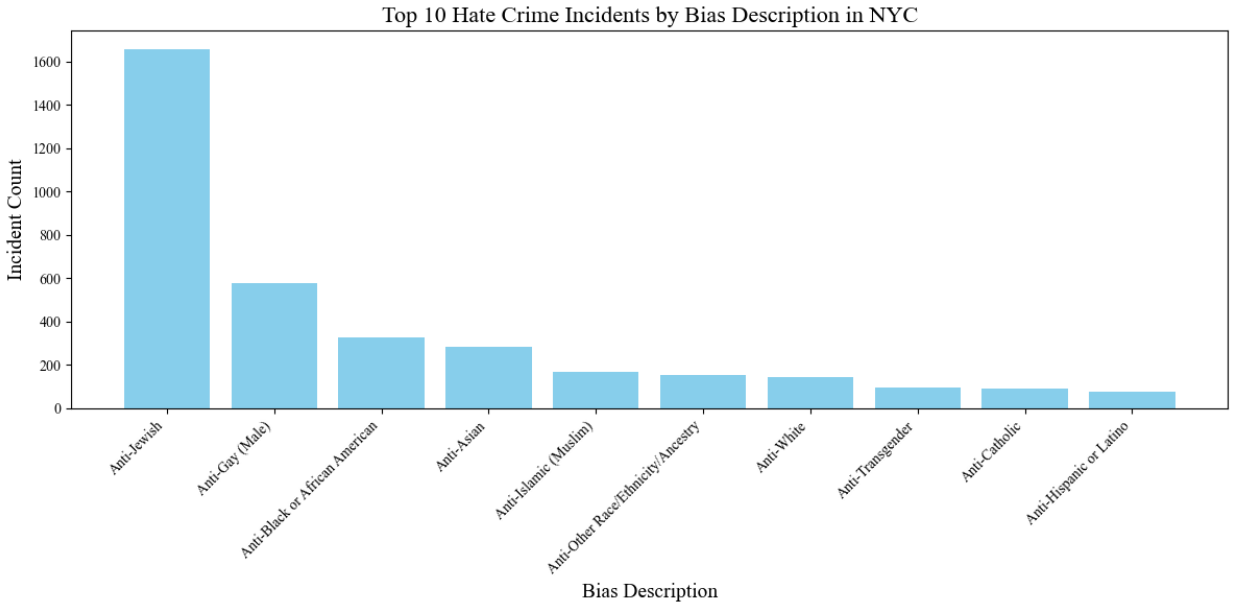
```
plt.figure(figsize=(12, 6))
plt.bar(top_10_bias_sfo['bias_desc'], top_10_bias_sfo['incident_count'], color=

# Angle x-axis labels
plt.xticks(rotation=45, ha='right')

plt.title('Hate Crime Incidents by Bias in SFO', fontname='Times New Roman', f
plt.xlabel('Bias', fontname='Times New Roman', fontsize=14)
plt.ylabel('Incident Count', fontname='Times New Roman', fontsize=14)
plt.tight_layout()  # Ensures the plot layout is adjusted to prevent clipping
plt.show()
```




In [157…

```
hatecrime_nyc.shape
nyc_target = hatecrime_nyc.groupby(['bias_desc'])['incident_id'].count().reset_
nyc_target_sorted = nyc_target.sort_values(by='incident_count', ascending=Fals
top_10_bias_nyc = nyc_target_sorted.head(10)
top_10_bias_nyc
plt.rcParams['font.family'] = 'Times New Roman'

# Plotting the bar graph
```

```
plt.figure(figsize=(12, 6))
plt.bar(top_10_bias_nyc['bias_desc'], top_10_bias_nyc['incident_count'], color=

# Angle x-axis labels
plt.xticks(rotation=45, ha='right')

plt.title('Top 10 Hate Crime Incidents by Bias Description in NYC', fontname='
plt.xlabel('Bias Description', fontname='Times New Roman', fontsize=14)
plt.ylabel('Incident Count', fontname='Times New Roman', fontsize=14)
plt.tight_layout()  # Ensures the plot layout is adjusted to prevent clipping
plt.show()
```



