Members: Rahat Naseem, Paru Banjara, Elaine Wu, Frederick Kusumo

Table of Contents:

Rahat Naseem: Questions 1 - 5, News article in question 3

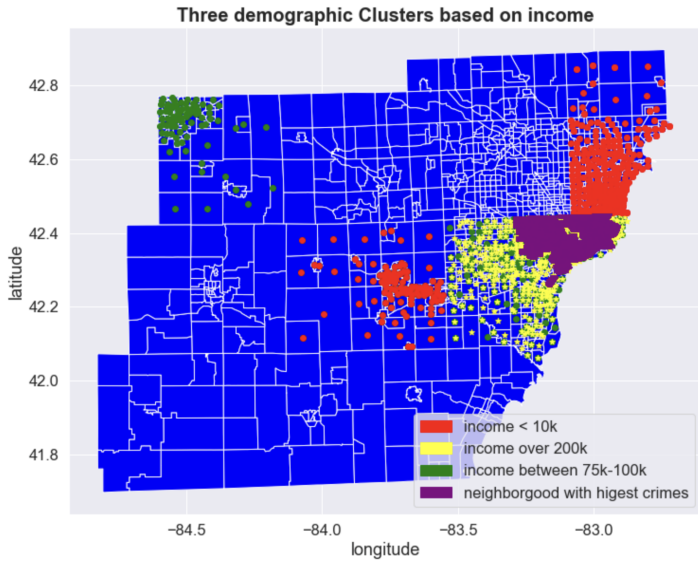
Paru Banjara: Questions 6 - 10, News article in question 7

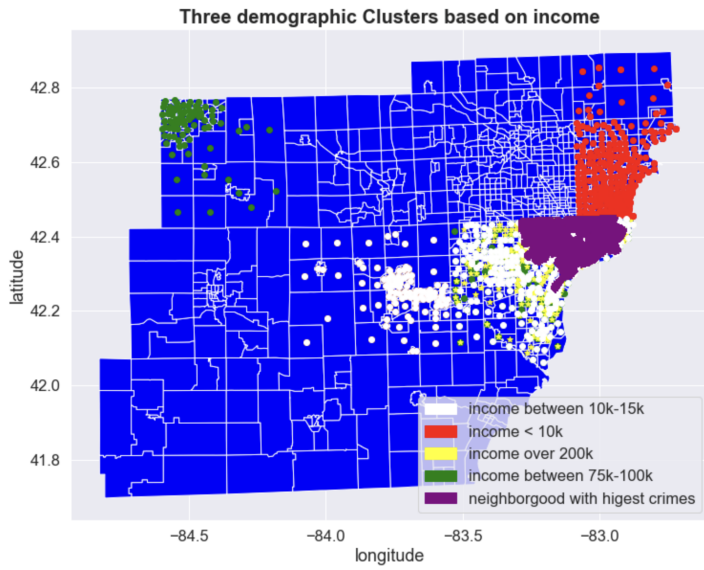
Elaine Wu: Questions 11 - 15, News article in question 15

Frederick Kusumo: Questions 16 - 20, News article in question 18

Question 1: (Rahat)

- **What is the question (~2 sentence)**
  - Are there any clusters of demographics based on various income levels and crime rates?
- **Why does it matter (~3 sentences)**
  - If different income levels and crime rates relate to each other in our datasets, then one variable can reveal more information about the demographics it represents. For example if we find a neighborhood with a higher crime rate, the neighborhood is more likely to have lower income levels .
- **How you approached answering the question at a high level (3-4 sentences)**
  - Find top 5 neighborhoods with most crimes in detroit crime data set and filter the dataset based on these neighborhoods to create small data set
  - Count the average number of crime in each neighborhood
  - Get the mean income level for each county and tract, merge it with the count shape dataframe on the county column and filter the data to South East counties income data.
  - Create small data sets for various income levels
  - Create spatial plots to visualize the relationship between income levels and crime rate.
- **The plot showing your best answer so far**

- 

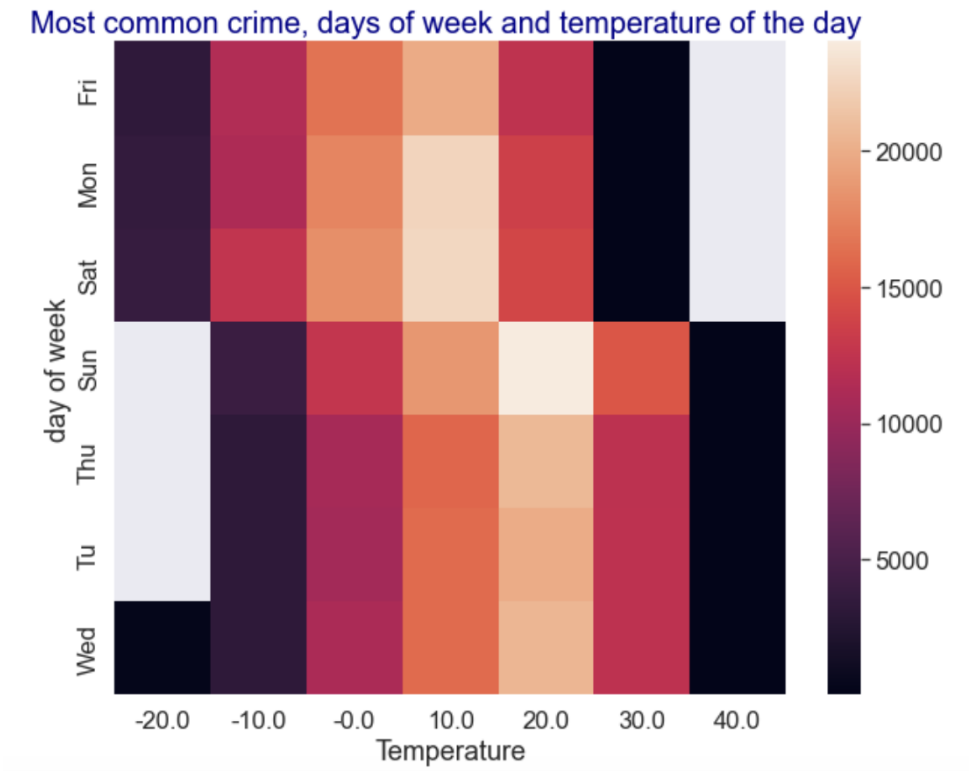
- 

- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - There are demographic clusters based on different income levels, but crime rates are higher in only one region.We can see the region' income levels are between 10k and over 200k. We can see the top red area with only under 10k income level has a total less than 1000 crimes in about 57 years as there are 57 unique years in our dataset.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**

- I would like to find the factors that can be responsible for having crime rate clusters in one area. I looked at only one top crime category . Next time, I can look at different categories of crime and see how they relate to different income levels.

Question 2: (Rahat)

- **What is the question (~2 sentence)**
  - On which day of week most crimes occur and what temperature ranges were when those crimes committed in Detroit?
- **Why does it matter (~3 sentences)**
  - Answering the above question can provide insight about people's behavior in relation to committing crimes and these negative behaviors are driven by extreme temperatures.
- **How you approached answering the question at a high level (3-4 sentences)**
  - I created a common column of day of year in both the Detroit crime data set and the Detroit temperature dataset by creating a datetime object. I merged both datasets on a common column. I counted the number of crimes reported for each crime category and found the most common crime category. I filtered the merged dataset based on the top crime category and created a day of week column and found the average number of crimes for each day of week. I rounded the temperature and used a rolling function to get better visualization as heatmap.
- **The plot showing your best answer so far**



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - On weekends and the first day of week more assaults were reported and the temperature range was 10.0 - 20.0 . I am not sure about the scale of temperature, it looks like it is in Celsius. So, people tend to be more aggressive when the temperatures are moderate. One reason might be that during extreme temperatures, people prefer to stay alone or maybe they are happier in summer and holiday seasons. The temperature suggests that less crimes were committed in summer and winter seasons.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
  - I would like to visualize which category of crimes occur on which hour of the day more often and then find the reasons behind the behavior.

Question 3 (Rahat)

- **What is the question (~2 sentence)**
  - Does higher blood lead levels affect college enrollment?
  - Is there any relationship between college enrollment and having higher levels of blood lead?
- **Why does it matter (~3 sentences)**
  - It is medically evident that exposure to lead during early childhood impact mental development leading to poor performance in school later on. So, my goal was to see if higher blood lead levels contribute to lower college enrollment as people with impaired mental capabilities could struggle to get higher education.
- **How you approached answering the question at a high level (3-4 sentences)**
  - I merged the college enrollment and blood lead levels data sets on common columns and constructed the regression model. I also conducted correlation between the college enrollment and blood lead levels to see how they correlate to each other as well as a regression plot to visualize the correlation.
- **The plot showing your best answer so far**
  - **Correlation between under 18 blood lead levels and college enrolment**

	PctUnder18EBLL	PctEnrl
PctUnder18EBLL	1.000000	-0.018633
PctEnrl	-0.018633	1.000000

  -
  - **Regression Model Summary**

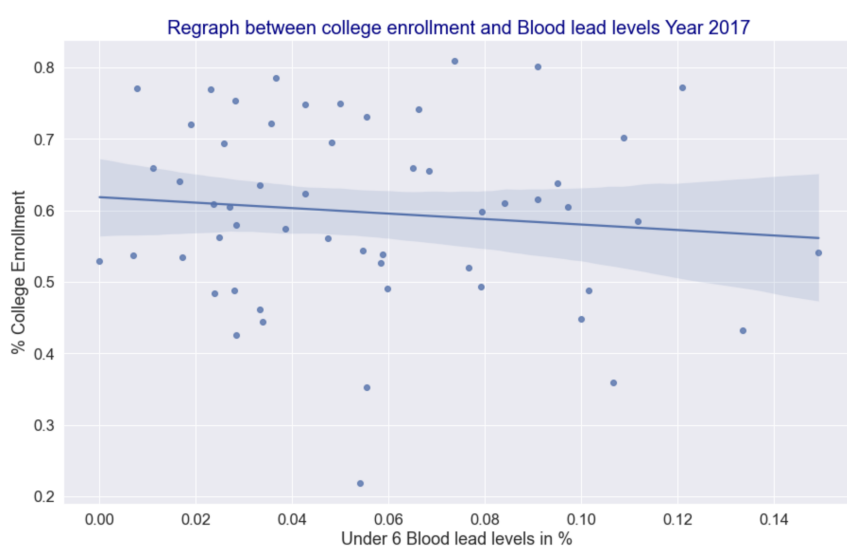
OLS Regression Results

Dep. Variable:	PctEnrl	**R-squared:**	0.021
Model:	OLS	**Adj. R-squared:**	-0.016
Method:	Least Squares	**F-statistic:**	0.5705
Date:	Wed, 19 Apr 2023	**Prob (F-statistic):**	0.569
Time:	10:13:38	**Log-Likelihood:**	36.398
No. Observations:	55	**AIC:**	-66.80
Df Residuals:	52	**BIC:**	-60.77
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.6165	0.033	18.724	0.000	0.550	0.683
PctEBLL	-0.8430	0.796	-1.060	0.294	-2.440	0.754
PctUnder18EBLL	0.5845	0.788	0.741	0.462	-0.998	2.167

Omnibus:	1.466	**Durbin-Watson:**	1.963
Prob(Omnibus):	0.480	**Jarque-Bera (JB):**	1.404
Skew:	-0.368	**Prob(JB):**	0.496
Kurtosis:	2.736	**Cond. No.**	61.2

-



-

- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**

- I did not get a clear answer for my question because my dataset was very small, it had only 54 entries. However,a slight negative correlation means when the blood lead levels were higher in the, the college enrollment was low. However, my results are not statistically significant and are not reliable because R-squared is zero and p value is higher than 0.05 based on the dataset I worked with. " Research suggests significant adverse health effects occur at blood lead levels below the current CDC blood lead action level".(Miranda et al. 2007). Thus, it is evident that lead exposure in early years of life negatively affects mental development leading to poor performance in school that ultimately leads to having less chances of getting enrolled into college.

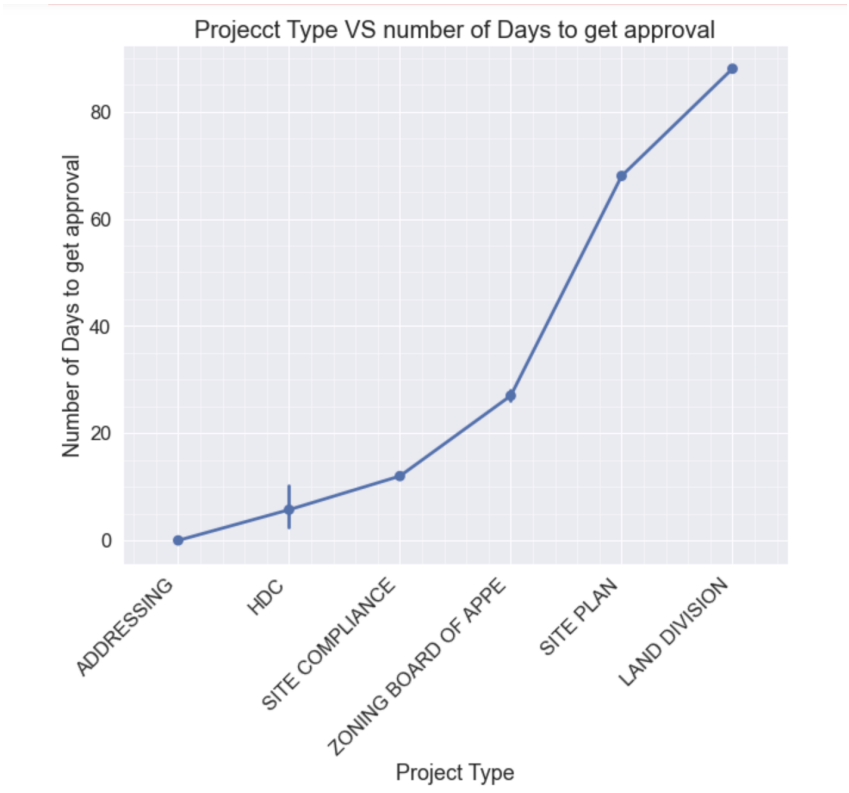
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
  - I would like to have very large data sets and construct the regression model to check my hypothesis. I want to have blood lead level data sets prior two years to that of college enrollment dataset years to get more reliable results.

Question 4 (Rahat)

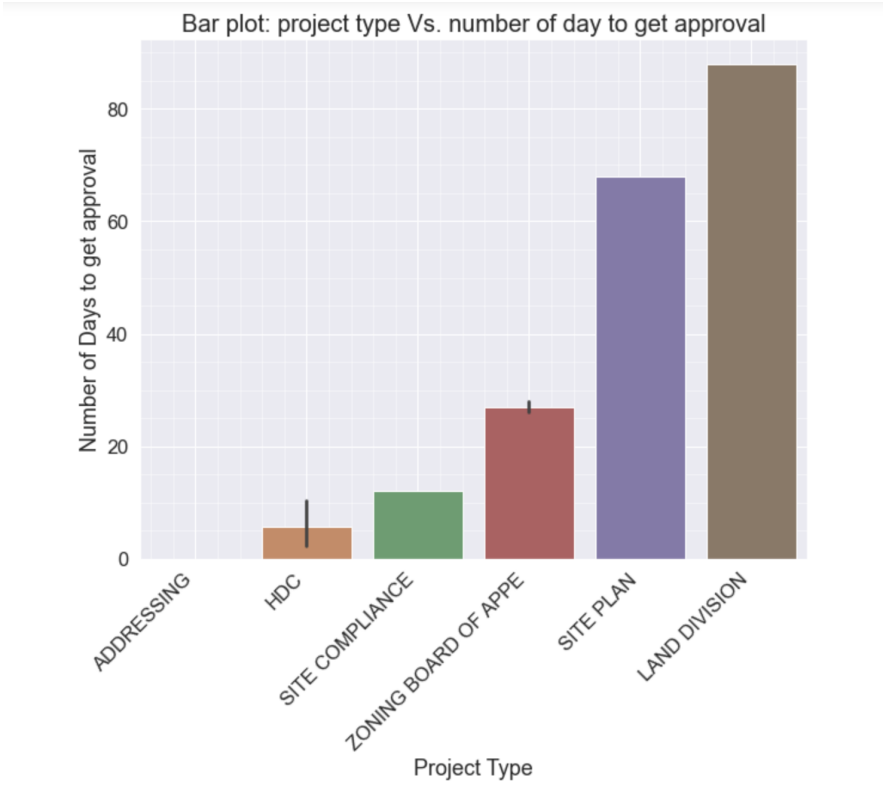
- **What is the question (~2 sentence)**
  - On average how long it takes for a project to get approved ?
  - What type of project gets approved quickly?
- **Why does it matter (~3 sentences)**
  - It will be useful to know what type of projects get approved quickly and if someone wants to do a business related to project development, they can see which areas they should focus on so their project can get approved quickly and they can complete the project on time.
- **How you approached answering the question at a high level (3-4 sentences)**
  - After getting the unique status values, I filtered the data frame based on status for approved projects and then found the number of days the project took to get approved by taking the difference of applied date and

approved date. Next I calculated the average number of days for each owner and project type and created a point plot and bar plot to visualize the results.

- **The plot showing your best answer so far**



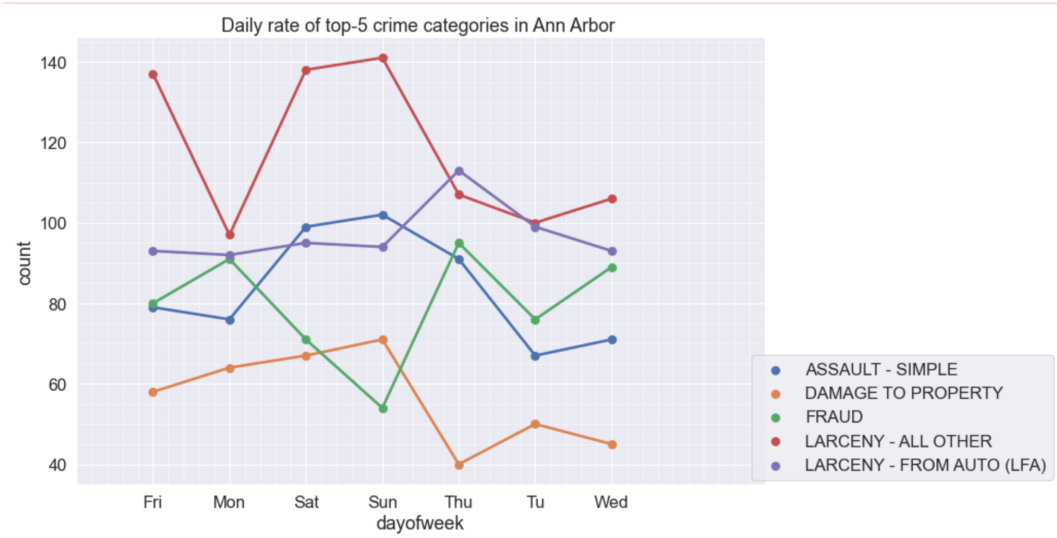
-

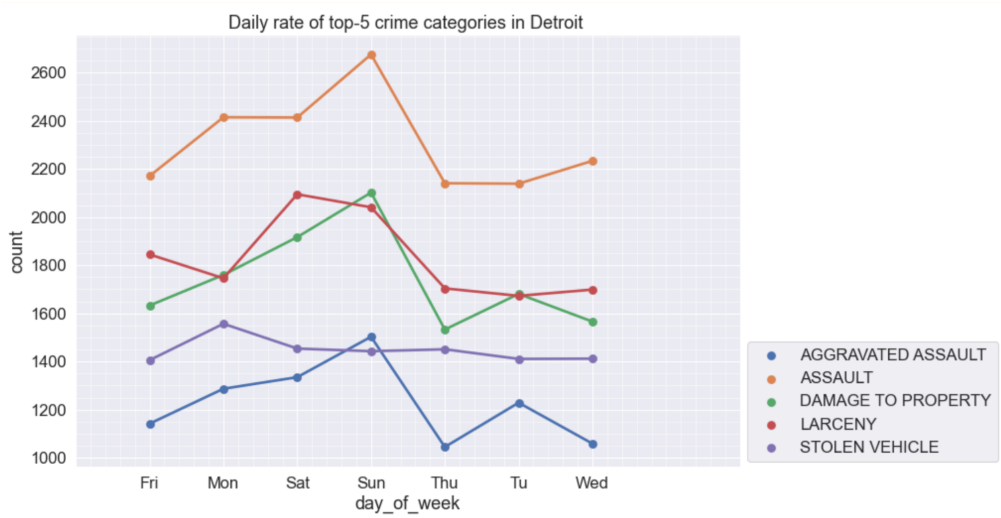


- ---
- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - Projects related to addressing are approved almost on the same day while HDC projects take only 5 - 7 days to get approved. However, land division projects get approved in more than 80 days. The graph makes sense because allocating addresses does not involve too many steps. It is possible that a computer system can generate addresses as soon as an application is submitted. Since land division required physical verification of the site by visiting and might involve more departments of the agency, that is why it took so long to get approval.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
  - I would like to have large datasets and some explanation from the department for delay so I can do some text analysis to see if negative sentiments in comments and feedback on application can predict how long it will take for a project to get approved.

Question 5 (Rahat)

- **What is the question (~2 sentence)**
  - What are trends in top 5 crimes in Ann Arbor for the year 2022, and 2023 on each day of week as compared to Detroit crime data sets for the same years?
- **Why does it matter (~3 sentences)**
  - It would be interesting to visualize how the crime rate differs across top 5 categories for 2022 and 2023 for the Detroit crime data set and Ann Arbor dataset. If it differs we can try to find the factors that are responsible for the difference.
- **How you approached answering the question at a high level (3-4 sentences)**
  - After creating a new column for day of week in Ann Arbor and Detroit Crimes datasets, filtered the data based on year 2022, and 2023, and figured out top 5 crime categories for each dataset. Filtered the dataset based on these top 5 categories for each dataset and calculated the number of crimes for each day of week and each category. Display the results by plotting pointplot graph for each data sets and compared the graphs.
- **The plot showing your best answer so far**



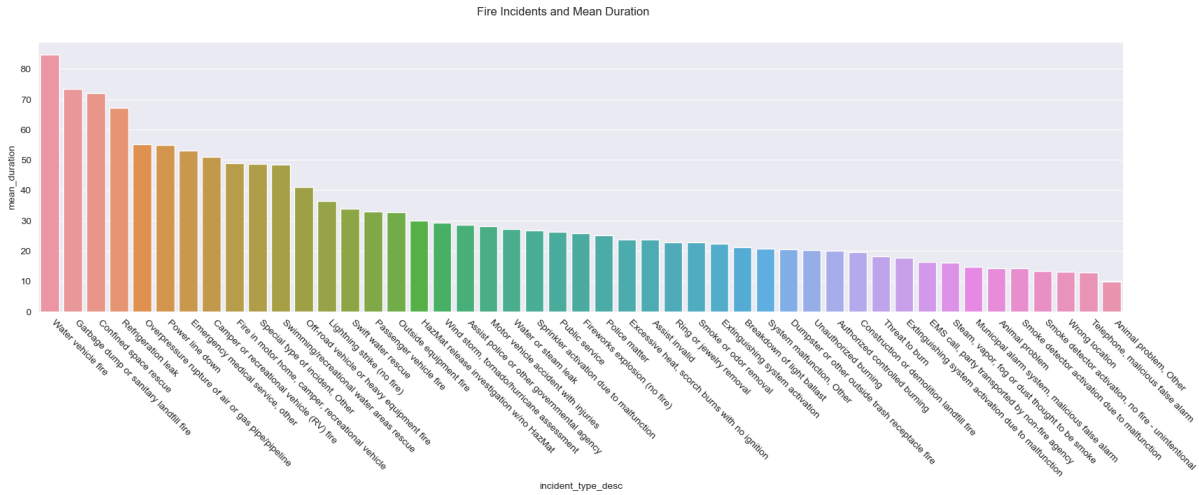


- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - In both datasets, most crimes occur over the weekends.In Detroit two crime categories of assaults were committed more often over the weekends, while in Ann Arbor usually frauds are committed on weekdays and larceny is most common crime over the weekends followed by larceny. Larceny is the second most common crime over weekends on both datasets. In Ann Arbor assault is the third most common crime while it is the most common crime in Detroit.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**

- I can analyze the same data sets for each month and see which crimes are most common in each month. I probably filter the datasets for some certain limit of total counts and then create spatial visualizations for each set.

