Auto thefts have been an alarming and on-going concern in the City of Toronto. We aim to analyze data to identify any correlations that these cases may have with break-and-enter cases and various neighbourhood statistics. We have conducted an analysis using three datasets on auto theft cases, break-and-enter cases, and neighbourhood statistics. Through our analysis, we were able to generate a dashboard to provide insights on the correlations related to the number of auto theft cases, as well as predictive statistics. The analysis was divided into five major components as per below.

1. *Technique-Significant Merging/Joining of Data*

Public historical data was used for this analysis report from two sources. We retrieved Auto Theft Open Data and Break & Enter Open Data through the Toronto Police Service and the City of Toronto Open Data portal. The Neighbourhood Profiles data was taken from a number of Census tables released by Statistics Canada from the City of Toronto Open Data portal. These datasets were then merged and refined to begin our analysis.

2. *Feature Engineering*

To gain a full grasp of car thefts and discover information that could help citizens protect their cars from being stolen, we transformed the raw data by creating new data points to tell a better story of the business problem. This included revamping and creating new data points. For instance, we further categorized the day of week column into weekday and weekend to analyze trends depending on the time of the week. Furthermore, we had a question: When is it most dangerous for residents to leave their cars parked outside? To answer this, we created a column that indicates the seasons in which the theft was recorded or carried out.

3. *Predictive Modelling*

Our objective was to predict the number of crimes based on the reported data for the period of 2019 to 2023 for neighborhoods in Toronto. We built a predictive model using variables such as: year, total age groups of the population, married common law rate, education rate, employment rate, average age, average total income in 2020. We refined the model by eliminating variables with no predictive power and developed a final model with influential variables to help us with crime prediction. An assortment of graphical tests was performed on the final model to validate the quality of the data. Our analysis shows that the total number of crime cases in 2024 is predicted to be 22,120, the auto theft cases for 2024 are anticipated to be 13,114, and the break-and-enter cases for 2024 are expected to be 6,162. There are limitations when it comes to predicting values such as the number of crime cases.

4. *Visualizing data for insight generation*

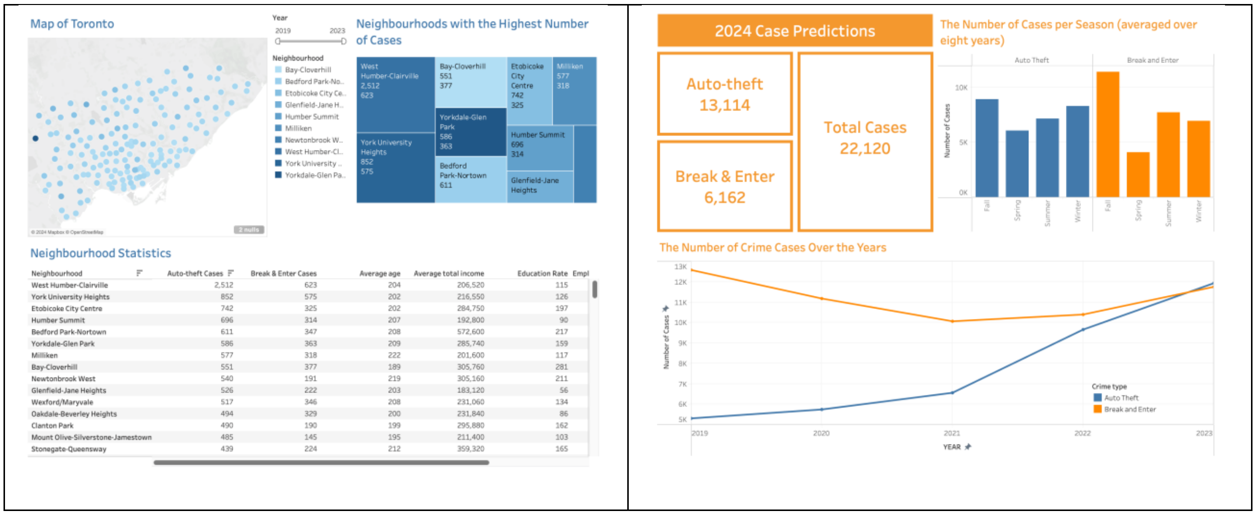
Based on our analysis, we have finalized 5 data visuals for insight generation. First, a line graph which displays the trend of auto theft crime as well as break & enter crimes over the last five years that is from 2019 to 2023. This allows us to identify that while the auto theft cases have been increasing over the years, break-and-enter cases show no trend. We have also included a bar graph which displays the average number of auto theft crimes and the average number of break-and-enter crimes over the period of eight years (2008 to 2016) per season. Through this visual, we can see which seasons on average have the most auto theft and break-and-enter cases for each of the seasons. Fall has the highest number of cases for both, and spring has the least. A map-tree visual was used to display the top 10 neighborhoods with the highest number of auto theft and break-and-enter cases. Using a map-tree makes it easy to visualize as the larger, darker squares represent the neighborhoods with the highest crime cases, and the neighbourhoods with fewer cases are represented by smaller, lighter blue squares. West Humber-Clairville is seen to have the highest number of total cases amongst all the neighbourhoods. Furthermore, we have generated a refined table which lists out the number of auto theft cases and break-and-enter cases per neighborhood. It also displays the average age of the population, average total income, average education rate, average employment rate and average married/common law rate per neighbourhood allowing the users to filter the table as they wish for insight generation. Lastly, we have created a map of Toronto depicting the number of cases in each neighbourhood allowing users to visualize what areas they should be more proactive in. The colour of the points on the map are shown darker as the number of cases are higher. Hovering over a point, displays the neighborhood name, latitude, longitude, number of auto theft cases, number of break-and-enter cases.

5. *Executive Dashboard*

Utilizing the visuals we have created and our analysis thus far, we were able to create an interactive dashboard to allow the users to gain insights on the current issue of auto theft cases in Toronto and how they may correlate with break-and-enter cases. We have displayed our findings into different components: analysis by neighbourhood and analysis by time. This will allow the audience to obtain an idea of during what and areas the cases are the highest making them more alert and proactive. The neighbourhood dashboard displays the map of Toronto and the tree-map with the top 10 neighbourhoods with the most cases. A filter was built into the dashboard to allow the users to filter the visualizations for a specific year(s). Below the map and tree-map, we have included the neighbourhood statistics table. The users can filter the table for any column of data they may desire. For instance, they can arrange the table in order of which neighbourhood has the highest employment rate. The time dashboard allows the audience to visualize how the number of cases correlate with time. We have included the line graph depicting the number of cases over the years, and the bar graph illustrating the number of average cases by season. Additionally, we have included our calculated values for the predicted number of auto theft cases, break-and-enter cases, and total cases for the year 2024.

Overall, our model can identify the auto theft trends in Toronto and provide us with detailed insight on the regions and time of crime. However, like numerous effective models, our prediction values have limitations which were identified in our analysis when looking at heteroskedasticity, normality, outliers, and low R-square. This may be a result of an underspecified model. Possible improvements to the model include increasing the number of relevant variables to the prediction, which we can implement when provided with more time and resources. The neighbourhood dashboard shows that West Humber-Clairville has the highest number of auto theft cases and total crime cases. However, it does not have the highest number of break-and-enter cases. Through the table, we can identify that there is not a strong correlation between auto theft and break-and-enter cases based on neighbourhoods. This indicates both two crime instances are independent from one another. This is further seen in the time dashboard through the line graph and bar graph.

Please find our final dashboard images below and refer to our technical report for details on our techniques.



Neighbourhood Statistics

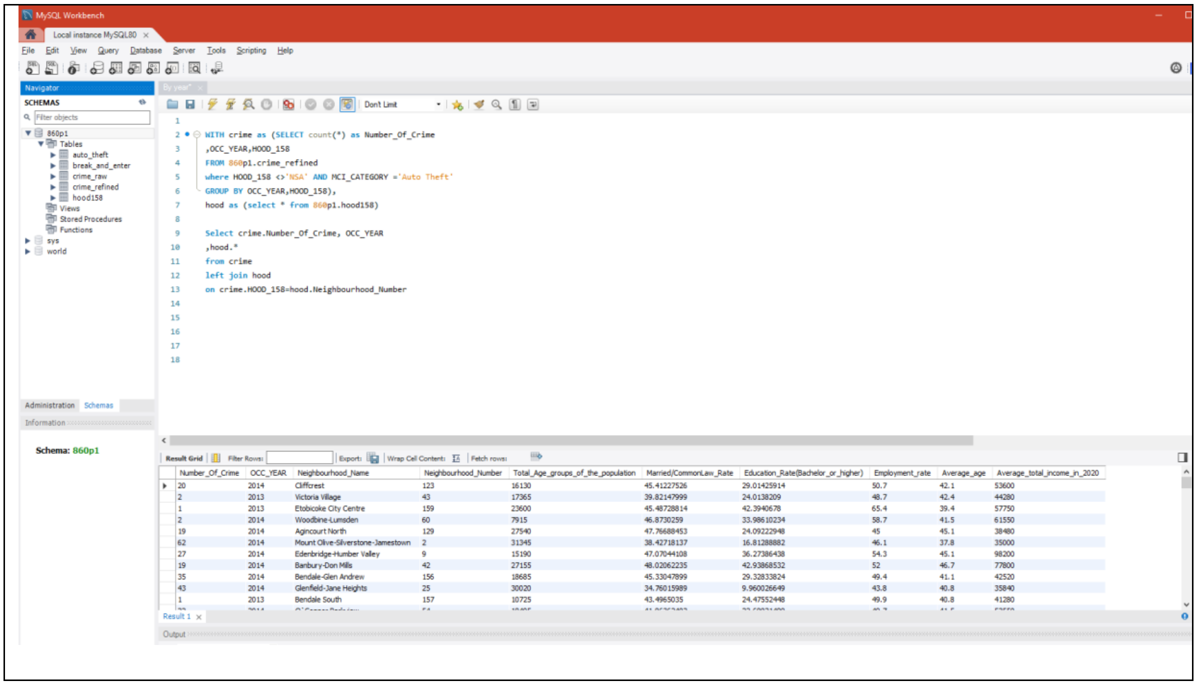
Neighbourhood	Auto-theft Cases	Break & Enter Cases	Average age	Average total income	Education Rate	Empl
West Humber-Clairville	2,512	623	204	206,520	115	
York University Heights	852	575	202	216,550	126	
Etobicoke City Centre	742	325	202	284,750	197	
Humber Summit	696	314	207	192,800	90	
Bedford Park-Nortown	611	347	208	572,600	217	
Yorkdale-Glen Park	586	363	209	285,740	159	
Milliken	577	318	222	201,600	117	
Bay-Cloverhill	551	377	189	305,760	281	
Newtonbrook West	540	191	219	305,160	211	
Glenfield-Jane Heights	526	222	203	183,120	56	
Wexford/Maryvale	517	346	208	231,060	134	
Oakdale-Beverley Heights	494	329	200	231,840	86	
Clanton Park	490	190	199	295,880	162	
Mount Olive-Silverstone-Jamestown	485	145	195	211,400	103	
Stonegate-Queensway	439	224	212	359,320	165	

Technical/Analytical Report

1. Technique-Significant Merging/Joining of Data

Raw Data Processing (prerequisites)

- A MYSQL data warehouse was built to for dataset storage.
- All dataset was extracted from above portals as three individual CSV files, then loaded into the MYSQL data warehouse using Table Data Import Wizard.
- Two crime Datasets were cleaned by removing non-GTA neighbourhoods, unnecessary columns, and then combined (UNION) with each other as crime table.
- The crime table was joined with the neighbourhood profile using the *neighbourhood ID* Key and grouped by for later visual and ML requirement.
- Refined datasets were extracted for later analysis.