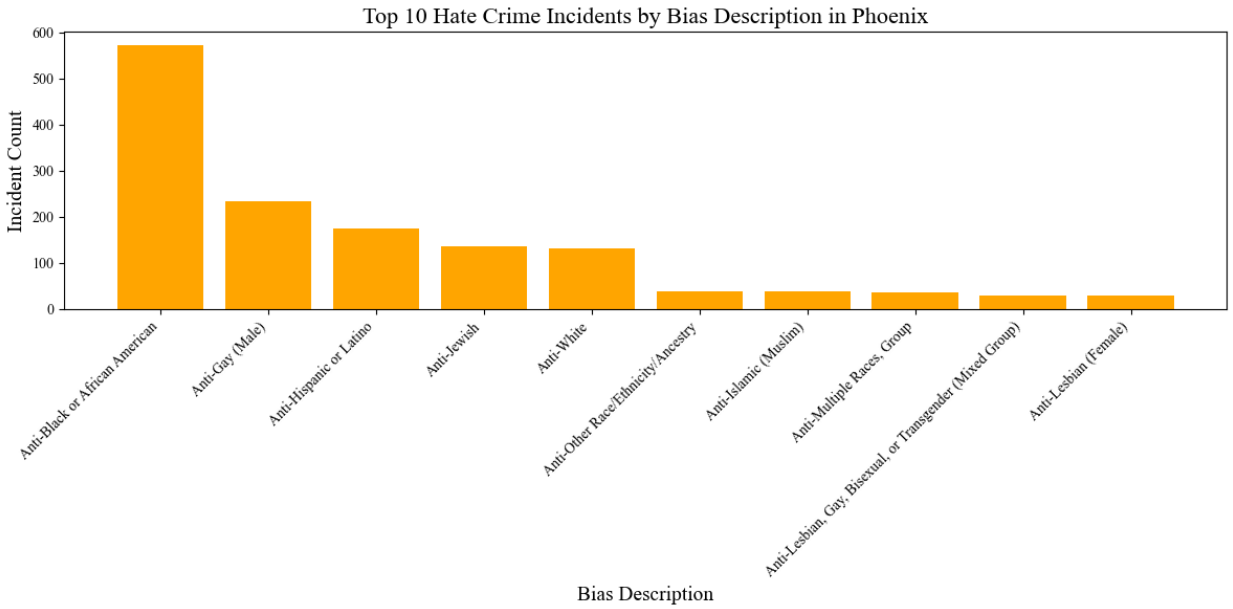

In [165…
```
hatecrime_px = hatecrime[hatecrime['pug_agency_name'].str.contains('Phoenix')]
hatecrime_px = hatecrime_px[hatecrime['agency_type_name'] == 'City']
hatecrime_px = hatecrime_px[hatecrime['state_abbr'] == 'AZ']
hatecrime_px.shape
```

```
/var/folders/tr/bl8c_0g517nfbgrdbn8f2b2w0000gn/T/ipykernel_25207/3050034670.p
y:2: UserWarning: Boolean Series key will be reindexed to match DataFrame inde
x.
  hatecrime_px = hatecrime_px[hatecrime['agency_type_name'] == 'City']
/var/folders/tr/bl8c_0g517nfbgrdbn8f2b2w0000gn/T/ipykernel_25207/3050034670.p
y:3: UserWarning: Boolean Series key will be reindexed to match DataFrame inde
x.
  hatecrime_px = hatecrime_px[hatecrime['state_abbr'] == 'AZ']
```

Out[165]:
```
(1621, 28)
```

In [175…
```
px_target = hatecrime_px.groupby(['bias_desc'])['incident_id'].count().reset_i
px_target_sorted = px_target.sort_values(by='incident_count', ascending=False)
top_10_bias_px = px_target_sorted.head(10)
top_10_bias_px
plt.rcParams['font.family'] = 'Times New Roman'

# Plotting the bar graph
plt.figure(figsize=(12, 6))
plt.bar(top_10_bias_px['bias_desc'], top_10_bias_px['incident_count'], color='

# Angle x-axis labels
plt.xticks(rotation=45, ha='right')
```

```
plt.title('Top 10 Hate Crime Incidents by Bias Description in Phoenix', fontnar
plt.xlabel('Bias Description', fontname='Times New Roman', fontsize=14)
plt.ylabel('Incident Count', fontname='Times New Roman', fontsize=14)
plt.tight_layout()  # Ensures the plot layout is adjusted to prevent clipping
plt.show()
```




In [177…
```
import geopandas as gpd
import urllib.request
import os
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import warnings
```

In [178…
```
stats.ttest_ind(hc_count.loc[hc_count['pug_agency_name']=='New York']['inciden
```

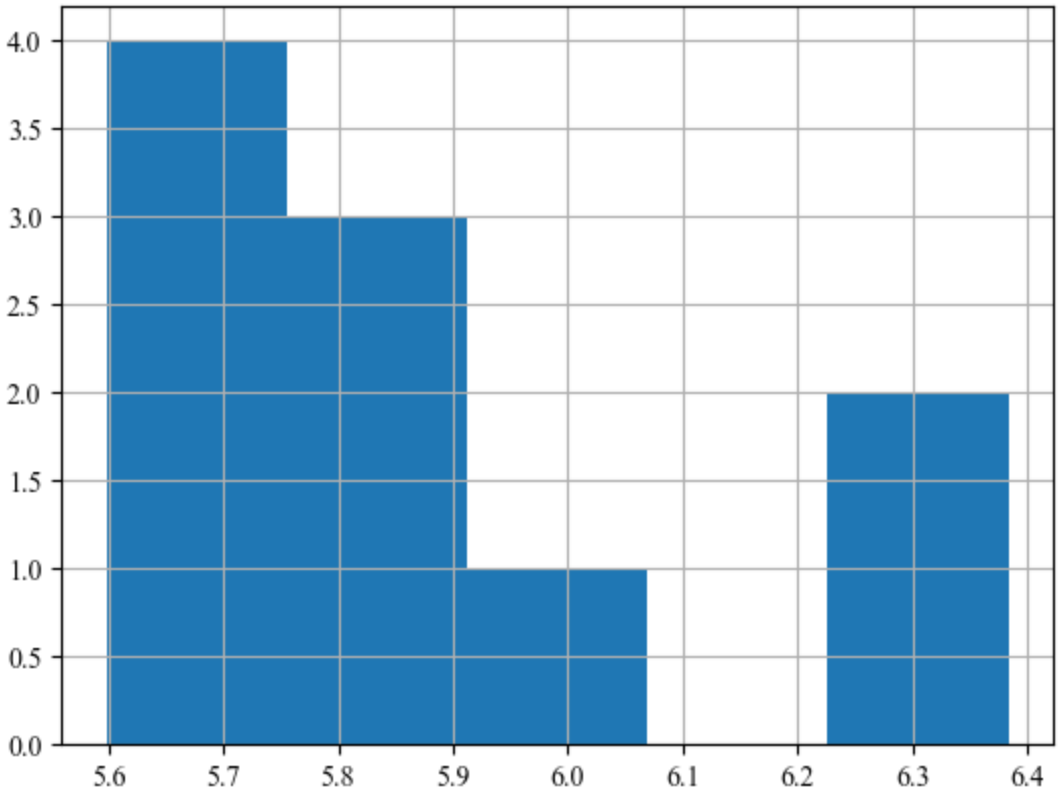
Out[178]:
```
Ttest_indResult(statistic=9.636031586405965, pvalue=1.5752538463642976e-08)
```