Question 6 (Paru)

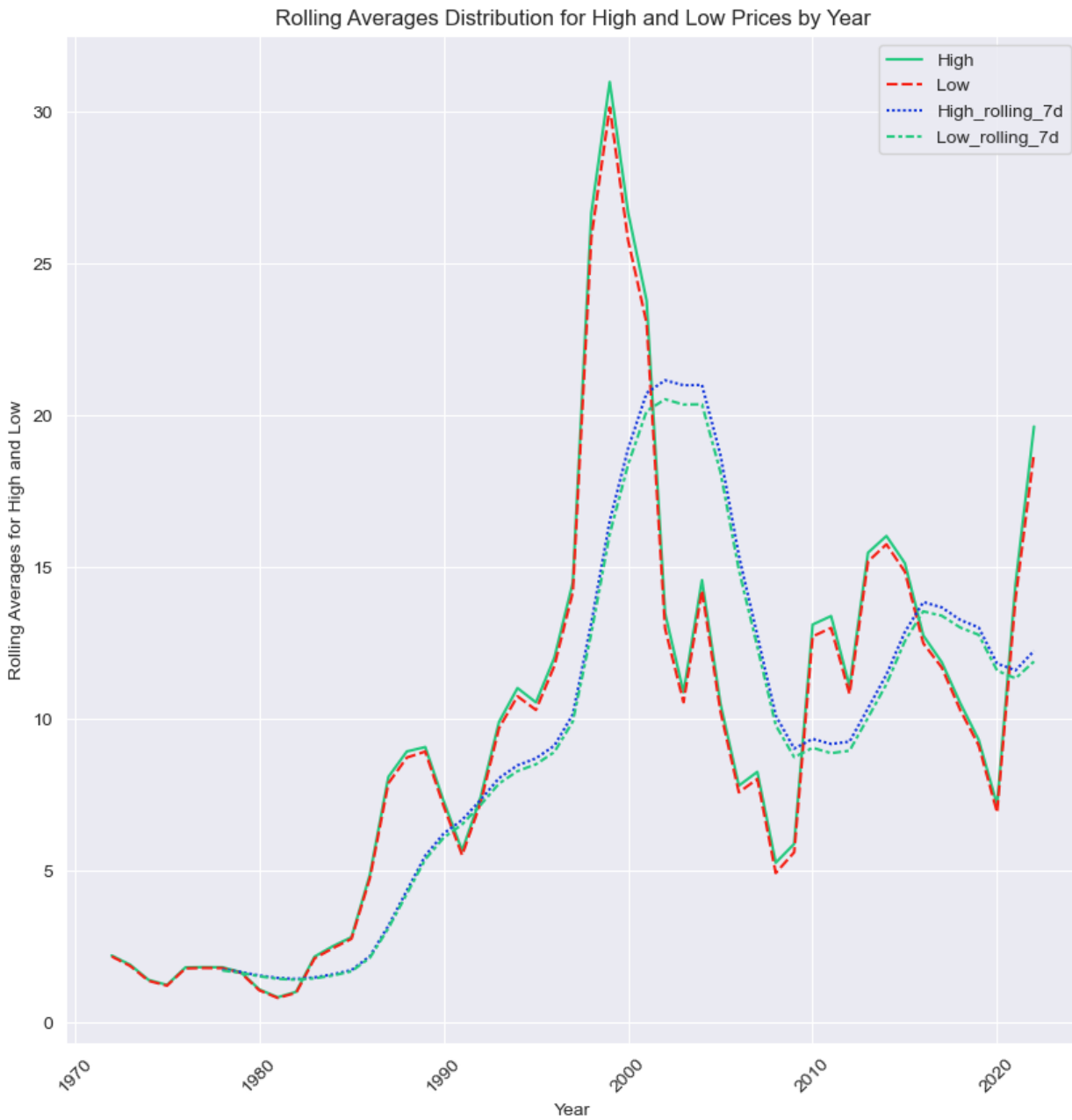
- **What is the question (~2 sentence)**
  - Identify different types of fire incidents. How long did it take for the fire to be disposed of from the time the fire incident was reported?
- **Why does it matter (~3 sentences)**
  - It is important to recognize how long the fire lasted to estimate the damage, injuries and fire related deaths. It allows for the concerned parties to allocate the resource accordingly. It is also important to note the type of fire because fire can spread quickly. If the fire occurred in a confined area, it can quickly cause extreme damage, but if it is a motor vehicle fire or open space fire, the damage may be less.
- **How you approached answering the question at a high level (3-4 sentences)**
  - In order to answer this question, I calculated the time difference between call datetime and cleared datetime. Then, I converted it to hours and used a 'group by' on the incident type description and the duration and sorted the values. The dataset had a large amount of data, so I only selected 30 percent sample data to plot the bar diagram using seaborn.
- **The plot showing your best answer so far**



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - Based on the plot, the duration of fire from the time of call datetime to the fire cleared datetime, vary by the type of fire. For example, the highest fire duration (more than 80 minutes) based on the plot is “water vehicle fire” and the lowest duration (10 minutes) is “animal problem, other”. It seems from the plot that a lot of the incidents descriptions with lower duration are actually not the fire issues. For instance, some of the incidents described are just the incident descriptions, not the actual fire, which are caused by false alarms, smoke detector malfunctions, fogs or dust, animal problems and so on. However, the incident descriptions with longer duration are in fact caused by actual fire, which include power line down, leakage of gas pipeline, vehicle fires and so on. This suggests that if the fire incident is of more serious nature, the duration is also longer.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
  - I would merge this dataset to the weather or any other datasets to find out different trends and patterns of fires and correlation of fires with other variables. My hypothesis is that fires mostly occur during the dry season, but it would be interesting to test my hypothesis to see if my hypothesis would be rejected or approved by the actual analysis.

Question 7 (Paru)

- **What is the question (~2 sentence)**
    - What are the rolling average trends of the high and low prices for Ford Stocks over the years from 1970 to 2020?
- **Why does it matter (~3 sentences)**
    - It is important for the stocks buyers and sellers to identify the trends and patterns of when the stock prices go high and when it can fall to make better purchase and sell decisions. It also allows the investors to perform further analysis of the causes for the rise and fall of the stock market.
- **How you approached answering the question at a high level (3-4 sentences)**
    - I approached this question by first converting the date to datetime and extracted the years. Then, used ‘group by’ year, high and low and got the mean. After that, I calculated the rolling 7 day average for both the high and low prices. Lastly, I made the line plot to plot the graph for high and low prices as well as for the high rolling 7 day and low rolling 7 day.
- **The plot showing your best answer so far**



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - Based on the plot, the average high and low prices remain about the same when we look at the plot just by high and low (red and green or blue and green plots) without comparing with the time. However, if we look at the average stock prices over a longer duration, such as by year, the stock

prices were low from 1970 to 1985 and the prices gradually started to increase after 1985. The prices started to rise after 1985 and peaked during 1998 to 2005. The rolling average curves are smoother, and still show the fluctuations in stock prices. The average high and low curves are pointy, but they show a lot of fluctuations in greater detail. “Rolling averages are useful for finding long-term trends otherwise disguised by occasional fluctuations” (Indeed Editorial Team, 2022). “Rolling averages can also help you determine the factors that cause your trends. This can help you prepare and make decisions for the future of your business” (Indeed Editorial Team, 2022). The graph also shows that the prices had dropped in 2008 and 2020. The rolling average graphs do not give the exact time frame like the other curve, but it is just helpful in observing the trends for when the rise and decline occurs, so that the investors can make informed decisions by looking at the trends.

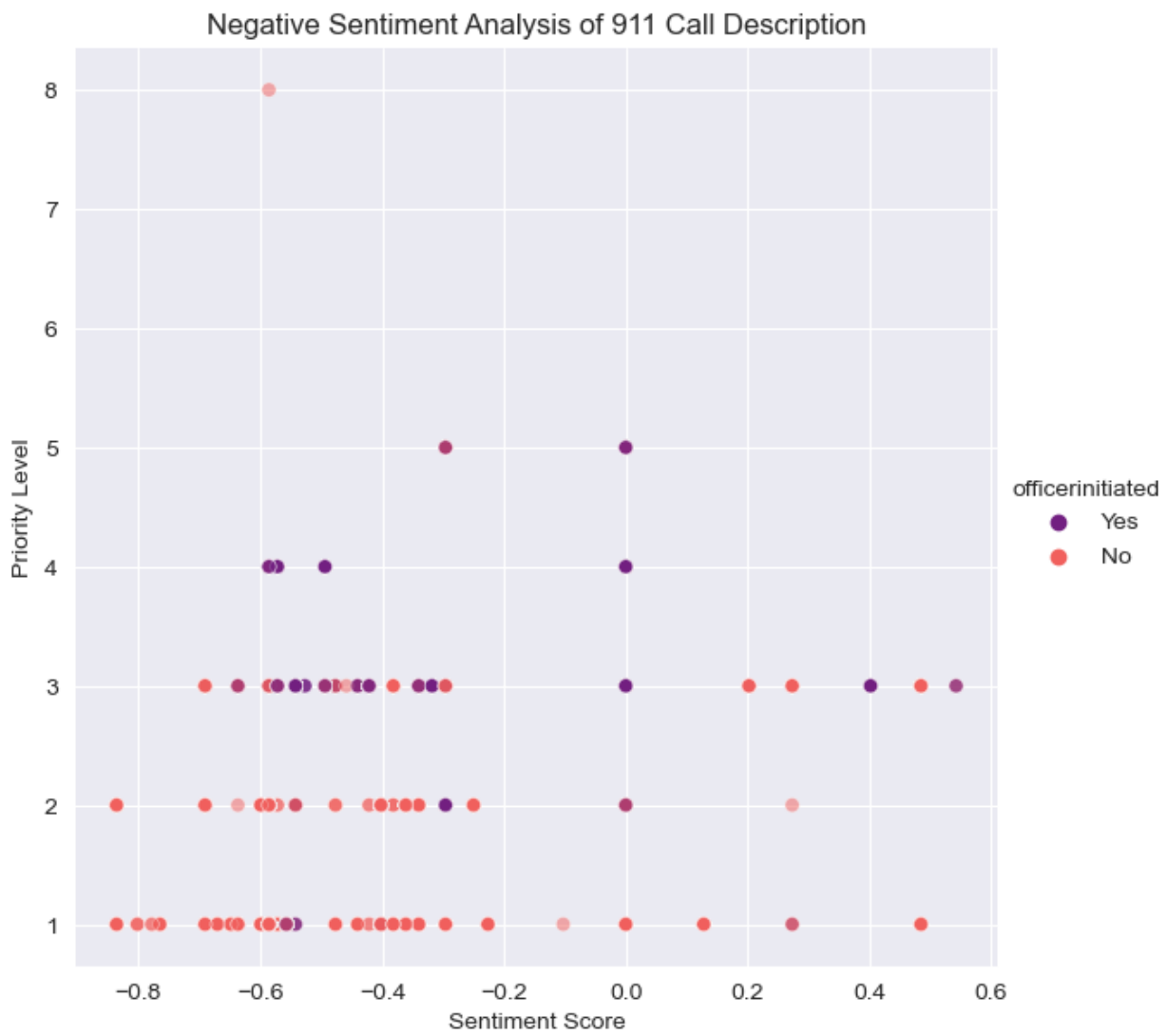
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - In order to dig deeper on this, I would try to find the trends of the stock prices of other companies to make comparisons about how different companies’ stock prices fluctuate over the years and try to identify the causes or factors leading to the fluctuations.

Question 8 (Paru)

- **What is the question (~2 sentence)**
    - Is there a positive relationship between negative sentiment of call description and type of the offense? Does more serious crime get a negative sentiment score?
- **Why does it matter (~3 sentences)**
    - It is important for the machine learning models to identify negative and positive sentiments, so that they could be used to detect crimes. It is valuable because machine learning models can alert humans quickly if

they are taught to analyze the positive and negative sentiments from the texts.

- **How you approached answering the question at a high level (3-4 sentences)**
  - I approached this question by first getting the sentiment score of call description using the sentiment intensity analyzer, polarity scores and compound. Then, I converted the priority into numeric and only selected the non na values. I also calculated the correlation between priority and sentiment score of call description. After all, I used seaborn relplot to plot the graph.
- **The plot showing your best answer so far**



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - My hypothesis for this question was that the more serious crimes would be given higher priority, so the sentiment score would also be negative. The plot also seems to indicate a similar result, but it isn't a very strong correlation. If we look at the plot, priority 1 has a lot of negative sentiment scores. As the priority increases the sentiment scores seem to shift towards the right. However, the result isn't very consistent because as we can see, priory 1 also has positive sentiment scores. There is one outlier at the top and some dots are aligning on the 0 sentiment score. One of the

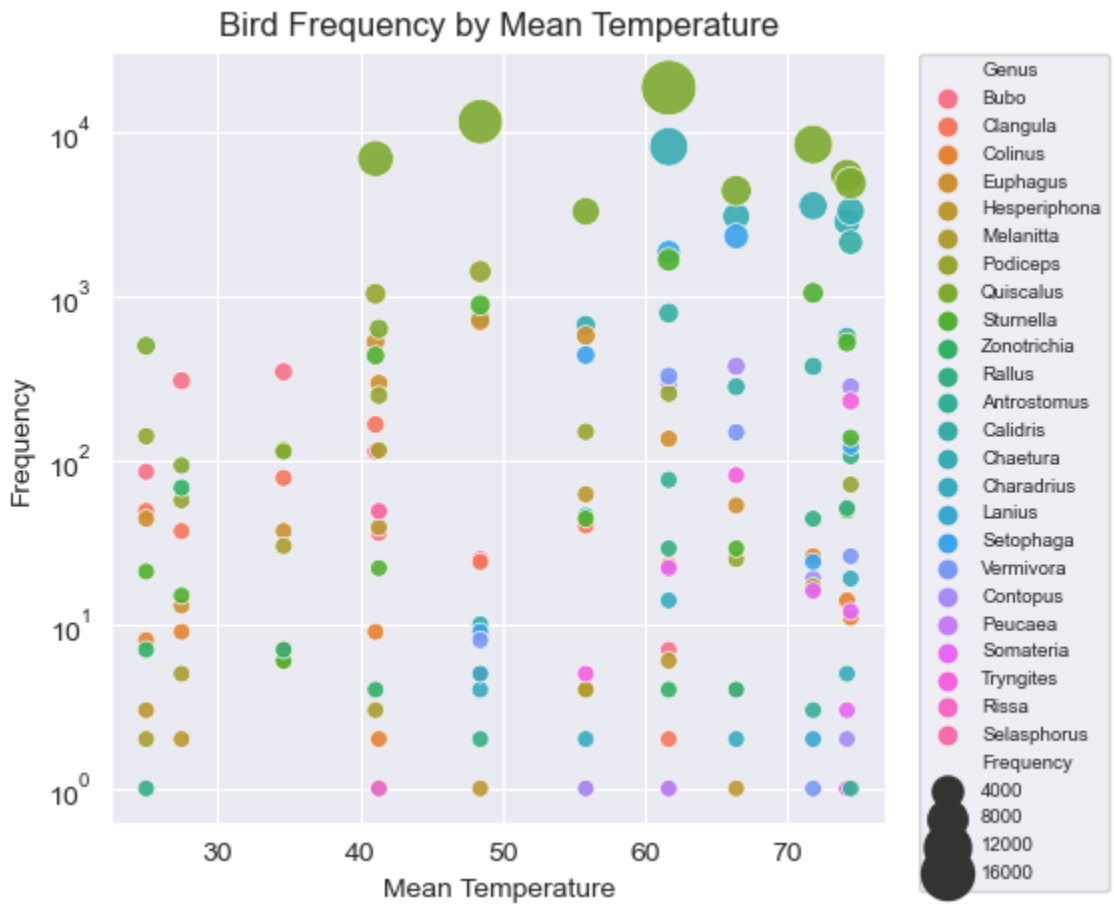
reasons for this kind of result is because the dataset did not contain very descriptive texts. The text descriptions were really short and were of repetitive nature. Because of that many results ended up being 0 and positive.

- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - If I had more time, I would definitely look into other datasets which had the larger descriptions and perform the sentiment analysis on those texts. I would also try to use a different plot.

Question 9 (Paru)

- **What is the question (~2 sentence)**
    - How does bird frequency change in relation to the change in temperature in south-east Michigan? Are some birds more migratory than others?
- **Why does it matter (~3 sentences)**
    - Birds are an important part of our environment and ecology because they help in seeds dispersal and other important things, which are vital to maintain the ecosystem. It is important to identify the birds' migratory patterns in order to investigate if the cause of their migration is related to change in temperature or other things. This question helps to identify which birds have greater tendency to migraterate by the change in temperature.
- **How you approached answering the question at a high level (3-4 sentences)**
    - In order to answer this question, I merged the two datasets - the michigan weather and michigan birds - by month after converting the string date object to datetime. I used group by to group the month, mean temperature and genus columns and aggregated them by genus count and used seaborn scatter plot to plot the diagram.

- **The plot showing your best answer so far**



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - Based on my plot, the birds in south-east Michigan are very migratory. For example, if we look at the plot, there are a lot of different kinds of birds when the temperature is warmer. Also, their frequencies are higher during the warmer climate, such as the green and blue dots at the top of the plot. A lot of birds that we see during the warmer temperatures are missing during the colder temperatures.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - I grouped the bird counts based on their genus, but if I wanted to dig deeper, I would also look into their species and make different plots for

their species to identify if some birds of the same genus have different migratory patterns than other species.

Question 10 (Paru)

- **What is the question (~2 sentence)**
    - How does the volume of the Ford stock change by the change in other variables? Are there any variables that are statistically significant at $p<0.05$? What is the status of standard errors?
- **Why does it matter (~3 sentences)**
    - It matters because it helps to investigate the factors that cause the stock volume to fluctuate. By identifying the variables that are correlated with the stock volume fluctuations, investors can predict if the stock value would go up or down and make the best investment decisions. For example, if there is a strong correlation of stock volume rising when the unemployment rate is low, then investestors can anticipate the stock values by looking at the unemployment rate.
- **How you approached answering the question at a high level (3-4 sentences)**
    - First, I converted the dates to datetime and merged the unemployment dataset to the ford stock dataset on the date. Then, I made a linear regression model using different independent variables with volume being a dependent variable. After that, I extracted the second column to only get the coefficients, standard-errors, variables, and p-value. Then, I again had to convert the data types of those variables to create an errorbar. Lastly, I created an errorbar using matplotlib.
- **The plot showing your best answer so far**

OLS Regression Results

Dep. Variable:	Volume	R-squared:	0.222
Model:	OLS	Adj. R-squared:	0.209
Method:	Least Squares	F-statistic:	16.58
Date:	Thu, 20 Apr 2023	Prob (F-statistic):	7.86e-17
Time:	15:54:07	Log-Likelihood:	-6634.5
No. Observations:	355	AIC:	1.328e+04
Df Residuals:	348	BIC:	1.331e+04
Df Model:	6		
Covariance Type:	nonrobust		

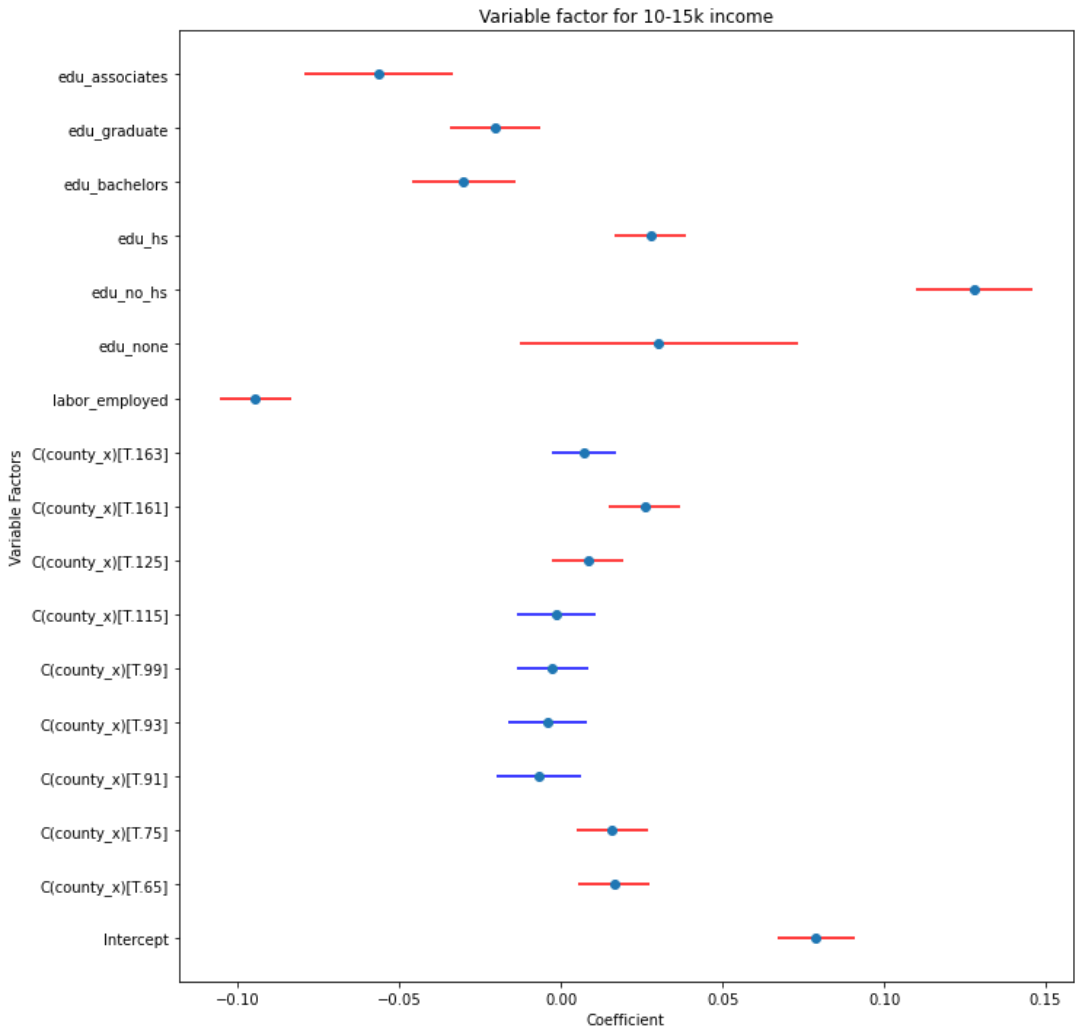
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.335e+07	8.77e+06	-2.663	0.008	-4.06e+07	-6.1e+06
High	6.948e+07	1.66e+07	4.174	0.000	3.67e+07	1.02e+08
Low	-4.875e+07	1.4e+07	-3.486	0.001	-7.63e+07	-2.12e+07
Open	1.629e+07	1.47e+07	1.110	0.268	-1.26e+07	4.51e+07
Close	-3.776e+07	1.5e+07	-2.517	0.012	-6.73e+07	-8.25e+06
Unemployment_Rate_MI	-2.084e+08	1.58e+08	-1.320	0.188	-5.19e+08	1.02e+08
Unemployment_Rate_US	9.121e+08	2.49e+08	3.669	0.000	4.23e+08	1.4e+09

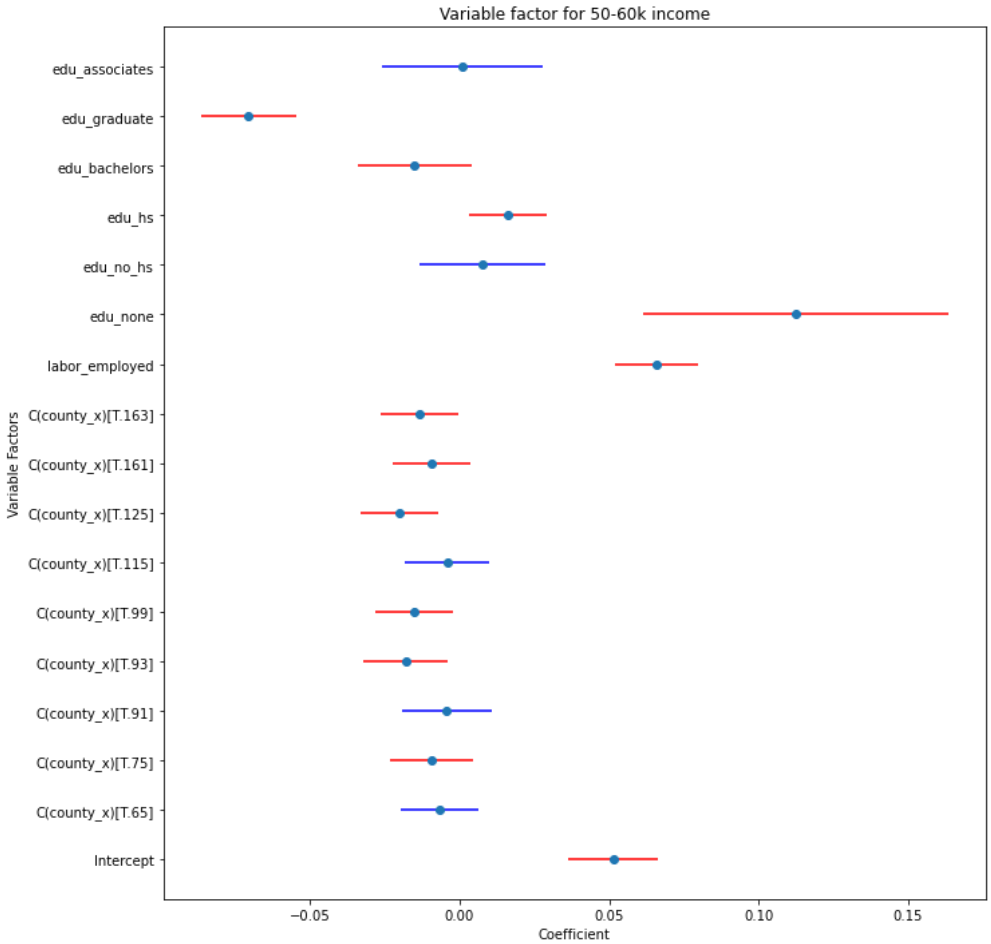
Omnibus:	148.258	Durbin-Watson:	0.701
Prob(Omnibus):	0.000	Jarque-Bera (JB):	929.482
Skew:	1.630	Prob(JB):	1.46e-202
Kurtosis:	10.226	Cond. No.	4.19e+03

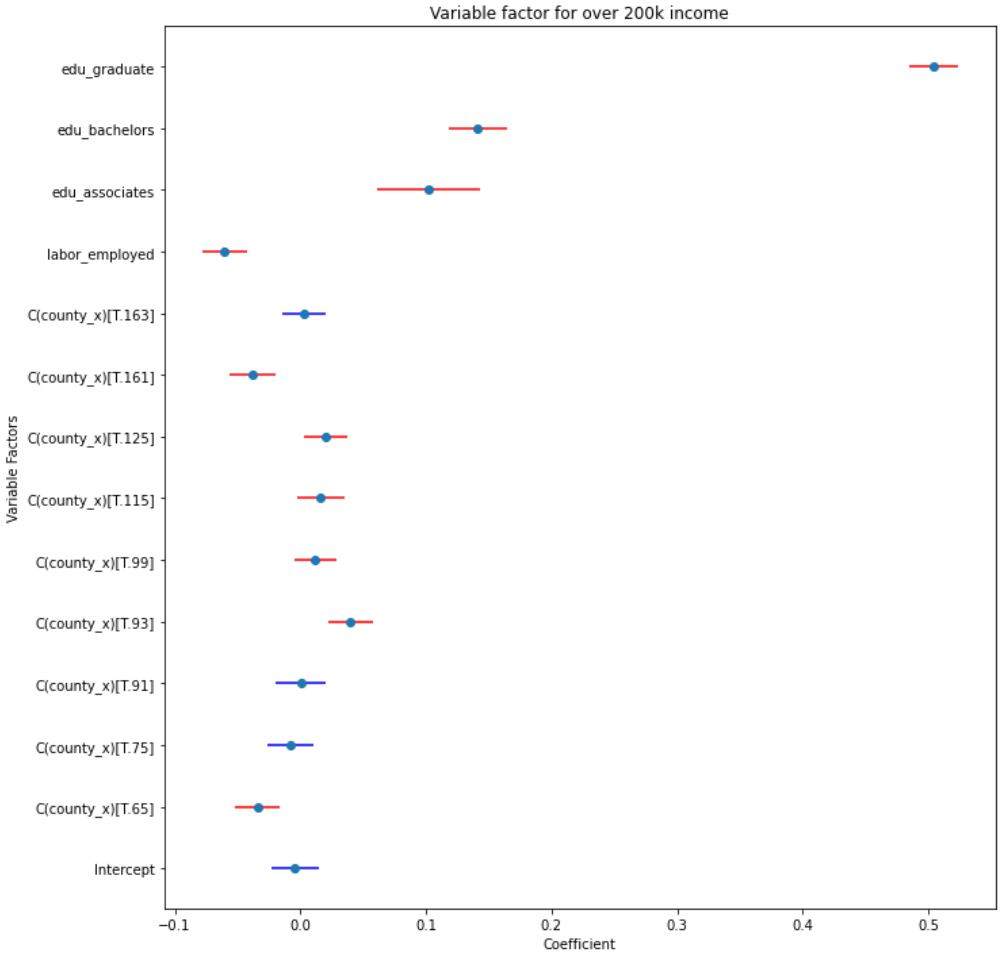
- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - Based on the plot, I see that the independent variables, such as high, low, open, and close have very small error bars, but the unemployment rates have large error bars. This indicates that the unemployment rate isn't a good predictor of change in stock volumes even though the p-value is statistically significant at $p<0.05$ for the variable 'employment rate US'. Also, the unemployment rate variables are further away from the axvline, while the other variables are around the axvline. The R-squared and intercepts seemed to change by the change in independent variables. This suggests that the independent variables are correlated with the stock volume, but the correlation isn't very strong.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
  - If I had more time, I would find other datasets to merge with these datasets. I would also use more independent variables to see how it would change the p-values, R-square and standard errors.