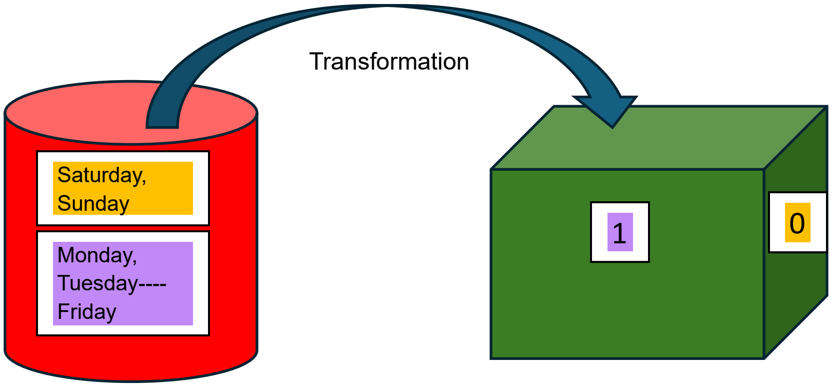
```
WITH crime as (SELECT count(*) as Number_Of_Crime
,OCC_YEAR,HOOD_158
FROM 860p1.crime_refined
where HOOD_158 <>'NSA' AND MCI_CATEGORY ='Auto Theft'
GROUP BY OCC_YEAR,HOOD_158),
hood as (select * from 860p1.hood158)

Select crime.Number_Of_Crime, OCC_YEAR
,hood.*
from crime
left join hood
on crime.HOOD_158=hood.Neighbourhood_Number
```

Number_Of_Crime	OCC_YEAR	Neighbourhood_Name	Neighbourhood_Number	Total_Age_groups_of_the_population	Married/CommonLaw_Rate	Education_Rate(Bachelor_or_higher)	Employment_rate	Average_age	Average_total_income_in_2020
20	2014	Cliffcrest	123	16130	45.41227526	29.01425914	50.7	42.1	53600
2	2013	Victoria Village	43	17365	39.82147999	24.0138209	48.7	42.4	44280
1	2013	Etobicoke City Centre	159	23600	45.48728814	42.3940678	65.4	39.4	57750
2	2014	Woodbine-Lumsden	60	7915	46.8730259	33.98610234	58.7	41.5	61550
19	2014	Agincourt North	129	27540	47.76688453	24.09222948	45	45.1	38480
62	2014	Mount Olive-Silverstone-Jamestown	2	31345	38.42718137	16.81288882	46.1	37.8	35000
27	2014	Edenbridge-Humber Valley	9	15190	47.07044108	36.27386438	54.3	45.1	98200
19	2014	Banbury-Don Mills	42	27155	48.02062235	42.93868532	52	46.7	77800
35	2014	Bendale-Glen Andrew	156	18685	45.33047899	29.32833824	49.4	41.1	42520
43	2014	Glenfield-Jane Heights	25	30020	34.76015989	9.960026649	43.8	40.8	35840
1	2013	Bendale South	157	10725	43.4965035	24.47552448	49.9	40.8	41280

2. Feature Engineering

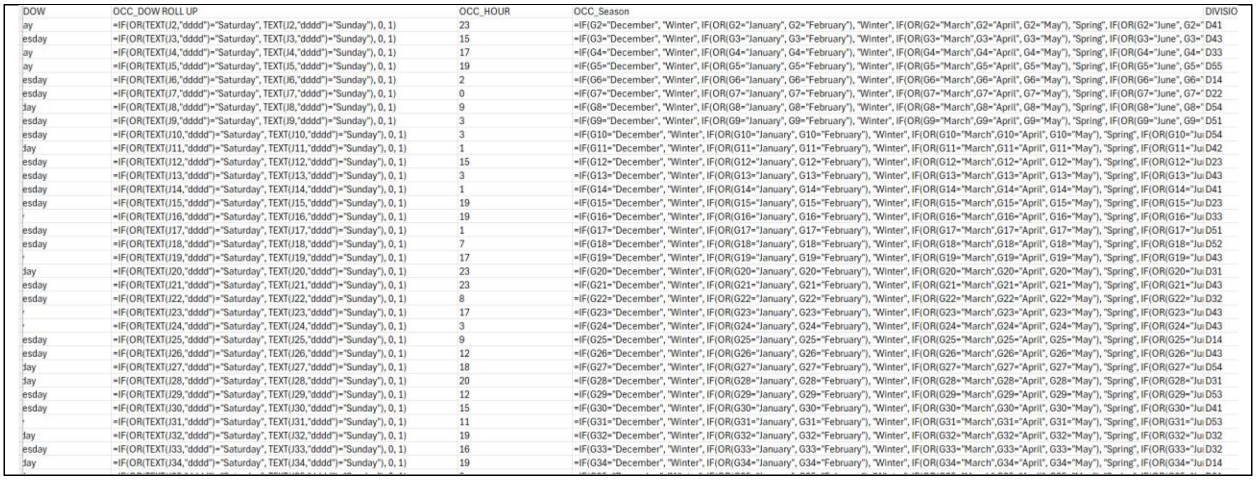
Revamping Categorical Variables: To ensure that we can apply our data to models for analysis, we converted the day of week column to give us a 1 when it is a weekday and 0 when it is a weekend to accurately evaluate how the trends change depending on the time of the week.



We also implemented the same process to derive the “Report Delay” data piece to help us understand what areas experienced the delays in reporting, which may suggest that the theft of a car has been discovered late; for example, stealing from a home may be discovered later and reported later than stealing from a shopping complex which is a temporary location.

Creating New Data Points:

We already have granular levels of data at this point, but we wanted to know if the story would change if we inculcated a larger grouping. So, by applying some excel formulas (depicted below), we created a column that indicates the seasons in which the theft was recorded or carried out.



DOW	OCC_DOW ROLL UP	OCC_HOUR	OCC_Season	DIVISIO
ay	=IF(OR(TEXT(J2,"dddd")="Saturday", TEXT(J2,"dddd")="Sunday"), 0, 1)	23	=IF(G2="December", "Winter", IF(OR(G2="January", G2="February"), "Winter", IF(OR(G2="March",G2="April", G2="May"), "Spring", IF(OR(G2="June", G2="	D41
esday	=IF(OR(TEXT(J3,"dddd")="Saturday", TEXT(J3,"dddd")="Sunday"), 0, 1)	15	=IF(G3="December", "Winter", IF(OR(G3="January", G3="February"), "Winter", IF(OR(G3="March",G3="April", G3="May"), "Spring", IF(OR(G3="June", G3="	D43
ay	=IF(OR(TEXT(J4,"dddd")="Saturday", TEXT(J4,"dddd")="Sunday"), 0, 1)	17	=IF(G4="December", "Winter", IF(OR(G4="January", G4="February"), "Winter", IF(OR(G4="March",G4="April", G4="May"), "Spring", IF(OR(G4="June", G4="	D33
ay	=IF(OR(TEXT(J5,"dddd")="Saturday", TEXT(J5,"dddd")="Sunday"), 0, 1)	19	=IF(G5="December", "Winter", IF(OR(G5="January", G5="February"), "Winter", IF(OR(G5="March",G5="April", G5="May"), "Spring", IF(OR(G5="June", G5="	D55
esday	=IF(OR(TEXT(J6,"dddd")="Saturday", TEXT(J6,"dddd")="Sunday"), 0, 1)	2	=IF(G6="December", "Winter", IF(OR(G6="January", G6="February"), "Winter", IF(OR(G6="March",G6="April", G6="May"), "Spring", IF(OR(G6="June", G6="	D14
esday	=IF(OR(TEXT(J7,"dddd")="Saturday", TEXT(J7,"dddd")="Sunday"), 0, 1)	0	=IF(G7="December", "Winter", IF(OR(G7="January", G7="February"), "Winter", IF(OR(G7="March",G7="April", G7="May"), "Spring", IF(OR(G7="June", G7="	D22
day	=IF(OR(TEXT(J8,"dddd")="Saturday", TEXT(J8,"dddd")="Sunday"), 0, 1)	9	=IF(G8="December", "Winter", IF(OR(G8="January", G8="February"), "Winter", IF(OR(G8="March",G8="April", G8="May"), "Spring", IF(OR(G8="June", G8="	D54
esday	=IF(OR(TEXT(J9,"dddd")="Saturday", TEXT(J9,"dddd")="Sunday"), 0, 1)	3	=IF(G9="December", "Winter", IF(OR(G9="January", G9="February"), "Winter", IF(OR(G9="March",G9="April", G9="May"), "Spring", IF(OR(G9="June", G9="	D51
esday	=IF(OR(TEXT(J10,"dddd")="Saturday", TEXT(J10,"dddd")="Sunday"), 0, 1)	3	=IF(G10="December", "Winter", IF(OR(G10="January", G10="February"), "Winter", IF(OR(G10="March",G10="April", G10="May"), "Spring", IF(OR(G10="Ju	D54
day	=IF(OR(TEXT(J11,"dddd")="Saturday", TEXT(J11,"dddd")="Sunday"), 0, 1)	1	=IF(G11="December", "Winter", IF(OR(G11="January", G11="February"), "Winter", IF(OR(G11="March",G11="April", G11="May"), "Spring", IF(OR(G11="Ju	D42
esday	=IF(OR(TEXT(J12,"dddd")="Saturday", TEXT(J12,"dddd")="Sunday"), 0, 1)	15	=IF(G12="December", "Winter", IF(OR(G12="January", G12="February"), "Winter", IF(OR(G12="March",G12="April", G12="May"), "Spring", IF(OR(G12="Ju	D23
esday	=IF(OR(TEXT(J13,"dddd")="Saturday", TEXT(J13,"dddd")="Sunday"), 0, 1)	3	=IF(G13="December", "Winter", IF(OR(G13="January", G13="February"), "Winter", IF(OR(G13="March",G13="April", G13="May"), "Spring", IF(OR(G13="Ju	D43
esday	=IF(OR(TEXT(J14,"dddd")="Saturday", TEXT(J14,"dddd")="Sunday"), 0, 1)	1	=IF(G14="December", "Winter", IF(OR(G14="January", G14="February"), "Winter", IF(OR(G14="March",G14="April", G14="May"), "Spring", IF(OR(G14="Ju	D41
esday	=IF(OR(TEXT(J15,"dddd")="Saturday", TEXT(J15,"dddd")="Sunday"), 0, 1)	19	=IF(G15="December", "Winter", IF(OR(G15="January", G15="February"), "Winter", IF(OR(G15="March",G15="April", G15="May"), "Spring", IF(OR(G15="Ju	D23
	=IF(OR(TEXT(J16,"dddd")="Saturday", TEXT(J16,"dddd")="Sunday"), 0, 1)	19	=IF(G16="December", "Winter", IF(OR(G16="January", G16="February"), "Winter", IF(OR(G16="March",G16="April", G16="May"), "Spring", IF(OR(G16="Ju	D33
esday	=IF(OR(TEXT(J17,"dddd")="Saturday", TEXT(J17,"dddd")="Sunday"), 0, 1)	1	=IF(G17="December", "Winter", IF(OR(G17="January", G17="February"), "Winter", IF(OR(G17="March",G17="April", G17="May"), "Spring", IF(OR(G17="Ju	D51
esday	=IF(OR(TEXT(J18,"dddd")="Saturday", TEXT(J18,"dddd")="Sunday"), 0, 1)	7	=IF(G18="December", "Winter", IF(OR(G18="January", G18="February"), "Winter", IF(OR(G18="March",G18="April", G18="May"), "Spring", IF(OR(G18="Ju	D52
	=IF(OR(TEXT(J19,"dddd")="Saturday", TEXT(J19,"dddd")="Sunday"), 0, 1)	17	=IF(G19="December", "Winter", IF(OR(G19="January", G19="February"), "Winter", IF(OR(G19="March",G19="April", G19="May"), "Spring", IF(OR(G19="Ju	D43
day	=IF(OR(TEXT(J20,"dddd")="Saturday", TEXT(J20,"dddd")="Sunday"), 0, 1)	23	=IF(G20="December", "Winter", IF(OR(G20="January", G20="February"), "Winter", IF(OR(G20="March",G20="April", G20="May"), "Spring", IF(OR(G20="Ju	D31
esday	=IF(OR(TEXT(J21,"dddd")="Saturday", TEXT(J21,"dddd")="Sunday"), 0, 1)	23	=IF(G21="December", "Winter", IF(OR(G21="January", G21="February"), "Winter", IF(OR(G21="March",G21="April", G21="May"), "Spring", IF(OR(G21="Ju	D43
esday	=IF(OR(TEXT(J22,"dddd")="Saturday", TEXT(J22,"dddd")="Sunday"), 0, 1)	8	=IF(G22="December", "Winter", IF(OR(G22="January", G22="February"), "Winter", IF(OR(G22="March",G22="April", G22="May"), "Spring", IF(OR(G22="Ju	D32
	=IF(OR(TEXT(J23,"dddd")="Saturday", TEXT(J23,"dddd")="Sunday"), 0, 1)	17	=IF(G23="December", "Winter", IF(OR(G23="January", G23="February"), "Winter", IF(OR(G23="March",G23="April", G23="May"), "Spring", IF(OR(G23="Ju	D43
	=IF(OR(TEXT(J24,"dddd")="Saturday", TEXT(J24,"dddd")="Sunday"), 0, 1)	3	=IF(G24="December", "Winter", IF(OR(G24="January", G24="February"), "Winter", IF(OR(G24="March",G24="April", G24="May"), "Spring", IF(OR(G24="Ju	D43
esday	=IF(OR(TEXT(J25,"dddd")="Saturday", TEXT(J25,"dddd")="Sunday"), 0, 1)	9	=IF(G25="December", "Winter", IF(OR(G25="January", G25="February"), "Winter", IF(OR(G25="March",G25="April", G25="May"), "Spring", IF(OR(G25="Ju	D14
esday	=IF(OR(TEXT(J26,"dddd")="Saturday", TEXT(J26,"dddd")="Sunday"), 0, 1)	12	=IF(G26="December", "Winter", IF(OR(G26="January", G26="February"), "Winter", IF(OR(G26="March",G26="April", G26="May"), "Spring", IF(OR(G26="Ju	D43
day	=IF(OR(TEXT(J27,"dddd")="Saturday", TEXT(J27,"dddd")="Sunday"), 0, 1)	18	=IF(G27="December", "Winter", IF(OR(G27="January", G27="February"), "Winter", IF(OR(G27="March",G27="April", G27="May"), "Spring", IF(OR(G27="Ju	D54
day	=IF(OR(TEXT(J28,"dddd")="Saturday", TEXT(J28,"dddd")="Sunday"), 0, 1)	20	=IF(G28="December", "Winter", IF(OR(G28="January", G28="February"), "Winter", IF(OR(G28="March",G28="April", G28="May"), "Spring", IF(OR(G28="Ju	D31
esday	=IF(OR(TEXT(J29,"dddd")="Saturday", TEXT(J29,"dddd")="Sunday"), 0, 1)	12	=IF(G29="December", "Winter", IF(OR(G29="January", G29="February"), "Winter", IF(OR(G29="March",G29="April", G29="May"), "Spring", IF(OR(G29="Ju	D53
esday	=IF(OR(TEXT(J30,"dddd")="Saturday", TEXT(J30,"dddd")="Sunday"), 0, 1)	15	=IF(G30="December", "Winter", IF(OR(G30="January", G30="February"), "Winter", IF(OR(G30="March",G30="April", G30="May"), "Spring", IF(OR(G30="Ju	D41
	=IF(OR(TEXT(J31,"dddd")="Saturday", TEXT(J31,"dddd")="Sunday"), 0, 1)	11	=IF(G31="December", "Winter", IF(OR(G31="January", G31="February"), "Winter", IF(OR(G31="March",G31="April", G31="May"), "Spring", IF(OR(G31="Ju	D53
day	=IF(OR(TEXT(J32,"dddd")="Saturday", TEXT(J32,"dddd")="Sunday"), 0, 1)	19	=IF(G32="December", "Winter", IF(OR(G32="January", G32="February"), "Winter", IF(OR(G32="March",G32="April", G32="May"), "Spring", IF(OR(G32="Ju	D32
esday	=IF(OR(TEXT(J33,"dddd")="Saturday", TEXT(J33,"dddd")="Sunday"), 0, 1)	16	=IF(G33="December", "Winter", IF(OR(G33="January", G33="February"), "Winter", IF(OR(G33="March",G33="April", G33="May"), "Spring", IF(OR(G33="Ju	D32
day	=IF(OR(TEXT(J34,"dddd")="Saturday", TEXT(J34,"dddd")="Sunday"), 0, 1)	19	=IF(G34="December", "Winter", IF(OR(G34="January", G34="February"), "Winter", IF(OR(G34="March",G34="April", G34="May"), "Spring", IF(OR(G34="Ju	D14