In [179…
```
stats.ttest_ind(hc_count.loc[hc_count['pug_agency_name']=='New York']['inciden
```

Out[179]:
```
Ttest_indResult(statistic=6.000008555980327, pvalue=1.1269773625052149e-05)
```

In [185…
```
np.log(hcnyc_count['incident_count']).hist(bins=5)
```

Out[185]:
```
<Axes: >
```




In [247…
```
MHI = pd.read_csv("MHI 2022.csv")
```

In [248…
```
MHI.head()
```

Out[248]:

| | City | State | MHI | Year |
|---|---|---|---|---|
| 0 | Auburn city | Alabama | 54,839 | 2022 |
| 1 | Birmingham city | Alabama | 39,326 | 2022 |
| 2 | Dothan city | Alabama | 53,929 | 2022 |
| 3 | Hoover city | Alabama | 103,194 | 2022 |
| 4 | Huntsville city | Alabama | 68,930 | 2022 |

In [244…
```
unique_agency_names = concat_hc['pug_agency_name'].unique()
```

In [245…
```
MHI_nyc = MHI[MHI['City'].str.contains('New York', case=False, regex=True, na=
MHI_sfo = MHI[MHI['City'].str.contains('San Francisco', case=False, regex=True
MHI_phx = MHI[MHI['City'].str.contains('Phoenix', case=False, regex=True, na=Fa
```

In [249…
```
MHI_scope = pd.concat([MHI_nyc, MHI_phx, MHI_sfo], ignore_index=True)
```

In [219…
```
MHI_scope = pd.concat([MHI_nyc, MHI_phx, MHI_sfo], ignore_index=True)
```

In [250…
```
MHI_scope
```

Out[250]:

| | City | State | MHI | Year |
|---|---|---|---|---|
| 0 | New York city | New York | 74,694 | 2022 |
| 1 | New York city | New York | 52996 | 2014 |
| 2 | New York city | New York | 55752 | 2015 |
| 3 | New York city | New York | 58856 | 2016 |
| 4 | New York city | New York | 60879 | 2017 |
| 5 | New York city | New York | 63799 | 2018 |
| 6 | New York city | New York | 69407 | 2019 |
| 7 | New York Mills city | Minnesota | 45000 | 2020 |
| 8 | West New York town | New Jersey | 64378 | 2020 |
| 9 | New York city | New York | 67046 | 2020 |
| 10 | New York Mills village | New York | 41549 | 2020 |
| 11 | New York city | New York | 67,997 | 2021 |
| 12 | New York city | New York | 52223 | 2013 |
| 13 | Phoenix city | Arizona | 75,969 | 2022 |
| 14 | Phoenix city | Arizona | 47929 | 2014 |
| 15 | Phoenix city | Arizona | 48452 | 2015 |
| 16 | Phoenix city | Arizona | 52062 | 2016 |
| 17 | Phoenix city | Arizona | 56696 | 2017 |
| 18 | Phoenix city | Arizona | 57957 | 2018 |
| 19 | Phoenix city | Arizona | 60931 | 2019 |
| 20 | Phoenix city | Arizona | 60914 | 2020 |
| 21 | Phoenix Lake CDP | California | 56641 | 2020 |
| 22 | Phoenix village | Illinois | 30455 | 2020 |
| 23 | Phoenix village | New York | 52159 | 2020 |
| 24 | Phoenix city | Oregon | 35641 | 2020 |
| 25 | Phoenixville borough | Pennsylvania | 85550 | 2020 |
| 26 | Phoenix city | Arizona | 68,435 | 2021 |
| 27 | Phoenix city | Arizona | 46601 | 2013 |
| 28 | San Francisco city | California | 136,692 | 2022 |
| 29 | San Francisco city | California | 85070 | 2014 |
| 30 | South San Francisco city | California | 86191 | 2014 |
| 31 | San Francisco city | California | 92094 | 2015 |
| 32 | South San Francisco city | California | 96822 | 2015 |
| 33 | San Francisco city | California | 103801 | 2016 |
| 34 | South San Francisco city | California | 90545 | 2016 |

| | City | State | MHI | Year |
|---|---|---|---|---|
| **35** | San Francisco city | California | 110816 | 2017 |
| **36** | South San Francisco city | California | 94459 | 2017 |
| **37** | San Francisco city | California | 112376 | 2018 |
| **38** | South San Francisco city | California | 102365 | 2018 |
| **39** | San Francisco city | California | 123859 | 2019 |
| **40** | South San Francisco city | California | 120573 | 2019 |
| **41** | San Francisco city | California | 119136 | 2020 |
| **42** | South San Francisco city | California | 106005 | 2020 |
| **43** | San Francisco city | California | 121,826 | 2021 |
| **44** | San Francisco city | California | 77485 | 2013 |
| **45** | South San Francisco city | California | 81361 | 2013 |