Question 11 (Elaine)

- **What is the question (~2 sentence)**
  - How does various education levels and labor employment affect different income levels in Michigan counties?
- **Why does it matter (~3 sentences)**
  - Some of the counties here in Michigan have a high average of income level where some counties are not. It is important to find out which counties satisfy or overall are closest to the hypothesized conditions. We can then find out from here what are the possible causes of why some counties have such high or low economic levels.
- **How you approached answering the question at a high level (3-4 sentences)**
  - In order to answer this question, I merged the acs_income and acs_work datasets based on the tract since both information are based on the same state and county. Then, I took out three income levels: 10-15k, 50-60k, and over 200k. From here I put each income level as a dependent variable then hypothesized what factors that might have closest linear relationship to. Afterward, I turned all three linear regressions into a dataframe containing the variables, coefficients, standard errors, and probabilities. Lastly, I did an errorbar plot based on coloring the difference if the p value is greater or less than 0.5.
- **The plot showing your best answer so far**


Variable factor for 10-15k income
edu_associates
edu_graduate
edu_bachelors
edu_hs
edu_no_hs
edu_none
labor_employed
C(county_x)[T.163]
C(county_x)[T.161]
C(county_x)[T.125]
C(county_x)[T.115]
C(county_x)[T.99]
C(county_x)[T.93]
C(county_x)[T.91]
C(county_x)[T.75]
C(county_x)[T.65]
Intercept
Variable Factors
−0.10
−0.05
0.00
0.05
0.10
0.15
Coefficient


Variable factor for 50-60k income
edu_associates
edu_graduate
edu_bachelors
edu_hs
edu_no_hs
edu_none
labor_employed
C(county_x)[T.163]
C(county_x)[T.161]
C(county_x)[T.125]
C(county_x)[T.115]
C(county_x)[T.99]
C(county_x)[T.93]
C(county_x)[T.91]
C(county_x)[T.75]
C(county_x)[T.65]
Intercept
Variable Factors
−0.05
0.00
0.05
0.10
0.15
Coefficient

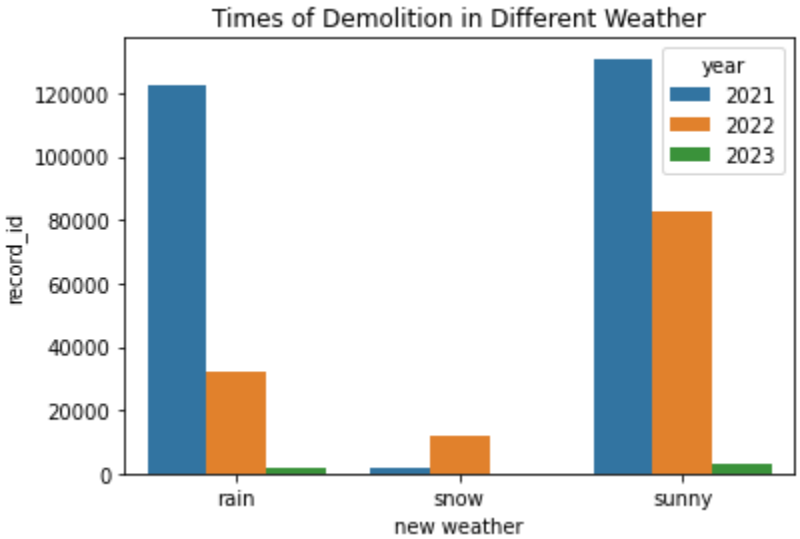

Variable factor for over 200k income
edu_graduate
edu_bachelors
edu_associates
labor_employed
C(county_x)[T.163]
C(county_x)[T.161]
C(county_x)[T.125]
C(county_x)[T.115]
C(county_x)[T.99]
C(county_x)[T.93]
C(county_x)[T.91]
C(county_x)[T.75]
C(county_x)[T.65]
Intercept
Variable Factors
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
Coefficient

- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - I did three different dependent variables based on the 10k, 50k, and over 200k. From each of the models, I imported different factors that I hypothesized would be rational to the dependent variable. For the 10-15k model, there are strong correlations on none_education, high school education or middle school education. The coefficient for that is positive. But, graduate education is a negative correlation and it's proven by when probability is less than 0.05, meaning there is no correlation at all. For the 50-60k model, I was almost right but there seems to be one factor off regarding to none_educaiton, thus I don't think there is a correct relation from income to all the variables. For the 200K model, my hypothesis assumption was correct where all factors about level of education have significant relationship to 200k income as well a positive correlation.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - I would like to find out the names of counties tract based on the model to identify which county has significant correlation that fits right into the model hypothesis. From here, I figure out the economic level of counties and why that happened.

Question 12 (Elaine)

- **What is the question (~2 sentence)**
    - What is the weather condition that demolition happened the most?
- **Why does it matter (~3 sentences)**
    - Demolitions happen all the time, but Michigan weather is weird that impedes the progress of wrecking it down. It is interesting to find out what type of weather conditions that allow demolition to happen the most. Thus it would give the government and state a sense on how to better coordinate to have demolition on which day and when not to.

- **How you approached answering the question at a high level (3-4 sentences)**
    - I approached the question by first applying the datetime object to both datasets' date columns. I also apply strftime on the demolition date to modify closely matched with the weather data date column. Then I inner merged the SE Michigan weather and Detroit demolition dataset together based on date. Next, I wrote and aggregated the function that goes through the SE Michigan Weather on snow and precipitation columns that would identify out snow, rain, and sunny. Then I applied a function to Michgan weather by creating a new column. Lastly, I grouped the year and weather type together on count of record id from demolition. Lastly, I barplot the result by showing the weather type in different years.
- **The plot showing your best answer so far**



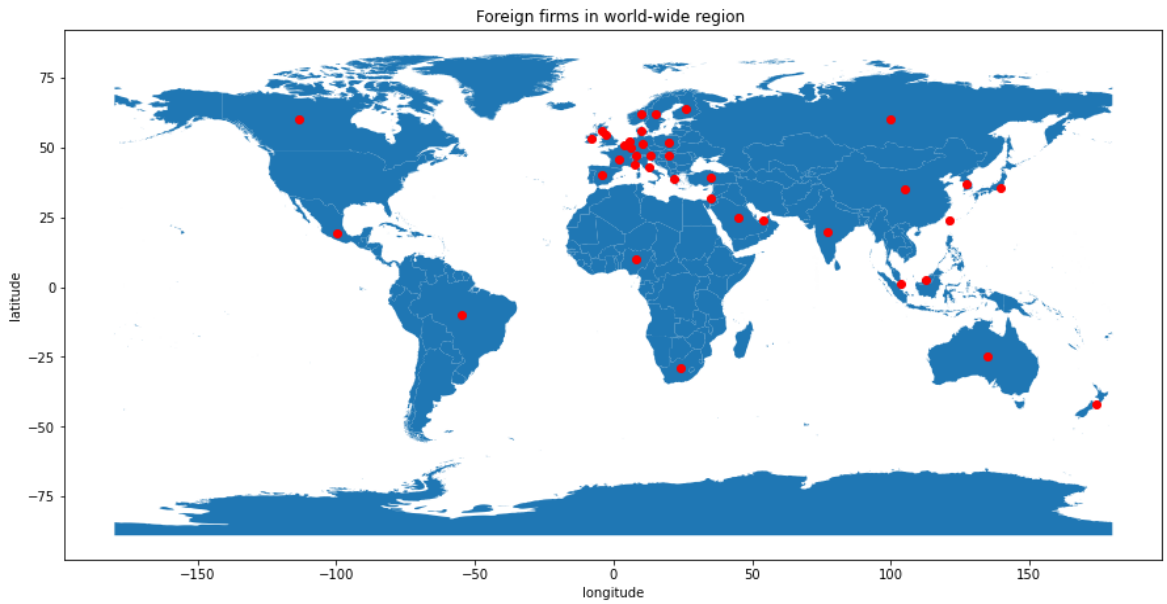
- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - Based on the plot, I can conclude that during the sunny weather condition is when demolition happens the most. From the graph, the colors are based on years and under the sunny weather, that is the highest count of demolition happened over the three years. Rainy season was followed by

that and the snow season was little to none demolition happened due to critical conditions or low temperatures. Which makes sense based on that.

- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - I would want to dig deeper on expanding more different types of weather from the weather data that might give a full detailed view of when demolition happened most under which weather conditions. It would be interesting to see if my thinking is right or different from the graph above.

Question 13 (Elaine)

- **What is the question (~2 sentence)**
    - Which part of the word region has the most international foreign firms residing here in SE Michigan?
- **Why does it matter (~3 sentences)**
    - It adds an economic value and importance for SE Michigan future planning on which area in the world that could make the best possible partner to improve the economy. Knowing the current foreign firms residing here in SE Michigan lets local companies know which world region they can have international partnerships with for easier access cooperation.
- **How you approached answering the question at a high level (3-4 sentences)**
    - I started off with a small function to aggregate through foreign own firms, which states the countries of the foreign firm. I sort through the values of the column and then group the X and Y coordinates together because I only need one unique country. After that, I apply my aggregate function to the grouping. Then I found a generalized world countries map. I also turned the foreign firm into a new geodataframe based on x and y coordinates. Lastly, I used sjoin skill that joined both datasets together. I used matplotlib to plot the datasets together.
- **The plot showing your best answer so far**



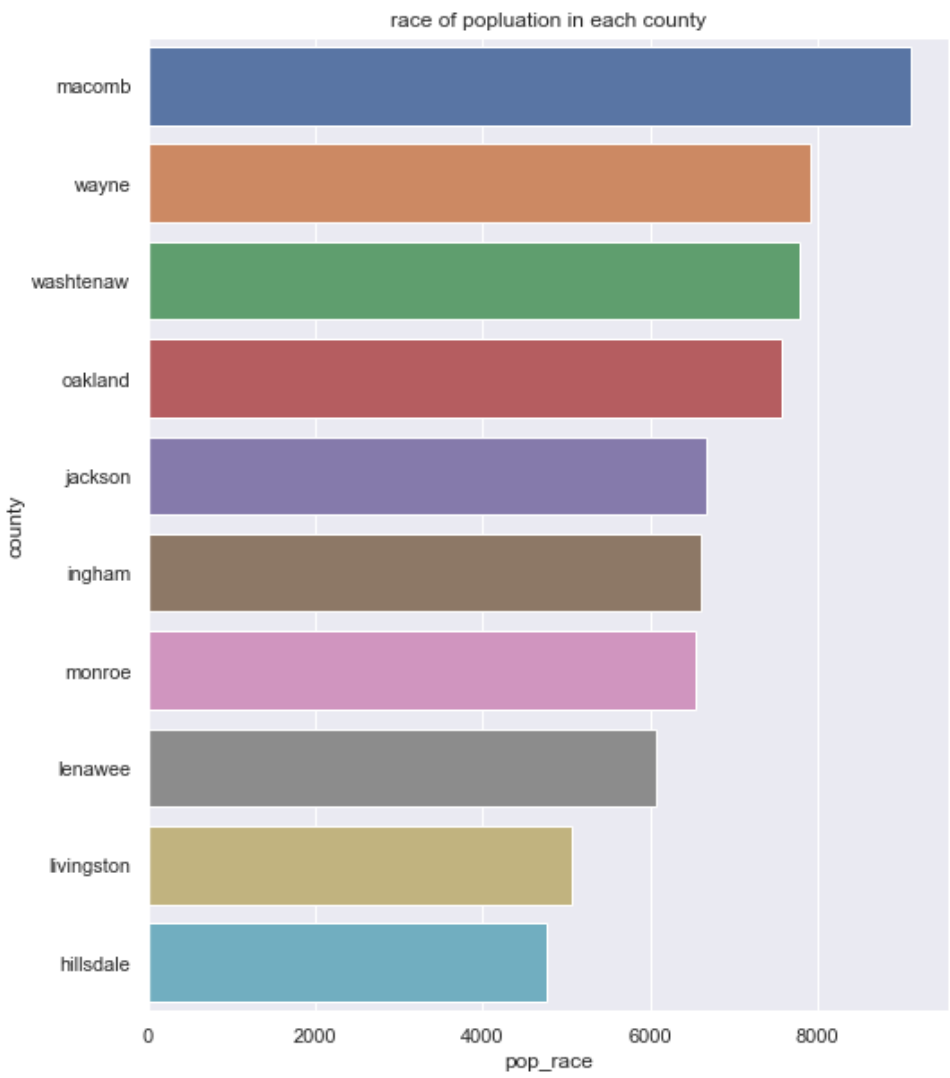
- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - From the sjoin plot, the cluster happened the most in European countries or Europe region where most foreign firms are generally from residing here in SE Michigan. Thus, it would give extra pieces of information to local firms where to look for international partnership. Then the rest are kind of scattered around the world where the firms originated from. But, Europe should be the focus to look for partnerships or have business interactions. According to the Oakland Michigan business development website, there is an overview of the county area where it locates the placement of foreignment and there is also an overview of a chart that counts each foreign firm. From the chart, about half of the countries doing international business development are from Europe area; for example, like Netherlands, Italy, Switzerland etc.(Oakland County). There also another article from Michigan Economic Development Corporation posted out the top ten countries that do investments here in Michigan. About half of the countries originate from the European region (Michigan business).

It's evident that the European region has the highest potential of doing business or investment with Michigan state.

- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
  - Next, I would also plot to see where each foreign firm is located here in SE Michigan. I would also dig through to see what are the top regions where most foreign firms are from. I see there are a lot of firms from Europe and from Asia, I want to know which countries have the highest count.

Question 14 (Elaine)

- **What is the question (~2 sentence)**
  - Which county has the highest population race in SE Michigan?
- **Why does it matter (~3 sentences)**
  - It would be beneficial to understand the census tract better on which county has the total overall race population. From here, It would give a lead on where to start the processing for incoming census counting. It also gives the general idea how many people needed to distribute in each county for tracking the count of census. In addition, the higher population means more people living in the county, and more diversity on racial populations. It would be great for political members to get the support from such a large crowd.
- **How you approached answering the question at a high level (3-4 sentences)**
  - I used the acs_demographic dataset where I split the tract_name that contain the county names and create new columns based on that. Then I group-by the county and apply lambda that filters through each county and sorts its values based on the max number of races. Then I seaborn barplot the diagram.
- **The plot showing your best answer so far**

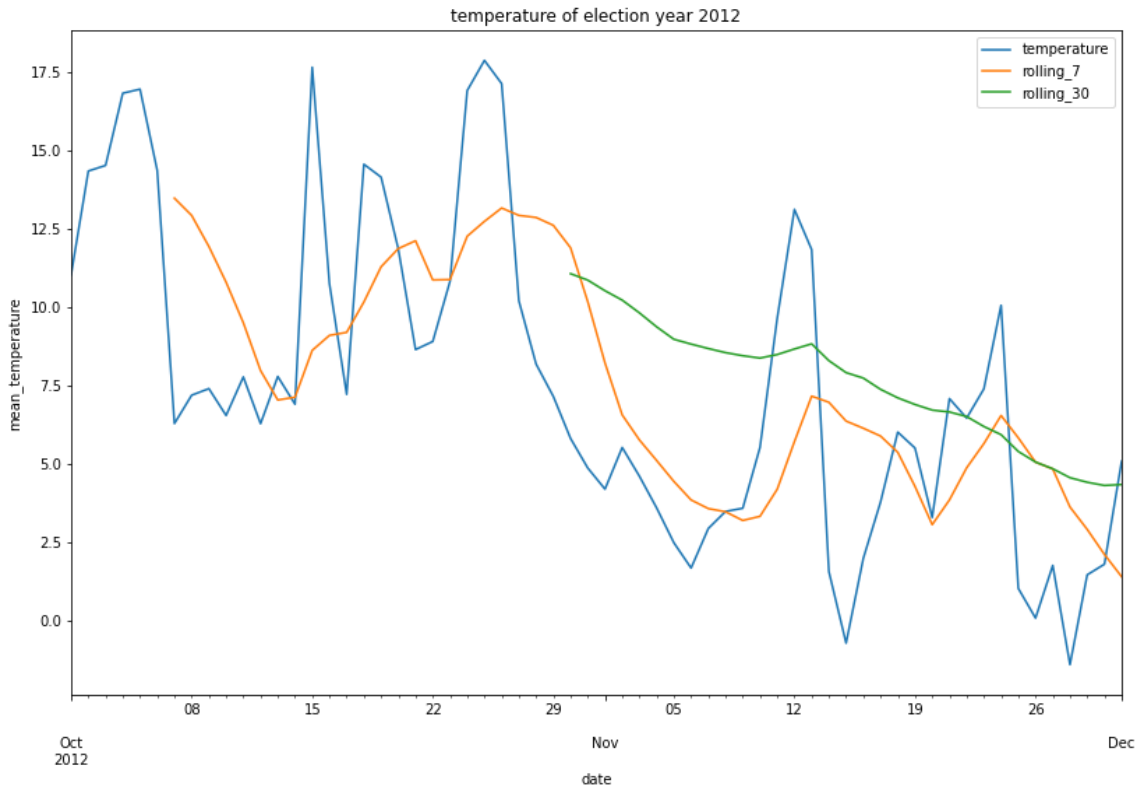


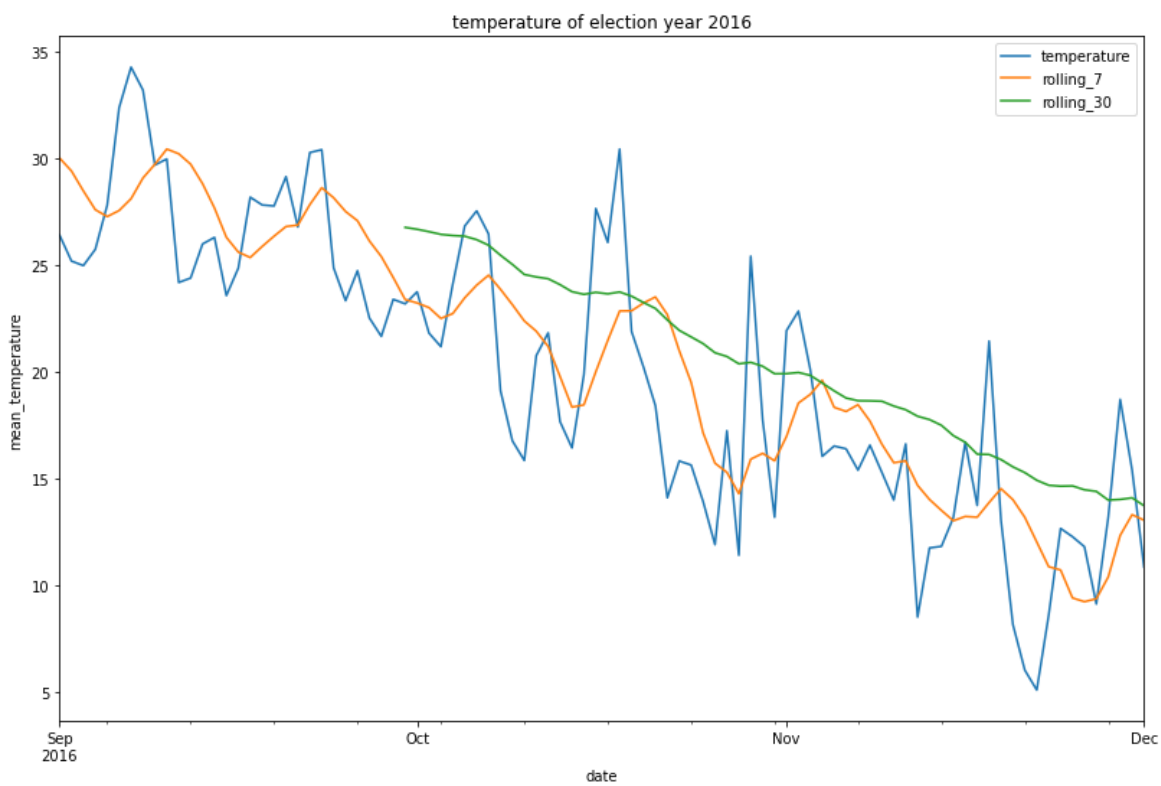
- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - From the graph above, it's not Wayne county that has the most population race. It is called Macomb county and has the highest population race in SE Michigan counties. This concludes that this country might contain diverse racial groups or contain a large crowd of certain races here. Then, second place overall was Wayne county and third was Washtenaw county. That was an interesting fact to see now knowing which county has the highest racial population. I thought it was Wayne county since, it have the highest population.

- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - I would dig deeper to showcase what the percentage of races in each county is. From here, I can find out which is the biggest race that resides in each county and figure why that might happen.

Question15 (Elaine)

- **What is the question (~2 sentence)**
    - Did the temperature rise from the year 2012 to the year 2016 election years?
- **Why does it matter (~3 sentences)**
    - Global warming has been ravaging all over the world in recent years where we are experiencing abnormally high temperatures or storms. There are times where November would be really warm and sunny or cold and freezing. It would be interesting to compare the 2012 and 2016 to see if there is a significant difference among these years on temperature which could be also leading to people coming out to vote.
- **How you approached answering the question at a high level (3-4 sentences)**
    - I took the approach to join the presidential election data set and detroit weather dataset together by year. I applied a datetime function to Detroit weather and merged. Thus, it filtered down to a two year data frame that fit the timing. Then, I create a small subset of dataframe from the filter dataframe group-by and get October of 2012 - December of 2012 and October of 2016 - December of 2016. Next, I used a rolling function on 7 days and 30 days to see the average overall temperature. At last, I subplotted both data frames.
- **The plot showing your best answer so far**


temperature of election year 2012
temperature
rolling_7
rolling_30
mean_temperature
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
08
15
22
29
05
12
19
26
Oct
2012
Nov
Dec
date


temperature of election year 2016
temperature
rolling_7
rolling_30
mean_temperature
35
30
25
20
15
10
5
Sep
2016
Oct
Nov
Dec
date

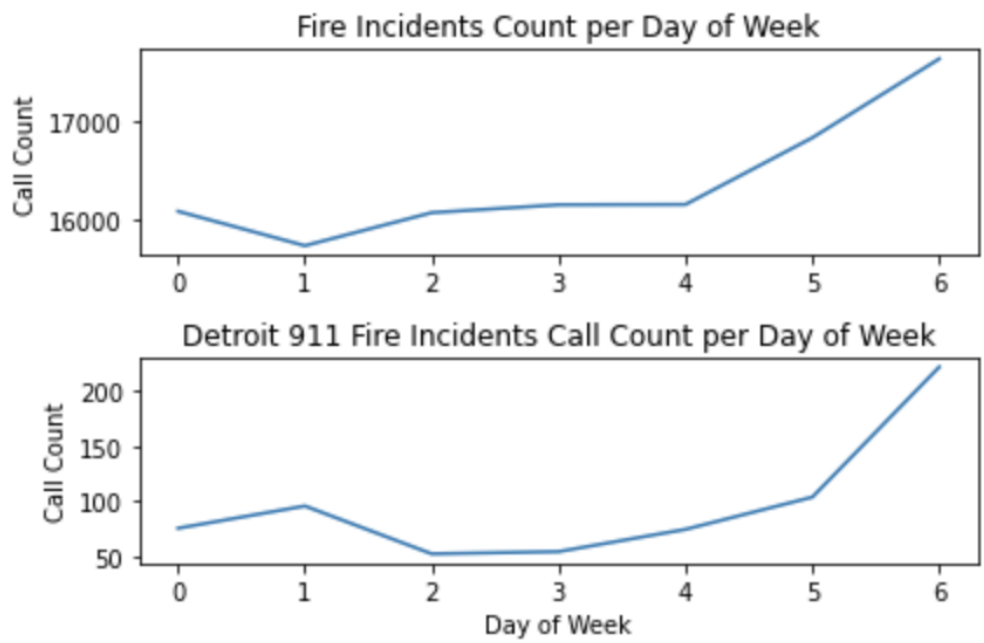
- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - Comparing both graphs from 2012 and 2016 from October to December, I can see the difference of messing temperature in 2012 compared to 2016. Though generally overall, I can see 2016 have higher average temperature mean than 2012. Looking at November month(election month), 2012 graph average mean are around 0 to 12.5 degree in Fahrenheit. In the 2016 graph, the average mean temperature in November was around 10-20 degrees Fahrenheit. It is said that temperatures have changed over the 4 years where it's slowly increasing year by year. It is an indication that global warming is emerging to cause rising temperatures.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - I want to count through each day in the election years to see specifically the period of time temperature and count if there are more total votes happening if the temperature is warmer or lower when temperature is cold.