3. Predictive Modelling

Crime prediction using reported data (2019-2023)

Predictive model:
We started our prediction activity by building a Linear Regression model using ‘statsmodels’ on Jupyter Notebook to predict the crime based on the reported data. To identify the significance, we performed an Ordinary Least Squares regression (OLS), a useful and effective method to assess the relationship between all independent variables used for the prediction (*Ordinary least squares regression (OLS),* n.d). We aim to keep variables that have P-values for the T-test of less than 0.05 (rejects ‘null hypothesis’) and reject the ones higher than 0.05. We found all the variables to be significant/ contributing to the model (P-value for the T-test less than 0.05); we are not removing any attributes from our model.

Here is what our model looks like:

$$\text{Number\_Of\_Crime} = \beta 0 + \beta 1 * \text{OCC\_YEAR} + \beta 2 * \text{Total\_Age\_groups\_of\_the\_population} + \\ \beta 3 * \text{Married\_CommonLaw\_Rate} + \beta 4 * \text{Education\_Rate} + \beta 5 * \text{Employment\_Rate} + \\ \beta 6 * \text{Average\_age} + \beta 7 * \text{Average\_total\_income\_in\_2020}$$

Data quality validation after the final model has been developed:

Heteroskedasticity detection:
To check for heteroscedasticity, we produced two plots: ‘Standardized Residuals’ vs. ‘Fitted value’ and ‘Root of Standardized Residuals’ vs. ‘Fitted value.’ The data points in both plots form a cone-shaped pattern, indicating heteroskedasticity. In addition, the results from the Breuch-Pagan Test also reflected heteroscedasticity.

Removal of Heteroskedasticity:
HCCME was used to correct the interference in the regression. HCCME produced new regression results; the results of the corrected model also confirm all variables of the corrected model are significant.

Normality Issue detection:
The Normal QQ Plot and Density Plot were plotted to identify normality issues. From the plots, the Normal QQ plot shows data points deviate from the diagonal line on the right side, while the Density plot shows slight right-skewness. This is an indication that the dataset has normality issues. It is not normally distributed.

Outlier detection:
Influence Plot was plotted to identify potential issues. The plot shows apparent outliers with high leverage and influence within the dataset. However, the outliers in the dataset need not be removed as the data points are valid.

Prediction for total crimes:
Using our corrected/ final model, a predictive function was employed to forecast the number of total crimes in 2024 per neighborhood, which is ~140. With that **predicted value of 140**, we multiplied by 158 neighborhoods and got 22,120 as the forecasted crime count for all

neighborhoods in 2024. In conclusion, the **predicted number of crimes for break-and-enter in 2024 is 22,120**.

Prediction for break-and-enter crimes:
Using our corrected/final model, a predictive function was employed to forecast the number of break-and-enter crimes in 2024 per neighborhood, which is ~39. With that **predicted value of 38**, we multiplied by 158 neighborhoods and got 6,162 as the forecasted crime count for all neighborhoods in 2024. In conclusion, the **predicted number of crimes for break-and-enter in 2024 is 6,162.**

Prediction for auto theft crimes:
Using our corrected/ final model, a predictive function was employed to forecast the number of auto theft crimes in 2024 per neighborhood, which is ~82. With that **predicted value of 82**, we multiplied by 158 neighborhoods and got 13,114 as the forecasted crime count for all neighborhoods in 2024. In conclusion, the **predicted number of crimes for break-and-enter in 2024 is 13,114.**

Transformations were attempted to 'fix' the issues:

- Log transformation on variables: Average_total_income_in_2020, Total_Age_groups_of_the_population.
- Log on log transformation on variables: Average_total_income_in_2020, Total_Age_groups_of_the_population.
- Square root transformation on variables: Average_total_income_in_2020, Total_Age_groups_of_the_population.

These have resulted in worse conditions or did not improve from the original output.

Total Crime Predictions Formulation and Python Results

The final model was built to **predict the number of crimes** after dropping insignificant variables: (Alpha α = 0.05)

Number_Of_Crime = $\beta 0$ + $\beta 1$ * OCC_YEAR + $\beta 2$ * Total_Age_groups_of_the_population + $\beta 3$ * Married_CommonLaw_Rate + $\beta 4$ * Education_Rate + $\beta 5$ * Employment_rate + $\beta 6$ * Average_total_income_in_2020.

R-squared: 0.311
F-test
H0: $\beta 0 = \beta 1 = \beta 2 = \beta 3 = \beta 4 = \beta 5 = \beta 6 = 0$
H1: {at least one} $\beta i \neq 0$, i = 1,2,3,4,5,6
P value for F-test: 3.05e-60 < 0.05. Therefore, the model is significant.

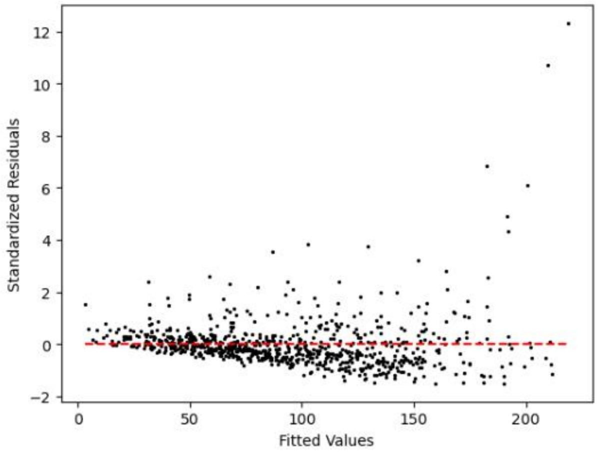
p-value for T-test:

- OCC_YEAR: 0.00
- Total_Age_groups_of_the_population: 0.000
- Married_CommonLaw_Rate: 0.001
- Education_Rate: 0.005

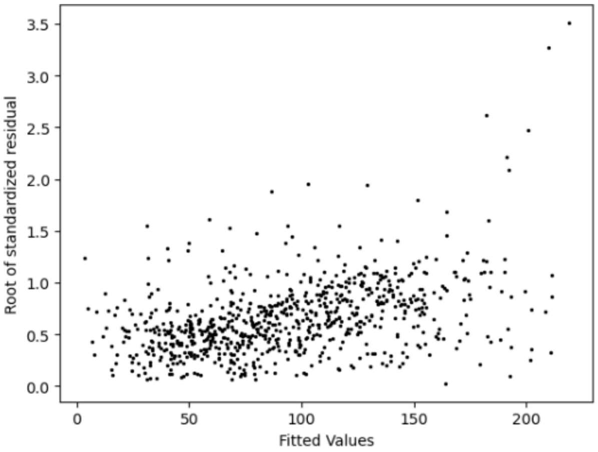
- Employment Rate: 0.043
- Average_total_income_in_2020: 0.000

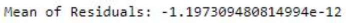
Therefore, all variables are significant and explain the number of crimes.

Residual vs Fitted Values Plot



Scale Location Plot

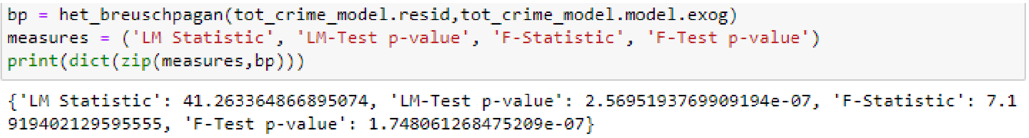




```
Mean of Residuals: -1.197309480814994e-12
```

The Mean of Residuals: -1.197309480814994e-12.
Both the Residual vs Fitted and Scale Location plots show a cone-shaped pattern.

Breuch-Pagan Test



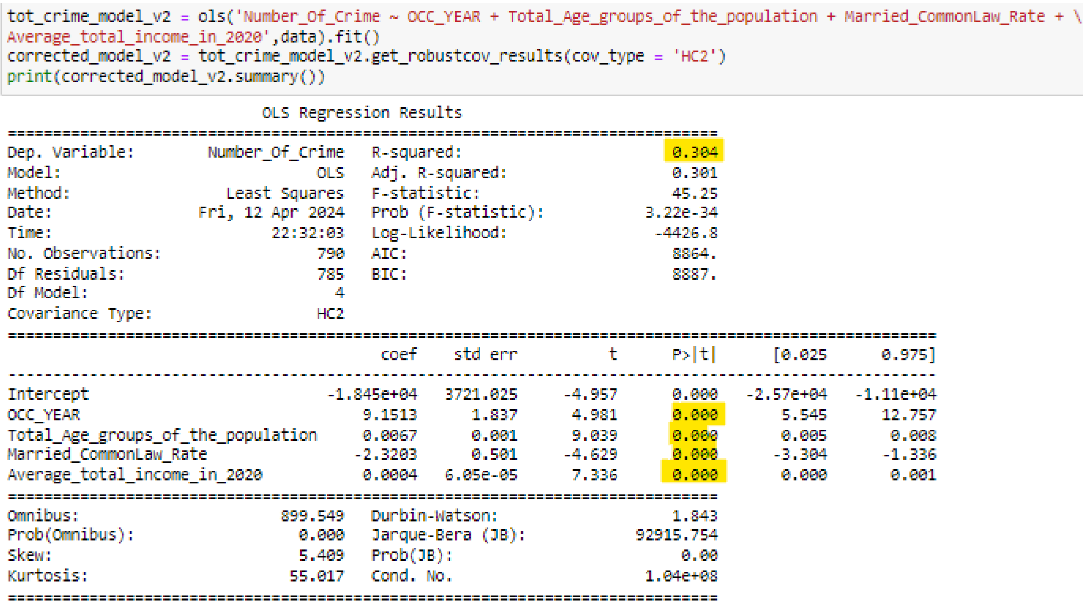
```
bp = het_breuschpagan(tot_crime_model.resid,tot_crime_model.model.exog)
measures = ('LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value')
print(dict(zip(measures,bp)))
```

```
{'LM Statistic': 41.263364866895074, 'LM-Test p-value': 2.5695193769909194e-07, 'F-Statistic': 7.1
919402129595555, 'F-Test p-value': 1.748061268475209e-07}
```

The Scale Location Plot and the Breuch-Pagan Test are effective ways to identify heteroskedasticity.