In [251…
```
unique_city = MHI_scope['City'].unique()
unique_city
```

Out[251]:
```
array(['New York city', 'New York Mills city', 'West New York town',
       'New York Mills village', 'Phoenix city', 'Phoenix Lake CDP',
       'Phoenix village', 'Phoenixville borough', 'San Francisco city',
       'South San Francisco city'], dtype=object)
```

In [252…
```
MHI_scope['City'] = MHI_scope['City'].map({
    'New York city': 'New York',
    'Phoenix city': 'Phoenix',
    'San Francisco city': 'San Francisco',
    'South San Francisco city': 'San Francisco'
})
```

In [255…
```
cities_to_keep = ['New York', 'Phoenix', 'San Francisco']
MHI_scope2 = MHI_scope[MHI_scope['City'].isin(cities_to_keep)]
MHI_scope2
```

Out[255]:

| | City | State | MHI | Year |
|---|---|---|---|---|
| 0 | New York | New York | 74,694 | 2022 |
| 1 | New York | New York | 52996 | 2014 |
| 2 | New York | New York | 55752 | 2015 |
| 3 | New York | New York | 58856 | 2016 |
| 4 | New York | New York | 60879 | 2017 |
| 5 | New York | New York | 63799 | 2018 |
| 6 | New York | New York | 69407 | 2019 |
| 9 | New York | New York | 67046 | 2020 |
| 11 | New York | New York | 67,997 | 2021 |
| 12 | New York | New York | 52223 | 2013 |
| 13 | Phoenix | Arizona | 75,969 | 2022 |
| 14 | Phoenix | Arizona | 47929 | 2014 |
| 15 | Phoenix | Arizona | 48452 | 2015 |
| 16 | Phoenix | Arizona | 52062 | 2016 |
| 17 | Phoenix | Arizona | 56696 | 2017 |
| 18 | Phoenix | Arizona | 57957 | 2018 |
| 19 | Phoenix | Arizona | 60931 | 2019 |
| 20 | Phoenix | Arizona | 60914 | 2020 |
| 24 | Phoenix | Oregon | 35641 | 2020 |
| 26 | Phoenix | Arizona | 68,435 | 2021 |
| 27 | Phoenix | Arizona | 46601 | 2013 |
| 28 | San Francisco | California | 136,692 | 2022 |
| 29 | San Francisco | California | 85070 | 2014 |
| 30 | San Francisco | California | 86191 | 2014 |
| 31 | San Francisco | California | 92094 | 2015 |
| 32 | San Francisco | California | 96822 | 2015 |
| 33 | San Francisco | California | 103801 | 2016 |
| 34 | San Francisco | California | 90545 | 2016 |
| 35 | San Francisco | California | 110816 | 2017 |
| 36 | San Francisco | California | 94459 | 2017 |
| 37 | San Francisco | California | 112376 | 2018 |
| 38 | San Francisco | California | 102365 | 2018 |
| 39 | San Francisco | California | 123859 | 2019 |
| 40 | San Francisco | California | 120573 | 2019 |
| 41 | San Francisco | California | 119136 | 2020 |

| | City | State | MHI | Year |
|---|---|---|---|---|
| **42** | San Francisco | California | 106005 | 2020 |
| **43** | San Francisco | California | 121,826 | 2021 |
| **44** | San Francisco | California | 77485 | 2013 |
| **45** | San Francisco | California | 81361 | 2013 |

In [257…

```
MHI_scope2['year_city'] = MHI_scope2['Year'].astype(str).str.cat(MHI_scope2['C
MHI_scope2
```

```
/var/folders/tr/bl8c_0g517nfbgrdbn8f2b2w0000gn/T/ipykernel_25207/1747338192.p
y:1: 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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  MHI_scope2['year_city'] = MHI_scope2['Year'].astype(str).str.cat(MHI_scope2
['City'].astype(str), sep='_')
```