Question 16 (Frederick)

- **What is the question (~2 sentence)**
    - Which day of the week has the highest calls regarding fire and how does it compare between the two different data sets [Fire & 911]?
- **Why does it matter (~3 sentences)**
    - It matters to find out regarding this trend because firefighters, police, and 911 dispatch officers would pay more attention and prepare on those days. Law enforcements or the local government could also investigate why those days have the more calls related to fire than usual, maybe it is because of an event that normally happens on those days and potentially find solutions on how to mitigate that issue. Moreover, this would also

enable decision makers like from 911 and local governments to research on things that could be done to reduce the number of fires generally.

- **How you approached answering the question at a high level (3-4 sentences)**
  - Prior to conducting any manipulation, I filtered the 911 data to just retrieve the ones that are regarding fire incidents. Both data sets contain a timeseries column and I made a new column in each data set to apply the column to a time series. I then made another column in each respective data set for the day of week based on the time series column I created. Next, I grouped the day of week and count the calldatetime column as it is the occurrence of each call. Lastly, I created a subplot to be able to plot both data in one plot.
- **The plot showing your best answer so far**



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - Based on the plot above, it can be seen that fire incidents are more frequent close to the weekend. It will then decrease and the lowest is around mid week which is Tuesday to Thursday. Though both data has

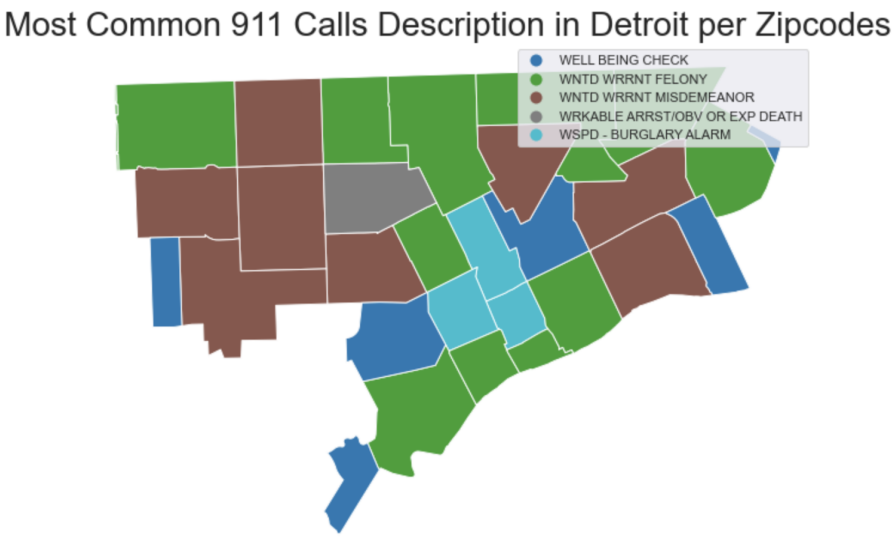
slight differences, but in general they are showing the same trend. As the Fire data set is range from a couple of years and the Detroit 911 data set is just within a 30 days time frame, this would explain the slight difference the graphs are. But, in general the idea would be that the trend can be followed from the fire data set.

- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
  - I could have possibly investigated more regarding the months or the years on which fire incidents are higher or lower. However, I would need to find a 911 data set that has readings for longer than a year.

Question17 (Frederick)

- **What is the question (~2 sentence)**
  - What 911 calls category is the most popular per zipcode?
- **Why does it matter (~3 sentences)**
  - This will be a great way for law enforcement to view on the most popular type of 911 calls made that is more popular in each zipcode. This will be able to let them investigate on what laws to tighten or allocating more crews to be dispatch for surveillance in the area. Furthermore, local government can also explore on what the society needs more.
- **How you approached answering the question at a high level (3-4 sentences)**
  - I found Detroit zipcode boundary online which I downloaded and imported from the shp file. Since some of the data sets do not contain some of the zip codes, I just made a new dataframe containing just the zipcodes from the zipcode boundary dataframe. I then merged this new dataframe with detroit 911 data to just get the lines containing the zipcodes that exist in the boundary dataframe. Then I counted the number of crimes by each category and just get the most occurrence crime for each zipcode. Lastly, I plot the data sets by using the zipcode boundaries as well as the most popular 911 calls description into a spatial map.

- **The plot showing your best answer so far**

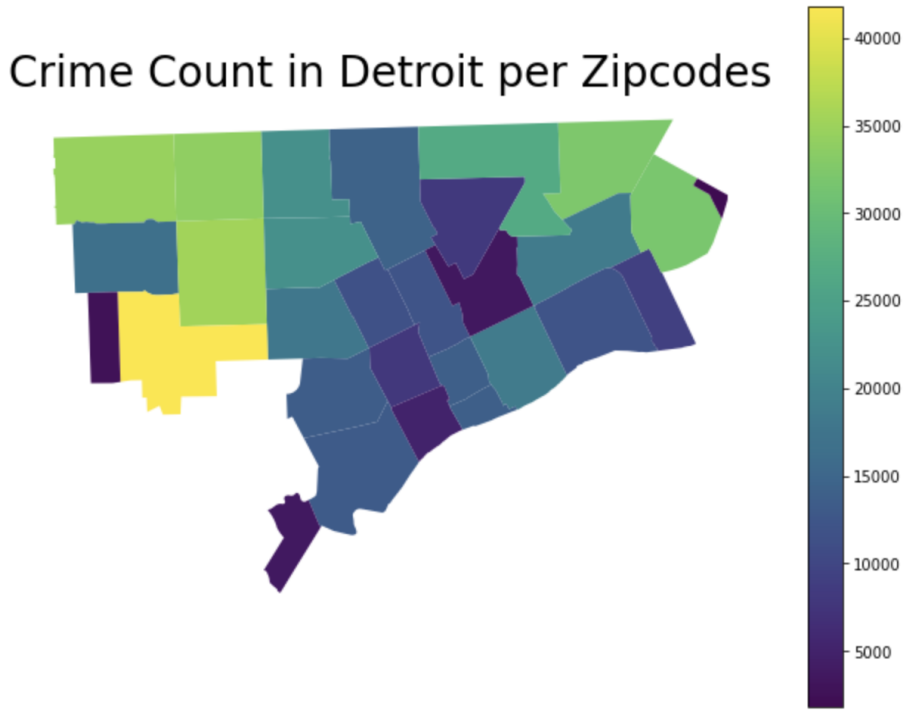


- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
  - Based on the graph above, it can be concluded that the two of the most common 911 calls in Detroit are because of Felony and Misdemeanor. Other popular 911 calls would be regarding Well Being Check, Workable Arrest and Burglary Alarm. There are generally more Felony related calls on the borders of Detroit and more Misdemeanor related calls in the West and East side of Detroit. But Midtown and the nearby zipcodes have Burglary Alarm as their most popular 911 call description.
- What would you differently or next if you wanted to dig deeper (~2 sentences)
  - I think it would be interesting to actually investigate the most popular actual crime. It would enable me to compare if these popular calls end up being crimes or just false alarm.

Question 18 (Frederick)

- **What is the question (~2 sentence)**
  - What is the crime count per Detroit zipcode?

- **Why does it matter (~3 sentences)**
    - This is important especially towards police departments and law enforcers as they can find out which zipcode actually needs the help most. They can also allocate more budget in order to increase labors around those area to keep it safer. Moreover, people visiting can distinguish which area to be more cautious of or which area to avoid.
- **How you approached answering the question at a high level (3-4 sentences)**
    - As mentioned from the previous question, some data sets do not contain a couple zipcode or a complete zipcode of Detroit. This means that I have to filter out the zipcodes that are only contained in the zip code boundary from the detroit rms data. Next, I grouped the zipcode and counted the number of crimes by the offense_description column. And lastly, I plot the data using the zipcode boundary and the count of offense_description per zipcode.
- **The plot showing your best answer so far**



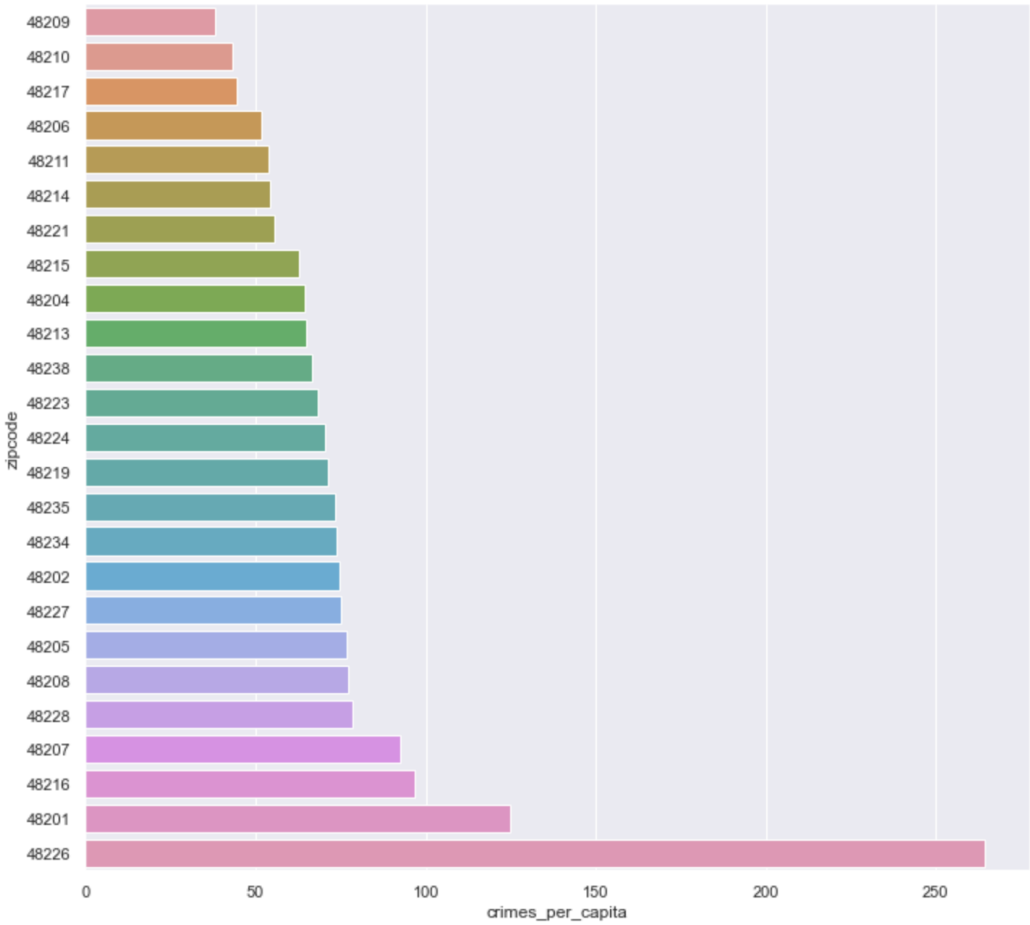
- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - The graph shows a map of Detroit with color indicating the crime count, which yellow is telling us the crime count is extremely high and blue is fairly low. From the graph above, we can see that a lot of the crimes actually happen in the boundaries of the Detroit city limit especially the Northwest side of Detroit and Northeast. It is definitely interesting to find out that the downtown area is actually not one of the area that has the highest crime count in the city. Hence, the most dangerous zipcode based on solely crime count would be the one on the West side in Detroit. According to neighbourhoodscout.com, the top most dangerous neighbourhood back in 2019 was the Tireman West neighbourhood. This is perfectly aligned with the result and plot I got from this particular question as Tireman West is located in the zipcode that is colored yellow in the plot. It is surprising that after 4 years the area that has the most crime count is still around the same area. Moreover, it is also surprising how the remaining neighbourhoods mentioned from the article are not downtown which also aligns with my plot as downtown is not the worse one compared to other neighborhoods (Michigan Radio, 2019).
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - It would be better to actually find out what is the crime count per capita which basically is more suitable to conclude whether a zipcode is actually more dangerous or safer. This will be done in the following question. It would also be better if there is another graph that is showing the zipcodes than just showing the map without any idea which area is indicating which zipcode.

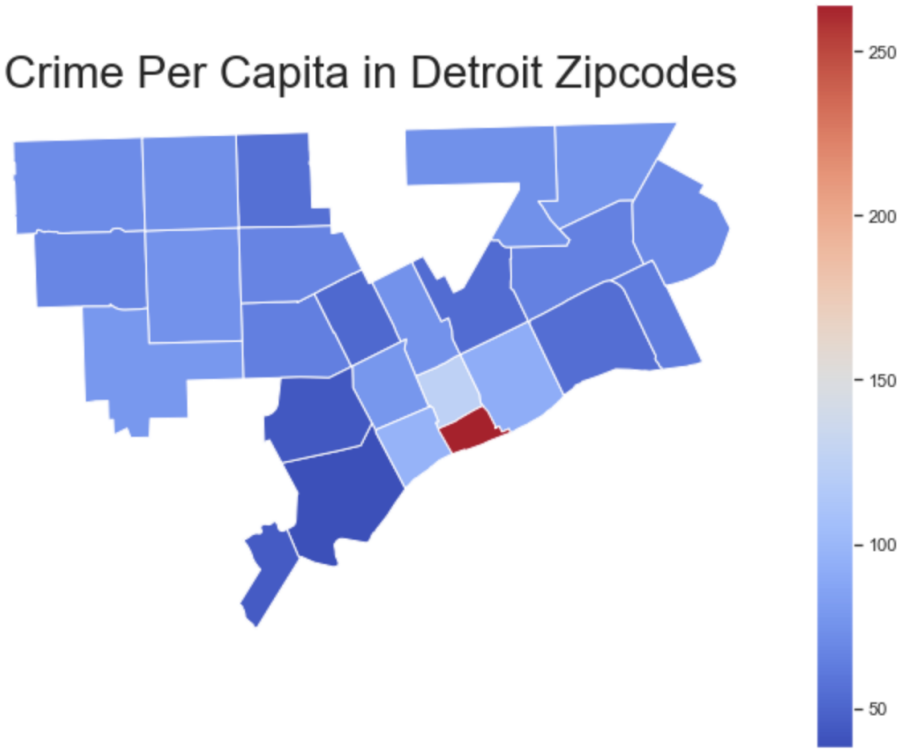
Question 19 (Frederick)

- **What is the question (~2 sentence)**

- What is the crime per capita in Detroit zipcodes?

- **Why does it matter (~3 sentences)**
  - This matters more than the previous question as this definitely tells a zipcode is safer or more dangerous. If the crime count is significantly higher than the population in it, it means that the zipcode is really dangerous, while if the crime count is high but the population is higher than the crime count, then a zipcode can be considered a bit safer than just the statistics of the crime count. This would be able to tell law enforcers and police departments to really focus on which part of the city desperately needs help and more allocation of resources and the tighten on law. Same like the previous question, visitors would be able to know which part of the city that is safe to visit and which one to avoid.
- **How you approached answering the question at a high level (3-4 sentences)**
  - I created a new dataframe based on the data I got from mapzicodes.com, which contains the population, median age, income per capita, median home value, and median gross rent of each zipcode. I then merged this new dataframe to the detroit rms data to have the full statistics of the zipcode in one dataframe. I then created a column to calculate the crimes per capita by dividing the population column from the crime count column. I then plot two graphs which one is showing a bar graph telling the crime per capita of all the zipcodes and another is a map plot showing the zipcode color coded with the crime per capita of each respective zipcode.
- **The plot showing your best answer so far**


48209
48210
48217
48206
48211
48214
48221
48215
48204
48213
48238
48223
48224
48219
48235
48234
48202
48227
48205
48208
48228
48207
48216
48201
48226
zipcode
0
50
100
150
200
250
crimes_per_capita



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - From the graphs above, it tells us that the most dangerous zipcode based on crimes per capita is 48226 which is downtown Detroit. This definitely makes sense as the number of people living in downtown is relatively lower than the number of crime count there. From the previous question, the zipcode in the Eastern Detroit showed that it has the highest crime count from the other zipcodes in Detroit, but in these graphs it seems like it is around the typical rate or crime per capita which means that it has a higher population relative to the crime count in that specifc zipcode. The zipcodes around downtown can be seen to also have a lighter shade of blue meaning that crime is more than the population meaning that it is more dangerous.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**

- I think it would be more interesting to find out if there is an affect between crime count and other variables in the dataframe like median age, income per capita, median home value, and median gross rent one by one. Creating more graphs would also be beneficial if wanting to analyze more of the relationship between these variables.

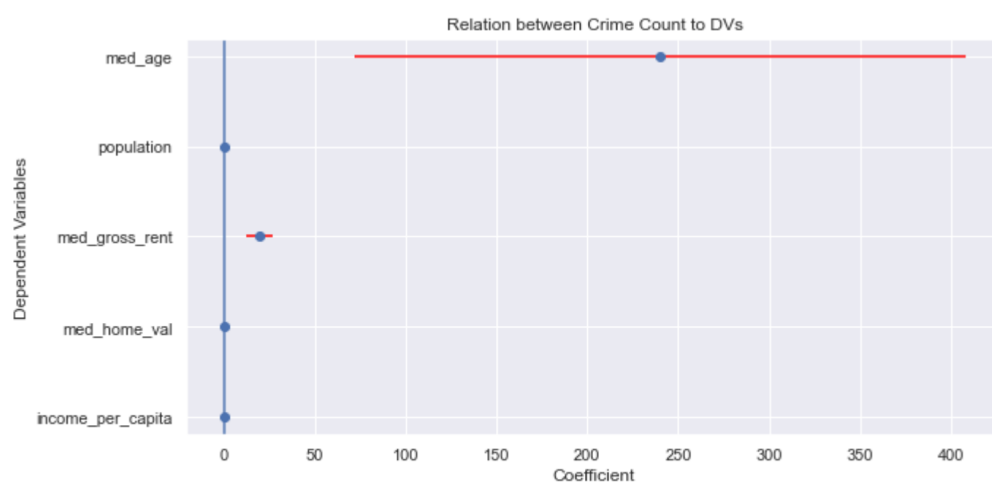
Question 20 (Frederick)

- **What is the question (~2 sentence)**
  - How does crime count affect the income per capita, median home value, median gross rent, population, and the median age of each Detroit zipcodes?
- **Why does it matter (~3 sentences)**
  - This matters because government can view this as a way to understand why crime is possibly really high in a certain area. Moreover, government could also see this opportunity to think of potential ways to allocate budget and try to invest certain resources and business towards those areas. Business owners could also use this as a way to determine which zipcode is the best for setting up their business.
- **How you approached answering the question at a high level (3-4 sentences)**
  - Since I already have the complete data frame from the previous question, I created a linear regression model and set the crime count as the independent variables and set the median age, income per capita, median home value, and the median gross rent as the dependent variable. And then I created a data frame from the model, and omitted the intercept in the new dataframe. I then reset the index to put make all the variables in a column and in the end I plot an errorbar plot. I also put a zero bar to treat it as a guideline in the plot.
- **The plot showing your best answer so far**

Dep. Variable:	crime_count	**R-squared:**	0.935
Model:	OLS	**Adj. R-squared:**	0.918
Method:	Least Squares	**F-statistic:**	54.75
Date:	Mon, 17 Apr 2023	**Prob (F-statistic):**	1.26e-10
Time:	13:40:53	**Log-Likelihood:**	-233.17
No. Observations:	25	**AIC:**	478.3
Df Residuals:	19	**BIC:**	485.6
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.856e+04	7584.617	-3.765	0.001	-4.44e+04	-1.27e+04
income_per_capita	-0.0537	0.019	-2.899	0.009	-0.092	-0.015
med_home_val	0.1079	0.026	4.200	0.000	0.054	0.162
med_gross_rent	19.6163	7.359	2.666	0.015	4.214	35.018
population	0.6675	0.055	12.034	0.000	0.551	0.784
med_age	239.7615	168.199	1.425	0.170	-112.282	591.806

Omnibus:	0.314	**Durbin-Watson:**	2.211
Prob(Omnibus):	0.855	**Jarque-Bera (JB):**	0.139
Skew:	-0.171	**Prob(JB):**	0.933
Kurtosis:	2.868	**Cond. No.**	9.61e+05



- **What is the answer to the question based on the plot (~1 paragraph of 4-5 sentences)**
    - The model doesn't really tell much especially for people who don't understand statistics and linear regression. The R value is also inaccurate as it is too high due to me using all the variables in the data frame. I plot the errorbar showing the variables relation with crime count. It can be seen that median age has the highest effect on crime count but it also has a really wide errorbar which makes it a little bit unprecise. The second most impactful variable is the median gross rent. And the rest variables are on the 0 bar meaning they don't really have much affect on crime count even though they have little to no errors.
- **What would you differently or next if you wanted to dig deeper (~2 sentences)**
    - I think it would be helpful to create individual plots respective to the relation of each variable with crime count. This would tell more specific whether they have a relationship or not.

Works Cited:

"Detroit Zip Code Map." *Find Cities, Counties and Zip Codes*, from http://www.mapszipcode.com/michigan/detroit-zip-code-map/.

Foreign Direct Investment: Michigan business. Michigan Economic Development Corporation (MEDC). (n.d.). Retrieved April 20, 2023, from https://www.michiganbusiness.org/why-michigan/fdi/

Indeed Editorial Team. (2022, June 24). *Rolling averages: What they are and how to calculate them - indeed*. Indeed. Retrieved April 20, 2023, from https://www.indeed.com/career-advice/career-development/what-is-rolling-average

Michigan Radio | By Mark Brush. "The 6 Most Dangerous Neighborhoods in Michigan." *Michigan Radio*, Retrieved 15 Jan. 2019, https://www.michiganradio.org/law/2013-05-31/the-6-most-dangerous-neighborhoods-in-michigan.

Miranda, M. L., Kim, D., Galeano, M. A. O., Paul, C. J., Hull, A. P., & Morgan, S. P. (2007, August). *The relationship between early childhood blood lead levels and performance on end-of-grade tests*. Environmental health perspectives. Retrieved April 19, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1940087/

Oakland County, MI. Foreign Firms. (n.d.). Retrieved April 20, 2023, from https://www.oakgov.com/business/business-development/international/foreign-firms

Appendix

```
sqlite-utils install -U
```

```
Collecting sqlite-utils
  Downloaded https://files.pythonhosted.org/packages/5c/f0/a3a7ccb4f2d7a7a7a
```

In 1519.

```
#how many projects have approved and how many days them to get approved
#total are projects 58

approved_projects = a2_planning_development[a2_planning_development['STATUS']== 'APPROVED'] | (a2_planning_development['STATUS']== 'APPROVED W/COND')]
#32 projects are approved
```

In 1520.

```
#approved_projects = a2_planning_development[a2_planning_development['APPROVED_DATE'] != 'NaT']
```

In 1143.

```
#approved_projects.columns
```

In 1522.

```
approved_projects['difference'] = approved_projects['APPROVED_DATE']-approved_projects['APPLIED_DATE']
#days
approved_projects['difference'] = approved_projects['difference'] /np.timedelta64(1,'D')
#a2_planning_development
```

```
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/4140554257.py:11: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  approved_projects['difference'] = approved_projects['APPROVED_DATE']-approved_projects['APPLIED_DATE']
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/4140554257.py:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  approved_projects['difference'] = approved_projects['difference'] /np.timedelta64(1,'D')
```

In 1523.

```
approved_projects = approved_projects.groupby(['OWNER_NAME', 'PROJECTTYPE'])['difference'].mean()
```

In 1524.

```
approved_projects = approved_projects.reset_index()
```

In 1527.

```
approved_projects.sort_values(by='difference', inplace=True)
approved_projects.head()
```

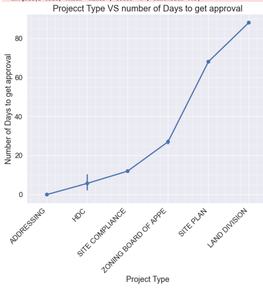
Out[1527]:

	OWNER_NAME	PROJECTTYPE	difference
0	325 E SUMMIT CONDOS, LLC	ADDRESSING	0.0

In 1119.

```
sns.set(rc={'figure.figsize':(15,9)),font_scale=1.5)
font = {'size'   : 20, 'family':'sans-serif','sans-serif':'Arial'}
plt.rc('font', **font)

fig, ax = plt.subplots(), figsize=(10,8))
g= sns.pointplot(data=approved_projects, x='PROJECTTYPE', y = 'difference', marker='*' )
plt.xticks(rotation = 45, ha='right')
g.set_title('Project Type VS number of days to get approval',fontsize=20)
                #,fontdict = {'fontsize': 20})
ax.set(xlabel='Project Type')
ax.set(ylabel='Number of Days to get approval')
plt.legend(bbox_to_anchor=(1.2, 0.15), loc='center', borderaxespad=0)
ax.get_xaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
ax.get_yaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
ax.grid(b=True, which='major', color='w', linewidth=1.0)
ax.grid(b=True, which='minor', color='w', linewidth=0.5)
```

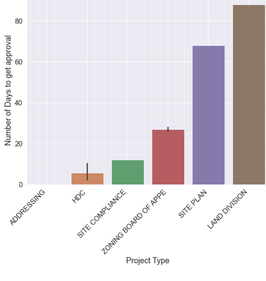
```
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/2249546807.py:15: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible' s
ince Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  ax.grid(b=True, which='major', color='w', linewidth=1.0)
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/2249546807.py:16: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible' s
ince Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  ax.grid(b=True, which='minor', color='w', linewidth=0.5)
```



In 1119.

```
sns.set(rc={'figure.figsize':(15,9)),font_scale=1.5)
font = {'size'   : 20, 'family':'sans-serif','sans-serif':'Arial'}
plt.rc('font', **font)
fig, ax = plt.subplots(), figsize=(10,8))
sns.barplot(data=approved_projects, x='PROJECTTYPE', y= 'difference')
plt.xticks(rotation = 45, ha='right')
ax.set_title('Bar plot: project type Vs. number of day to get approval',fontsize=20)
ax.set(xlabel='Project Type')
ax.set(ylabel='Number of Days to get approval')
ax.get_xaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
ax.get_yaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
ax.grid(b=True, which='major', color='w', linewidth=1.0)
ax.grid(b=True, which='minor', color='w', linewidth=0.5)
```

```
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/2235534087.py:12: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible'
since Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  ax.grid(b=True, which='major', color='w', linewidth=1.0)
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/2235534087.py:13: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible'
since Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  ax.grid(b=True, which='minor', color='w', linewidth=0.5)
```



Question 5

What are trends in top 5 crimes in Ann Arbor for the year 2022, and 2023 on each day of week as compared to Detroit crime data sets for the same years?

In 1117.

```
ann_crimes_df = pd.read_csv('AnnArbor-PartABCrimeOffenses.csv')
```

In 1143.

```
ann_crimes_df['timestamp'] = pd.to_datetime(ann_crimes_df['DATE'])
ann_crimes_df['dayofweek'] = ann_crimes_df['timestamp'].dt.dayofweek
ann_crimes_df['year'] = ann_crimes_df['timestamp'].dt.year
#ann_crimes_df['dayofweek']
#ann_crimes_df['year']
```

In 1143.

```
days = pd.Series({0:'Mon', 1:'Tu', 2:'Wed', 3:'Thu',4:'Fri',5:'Sat', 6:'Sun'})
ann_crimes_df['dayofweek'] = ann_crimes_df['dayofweek'].map(days)
#ann_crimes_df['dayofweek']
```

In 1113.

```
ann_crimes_df.groupby(['CATEGORY'])['CASE_NUM'].count().reset_index().sort_values(by='CASE_NUM',ascending=False).head(5)
```

Out[1136]:

	CATEGORY	CASE_NUM
17	LARCENY - ALL OTHER	826
18	LARCENY - FROM AUTO (LFA)	679
3	ASSAULT - SIMPLE	585
12	FRAUD	556
6	DAMAGE TO PROPERTY	395

In 1113.

```
ann_crimes_df5 = ann_crimes_df[(ann_crimes_df['CATEGORY'] =='LARCENY - ALL OTHER') |(ann_crimes_df['CATEGORY'] =='LARCENY - FROM AUTO (LFA)')| (ann_crimes_df['CATEGORY'
```

In 1113.

```
ann_crimes_df_top_5 = ann_crimes_df5.groupby(['CATEGORY','dayofweek')]['CASE_NUM'].count().reset_index()
ann_crimes_df_top_5.head(1)
```

Out[1138]:

	CATEGORY	dayofweek	CASE_NUM
0	ASSAULT - SIMPLE	Fri	79

In 1113.

```
ann_crimes_df_top_5.rename(columns={'CASE_NUM':'count'}, inplace=True )
```

Out[1139]:

	CATEGORY	dayofweek	count
0	ASSAULT - SIMPLE	Fri	79

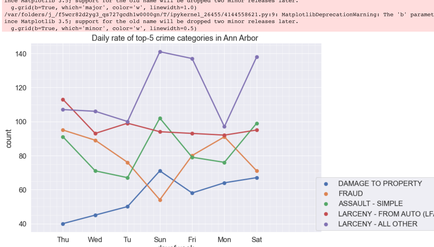
In 1128.

```
ann_crimes_df_top_5 = ann_crimes_df_top_5.sort_values(by='count')
```

In 1128.

```
sns.set(rc={'figure.figsize':(12,8)),font_scale=1.5)
g= sns.pointplot(data=ann_crimes_df_top_5, x='dayofweek',y='count',hue='CATEGORY')
g.set_xlim(-1, 8)
g.set_title('Daily rate of top-5 crime categories in Ann Arbor ')
plt.legend(bbox_to_anchor=(1.2, 0.15), loc='center', borderaxespad=0)
g.get_xaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
g.get_yaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
g.grid(b=True, which='major', color='w', linewidth=1.0)
g.grid(b=True, which='minor', color='w', linewidth=0.5)
```

```
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/4144559621.py:8: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible' s
ince Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  g.grid(b=True, which='major', color='w', linewidth=1.0)
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/4144559621.py:9: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible' s
ince Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  g.grid(b=True, which='minor', color='w', linewidth=0.5)
```



In 1144.

```
detroit_crimes_df = pd.read_csv('Detroit-RMS_Crime_Incidents.csv')
#detroit_crimes_df.columns
```

```
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/3704404747.py:1: DtypeWarning: Columns (8,15,16) have mixed types. Specify dtype option on import or se
t low_memory=False.
  detroit_crimes_df = pd.read_csv('Detroit-RMS_Crime_Incidents.csv')
```

In 1144.

```
detroit_crimes_df['Date'] = pd.to_datetime(detroit_crimes_df['incident_timestamp'])
detroit_crimes_df['day_of_year'] = detroit_crimes_df['Date'].dt.dayofyear
detroit_crimes_df['day_of_week'] = detroit_crimes_df['Date'].dt.dayofweek
```

In 1144.

```
day_series = pd.Series({0:'Mon', 1:'Tu', 2:'Wed', 3:'Thu',4:'Fri',5:'Sat', 6:'Sun'})
detroit_crimes_df['day_of_week'] = detroit_crimes_df['day_of_week'].map(day_series)
#detroit_crimes_df['day_of_week']
```

In 1144.

```
detroit_crimes_top_categories = detroit_crimes.groupby('offense_category')['crime_id'].count().sort_values(ascending=False)
#detroit_crimes_top_categories.head(5)
```

In 1144.

```
#detroit_crimes_df.year.unique()
```

In 1145.

```
detroit_crimes_top_categories = detroit_crimes_df[(detroit_crimes_df['offense_category'] =='ASSAULT') |(detroit_crimes_df['offense_category'] =='LARCENY')| (detroit_cri
```

In 1145.

```
detroit_crimes_top_categories = detroit_crimes_top_categories[(detroit_crimes_top_categories['year'] == 2022) | (detroit_crimes_top_categories['year'] == 2023)]
```

In 1145.

```
#detroit_crimes_top_categories.columns
```

In 1145.

```
detroit_crimes_top_categories = detroit_crimes_top_categories.groupby(['offense_category','day_of_week')]['crime_id'].count().reset_index()
```

In 1120.

```
detroit_crimes_top_categories.rename(columns={'crime_id':'count'}, inplace=True )
detroit_crimes_top_categories.head(1)
```

Out[1204]:

	offense_category	day_of_week	count
4	AGGRAVATED ASSAULT	Thu	1045

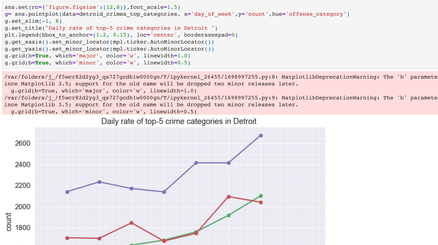
In 1121.

```
detroit_crimes_top_categories = detroit_crimes_top_categories.sort_values(by='count')
```

In 1121.

```
sns.set(rc={'figure.figsize':(12,8)),font_scale=1.5)
g= sns.pointplot(data=detroit_crimes_top_categories, x='day_of_week',y='count',hue='offense_category')
g.set_xlim(-1, 8)
g.set_title('Daily rate of top-5 crime categories in Detroit ')
plt.legend(bbox_to_anchor=(1.2, 0.15), loc='center', borderaxespad=0)
g.get_xaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
g.get_yaxis().set_minor_locator(mpl.ticker.AutoMinorLocator())
g.grid(b=True, which='major', color='w', linewidth=1.0)
g.grid(b=True, which='minor', color='w', linewidth=0.5)
```

```
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/1698997255.py:8: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible' s
ince Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  g.grid(b=True, which='major', color='w', linewidth=1.0)
/var/folders/j_f5wcr82d2yq1_qs727qcdhlw0000gn/T/ipykernel_26455/1698997255.py:9: MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed 'visible' s
ince Matplotlib 3.5; support for the old name will be dropped two minor releases later.
  g.grid(b=True, which='minor', color='w', linewidth=0.5)
```



In [ ]:

In [ ]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
import datetime
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from nltk.sentiment import SentimentIntensityAnalyzer
```

In [ ]:
```python
detroit_911_calls = pd.read_csv('Detroit-911_Calls_for_Service_(Last_30_Days).csv')
detroit_fire = pd.read_csv('Detroit-Fire_Incidents.csv')
ford_stocks = pd.read_csv("Ford-Stock-Prices.csv")
df_michigan_weather = pd.read_csv('se-michigan-weather-data.csv')
df_michigan_bird = pd.read_csv('se-michigan_birding_data.csv', sep='\t')
detroit_trades_permits = pd.read_csv('Detroit-TradesPermits.csv')
detroit_demolition_permits = pd.read_csv('Detroit0DemolitionPermits.csv')
df_unemployment = pd.read_csv('Unemployment_Rate.csv')
```

```
C:\Users\paru\AppData\Local\Temp\ipykernel_8656\1637098283.py:2: DtypeWarning: Columns (14) have mixed types. Specify dtype option
on import or set low_memory=False.
  detroit_fire = pd.read_csv('Detroit-Fire_Incidents.csv')
C:\Users\paru\AppData\Local\Temp\ipykernel_8656\1637098283.py:4: DtypeWarning: Columns (15) have mixed types. Specify dtype option
on import or set low_memory=False.
  df_michigan_weather = pd.read_csv('se-michigan-weather-data.csv')
```

Question 1.Group by and aggregate function

Identify different types of fires incidents. How long did it take for the fire to be disposed from the time the fire incident was reported?

In [ ]:
```python
df = detroit_fire.copy()

df['call_datetime'] = pd.to_datetime(df['call_datetime'])

df['cleared_datetime'] = pd.to_datetime(df['cleared_datetime'])

def calculate_difference(datetime):
    return (datetime['cleared_datetime'] - datetime['call_datetime']).seconds / 60

df['time_to_dispose'] = df.apply(calculate_difference, axis=1)

df = df.groupby('incident_type_desc')['time_to_dispose'].mean()
```

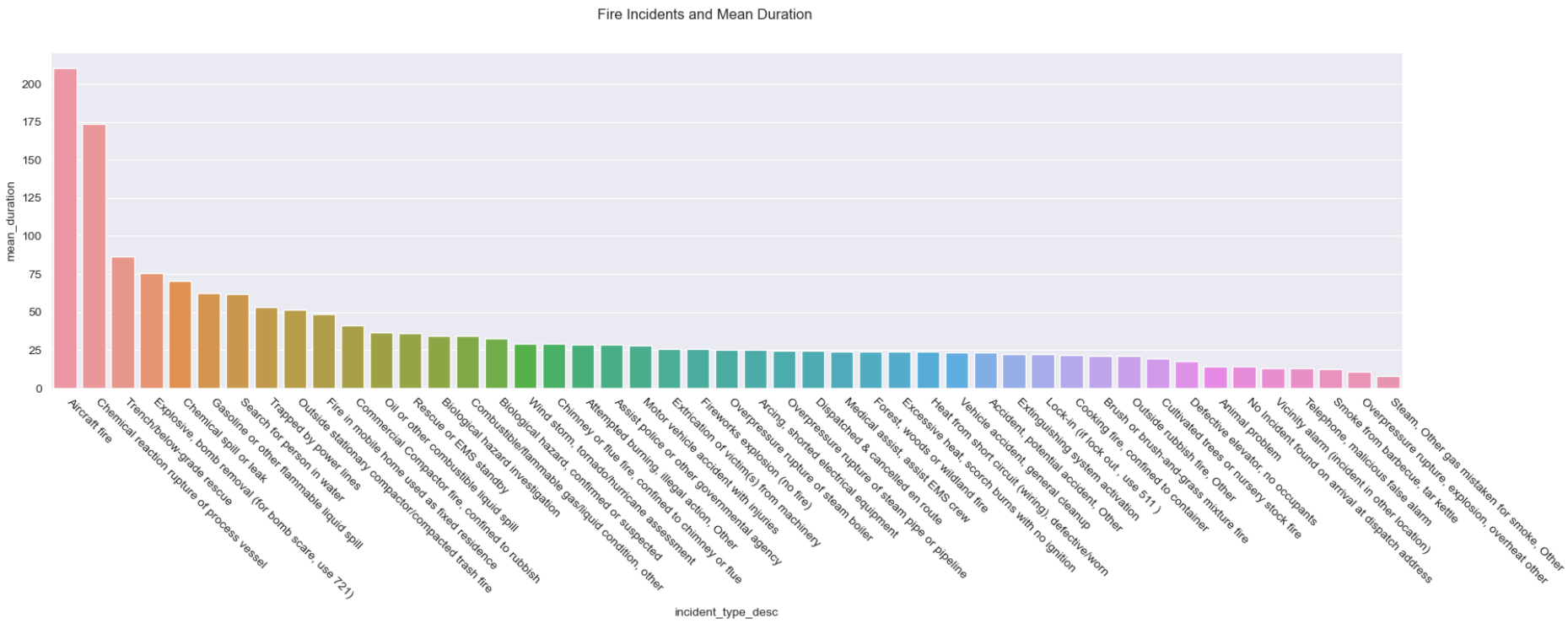
```
df = df.sample(frac=.3).sort_values(ascending=False)

df = df.to_frame(name='mean_duration')

df = df.reset_index()

df_avg_duration = df.copy()

df.head(2)
```

Out[ ]:

	incident_type_desc	mean_duration
0	Aircraft fire ...	210.166667
1	Chemical reaction rupture of process vessel ...	173.491667

In [ ]:

```
df = df_avg_duration.copy()

sns.set_style("darkgrid")

fig, ax = plt.subplots(1, 1)

fig.set_size_inches((20,5))

sns.barplot(data=df, y='mean_duration', x='incident_type_desc', ax=ax)

fig.suptitle('Fire Incidents and Mean Duration')

_ = ax.set_xticks(ticks=ax.get_xticks(), labels=ax.get_xticklabels(), rotation=-45, ha='left')
```



2. Time series analysis requiring rolling average and groupby

What are the rolling average trends of the high and low prices for Ford Stocks over the years from 1970 to 2020?

In [ ]:
```
df = ford_stocks.copy()

# convert the date to datetime
df['New_Date'] = df['Date'].apply(pd.to_datetime)

df['year'] = df['New_Date'].dt.year

#set the date as index
df = df.set_index(df['New_Date'])

df = df.groupby('year')[['High', 'Low']].mean()

df['High_rolling_7d'] = df['High'].rolling(7).mean()

df['Low_rolling_7d'] = df['Low'].rolling(7).mean()

df_rolling_7 = df.copy()
```

```
df.head(2)
```

Out[ ]:

	High	Low	High_rolling_7d	Low_rolling_7d
year				
1972	2.220183	2.188826	NaN	NaN
1973	1.915582	1.875917	NaN	NaN

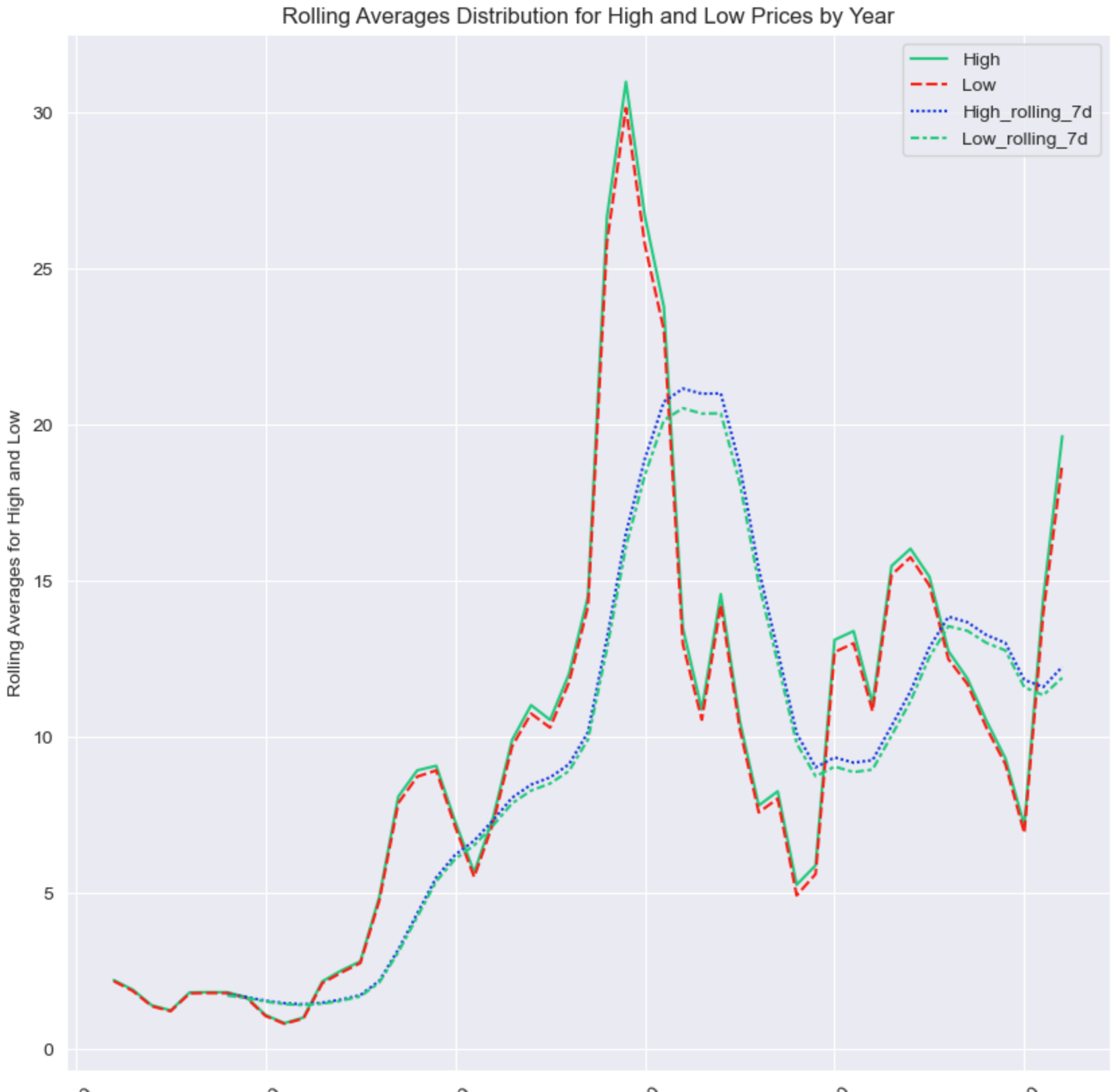
In [ ]:

```
df = df_rolling_7.copy()

fig, ax = plt.subplots(1, 1)

fig.set_size_inches((10,10))

ax = sns.lineplot(data=df, palette=['#24c97f','#f52116','#1438db'])

ax.tick_params(axis='x', rotation=45)

ax.set(title="Rolling Averages Distribution for High and Low Prices by Year", xlabel='Year', ylabel='Rolling Averages for High and
```

```
C:\Users\paru\AppData\Local\Temp\ipykernel_8656\4038688487.py:7: UserWarning:
The palette list has fewer values (3) than needed (4) and will cycle, which may produce an uninterpretable plot.
  ax = sns.lineplot(data=df, palette=['#24c97f','#f52116','#1438db'])
```

Out[ ]:

```
[Text(0.5, 1.0, 'Rolling Averages Distribution for High and Low Prices by Year'),
 Text(0.5, 0, 'Year'),
 Text(0, 0.5, 'Rolling Averages for High and Low')]
```


Rolling Averages Distribution for High and Low Prices by Year
High
Low
High_rolling_7d
Low_rolling_7d
Rolling Averages for High and Low
0
5
10
15
20
25
30

3. Text analysis requiring inferring sentiment aspect of text

Is there a positive relationship between negative sentiment of call description and type of the offense? Does more serious crime get a negative sentiment score?

In [ ]:
```
df = detroit_911_calls.copy()

sa = SentimentIntensityAnalyzer()

df['sentiment_score'] = df['calldescription'].apply(lambda x:sa.polarity_scores(x)['compound'])

df = df.loc[df['priority'].notna()]

df['priority'] = pd.to_numeric(df['priority'], errors='coerce')

df = df.loc[df['priority'].notna()]

df_category = df.copy()

df_category.head(2)
```

Out[ ]:

	X	Y	incident_id	agency	incident_address	zip_code	priority	callcode	calldescription	category	...	totalresponsetime
0	-83.108512	42.373040	202236400038	DPD	Linwood St & Clairmount St	48206	3.0	SA	SPECIAL ATTENTION	SPCL ATT	...	0.0
1	-83.160790	42.425981	202236400039	DPD	Pickford St & Wyoming St	48221	3.0	935030	REMARKS	REMARKS	...	84.2

2 rows × 27 columns

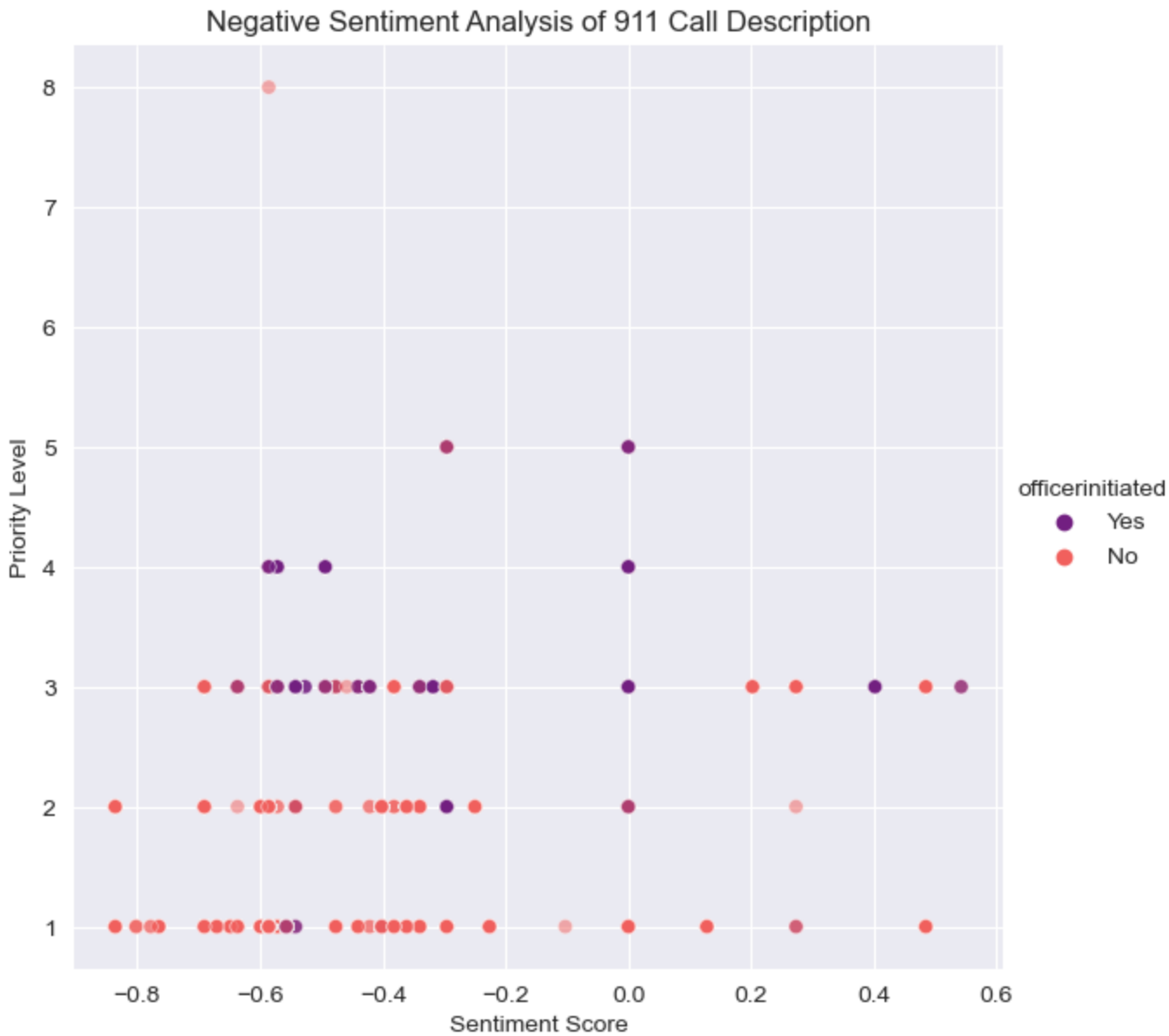
In [ ]:
```
df = df_category.copy()

df['priority'].corr(df['sentiment_score'])
```

Out[ ]: 0.3911946764465255

In [ ]:
```
df = df_category.copy()
```

```
ax = sns.relplot(x="sentiment_score", y="priority", hue='officerinitiated', palette=("magma"),
            sizes=(40, 400), alpha=.5,
            height=6, data=df)


ax.set(title="Negative Sentiment Analysis of 911 Call Description", xlabel='Sentiment Score', ylabel='Priority Level')
```

Out[ ]: <seaborn.axisgrid.FacetGrid at 0x2620421cf40>



4. groupby and aggregate

How does bird frequency change in relation to the change in temperature in south-east Michigan? Are some birds more migratory than others?

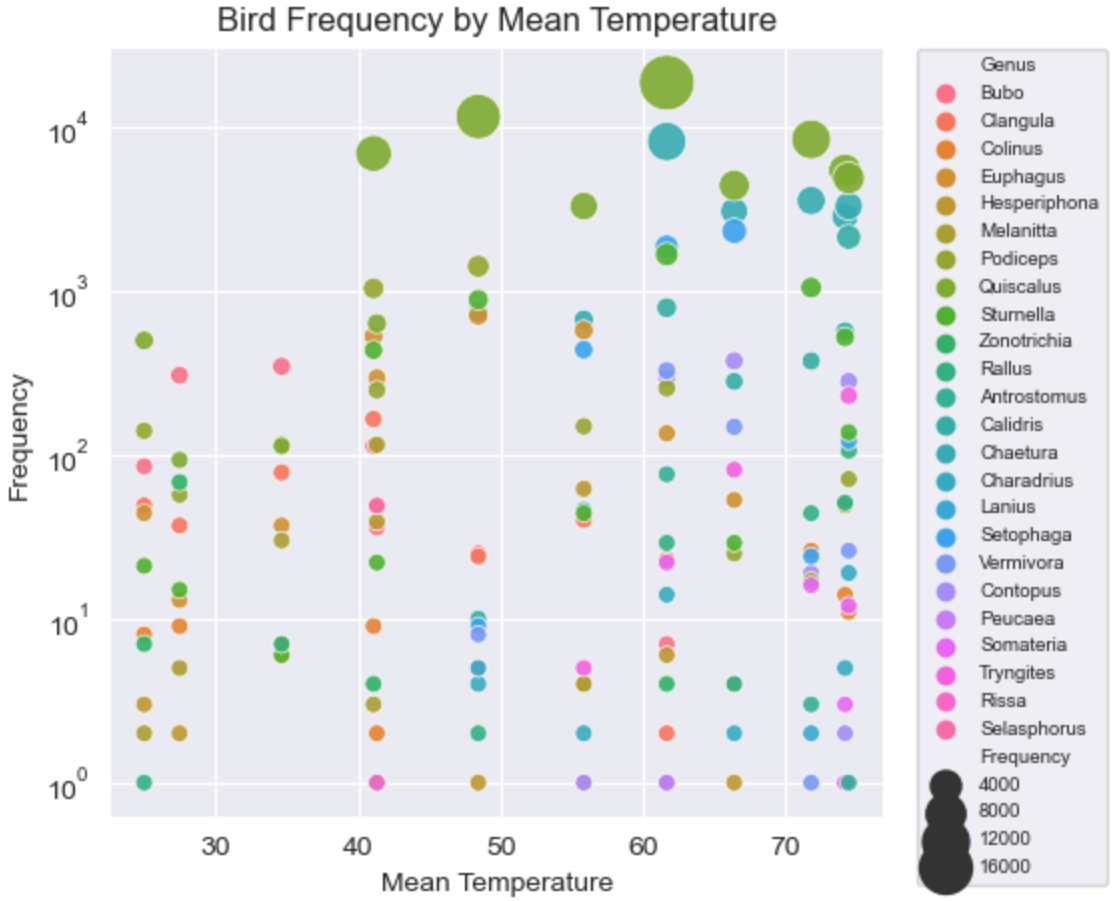
In [ ]:
```
df = df_michigan_bird.copy()

df = df[['genus', 'eventDate']]

df['month'] = pd.to_datetime(df['eventDate']).dt.month

df = df[['genus', 'month']]

df_species_month = df.copy()

df = df_michigan_weather.copy()

df = df[['DATE', 'TAVG']]

df = df.dropna()

df['month'] = pd.to_datetime(df['DATE']).dt.month

df['mean_temp'] = pd.to_numeric(df['TAVG'])

df = df.groupby(['month'])['mean_temp'].agg(['mean'])

df = df.reset_index()

df = df.rename({'mean': 'mean_temp'}, axis=1)

df = df.merge(df_species_month, how='inner')

df = df.groupby(['month', 'mean_temp', 'genus'])['genus'].agg(lambda x: x.shape[0])

df = df.to_frame(name='count').reset_index()

df_genus = df.copy()

df.head(2)
```

Out[ ]:

	month	mean_temp	genus	count
0	1	27.448718	Bubo	305
1	1	27.448718	Clangula	37

In [ ]:
```
df = df_genus.copy()
```

```python
fig, ax = plt.subplots(1)

fig.set_size_inches(5,5)

ax.set_xlabel('Mean Temperature')

ax.set_ylabel('Frequency')

sns.set_style("darkgrid")

ax = sns.scatterplot(data=df, x="mean_temp", y="count", hue="genus", alpha=0.9, size='count', sizes=(40, 400), ax=ax)

h, l = ax.get_legend_handles_labels()

l = [{'genus': 'Genus', 'count': 'Frequency'}.get(label, label) for label in l]

sns.set_style("darkgrid")

ax.legend(loc='upper right', bbox_to_anchor=(1.3, 1.01), fontsize='x-small', handles=h, labels=l)

ax.set_yscale('log')

_ = ax.set_title('Bird Frequency by Mean Temperature')
```



5. Linear regression analysis involving more than 5 independent variables

How does the volume of the Ford stock changes by the change in other variables? Are there variables that are statistically significant at $p<0.05$? What is the status of standard errors?

In [ ]:
```
df = df_unemployment.copy()

df['Calendar Date'] = pd.to_datetime(df['Calendar Date'])

df_conversion_unemp = df.copy()

df = ford_stocks.copy()

df['Date'] = pd.to_datetime(df['Date'])

df_conversion_stock = df.copy()
```

In [ ]:
```
df = df_conversion_unemp.copy()

df1 = df_conversion_stock.copy()

df = df.merge(df1, how='inner', left_on='Calendar Date', right_on='Date')

combined = df.copy()

df.head(2)
```

Out[ ]:

	Calendar Date	Date_x	Unemployment_Rate_MI	Unemployment_Rate_US	Rank	Highlights	Date_y	Open	High	Low	Close	C
0	1976-03-01	Mar-76	0.099	0.076	NaN	NaN	1976-03-01	1.658506	1.674727	1.646341	1.666617	0.112
1	1976-04-01	Apr-76	0.098	0.077	NaN	NaN	1976-04-01	1.849093	1.861258	1.836928	1.840983	0.124

In [ ]:
```
df = combined.copy()

model = smf.ols('Volume ~ High + Low + Open + Close + Unemployment_Rate_MI + Unemployment_Rate_US', data=df)

result = model.fit()

result.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	Volume	**R-squared:**	0.222
Model:	OLS	**Adj. R-squared:**	0.209
Method:	Least Squares	**F-statistic:**	16.58
Date:	Thu, 20 Apr 2023	**Prob (F-statistic):**	7.86e-17
Time:	15:56:59	**Log-Likelihood:**	-6634.5
No. Observations:	355	**AIC:**	1.328e+04
Df Residuals:	348	**BIC:**	1.331e+04
Df Model:	6		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-2.335e+07	8.77e+06	-2.663	0.008	-4.06e+07	-6.1e+06
High	6.948e+07	1.66e+07	4.174	0.000	3.67e+07	1.02e+08
Low	-4.875e+07	1.4e+07	-3.486	0.001	-7.63e+07	-2.12e+07
Open	1.629e+07	1.47e+07	1.110	0.268	-1.26e+07	4.51e+07
Close	-3.776e+07	1.5e+07	-2.517	0.012	-6.73e+07	-8.25e+06
Unemployment_Rate_MI	-2.084e+08	1.58e+08	-1.320	0.188	-5.19e+08	1.02e+08
Unemployment_Rate_US	9.121e+08	2.49e+08	3.669	0.000	4.23e+08	1.4e+09

Omnibus:	148.258	**Durbin-Watson:**	0.701
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	929.482
Skew:	1.630	**Prob(JB):**	1.46e-202
Kurtosis:	10.226	**Cond. No.**	4.19e+03

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

[2] The condition number is large, 4.19e+03. This might indicate that there are strong multicollinearity or other numerical problems.

In [ ]:
```
df = combined.copy()

model = smf.ols('Volume ~ High + Low + Open + Close + Unemployment_Rate_MI + Unemployment_Rate_US', data=df)

result = model.fit()

result.summary()
df = pd.DataFrame(result.summary().tables[1][1:])
df = df.rename(columns={0:'Variable', 1:'Coefficient', 2:'Standard Error', 4:'p-value'}).drop(columns=[3, 5, 6])

df_results = df.copy()
```

In [ ]:
```
df = df_results.copy()

df['Variable'] = df['Variable'].astype(str)
df['Coefficient'] = pd.to_numeric(df['Coefficient'].apply(lambda x: x.data))
df['Standard Error'] = pd.to_numeric(df['Standard Error'].apply(lambda x: x.data))
df['p-value'] = pd.to_numeric(df['p-value'].apply(lambda x: x.data))

df = df.iloc[1:]
df_conversion_types = df.copy()

df.head(2)
```

Out[ ]:

	Variable	Coefficient	Standard Error	p-value
1	High	69480000.0	16600000.0	0.000
2	Low	-48750000.0	14000000.0	0.001

In [ ]:
```
df =df_conversion_types.copy()

plt.figure(figsize=(10, 5))

sns.set_style("darkgrid")

color = [('red' if p_val < 1 else 'blue') for p_val in df['p-value']]

plt.errorbar(data=df, x='Coefficient', y='Variable', xerr='Standard Error', ecolor=color, ms=2, mew=8, fmt='o')

plt.suptitle('Standard Errors for Stock Volume and Other Variables')

plt.xlabel('Coefficient')
```

```
plt.ylabel('Independent Variables')

plt.axvline(linewidth = 1, linestyle ="--", color ='black')
```

Out[ ]: <matplotlib.lines.Line2D at 0x26204076bb0>

In [ ]:

Name: Elaine Wu

In [1]:
```
import pandas as pd
import datetime as dt
import seaborn as sns
import datetime as datetime
import statsmodels.api as sm
import statsmodels.formula.api as smf
from matplotlib import pyplot as plt
import geopandas as gpd
from shapely.geometry import Point
```

In [2]:
```
work = pd.read_csv('acs_work.csv')
income = pd.read_csv('acs_income.csv')
```

In [5]:
```
work_income = pd.merge(income, work, how='inner', on='tract')
```

Q1: how does various education levels and labor employment effects the different income levels in michigan counties"?

In [6]:
```
model = smf.ols('income_over_200k ~ labor_employed+C(county_x)+edu_associates+edu_bachelors'
                +'+edu_graduate', data=work_income).fit()
model.summary()
```

Out[6]:

OLS Regression Results

Dep. Variable:	income_over_200k	**R-squared:**	0.575
Model:	OLS	**Adj. R-squared:**	0.572
Method:	Least Squares	**F-statistic:**	161.8
Date:	Thu, 20 Apr 2023	**Prob (F-statistic):**	2.83e-277
Time:	15:52:50	**Log-Likelihood:**	2198.8
No. Observations:	1566	**AIC:**	-4370.
Df Residuals:	1552	**BIC:**	-4295.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.0042	0.019	-0.227	0.820	-0.041	0.032
C(county_x)[T.65]	-0.0340	0.018	-1.922	0.055	-0.069	0.001
C(county_x)[T.75]	-0.0080	0.018	-0.442	0.658	-0.044	0.028
C(county_x)[T.91]	0.0004	0.020	0.022	0.983	-0.040	0.040
C(county_x)[T.93]	0.0401	0.018	2.176	0.030	0.004	0.076
C(county_x)[T.99]	0.0120	0.017	0.701	0.483	-0.022	0.045
C(county_x)[T.115]	0.0164	0.019	0.859	0.391	-0.021	0.054
C(county_x)[T.125]	0.0201	0.017	1.176	0.240	-0.013	0.054
C(county_x)[T.161]	-0.0380	0.018	-2.128	0.033	-0.073	-0.003
C(county_x)[T.163]	0.0029	0.017	0.175	0.861	-0.030	0.036
labor_employed	-0.0602	0.018	-3.413	0.001	-0.095	-0.026
edu_associates	0.1026	0.041	2.505	0.012	0.022	0.183
edu_bachelors	0.1412	0.023	6.075	0.000	0.096	0.187
edu_graduate	0.5043	0.019	26.096	0.000	0.466	0.542

Omnibus:	274.920	**Durbin-Watson:**	1.325
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1338.963
Skew:	0.740	**Prob(JB):**	1.77e-291
Kurtosis:	7.282	**Cond. No.**	44.7

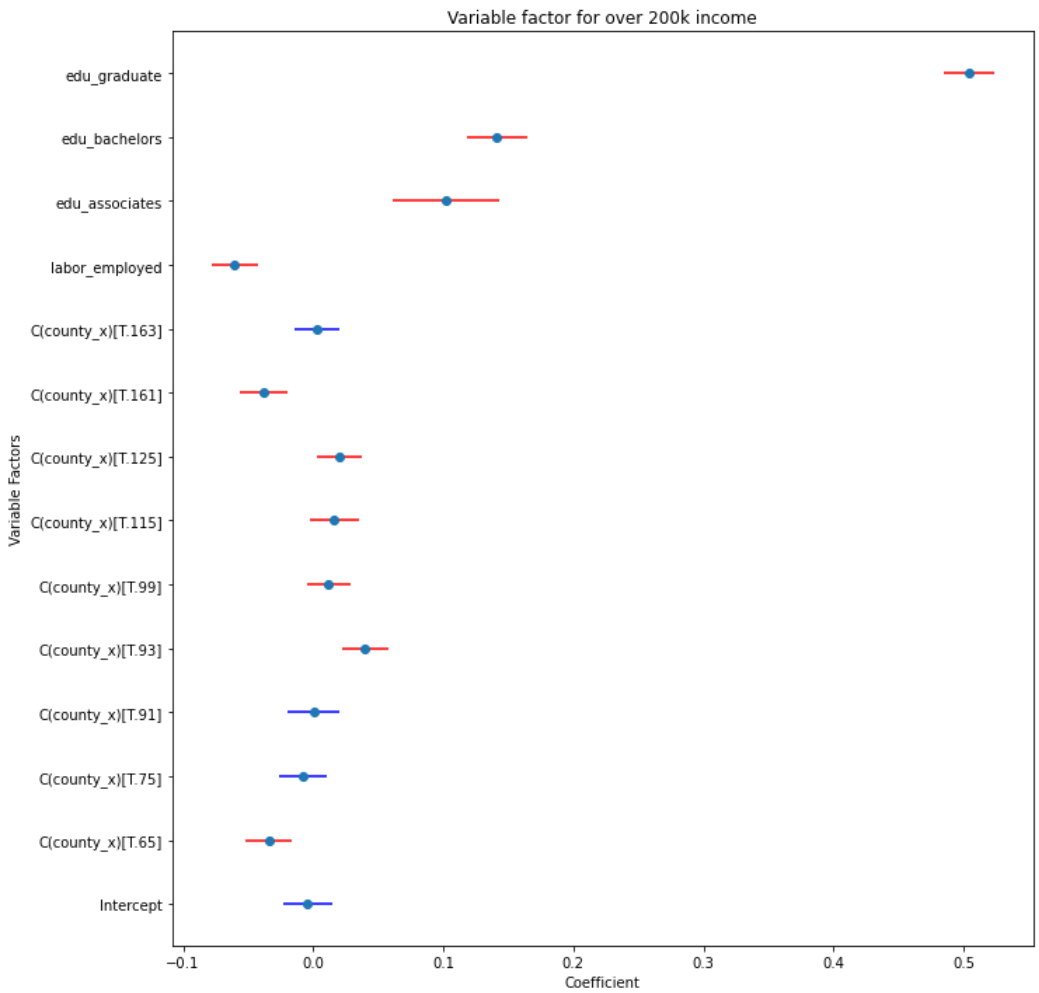
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [6]:
```
result_model = model.summary().tables[1].as_html()
result_model = pd.read_html(result_model, header =0, index_col=0)[0]
result_model_df=result_model[['coef', 'std err', 'P>|t|']]
result_model_df.reset_index(inplace=True)
result_model_df.rename(columns={'index':'variable'}, inplace=True)
```

```
/var/folders/xr/gjwnmnh16yq74f56w_0pz5tr0000gn/T/ipykernel_15748/2540478157.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  result_model_df.rename(columns={'index':'variable'}, inplace=True)
```

In [7]:
```
plt.figure(figsize=(11,12))
color = [('red' if p<0.5 else 'blue') for p in result_model_df['P>|t|']]
plt.errorbar(data=result_model_df, x='coef', y='variable',xerr='std err', ecolor=color,
             fmt='o')
plt.xlabel('Coefficient')
plt.ylabel('Variable Factors')
plt.title('Variable factor for over 200k income')
```

Out[7]:
```
Text(0.5, 1.0, 'Variable factor for over 200k income')
```



In [7]:
```
model_2 = smf.ols('income_10k_15k ~ labor_employed+C(county_x)+edu_none+edu_no_hs+edu_hs+edu_graduate'
                  +'+edu_associates', data=work_income).fit()
model_2.summary()
```

Out[7]:

OLS Regression Results

Dep. Variable:	income_10k_15k	**R-squared:**	0.301
Model:	OLS	**Adj. R-squared:**	0.294
Method:	Least Squares	**F-statistic:**	44.39
Date:	Thu, 20 Apr 2023	**Prob (F-statistic):**	7.01e-109
Time:	15:53:23	**Log-Likelihood:**	2936.7
No. Observations:	1566	**AIC:**	-5841.
Df Residuals:	1550	**BIC:**	-5756.
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0487	0.019	2.521	0.012	0.011	0.087
C(county_x)[T.65]	0.0165	0.011	1.486	0.137	-0.005	0.038
C(county_x)[T.75]	0.0158	0.011	1.388	0.165	-0.007	0.038
C(county_x)[T.91]	-0.0068	0.013	-0.536	0.592	-0.032	0.018
C(county_x)[T.93]	-0.0041	0.012	-0.356	0.722	-0.027	0.018
C(county_x)[T.99]	-0.0027	0.011	-0.249	0.803	-0.024	0.018
C(county_x)[T.115]	-0.0014	0.012	-0.118	0.906	-0.025	0.022
C(county_x)[T.125]	0.0084	0.011	0.789	0.430	-0.013	0.029
C(county_x)[T.161]	0.0260	0.011	2.329	0.020	0.004	0.048
C(county_x)[T.163]	0.0071	0.010	0.675	0.500	-0.014	0.028
labor_employed	-0.0946	0.011	-8.388	0.000	-0.117	-0.073
edu_none	0.0605	0.051	1.177	0.240	-0.040	0.161
edu_no_hs	0.1581	0.021	7.627	0.000	0.117	0.199
edu_hs	0.0580	0.015	3.744	0.000	0.028	0.088
edu_graduate	0.0099	0.024	0.421	0.674	-0.036	0.056
edu_associates	-0.0263	0.030	-0.872	0.383	-0.085	0.033

Omnibus:	709.063	**Durbin-Watson:**	1.677
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	6633.845
Skew:	1.876	**Prob(JB):**	0.00
Kurtosis:	12.359	**Cond. No.**	77.2

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

In [9]:

```
result_model_2 = model_2.summary().tables[1].as_html()
result_model_2 = pd.read_html(result_model_2, header =0, index_col=0)[0]
result_model_df_2=result_model_2[['coef', 'std err', 'P>|t|']]
result_model_df_2.reset_index(inplace=True)
result_model_df_2.rename(columns={'index':'variable'}, inplace=True)
result_model_df_2
```

```
/var/folders/xr/gjwnmnh16yq74f56w_0pz5tr0000gn/T/ipykernel_15748/1749436572.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  result_model_df_2.rename(columns={'index':'variable'}, inplace=True)
```

Out[9]:

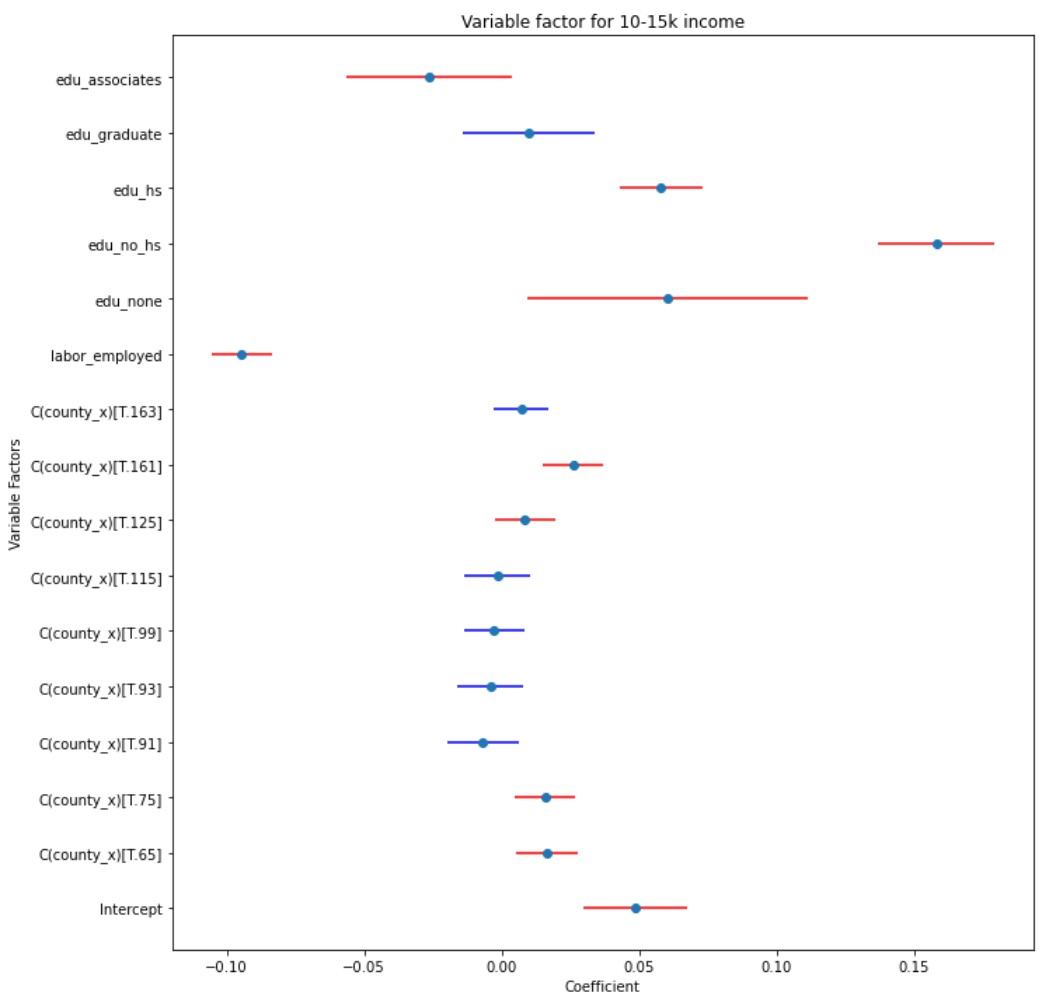
	variable	coef	std err	P>|t|
0	Intercept	0.0487	0.019	0.012
1	C(county_x)[T.65]	0.0165	0.011	0.137
2	C(county_x)[T.75]	0.0158	0.011	0.165
3	C(county_x)[T.91]	-0.0068	0.013	0.592
4	C(county_x)[T.93]	-0.0041	0.012	0.722
5	C(county_x)[T.99]	-0.0027	0.011	0.803
6	C(county_x)[T.115]	-0.0014	0.012	0.906
7	C(county_x)[T.125]	0.0084	0.011	0.430
8	C(county_x)[T.161]	0.0260	0.011	0.020
9	C(county_x)[T.163]	0.0071	0.010	0.500
10	labor_employed	-0.0946	0.011	0.000
11	edu_none	0.0605	0.051	0.240
12	edu_no_hs	0.1581	0.021	0.000
13	edu_hs	0.0580	0.015	0.000
14	edu_graduate	0.0099	0.024	0.674
15	edu_associates	-0.0263	0.030	0.383

In [10]:

```
plt.figure(figsize=(11,12))
color = [('red' if p<0.5 else 'blue') for p in result_model_df_2['P>|t|']]
plt.errorbar(data=result_model_df_2, x='coef', y='variable',xerr='std err', ecolor=color,
             fmt='o')
plt.xlabel('Coefficient')
plt.ylabel('Variable Factors')
plt.title('Variable factor for 10-15k income')
```

Out[10]:

```
Text(0.5, 1.0, 'Variable factor for 10-15k income')
```



In [8]:

```
model_3 = smf.ols('income_50k_60k ~ labor_employed+C(county_x)+edu_none+edu_no_hs+edu_hs+edu_bachelors'
                  +'+edu_graduate+edu_associates', data=work_income).fit()
model_3.summary()
```

Out[8]:

OLS Regression Results

Dep. Variable:	income_50k_60k	**R-squared:**	0.080
Model:	OLS	**Adj. R-squared:**	0.071
Method:	Least Squares	**F-statistic:**	8.973
Date:	Thu, 20 Apr 2023	**Prob (F-statistic):**	2.55e-20
Time:	15:53:47	**Log-Likelihood:**	2652.7
No. Observations:	1566	**AIC:**	-5273.
Df Residuals:	1550	**BIC:**	-5188.
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0513	0.015	3.453	0.001	0.022	0.080
C(county_x)[T.65]	-0.0067	0.013	-0.507	0.612	-0.033	0.019
C(county_x)[T.75]	-0.0093	0.014	-0.680	0.496	-0.036	0.017
C(county_x)[T.91]	-0.0043	0.015	-0.282	0.778	-0.034	0.026
C(county_x)[T.93]	-0.0180	0.014	-1.304	0.193	-0.045	0.009
C(county_x)[T.99]	-0.0153	0.013	-1.198	0.231	-0.040	0.010
C(county_x)[T.115]	-0.0042	0.014	-0.297	0.767	-0.032	0.024
C(county_x)[T.125]	-0.0201	0.013	-1.570	0.117	-0.045	0.005
C(county_x)[T.161]	-0.0092	0.013	-0.690	0.491	-0.036	0.017
C(county_x)[T.163]	-0.0133	0.013	-1.060	0.289	-0.038	0.011
labor_employed	0.0659	0.014	4.873	0.000	0.039	0.092
edu_none	0.1123	0.051	2.181	0.029	0.011	0.213
edu_no_hs	0.0075	0.021	0.355	0.722	-0.034	0.049
edu_hs	0.0162	0.013	1.277	0.202	-0.009	0.041
edu_bachelors	-0.0150	0.019	-0.787	0.432	-0.053	0.022
edu_graduate	-0.0705	0.016	-4.348	0.000	-0.102	-0.039
edu_associates	0.0009	0.027	0.033	0.974	-0.052	0.054

Omnibus:	2004.930	**Durbin-Watson:**	1.872
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	955216.058
Skew:	6.368	**Prob(JB):**	0.00
Kurtosis:	123.321	**Cond. No.**	7.05e+15

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 5.93e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

In [12]:

```
result_model_3 = model_3.summary().tables[1].as_html()
result_model_3 = pd.read_html(result_model_3, header =0, index_col=0)[0]
result_model_df_3=result_model_3[['coef', 'std err', 'P>|t|']]
result_model_df_3.reset_index(inplace=True)
result_model_df_3.rename(columns={'index':'variable'}, inplace=True)
```

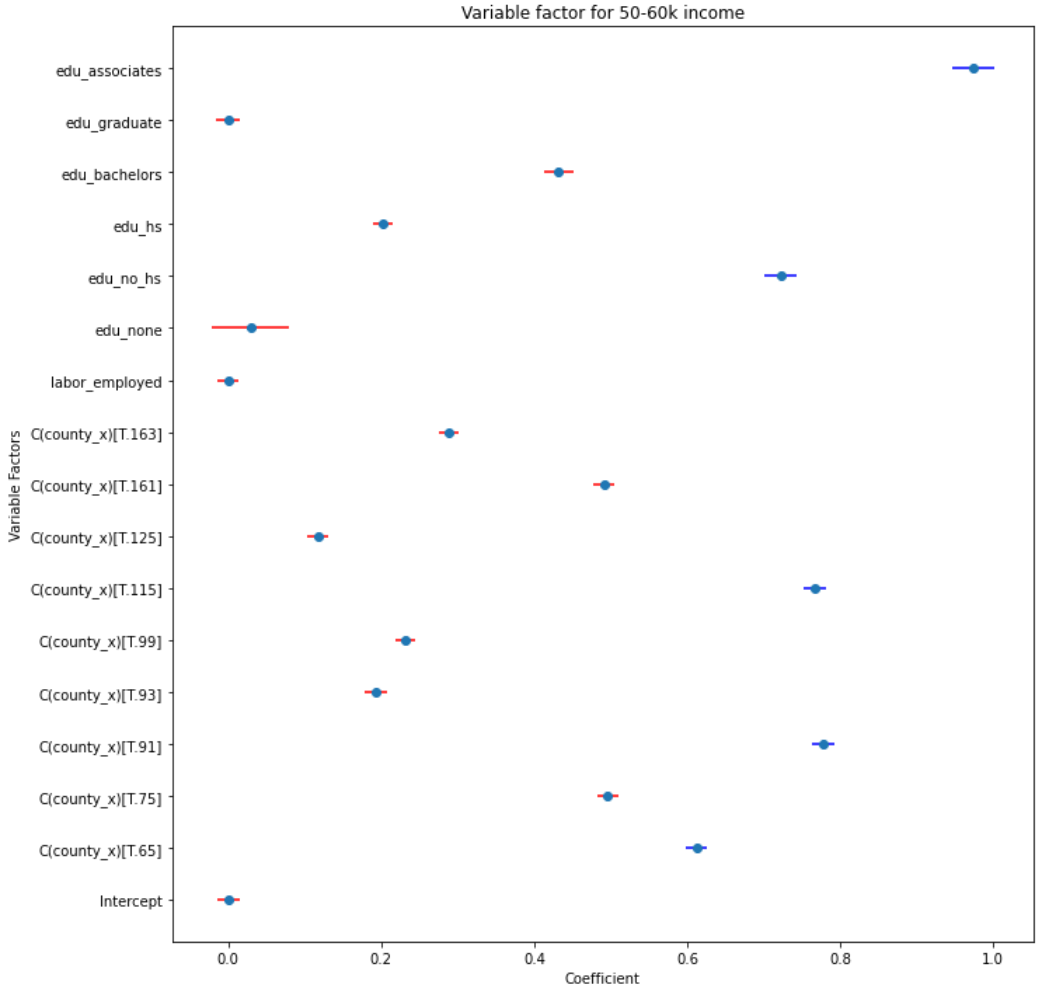
```
/var/folders/xr/gjwnmnh16yq74f56w_0pz5tr0000gn/T/ipykernel_15748/1143228312.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  result_model_df_3.rename(columns={'index':'variable'}, inplace=True)
```

In [13]:

```
plt.figure(figsize=(11,12))
color = [('red' if p<0.5 else 'blue') for p in result_model_df_3['P>|t|']]
plt.errorbar(data=result_model_df_3, x='P>|t|', y='variable',xerr='std err', ecolor=color,
             fmt='o')
plt.xlabel('Coefficient')
plt.ylabel('Variable Factors')
plt.title('Variable factor for 50-60k income')
```

Out[13]:

```
Text(0.5, 1.0, 'Variable factor for 50-60k income')
```



Q2: What is the weather condition that demolition happended the most?

In [14]:
```
michigan_weather = pd.read_csv('se-michigan-weather-data.csv', dtype={'TAVG_ATTRIBUTES': 'str'})
demolition = pd.read_csv('Detroit0DemolitionPermits.csv')
foreign_firms = pd.read_csv('Foreign_Firm_Locations.csv')
```

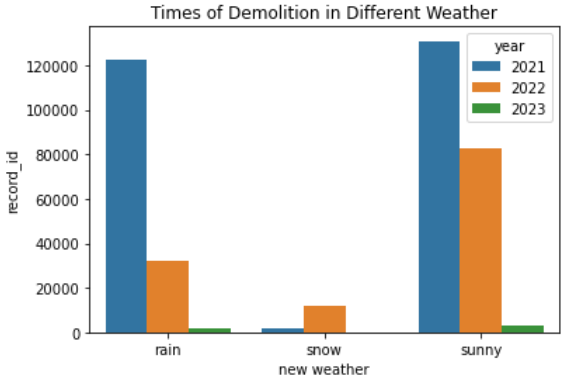
In [15]:
```
def weather(x):
    if x['SNOW'] > 0:
        return 'snow'
    elif x['SNOW'] > 0 and x['PRCP'] > 0:
        return 'mixed weather'
    elif x['PRCP'] > 0:
        return 'rain'
    return 'sunny'
michigan_weather['new weather'] = michigan_weather.apply(weather, axis=1)
```

In [16]:
```
michigan_weather['NEW_DATE'] = michigan_weather['DATE'].apply(pd.to_datetime)
demolition['new_date'] = demolition['date_status'].apply(pd.to_datetime)
demolition['Date'] = demolition['new_date'].dt.strftime("%Y-%m-%d")
demolition['Date'] = pd.to_datetime(demolition['Date'])
demolition['year'] = demolition['Date'].dt.year
```

In [17]:
```
new_df = pd.merge(michigan_weather, demolition, how='inner', left_on='NEW_DATE', right_on='Date')
```

In [19]:
```
new_df_weather = new_df.groupby(['year', 'new weather'])['record_id'].count()
new_df = new_df_weather.reset_index()
sns.barplot(data=new_df, x='new weather', y='record_id', hue='year').set(title='Times of Demolition in Different Weather')
```

Out[19]:
```
[Text(0.5, 1.0, 'Times of Demolition in Different Weather')]
```



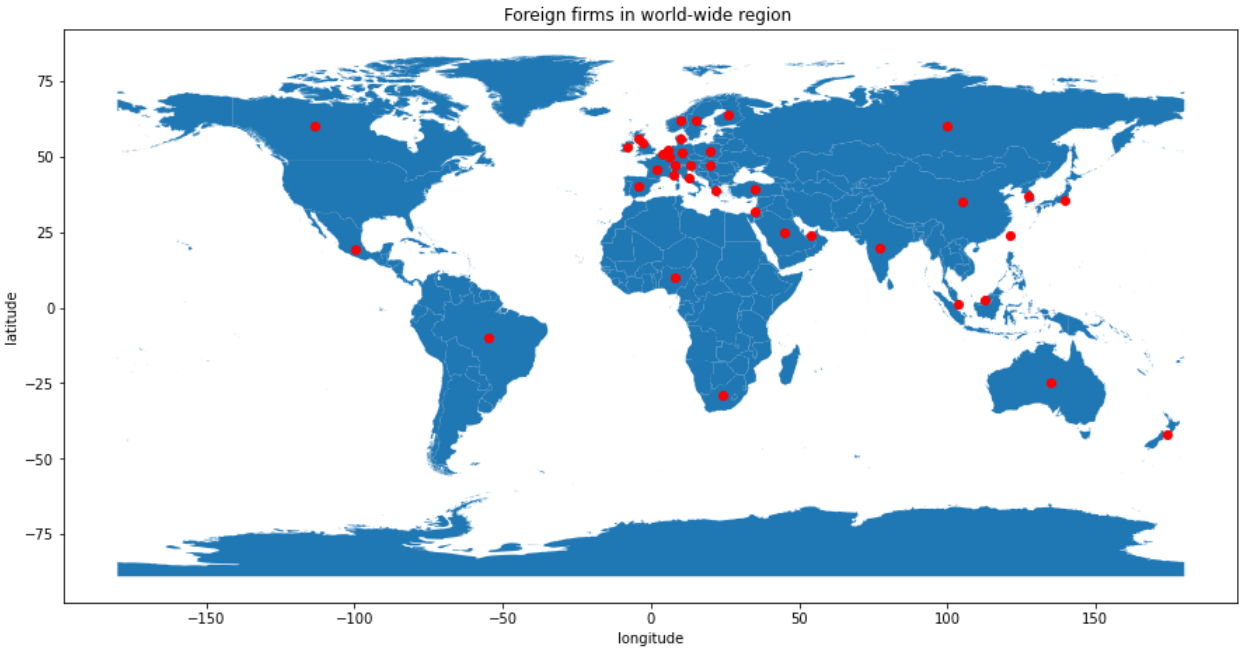
Q3:Which part of the word region has the most international foreign firms residing here in SE Michigan?

In [21]:
```
def max(s):
    x = s.sort_values(by='Foreign_Owned_Firms', ascending=False)
    return x[:1]
new = foreign_firms.groupby(['X', 'Y'])
new = new.apply(max)
```

In [22]:
```
countries_map= gpd.read_file('World_Countries_(Generalized)')
data = gpd.GeoDataFrame(new, geometry=gpd.points_from_xy(new.X, new.Y))
data.crs='EPSG:4326'
geo_pan = gpd.sjoin(countries_map, data)
```

In [23]:
```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(1, figsize=(15,15))
plt.title('Foreign firms in world-wide region ')
plt.xlabel('longitude')
plt.ylabel('latitude')
countries_map.plot(ax=ax)
data.plot(ax=ax, marker='o', color='red')
```

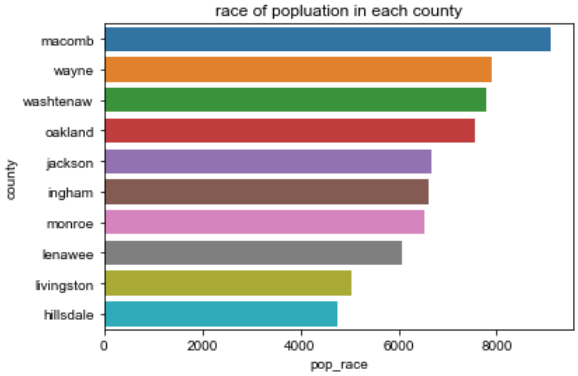
Out[23]:
```
<AxesSubplot:title={'center':'Foreign firms in world-wide region '}, xlabel='longitude', ylabel='latitude'>
```



Q4:Which county has the highest population race in SE Michigan?

In [24]:
```
demo = pd.read_csv('acs_demographics.csv')
def sort_county(county):
    a = county.lower().split()
    return a[3]
demo['county'] = demo['tract_name'].apply(sort_county)
```

In [25]:
```
group = demo.groupby('county')
new = group.apply(lambda g: g[g['pop_race'] == g['pop_race'].max()]).sort_values(ascending=False, by='pop_race')

sns.barplot(data=new, x='pop_race', y='county').set(title='race of popluation in each county')
sns.set(rc={'figure.figsize':(8,10)})
```



Q5:Did the temperature rise from the year 2012 to the year 2016 election years?

In [26]:
```
election = pd.read_csv('pres_election_2000-2020.csv')
temperature = pd.read_csv('detroit-weather-complete.csv')
```

In [28]:
```
temperature['date'] = temperature['date'].apply(pd.to_datetime)
temperature['year'] = temperature.date.dt.year
election_temperature = pd.merge(election, temperature, on='year')
```

In [29]:
```
election_temperature=election_temperature[['date','county_name', 'temperature']]
temperature = election_temperature.groupby('date').agg({'temperature':'mean'})
pd.DataFrame(temperature).reset_index()
```

Out[29]:

	date	temperature
0	2012-10-01	11.036840
1	2012-10-02	14.340558
2	2012-10-03	14.518382
3	2012-10-04	16.820351
4	2012-10-05	16.948431
...	...	...
453	2016-12-27	8.058958
454	2016-12-28	7.045372
455	2016-12-29	10.322896
456	2016-12-30	6.621532
457	2016-12-31	9.905723

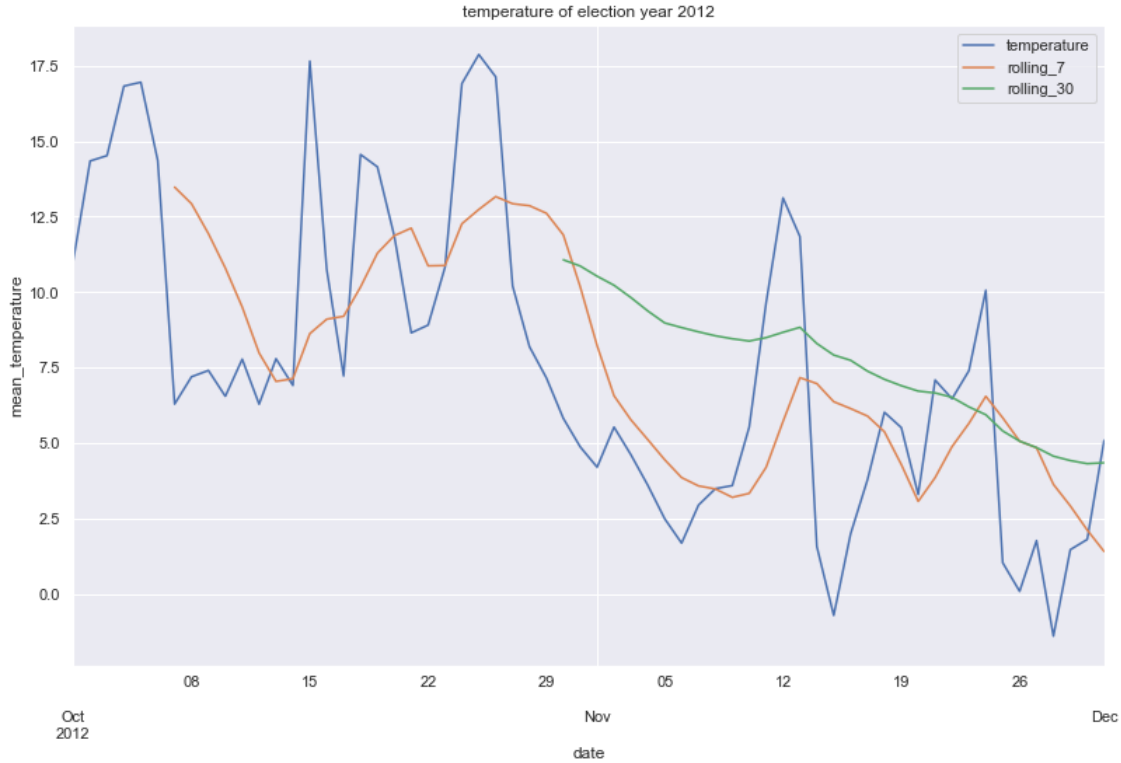
458 rows × 2 columns

In [30]:
```
data_2012 = temperature.loc['2012-09-01': '2012-12-01']
data_2012['rolling_7'] = data_2012['temperature'].rolling(7).mean()
data_2012['rolling_30'] = data_2012['temperature'].rolling(30).mean()
```

```
/var/folders/xr/gjwnmnh16yq74f56w_0pz5tr0000gn/T/ipykernel_15662/2752470007.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  data_2012['rolling_7'] = data_2012['temperature'].rolling(7).mean()
/var/folders/xr/gjwnmnh16yq74f56w_0pz5tr0000gn/T/ipykernel_15662/2752470007.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  data_2012['rolling_30'] = data_2012['temperature'].rolling(30).mean()
```

In [31]:
```
fig, ax = plt.subplots(nrows=1, figsize=(12,8), sharey=True)
fig.tight_layout(pad=4)
data_2012.plot(ax=ax, title='temperature of election year 2012')
plt.ylabel('mean_temperature')
```

Out[31]:
```
Text(79.125, 0.5, 'mean_temperature')
```

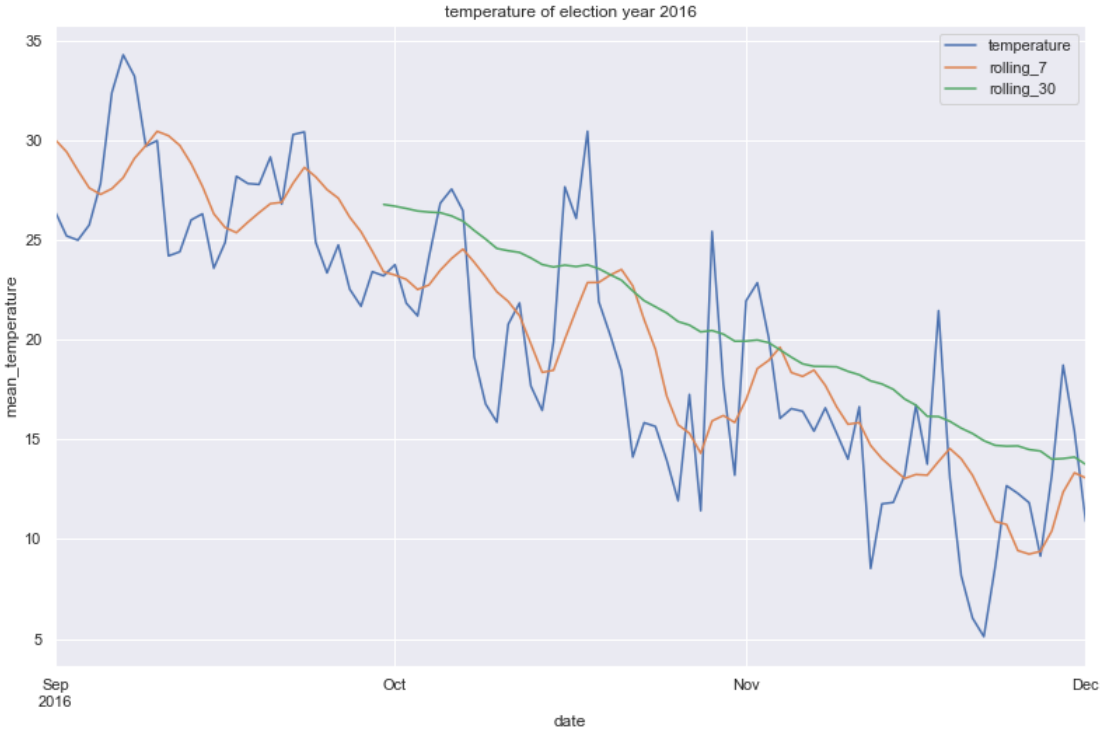


In [33]:
```
temperature_date=election_temperature.groupby('date').agg({'temperature':'mean'})
temperature_date
year_2016=temperature_date.loc['2016-09-01': '2016-12-01']
year_2016['rolling_7']=temperature_date['temperature'].rolling(window =7).mean()
year_2016['rolling_30']=year_2016['temperature'].rolling(30).mean()
```

```
/var/folders/xr/gjwnmnh16yq74f56w_0pz5tr0000gn/T/ipykernel_15662/2981820654.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  year_2016['rolling_7']=temperature_date['temperature'].rolling(window =7).mean()
/var/folders/xr/gjwnmnh16yq74f56w_0pz5tr0000gn/T/ipykernel_15662/2981820654.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
-copy
  year_2016['rolling_30']=year_2016['temperature'].rolling(30).mean()
```

In [34]:
```
fig, ax = plt.subplots(nrows=1, figsize=(12,8), sharey=True)
fig.tight_layout(pad=4)
year_2016.plot(ax=ax, title='temperature of election year 2016')
plt.ylabel('mean_temperature')
```

Out[34]:
```
Text(79.125, 0.5, 'mean_temperature')
```


temperature of election year 2016
temperature
rolling_7
rolling_30
mean_temperature
35
30
25
20
15
10
5
Sep
2016
Oct
Nov
Dec
date

In [ ]:

Detroit Crimes and Incidents

Name: Frederick Kusumo

In [1]:
```
import pandas as pd
import seaborn as sns
import datetime
import matplotlib.pyplot as plt
import geopandas as gpd
import shapely
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

In [2]:
```
rms = pd.read_csv('Detroit-RMS_Crime_Incidents.csv')
rms.head()
```

```
/var/folders/x2/71_6kf9x3yd4ksy07bczzct00000gn/T/ipykernel_51439/4034092821.py:1: DtypeWarning: Columns (8,15,
16) have mixed types. Specify dtype option on import or set low_memory=False.
  rms = pd.read_csv('Detroit-RMS_Crime_Incidents.csv')
```

Out[2]:

	X	Y	crime_id	report_number	address	offense_description	offense_category	state_offense_code	arrest_cha
0	-83.163011	42.427745	3535020	2008170095	Margareta St & Ilene St	ARSON	ARSON	2099	20
1	-83.131659	42.307950	3006180	1612260018	Lawndale St & Senator St	ARSON	ARSON	2099	20
2	-83.204470	42.439354	3018963	1701280234	Prevost St & Fargo St	LARCENY - OTHER	LARCENY	2307	23
3	-83.018830	42.355350	3019773	1701310083	Burlage Pl & Hendricks St	LARCENY - OTHER	LARCENY	2307	23
4	-82.996568	42.368237	3021816	1702060055	Goethe Ave & Crane St	ROBBERY	ROBBERY	1201	12

5 rows × 25 columns

In [3]:
```
fire = pd.read_csv('Detroit-Fire_Incidents.csv')
fire.head()
```

```
/var/folders/x2/71_6kf9x3yd4ksy07bczzct00000gn/T/ipykernel_51439/2075665903.py:1: DtypeWarning: Columns (14) h
ave mixed types. Specify dtype option on import or set low_memory=False.
  fire = pd.read_csv('Detroit-Fire_Incidents.csv')
```

Out[3]:

	exposure	incident_address	incident_number	incident_type_desc	property_use	engine_area	call_datetime	dispatch_datetime	ar
0	No	16888 RIVERVIEW ST	18-0032836	Fire, Other ...	1 or 2 family dwelling ...	E54	2018/09/06 12:13:21+00	2018/09/06 12:19:19+00	
1	No	18469 GREENVIEW AVE	19-0019239	Fire, Other ...	1 or 2 family dwelling ...	E59	2019/06/03 06:46:14+00	2019/06/03 06:48:19+00	
2	No	2930 CODY ST	20-0010724	Fire, Other ...	1 or 2 family dwelling ...	E56	2020/04/05 10:44:09+00	2020/04/05 10:46:02+00	
3	No	11076 NOTTINGHAM RD	20-0003714	Fire, Other ...	1 or 2 family dwelling ...	E58	2020/02/01 19:43:20+00	2020/02/01 19:44:40+00	
4	No	18844 HULL ST	20-0029880	Fire, Other ...	1 or 2 family dwelling ...	E44	2020/08/31 20:22:19+00	2020/08/31 20:23:30+00	

In [4]:
```
nine11 = pd.read_csv('Detroit-911_Calls_for_Service_(Last_30_Days).csv')
nine11.head()
```

Out[4]:

	X	Y	incident_id	agency	incident_address	zip_code	priority	callcode	calldescription	category	...	trav
0	-83.108512	42.373040	202236400038	DPD	Linwood St & Clairmount St	48206	3	SA	SPECIAL ATTENTION	SPCL ATT	...	
1	-83.160790	42.425981	202236400039	DPD	Pickford St & Wyoming St	48221	3	935030	REMARKS	REMARKS	...	
2	-83.066505	42.339483	202236400040	DPD	S M 10 Service Drive & Grand River Ave	48201	5	376050	INVESTIGATE AUTO	INVAUTO	...	
3	-83.088657	42.425048	202236400042	DPD	Hull St & E Nevada St	48203	3	W4807	START OF SHIFT INFORMATION	STRTSHFT	...	
4	-82.993417	42.407280	202236400043	DPD	E Outer Dr & Gunston Ave	48213	3	W8190	TOWING DETAIL	TOW	...	

5 rows × 26 columns

1. Which day of the week has the highest calls regarding fire and how does it compare with between two different data sets [Fire, 911]?

In [5]:
```
fire['calldatetime'] = pd.to_datetime(fire['call_datetime'])
fire['dayofweek'] = fire.calldatetime.dt.dayofweek
fire['month'] = fire.calldatetime.dt.month
```

In [6]:
```
nine11['calldatetime'] = pd.to_datetime(nine11['call_timestamp'])
nine11['dayofweek'] = nine11.calldatetime.dt.dayofweek
nine11['month'] = nine11.calldatetime.dt.month
nine11_fire = nine11[nine11['calldescription'].str.contains('FIRE')  &
                     (~nine11['category'].str.contains('SHOTS FIRED'))]
nine11_fire.head()
```

Out[6]:

	X	Y	incident_id	agency	incident_address	zip_code	priority	callcode	calldescription	category	...	tot
55	-83.245381	42.350157	202236400132	DPD	Tireman St & Burt Rd	48228	1	342010	SHOTS FIRED IP	SHOTS IP	...	
97	-82.997225	42.432840	202236400210	DPD	Hamburg St & Eastwood St	48205	1	342010	SHOTS FIRED IP	SHOTS IP	...	
209	-83.026325	42.381759	202236400418	DPD	Lambert St & Helen St	48211	1	342010	SHOTS FIRED IP	SHOTS IP	...	
219	-83.122808	42.384118	202236400444	DPD	Wildemere St & Richton St	48206	1	342010	SHOTS FIRED IP	SHOTS IP	...	
237	-83.124365	42.309188	202236400481	DPD	Gartner St & Evans St	48209	1	342010	SHOTS FIRED IP	SHOTS IP	...	

5 rows × 29 columns

In [7]:
```
f = fire.groupby('dayofweek')['calldatetime'].count()
f = f.reset_index()
f
```

Out[7]:

	dayofweek	calldatetime
0	0	16092
1	1	15743
2	2	16078
3	3	16158
4	4	16161
5	5	16838
6	6	17638

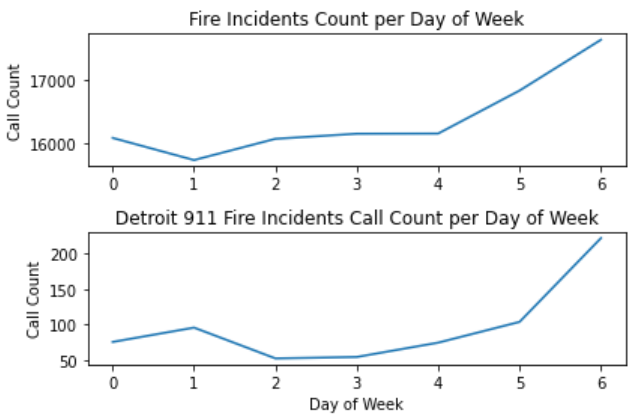
In [8]:
```
n = nine11_fire.groupby('dayofweek')['calldatetime'].count()
n = n.reset_index()
n
```

Out[8]:

	dayofweek	calldatetime
0	0	76
1	1	96
2	2	53
3	3	55
4	4	75
5	5	104
6	6	221

In [9]:

```
plt.subplot(2, 1, 1)
plt.plot(f.dayofweek, f.calldatetime)
plt.title('Fire Incidents Count per Day of Week')
plt.ylabel("Call Count")

plt.subplot(2, 1, 2)
plt.plot(n.dayofweek, n.calldatetime)
plt.title('Detroit 911 Fire Incidents Call Count per Day of Week')

plt.xlabel("Day of Week")
plt.ylabel("Call Count")
plt.tight_layout()

plt.show()
```



2. What 911 calls category is the most common per zipcode?

In [10]:

```
zip_boundary = gpd.read_file('zip_codes.shp')
zip_boundary['zipcode'] = zip_boundary['zipcode'].astype(int)
zip_boundary = zip_boundary.sort_values('zipcode')
zip_boundary['zipcode'] = zip_boundary['zipcode'].astype(str)
zip_boundary = zip_boundary.drop([27], axis=0)
zip_boundary.set_axis(range(len(zip_boundary)), inplace=True)
zip_boundary = zip_boundary.drop(columns='objectid')
zip_boundary.head()
```

Out[10]:

	zipcode	SHAPE_Leng	SHAPE_Area	geometry
0	48201	0.098797	0.000560	POLYGON ((-83.07770 42.35254, -83.07759 42.352...
1	48202	0.140343	0.000976	POLYGON ((-83.10059 42.38864, -83.07974 42.396...
2	48203	0.214615	0.002394	POLYGON ((-83.13333 42.44623, -83.08344 42.447...
3	48204	0.160062	0.001434	POLYGON ((-83.16866 42.38067, -83.16794 42.380...
4	48205	0.197386	0.001772	POLYGON ((-83.01397 42.41972, -83.00406 42.419...

In [11]:

```
zipcodes = [int(x) for x in zip_boundary['zipcode']]

detroit_zips = pd.DataFrame(zipcodes, columns=['zip_code'])
detroit_zips = detroit_zips.sort_values('zip_code')
detroit_zips.head()
```

Out[11]:

	zip_code
0	48201
1	48202
2	48203
3	48204
4	48205

In [12]:

```
detroit_911 = detroit_zips.merge(nine11, on='zip_code', how='inner')
detroit_911.head()
```

Out[12]:

	zip_code	X	Y	incident_id	agency	incident_address	priority	callcode	calldescription	category	...	total
0	48201	-83.066505	42.339483	202236400040	DPD	S M 10 Service Drive & Grand River Ave	5	376050	INVESTIGATE AUTO	INVAUTO	...	
1	48201	-83.066505	42.339483	202236400062	DPD	S M 10 Service Drive & Grand River Ave	3	935030	REMARKS	REMARKS	...	
2	48201	-83.066505	42.339483	202236400063	DPD	S M 10 Service Drive & Grand River Ave	3	935030	REMARKS	REMARKS	...	
3	48201	-83.052412	42.349718	202236400079	DPD	Mack Ave & Beaubien St	4	347040	ASSAULT AND BATTERY REPORT	ABRPT	...	2
4	48201	-83.068873	42.355196	202236400181	DPD	W Warren Ave & 2nd Ave	3	SA	SPECIAL ATTENTION	SPCL ATT	...	

5 rows × 29 columns

In [13]:

```
def top(x):
    x = x.sort_values(by='calldescription', ascending=False)
    return x[:1]

topcalldesc = detroit_911.groupby('zip_code')
topcalldesc = topcalldesc.apply(top)
topcalldesc = topcalldesc[['calldescription', 'longitude', 'latitude']]
topcalldesc = topcalldesc.reset_index()
topcalldesc = topcalldesc[['zip_code', 'calldescription', 'longitude', 'latitude']]
topcalldesc['zip_code'] = topcalldesc['zip_code'].astype(str)
topcalldesc.head()
```

Out[13]:

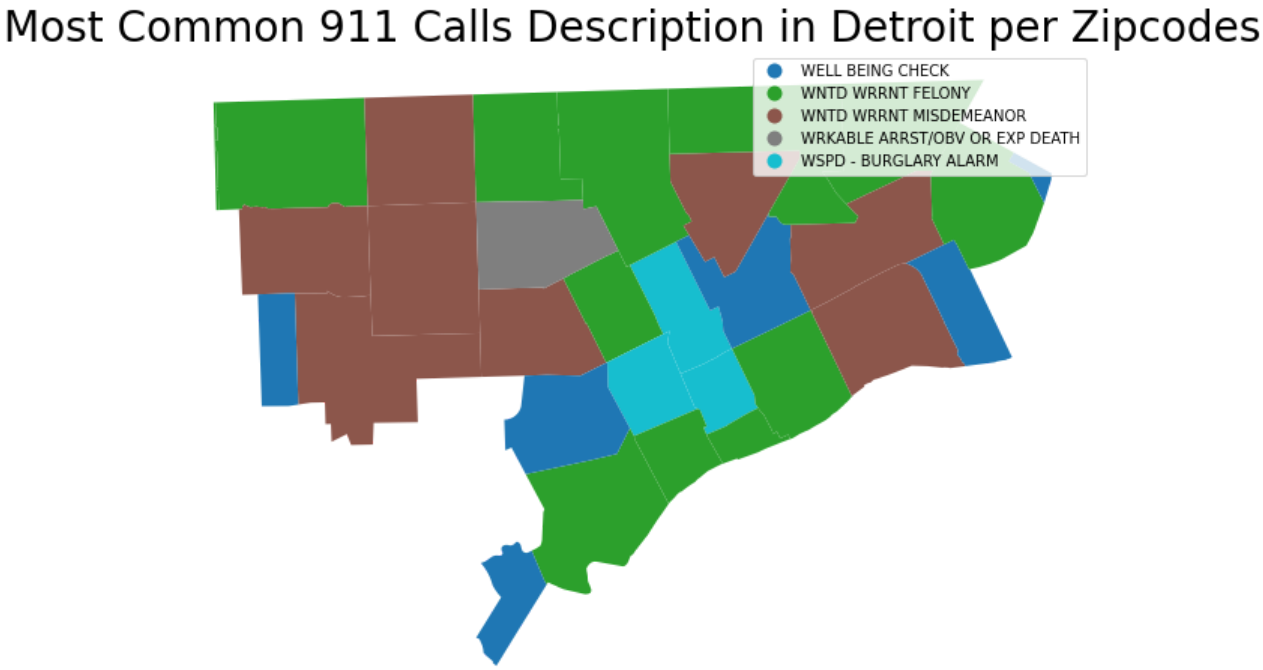
	zip_code	calldescription	longitude	latitude
0	48201	WSPD - BURGLARY ALARM	-83.062209	42.357754
1	48202	WSPD - BURGLARY ALARM	-83.071871	42.365117
2	48203	WNTD WRRNT FELONY	-83.116367	42.414367
3	48204	WNTD WRRNT MISDEMEANOR	-83.138197	42.353949
4	48205	WNTD WRRNT FELONY	-82.996198	42.438303

In [14]:

```
fig, ax = plt.subplots(1, figsize=(11, 8.5))
ax.axis('off')
ax.set_title('Most Common 911 Calls Description in Detroit per Zipcodes', fontsize=28)
zip_boundary.plot(ax=ax, column = topcalldesc.calldescription, legend=True)
# ax.legend(loc=0)
#plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left', borderaxespad=0.)
# plt.legend(loc='lower right')
#plt.rc()'legend',fontsize='small')
```

Out[14]:

```
<AxesSubplot:title={'center':'Most Common 911 Calls Description in Detroit per Zipcodes'}>
```



3. What is the crime count per zipcode in Detroit?

In [15]:
```
detroit_zips = zip_boundary['zipcode'].tolist()
detroit_zips
```

Out[15]:
```
['48201',
 '48202',
 '48203',
 '48204',
 '48205',
 '48206',
 '48207',
 '48208',
 '48209',
 '48210',
 '48211',
 '48212',
 '48213',
 '48214',
 '48215',
 '48216',
 '48217',
 '48219',
 '48221',
 '48223',
 '48224',
 '48226',
 '48227',
 '48228',
 '48234',
 '48235',
 '48236',
 '48238',
 '48239',
 '48240']
```

In [16]:
```
detroit_rms = rms.copy()
detroit_rms['zip_code'] = detroit_rms['zip_code'].astype(str)
detroit_rms = detroit_rms[detroit_rms['zip_code'].str.contains("|".join(detroit_zips))]
detroit_rms.head()
```

Out[16]:

	X	Y	crime_id	report_number	address	offense_description	offense_category	state_offense_code	arrest_cha
0	-83.163011	42.427745	3535020	2008170095	Margareta St & Ilene St	ARSON	ARSON	2099	20
1	-83.131659	42.307950	3006180	1612260018	Lawndale St & Senator St	ARSON	ARSON	2099	20
2	-83.204470	42.439354	3018963	1701280234	Prevost St & Fargo St	LARCENY - OTHER	LARCENY	2307	23
3	-83.018830	42.355350	3019773	1701310083	Burlage Pl & Hendricks St	LARCENY - OTHER	LARCENY	2307	23
4	-82.996568	42.368237	3021816	1702060055	Goethe Ave & Crane St	ROBBERY	ROBBERY	1201	12

5 rows × 25 columns

In [17]:

```
det_rms = detroit_rms.groupby('zip_code')['offense_description'].count()
det_rms = det_rms.reset_index()
det_rms.head()
```

Out[17]:

	zip_code	offense_description
0	48201	13798
1	48202	12160
2	48203	14413
3	48204	18033
4	48205	32220

In [18]:

```
d_zip = zip_boundary.copy()
d_zip = d_zip.drop([29], axis=0)
d_zip.head()
```

Out[18]:

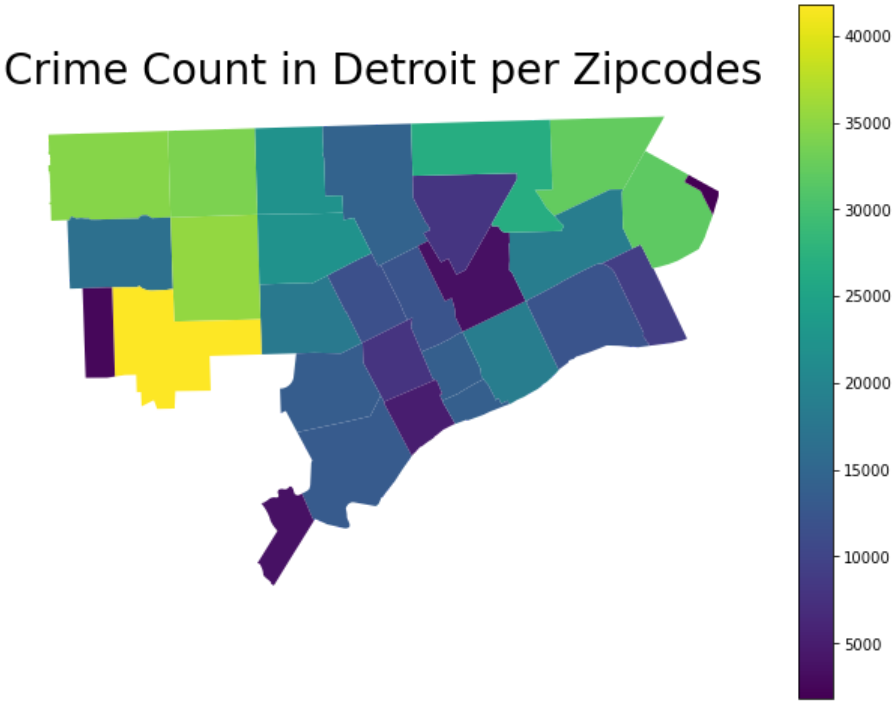
	zipcode	SHAPE_Leng	SHAPE_Area	geometry
0	48201	0.098797	0.000560	POLYGON ((-83.07770 42.35254, -83.07759 42.352...
1	48202	0.140343	0.000976	POLYGON ((-83.10059 42.38864, -83.07974 42.396...
2	48203	0.214615	0.002394	POLYGON ((-83.13333 42.44623, -83.08344 42.447...
3	48204	0.160062	0.001434	POLYGON ((-83.16866 42.38067, -83.16794 42.380...
4	48205	0.197386	0.001772	POLYGON ((-83.01397 42.41972, -83.00406 42.419...

In [19]:

```
fig, ax = plt.subplots(1, figsize=(11,8.5))
ax.axis('off')
ax.set_title('Crime Count in Detroit per Zipcodes', fontsize=28)
d_zip.plot(ax=ax, column = det_rms.offense_description, legend=True)
```

Out[19]:

```
<AxesSubplot:title={'center':'Crime Count in Detroit per Zipcodes'}>
```



4. What is the crimes per capita based on each zipcode in Detroit?

Data is from http://www.mapszipcode.com/michigan/detroit-zip-code-map/

In [20]:
```
zips = d_zip['zipcode'].tolist()
zips.remove('48203')
zips.remove('48212')
zips.remove('48236')
zips.remove('48239')
population = [11033, 16265, 28011, 41947, 22436, 20085, 10084, 34583, 31141, 6318, 28729, 22673, 14568, 5117,
              48391, 40152, 24185, 45296, 5232, 47007, 53145, 36211, 46133, 33116]
med_age = [35.9, 37.8, 39.5, 29.7, 34.6, 43.4, 36.2, 28, 27, 34, 32.9, 42.6, 34.8, 34.9, 42.7, 35.8, 39.5, 34.
           40.7, 34.6, 30.4, 35.3, 40.1, 36.8]
income_per_capita = [15969, 16395, 13078, 12887, 12986, 18955, 12611, 11438, 9919, 11061, 11248, 17945, 12679,
                     14349, 17351, 201888, 17363, 14928, 30904, 15013, 13359, 13400, 18343, 13287]
med_home_val = [166400, 72900, 49400, 56300, 59700, 62600, 61300, 51400, 41500, 34800, 45200, 54600, 45800, 53
            54300, 70300, 83500, 68700, 63200, 123200, 62100, 52100, 57200, 74200, 51300]
med_gross_rent = [501, 600, 737, 926, 694, 726, 645, 634, 762, 741, 761, 652, 794, 623, 865, 811, 840, 809, 88
                  874, 864, 831, 878, 722]
```

In [21]:
```
data = {'zipcode': zips,
        'population': population,
        'med_age': med_age,
        'income_per_capita': income_per_capita,
        'med_home_val': med_home_val,
        'med_gross_rent': med_gross_rent}

zips_info = pd.DataFrame(data)
zips_info.head()
```

Out[21]:

	zipcode	population	med_age	income_per_capita	med_home_val	med_gross_rent
0	48201	11033	35.9	15969	166400	501
1	48202	16265	37.8	16395	72900	600
2	48204	28011	39.5	13078	49400	737
3	48205	41947	29.7	12887	56300	926
4	48206	22436	34.6	12986	59700	694

In [22]:
```
zips_fullinfo = zips_info.merge(det_rms, how='left', left_on='zipcode', right_on='zip_code').drop(
    columns = ['zip_code'])
zips_fullinfo.rename({'offense_description': 'crime_count'}, axis=1, inplace=True)
zips_fullinfo.head()
```

Out[22]:

	zipcode	population	med_age	income_per_capita	med_home_val	med_gross_rent	crime_count
0	48201	11033	35.9	15969	166400	501	13798
1	48202	16265	37.8	16395	72900	600	12160
2	48204	28011	39.5	13078	49400	737	18033
3	48205	41947	29.7	12887	56300	926	32220
4	48206	22436	34.6	12986	59700	694	11643

In [23]:

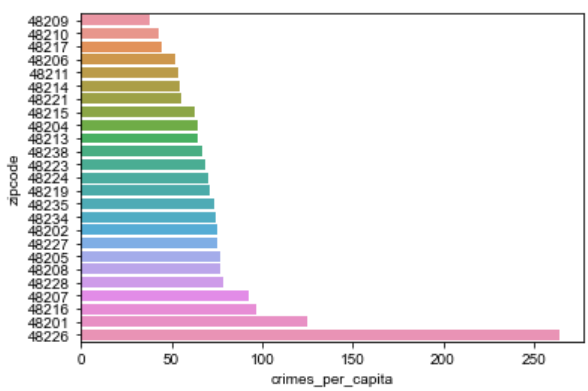
```
zips_fullinfo['crimes_per_capita'] = zips_fullinfo['crime_count'] / zips_fullinfo['population'] * 100
zips_fullinfo.dropna(inplace=True)
zips_fullinfo.head()
```

Out[23]:

	zipcode	population	med_age	income_per_capita	med_home_val	med_gross_rent	crime_count	crimes_per_capita
0	48201	11033	35.9	15969	166400	501	13798	125.061180
1	48202	16265	37.8	16395	72900	600	12160	74.761758
2	48204	28011	39.5	13078	49400	737	18033	64.378280
3	48205	41947	29.7	12887	56300	926	32220	76.811214
4	48206	22436	34.6	12986	59700	694	11643	51.894277

In [24]:

```
order = zips_fullinfo.sort_values(by=['crimes_per_capita'])
sns.barplot(data=order, x=order.crimes_per_capita, y=order.zipcode)
sns.set(rc={'figure.figsize':(40,11)})
```



In [25]:

```
det_zip = d_zip
det_zip = det_zip.drop([2], axis=0)
det_zip = det_zip.drop([11], axis=0)
det_zip = det_zip.drop([26], axis=0)
det_zip = det_zip.drop([28], axis=0)
det_zip.set_axis(range(len(det_zip)), inplace=True)
det_zip.head()
```

Out[25]:

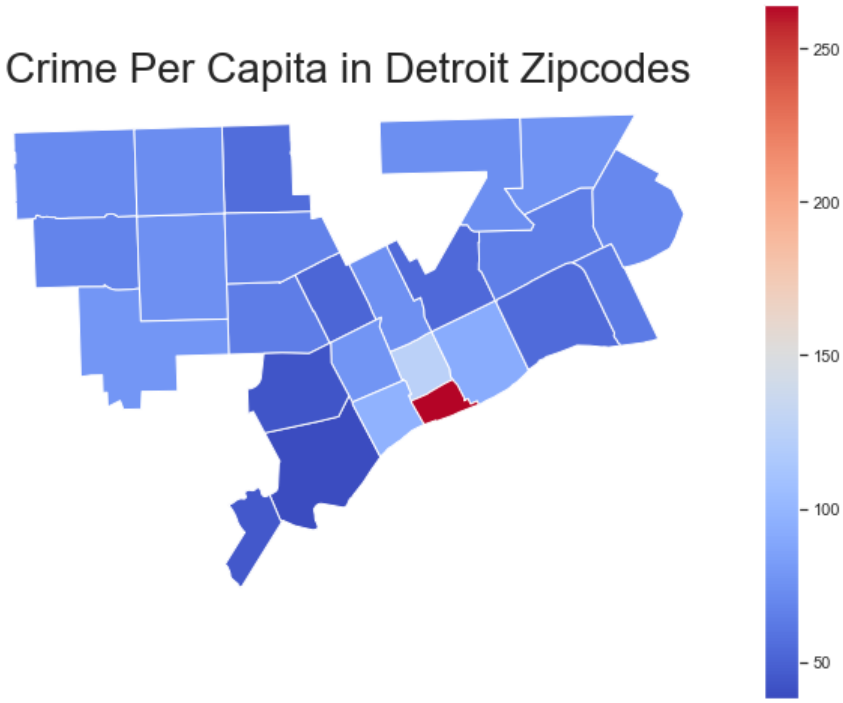
	zipcode	SHAPE_Leng	SHAPE_Area	geometry
0	48201	0.098797	0.000560	POLYGON ((-83.07770 42.35254, -83.07759 42.352...
1	48202	0.140343	0.000976	POLYGON ((-83.10059 42.38864, -83.07974 42.396...
2	48204	0.160062	0.001434	POLYGON ((-83.16866 42.38067, -83.16794 42.380...
3	48205	0.197386	0.001772	POLYGON ((-83.01397 42.41972, -83.00406 42.419...
4	48206	0.123802	0.000914	POLYGON ((-83.13053 42.38420, -83.13024 42.384...

In [26]:

```
fig, ax = plt.subplots(1, figsize=(11,8.5))
ax.axis('off')
ax.set_title('Crime Per Capita in Detroit Zipcodes', fontsize=28)
det_zip.plot(ax=ax, column = zips_fullinfo['crimes_per_capita'], legend=True, cmap='coolwarm')
```

Out[26]:

```
<AxesSubplot:title={'center':'Crime Per Capita in Detroit Zipcodes'}>
```



5. How does Crime Count affect the Income per capita, Median Home Value, Median Gross Rent, Population, and Median Age of each zipcode in Detroit?

In [27]:
```
model = smf.ols('crime_count ~ income_per_capita + med_home_val + med_gross_rent + population + med_age',
                data=zips_fullinfo).fit()
model.summary()
```

Out[27]:

OLS Regression Results

Dep. Variable:	crime_count	**R-squared:**	0.935
Model:	OLS	**Adj. R-squared:**	0.918
Method:	Least Squares	**F-statistic:**	54.75
Date:	Thu, 20 Apr 2023	**Prob (F-statistic):**	1.26e-10
Time:	15:28:55	**Log-Likelihood:**	-233.17
No. Observations:	25	**AIC:**	478.3
Df Residuals:	19	**BIC:**	485.6
Df Model:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-2.856e+04	7584.617	-3.765	0.001	-4.44e+04	-1.27e+04
income_per_capita	-0.0537	0.019	-2.899	0.009	-0.092	-0.015
med_home_val	0.1079	0.026	4.200	0.000	0.054	0.162
med_gross_rent	19.6163	7.359	2.666	0.015	4.214	35.018
population	0.6675	0.055	12.034	0.000	0.551	0.784
med_age	239.7615	168.199	1.425	0.170	-112.282	591.806

Omnibus:	0.314	**Durbin-Watson:**	2.211
Prob(Omnibus):	0.855	**Jarque-Bera (JB):**	0.139
Skew:	-0.171	**Prob(JB):**	0.933
Kurtosis:	2.868	**Cond. No.**	9.61e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.61e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [28]:

```
linear_model_df = (model.summary2().tables[1])
linear_model_df.reset_index(inplace=True)
linear_model_df.rename(columns={'index':'Variable'}, inplace=True)
linear_model_df.drop([0], axis=0, inplace=True)
linear_model_df
```

Out[28]:

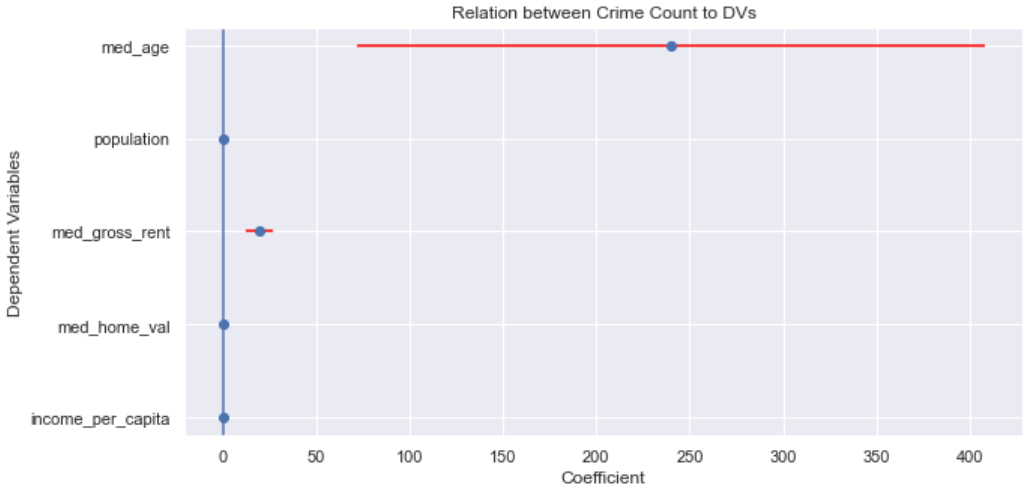
	Variable	**Coef.**	**Std.Err.**	**t**	**P>|t|**	**[0.025**	**0.975]**
1	income_per_capita	-0.053655	0.018506	-2.899268	9.193456e-03	-0.092390	-0.014921
2	med_home_val	0.107911	0.025695	4.199745	4.855986e-04	0.054132	0.161691
3	med_gross_rent	19.616277	7.358627	2.665752	1.527527e-02	4.214493	35.018061
4	population	0.667454	0.055464	12.033913	2.470263e-10	0.551365	0.783542
5	med_age	239.761533	168.198747	1.425466	1.702449e-01	-112.282490	591.805557

In [30]:

```
fig, ax = plt.subplots(1, figsize=(10,5))
ax.errorbar(data = linear_model_df, x='Coef.', y='Variable', xerr='Std.Err.', fmt='o',
            ecolor='red')
ax.axvline(0)
ax.set_title('Relation between Crime Count to DVs')
ax.set_xlabel('Coefficient')
ax.set_ylabel('Dependent Variables')
#if ahead really affect high val, but has a lot of errors
#zero how big coef, i close to zero not much affect
#0 but low error, no0t having much affect
#med_age huge affect in predictions i
```

Out[30]:

```
Text(0, 0.5, 'Dependent Variables')
```


Relation between Crime Count to DVs
med_age
population
med_gross_rent
med_home_val
income_per_capita
Dependent Variables
0
50
100
150
200
250
300
350
400
Coefficient