- Scale Location Plot: the residuals increase as the fitted values increase, which is indicative of heteroskedasticity.
- Breuch-Pagan Test:
  - 'LM-Test p-value': 2.5695193769909194e-07
  - 'F-Test p-value': 1.748061268475209e-07
  - They are both less than 0.05; the model is found to be heteroskedastic.

Heteroskedasticity resolution:
HCCME to resolve the heteroskedasticity: get_robustcov_results(cov_type = 'HC2')



```
tot_crime_model_v2 = ols('Number_Of_Crime ~ OCC_YEAR + Total_Age_groups_of_the_population + Married_CommonLaw_Rate + \
Average_total_income_in_2020',data).fit()
corrected_model_v2 = tot_crime_model_v2.get_robustcov_results(cov_type = 'HC2')
print(corrected_model_v2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Number_Of_Crime   R-squared:                       0.304
Model:                            OLS   Adj. R-squared:                  0.301
Method:                 Least Squares   F-statistic:                     45.25
Date:                Fri, 12 Apr 2024   Prob (F-statistic):           3.22e-34
Time:                        22:32:03   Log-Likelihood:                -4426.8
No. Observations:                 790   AIC:                             8864.
Df Residuals:                     785   BIC:                             8887.
Df Model:                           4
Covariance Type:                  HC2
======================================================================================================
                                         coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------------
Intercept                          -1.845e+04   3721.025     -4.957      0.000   -2.57e+04   -1.11e+04
OCC_YEAR                               9.1513      1.837      4.981      0.000       5.545      12.757
Total_Age_groups_of_the_population     0.0067      0.001      9.039      0.000       0.005       0.008
Married_CommonLaw_Rate                -2.3203      0.501     -4.629      0.000      -3.304      -1.336
Average_total_income_in_2020           0.0004   6.05e-05      7.336      0.000       0.000       0.001
==============================================================================
Omnibus:                      899.549   Durbin-Watson:                   1.843
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            92915.754
Skew:                           5.409   Prob(JB):                         0.00
Kurtosis:                      55.017   Cond. No.                     1.04e+08
==============================================================================
```

R-squared: 0.304

P value for F-test: 3.22e-34 < 0.05. Therefore, the model is significant.

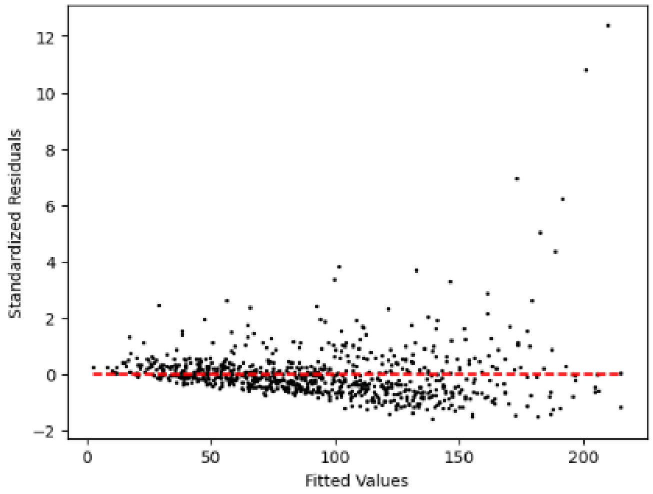
p-value for T-test:

- OCC_YEAR: 0.000
- Total_Age_groups_of_the_population: 0.000
- Married_CommonLaw_Rate: 0.000
- Average_total_income_in_2020: 0.000

Final model after correcting for heteroscedasticity:

Number_Of_Crime = $\beta 0$ + $\beta 1$ * OCC_YEAR + $\beta 2$ * Total_Age_groups_of_the_population + $\beta 3$ * Married_CommonLaw_Rate + $\beta 4$* Average_total_income_in_2020''

Residual vs Fitted Values Plot

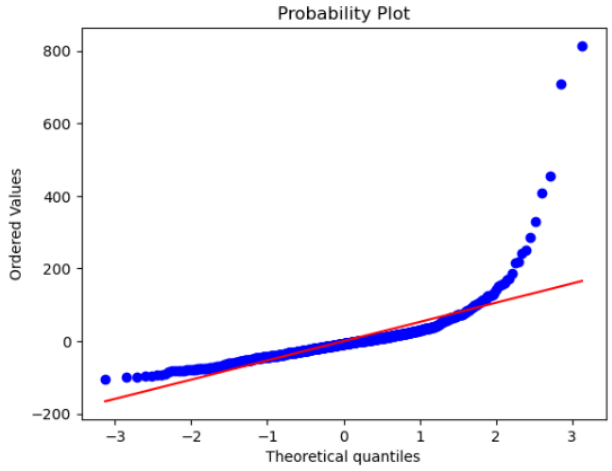


```
Mean of Residuals: -4.466885370732863e-13
```

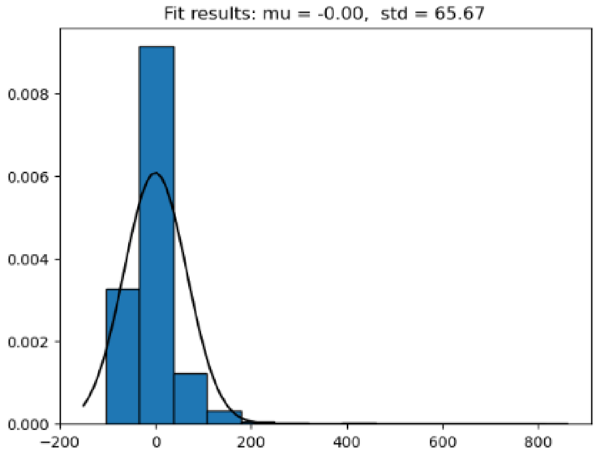
The Mean of Residuals: -4.466885370732863e-13

The Residual vs Fitted plot still shows a cone-shaped pattern after using HCCME to correct for heteroskedasticity.

Normal-QQ Plot



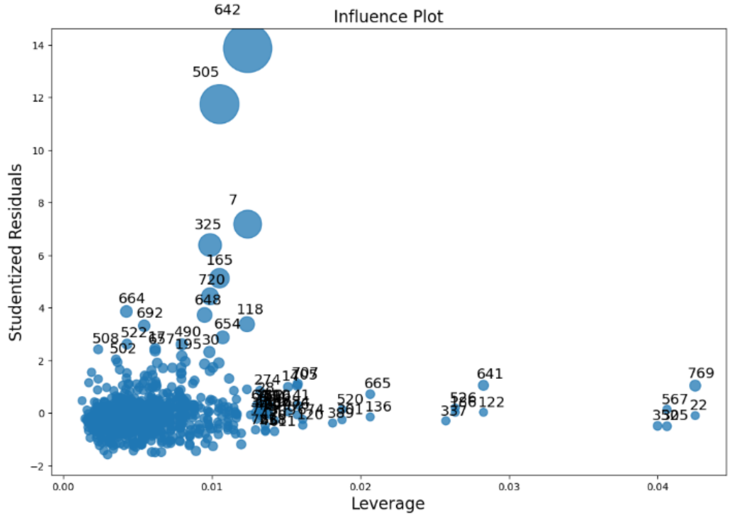
Density Plot



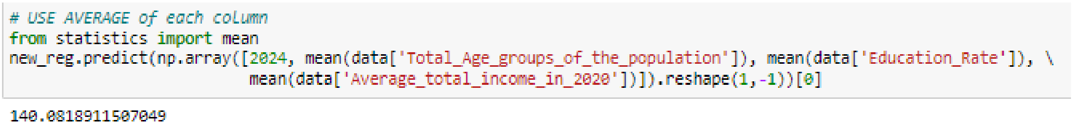
We can see from the plots above that the dataset exhibits strong normality issues.

- The Normal QQ plot shows that some data points deviate from the diagonal line on the right side. This is an indication that the dataset has normality issues. It is not normally distributed.
- The Density plot shows right-skewness. It is an asymmetric histogram.

Residual vs Leverage



We can see from the Influence Plot that the dataset has data points with high leverage and influence.



```
# USE AVERAGE of each column
from statistics import mean
new_reg.predict(np.array([2024, mean(data['Total_Age_groups_of_the_population']), mean(data['Education_Rate']), \
                          mean(data['Average_total_income_in_2020'])]).reshape(1,-1))[0]
```

```
140.0818911507049
```

Prediction for the number of total crimes for 2024:

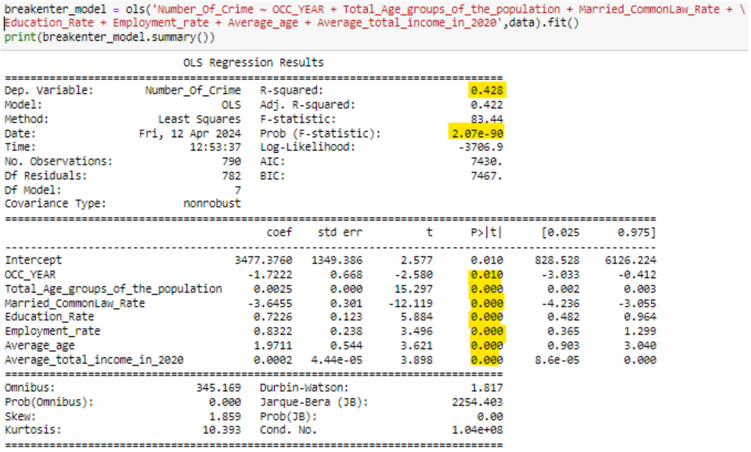
- The average of all individual attributes was used for prediction: Total_Age_groups_of_the_population, Married_CommonLaw_Rate, Education_Rate, Employment_rate, Average_age, Average_total_income_in_2020.
- There are 158 neighborhoods in the dataset.

With that **predicted value of 140,** we multiplied by 158 neighborhoods and **got 22,120 as the forecasted crime count for all neighborhoods in 2024.**

Break-and-Enter Predictions Formulation and Python Results

The final model was built to **predict the number of break-and-enter crimes**: (Alpha α = 0.05)

Number_Of_Crime = $\beta0$ + $\beta1$ * OCC_YEAR + $\beta2$ * Total_Age_groups_of_the_population + $\beta3$ * Married_CommonLaw_Rate + $\beta4$ * Education_Rate + $\beta5$ * Employment_rate + $\beta6$ * Average_age + $\beta7$ * Average_total_income_in_2020



```
breakenter_model = ols('Number_Of_Crime ~ OCC_YEAR + Total_Age_groups_of_the_population + Married_CommonLaw_Rate + \
Education_Rate + Employment_rate + Average_age + Average_total_income_in_2020',data).fit()
print(breakenter_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Number_Of_Crime   R-squared:                       0.428
Model:                            OLS   Adj. R-squared:                  0.422
Method:                 Least Squares   F-statistic:                     83.44
Date:                Fri, 12 Apr 2024   Prob (F-statistic):           2.07e-90
Time:                        12:53:37   Log-Likelihood:                -3706.9
No. Observations:                 790   AIC:                             7430.
Df Residuals:                     782   BIC:                             7467.
Df Model:                           7
Covariance Type:            nonrobust
======================================================================================================
                                         coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------------
Intercept                           3477.3760   1349.386      2.577      0.010     828.528    6126.224
OCC_YEAR                              -1.7222      0.668     -2.580      0.010      -3.033      -0.412
Total_Age_groups_of_the_population     0.0025      0.000     15.297      0.000       0.002       0.003
Married_CommonLaw_Rate                -3.6455      0.301    -12.119      0.000      -4.236      -3.055
Education_Rate                         0.7226      0.123      5.884      0.000       0.482       0.964
Employment_rate                        0.8322      0.238      3.496      0.000       0.365       1.299
Average_age                            1.9711      0.544      3.621      0.000       0.903       3.040
Average_total_income_in_2020           0.0002   4.44e-05      3.898      0.000     8.6e-05       0.000
==============================================================================
Omnibus:                      345.169   Durbin-Watson:                   1.817
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2254.403
Skew:                           1.859   Prob(JB):                         0.00
Kurtosis:                      10.393   Cond. No.                     1.04e+08
==============================================================================
```

R-squared: 0.428

F-test

H0: $\beta0 = \beta1 = \beta2 = \beta3 = \beta4 = \beta5 = \beta6 = \beta7 = 0$

H1: {at least one} $\beta i \neq 0$, i = 1,2,3,4,5,6,7

P value for F-test: 2.07e-90 < 0.05

Therefore, the model is significant.

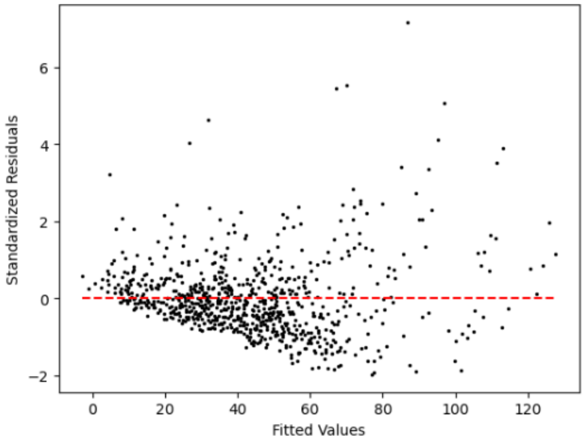
p-value for T-test:

- OCC_YEAR: 0.010
- Total_Age_groups_of_the_population: 0.000
- Married_CommonLaw_Rate: 0.000
- Education_Rate: 0.000
- Employment_rate: 0.000
- Average_age: 0.000

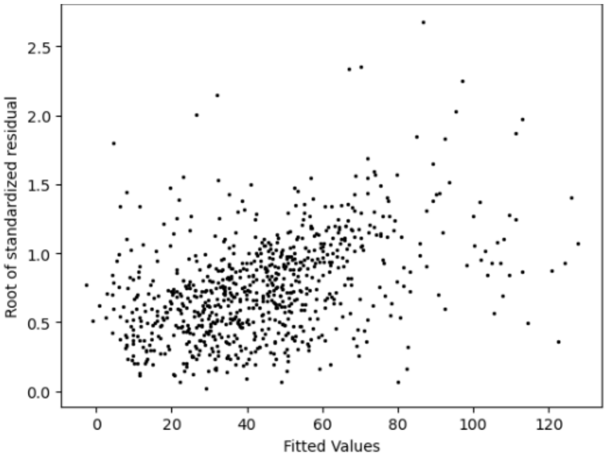
- Average_total_income_in_2020: 0.000

Therefore, all variables are significant as they all have p-value < 0.05.

Residual vs Fitted Values Plot (LEFT) **Scale Location Plot** (RIGHT)



```
Mean of Residuals: -4.93314531278101e-13
```



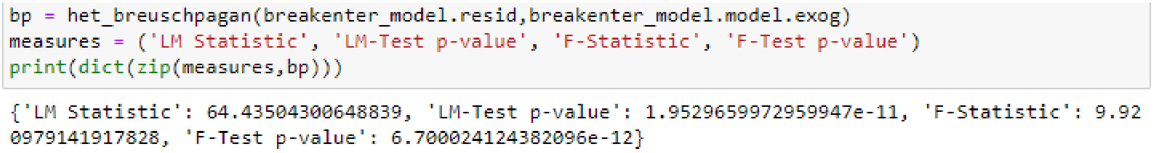
The (LEFT) Residual vs Fitted Value plot

- The Mean of Residuals: -4.93314531278101e-13
- It shows an apparent cone-shaped pattern.

The (RIGHT) Residual vs. Fitted values plot

- It shows that the data points are scattered around the horizontal line at x=0, but the points are concentrated on the left and scatter and pan out towards the right.

Breuch-Pagan Test



```
bp = het_breuschpagan(breakenter_model.resid,breakenter_model.model.exog)
measures = ('LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value')
print(dict(zip(measures,bp)))
```

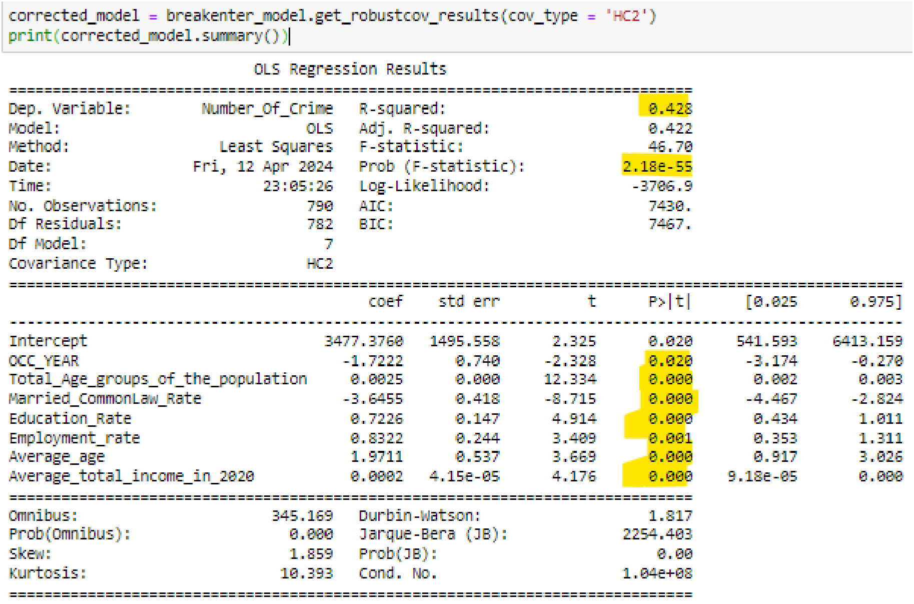
```
{'LM Statistic': 64.43504300648839, 'LM-Test p-value': 1.9529659972959947e-11, 'F-Statistic': 9.92
0979141917828, 'F-Test p-value': 6.700024124382096e-12}
```

The Scale Location Plot and the Breuch-Pagan Test are effective ways to identify heteroskedasticity.

- Scale Location Plot: the residuals increase as the fitted values increase, which indicates heteroskedasticity.
- Breuch-Pagan Test:
  - 'LM-Test p-value': 1.9529659972959947e-11
  - 'F-Test p-value': 6.700024124382096e-12
  - They are both less than 0.05, the model is found to be heteroskedastic.

Heteroskedasticity resolution:

HCCME to resolve the heteroskedasticity: get_robustcov_results(cov_type = 'HC2')



```
corrected_model = breakenter_model.get_robustcov_results(cov_type = 'HC2')
print(corrected_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Number_Of_Crime   R-squared:                       0.428
Model:                            OLS   Adj. R-squared:                  0.422
Method:                 Least Squares   F-statistic:                     46.70
Date:                Fri, 12 Apr 2024   Prob (F-statistic):           2.18e-55
Time:                        23:05:26   Log-Likelihood:                -3706.9
No. Observations:                 790   AIC:                             7430.
Df Residuals:                     782   BIC:                             7467.
Df Model:                           7
Covariance Type:                  HC2
======================================================================================================
                                         coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------------
Intercept                           3477.3760   1495.558      2.325      0.020     541.593    6413.159
OCC_YEAR                              -1.7222      0.740     -2.328      0.020      -3.174      -0.270
Total_Age_groups_of_the_population     0.0025      0.000     12.334      0.000       0.002       0.003
Married_CommonLaw_Rate                -3.6455      0.418     -8.715      0.000      -4.467      -2.824
Education_Rate                         0.7226      0.147      4.914      0.000       0.434       1.011
Employment_rate                        0.8322      0.244      3.409      0.001       0.353       1.311
Average_age                            1.9711      0.537      3.669      0.000       0.917       3.026
Average_total_income_in_2020           0.0002   4.15e-05      4.176      0.000    9.18e-05       0.000
==============================================================================
Omnibus:                      345.169   Durbin-Watson:                   1.817
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2254.403
Skew:                           1.859   Prob(JB):                         0.00
Kurtosis:                      10.393   Cond. No.                     1.04e+08
==============================================================================
```

R-squared: 0.428

F-test

H0: $\beta0 = \beta1 = \beta2 = \beta3= \beta4 = \beta5 = \beta6 = \beta7 = 0$

H1: {at least one} $\beta i \neq 0$, i = 1,2,3,4,5,6,7

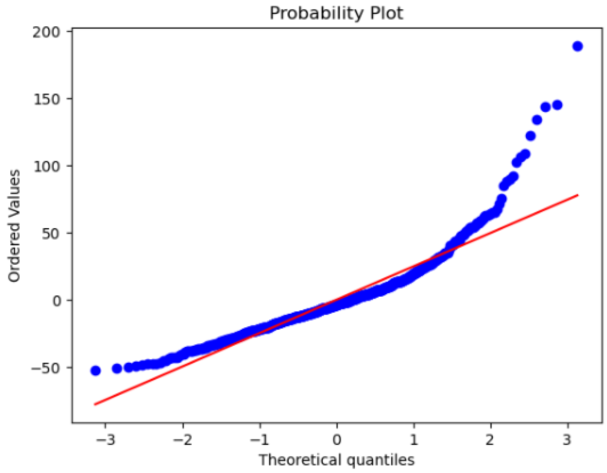
P value for F-test: 2.18e-55 < 0.05. Therefore, the model is significant.

P value for T-test:

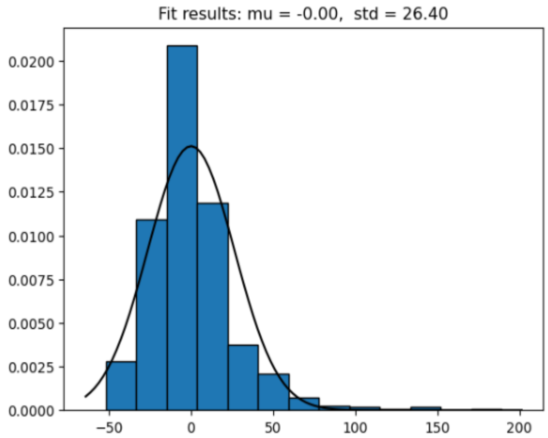
- OCC_YEAR: 0.20
- Total_Age_groups_of_the_population: 0.000
- Married_CommonLaw_Rate: 0.000
- Education_Rate: 0.000
- Employment_rate: 0.001
- Average_age: 0.000
- Average_total_income_in_2020: 0.000

Therefore, all variables are significant as their p-values are less than 0.05.

Normal-QQ Plot



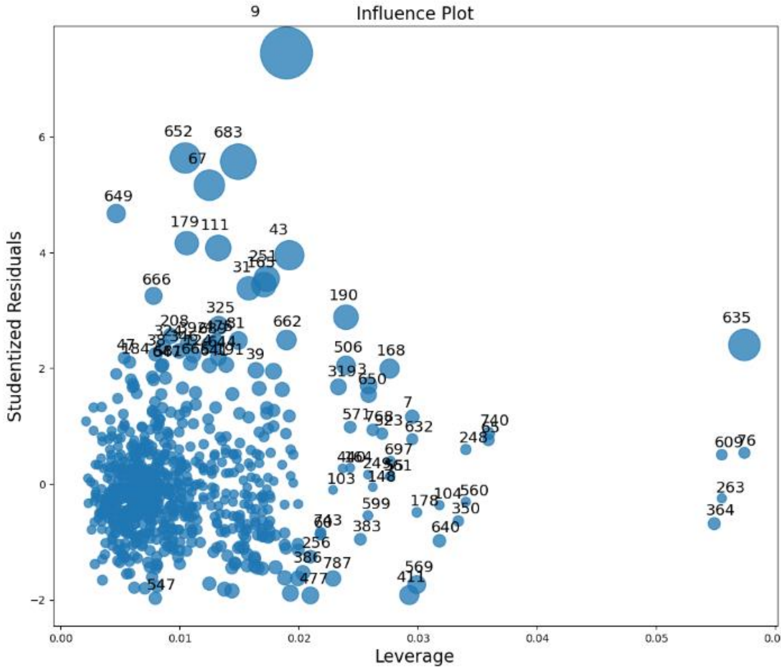
Density Plot



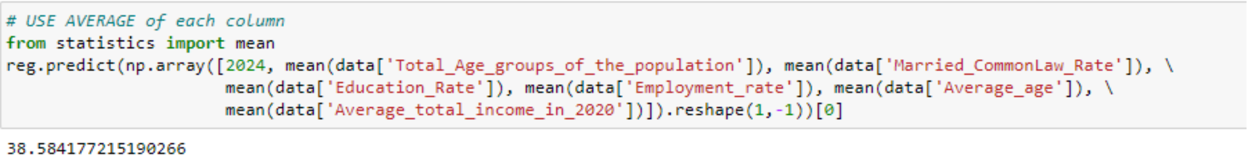
We can see from the plots above that the dataset exhibits normality issues.

- The Normal QQ plot shows that some data points deviate from the diagonal line on the right side. This is an indication that the dataset has normality issues. It is not normally distributed.
- The Density plot shows slight right-skewness. It is an asymmetric histogram.

Influence Plot



The plot shows apparent outliers with high leverage and influence within the dataset. However, the outliers in the dataset need not be removed as the data points are valid.



```
# USE AVERAGE of each column
from statistics import mean
reg.predict(np.array([2024, mean(data['Total_Age_groups_of_the_population']), mean(data['Married_CommonLaw_Rate']), \
                      mean(data['Education_Rate']), mean(data['Employment_rate']), mean(data['Average_age']), \
                      mean(data['Average_total_income_in_2020'])]).reshape(1,-1))[0]
```

```
38.584177215190266
```

Prediction for the number of break & enter crimes for 2024:

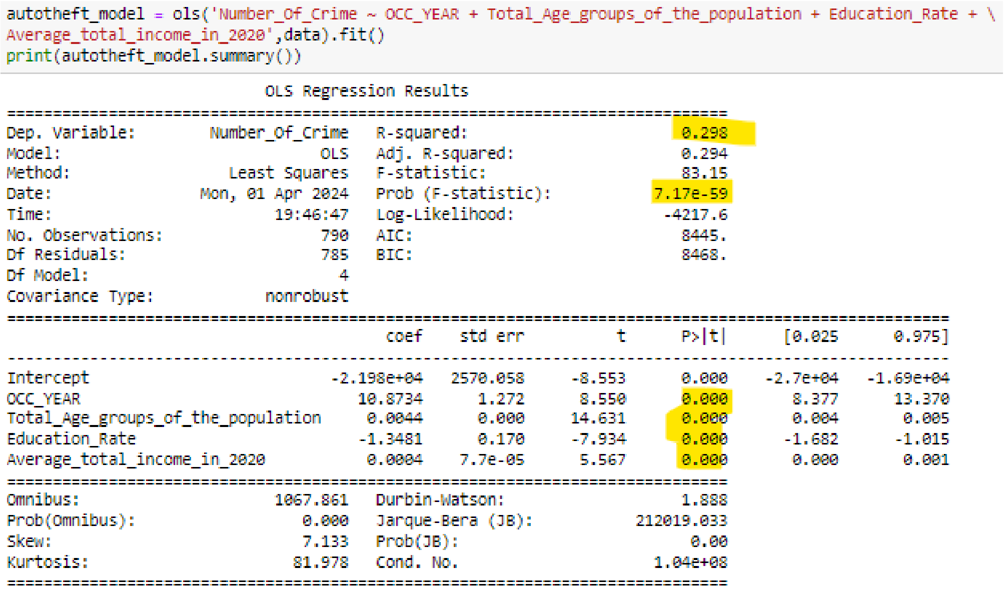
- The average of all individual attributes was used for prediction: Total_Age_groups_of_the_population, Married_CommonLaw_Rate, Education_Rate, Employment_rate, Average_age, Average_total_income_in_2020.
- There are 158 neighborhoods in the dataset.

With that **predicted value of 38**, we multiplied by 158 neighborhoods and **got 6,162 as the forecasted crime count for all neighborhoods in 2024.**

Auto theft Predictions Formulation and Python Results

The final model was built to **predict the number of auto theft crimes** after dropping insignificant variables: (Alpha α = 0.05)

Number_Of_Crime = $\beta 0 + \beta 1$ * OCC_YEAR + $\beta 2$ * Total_Age_groups_of_the_population + $\beta 3$ * Education_Rate + $\beta 4$ * Average_total_income_in_2020.



```
autotheft_model = ols('Number_Of_Crime ~ OCC_YEAR + Total_Age_groups_of_the_population + Education_Rate + \
Average_total_income_in_2020',data).fit()
print(autotheft_model.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:        Number_Of_Crime   R-squared:                       0.298
Model:                            OLS   Adj. R-squared:                  0.294
Method:                 Least Squares   F-statistic:                     83.15
Date:                Mon, 01 Apr 2024   Prob (F-statistic):           7.17e-59
Time:                        19:46:47   Log-Likelihood:                -4217.6
No. Observations:                 790   AIC:                             8445.
Df Residuals:                     785   BIC:                             8468.
Df Model:                           4                                         
Covariance Type:            nonrobust                                         
======================================================================================================
                                         coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------------
Intercept                          -2.198e+04   2570.058     -8.553      0.000    -2.7e+04   -1.69e+04
OCC_YEAR                              10.8734      1.272      8.550      0.000       8.377      13.370
Total_Age_groups_of_the_population     0.0044      0.000     14.631      0.000       0.004       0.005
Education_Rate                        -1.3481      0.170     -7.934      0.000      -1.682      -1.015
Average_total_income_in_2020           0.0004    7.7e-05      5.567      0.000       0.000       0.001
==============================================================================
Omnibus:                     1067.861   Durbin-Watson:                   1.888
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           212019.033
Skew:                           7.133   Prob(JB):                         0.00
Kurtosis:                      81.978   Cond. No.                     1.04e+08
==============================================================================
```

R-squared: 0.298

F-test

H0: $\beta 0 = \beta 1 = \beta 2 = \beta 3 = \beta 4 = 0$

H1: {at least one} $\beta i \neq 0$, i = 1,2,3,4

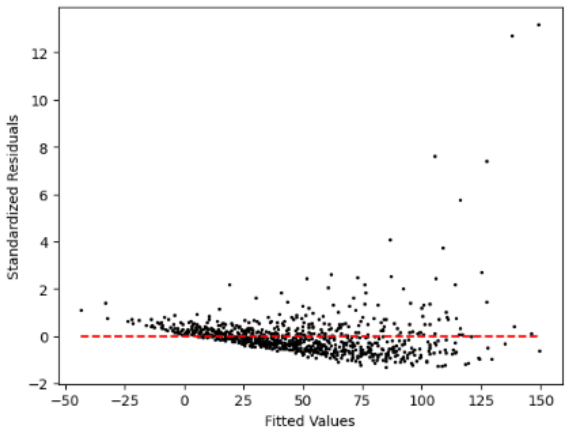
P value for F-test: 7.17e-59 < 0.05. Therefore, the model is significant.

P value for T-test:

- OCC_YEAR: 0.000
- Total_Age_groups_of_the_population: 0.000
- Education_Rate: 0.000
- Average_total_income_in_2020: 0.000

Therefore, all variables are significant as their p-values are less than 0.05.

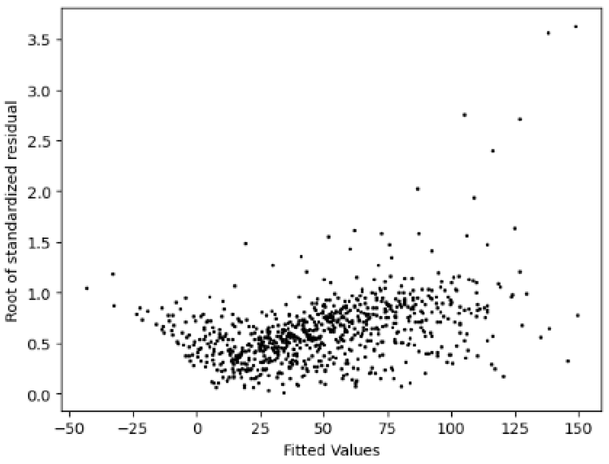
Residual vs Fitted Values Plot



```
Mean of Residuals: -1.3999310852606085e-12
```

(LEFT)

Scale Location Plot



(RIGHT)

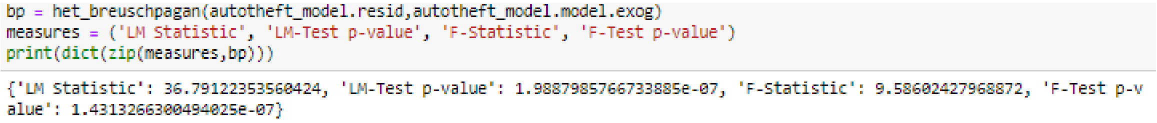
The (LEFT) Residual vs Fitted Value plot

- The Mean of Residuals: -1.3999310852606085e-12
- It shows an apparent cone-shaped pattern.

The (RIGHT) Residual vs. Fitted values plot

- It shows an apparent cone-shaped pattern where the points are concentrated on the around x=0 and scatter and pan out towards the right.

Breuch-Pagan Test



```
bp = het_breuschpagan(autotheft_model.resid,autotheft_model.model.exog)
measures = ('LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value')
print(dict(zip(measures,bp)))
```

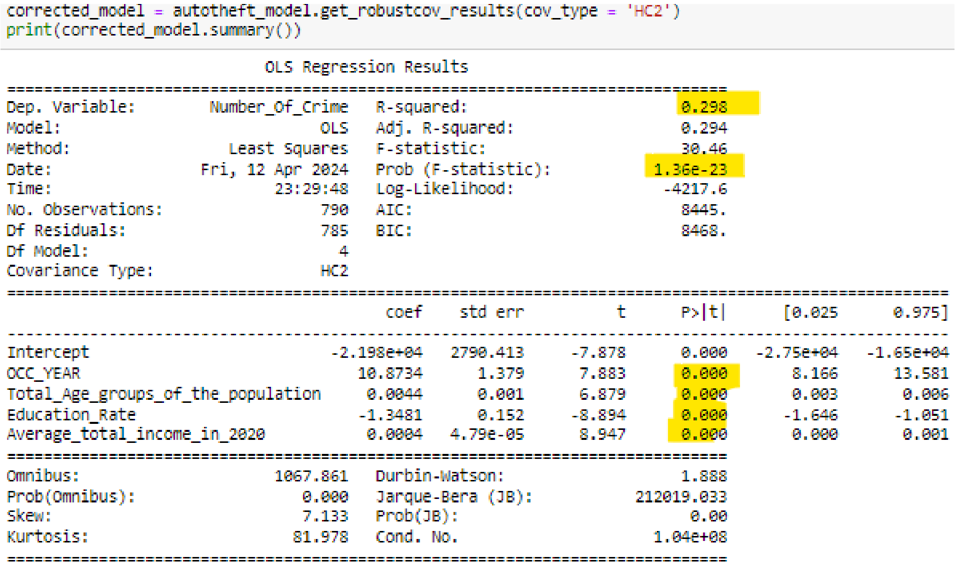
```
{'LM Statistic': 36.79122353560424, 'LM-Test p-value': 1.9887985766733885e-07, 'F-Statistic': 9.58602427968872, 'F-Test p-v
alue': 1.4313266300494025e-07}
```

The Scale Location Plot and the Breuch-Pagan Test are effective ways to identify heteroskedasticity.

- Scale Location Plot: the residuals increase as the fitted values increase, which indicates heteroskedasticity.
- Breuch-Pagan Test:

- 'LM-Test p-value': 1.9887985766733885e-07
- 'F-Test p-value': 1.4313266300494025e-07
- They are both less than 0.05, the model is found to be heteroskedastic.

Heteroskedasticity resolution:
HCCME to resolve the heteroskedasticity: get_robustcov_results(cov_type = 'HC2')



```
corrected_model = autotheft_model.get_robustcov_results(cov_type = 'HC2')
print(corrected_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Number_Of_Crime   R-squared:                       0.298
Model:                            OLS   Adj. R-squared:                  0.294
Method:                 Least Squares   F-statistic:                     30.46
Date:                Fri, 12 Apr 2024   Prob (F-statistic):           1.36e-23
Time:                        23:29:48   Log-Likelihood:                -4217.6
No. Observations:                 790   AIC:                             8445.
Df Residuals:                     785   BIC:                             8468.
Df Model:                           4
Covariance Type:                  HC2
=======================================================================================================
                                          coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------
Intercept                           -2.198e+04   2790.413     -7.878      0.000   -2.75e+04   -1.65e+04
OCC_YEAR                               10.8734      1.379      7.883      0.000       8.166      13.581
Total_Age_groups_of_the_population      0.0044      0.001      6.879      0.000       0.003       0.006
Education_Rate                         -1.3481      0.152     -8.894      0.000      -1.646      -1.051
Average_total_income_in_2020            0.0004   4.79e-05      8.947      0.000       0.000       0.001
==============================================================================
Omnibus:                     1067.861   Durbin-Watson:                   1.888
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           212019.033
Skew:                           7.133   Prob(JB):                         0.00
Kurtosis:                      81.978   Cond. No.                     1.04e+08
==============================================================================
```

R-squared: 0.298

F-test

H0: $\beta 0 = \beta 1 = \beta 2 = \beta 3 = \beta 4 = \beta 5 = \beta 6 = \beta 7 = 0$

H1: {at least one} $\beta i \neq 0$, i = 1,2,3,4,5,6,7

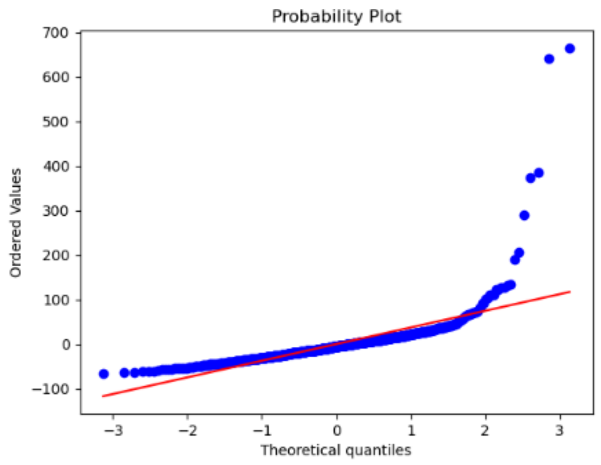
P value for F-test: 2.18e-55- model is significant

P value for T-test:

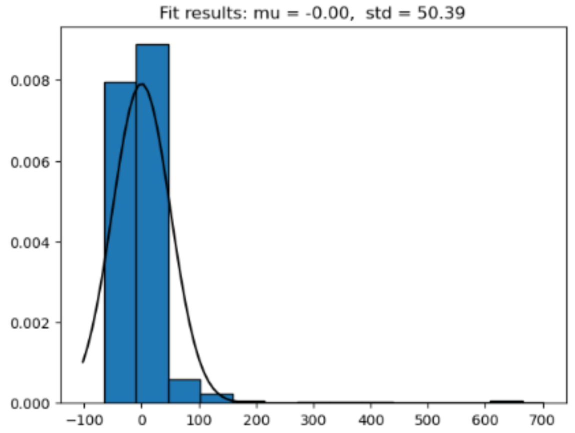
- OCC_YEAR: 0.000
- Total_Age_groups_of_the_population: 0.000
- Education_Rate: 0.000
- Average_total_income_in_2020: 0.000

Therefore, all variables are significant as their p-values are less than 0.05.