Out[257]:

| | City | State | MHI | Year | year_city |
|---|---|---|---|---|---|
| 0 | New York | New York | 74,694 | 2022 | 2022_New York |
| 1 | New York | New York | 52996 | 2014 | 2014_New York |
| 2 | New York | New York | 55752 | 2015 | 2015_New York |
| 3 | New York | New York | 58856 | 2016 | 2016_New York |
| 4 | New York | New York | 60879 | 2017 | 2017_New York |
| 5 | New York | New York | 63799 | 2018 | 2018_New York |
| 6 | New York | New York | 69407 | 2019 | 2019_New York |
| 9 | New York | New York | 67046 | 2020 | 2020_New York |
| 11 | New York | New York | 67,997 | 2021 | 2021_New York |
| 12 | New York | New York | 52223 | 2013 | 2013_New York |
| 13 | Phoenix | Arizona | 75,969 | 2022 | 2022_Phoenix |
| 14 | Phoenix | Arizona | 47929 | 2014 | 2014_Phoenix |
| 15 | Phoenix | Arizona | 48452 | 2015 | 2015_Phoenix |
| 16 | Phoenix | Arizona | 52062 | 2016 | 2016_Phoenix |
| 17 | Phoenix | Arizona | 56696 | 2017 | 2017_Phoenix |
| 18 | Phoenix | Arizona | 57957 | 2018 | 2018_Phoenix |
| 19 | Phoenix | Arizona | 60931 | 2019 | 2019_Phoenix |
| 20 | Phoenix | Arizona | 60914 | 2020 | 2020_Phoenix |
| 24 | Phoenix | Oregon | 35641 | 2020 | 2020_Phoenix |
| 26 | Phoenix | Arizona | 68,435 | 2021 | 2021_Phoenix |
| 27 | Phoenix | Arizona | 46601 | 2013 | 2013_Phoenix |
| 28 | San Francisco | California | 136,692 | 2022 | 2022_San Francisco |
| 29 | San Francisco | California | 85070 | 2014 | 2014_San Francisco |
| 30 | San Francisco | California | 86191 | 2014 | 2014_San Francisco |
| 31 | San Francisco | California | 92094 | 2015 | 2015_San Francisco |
| 32 | San Francisco | California | 96822 | 2015 | 2015_San Francisco |
| 33 | San Francisco | California | 103801 | 2016 | 2016_San Francisco |
| 34 | San Francisco | California | 90545 | 2016 | 2016_San Francisco |
| 35 | San Francisco | California | 110816 | 2017 | 2017_San Francisco |
| 36 | San Francisco | California | 94459 | 2017 | 2017_San Francisco |
| 37 | San Francisco | California | 112376 | 2018 | 2018_San Francisco |
| 38 | San Francisco | California | 102365 | 2018 | 2018_San Francisco |
| 39 | San Francisco | California | 123859 | 2019 | 2019_San Francisco |
| 40 | San Francisco | California | 120573 | 2019 | 2019_San Francisco |
| 41 | San Francisco | California | 119136 | 2020 | 2020_San Francisco |

| | City | State | MHI | Year | year_city |
|---|---|---|---|---|---|
| **42** | San Francisco | California | 106005 | 2020 | 2020_San Francisco |
| **43** | San Francisco | California | 121,826 | 2021 | 2021_San Francisco |
| **44** | San Francisco | California | 77485 | 2013 | 2013_San Francisco |
| **45** | San Francisco | California | 81361 | 2013 | 2013_San Francisco |

In [228…

```
hc_count.rename(columns={'pug_agency_name': 'City'}, inplace=True)
hc_count.rename(columns={'data_year': 'Year'}, inplace=True)
hc_count['year_city'] = hc_count['Year'].astype(str).str.cat(hc_count['City'].a
hc_count
```

Out[228]:

| | Year | City | incident_count | year_city |
|---|---|---|---|---|
| 0 | 2013 | New York | 314 | 2013_New York |
| 1 | 2013 | Phoenix | 123 | 2013_Phoenix |
| 2 | 2013 | San Francisco | 24 | 2013_San Francisco |
| 3 | 2014 | New York | 307 | 2014_New York |
| 4 | 2014 | Phoenix | 183 | 2014_Phoenix |
| 5 | 2014 | San Francisco | 22 | 2014_San Francisco |
| 6 | 2015 | New York | 307 | 2015_New York |
| 7 | 2015 | Phoenix | 231 | 2015_Phoenix |
| 8 | 2015 | San Francisco | 28 | 2015_San Francisco |
| 9 | 2016 | New York | 361 | 2016_New York |
| 10 | 2016 | Phoenix | 174 | 2016_Phoenix |
| 11 | 2016 | San Francisco | 36 | 2016_San Francisco |
| 12 | 2017 | New York | 318 | 2017_New York |
| 13 | 2017 | Phoenix | 219 | 2017_Phoenix |
| 14 | 2017 | San Francisco | 43 | 2017_San Francisco |
| 15 | 2018 | New York | 351 | 2018_New York |
| 16 | 2018 | Phoenix | 107 | 2018_Phoenix |
| 17 | 2018 | San Francisco | 68 | 2018_San Francisco |
| 18 | 2019 | New York | 423 | 2019_New York |
| 19 | 2019 | Phoenix | 151 | 2019_Phoenix |
| 20 | 2019 | San Francisco | 64 | 2019_San Francisco |
| 21 | 2020 | New York | 270 | 2020_New York |
| 22 | 2020 | Phoenix | 187 | 2020_Phoenix |
| 23 | 2020 | San Francisco | 54 | 2020_San Francisco |
| 24 | 2021 | New York | 513 | 2021_New York |
| 25 | 2021 | Phoenix | 140 | 2021_Phoenix |
| 26 | 2021 | San Francisco | 114 | 2021_San Francisco |
| 27 | 2022 | New York | 592 | 2022_New York |
| 28 | 2022 | Phoenix | 106 | 2022_Phoenix |
| 29 | 2022 | San Francisco | 36 | 2022_San Francisco |

In [266…

```
common_column = 'year_city'
merged_df = pd.merge(MHI_scope2, hc_count, on=common_column, how='left')
merged_df.dtypes
```

Out[266]:
```
City_x            object
State             object
MHI               object
Year_x             int64
year_city         object
Year_y             int64
City_y            object
incident_count     int64
dtype: object
```

In [310…
```
merged_df['MHI'] = pd.to_numeric(merged_df['MHI'], errors='coerce').astype('in
```

In [311…
```
merged_df = merged_df.dropna(subset=['MHI'])
merged_df
```

Out[311]:

| | City_x | State | MHI | Year_x | year_city | Year_y | City_y | incident_count |
|---|---|---|---|---|---|---|---|---|
| 1 | New York | New York | 52996 | 2014 | 2014_New York | 2014 | New York | 307 |
| 2 | New York | New York | 55752 | 2015 | 2015_New York | 2015 | New York | 307 |
| 3 | New York | New York | 58856 | 2016 | 2016_New York | 2016 | New York | 361 |
| 4 | New York | New York | 60879 | 2017 | 2017_New York | 2017 | New York | 318 |
| 5 | New York | New York | 63799 | 2018 | 2018_New York | 2018 | New York | 351 |
| 6 | New York | New York | 69407 | 2019 | 2019_New York | 2019 | New York | 423 |
| 7 | New York | New York | 67046 | 2020 | 2020_New York | 2020 | New York | 270 |
| 9 | New York | New York | 52223 | 2013 | 2013_New York | 2013 | New York | 314 |
| 11 | Phoenix | Arizona | 47929 | 2014 | 2014_Phoenix | 2014 | Phoenix | 183 |
| 12 | Phoenix | Arizona | 48452 | 2015 | 2015_Phoenix | 2015 | Phoenix | 231 |
| 13 | Phoenix | Arizona | 52062 | 2016 | 2016_Phoenix | 2016 | Phoenix | 174 |
| 14 | Phoenix | Arizona | 56696 | 2017 | 2017_Phoenix | 2017 | Phoenix | 219 |
| 15 | Phoenix | Arizona | 57957 | 2018 | 2018_Phoenix | 2018 | Phoenix | 107 |
| 16 | Phoenix | Arizona | 60931 | 2019 | 2019_Phoenix | 2019 | Phoenix | 151 |
| 17 | Phoenix | Arizona | 60914 | 2020 | 2020_Phoenix | 2020 | Phoenix | 187 |
| 18 | Phoenix | Oregon | 35641 | 2020 | 2020_Phoenix | 2020 | Phoenix | 187 |
| 20 | Phoenix | Arizona | 46601 | 2013 | 2013_Phoenix | 2013 | Phoenix | 123 |
| 22 | San Francisco | California | 85070 | 2014 | 2014_San Francisco | 2014 | San Francisco | 22 |
| 23 | San Francisco | California | 86191 | 2014 | 2014_San Francisco | 2014 | San Francisco | 22 |
| 24 | San Francisco | California | 92094 | 2015 | 2015_San Francisco | 2015 | San Francisco | 28 |
| 25 | San Francisco | California | 96822 | 2015 | 2015_San Francisco | 2015 | San Francisco | 28 |
| 26 | San Francisco | California | 103801 | 2016 | 2016_San Francisco | 2016 | San Francisco | 36 |
| 27 | San Francisco | California | 90545 | 2016 | 2016_San Francisco | 2016 | San Francisco | 36 |
| 28 | San Francisco | California | 110816 | 2017 | 2017_San Francisco | 2017 | San Francisco | 43 |
| 29 | San Francisco | California | 94459 | 2017 | 2017_San Francisco | 2017 | San Francisco | 43 |
| 30 | San | California | 112376 | 2018 | 2018_San | 2018 | San | 68 |

| | City_x | State | MHI | Year_x | year_city | Year_y | City_y | incident_count |
|---|---|---|---|---|---|---|---|---|
| | Francisco | | | | Francisco | | Francisco | |
| **31** | San Francisco | California | 102365 | 2018 | 2018_San Francisco | 2018 | San Francisco | 68 |
| **32** | San Francisco | California | 123859 | 2019 | 2019_San Francisco | 2019 | San Francisco | 64 |
| **33** | San Francisco | California | 120573 | 2019 | 2019_San Francisco | 2019 | San Francisco | 64 |
| **34** | San Francisco | California | 119136 | 2020 | 2020_San Francisco | 2020 | San Francisco | 54 |
| **35** | San Francisco | California | 106005 | 2020 | 2020_San Francisco | 2020 | San Francisco | 54 |
| **37** | San Francisco | California | 77485 | 2013 | 2013_San Francisco | 2013 | San Francisco | 24 |
| **38** | San Francisco | California | 81361 | 2013 | 2013_San Francisco | 2013 | San Francisco | 24 |

In [312…
```
merged_df.describe()
```

Out[312]:

| | MHI | Year_x | Year_y | incident_count |
|---|---|---|---|---|
| **count** | 33.000000 | 33.000000 | 33.000000 | 33.000000 |
| **mean** | 77306.030303 | 2016.606061 | 2016.606061 | 148.212121 |
| **std** | 25318.444257 | 2.370910 | 2.370910 | 123.277826 |
| **min** | 35641.000000 | 2013.000000 | 2013.000000 | 22.000000 |
| **25%** | 56696.000000 | 2015.000000 | 2015.000000 | 43.000000 |
| **50%** | 69407.000000 | 2017.000000 | 2017.000000 | 107.000000 |
| **75%** | 96822.000000 | 2019.000000 | 2019.000000 | 231.000000 |
| **max** | 123859.000000 | 2020.000000 | 2020.000000 | 423.000000 |

In [313…
```
merged_df[['Year_x','incident_count','MHI']].corr()
```

Out[313]:

| | Year_x | incident_count | MHI |
|---|---|---|---|
| **Year_x** | 1.000000 | 0.090213 | 0.269927 |
| **incident_count** | 0.090213 | 1.000000 | -0.662141 |
| **MHI** | 0.269927 | -0.662141 | 1.000000 |

In [314…
```
#introduce a custom function performing distribution analysis
import seaborn as sns
def distribution_analysis(x, log_scale = False, fit_distribution = 'None', bins
    #x - array of observations
    #log_scale - analyze distribution of log(x) if True
    #fit_distribution - fit the distribution ('normal', 'gev' or 'pareto') or
    #bins - how many bins to use for binning the data
    #vis_means - show mean and std lines if True
    #vis_curve - show interpolated distribution curve over the histogram bars
```

```
    #print_outputs - print mean, std and percentiles

    if log_scale:
        x1 = np.log10(x) #convert data to decimal logarithms
        xlabel = 'log(values)' #reflect in x labels
    else:
        x1 = x #leave original scale
        xlabel = 'values'
    mu = x1.mean() #compute the mean
    if log_scale: #if logscale, output all three - log mean, its original scale
        print('Log mean = {:.2f}({:.2f}), mean = {:.2f}'.format(mu,10**mu,x.mea
    else:
        print('Mean = {:.2f}'.format(mu)) #otherwise print mean
    sigma = x1.std() #compute and output standard deviation
    print('Standard deviation = {:.2f}'.format(sigma))
    for p in [1,5,25,50,75,95,99]: #output percentile values
        print('{:d} percentile = {:.2f}'.format(p,np.percentile(x,p)))

    #visualize histogram and the interpolated line (if vis_curve=True) using se
    sns.distplot(x1, hist=True, kde=vis_curve,
        bins=bins,color = 'darkblue',
        hist_kws={'edgecolor':'black'},
        kde_kws={'linewidth': 4})

    #show vertical lines for mean and std if vis_means = True
    if vis_means:
        plt.axvline(mu, color='r', ls='--', lw=2.0)
        plt.axvline(mu-sigma, color='g', ls='--', lw=2.0)
        plt.axvline(mu+sigma, color='g', ls='--', lw=2.0)

    ylim = plt.gca().get_ylim() #keep the y-range of original distribution dens
    #(to make sure the fitted distribution would not affect it)

    h = np.arange(mu - 3 * sigma, mu + 3 * sigma, sigma / 100) #3-sigma visual
    pars = None #fitted distribution parameters

    #fit and visualize the theoretic distribution
    if fit_distribution == 'normal':
        pars = norm.fit(x1)
        plt.plot(h,norm.pdf(h,*pars),'r')
    elif fit_distribution == 'gev':
        pars = gev.fit(x1)
        plt.plot(h,gev.pdf(h,*pars),'r')
    elif fit_distribution == 'pareto':
        pars = pareto.fit(x1)
        plt.plot(h,pareto.pdf(h,*pars),'r')

    plt.xlabel(xlabel) #add x label
    plt.ylim(ylim) #restore the y-range of original distribution density values
    plt.show()
    return pars
```

In [315…

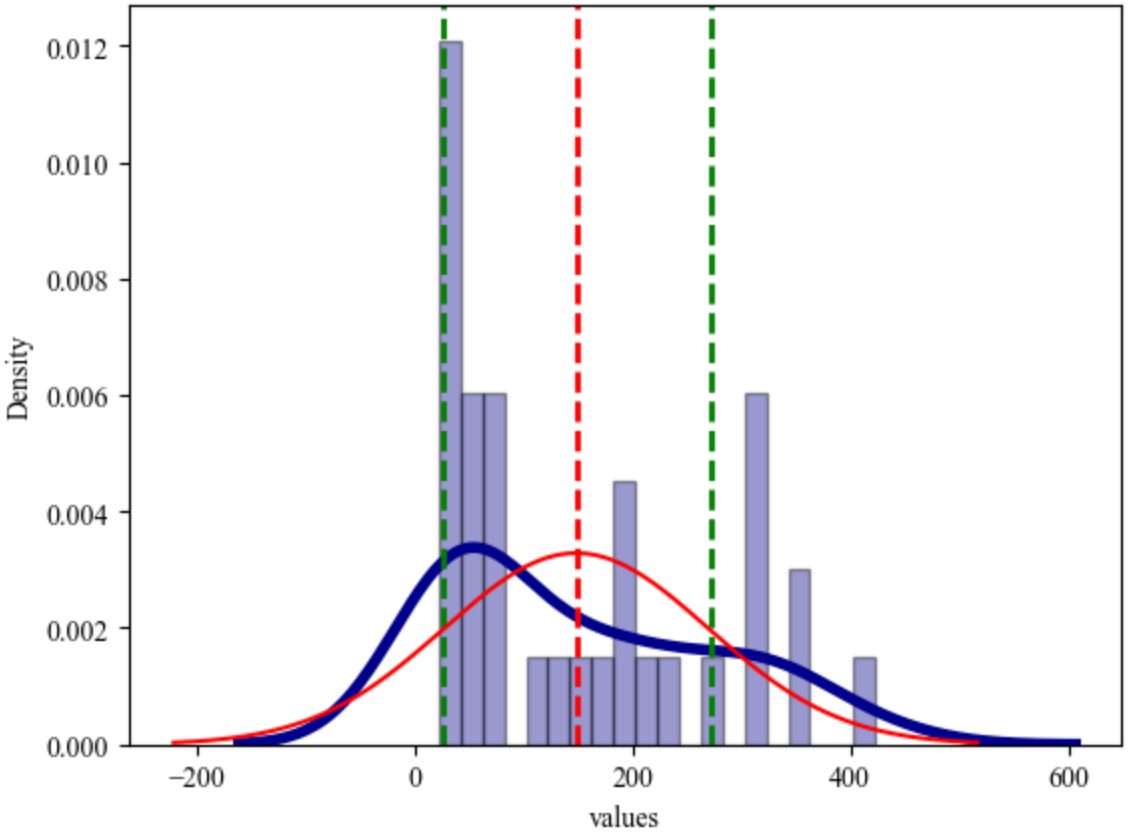
```
from scipy.stats import norm #normal
from scipy.stats import genextreme as gev #generalized extreme value
from scipy.stats import pareto
```

In [316…

```
distribution_analysis(merged_df.incident_count, fit_distribution='normal', bins
```

```
Mean = 148.21
Standard deviation = 123.28
1 percentile = 22.00
5 percentile = 23.20
25 percentile = 43.00
50 percentile = 107.00
75 percentile = 231.00
95 percentile = 355.00
99 percentile = 403.16
```

```
/var/folders/tr/bl8c_0g517nfbgrdbn8f2b2w0000gn/T/ipykernel_25207/2988630648.p
y:29: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x1, hist=True, kde=vis_curve,
```




Out[316]: (148.21212121212122, 121.3956111172507)

In [317…
```
from sklearn.linear_model import LinearRegression
```

In [318…
```
lm = LinearRegression(fit_intercept=False).fit(merged_df[['Year_x']], merged_d
```

In [319…
```
lm.coef_
```

Out[319]: array([0.07350201])

In [320…
```
import statsmodels.formula.api as smf
lm = smf.ols(formula='Year_x~incident_count', data = merged_df).fit()
```

In [321…
```
print(lm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Year_x   R-squared:                       0.008
Model:                            OLS   Adj. R-squared:                 -0.024
Method:                 Least Squares   F-statistic:                    0.2544
Date:                Sun, 10 Dec 2023   Prob (F-statistic):              0.618
Time:                        11:49:42   Log-Likelihood:                -74.670
No. Observations:                  33   AIC:                             153.3
Df Residuals:                      31   BIC:                             156.3
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.
975]
--------------------------------------------------------------------------------
Intercept       2016.3489      0.659   3059.397      0.000    2015.005    201
7.693
incident_count     0.0017      0.003      0.504      0.618      -0.005
0.009
==============================================================================
Omnibus:                        7.595   Durbin-Watson:                   0.935
Prob(Omnibus):                  0.022   Jarque-Bera (JB):                2.164
Skew:                          -0.024   Prob(JB):                        0.339
Kurtosis:                       1.746   Cond. No.                         302.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correct
ly specified.
```

In [322…
```
lm2 = LinearRegression(fit_intercept=False).fit(merged_df[['MHI']], merged_df[
```

In [323…
```
lm2.coef_
```

Out[323]:
```
array([0.00143284])
```

In [349…
```
lm2 = smf.ols(formula='incident_count~MHI', data = merged_df).fit()
```

In [350…
```
print(lm2.summary())
```

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | incident_count | R-squared: | 0.438 |
| Model: | OLS | Adj. R-squared: | 0.420 |
| Method: | Least Squares | F-statistic: | 24.20 |
| Date: | Sun, 10 Dec 2023 | Prob (F-statistic): | 2.70e-05 |
| Time: | 15:04:10 | Log-Likelihood: | -195.67 |
| No. Observations: | 33 | AIC: | 395.3 |
| Df Residuals: | 31 | BIC: | 398.3 |
| Df Model: | 1 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | 397.4486 | 53.232 | 7.466 | 0.000 | 288.882 | 506.015 |
| MHI | -0.0032 | 0.001 | -4.920 | 0.000 | -0.005 | -0.002 |

| | | | |
|---|---|---|---|
| Omnibus: | 3.278 | Durbin-Watson: | 0.495 |
| Prob(Omnibus): | 0.194 | Jarque-Bera (JB): | 2.626 |
| Skew: | 0.690 | Prob(JB): | 0.269 |
| Kurtosis: | 2.926 | Cond. No. | 2.65e+05 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.65e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [ ]: