Table of Contents:

Introduction: Background Information, Context, and Research Question

The prevalence of gun violence continues to be a problem America cannot seem to avoid. In 2023, on average, 118 people a day died from a gun-related incident (Helmke and Song). Americans are 25 times more likely to be shot and killed than others in high-income countries (BRADY). Gun crime violence itself is a major public health problem with staggering economic consequences to society, the healthcare system, and individuals. But gun violence is an incredibly broad topic, and so a focus was put on a city known for its devastating levels of violence: Chicago. In 2023, there were 2,684 shooting victimizations in the city of Chicago alone ("City of Chicago Violence Reduction Dashboard"). Chicago will be the lens through which gun violence is examined, with the hope that some insights from Chicago can be useful when thinking about gun violence across the country.

Therefore, the writers of this report were concerned with addressing the following research question: ***"How has gun crime in Chicago trended from 2010-2023 and how might its prevalence be explained through metrics such as racial composition, educational attainment, sex, age, and geographical distribution?"***

To answer this research question, it was decided to examine data from the Chicago Data Portal related to gunshot injuries (recorded from 1996 onwards), their corresponding crime category, and location of origin to geographically map how gunshot injuries (which serves as a proxy on gun violence) have been distributed and concentrated over the years.

Furthermore, 8+ datasets from Chicago Public Schools (CPS) which list school performance metrics by the years will be utilized to attempt to study the relationship between educational

attainment (composed of metrics such as test score performance, family involvement, suspensions per capita, etc) and the geographic distribution of and correlation with violent crime in Chicago.

Using educational attainment, racial composition, age, sex, and geographical distribution will provide a comprehensive understanding of gun crime in Chicago over the years, allowing a comprehensive exploration of the determinants of violent crime, as well as how city resources can be directed to neighborhoods with high violent crime.

All datasets used in this report are sourced from the City of Chicago's Data Portal, which was initiated by Chicago Mayor Rahm Emanuel in 2012 and has since then served to promote easy access to government data related to public safety, crime, sanitation, and more. Based on initial exploratory data analysis, it was chosen to primarily explore gun violence through the lens of race, zipcode, educational attainment (measured through average test scores), and age group. This was due to inferences made from correlation matrices created (refer EDA section). Refer to full data cleaning and EDA file using the links provided.

Key terms:

Gunshot Injury: This is represented by the column "GUNSHOT_INJURY_I" in the crime dataset and the main dependent variable studied. Originally with "YES" and "NO" and converted to 1 and 0, this is a binary variable, stating if the crime in Chicago was a gunshot injury or another form of violence. The analysis being conducted only considers data where the binary value is 1, as the purpose of this analysis is to analyze gun violence. A value of 0 indicates that no gun crime occurred in the event of a violent crime.

Average Rank: A manually made column titled "AverageRank" or "AR" depending on dataset being used. "AR" quantifies the education level of a zipcode yearly. Average high school standardized test scores (SAT, ACT, and alternative metrics by Chicago Public Schools) were the proxy for determining if a school district had better education than another, and was used because it was very commonly reported, and the standardization of these tests make the results more comparable. Each high school was ranked based on their average test scores, with 1 being the highest ranked school with the best education. These ranks were averaged for schools within a zipcode to find the average ranking per year of each school in a zipcode. This was created because it was believed that level of education is negatively correlated with gunshot prevalence. Refer to data cleaning notebook to view comprehensive creation of this metric.

Race: The racial composition of the victim of guncrime (represented by the column "Race"). 7 race categories are recorded, namely: Black, White Hispanic, White, American Indian, Asian and Pacific Islander, Black Hispanic, & Unknown.

Age: The age of the victim of guncrime (represented by the column "Age"). 9 age group categories are recorded, namely: 0-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80+, & Unknown

p-value: A probability value which explores the likelihood of obtaining a conclusion under the null hypothesis. The writers of this report choose to reject the null hypothesis if a p-value is lower than 0.05.

EDA: Acronym for Exploratory Data Analysis

CPS: Acronym for Chicago Public Schools

Data Description, Data Cleaning, EDA

Data Description:

Here is a description of the first dataset used for analysis:

a. dataSet.csv - Raw dataset obtained from the Chicago Data Portal, this dataset contains information regarding gunshot injuries, racial composition of victims, age, and location of gun crime in Chicago from 1996 to 2023.

This dataset initially included 38 attributes ('CASE_NUMBER', 'DATE', 'BLOCK', 'VICTIMIZATION_PRIMARY', 'INCIDENT_PRIMARY', 'GUNSHOT_INJURY_I', 'UNIQUE_ID', 'ZIP_CODE', 'WARD', 'COMMUNITY_AREA', 'STREET_OUTREACH_ORGANIZATION', 'AREA', 'DISTRICT', 'BEAT', 'AGE', 'SEX', 'RACE', 'VICTIMIZATION_FBI_CD', 'INCIDENT_FBI_CD', 'VICTIMIZATION_FBI_DESCR', 'INCIDENT_FBI_DESCR', 'VICTIMIZATION_IUCR_CD', 'INCIDENT_IUCR_CD', 'VICTIMIZATION_IUCR_SECONDARY', 'INCIDENT_IUCR_SECONDARY', 'HOMICIDE_VICTIM_FIRST_NAME', 'HOMICIDE_VICTIM_MI', 'HOMICIDE_VICTIM_LAST_NAME', 'MONTH', 'DAY_OF_WEEK', 'HOUR', 'LOCATION_DESCRIPTION', 'STATE_HOUSE_DISTRICT', 'STATE_SENATE_DISTRICT', 'UPDATED', 'LATITUDE', 'LONGITUDE', 'LOCATION') and 60412 rows (entries related to gunshot injuries from 1996 - 2023)

The purpose of this dataset is to comply with legal requirements to reveal crime data to the general public, and was created with the intention of providing transparency between the City of Chicago and taxpayers. The dataset was funded using taxpayer funds and compiled by the Chicago Police Department.

The writers of this report posit that this dataset is accurately reported. This is because it measures major crime, which is normally not underreported. Furthermore, the Chicago Police Department does not have an incentive to deliberately leave out information as its focus is to make communities safer, and it can only do this by providing transparency. It is also not legally allowed to "hide" major crime information.

The dataset was initially collected in a very raw format. Column names were not particularly readable, and row entries were unstandardized (for example, one row stated "BLK" while another stated "blk"). The date column was not in datetime format. Nevertheless, the dataset provided us with 60000+ data points which, when eventually cleaned, would provide us substantial insight.

The raw dataset can be found at https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/dataSet.csv

After preprocessing and cleaning dataSet.csv, a new main dataframe is created labelled mainDataSet, along with yearly dataframes and metric specific dataframes (the ones used for the analysis in this final report are listed underneath when imported).

b. Chicago Public Schools CSVs - This is a collection of 8 datasets collected from Chicago Public Schools which document performance indicators for public schools in the City of Chicago for academic years ranging from 2011 - 2023. Each dataset represents one specific academic year.

Each dataset had different attributes, but could be categorized into the following fields - test score performance, safety and cleanliness scores, family engagement, and misconduct rates along with corresponding school names, categories, and zipcodes. There were approximately > 500 rows in each dataset, each corresponding to a specific school and its performance metrics.

The purpose of this dataset is to comply with legal requirements to reveal educational data to the general public, and was created with the intention of providing transparency between the City of Chicago and taxpayers. It is also to identify which schools are underperforming and thus need to be focused on improving. The dataset was funded using taxpayer funds and compiled by Chicago Public Schools. We posit that it is unlikely that Chicago Public Schools, as mentioned above with Chicago Police Department, would deliberately withhold information as its main goal is to provide transparency and identify underperformance. However, it is certainly possible that some of the data could be potentially biased. This is because CPS does not necessarily collect some of the data itself, but receives it from individual schools. Thus, different individual schools may have different standards for categories such as suspensions, leading to discrepancies in reporting. Furthermore, some years data was missing, which required us to undertake extrapolation and imputation (discussed in data limitations).

NOTE: All datasets came with little to no preprocessing, requiring substantial effort in the preprocessing phase itself. Categories across datasets were not consistent and were changing throughout. For example, in one year, CPS measured test performance through ACT scores but then measured it through ISAT the year after. Furthermore, each dataset had certain columns where there was substantial data missing, rendering them unusable for analysis. Lastly, each row's entry itself was unstandardized and differed by year. For example, rows in one year for family engagement included specific numbers, while for the other years included broad categories. This required us to perform substantial cleaning and eventually create our own rankings to use for EDA. The datasets can be found at:

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools_-_High_School_Progress_Report__2013-2014_.csv

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools_-_Progress_Report_Cards__2011-2012_.csv

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools_-_School_Progress_Reports_SY1516.csv

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools_-_School_Progress_Reports_SY1617_20241015.csv

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools_-_School_Progress_Reports_SY1819_20241015.csv

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools_-_School_Progress_Reports_SY2122_20241015.csv

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools_-_School_Progress_Reports_SY2324.csv

https://github.com/Abhaygups/INFO-2950-Final-Project/blob/main/Chicago_Public_Schools___High_School_Progress_Report_Card__2012-2013_.csv

Data Cleaning:

The data cleaning process was multifaceted and complicated. Significant time and effort went into preprocessing before embarking on EDA and Hypothesis testing. It involved the following steps.

Step 1: All dataframes were imported in raw format

Step 2: dataSet.csv was first preprocessed. This involved standardizing row entries (i.e. renaming 'BLK', 'Blk', and 'BLACK' to 'Black', and so on so forth), separating datetimes, and dropping unnecessary columns.

Step 3: 8 dataframes for school data (allSchool_11_12, highSchool_12_13, highSchool_13_14, allSchool_15_16, allSchool_16_17, allSchool_18_19, allSchool_21_22, allSchool_23_24) are then preprocessed. This involved the same steps as in step 2, and substantially more standaridization. Columns across datasets were different and thus research needed to be conducted to identify common columns and name them accordingly. Furthermore, each dataset had different metrics for measuring performance. This required the writers of this report to create manual rankings that incorporated all metrics to produce a universal metric. This was, in an abstract sense, undertaken by averaging test scores for each year, and then forming rankings relative to zipcodes. Refer to cleaning notebook for more in-depth analysis.

Step 4: The same methods for creating a test metric were also applied to other 3 metrics - family involvement, educational attainment/test scores, and used ranking to standardize them across all schools.

__Here is the link to the iypnb file containing the full cleaning process:__ PhaseVCleaning

Import Block - Importing all Necessary Libraries and Packages

In [269…
```
import pandas as pd
import numpy as np
import datetime as dt
import duckdb as db
import seaborn as sns
import matplotlib.pyplot as plt
import folium as folium
import math as math
pd.options.mode.copy_on_write = False
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error \
as mse, root_mean_squared_error as rmse, mean_absolute_error \
as mae, mean_absolute_percentage_error
from sklearn.model_selection import train_test_split, \
    KFold, cross_val_score, cross_val_predict
import statsmodels.api as sm

import folium.plugins as plugins
```

Reading in primary datasets which will be used for model creation and analysis

In [271…
```
Test_Gunshots_11_12 = pd.read_csv("Test_Gunshots_11_12")
Test_Gunshots_12_13 = pd.read_csv("Test_Gunshots_12_13")
Test_Gunshots_13_14 = pd.read_csv("Test_Gunshots_13_14")
Test_Gunshots_15_16 = pd.read_csv("Test_Gunshots_15_16")
Test_Gunshots_16_17 = pd.read_csv("Test_Gunshots_16_17")
Test_Gunshots_18_19 = pd.read_csv("Test_Gunshots_18_19")
Test_Gunshots_21_22 = pd.read_csv("Test_Gunshots_21_22")
Test_Gunshots_23_24 = pd.read_csv("Test_Gunshots_23_24")
mainGunshotByRaceDataFrame = pd.read_csv("mainGunshotByRaceDataFrame")
gunshotByZipcode2013 = pd.read_csv("gunshotByZipcode2013")
mainDataSet = pd.read_csv("mainDataSet")
data2023 = pd.read_csv("data2023")
data2021 = pd.read_csv("data2021")
data2018 = pd.read_csv("data2018")
data2016 = pd.read_csv("data2016")
```

```
data2015 = pd.read_csv("data2015")
data2013 = pd.read_csv("data2013")
data2012 = pd.read_csv("data2012")
data2011 = pd.read_csv("data2011")
mergedZipcodesTestRank = pd.read_csv("mergedZipcodesTestRank")
mainGunshotByAgeDataFrame = pd.read_csv("mainGunshotByAgeDataFrame")
```

Relevant EDA:

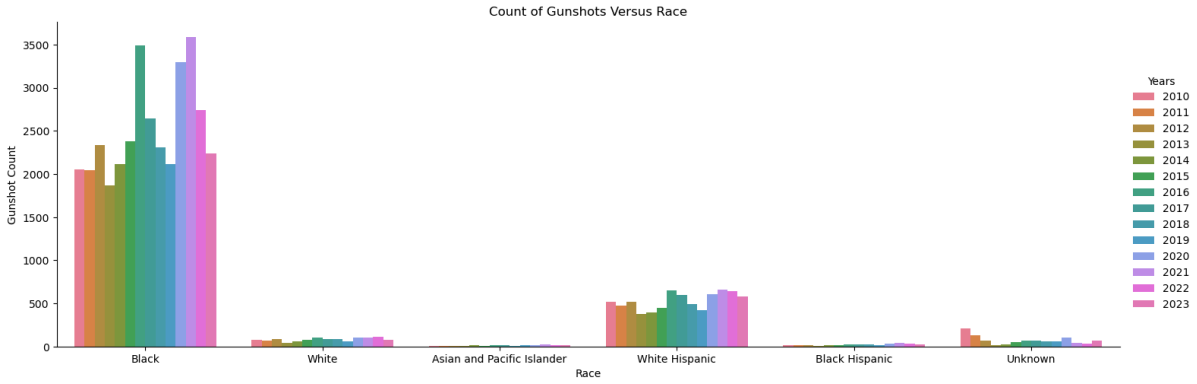
The following visualizations display some of the most important exploratory data analysis conducted. This analysis eventually led to the creation of the research question and hypotheses. Refer to EDA notebook for full EDA.

EDA was conducted on how different demographics may be affected by major crime in Chicago. The primary motivation was to understand if key demographic metrics indeed could be used to analyze violent crime. If a metric showed a specific concentration/trend in violent crime, there would be a higher inclination to use it in research and perform regressions in the future.

The graph shows that African Americans appear disproportionately affected by gun violence in Chicago. This was taken into consideration when establishing "Race" as a potential category to include in the research question.

In [276…

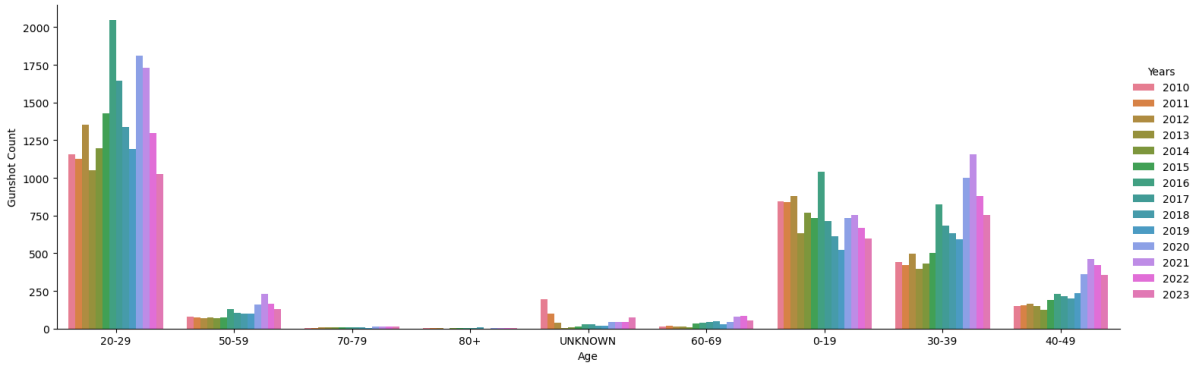
```
mainGunshotByRaceDataFrameMelted = pd.melt\
(mainGunshotByRaceDataFrame, id_vars='Race', \
 var_name='Years', value_name='Gunshot Count' )
gunshotCatPlotRace = sns.catplot\
(data=mainGunshotByRaceDataFrameMelted, \
 x='Race', y='Gunshot Count', kind='bar', \
 hue='Years', aspect=3)
plt.title('Count of Gunshots Versus Race');
```



The graph shows that one's age group may have an effect on their likelihood of experiencing gun violence. A decision was hence made to include this in analysis.

In [278…

```
#Age

mainGunshotByAgeDataFrameMelted = \
pd.melt(mainGunshotByAgeDataFrame, \
        id_vars='Age', var_name='Years', \
        value_name='Gunshot Count' )

gunshotCatPlotAge = \
 sns.catplot(data=mainGunshotByAgeDataFrameMelted, \
             x='Age', y='Gunshot Count', \
             kind='bar', hue='Years', aspect=3)
```



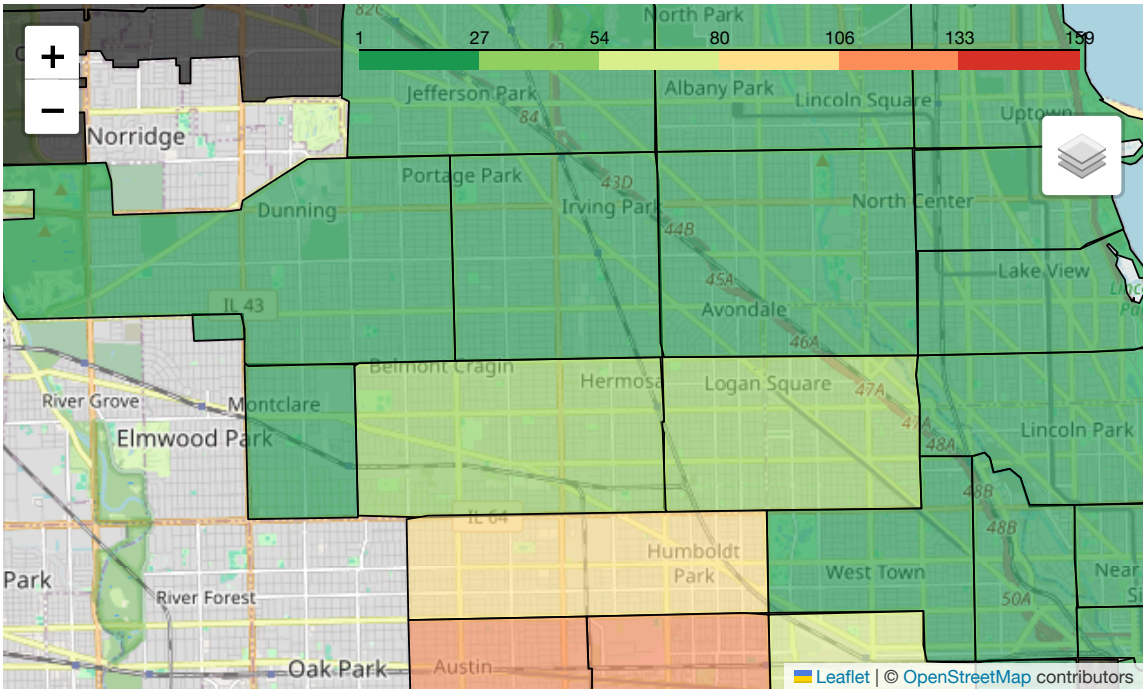
Geographical Mapping

Given Chicago's historical redlining and segregational practices, there was a curiousity to see if there were concentrations of violent crime in specific areas, or if they were randomly dispersed. To do this, geographical maps for every year were created to observe where violent crime tended to be the most concentrated. If indeed there was a trend, there would be an inclination towards including zipcode as a potential metric for analyzing gun crime.

In [280…

```
gunshotInjury2013ChicagoMap = \
folium.Map(location=[41.88407, -87.6333], zoom_start=12)
folium.Choropleth(geo_data='Zip_Codes_Chicago.geojson', \
                  key_on='feature.properties.zip', \
                  data=gunshotByZipcode2013, columns=['Zip Code', \
                 'Gunshot Injury I'], fill_color='RdYlGn_r').\
add_to(gunshotInjury2013ChicagoMap)
folium.LayerControl().add_to(gunshotInjury2013ChicagoMap)

gunshotInjury2013ChicagoMap
```

Out[280…



This visualization shows that gunshot injuries appear to be heavily affected by the zipcode. Certain zipcodes experience a higher proportion of gunshot injuries, implying that one's zipcode may have a substantial effect on experiencing gun violence.

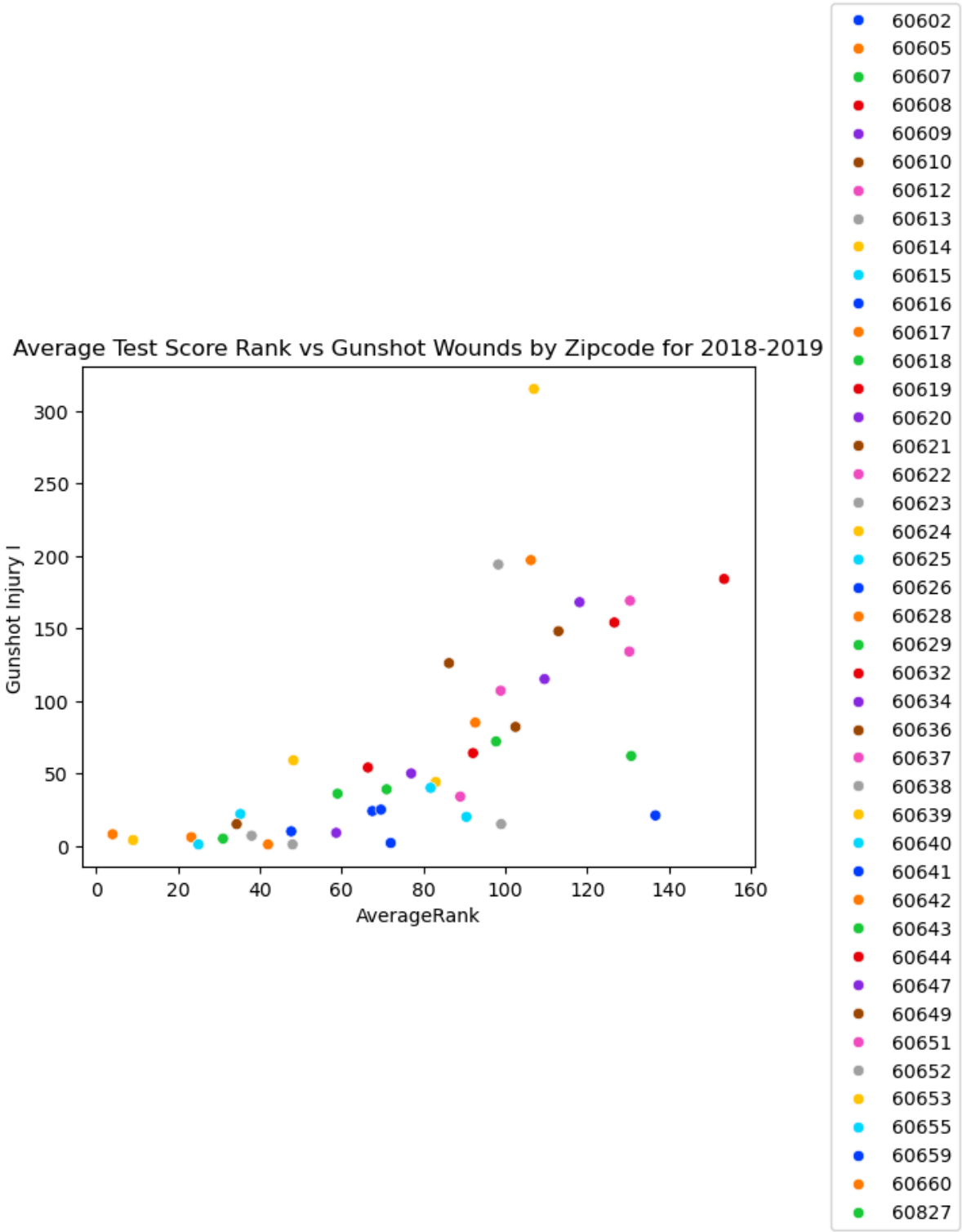
Scatterplots for Test Score Rank vs Gunshot Wounds by Zipcode

In trying to understand if test score rank in an area(our metric for the level of education) affects the chances of being the victim of a major crime, for every year, a scatterplot with the average

testing ranking on the x axis and gunshot injury on the y axis was created. The primary motivation was that better school districts with higher test metrics would have lower crime rates. It appears that schools who performed better academically were also associated with having fewer gunshot injuries in the zipcode. The zipcodes that had schools with the highest testing averages had lower rates of gunshot violence. However, there appears to be a curve for the graphs, suggesting some transformations need to be done if a linear regression were to be run. The following graph underneath displays a visualization of the year 2018-2019 for demonstration purposes. It is necessary to include all the zipcodes to show the distribution of points.

In [283…

```
g = sns.scatterplot(x = Test_Gunshots_18_19["AverageRank"], \
y = Test_Gunshots_18_19["Gunshot Injury I"], \
hue = Test_Gunshots_18_19["Zip"], legend = "full", palette = "bright");
plt.title("Average Test Score Rank vs Gunshot Wounds by Zipcode for 2018-2019")
g.legend(loc='center left', bbox_to_anchor=(1.1, 0.5), ncol=1);
```



Correlation Matrices of Gunshot Injury and Average Academic Rank by Year

Since the previous scatterplots gave an idea of what variables to look at, correlation matrices were created to try and further quantify the relationship between these variables. First, correlation matrices with gunshot injury and academic ranking were looked at because these two variables visually had the largest association. Overall, most of these correlations were between 0.5-0.7, indicating a moderate linear relationship between these variables.

In [285…

```
corr_2011 = Test_Gunshots_11_12.loc[:,\
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot Injury \
and Average Rank for 2011-2012")
print(corr_2011)
corr_2012 = Test_Gunshots_12_13.loc[:,\
```

```
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot \
Injury and Average Rank for 2012-2013")
print(corr_2012)
corr_2013 = Test_Gunshots_13_14.loc[:,\
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot \
Injury and Average Rank for 2013-2014")

print(corr_2013)
corr_2015 = Test_Gunshots_15_16.loc[:,\
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot \
Injury and Average Rank for 2015-2016")

print(corr_2015)
corr_2016 = Test_Gunshots_16_17.loc[:,\
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot Injury \
and Average Rank for 2016-2017")

print(corr_2016)
corr_2018 = Test_Gunshots_18_19.loc[:,\
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot \
Injury and Average Rank for 2018-2019")

print(corr_2018)
corr_2021 = Test_Gunshots_21_22.loc[:,\
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot \
Injury and Average Rank for 2021-2022")

print(corr_2021)
corr_2023 = Test_Gunshots_23_24.loc[:,\
["Gunshot Injury I", "AverageRank"]].corr()
print("Correlation Matrix Between Gunshot \
Injury and Average Rank for 2023-2024")
print(corr_2023)
```

```
Correlation Matrix Between Gunshot Injury and Average Rank for 2011-2012
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.554377
AverageRank               0.554377     1.000000
Correlation Matrix Between Gunshot Injury and Average Rank for 2012-2013
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.556686
AverageRank               0.556686     1.000000
Correlation Matrix Between Gunshot Injury and Average Rank for 2013-2014
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.508065
AverageRank               0.508065     1.000000
Correlation Matrix Between Gunshot Injury and Average Rank for 2015-2016
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.593186
AverageRank               0.593186     1.000000
Correlation Matrix Between Gunshot Injury and Average Rank for 2016-2017
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.582791
AverageRank               0.582791     1.000000
Correlation Matrix Between Gunshot Injury and Average Rank for 2018-2019
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.674299
AverageRank               0.674299     1.000000
Correlation Matrix Between Gunshot Injury and Average Rank for 2021-2022
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.600612
AverageRank               0.600612     1.000000
Correlation Matrix Between Gunshot Injury and Average Rank for 2023-2024
                  Gunshot Injury I  AverageRank
Gunshot Injury I          1.000000     0.668765
AverageRank               0.668765     1.000000
```

> # *Preregistration of Hypotheses*

Subquestion 1: How do the months of the year differ in terms of the probability of gun violence that occurs in that month?

Null Hypothesis:

The month of the year has no significant (linear) relation to the probability of gunshot violence at that month, and the regression model has no useful predictors for predicting gunshot violence. In particular, βJanuary = βFebruary = βMarch = βApril = βMay = βJune = βAugust = βSeptember = βOctober = βNovember = βDecember=0

Alternative Hypothesis:

The linear regression’s fit is better than that of the intercept only model. This means that at least one of the month's coefficients is useful in predicting gunshot violence. In particular, at least one of the β values is not zero.

Explanation for Hypothesis:

It is hypothesized that there will be certain months of the year which experience a higher rate of gun violence as compared to other months. Past research into gun violence in Chicago has indicated that winter months tend to experience lower gun violence rates due to low temperatures. Hotter temperatures lead to shorter tempers, leading to more crime. It is also

hypothesized that the month of July may witness an uptick in gun violence rates due to summer breaks in the Chicago Public School system. The summer also has more gatherings of people, and possibly more drinking, leading to a greater probability of catastrophe.

Analytical Method:

It is chosen to run a multivariate linear regression with the inputs being binary variables for each month. This means that there will be 11 binary variables total and the reference variable will be January, so if all the binary variables have input values of 0, then the month is the reference month (January).

Subquestion 2: How does a zipcode's test-based academic ranking affect the frequency of gun violence in that zipcode?

Null Hypothesis:

The coefficient of the predictor – ranking based on test scores ("AverageRank") – is 0 indicating that there is no significant relationship between a zipcode's test scores and the frequency of gunshot violence.

Alternative Hypothesis:

The coefficient of the predictor (ranking based on test scores/ "Average Rank") ≠ 0 with a p-value of < 0.05/8, or 0.00625, which indicates the likelihood of there being a relationship between a zipcode's test scores and the frequency of gunshot violence.

Explanation for Hypothesis:

It is hypothesized that there will be a positive linear relationship between the frequency of gunshot violence and a higher educational average rank (indicating worse test scores). This is because lower test scores are often associated with university entrance and high school graduation rates, which could lead to either increased involvement in groups committing gun violence or simply increased vulnearability to gun violence due to a lack of social mobility. Research has been conducted to show that educational pathways are significantly associated with changes in crime, with downward education trends being associated with an increase in crime and upward trends being predictive of lower rates of crime. Additionally, research also shows that children's exposure to violence in their neighborhood is associated with worse performance in school due to the disruptive effect it has on their lives (Swisher). This research report aims to see to what extent these trends are present in Chicago zipcodes.

Analytical Method:

It was chosen to use a linear regression because this involved analyzing the effect of a singular input variable ("AverageRank") on gunshot violence frequency. This model was selected over a logistic regression because of the lack of probabilities involved, as outputs are simply frequences/number of occurrences of gunshot violence. This means that there is one independent variable "AverageRank" and one output.

*Subquestion 3: How does the interplay between race, age group, and zipcode affect one's likelihood of

experiencing gun violence?*

Null Hypothesis:

There is no relationship between one's race, age group, and zipcode in affecting their likelihood of experiencing gun violence. This will indicate that the coefficients for all predictor variables will be 0, indicating that there is no relationship between race, age-group, and zipcode and the likelihood of experiencing gun violence.

Alternative Hypothesis:

There is a significant relationship between one's race, age group, or zipcode that affects their likelihood of experiencing gun violence. In particular, predictor variables with coefficients not equal to 0 with significance values of $p < 0.05$ will allow a conclusion that certain races, age groups, and zipcodes may play a role in affecting gun violence likelihood. The null hypothesis is rejected if there is at least one predictor from race, one predictor from age group, one predictor from zipcode whose coefficients are not equal to 0 and whose p values are less than 0.05, and if the model as a whole is significant, as shown from an F-test.

Explanation for Hypothesis:

It is hypothesized that at least one of these predictors from each category (race, zipcode, age group) will be significant due to the EDA conducted prior to starting the analysis. The analysis showed that specific races, zipcodes, and age groups had disproportionately higher rates of gunviolence than others.

Analytical Method:

It is chosen to run a multivariate Negative Binomial linear regression with the inputs being race categories, age group categories, average test score for the particular zipcode, zipcode, and year of occurrence. The reference variable for race was chosen to be white, the reference variable for age group was chosen to be 80+, the reference variable for zipcode was chosen to be 60605, and the reference variable for year was chosen to be 2011. In particular, the reference variable thus becomes a white male in the age group 80+ residing in zipcode 60605 in the year 2011.

> # *Data Analysis:*

Hypothesis 1:

Null Hypothesis:

The month of the year has no significant (linear) relation to the probability of gunshot violence at that month, and the regression model has no useful predictors for predicting gunshot violence. In particular, $\beta_{January} = \beta_{February} = \beta_{March} = \beta_{April} = \beta_{May} = \beta_{June} = \beta_{August} = \beta_{September} = \beta_{October} = \beta_{November} = \beta_{December}=0$

Alternative Hypothesis:

The linear regression’s fit is better than that of the intercept only model. This means that at least one of the month's coefficients is useful in predicting gunshot violence. In particular, at least one of the β values is not zero.

Overview:

A dataframe was formed with the columns of gunshot injury, month, and year for years past 2010. Dummy variables were created from the month column to represent each month with 1s and 0s. Then a probability column was made through aggregating gunshot counts when grouping by month and year and dividing by the total gunshots per year. First, a standard scikit-learn linear regression was conducted with months as the input and probability of gunshot as the output and the data was split into a training set (70%) and a training set (30%). Then, the model's coefficient and intercept were found and the residual plot of the test set was graphed to ensure that the data was not heteroskedastic. Because the data was randomly distributed and not skewed, transformations were not applied. Predictions, summarizations, and limitations were found for the regression. Then, an OLS regression was run to test significance on the model, and evaluation metrics were calculated.

Dummy Creation

Dummy columns were created for each month, and the first column representing January was dropped. These dummy columns were concatenated with the year and gunshot injury columns.

In [292…

```
months_gunshots = db.sql("""SELECT "Gunshot Injury I", Month, Year
                            FROM mainDataSet
                            WHERE Year >= 2010""").df()
month_dummies = pd.get_dummies(months_gunshots["Month"], \
                               drop_first = True)
#converts the dummy variables to 1 and 0 instead of true and false
month_dummies = month_dummies*1
months_df = pd.concat([months_gunshots["Gunshot Injury I"],\
month_dummies],axis = 1)
months_df = pd.concat([months_gunshots["Year"],\
months_df],axis = 1)
```

Aggregation

The dataframe with the columns of gunshot injury, month, and year was filtered for when gunshot injury is 1 (when the crime is gun-related). Then the number of gunshots per month and year for years past 2010 was calculated. Afterwards, a separate dataframe was made with just gunshot occurrences by year. Then probabilities for each set of month and year were calculated by dividing the number of occurences in that month for that year by the number of gunshot occurrences that year. This allows for the probability column being the probability of being shot in a month in a given year. These probabilities were concatenated to the original dataframe.