Normal-QQ Plot



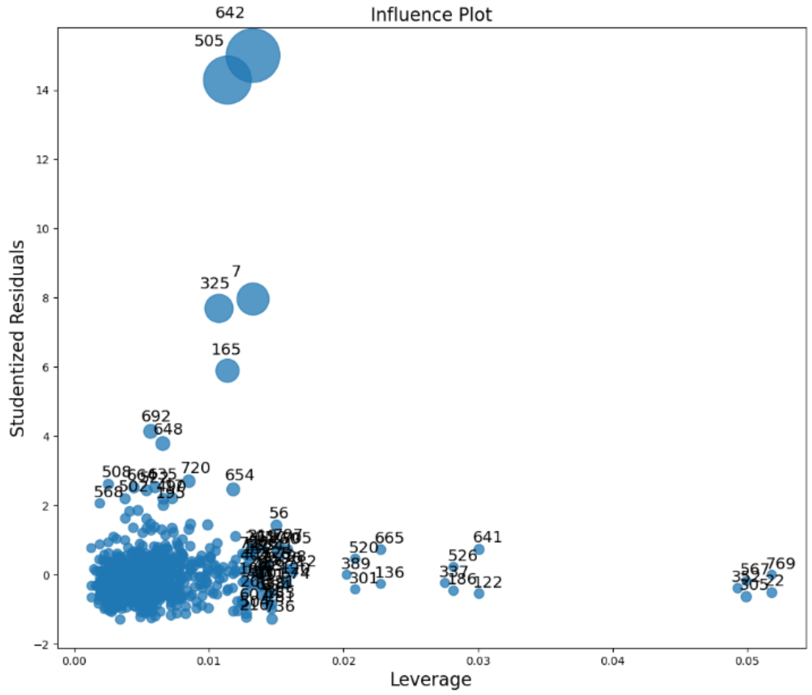
Density Plot



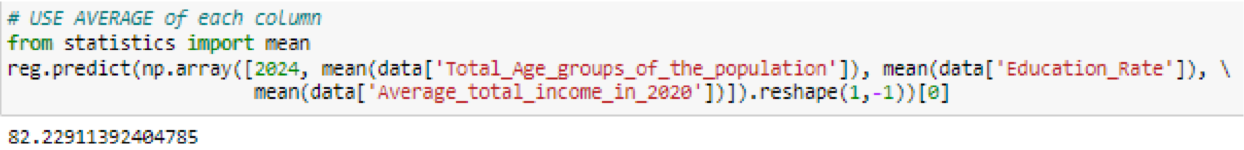
We can see from the plots above that the dataset exhibits strong normality issues.

- The Normal QQ plot shows that some data points deviate from the diagonal line on the right side. This is an indication that the dataset has normality issues. It is not normally distributed.
- The Density plot shows strong right-skewness.

Influence Plot



The plot shows apparent outliers with high leverage and influence within the dataset. However, the outliers in the dataset need not be removed as the data points are valid.



```
# USE AVERAGE of each column
from statistics import mean
reg.predict(np.array([2024, mean(data['Total_Age_groups_of_the_population']), mean(data['Education_Rate']), \
                      mean(data['Average_total_income_in_2020'])]).reshape(1,-1))[0]
```

```
82.22911392404785
```

Prediction for the number of auto theft crimes for 2024:

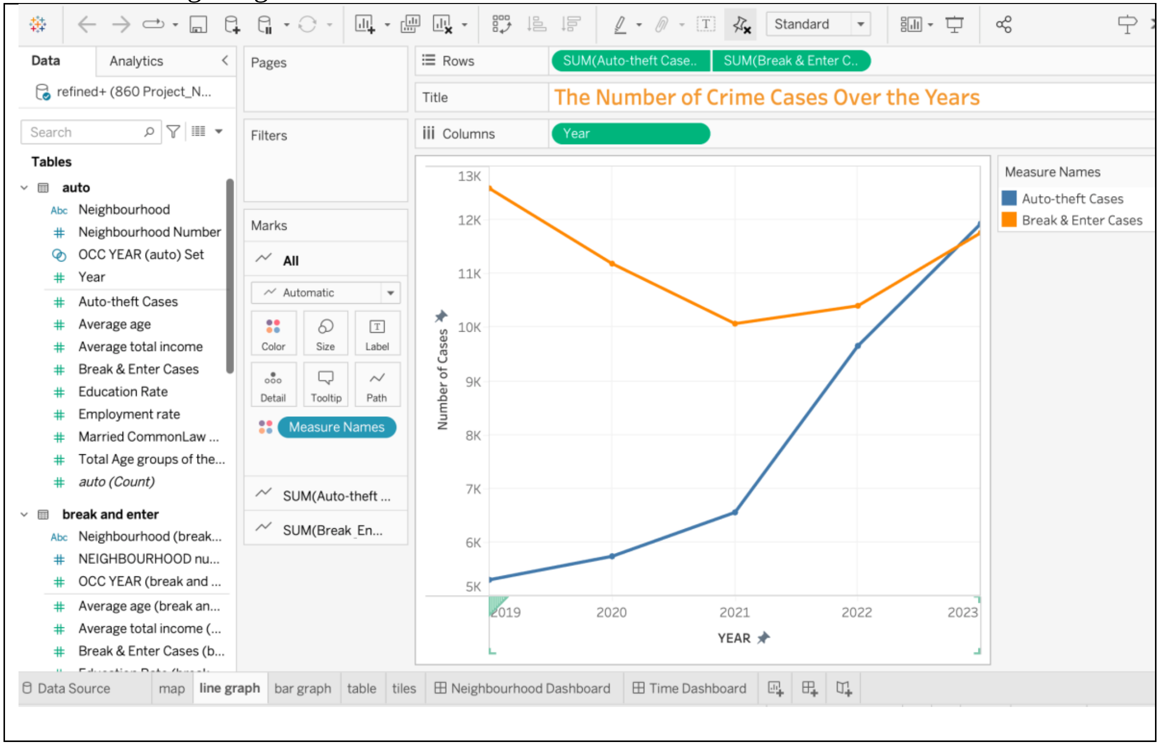
- The average of all individual attributes was used for prediction: Total_Age_groups_of_the_population, Married_CommonLaw_Rate, Education_Rate, Employment_rate, Average_age, Average_total_income_in_2020.
- There are 158 neighborhoods in the dataset.

With that **predicted value of 82**, we multiplied by 158 neighborhoods and **got 13,114 as the forecasted crime count for all neighborhoods in 2024.**

4. *Visualizing Data for Insight Generation*

Tableau was used for all visualizations.

- Line graph
  This graph is constructed using the auto table from the Data Source tab. Auto theft cases and Break & Enter cases are the measure fields that are getting used in rows. Year is the dimension attribute that is getting used in columns.



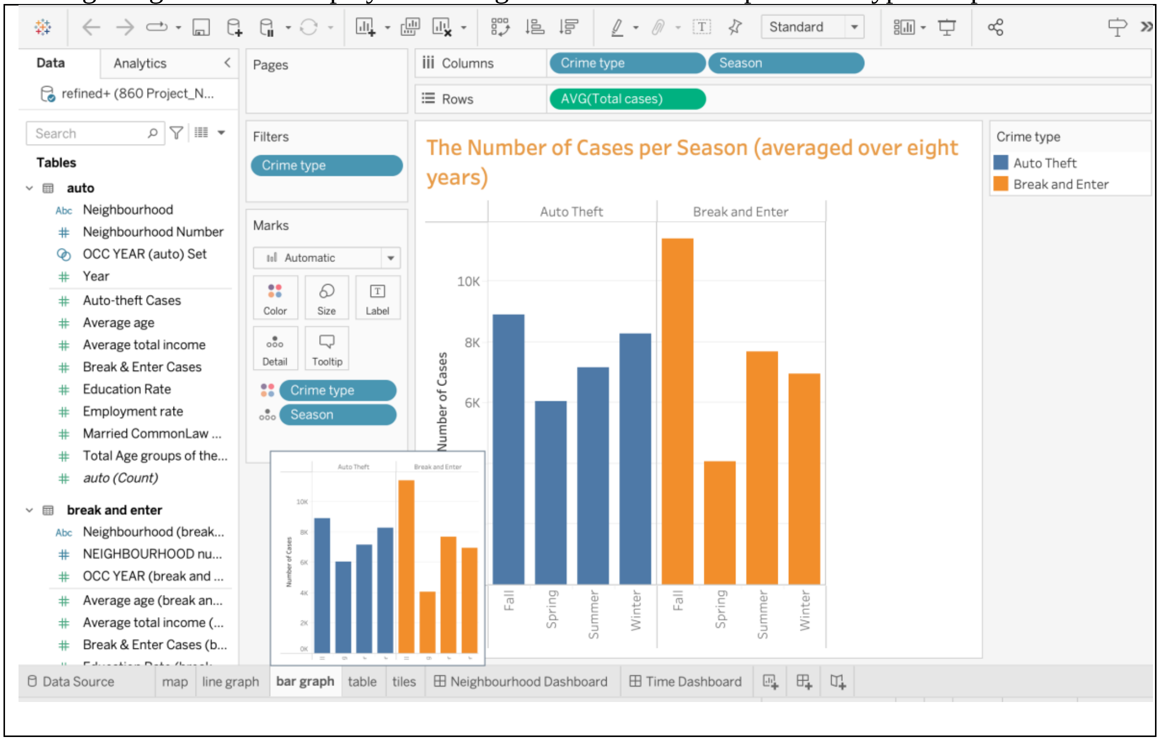
Rows: SUM(Auto-theft Case.. | SUM(Break & Enter C..

Title: The Number of Crime Cases Over the Years

Columns: Year

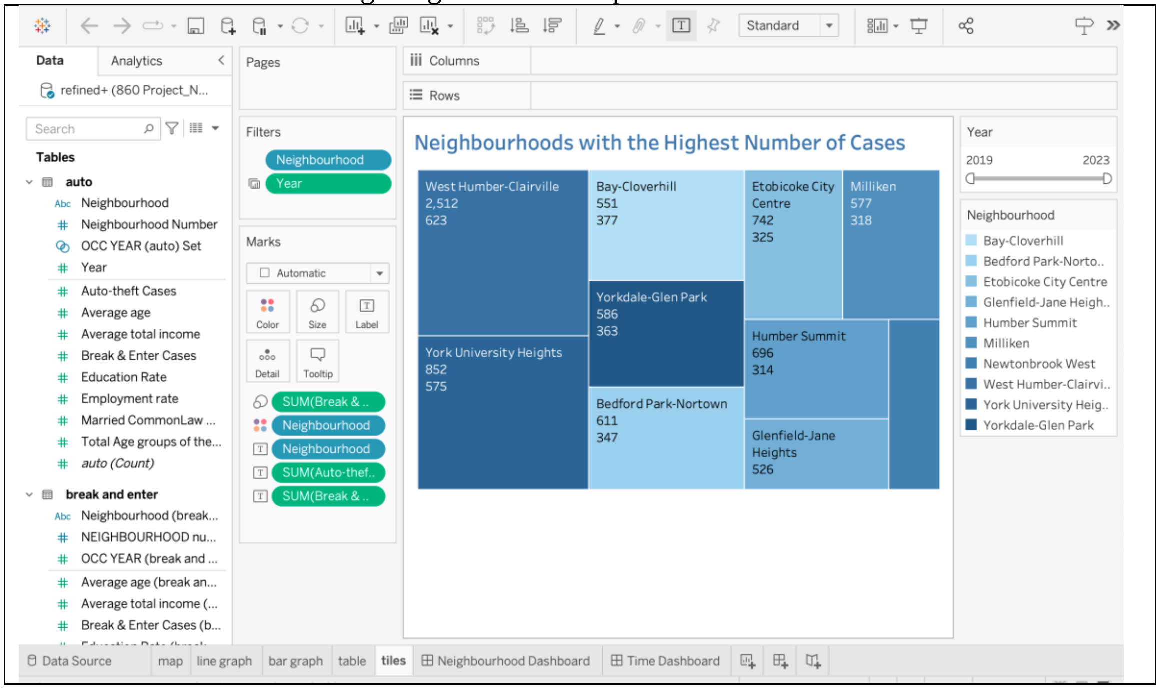
Marks: All — Automatic; Measure Names; SUM(Auto-theft ...; SUM(Break_En...