In [294…

```
#selecting gunshot related crimes
gunshots = db.sql("""SELECT *
                        FROM months_df
                        WHERE "Gunshot Injury I" = 1
                        """).df()

#counting the number of crimes by year and month
gunshots_by_month = db.sql('''SELECT *,
                    COUNT(*) AS "Occurrence" ,
                        FROM gunshots ,
                        GROUP BY * ORDER BY *''').df()

#only counting the instances where gunshot crimes occurred
yearly_gunshots = db.sql("""SELECT "Gunshot Injury I", Year
                            FROM mainDataSet
                            WHERE "Gunshot Injury I" = 1
                            """).df()
```

```
#yearly total of gunshot crimes every year
yearly_gunshots_total = db.sql('''SELECT *,
                    COUNT(*) AS "Occurrence" ,
                        FROM yearly_gunshots ,
                        GROUP BY * ORDER BY *''').df()

#making empty column for probabilities
gunshots_by_month["probability"] = np.zeros(len(gunshots_by_month))
gunshots_array = np.array(gunshots_by_month).astype("int")
yearly_gunshots_total = np.array(yearly_gunshots_total)

probabilities = [] #array for probabilities

for i in range(len(gunshots_by_month)):
     #finding denominator for each month of a year
    denominator = yearly_gunshots_total[gunshots_array[i,0]-1991,2]
    #adding the correct probability by dividing by the denominator
    probabilities.append(gunshots_array[i,13]/denominator*1.0)

#adding probabilities to dataframe
gunshots_by_month["probability"] = probabilities
#filtering out the other columns
X = gunshots_by_month.loc[:,~gunshots_by_month.\
columns.isin(["Gunshot Injury I","Occurrence", "probability", "Year"])]
print(gunshots_by_month.head())
```

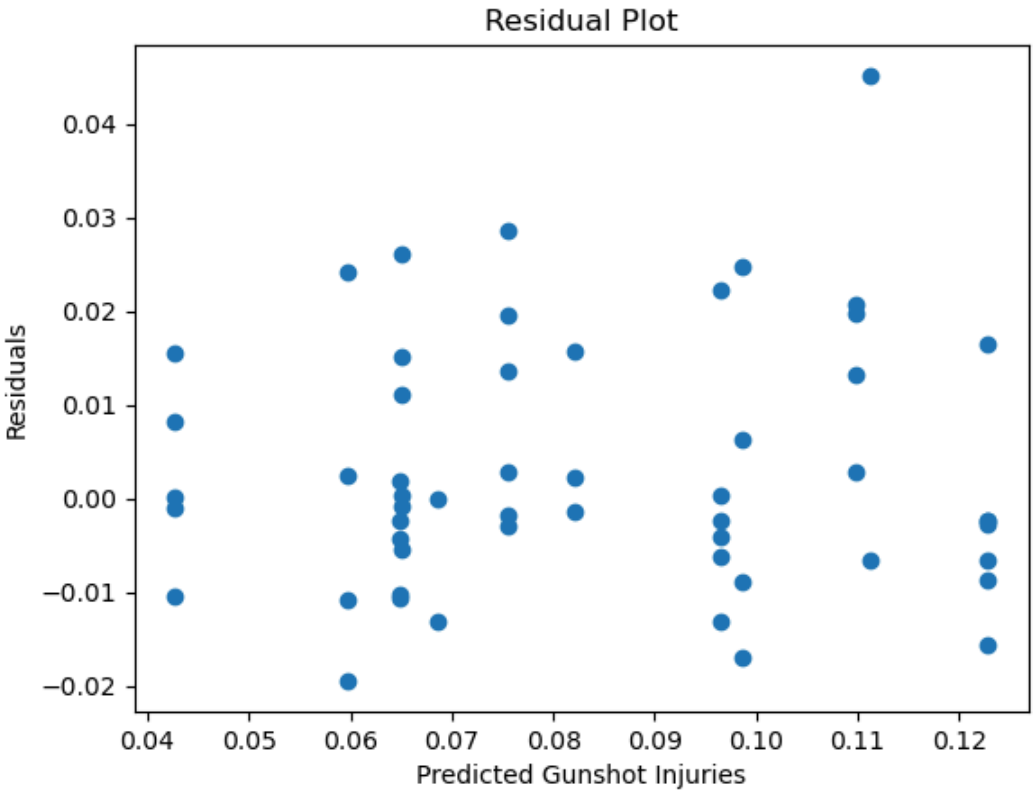
```
   Year  Gunshot Injury I  2  3  4  5  6  7  8  9  10  11  12  Occurrence  \
0  2010                 1  0  0  0  0  0  0  0  0   0   0   0         137
1  2010                 1  0  0  0  0  0  0  0  0   0   0   1         135
2  2010                 1  0  0  0  0  0  0  0  0   0   1   0         176
3  2010                 1  0  0  0  0  0  0  0  0   1   0   0         276
4  2010                 1  0  0  0  0  0  0  0  1   0   0   0         260

   probability
0     0.048946
1     0.048232
2     0.062880
3     0.098607
4     0.092890
```

Model Creation

A train test split was created with 30% being in the test set and 70% in the training set. A linear regression was run with the X inputs being the dummy variables for the months, and the outcome being the probability column. A residual plot was made comparing the predictions on the test set and the actual outcomes for the test set to determine if a linear regression can be used.

In [296…

```
X_train, X_test, y_train, y_test = \
train_test_split(X,gunshots_by_month["probability"], test_size = 0.3)

model = LinearRegression().fit(X_train,y_train)
train_predictions = model.predict(X_train)
test_predictions = model.predict(X_test)

residuals = y_test - test_predictions
plt.scatter(test_predictions, residuals)
plt.xlabel("Predicted Gunshot Injuries")
plt.ylabel("Residuals")
plt.title("Residual Plot")
plt.show()


print("The model's coefficients are " + str(model.coef_))
print("The model's intercept is " + str(model.intercept_))
```




```
The model's coefficients are [-0.0170011   0.00518321  0.01579046  0.0389616   0.05
157126  0.0631009
  0.04999302  0.03674993  0.02243043  0.00888224  0.00525014]
The model's intercept is 0.05973897160826529
```

Summarizing The Regression:

For each coefficient that was not the reference variable, it was interpreted in relation to the reference variable of January. Since each indicator variable is binary, that just means when it is a certain month, the coefficient tells us how the probability of a gunshot-related incident changes when the month changes from January to another month.

In [298…

```
february_p = model.coef_[0]*-1
print(f"The model predicts that when it is Feburary, \
as opposed to January, the probability of a gunshot-related accident \
in Chicago decreases by {february_p:.3f}")
march_p = model.coef_[1]
print(f"The model predicts that when it is March, \
as opposed to January, the probability of a \
gunshot-related accident in Chicago increases by {march_p:.3f}")
april_p = model.coef_[2]
print(f"The model predicts that when it is April, \
as opposed to January, the probability of a gunshot-related \
accident in Chicago increases by {april_p:.3f}")
may_p = model.coef_[3]
print(f"The model predicts that when it is May, as \
opposed to January, the probability of a gunshot-related \
accident in Chicago increases by {may_p:.3f}")
june_p = model.coef_[4]
print(f"The model predicts that when it is June, as \
opposed to January, the probability of a gunshot-related \
accident in Chicago increases by {june_p:.3f}")
july_p = model.coef_[5]
print(f"The model predicts that when it is July, \
as opposed to January, the probability of a gunshot-related accident \
in Chicago increases by {july_p:.3f}")
august_p = model.coef_[6]
print(f"The model predicts when it is August, as opposed to January, \
```

```
the probability of a gunshot-related accident in Chicago increases by {august_p:.3f
september_p = model.coef_[7]
print(f"The model predicts when it is September, \
as opposed to January, \
the probability of a gunshot-related accident in Chicago increases by {september_p:
october_p = model.coef_[8]
print(f"The model predicts when it is October, \
as opposed to January, the probability of a gunshot-related accident \
in Chicago increases by {october_p:.3f}")
november_p =  model.coef_[9]
print(f"The model predicts when it is November, \
as opposed to January, the probability of a gunshot-related accident \
in Chicago increases by {november_p:.3f}")
december_p = model.coef_[10]
print(f"The model predicts when it is December, as \
opposed to January, the probability of a gunshot-related \
accident in Chicago increases by {december_p:.3f}")
```

```
The model predicts that when it is Feburary, as opposed to January, the probability
of a gunshot-related accident in Chicago decreases by 0.017
The model predicts that when it is March, as opposed to January, the probability of
a gunshot-related accident in Chicago increases by 0.005
The model predicts that when it is April, as opposed to January, the probability of
a gunshot-related accident in Chicago increases by 0.016
The model predicts that when it is May, as opposed to January, the probability of a
gunshot-related accident in Chicago increases by 0.039
The model predicts that when it is June, as opposed to January, the probability of
a gunshot-related accident in Chicago increases by 0.052
The model predicts that when it is July, as opposed to January, the probability of
a gunshot-related accident in Chicago increases by 0.063
The model predicts when it is August, as opposed to January, the probability of a g
unshot-related accident in Chicago increases by 0.050
The model predicts when it is September, as opposed to January, the probability of
a gunshot-related accident in Chicago increases by 0.037
The model predicts when it is October, as opposed to January, the probability of a
gunshot-related accident in Chicago increases by 0.022
The model predicts when it is November, as opposed to January, the probability of a
gunshot-related accident in Chicago increases by 0.009
The model predicts when it is December, as opposed to January, the probability of a
gunshot-related accident in Chicago increases by 0.005
```

Predictions for Model:

Since these are all dummy variables, the correct coefficient was set equal to 1 and then added to the intercept to determine the probability of a gunshot-related accident in Chicago for that month. Since January is the reference variable, its value is when every other dummy variable is 0, which is the intercept.

In [300…

```
january_p = model.intercept_ #this is our reference variable
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in January is {january_p:.3f}")
february_p = model.intercept_ + model.coef_[0]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in February is {february_p:.3f}")
march_p = model.intercept_ + model.coef_[1]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in March is {march_p:.3f}")
april_p = model.intercept_ + model.coef_[2]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in April is {april_p:.3f}")
may_p = model.intercept_ + model.coef_[3]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in May is {may_p:.3f}")
june_p = model.intercept_ + model.coef_[4]
print(f"The model predicts that the probability of a \
```

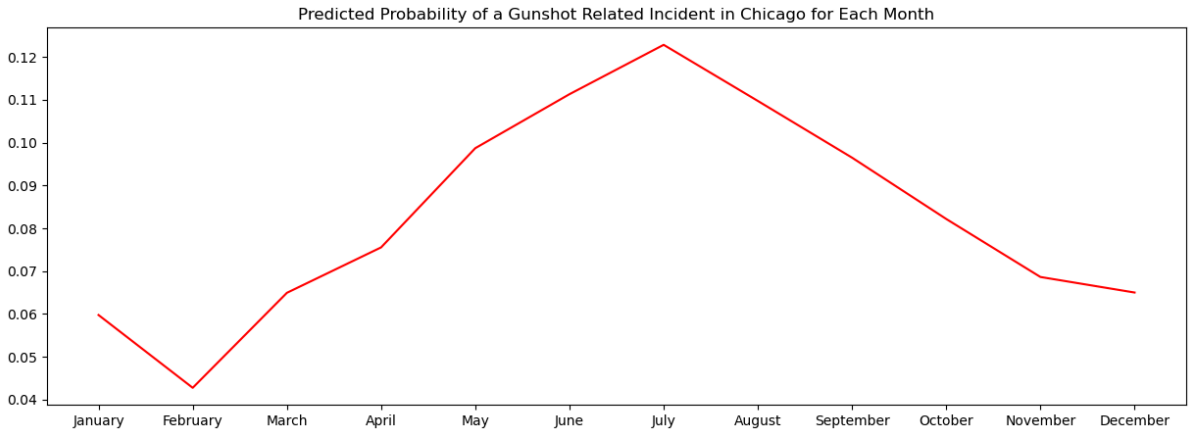
```
gunshot-related accident in Chicago in June is {june_p:.3f}")
july_p = model.intercept_ + model.coef_[5]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in July is {july_p:.3f}")
august_p = model.intercept_ + model.coef_[6]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in August is {august_p:.3f}")
september_p = model.intercept_ + model.coef_[7]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in September is {september_p:.3f}")
october_p = model.intercept_ + model.coef_[8]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in October is {october_p:.3f}")
november_p = model.intercept_ + model.coef_[9]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in November is {november_p:.3f}")
december_p = model.intercept_ + model.coef_[10]
print(f"The model predicts that the probability of a \
gunshot-related accident in Chicago in December is {december_p:.3f}")
```

```
The model predicts that the probability of a gunshot-related accident in Chicago in
January is 0.060
The model predicts that the probability of a gunshot-related accident in Chicago in
February is 0.043
The model predicts that the probability of a gunshot-related accident in Chicago in
March is 0.065
The model predicts that the probability of a gunshot-related accident in Chicago in
April is 0.076
The model predicts that the probability of a gunshot-related accident in Chicago in
May is 0.099
The model predicts that the probability of a gunshot-related accident in Chicago in
June is 0.111
The model predicts that the probability of a gunshot-related accident in Chicago in
July is 0.123
The model predicts that the probability of a gunshot-related accident in Chicago in
August is 0.110
The model predicts that the probability of a gunshot-related accident in Chicago in
September is 0.096
The model predicts that the probability of a gunshot-related accident in Chicago in
October is 0.082
The model predicts that the probability of a gunshot-related accident in Chicago in
November is 0.069
The model predicts that the probability of a gunshot-related accident in Chicago in
December is 0.065
```

A graph of the predicted probabilities is shown below, indicating that indeed July has the highest predicted probability of a gunshot related incident in Chicago.

In [302…

```
fig, ax = plt.subplots(figsize=(15, 5))
ax.plot(["January", "February", "March", "April", \
         "May", "June", "July", "August", "September", \
         "October", "November", "December"],
         [january_p,february_p,march_p,april_p,\
          may_p,june_p,july_p, august_p, september_p, \
          october_p, november_p, december_p],
         color = "red")
plt.title("Predicted Probability of a Gunshot \
Related Incident in Chicago for Each Month")
plt.show()
```



Model Significance & Evaluation:

In [304…
```
X = sm.add_constant(X)
model = sm.OLS(gunshots_by_month["probability"],X)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            probability   R-squared:                       0.797
Model:                            OLS   Adj. R-squared:                  0.783
Method:                 Least Squares   F-statistic:                     59.14
Date:                Mon, 09 Dec 2024   Prob (F-statistic):           1.31e-51
Time:                        20:19:06   Log-Likelihood:                534.62
No. Observations:                 178   AIC:                            -1045.
Df Residuals:                     166   BIC:                            -1007.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0595      0.003     18.535      0.000       0.053       0.066
2             -0.0159      0.005     -3.510      0.001      -0.025      -0.007
3              0.0037      0.005      0.818      0.414      -0.005       0.013
4              0.0200      0.005      4.408      0.000       0.011       0.029
5              0.0396      0.005      8.714      0.000       0.031       0.049
6              0.0544      0.005     11.980      0.000       0.045       0.063
7              0.0619      0.005     13.629      0.000       0.053       0.071
8              0.0540      0.005     11.895      0.000       0.045       0.063
9              0.0368      0.005      8.103      0.000       0.028       0.046
10             0.0238      0.005      5.235      0.000       0.015       0.033
11             0.0082      0.005      1.773      0.078      -0.001       0.017
12             0.0088      0.005      1.904      0.059      -0.000       0.018
==============================================================================
Omnibus:                        7.484   Durbin-Watson:                   1.773
Prob(Omnibus):                  0.024   Jarque-Bera (JB):               12.721
Skew:                           0.115   Prob(JB):                      0.00173
Kurtosis:                       4.289   Cond. No.                         12.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
```

A statsmodel OLS regression was used to help evaluate statistical significance. The F statistic was calculated by statsmodels to be 59.14, with a p-value of 1.31*e-51. This p-value is extremely close to 0, which is less than the alpha level of 0.05, meaning the null hypothesis can be rejected. This means that at least one of the predictors is useful in predicting the probability of a gunshot-related crime.

In [306…
```
print(f"Train MSE: {round(mse(y_train, train_predictions), 4)}")
print(f"Test MSE: {round(mse(y_test, test_predictions), 4)}")

print(f"Train MAE: {round(mae(y_train, train_predictions), 4)}")
print(f"Test MAE: {round(mae(y_test, test_predictions), 4)}")
```

```
Train MSE: 0.0001
Test MSE: 0.0002
Train MAE: 0.009
Test MAE: 0.0102
```

The MSE and MAE were all found for the training set and the test set. The mean squared error (MSE) measures the average squared difference between predicted values and actual values in a dataset. Having a lower MSE is better as it indicates the actual and predicted values are closer together. The mean absolute error (MAE) is a metric measuring the average magnitude of error between actual and predicted values. Having a lower MAE is better as it means the magnitude of error for each prediction is lower on average. Having low values of these metrics for the training set ensures that the model is accurate, while having low values of these metrics for the test set ensures that the model is generalizable. All of these values seem to be pretty low, indicating that the model is both accurate and generalizable.

Limitations:

For the linear regression used in hypothesis 1, utilizing months as the input variable means the interpretations are unintuitive at times. Summarizing a prediction as when it is one month versus another month inherently does not mean very much. Being a month means a large variety of things and is not one measurable quantitative variable, meaning there are probably some factors causing the relationships between a month having a higher probability of gunshot accident that is not the month itself (e.g weather). Ideally, with more time, the true factors driving the seasonality in gunshot-related violence were explored, instead of just seeing the discrepancies in crime per month. The years are also restricted to 2010 and after because the crime reporting metrics changed at that time, but having more data from previous years would make the model predict more accurately. There is no way to use hypothesis testing with the tools from the course to compare the magnitudes of coefficients against each other, so the only thing that could be used to compare how the months generally fared were the predictions from the model. An F test also only tests if at least one of the coefficients is relevant, but does not tell researchers which ones.

Hypothesis 2

Null Hypothesis:

The coefficient of the predictor – ranking based on test scores ("AverageRank") – is 0 indicating that there is no significant relationship between a zipcode's test scores and the frequency of gunshot violence.

Alternative Hypothesis:

The coefficient of the predictor (ranking based on test scores/ "Average Rank") ≠ 0 with a p-value of < 0.05/8, or 0.00625, which indicates the likelihood of there being a relationship between a zipcode's test scores and the frequency of gunshot violence.

Overview: Implementation and Evaluation of Model Generalization

A linear regression was chosen because this involved analyzing the effect of a singular input variable ("AverageRank") on gunshot violence frequency. To do this, a function was written that

could be applied to the data from each year for efficiency.

First, a standard scikit-learn linear regression was conducted and the year's data was split into a training set (80%) and a test set (20%). Then, the model's coefficient and intercept were obtained and the residual plot of the test set was graphed to ensure that the data was not heteroskedastic. Because the data was randomly distributed and not skewed, transformations were not applied. Therefore, the model remains a classic linear regression.

Second, to test the model's generalization, root mean square error and mean absolute error were calculated, both for the test set itself and along with a K-fold cross-validation across 6 folds. An average was taken of the results from all 6 folds. Having reasonable or low values of these metrics indicates that the model is generalizable to broader data.

__Note to reader: Since this hypothesis exploration requires the use of a comprehensive function, the structure of this section is different from that of hypothesis 1 or 3. Certain subsections are clubbed together. The use of a comprehensive function helps with readability and conciseness as without it, individual models would have to be created for all 8 years, which would be redundant.__

Model Creation and Evaluation Function

The following function (year_regression) both creates the linear regression model and tests how well it generalizes using the metrics listed above and K-fold cross-validation. These aspects are combined to make it easier to call the function on different datasets since there are multiple years of data, and to easily obtain a summary, prediction, and generalization of each year. The function also rounds coefficients, intercepts, RMSE, and MAE to 3 decimal places for better readability and consistency.

In [315…

```
def year_regression(df):
#Basic linear regression to get coefficient and intercept for further interpretatic
    X = df["AverageRank"].values.reshape(-1, 1)
    y = df["Gunshot Injury I"].values.reshape(-1, 1)
    X_train, X_test, y_train, y_test = train_test_split\
    (X, y, test_size=0.2)
    model = LinearRegression().fit(X_train, y_train)
    m = model.coef_[0][0]
    a = model.intercept_[0]
    y_pred = model.predict(X_test)
    residuals = y_test - y_pred
    plt.figure(1)
    plt.scatter(y_pred, residuals)
    plt.xlabel("Predicted Gunshot Injuries")
    plt.ylabel("Residuals")
    print(f"The model's coefficient is {m:.3f}")
    print(f"The model's intercept is {a:.3f}")

#Testing model generalization
    RMSE = rmse(X_test, y_pred)
    MAE = mae(X_test, y_pred)
    print(f"This model's root mean squared error for the test set is {RMSE:.3f}")
    print(f"This model's mean absolute error for the test set is {MAE:.3f}")


#Testing model validation further through K-fold cross-validation:
    rmse_scores = -(cross_val_score(model, X, y, cv=6, \
    scoring='neg_root_mean_squared_error'))
    mae_scores = -(cross_val_score(model, X, y, cv=6, \
    scoring='neg_mean_absolute_error'))
    rmse_average_score = np.mean(rmse_scores)
```

```
    mae_average_score = np.mean(mae_scores)
    print(f"RMSE scores across 6 folds: {rmse_average_score:.3f}")
    print(f"MAE scores across 6 folds: {mae_average_score:.3f}")

#Final print of prediction:
    print(f"This model predicts that, for a 1 unit increase in AverageRank, the num
```

Example of Output from Function, with Corresponding Residual Plot and Evaluation of Generalizability

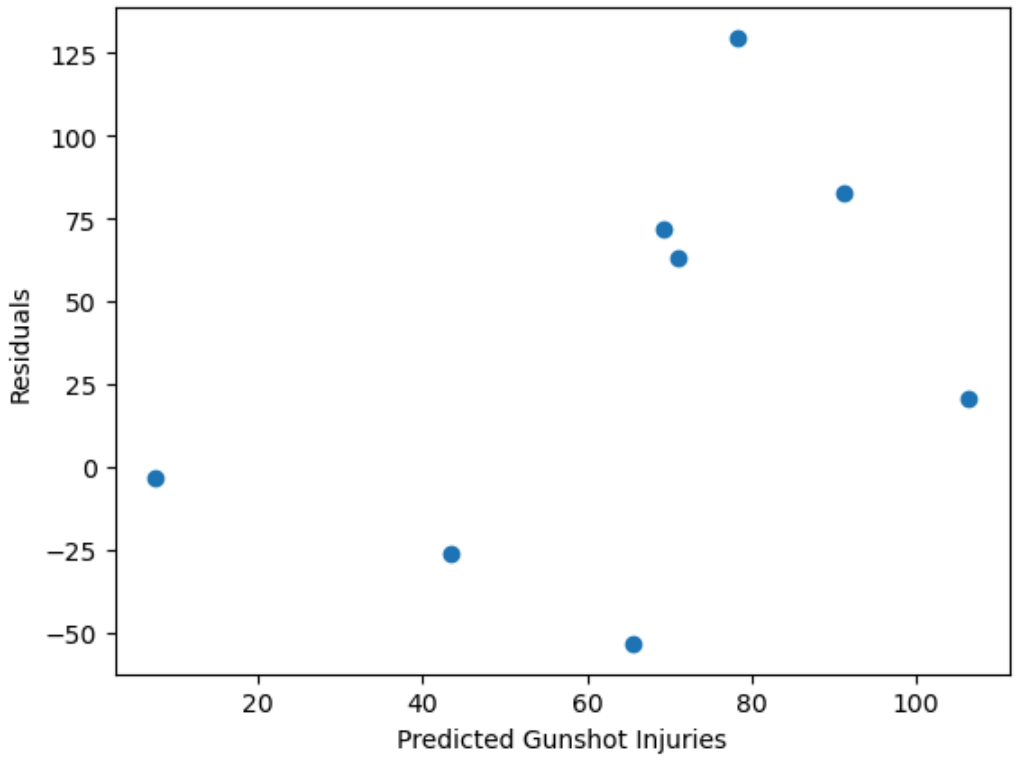
Here, the regression is run on data from 2012 - 2013 as an example to test the regression and to evaluate its generalizability.

The output includes a residual plot drawn from the test set that is clearly random and lacks heteroskedasticity, and properly determines a coefficient and an intercept.

The model generalizes well and is not overfitted as the model's root mean squared error for the test set is approximately 25.269 and its mean absolute error for the test set is approximately 22.890. Using K-fold cross-validation on 6 folds, it is clear that the model has reasonably low average RMSE and MAE scores, indicating that it generalizes well and is not overfitted. The relatively low root mean squared error scores indicate that the model makes fairly accurate predictions and fits the data well. The low mean absolute error scores indicate that predicted results are similar to true results, and that the model is accurate. These metrics were chosen to assess the model's overall accuracy in predicting gunshot violence and to check that predicted values were similar to actual values.

In [318…
```
year_regression(Test_Gunshots_12_13)
```

```
The model's coefficient is 1.610
The model's intercept is -3.751
This model's root mean squared error for the test set is 25.269
This model's mean absolute error for the test set is 22.890
RMSE scores across 6 folds: 55.005
MAE scores across 6 folds: 44.754
This model predicts that, for a 1 unit increase in AverageRank, the number of gunsh
ot injuries increases by 1.610
```



Model Significance:

Second, a statsmodel OLS linear regression was used in the following function (evaluation) to help evaluate statistical significance. The Bonferroni method of correction was used to properly scale the required p-value threshold for statistical significance down in order to preserve only true positives and reduce false positives. This means that, since 8 hypothesis tests are to be run for 8 different year datasets, each calculated p-value must be less than chosen a/8. The chosen value is 0.05, meaning each p-value must be below 0.05/8, or 0.00625.

In [320…
```
def significance(df):
#OLS regression and summary statistics for hypothesis testing
    X1 = df["AverageRank"].values.reshape(-1, 1)
    y1 = df["Gunshot Injury I"].values.reshape(-1, 1)
    X1_train, X1_test, y1_train, y1_test = \
    train_test_split(X1, y1, test_size=0.2)
    X1_train_with_constant = sm.add_constant(X1_train)
    X1_test_with_constant = sm.add_constant(X1_test)
    model1 = sm.OLS(y1_train, X1_train_with_constant)
    results = model1.fit()
    print(results.summary())
```

Below, as an example, the function is again run on data from 2012-2013. It is clear that the resulting p-value is 0.003, which is below the threshold of 0.00625, indicating that the coefficient is statistically significant and that there is evidence that the null hypothesis of AverageRank having no effect on gunshot violence can be rejected.

In [322…
```
significance(Test_Gunshots_12_13)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.290
Model:                            OLS   Adj. R-squared:                  0.262
Method:                 Least Squares   F-statistic:                     10.60
Date:                Mon, 09 Dec 2024   Prob (F-statistic):            0.00314
Time:                        20:19:06   Log-Likelihood:                -151.47
No. Observations:                  28   AIC:                             306.9
Df Residuals:                      26   BIC:                             309.6
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.8784     26.479     -0.071      0.944     -56.307      52.550
x1             1.7159      0.527      3.256      0.003       0.633       2.799
==============================================================================
Omnibus:                        0.939   Durbin-Watson:                   1.488
Prob(Omnibus):                  0.625   Jarque-Bera (JB):                0.903
Skew:                           0.383   Prob(JB):                        0.637
Kurtosis:                       2.566   Cond. No.                         125.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
```

Here is a function that shows the p-values for the coefficients for all 8 years so that it can be confirmed that the p-values for all 8 years are below the threshold required given the Bonferroni correction (0.00625). Each p-value is rounded to 5 decimal places for consistency and readability, "year 1" is the first dataframe passed in as a parameter, or 2011 - 2012 in this case.

It can be seen that for each year, the p-value of the coefficient derived was under 0.00625 and extremely close to 0, indicating that there is strong evidence that the coefficient is significant, allowing a rejection of the null hypothesis and an acceptance of the alternative hypothesis. Several years are shown as examples below. While the rest of the years are not shown in order to avoid excessive scrolling and redundancy, results are consistent across every year. Similar to year_regression, a function was created to test model significance so that different datasets for different years could be passed in and evaluated for significance easily and in a readable form.

In [324…
```
def confirm_years_significance(df1, df2, df3, df4, df5, df6, df7, df8):
    x = [df1, df2, df3, df4, df5, df6, df7, df8]
    p_values_list = []
    k = 0
    for df in x:
        X1 = df["AverageRank"].values.reshape(-1, 1)
        y1 = df["Gunshot Injury I"].values.reshape(-1, 1)
        X1_train, X1_test, y1_train, y1_test = \
        train_test_split(X1, y1, test_size=0.2)
        X1_train_with_constant = sm.add_constant(X1_train)
        X1_test_with_constant = sm.add_constant(X1_test)
        model1 = sm.OLS(y1_train, X1_train_with_constant)
        results = model1.fit()
        p_value_coefficient = results.pvalues[1]
        print(f"The p-value of the coefficient for AverageRank of year {k}  is {p_v
        p_values_list.append(p_value_coefficient)
        p_values_df = pd.DataFrame(p_values_list)
        k = k + 1
    print(p_values_df);
```

Note: In this case, year 0 corresponds to 2011-2012 (the first year), year 1 corresponds to 2012-2013, year 2 corresponds to 2013-2014, year 3 corresponds to 2015-2016, year 4 responds to 2016-2017, year 5 corresponds to 2018-2019, year 6 corresponds to 2021-2022, and year 7 corresponds to 2023-2024. This simply illustrates that all 8 years successfuly have a p-value that is lower than 0.00625.

In [326…
```
confirm_years_significance(Test_Gunshots_11_12, Test_Gunshots_12_13, \
Test_Gunshots_13_14, Test_Gunshots_15_16, Test_Gunshots_16_17, \
Test_Gunshots_18_19, Test_Gunshots_21_22, Test_Gunshots_23_24)
```

```
The p-value of the coefficient for AverageRank of year 0  is 0.00196
The p-value of the coefficient for AverageRank of year 1  is 0.00188
The p-value of the coefficient for AverageRank of year 2  is 0.00738
The p-value of the coefficient for AverageRank of year 3  is 0.00033
The p-value of the coefficient for AverageRank of year 4  is 0.00079
The p-value of the coefficient for AverageRank of year 5  is 0.00004
The p-value of the coefficient for AverageRank of year 6  is 0.00000
The p-value of the coefficient for AverageRank of year 7  is 0.00001
          0
0  0.001961
1  0.001877
2  0.007379
3  0.000333
4  0.000788
5  0.000037
6  0.000001
7  0.000008
```

Generalized Model Summary and Predictions:

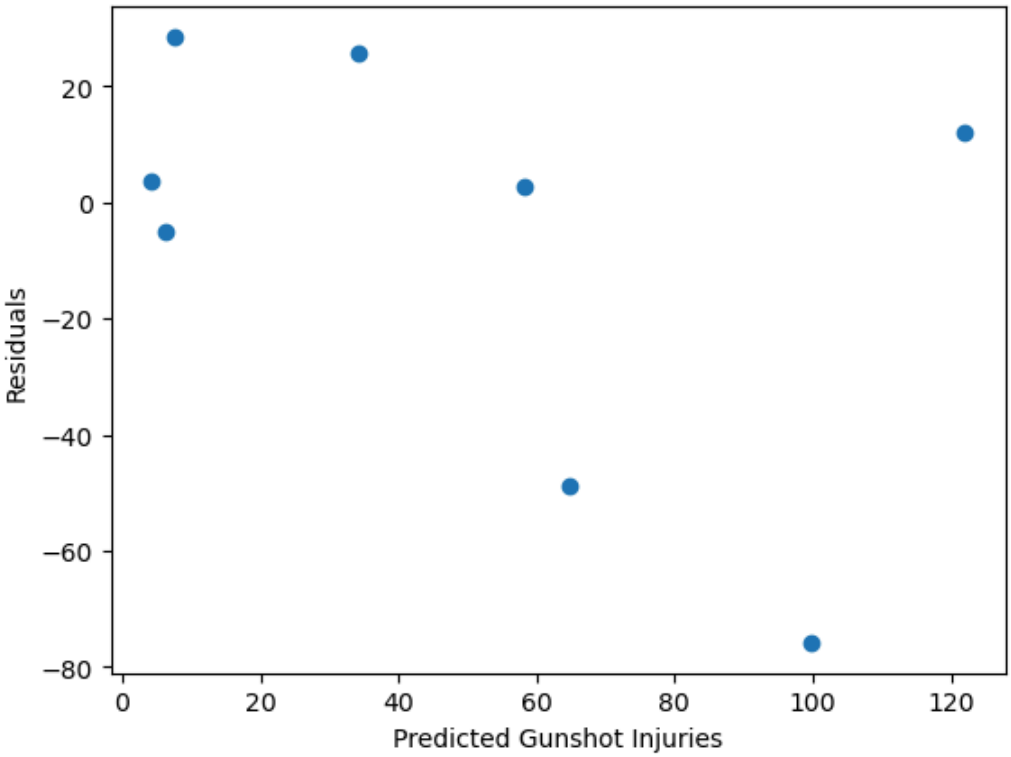
To interpret the model, an increase in one unit of AverageRank (the input) results in a corresponding increase of the coefficient value in output. This means that for an increase in

AverageRank of 1 unit, the number of gunshot occurences is likely to increase by the coefficient, reiterating that positive linear relationship.

In the below example the function year_regression is run on data from 2011 - 2012. Because the model has a coefficient of 1.952 and an intercept of -21.145, it can be predicted that for an increase in AverageRank of 1 unit, the number of gunshot occurences is likely to increase by 1.952, and an AverageRank input of value 0 yields a score of -21.145. The negative intercept and its interpretation will be discussed later in the model limitations section. If the AverageRank input value is equal to 20, for example, this model predicts that the corresponding frequency of gunshot violence occurrences will be -21.145 + 20*(1.952), or 17.895.

In [328…
```
year_regression(Test_Gunshots_11_12)
```

```
The model's coefficient is 1.952
The model's intercept is -21.145
This model's root mean squared error for the test set is 24.393
This model's mean absolute error for the test set is 19.297
RMSE scores across 6 folds: 51.813
MAE scores across 6 folds: 41.545
This model predicts that, for a 1 unit increase in AverageRank, the number of gunsh
ot injuries increases by 1.952
```



Limitations:

In this linear regression model, a negative intercept can sometimes be generated. This is not logical as it indicates that a low average rank (high test scores) will be associated with negative gun violence, which is not possible in the real world. Therefore, the model is restricted in the range of values that it can take as inputs for academic ranking. Additionally, because each year has a limited number of zipcodes, there were therefore a limited number of datapoints, resulting in the model being trained on fewer data points and tested on even fewer. For a model to be better fitted, it would be more ideal to have more data points to train on. Finally, alternative models could do well to include other predictor variables which could potentially help with analysis, but this model was specifically designed to test the impact of educational attainment on gunshot frequency.

Hypothesis 3

Null Hypothesis:

There is no relationship between one's race, age group, and zipcode in affecting their likelihood of experiencing gun violence. This will indicate that the coefficients for all predictor variables will be 0, indicating that there is no relationship between race, age-group, and zipcode and the likelihood of experiencing gun violence.

Alternative Hypothesis:

There is a significant relationship between one's race, age group, or zipcode that affects their likelihood of experiencing gun violence. In particular, predictor variables with coefficients not equal to 0 with significance values of $p < 0.05$ will allow a conclusion that certain races, age groups, and zipcodes may play a role in affecting gun violence likelihood. The null hypothesis is rejected if there is at least one predictor from race, one predictor from age group, one predictor from zipcode whose coefficients are not equal to 0 and whose p values are less than 0.05, and if the model as a whole is significant, as shown from an F-test.

Explanation for Hypothesis:

It is hypothesized that at least one of these predictors from each category (race, zipcode, age group) will be significant due to the EDA conducted prior to starting the analysis. The analysis showed that specific races, zipcodes, and age groups had disproportionately higher rates of gunviolence than others.

Overview:

First, using the dataframes created in Phase II, new dataframes were derived and prepared for use in analysis. Using derived dataframes, dummies were created, multiple different models were tested and residual plots before finalizing a decision to use a Negative Binomial Regression Model. The model was then interpreted and summarized, using t-tests for individual coefficients and an F-test for the model as a whole, and evaluation metrics were applied. Exploring this hypothesis required substantial data aggregation and analytical decision-making (such as whether to use averages and methods to group the data by) and model exploration.

Data Aggregation:

A dataframe was created using data from the years 2023, 2021, 2018, 2016, 2015, 2013, 2012, and 2011. This years were chosen due to the unavailability of educational data in other years. Thus, eventhough data for gunshot injuries was available for every year, the lack of availability of educational data limited the scope of data being used (discussed further in limitations). The dataframe created is also filtered out for males and gunshot injuries being 1 to address the requirements of the sub-question.

In [333…

```
DataSetH3 = data2023[['Zip Code','Sex', 'Race', 'Age','Year', 'Gunshot Injury I']]
DataSetH3 = pd.concat([
    DataSetH3,
    data2021[['Zip Code','Sex', 'Race', 'Age', 'Gunshot Injury I', 'Year']],
    data2018[['Zip Code','Sex', 'Race', 'Age', 'Gunshot Injury I', 'Year']],
    data2016[['Zip Code','Sex', 'Race', 'Age', 'Gunshot Injury I', 'Year']],
    data2015[['Zip Code','Sex', 'Race', 'Age', 'Gunshot Injury I', 'Year']],
    data2013[['Zip Code','Sex', 'Race', 'Age', 'Gunshot Injury I', 'Year']],
    data2012[['Zip Code','Sex', 'Race', 'Age', 'Gunshot Injury I', 'Year']],
```

```
    data2011[['Zip Code','Sex', 'Race', 'Age', 'Gunshot Injury I', 'Year']]
])
```

In [334…
```
DataSetH3 = DataSetH3[DataSetH3['Sex'] == 'M']
DataSetH3 = DataSetH3[DataSetH3['Gunshot Injury I'] == 1]
```

In [335…
```
mergedZipcodesTestRank = mergedZipcodesTestRank.set_index(["Zip"])
mergedZipcodesTestRankedAverage = \
mergedZipcodesTestRank.mean(axis=1)
mergedZipcodesTestRankedAverage = \
pd.DataFrame(mergedZipcodesTestRankedAverage)
mergedZipcodesTestRankedAverage = \
mergedZipcodesTestRankedAverage.reset_index()
mergedZipcodesTestRankedAverage = \
mergedZipcodesTestRankedAverage.rename(columns = {"Zip" : "Zip Code"})
mergedZipcodesTestRankedAverage["Zip Code"] = \
mergedZipcodesTestRankedAverage["Zip Code"].apply(lambda x: float(x))
mergedZipcodesTestRankedAverage = \
mergedZipcodesTestRankedAverage.rename(columns={0 : "AR"})
```

In [336…
```
DataSetH3
```

Out[336…

	Zip Code	Sex	Race	Age	Year	Gunshot Injury I
0	60641.0	M	White Hispanic	40-49	2023	1
1	60617.0	M	Black	0-19	2023	1
2	60619.0	M	Asian and Pacific Islander	20-29	2023	1
3	60649.0	M	Black	0-19	2023	1
4	60608.0	M	White	50-59	2023	1
…	…	…	…	…	…	…
2734	60609.0	M	Black	50-59	2011	1
2735	60643.0	M	Black	20-29	2011	1
2736	60651.0	M	Black	40-49	2011	1
2737	60621.0	M	Black	30-39	2011	1
2739	60644.0	M	Black	20-29	2011	1

22104 rows × 6 columns

Using the dataset above, a new dataset is created which counts the number of gunshot injuries grouped by zipcode, race, year, and age. The count of each data point is recorded and listed as "Occurrence".

In [338…
```
DataSetH3Grouped \
    = db.sql('''SELECT "Zip Code",\
              "Race", "Age","Year", COUNT(*) AS\
              "Occurrence" FROM DataSetH3 GROUP BY "Zip Code", \
             "Race", "Age", "Year" ORDER BY "Zip Code", "Race", "Age", \
             "Year"''').df()
```

In [339…
```
DataSetH3Grouped
```

Out[339…

	Zip Code	Race	Age	Year	Occurrence
0	60601.0	Black	0-19	2018	1
1	60601.0	Black	0-19	2021	1
2	60601.0	Black	20-29	2023	1
3	60601.0	Black	30-39	2018	1
4	60601.0	Black	30-39	2021	2
...	...	...	...	...	...
2896	60827.0	Unknown	20-29	2018	2
2897	60827.0	White	20-29	2011	2
2898	60827.0	White	40-49	2011	1
2899	60827.0	White	UNKNOWN	2011	1
2900	60827.0	White Hispanic	30-39	2011	1

2901 rows × 5 columns

In [340…

```
DataSetH3Grouped = pd.merge\
(DataSetH3Grouped, \
 mergedZipcodesTestRankedAverage, how = 'inner', \
    on = 'Zip Code')
```

A function to generate residual plots is also created for ease of use.

In [342…

```
def generate_residual_plot(pred, resid):
    scatterPlotPred = \
          sns.scatterplot(x = pred, y = resid, marker="o")
    scatterPlotPred.set_xlabel("Predicted Occurrence")
    scatterPlotPred.set_ylabel("Residuals")
    plt.axhline(y = 0, color = "black")

    return scatterPlotPred
```

Dummy Creation:

Zipcodes, Age Groups, Years, and Race are categorical data, and hence must be converted to dummy form in order to be used in the model. The reference variable is a white male in the age group 80+ residing in the zipcode 60605.0 in the year 2011. The reference is thus "a white male of age 80+ residing in the zipcode 60605 in the year 2011". These specific variables were chosen as the zipcode 60605 appeared to have the lowest average occurrence of gunshot injuries over 8 years at 1.25. Similarly, the age group 80+ appeared to have the lowest average occurrence of gunshot injuries as compared to other age groups. The race and year were arbitrarily chosen.

In [344…

```
zipcodeDummy = pd.get_dummies(DataSetH3Grouped['Zip Code'], \
dtype=int, drop_first=False)
zipcodeDummy = zipcodeDummy.drop(columns = [60605.0])
raceDummy = pd.get_dummies(DataSetH3Grouped['Race'], \
dtype=int, drop_first=False)
raceDummy = raceDummy.drop(columns = ['White'])
ageDummy = pd.get_dummies(DataSetH3Grouped['Age'], \
dtype=int, drop_first=False)
ageDummy=ageDummy.drop(columns = ['80+'])
yearDummy = pd.get_dummies(DataSetH3Grouped['Year'],\
                           dtype=int, drop_first=False)
yearDummy = yearDummy.drop(columns=2011)
```

The dummy variables are concatenated together with average test rank to create an input dataframe named X. The output variable is average gunshot occurrence, named y.

In [346…
```
X = pd.concat([raceDummy, zipcodeDummy, ageDummy, \
DataSetH3Grouped['AR'], yearDummy], axis = 1)
y = DataSetH3Grouped['Occurrence']
```

In [347…
```
X
```

Out[347…

	American Indian	Asian and Pacific Islander	Black	Black Hispanic	Unknown	White Hispanic	60607.0	60608.0	60609.0
0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...
2399	0	0	0	0	0	0	0	0	0
2400	0	0	0	0	0	0	0	0	0
2401	0	0	0	0	0	1	0	0	0
2402	0	0	0	0	0	1	0	0	0
2403	0	0	0	0	0	1	0	0	0

2404 rows × 60 columns

In [348…
```
y
```

Out[348…
```
0       1
1       2
2       1
3       1
4       2
       ..
2399    1
2400    1
2401    1
2402    2
2403    1
Name: Occurrence, Length: 2404, dtype: int64
```

Model Creation Process:

The initial reasoning (as provided in Phase III) was to use a multinomial logistic regression to predict the likelihood of being a victim of a gunshot injury. However, after additional research, it was concluded that such a model may not be conducive towards the outcome of the experiment. This was due to a variety of reasons. The first reason was that logistic regression outputs probabilities based on a binary value (1 or 0). In the case of the dataset being used, this proved to be unrealistic. This was because the dataset only included data on crimes involving gunshot injuries, and not other forms of crime. It hence became difficult to conclude what "0" would entail and what "1" would entail. Furthermore, since a logistic regression is supposed to measure a

binary value, it could not be applied to the dataset as the data itself was not encoded in 0s and 1s.

Hence, a renewed effort was undertaken to identify the best model to use for the data.

At first, an Ordinary Least Squares Regression Method was implemented. However, the resulting residual plot showed a positive linear trend between the residuals and test values. The output data was then log transformed and an Ordinary Least Squares regression was once again fit. However, this resulted in an even stronger positive linear trend between the residuals and test values.

Hence, with further research, it was decided to use a model which specifically modeled count data (as gunshot occurrences could be seen as akin to being a count). Hence, a Generalized Linear Model was implemented with the underlying distribution resembling that of a Poisson Distribution. This was a substantial improvement from the OLS models, but the residual plot still showed the residuals to be heteroskedastic. This was because perhaps the mean of the output data was not equal to its variance, as is expected in a poisson distribution. Next, a Negative Binomial Regression was used as the distribution is more flexible when it comes to recognizing variance. This led to the residual plot being more random, with a little heteroskedasticity. Lastly, a weight was applied to all training points, to reveal a model which fit the data the best out of all of the previous ones. This section walks the reader through this process.

The following section walks the reader through the model creation process and provides additional explanation.

First, a train test split was created with the test size being 0.3. An intercept is added as Statsmodels packages do not automatically add intercepts.

In [351…
```
XCopy = X
XCopy = XCopy.assign(intercept = 1)
X_train, X_test, y_train, y_test = \
train_test_split(X, y, test_size=0.3)
```

Now, an OLS model is trained and fit to the data. It was first thought to use an OLS as it minimizes the sum of squared differences between observed and predicted values. A key assumption when using OLS is that the the variance of the error term is the same for all values (i.e. the data is homoskedastic), there is a linear trend between the input variable and the output variable, and that the errors are normally distributed. The following model creation checks for these.

In [353…
```
modOLS = sm.OLS(y_train, X_train)
modOLSfit = modOLS.fit()
```

In [354…
```
print(modOLSfit.summary())
```

OLS Regression Results

Dep. Variable:	Occurrence	R-squared:	0.430
Model:	OLS	Adj. R-squared:	0.410
Method:	Least Squares	F-statistic:	21.11
Date:	Mon, 09 Dec 2024	Prob (F-statistic):	6.47e-157
Time:	20:19:07	Log-Likelihood:	-6482.2
No. Observations:	1682	AIC:	1.308e+04
Df Residuals:	1623	BIC:	1.340e+04
Df Model:	58		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
American Indian	-7.6560	8.323	-0.920	0.358	-23.981	8.669
Asian and Pacific Islander	-0.3370	1.982	-0.170	0.865	-4.224	3.550
Black	13.2126	0.972	13.594	0.000	11.306	15.119
Black Hispanic	-4.8873	1.580	-3.094	0.002	-7.986	-1.789
Unknown	-5.1861	1.548	-3.351	0.001	-8.222	-2.150
White Hispanic	3.5248	1.002	3.517	0.000	1.559	5.490
60607.0	17.9431	5.843	3.071	0.002	6.483	29.403
60608.0	435.3063	73.226	5.945	0.000	291.680	578.933
60609.0	515.7217	86.020	5.995	0.000	347.000	684.443
60610.0	88.5588	16.899	5.241	0.000	55.413	121.704
60612.0	425.0918	71.386	5.955	0.000	285.074	565.109
60613.0	145.4570	27.068	5.374	0.000	92.364	198.550
60614.0	23.0548	6.521	3.535	0.000	10.264	35.845
60615.0	347.8475	59.335	5.862	0.000	231.466	464.229
60616.0	530.0217	90.012	5.888	0.000	353.469	706.574
60617.0	482.5379	80.521	5.993	0.000	324.602	640.474
60618.0	159.4761	27.674	5.763	0.000	105.195	213.757
60619.0	616.6602	102.201	6.034	0.000	416.200	817.120
60620.0	409.2210	67.122	6.097	0.000	277.566	540.876
60621.0	599.8635	98.815	6.071	0.000	406.044	793.683
60622.0	447.5250	75.796	5.904	0.000	298.857	596.193
60623.0	431.4286	71.403	6.042	0.000	291.378	571.480
60624.0	481.7337	79.069	6.093	0.000	326.645	636.823
60625.0	113.0447	20.414	5.538	0.000	73.003	153.086
60626.0	308.1474	53.150	5.798	0.000	203.898	412.397

60628.0	466.5369	76.578	6.092	0.000	316.334 616.739
60629.0	411.9946	68.840	5.985	0.000	276.970 547.019
60631.0	-1.317e-12	2.19e-13	-6.021	0.000	-1.75e-12 -8.88e-13
60632.0	242.0435	40.810	5.931	0.000	161.997 322.090
60634.0	193.9650	33.544	5.782	0.000	128.171 259.759
60636.0	375.1669	61.534	6.097	0.000	254.473 495.861
60637.0	508.3880	84.171	6.040	0.000	343.292 673.484
60638.0	122.8043	22.282	5.511	0.000	79.099 166.510
60639.0	261.6553	44.430	5.889	0.000	174.508 348.802
60640.0	315.3515	54.248	5.813	0.000	208.948 421.755
60641.0	368.5677	62.303	5.916	0.000	246.365 490.770
60643.0	530.7136	89.280	5.944	0.000	355.597 705.830
60644.0	651.3582	107.524	6.058	0.000	440.457 862.260
60649.0	542.4867	90.480	5.996	0.000	365.017 719.957
60652.0	479.9059	81.641	5.878	0.000	319.773 640.039
60653.0	299.9297	50.964	5.885	0.000	199.967 399.893
60655.0	49.6773	13.998	3.549	0.000	22.222 77.133
60659.0	164.1776	29.151	5.632	0.000	107.001 221.354
60660.0	109.3838	20.516	5.332	0.000	69.142 149.625
0–19	20.9478	3.486	6.009	0.000	14.110 27.786
20–29	25.7083	3.473	7.401	0.000	18.895 32.521
30–39	18.1340	3.479	5.212	0.000	11.310 24.958
40–49	12.0427	3.495	3.446	0.001	5.187 18.898
50–59	8.2825	3.548	2.334	0.020	1.323 15.242
60–69	5.3001	3.657	1.449	0.147	-1.872 12.472
70–79	3.5111	4.228	0.830	0.406	-4.783 11.805
UNKNOWN	4.2622	4.456	0.956	0.339	-4.478 13.002
AR	-17.5049	2.922	-5.990	0.000	-23.237 -11.773
2012	1.8536	1.212	1.529	0.126	-0.524 4.232
2013	-0.3949	1.247	-0.317	0.751	-2.841 2.051
2015	0.8486	1.249	0.679	0.497	-1.601 3.298
2016	4.3772	1.182	3.704	0.000	2.059 6.695
2018	0.4416	1.217	0.363	0.717	-1.945 2.828
2021	3.9568	1.148	3.447	0.001	1.705

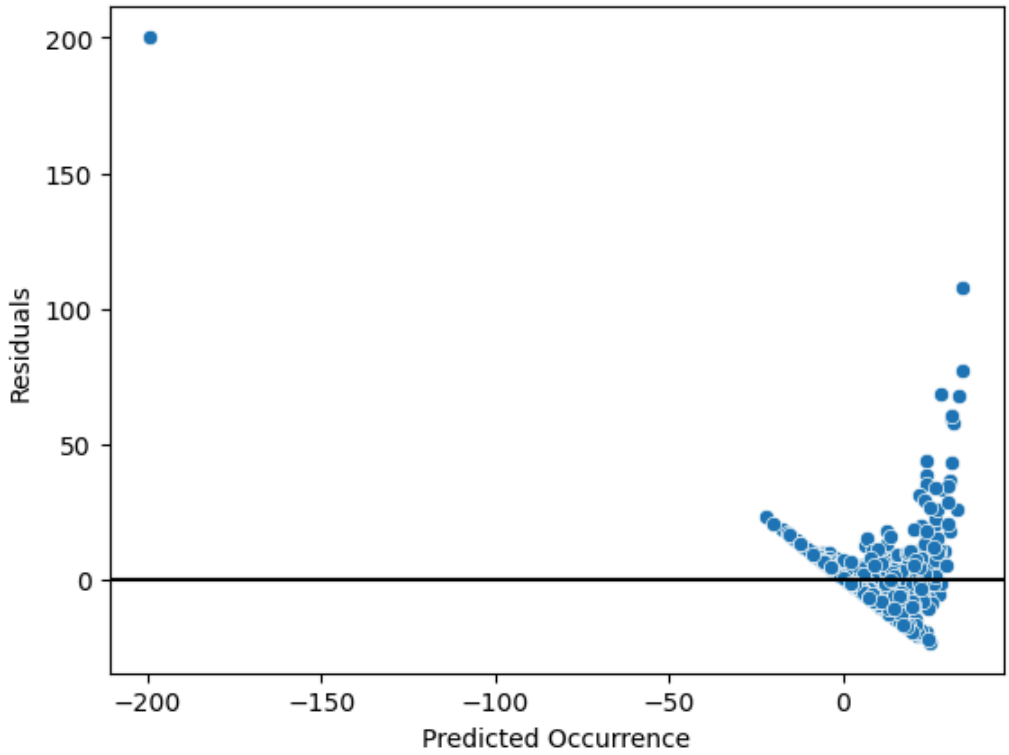
```
6.208
2023                          0.3558      1.186      0.300      0.764      -1.971
2.683
==============================================================================
Omnibus:                     1146.034   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            25034.344
Skew:                           2.877   Prob(JB):                         0.00
Kurtosis:                      21.003   Cond. No.                     1.07e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
[2] The smallest eigenvalue is  9e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Occurrences using X_test are predicted.

In [356…
```
ypredictOLS = modOLSfit.predict(X_test)
```

In [357…
```
generate_residual_plot(ypredictOLS, y_test-ypredictOLS)
```

Out[357… `<Axes: xlabel='Predicted Occurrence', ylabel='Residuals'>`



Note that the residual plot shows the plotted values to be highly heteroskedastic. This violates a key assumption in using OLS. Note that the data may also not have a linear trend. This is because the data features multiple outliers with substantially extreme values (as there are certain zipcodes and races in Chicago which experience substantially more gunshot injuries than other zipcodes as shown in the EDA). Nevertheless, an attempt will be made to log-transform the output values as this can potentially reduce the size of outliers. Note that the same train test split is used as an entirely new model is being created. Note that log transformations are also applied after splitting the data due to common practice, so as to avoid the chance of data leakage. Although in this case, it may not entirely matter whether the data is transformed before versus after splitting, it certainly matters when weights are being considered (as shown in further below cells).

In [359…

```
modlog = sm.OLS(np.log(y_train), X_train)
modlogfit = modlog.fit()
print(modlogfit.summary())
```

OLS Regression Results

Dep. Variable:	Occurrence	R-squared:	0.593
Model:	OLS	Adj. R-squared:	0.579
Method:	Least Squares	F-statistic:	40.83
Date:	Mon, 09 Dec 2024	Prob (F-statistic):	5.37e-272
Time:	20:19:07	Log-Likelihood:	-2042.3
No. Observations:	1682	AIC:	4203.
Df Residuals:	1623	BIC:	4523.
Df Model:	58		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
American Indian	-1.0735	0.594	-1.807	0.071	-2.239	0.092
Asian and Pacific Islander	-0.0411	0.141	-0.291	0.771	-0.319	0.236
Black	1.6492	0.069	23.770	0.000	1.513	1.785
Black Hispanic	-0.5065	0.113	-4.491	0.000	-0.728	-0.285
Unknown	-0.4895	0.110	-4.431	0.000	-0.706	-0.273
White Hispanic	0.8143	0.072	11.382	0.000	0.674	0.955
60607.0	2.1237	0.417	5.092	0.000	1.306	2.942
60608.0	49.1556	5.227	9.404	0.000	38.903	59.409
60609.0	58.1625	6.141	9.472	0.000	46.118	70.207
60610.0	10.4359	1.206	8.651	0.000	8.070	12.802
60612.0	47.9473	5.096	9.409	0.000	37.952	57.943
60613.0	16.3848	1.932	8.479	0.000	12.595	20.175
60614.0	2.4803	0.466	5.328	0.000	1.567	3.393
60615.0	39.3901	4.236	9.300	0.000	31.082	47.698
60616.0	59.7129	6.426	9.293	0.000	47.110	72.316
60617.0	54.3383	5.748	9.453	0.000	43.064	65.613
60618.0	18.0540	1.976	9.139	0.000	14.179	21.929
60619.0	68.9671	7.296	9.453	0.000	54.657	83.277
60620.0	45.4683	4.792	9.489	0.000	36.070	54.867
60621.0	66.8659	7.054	9.479	0.000	53.030	80.702
60622.0	50.4242	5.411	9.319	0.000	39.811	61.037
60623.0	48.4173	5.097	9.499	0.000	38.420	58.415
60624.0	53.4341	5.644	9.467	0.000	42.363	64.505
60625.0	12.8361	1.457	8.808	0.000	9.978	15.694
60626.0	34.8717	3.794	9.191	0.000	27.430	42.314

60628.0	51.8908	5.467	9.492	0.000	41.168 62.613
60629.0	46.4593	4.914	9.454	0.000	36.820 56.098
60631.0	-1.471e-13	1.56e-14	-9.420	0.000	-1.78e-13 -1.16e-13
60632.0	27.3044	2.913	9.372	0.000	21.590 33.019
60634.0	21.7004	2.395	9.062	0.000	17.004 26.397
60636.0	41.8309	4.393	9.523	0.000	33.215 50.447
60637.0	56.8040	6.009	9.454	0.000	45.018 68.589
60638.0	13.6384	1.591	8.574	0.000	10.518 16.758
60639.0	29.6906	3.172	9.361	0.000	23.470 35.912
60640.0	35.7636	3.873	9.235	0.000	28.168 43.359
60641.0	41.5004	4.448	9.331	0.000	32.777 50.224
60643.0	59.9094	6.373	9.400	0.000	47.409 72.410
60644.0	72.4923	7.676	9.444	0.000	57.437 87.548
60649.0	60.8717	6.459	9.424	0.000	48.203 73.541
60652.0	54.0907	5.828	9.281	0.000	42.659 65.522
60653.0	34.0636	3.638	9.363	0.000	26.928 41.200
60655.0	5.6522	0.999	5.656	0.000	3.692 7.612
60659.0	18.2792	2.081	8.784	0.000	14.198 22.361
60660.0	12.2399	1.465	8.357	0.000	9.367 15.113
0–19	2.4160	0.249	9.708	0.000	1.928 2.904
20–29	2.6840	0.248	10.825	0.000	2.198 3.170
30–39	2.1867	0.248	8.804	0.000	1.700 2.674
40–49	1.6121	0.250	6.461	0.000	1.123 2.101
50–59	1.2166	0.253	4.803	0.000	0.720 1.713
60–69	0.7031	0.261	2.694	0.007	0.191 1.215
70–79	0.3508	0.302	1.162	0.245	-0.241 0.943
UNKNOWN	0.3294	0.318	1.036	0.301	-0.294 0.953
AR	-1.9627	0.209	-9.409	0.000	-2.372 -1.554
2012	0.0635	0.087	0.733	0.463	-0.106 0.233
2013	-0.0745	0.089	-0.837	0.403	-0.249 0.100
2015	0.0619	0.089	0.694	0.488	-0.113 0.237
2016	0.1658	0.084	1.966	0.049	0.000 0.331
2018	-0.0656	0.087	-0.756	0.450	-0.236 0.105
2021	0.2654	0.082	3.239	0.001	0.105

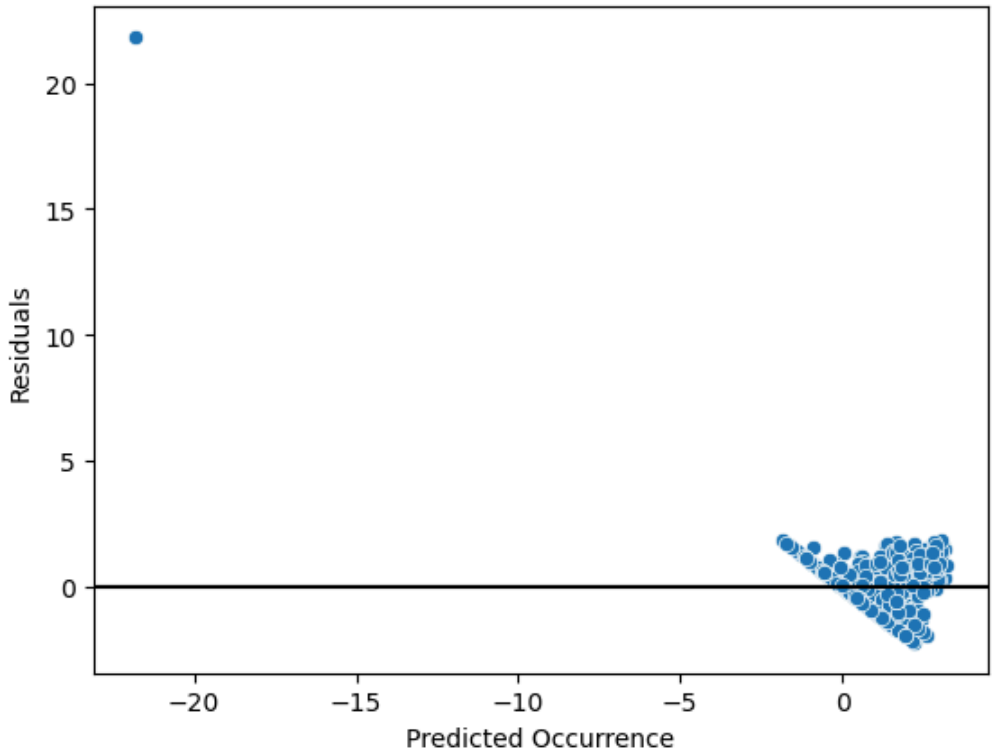
```
0.426
2023                          0.0587      0.085      0.693      0.489      -0.107
0.225
==============================================================================
Omnibus:                       13.572   Durbin-Watson:                   1.923
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               12.358
Skew:                          -0.163   Prob(JB):                      0.00207
Kurtosis:                       2.736   Cond. No.                     1.07e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
[2] The smallest eigenvalue is  9e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

The log output is now predicted using X_test

In [361…
```
ypredlog = modlogfit.predict(X_test)
```

In [362…
```
generate_residual_plot(ypredlog, np.log(y_test)-ypredlog)
```

Out[362… `<Axes: xlabel='Predicted Occurrence', ylabel='Residuals'>`



The residual plot appears to have the same level of randomness as the previous residual plot. It is hence clear that applying a log-transform is not feasible in this case. Hence, in both cases, certain key assumptions have been violated which make it not feasible to use an OLS model. A decision was hence made to use a "Generalized Linear Model". GLMs provide substantial flexibility with modeling data as compared to OLS models. There are numerous reasons why. Firstly, GLMs do not require the response and errors to be normally distributed. Secondly, there are a wide variety of families which can be used with GLM (such as Poisson and Negative Binomial as described below). Thirdly, maximum likelihood estimation is used in fitting the model. This means that even if the underlying data is not normally distributed, as the sample size increases (in the case of this data, it is a population with size 2404), estimated model coefficients begin to approach a normal distribution. Since the size of the data's population is large, the writers of this report can rely on the Central Limit Theorem and make use of the GLM.

A Poisson distribution GLM is first explored. This is because "Occurrences" are in fact count data. This was indeed another issue with using OLS. Poisson distributions on the other hand are specifically accustomed to count data.

In [365…

```
X_train, X_test, y_train, y_test = train_test_split(XCopy, y, test_size=0.3)
modPoissonGLM = sm.GLM(y_train, X_train, family = sm.families.Poisson())
modPoissonGLMfit = modPoissonGLM.fit()
print(modPoissonGLMfit.summary())
```

Generalized Linear Model Regression Results

Dep. Variable:	Occurrence	No. Observations:	1682
Model:	GLM	Df Residuals:	1622
Model Family:	Poisson	Df Model:	59
Link Function:	Log	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-5064.9
Date:	Mon, 09 Dec 2024	Deviance:	4930.0
Time:	20:19:08	Pearson chi2:	5.46e+03
No. Iterations:	7	Pseudo R-squ. (CS):	1.000
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
American Indian	-0.8868	0.711	-1.247	0.212	-2.280	0.507
Asian and Pacific Islander	-0.4343	0.155	-2.810	0.005	-0.737	-0.131
Black	2.3645	0.062	37.872	0.000	2.242	2.487
Black Hispanic	-0.5983	0.123	-4.877	0.000	-0.839	-0.358
Unknown	-0.7192	0.119	-6.040	0.000	-0.953	-0.486
White Hispanic	1.3085	0.065	20.228	0.000	1.182	1.435
60607.0	0.1731	0.220	0.787	0.431	-0.258	0.604
60608.0	-0.0439	0.053	-0.829	0.407	-0.148	0.060
60609.0	0.1059	0.055	1.930	0.054	-0.002	0.213
60610.0	0.2348	0.191	1.227	0.220	-0.140	0.610
60612.0	0.2643	0.055	4.814	0.000	0.157	0.372
60613.0	-1.0956	0.277	-3.959	0.000	-1.638	-0.553
60614.0	-0.2224	0.273	-0.814	0.416	-0.758	0.313
60615.0	-0.1180	0.073	-1.615	0.106	-0.261	0.025
60616.0	-1.7972	0.119	-15.052	0.000	-2.031	-1.563
60617.0	0.0935	0.049	1.915	0.056	-0.002	0.189
60618.0	0.3739	0.141	2.658	0.008	0.098	0.650
60619.0	0.0903	0.090	1.001	0.317	-0.087	0.267
60620.0	1.0602	0.039	27.254	0.000	0.984	1.136
60621.0	0.2023	0.081	2.499	0.012	0.044	0.361
60622.0	-0.7429	0.081	-9.176	0.000	-0.902	-0.584
60623.0	0.7911	0.037	21.303	0.000	0.718	0.864
60624.0	0.8697	0.039	22.249	0.000	0.793	0.946
60625.0	0.1491	0.176	0.846	0.397	-0.196	0.494
60626.0	-0.4573	0.102	-4.495	0.000	-0.657	-0.258

60628.0	0.8768	0.038	23.286	0.000	0.803 0.951
60629.0	0.3575	0.054	6.661	0.000	0.252 0.463
60631.0	0.3192	0.999	0.320	0.749	-1.639 2.278
60632.0	1.1073	0.099	11.139	0.000	0.912 1.302
60634.0	0.0839	0.184	0.456	0.648	-0.277 0.444
60636.0	1.1578	0.047	24.782	0.000	1.066 1.249
60637.0	0.3791	0.053	7.111	0.000	0.275 0.484
60638.0	-0.3250	0.231	-1.408	0.159	-0.777 0.127
60639.0	0.5335	0.091	5.832	0.000	0.354 0.713
60640.0	-0.3715	0.099	-3.767	0.000	-0.565 -0.178
60641.0	-0.2968	0.104	-2.840	0.005	-0.502 -0.092
60643.0	-0.5214	0.076	-6.867	0.000	-0.670 -0.373
60644.0	-0.1152	0.102	-1.129	0.259	-0.315 0.085
60649.0	-0.0884	0.067	-1.321	0.186	-0.220 0.043
60652.0	-1.3349	0.093	-14.300	0.000	-1.518 -1.152
60653.0	0.3388	0.086	3.930	0.000	0.170 0.508
60655.0	-1.7794	1.013	-1.757	0.079	-3.764 0.205
60659.0	-0.5702	0.218	-2.621	0.009	-0.997 -0.144
60660.0	-0.4692	0.236	-1.985	0.047	-0.933 -0.006
0-19	3.1737	0.290	10.962	0.000	2.606 3.741
20-29	3.7444	0.289	12.940	0.000	3.177 4.312
30-39	2.9839	0.290	10.302	0.000	2.416 3.552
40-49	2.0945	0.291	7.208	0.000	1.525 2.664
50-59	1.3616	0.294	4.634	0.000	0.786 1.937
60-69	0.7363	0.302	2.439	0.015	0.145 1.328
70-79	0.1964	0.354	0.555	0.579	-0.498 0.890
UNKNOWN	0.2298	0.373	0.616	0.538	-0.502 0.961
AR	0.0784	0.008	10.104	0.000	0.063 0.094
2012	0.1377	0.035	3.900	0.000	0.068 0.207
2013	-0.1098	0.036	-3.051	0.002	-0.180 -0.039
2015	0.0511	0.036	1.434	0.152	-0.019 0.121
2016	0.4215	0.032	13.312	0.000	0.359 0.484
2018	-0.0154	0.035	-0.436	0.663	-0.085 0.054
2021	0.3642	0.032	11.355	0.000	0.301

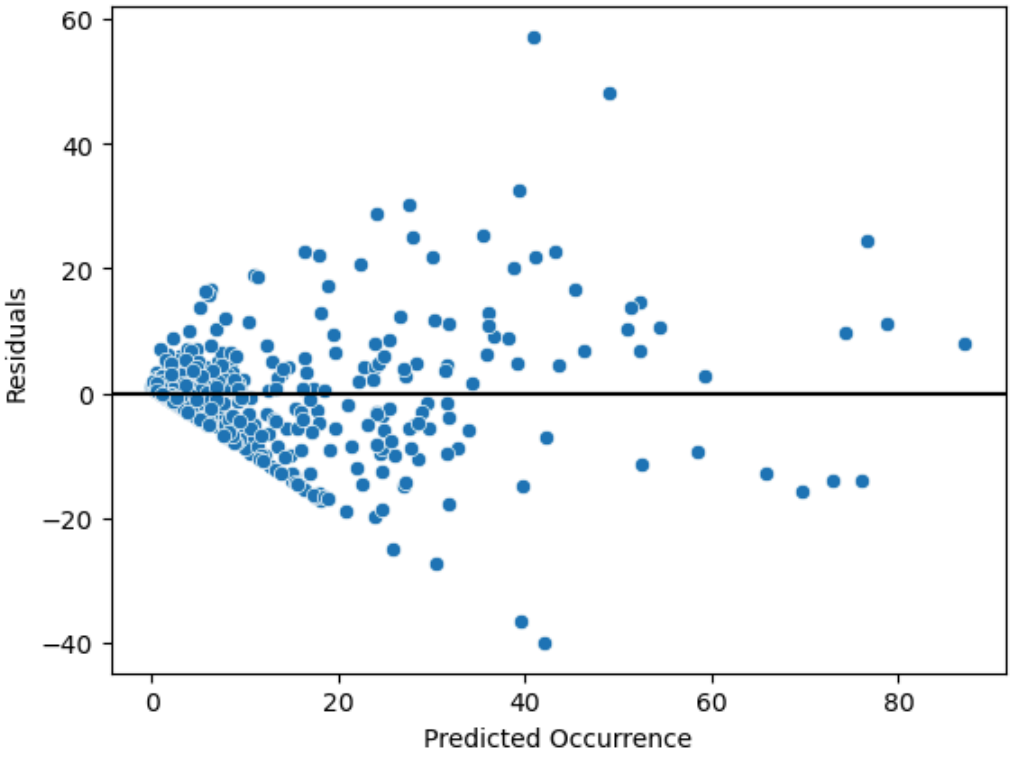
```
0.427
2023                          -0.0383       0.034     -1.121       0.262       -0.105
0.029
intercept                     -5.0660       0.357    -14.189       0.000       -5.766
-4.366
==============================================================================
===========
```

The output is now predicted for X_test

In [367…
```
ypredpoissonglm = modPoissonGLMfit.predict(X_test)
```

In [368…
```
generate_residual_plot(ypredpoissonglm, y_test - ypredpoissonglm)
```

Out[368… `<Axes: xlabel='Predicted Occurrence', ylabel='Residuals'>`



An analysis of the residual plot reveals the variance of the residuals to be more uniform as compared to when using OLS. However, the data is still heteroskedastic, as shown by the funnel shape. There may be a variety of reasons for this. Firstly, a poisson distribution assumes that the mean of the data is equal to its variance. In this case, the variance is substantially different from the mean, which indicates that the underlying distribution of the data is not poisson. This also indicates that the data is "over-dispersed". Refer to the code below.

In [370…
```
print(f"The expected value of the data is {np.mean(y)}")
print(f"The variance of the data is {np.var(y)}")
```

```
The expected value of the data is 8.418885191347753
The variance of the data is 231.75839542941975
```

A Negative Binomial Regression is now used. This is because a Negative Binomial Distribution does not require variance to be equal to mean. It is thus more robust to deal with overdispersed data (refer to above explanation).

In [372…
```
modnegbinGLM = sm.GLM(y_train, X_train, family = sm.families.NegativeBinomial())
modnegbinGLMfit = modnegbinGLM.fit()
print(modnegbinGLMfit.summary())
```

Generalized Linear Model Regression Results

Dep. Variable:	Occurrence	No. Observations:	1682
Model:	GLM	Df Residuals:	1622
Model Family:	NegativeBinomial	Df Model:	59
Link Function:	Log	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-4347.6
Date:	Mon, 09 Dec 2024	Deviance:	679.11
Time:	20:19:08	Pearson chi2:	705.
No. Iterations:	13	Pseudo R-squ. (CS):	0.7028
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
American Indian	-0.9901	1.013	-0.977	0.328	-2.975	0.995
Asian and Pacific Islander	-0.1626	0.231	-0.704	0.481	-0.615	0.290
Black	1.9036	0.107	17.757	0.000	1.693	2.114
Black Hispanic	-0.6261	0.185	-3.387	0.001	-0.988	-0.264
Unknown	-0.6892	0.184	-3.742	0.000	-1.050	-0.328
White Hispanic	1.0599	0.110	9.603	0.000	0.844	1.276
60607.0	0.2941	0.400	0.736	0.462	-0.489	1.078
60608.0	0.1221	0.142	0.858	0.391	-0.157	0.401
60609.0	0.1160	0.135	0.858	0.391	-0.149	0.381
60610.0	0.2062	0.380	0.542	0.588	-0.539	0.952
60612.0	0.1850	0.169	1.092	0.275	-0.147	0.517
60613.0	-0.8352	0.435	-1.920	0.055	-1.688	0.017
60614.0	0.0116	0.443	0.026	0.979	-0.857	0.880
60615.0	-0.1081	0.187	-0.577	0.564	-0.475	0.259
60616.0	-1.4411	0.230	-6.258	0.000	-1.892	-0.990
60617.0	-0.0035	0.143	-0.024	0.981	-0.285	0.278
60618.0	0.6098	0.260	2.342	0.019	0.099	1.120
60619.0	0.0394	0.208	0.190	0.849	-0.368	0.447
60620.0	0.8643	0.152	5.668	0.000	0.565	1.163
60621.0	0.1226	0.198	0.618	0.537	-0.266	0.512
60622.0	-0.5003	0.171	-2.929	0.003	-0.835	-0.165
60623.0	0.6522	0.125	5.232	0.000	0.408	0.897
60624.0	0.5714	0.135	4.242	0.000	0.307	0.835
60625.0	0.3019	0.313	0.963	0.335	-0.312	0.916
60626.0	-0.3891	0.209	-1.865	0.062	-0.798	0.020

60628.0	0.6387	0.146	4.384	0.000	0.353 0.924
60629.0	0.3157	0.149	2.116	0.034	0.023 0.608
60631.0	−0.0778	1.420	−0.055	0.956	−2.861 2.706
60632.0	1.1046	0.216	5.102	0.000	0.680 1.529
60634.0	0.1277	0.325	0.392	0.695	−0.510 0.766
60636.0	0.9676	0.164	5.891	0.000	0.646 1.289
60637.0	0.2314	0.168	1.377	0.169	−0.098 0.561
60638.0	−0.0824	0.377	−0.219	0.827	−0.821 0.656
60639.0	0.6330	0.190	3.330	0.001	0.260 1.006
60640.0	−0.2014	0.220	−0.917	0.359	−0.632 0.229
60641.0	−0.1706	0.197	−0.867	0.386	−0.556 0.215
60643.0	−0.4309	0.199	−2.166	0.030	−0.821 −0.041
60644.0	−0.2842	0.224	−1.270	0.204	−0.723 0.155
60649.0	−0.1272	0.183	−0.696	0.487	−0.486 0.231
60652.0	−1.1607	0.194	−5.978	0.000	−1.541 −0.780
60653.0	0.3931	0.219	1.799	0.072	−0.035 0.821
60655.0	−1.7492	1.444	−1.211	0.226	−4.580 1.081
60659.0	−0.2832	0.345	−0.822	0.411	−0.959 0.392
60660.0	−0.2862	0.403	−0.710	0.478	−1.077 0.504
0–19	2.9155	0.428	6.812	0.000	2.077 3.754
20–29	3.3216	0.427	7.771	0.000	2.484 4.159
30–39	2.6793	0.428	6.266	0.000	1.841 3.517
40–49	1.9548	0.429	4.552	0.000	1.113 2.796
50–59	1.3663	0.435	3.141	0.002	0.514 2.219
60–69	0.8281	0.447	1.852	0.064	−0.048 1.705
70–79	0.2502	0.518	0.483	0.629	−0.765 1.265
UNKNOWN	0.4675	0.544	0.859	0.390	−0.599 1.534
AR	0.0696	0.014	5.011	0.000	0.042 0.097
2012	0.0594	0.121	0.489	0.625	−0.179 0.297
2013	−0.1174	0.122	−0.965	0.335	−0.356 0.121
2015	−0.0085	0.119	−0.071	0.943	−0.242 0.225
2016	0.2341	0.114	2.056	0.040	0.011 0.457
2018	−0.0318	0.117	−0.271	0.786	−0.262 0.198
2021	0.3259	0.112	2.909	0.004	0.106

```
0.545
2023                          0.0304       0.113       0.268       0.789       -0.192
0.253
intercept                     -4.1323      0.571      -7.242       0.000       -5.251
-3.014
==============================================================================
===========
```

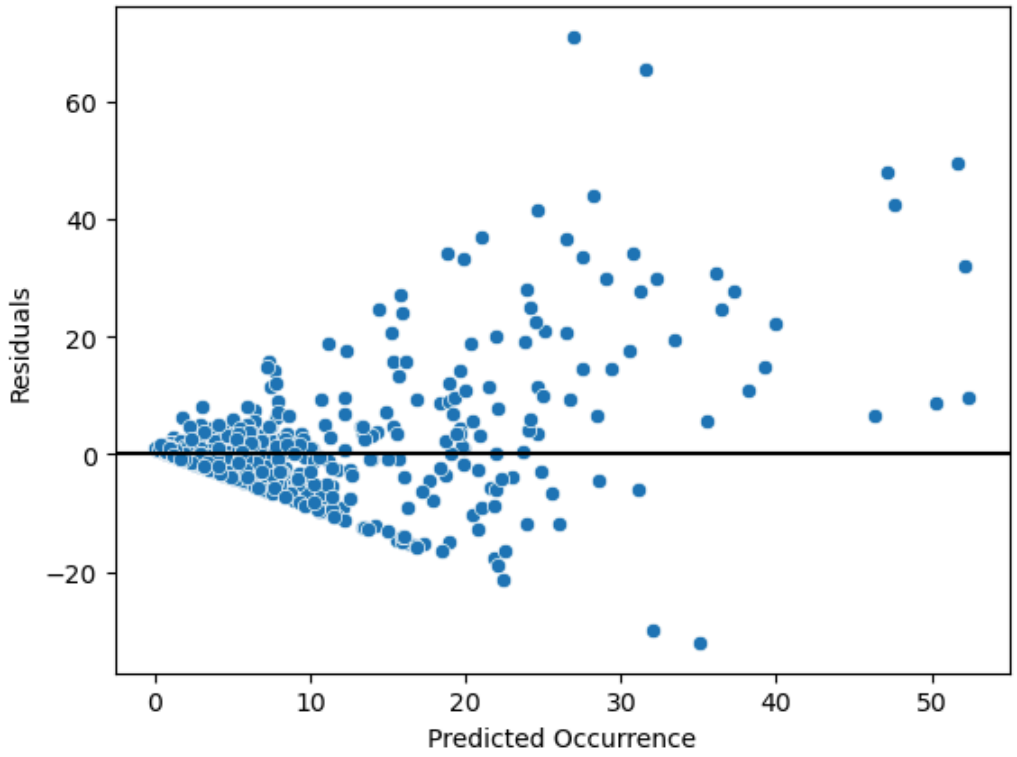
```
/opt/anaconda3/lib/python3.12/site-packages/statsmodels/genmod/families/family.py:1
367: ValueWarning: Negative binomial dispersion parameter alpha not set. Using defa
ult value alpha=1.0.
  warnings.warn("Negative binomial dispersion parameter alpha not "
```

The output data is now predicted using X_test

In [374…
```
yprednegbin = modnegbinGLMfit.predict(X_test)
```

In [375…
```
generate_residual_plot(yprednegbin, y_test-yprednegbin)
```

Out[375… <Axes: xlabel='Predicted Occurrence', ylabel='Residuals'>



On closer analysis, the residual plot looks less random than in the previous model, but still shows the data as heteroskedastic. This is because perhaps the Negative Binomial regression better handles overdispersed data, but that the variance is still substantially higher than the mean. After research, a conclusion was made to add "weights" to each of the output variables, so as to minimize the effect of outliers. This was chosen to be (1/(0.1 + (mean of y training data - occurrence value))). Thus, if a data point has a higher occurrence value from the mean occurrence, it will be penalized to a greater extent than a data point with an occurrence value that is higher than the mean occurrence but not as high as the other point. This ensures that the effect of outliers is reduced, and reduces overdispersion of the data being used to train the model. Note that 0.1 is present to serve as a buffer for values which may be extremely close to the mean. It reduces such values from significantly overpowering the regression model.

This decision was made as a logarithmic transformation was not possible to use in the Negative Binomial regression. Furthermore, this specific statistic was chosen as it would be able to minimize the effect of outliers and thus create a better fit. Note that since there is no universally accepted standard for what this "weight" could entail, it was arbitrarily decided to use the mean.

In [377…
```
weights = (1 / (0.1+(abs(((np.mean(y_train) - y_train))))))
```

In [378…
```
weights
```

Out[378…
```
2153    0.281403
844     0.132387
712     0.152587
2031    0.152587
4       0.152587
          ...
2033    0.150458
1970    0.132387
2393    0.132387
1635    0.219605
438     0.132387
Name: Occurrence, Length: 1682, dtype: float64
```

The Negative Binomial model is now created. The weights are inputted into the "freq_weights" category. Note that "var_weights" is not used instead as the endogenous variables do not reflect averages. Freq_weights is instead used as the endogenous variables are aggregated gunshots and indicate how many cases in the population a given observation represents. Lastly, freq_weights does not transform the output and instead maintains the output on the original scale, which makes interpretation simplistic.

In [380…
```
modnegbinweightsGLM = sm.GLM(y_train, X_train, family = \
sm.families.NegativeBinomial(), freq_weights=weights)
modnegbinweightsGLMfit = modnegbinweightsGLM.fit()
print(modnegbinweightsGLMfit.summary())
```

Generalized Linear Model Regression Results

Dep. Variable:	Occurrence	No. Observations:	1682
Model:	GLM	Df Residuals:	307.79
Model Family:	NegativeBinomial	Df Model:	59
Link Function:	Log	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-988.81
Date:	Mon, 09 Dec 2024	Deviance:	94.261
Time:	20:19:09	Pearson chi2:	89.4
No. Iterations:	9	Pseudo R-squ. (CS):	0.08669
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
American Indian	-0.8572	2.772	-0.309	0.757	-6.290	4.575
Asian and Pacific Islander	-0.1770	0.621	-0.285	0.776	-1.394	1.040
Black	1.7035	0.272	6.256	0.000	1.170	2.237
Black Hispanic	-0.5019	0.494	-1.016	0.310	-1.470	0.466
Unknown	-0.5369	0.487	-1.102	0.271	-1.492	0.418
White Hispanic	1.2006	0.273	4.395	0.000	0.665	1.736
60607.0	0.3463	0.903	0.384	0.701	-1.423	2.116
60608.0	0.1406	0.272	0.518	0.605	-0.392	0.673
60609.0	0.0713	0.287	0.248	0.804	-0.491	0.634
60610.0	0.4902	0.758	0.647	0.518	-0.995	1.975
60612.0	0.2160	0.327	0.662	0.508	-0.424	0.856
60613.0	-0.7884	1.133	-0.696	0.487	-3.009	1.433
60614.0	-0.2479	1.117	-0.222	0.824	-2.436	1.941
60615.0	0.0150	0.368	0.041	0.967	-0.706	0.736
60616.0	-0.9451	0.443	-2.131	0.033	-1.814	-0.076
60617.0	0.1018	0.259	0.393	0.694	-0.406	0.610
60618.0	0.6465	0.575	1.125	0.261	-0.480	1.773
60619.0	0.0539	0.443	0.122	0.903	-0.815	0.923
60620.0	0.6906	0.344	2.010	0.044	0.017	1.364
60621.0	0.0229	0.501	0.046	0.963	-0.959	1.004
60622.0	-0.1893	0.325	-0.582	0.561	-0.827	0.448
60623.0	0.5131	0.285	1.803	0.071	-0.045	1.071
60624.0	0.3082	0.321	0.961	0.337	-0.320	0.937
60625.0	0.2692	0.734	0.367	0.714	-1.170	1.708
60626.0	-0.1840	0.434	-0.424	0.671	-1.034	0.666

60628.0	0.3611	0.393	0.919	0.358	−0.409
1.131					
60629.0	0.2278	0.278	0.820	0.412	−0.317
0.773					
60631.0	−0.5926	3.890	−0.152	0.879	−8.217
7.032					
60632.0	0.8302	0.510	1.629	0.103	−0.169
1.829					
60634.0	0.0279	0.772	0.036	0.971	−1.486
1.542					
60636.0	0.6220	0.408	1.526	0.127	−0.177
1.421					
60637.0	0.1848	0.387	0.478	0.633	−0.573
0.943					
60638.0	−0.2330	0.939	−0.248	0.804	−2.073
1.607					
60639.0	0.4989	0.403	1.238	0.216	−0.291
1.289					
60640.0	−0.1333	0.475	−0.281	0.779	−1.065
0.798					
60641.0	−0.0657	0.407	−0.161	0.872	−0.863
0.731					
60643.0	−0.1596	0.413	−0.386	0.699	−0.970
0.651					
60644.0	−0.4351	0.548	−0.795	0.427	−1.508
0.638					
60649.0	−0.1111	0.467	−0.238	0.812	−1.026
0.804					
60652.0	−0.7688	0.374	−2.055	0.040	−1.502
−0.036					
60653.0	0.2597	0.441	0.589	0.556	−0.604
1.124					
60655.0	−1.5256	3.945	−0.387	0.699	−9.257
6.206					
60659.0	−0.4529	0.888	−0.510	0.610	−2.193
1.287					
60660.0	−0.4204	1.003	−0.419	0.675	−2.386
1.545					
0–19	2.5196	1.160	2.172	0.030	0.246
4.793					
20–29	2.7808	1.159	2.399	0.016	0.509
5.053					
30–39	2.4050	1.158	2.077	0.038	0.136
4.674					
40–49	1.9195	1.153	1.665	0.096	−0.341
4.179					
50–59	1.4389	1.164	1.236	0.217	−0.843
3.721					
60–69	0.8349	1.201	0.695	0.487	−1.520
3.189					
70–79	0.2077	1.404	0.148	0.882	−2.544
2.959					
UNKNOWN	0.4281	1.477	0.290	0.772	−2.467
3.323					
AR	0.0556	0.033	1.711	0.087	−0.008
0.119					
2012	−0.0185	0.257	−0.072	0.943	−0.522
0.485					
2013	−0.0953	0.254	−0.376	0.707	−0.592
0.402					
2015	−0.0538	0.244	−0.220	0.826	−0.533
0.425					
2016	0.0863	0.249	0.347	0.728	−0.401
0.574					
2018	−0.0592	0.244	−0.242	0.808	−0.538
0.419					
2021	0.1807	0.242	0.747	0.455	−0.293

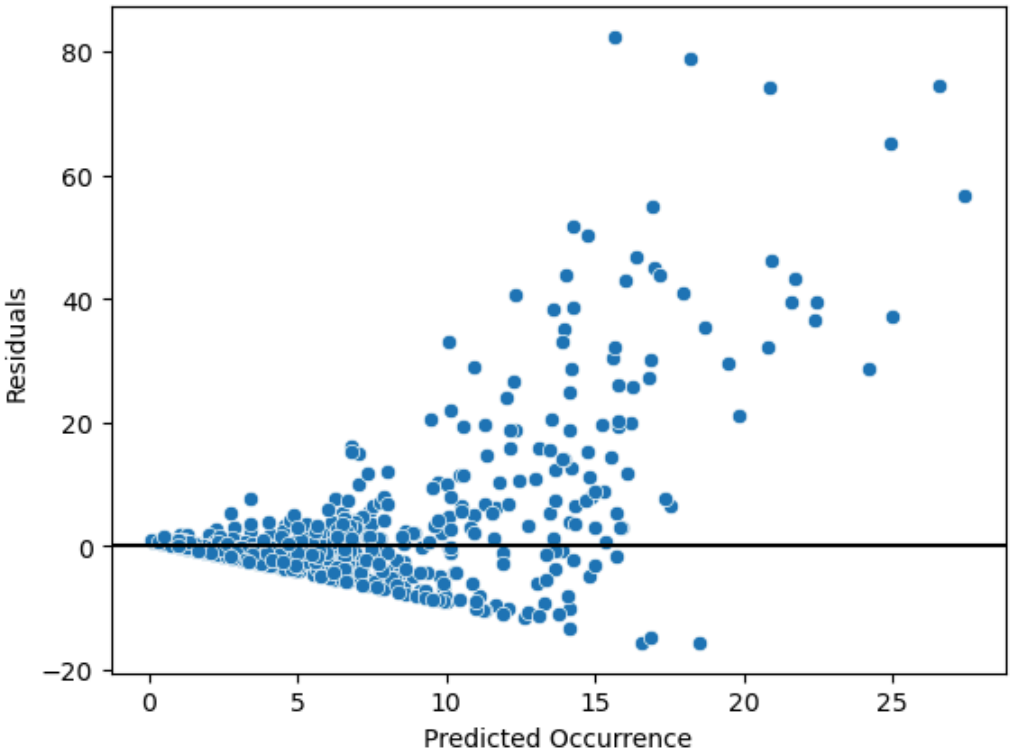
```
0.655
2023                          0.0208       0.244       0.085       0.932       -0.458
0.499
intercept                     -3.4036      1.462      -2.329       0.020       -6.268
-0.539
==============================================================================
===========
```

```
/opt/anaconda3/lib/python3.12/site-packages/statsmodels/genmod/families/family.py:1
367: ValueWarning: Negative binomial dispersion parameter alpha not set. Using defa
ult value alpha=1.0.
  warnings.warn("Negative binomial dispersion parameter alpha not "
```

In [381…
```
yprednegbinglm = modnegbinweightsGLMfit.predict(X_test)
```

In [382…
```
generate_residual_plot(yprednegbinglm, y_test-yprednegbinglm)
```

Out[382… `<Axes: xlabel='Predicted Occurrence', ylabel='Residuals'>`

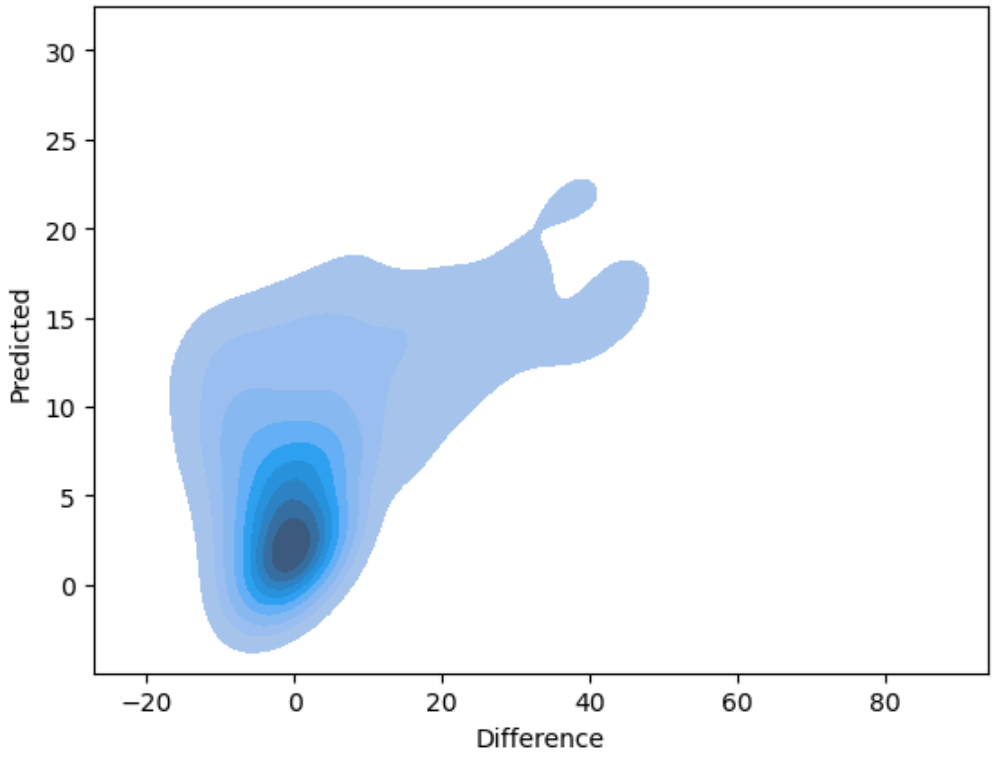


Note that the residual plot here is more random. Given the current constraints which the writers of this report have, it proves unfeasible to attempt other models such as stacked regression, which could potentially make the residuals even less heteroskedastic (discussed in model limitations). However, it is still possible make relatively accurate interpretations due to lower MAE and RMSE values (discussed under).

A Kernel Density Estimate Plot was created to visualize the density of residuals.

In [385…
```
dataFramewithResiduals = pd.concat(\
[yprednegbinglm, y_test-yprednegbinglm], axis = 1)
dataFramewithResiduals = dataFramewithResiduals.\
rename(columns = {0: "Predicted", 1 : "Difference"})

sns.kdeplot(data = dataFramewithResiduals, \
x = "Difference", y = "Predicted", shade = True, shade_lowest = False);
```

```
/var/folders/kp/bysc8b9n3tb6dh69jj57jvhr0000gn/T/ipykernel_9360/1549912163.py:6: Us
erWarning:

`shade_lowest` has been replaced by `thresh`; setting `thresh=0.05.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data = dataFramewithResiduals, \
/var/folders/kp/bysc8b9n3tb6dh69jj57jvhr0000gn/T/ipykernel_9360/1549912163.py:6: Fu
tureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data = dataFramewithResiduals, \
```



Given that a high density of residuals appear to be concentrated around 0, it was concluded that this particular weighted model was best suited towards modeling the data.

Evaluation

In evaluating the model, it was decided to use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Mean Absolute Error measures the absolute average difference between the predicted values and the actual target values. Root Mean Squared Error measures the average difference between values predicted by a model and the actual model. A lower value for both indicate that the model is more accurate.

In [388…
```
mae(y_test, yprednegbinglm)
```

Out[388…
```
5.391847304569862
```

In [389…
```
rmse(y_test, yprednegbinglm)
```

Out[389…
```
11.727958839961497
```

With both metrics below 20, we can conclude that the model is relatively accurate. The RMSE of 11.73 indicates that there are some outliers with substantial values. Yet, the MAE, since it weighs outliers less, has a lower score, indicating that the data has relatively few outliers, but that those

outliers are substantial in magnitude. They may be so substaantial such that the model is not able to fully capture their effect.

Significance

The following code lists the predictors whose coefficients were not equal to 0 and whose p values were less than 0.05.

In [393…
```
dataFrameWithSignificantPredictors = pd.DataFrame\
(columns=['Predictor', 'Coefficient', 'p Value'])

modnegbinweightsGLMfitPval = modnegbinweightsGLMfit.pvalues.reset_index(drop=False)


for x in range(0,len(modnegbinweightsGLMfit.params)):
    if(modnegbinweightsGLMfitPval.iloc[x][0] < 0.05):
            dfTemp = pd.DataFrame({'Predictor': \
            [modnegbinweightsGLMfit.model.exog_names[x]], 'Coefficient' : \
            [modnegbinweightsGLMfit.params.iloc[x]], 'p Value' \
            : [modnegbinweightsGLMfit.pvalues.iloc[x]]})
            dataFrameWithSignificantPredictors =  \
        dataFrameWithSignificantPredictors = \
        pd.concat([dataFrameWithSignificantPredictors, dfTemp])

dataFrameWithSignificantPredictors
```

```
/var/folders/kp/bysc8b9n3tb6dh69jj57jvhr0000gn/T/ipykernel_9360/1806929587.py:15: F
utureWarning: The behavior of DataFrame concatenation with empty or all-NA entries
is deprecated. In a future version, this will no longer exclude empty or all-NA col
umns when determining the result dtypes. To retain the old behavior, exclude the re
levant entries before the concat operation.
  pd.concat([dataFrameWithSignificantPredictors, dfTemp])
```

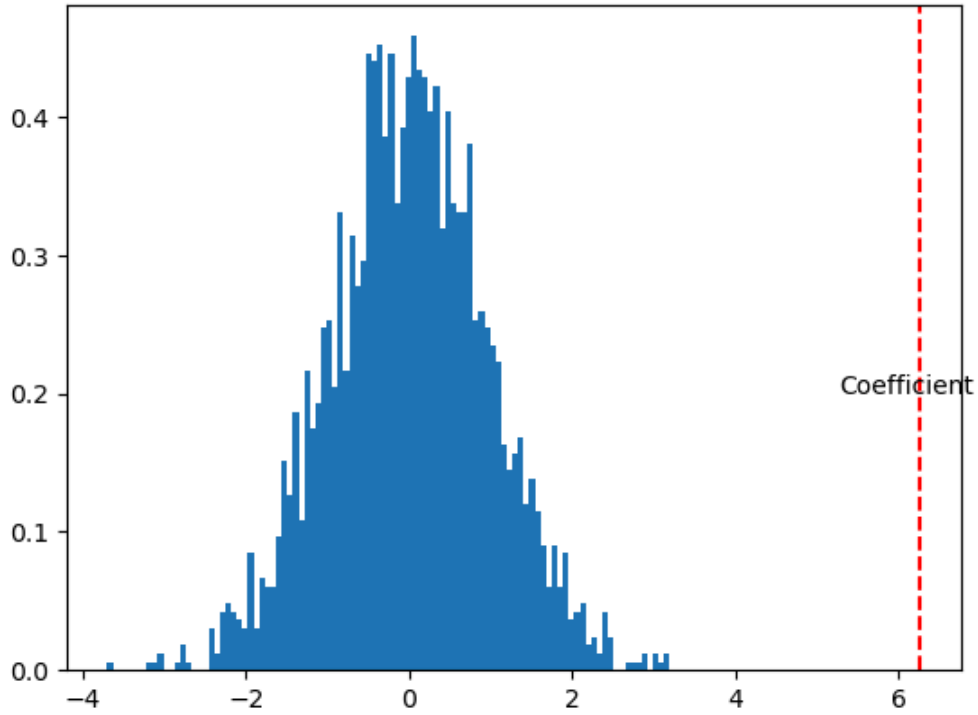
Out[393…

	Predictor	Coefficient	p Value
0	Black	1.703472	3.939257e-10
0	White Hispanic	1.200577	1.105466e-05
0	60616.0	-0.945096	3.305423e-02
0	60620.0	0.690563	4.439447e-02
0	60652.0	-0.768812	3.984051e-02
0	0-19	2.519574	2.985366e-02
0	20-29	2.780806	1.645679e-02
0	30-39	2.405004	3.779152e-02
0	intercept	-3.403634	1.987189e-02

Note that visualizations were also created to understand the deviation of predictors from a student's t distribution. Below is one visualization for the a significant predictor, which was being Black (for demonstrational purposes. The same code can be used for every other significant predictor as well).

In [509…
```
tStat = 1.703472/0.272

degreesOfFreedom = 2404-len(modnegbinweightsGLMfit.params)-1

tdist = np.random.standard_t(df = degreesOfFreedom, size = 2404)
h = plt.hist(tdist, bins=100, density=True)
```

```
plt.axvline(tStat, color = 'r', linestyle = "--")
plt.annotate('Coefficient', (tStat-1,0.2));
```



Note that there were 8 predictors which the model predicted as significant. This means that their p values were less than 0.05. The alternative hypothesis stated that at least one predictor from age group, zipcode, and race must be significant for the null hypothesis to be rejected. Note that this is true. Black is an example of a predictor from race which is considered significant. 60616 is an example of a predictor from zipcode which is considered significant. 20-29 is an example of a predictor from age group which is considered significant. There is thus interplay between year, zipcode, age group, and average test score in one's likelihood of experiencing gun violence.

Note that the model itself is also considered significant. Consult the F-test conducted beneath.

In [399…
```
list = np.identity(len(modnegbinweightsGLMfit.params))
F = np.identity(len(modnegbinweightsGLMfit.params))
F = F[1:,:]
print(modnegbinweightsGLMfit.f_test(F))
```

```
<F test: F=16.641137888739415, p=1.3682182239449465e-66, df_denom=308, df_num=59>
/opt/anaconda3/lib/python3.12/site-packages/statsmodels/base/model.py:1894: ValueWa
rning: covariance of constraints does not have full rank. The number of constraints
is 60, but rank is 59
  warnings.warn('covariance of constraints does not have full '
```

Note that the F statistic is substantially high at nearly 16.64. This indicates a p value which is substantially lower than 0.05. This indicates that the model as a whole is significant and is thus reliable to use when exploring the frequency of gun violence based on the predictors in the model.

Thus, it is safe to say that the null hypothesis can be rejected. There is at least one predictor from race, age group, and zipcode which is significant, and the F-test shows the model as a whole to be significant. This model can hence be used for prediction and the null hypothesis can be rejected.

Interpretation

The predictors which were considered significant will be used for interpretation, along with educational attainment. Note that the model shows a relationship between significant predictor variables such as race, zipcode, and age group, and one's likelihood of experiencing gun violence. First consider the reference variable. A white male of age 80+ residing in the zipcode 60605 in the year 2011 would experience -3.403634 + 0.05563877619007521 * 1.625 = -3.31322098869 gunshots. Note that educational attainment also does not need to be used for this prediction as it is not considered significant; the difference (as seen in calculations) is negligible. Contrast this result with a black male of age 80+ in the zipcode 60605 in the year 2011, who would experience -3.403634 + 1.703472 + 0.05563877619007521 * 1.625 = -1.60974898869 gunshots. This represents a 51.4144% increase. Hence, a black male experiences a greater frequency of gun violence as compared to a white male, even if they are in the same zipcode and of the same age range.

Consider now the interplay between race, zipcode, age group, average test score. In 2011, a black male in the age group 20-29, residing in the zipcode 60652 (which has an average test rank of 28.875) would experience -3.403634 + 0.05563877619007521 * 28.875 + 1.703472 + 2.780806 - 0.768812 = 1.9840166249 gunshots. This represents a 159.882% increase compared to the reference variable.

Note that, in reality, this model is not realistic for negative values. However, it serves to capture the grave discrepancy between different demographics in Chicago in experiencing gun violence. There is interplay between zipcode, test score, age group, and race.

Model Limitations:

The model created above comes with certain limitations, namely:

Value prediction - as mentioned above, this model, while effective for relative prediction purposes, can predict negative values, which in reality is unrealistic. This could be addressed in the future by scaling output values to be 0 or greater.

Model selection - The model chosen is a linear model which assumes that the data resembles a distribution akin to the Negative Binomial Distribution. While this is a better approach than using OLS or a Poisson GLM, this model still does not capture the complete pattern of the data. Future solutions may be to use stack regression or machine learning methods such as random forests.

Use of weights - Even with the use of weights, certain heteroskedasticity exists. Furthermore, it is certainly possible that a better weight function could have been used.

Conclusions

It can hence be concluded that gun violence in Chicago is a multi-faceted problem. The analysis conducted shows novel approaches towards exploring and explaining gun crime in Chicago. It was determined that race appears to be the most significant predictor of one's chances of experiencing gun violence. This finding may encourage local government leaders to identify methods to reduce such discrepancies. Additionally, it is also noted that gun violence appears to trend in Chicago depending on the month of the year, with upticks in July. Local legislators should carefully outline solutions which can be implemented in specific months of the year that can help quell violence. Furthermore, it is demonstrated that the frequency of gun violence appears to increase as test scores and educational attainment decrease. This conclusion may be relevant to school administrators and policymakers, prompting more investment in educational resources as a way to minimize violence.

Overall Data Limitations

Data limitations:

a. The first data limitation that we encountered is in dataSet.csv, the dataset containing Chicago crime information. The dataset as a whole is accurate due to the fact that it is constantly refreshed and modified once the Chicago Police Department has had time to gather relevant information, but the dataset may not always reflect the most updated information as there is a time lag as investigations into the incidents develop. Another limitation regarding this dataset is that there may be missing data regarding certain information like the name of the victimized individual that is not released due to privacy reasons.

b. The second data limitation that we encountered is in the collection of 8 Chicago Public School CSVs documenting academic years ranging from 2011-2013. Due to a lack of publicly accessible data online, we were unable to acquire datasets for years 2014-2015, 2017-2018, 2019-2020, and 2020-2021. The absence of the latter two is presumably due to COVID-19. This missing data presents an obstacle to trying to create time-series graphs, and may cause trends across years to be skewed due to gaps. Additionally, within these datasets, there is occasionally incomplete data or inconsistent methods of recording data. One example of this is with SAT test scores, as some schools simply do not report them. We decided to perform data imputation using averages, filling in missing rows by the average values of all other schools in the same zipcode.

c. For hypothesis 3 in particular: a closer analysis of the data reveals few outliers but of substantial magnitude. This could be contirbuting to the heteroskedasticity seen in the model. This hence may be a limitation of the data itself, where gun violence in Chicago is concentrated specifically in few places or in certain demographics, leading to few outliers of substantial magnitude.

Acknowledgements and Bibliography

"Are Frequency Weights and Sampling Weights, in Practice, the Same Thing?" Cross Validated, 1 Aug. 1963, stats.stackexchange.com/questions/354689/are-frequency-weights-and-sampling-weights-in-practice-the-same-thing. Accessed 08 Dec. 2024.

BRADY. "Statistics." Brady, 2022, www.bradyunited.org/resources/statistics.

"City of Chicago Violence Reduction Dashboard." www.chicago.gov, www.chicago.gov/city/en/sites/vrd/home.html.

Helmke, Paul, and Zirui Song. "Gun Violence: The Impact on Society." NIHCM, 16 Jan. 2024, nihcm.org/publications/gun-violence-the-impact-on-society.

Ostio, Ostio, and tania tania 2. "Statsmodels + Var_weights + Cross_val_score." Stack Overflow, 1 Nov. 1965, stackoverflow.com/questions/63910347/statsmodels-var-weights-cross-val-score. Accessed 21 Nov. 2024.

"6.1 - Introduction to Glms: Stat 504." PennState: Statistics Online Courses, 2024, online.stat.psu.edu/stat504/lesson/6/6.1#:~:text=Summary%20of%20advantages%20of%20GLMs,u Accessed 09 Dec. 2024.

"Statsmodels.Genmod.Generalized_linear_model.GLM¶." Statsmodels.Genmod.Generalized_linear_model.GLM - Statsmodels 0.14.4, 2017,

www.statsmodels.org/stable/generated/statsmodels.genmod.generalized_linear_model.GLM.html. Accessed 08 Dec. 2024.

“Statsmodels.Genmod.Generalized_linear_model.Glmresults.F_test¶.” Statsmodels.Genmod.Generalized_linear_model.GLMResults.F_test - Statsmodels 0.8.0 Documentation, 2017, www.statsmodels.org/0.8.0/generated/statsmodels.genmod.generalized_linear_model.GLMResults.f_ Accessed 08 Dec. 2024.

Swisher RR, Dennison CR. Educational Pathways and Change in Crime Between Adolescence and Early Adulthood. J Res Crime Delinq. 2016 Nov;53(6):840-871. doi: 10.1177/0022427816645380. Epub 2016 May 4. PMID: 28348441; PMCID: PMC5365088., https://pmc.ncbi.nlm.nih.gov/articles/PMC5365088/

Teja, Ravi, and Vladislav Gladkikh. “How to Find the Weights for Weighted Least Squares Regression?” Data Science Stack Exchange, 1 Dec. 1965, datascience.stackexchange.com/questions/82570/how-to-find-the-weights-for-weighted-least-squares-regression. Accessed 21 Nov. 2024.

Venkataramana, Venkataramana. “When Doing a T-Test for the Significance of a Regression Coefficient, Why Is the Number of Degrees of Freedom $N - P - 1$?” Cross Validated, 20 Sept. 2017, stats.stackexchange.com/questions/93130/when-doing-a-t-test-for-the-significance-of-a-regression-coefficient-why-is-the. Accessed 09 Dec. 2024.

“Why Shootings and Violence Increase in the Summer Months.” PBS News, 3 July 2024, www.pbs.org/newshour/nation/why-shootings-and-violence-increase-in-the-summer-months.

Yashmeet Singh, "K-fold Cross-validation using Python and Scikit-Learn", 29 May 2022, https://proclusacademy.com/blog/practical/k-fold-cross-validation-sklearn/

Artificial Intelligence Acknowledgment Note: ChatGPT was used in one case of conceptual understanding, which was to understand how weights would be applied to reduce randomness of residuals. With conceptual understanding, it become easier to understand the other secondary sources (cited above) consulted, and identify a method to weight variables.

Special acknowledgements to Joyce Yan & Nicolas Guerra for their constant support and guidance throughout the production of this report.