- Bar graph
  This graph is constructed using the refined table from the Data Source tab. Crime type and season are the attribute variables that are getting used as columns. Number of crimes is the measure field that is getting in rows to display the average number of crimes per crime type and per seasons.



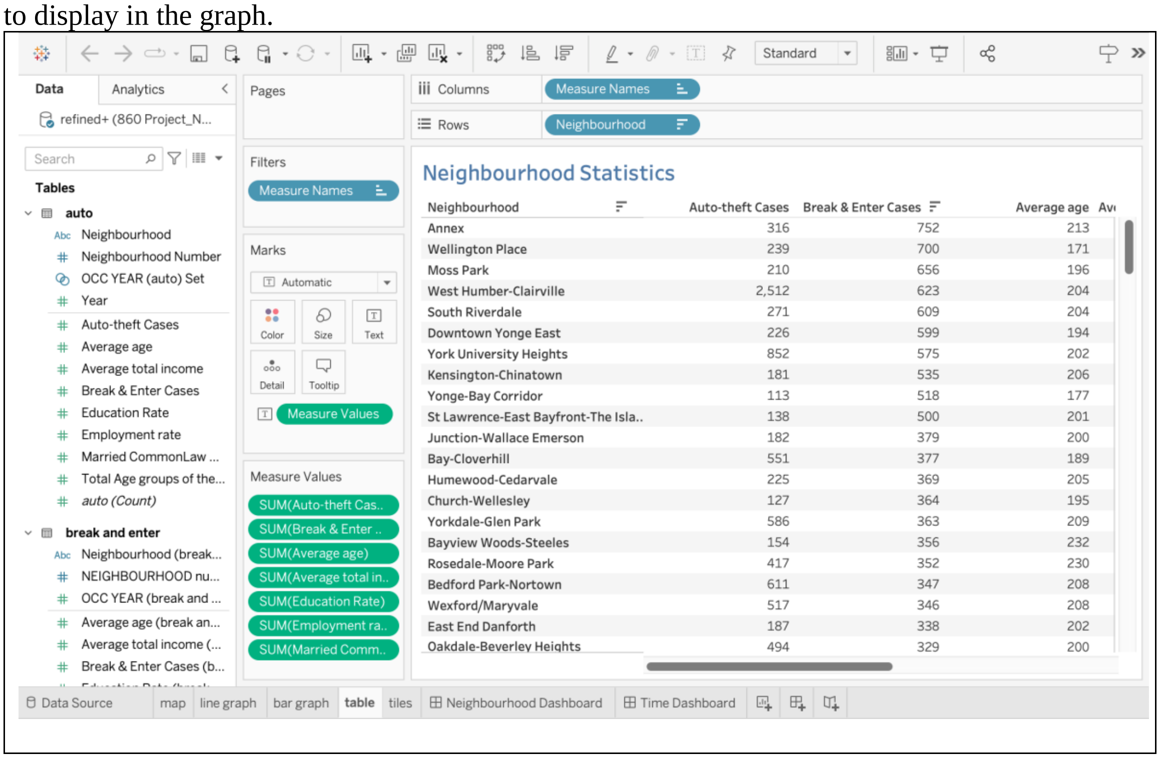
Columns: Crime type, Season
Rows: AVG(Total cases)
Filters: Crime type
Marks: Automatic; Color: Crime type; Detail: Season

- Tree-map
  This graph is constructed using the auto table from the Data Source tab. Auto theft cases and Break & Enter cases are the measure fields that are getting used in this graph. Neighbourhood is the dimension attribute that is getting used in this report.



Filters: Neighbourhood, Year
Marks: Automatic; Size: SUM(Break & ..; Color: Neighbourhood; Label: Neighbourhood, SUM(Auto-thef.., SUM(Break & ..
Year: 2019 – 2023

- Neighborhood Statistics Table
  We have used the auto theft, break & enter, and refined tables available from the Data Source tab to build this report. We have taken the raw files and aggregated the data per neighborhood level to display in the graph.



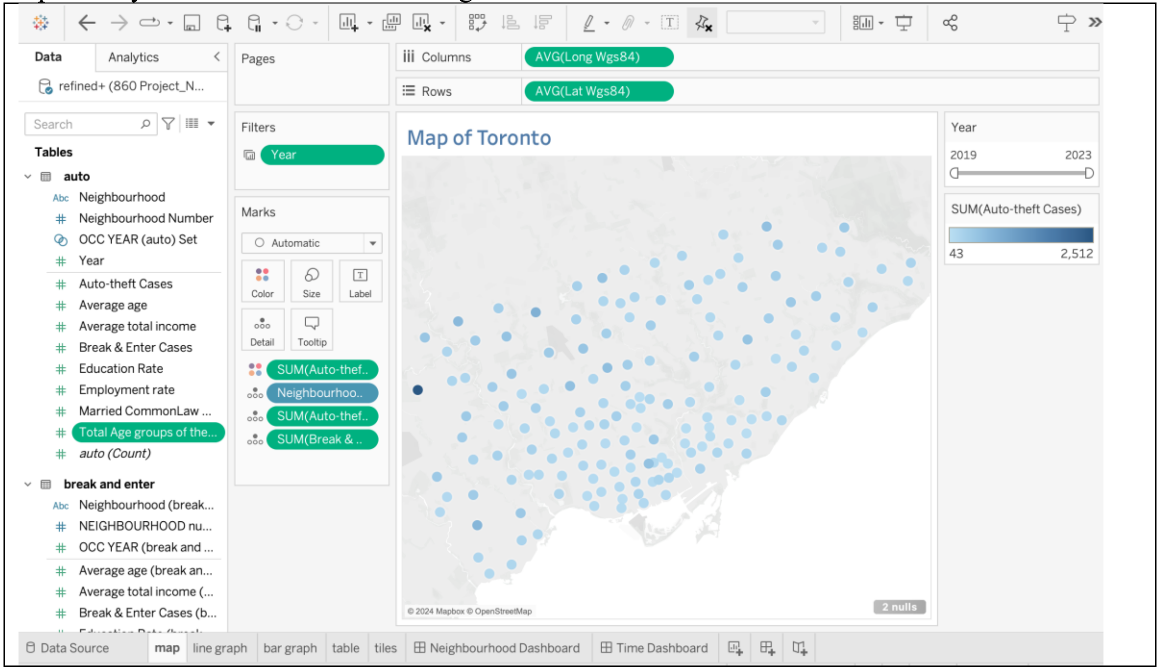
Columns: Measure Names

Rows: Neighbourhood

Neighbourhood Statistics

Neighbourhood	Auto-theft Cases	Break & Enter Cases	Average age	Av
Annex	316	752	213	
Wellington Place	239	700	171	
Moss Park	210	656	196	
West Humber-Clairville	2,512	623	204	
South Riverdale	271	609	204	
Downtown Yonge East	226	599	194	
York University Heights	852	575	202	
Kensington-Chinatown	181	535	206	
Yonge-Bay Corridor	113	518	177	
St Lawrence-East Bayfront-The Isla..	138	500	201	
Junction-Wallace Emerson	182	379	200	
Bay-Cloverhill	551	377	189	
Humewood-Cedarvale	225	369	205	
Church-Wellesley	127	364	195	
Yorkdale-Glen Park	586	363	209	
Bayview Woods-Steeles	154	356	232	
Rosedale-Moore Park	417	352	230	
Bedford Park-Nortown	611	347	208	
Wexford/Maryvale	517	346	208	
East End Danforth	187	338	202	
Oakdale-Beverley Heights	494	329	200	

- Map of Toronto
  This map is constructed using the auto table and refined table from the Data Source tab. Lat Wgs84 and Long Wgs84 and the fields from the refined table getting used as rows and columns respectively and other fields are coming from the auto table.



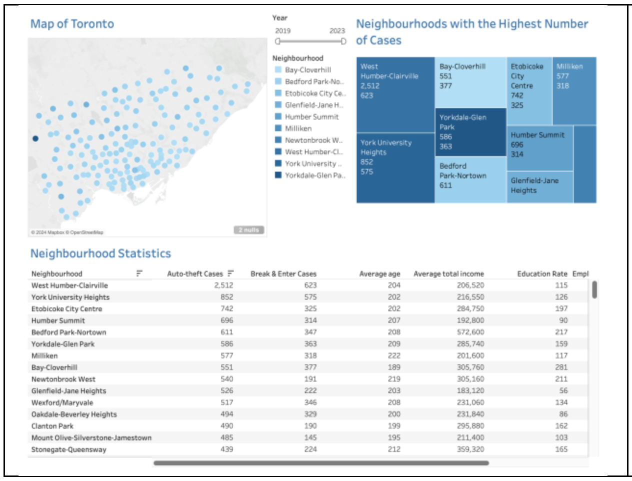
Columns: AVG(Long Wgs84)

Rows: AVG(Lat Wgs84)

Map of Toronto

5.Executive Dashboard

We have utilized Tableau to create our dashboard. We used objects such as containers to organize our visual data. Additionally, we used text boxes to display our predicted values for the number of cases in 2024. A filter for ‘year’ was added on the dashboard which is depicted on the image on the left. This filter is set to filter both the map (left) and the tree-map (right) making the dashboard more cohesive. Each dashboard created only includes three main visuals to make it easy for the executive team to follow the story and not get distracted or confused.

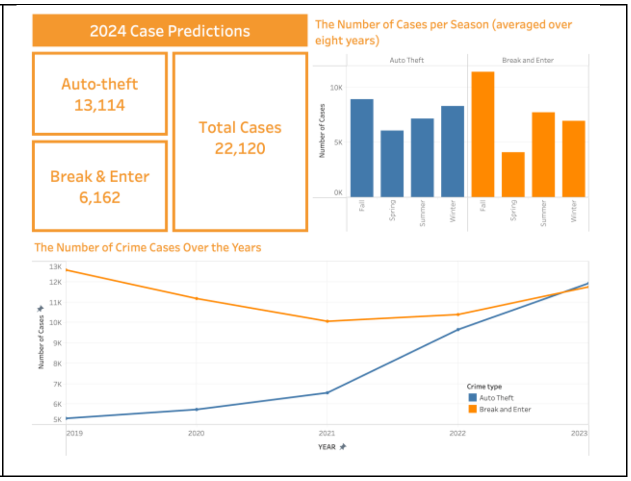


Map of Toronto

Neighbourhoods with the Highest Number of Cases

Neighbourhood Statistics

Neighbourhood	Auto-theft Cases	Break & Enter Cases	Average age	Average total income	Education Rate	Empl
West Humber-Clairville	2,512	623	204	206,520	115	
York University Heights	852	575	202	216,550	126	
Etobicoke City Centre	742	325	202	284,750	197	
Humber Summit	696	314	207	192,800	90	
Bedford Park-Nortown	611	347	208	572,600	217	
Yorkdale-Glen Park	586	363	209	285,740	159	
Milliken	577	318	222	201,600	117	
Bay-Cloverhill	551	377	189	305,760	281	
Newtonbrook West	540	191	219	305,160	211	
Glenfield-Jane Heights	526	222	203	183,120	56	
Wexford/Maryvale	517	346	208	231,060	134	
Oakdale-Beverley Heights	494	329	200	231,840	86	
Clanton Park	490	190	199	295,880	162	
Mount Olive-Silverstone-Jamestown	485	145	195	211,400	103	
Stonegate-Queensway	439	224	212	359,320	165	



2024 Case Predictions

Auto-theft
13,114

Break & Enter
6,162

Total Cases
22,120

The Number of Cases per Season (averaged over eight years)

The Number of Crime Cases Over the Years

References

Ordinary least squares regression (OLS). XLSTAT, Your data analysis solution. (n.d.). https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols