Matt Newman (S4100701)

University of Gloucestershire

CT5018 Assignment

Data Analytics

Contents

[Introduction 3](#_Toc122090558)

[Data Cleaning and Processing 3](#_Toc122090559)

[Choosing the sample 3](#_Toc122090560)

[Identifying and removing missing data items 4](#_Toc122090561)

[Dropping irrelevant data 4](#_Toc122090562)

[Cleaning outliers 4](#_Toc122090563)

[Merging the spreadsheets 5](#_Toc122090564)

[Dataset Analysis 6](#_Toc122090565)

[Mean, Median and Mode 6](#_Toc122090566)

[Standard Deviation and Variance 7](#_Toc122090567)

[Correlation 7](#_Toc122090568)

[Hypothesis Testing 8](#_Toc122090569)

[Predictive Models 9](#_Toc122090570)

[Regression 9](#_Toc122090571)

[Comparison of Techniques 12](#_Toc122090572)

[Association Rules 12](#_Toc122090573)

[Comparison of Techniques 14](#_Toc122090574)

[Conclusion 14](#_Toc122090575)

[References 16](#_Toc122090576)

[Appendices 19](#_Toc122090577)

[Appendix 1: March 2018 Boxplots Before and After Outlier Cleaning 19](#_Toc122090578)

[Appendix 2: March 2019 Boxplots Before and After Data Cleaning 24](#_Toc122090579)

[Appendix 3: Test of Linear Regression Model Upon Mean Value of Violence With Injury 29](#_Toc122090580)

[Appendix 4: Test of Quantile Regression Model Upon Mean Value of Violence With Injury 30](#_Toc122090581)

[Appendix 5: Steps Taken to Convert Numerical Data into Categorical Data for Association Rules Models 31](#_Toc122090582)

[Appendix 6: Apriori Association Rules Model 32](#_Toc122090583)

[Appendix 7: Correlation Heatmap 33](#_Toc122090584)

[Appendix 8: Covariance Heatmap 34](#_Toc122090585)

[Appendix 9: Data Cleaning for March 2018 Sample 35](#_Toc122090586)

[Appendix 10: Data Cleaning for March 2019 Sample 40](#_Toc122090587)

[Appendix 11: FP-growth Association Rules Model 45](#_Toc122090588)

[Appendix 12: High and Low Values 46](#_Toc122090589)

[Appendix 13: Linear Regression Model 48](#_Toc122090590)

[Appendix 14: Means 50](#_Toc122090591)

[Appendix 15: Medians 52](#_Toc122090592)

[Appendix 16: Merging March 2018 and March 2019 Samples 54](#_Toc122090593)

[Appendix 17: Missing Values Heatmap for March 2018 Sample 55](#_Toc122090594)

[Appendix 18: Missing Values Heatmap for March 2019 Sample 56](#_Toc122090595)

[Appendix 19: Modes 57](#_Toc122090596)

[Appendix 20: Pearson Correlation Coefficient Test for Violence with Injury and Violence without Injury 59](#_Toc122090597)

[Appendix 21: Quantile Regression Model 61](#_Toc122090598)

[Appendix 22: Standard Deviations 63](#_Toc122090599)

[Appendix 23: Variances 65](#_Toc122090600)

# Introduction

Violence against the person is a wide-ranging umbrella that encompasses several different crimes. Some of these crimes are likely to have a relationship with one another. For instance, recent ONS data suggested that homicide and knife crime could be linked, as they tended to increase in prevalence together and at similar rates (Stripe, 2021). This report aims to find which violent crimes are most strongly related. A deep analysis entailing many different techniques will be performed upon the different sub-crimes within the umbrella of violence against the person to see which crimes share the strongest relationship.

# Data Cleaning and Processing

Before analysis and testing were performed, steps were taken to clean and process the dataset.

## Choosing the sample

The first step was to choose the 24-month period to examine. For this investigation, statistics for the years ending March 2018 and March 2019 were selected. This was because the samples from 2020 onwards were likely to have been affected by the COVID-19 pandemic, when social restrictions skewed crime rates and relationships. Assault reduced by 35%, while homicide only reduced by 14% (Nivette *et al.*, 2021). This evidences that the pandemic had different effects upon different types of violent crime, so using pandemic data would have drawn potentially inaccurate conclusions about relationships. However, the pandemic may have a long-lasting effect on social contact levels and by extension, crime. A study implied that even when social curbs were relaxed, pre-COVID communication levels did not return (Liu *et al.*, 2021), which would encapsulate that any impact on crime could be long-term. Nonetheless, it is still too early to tell whether COVID-19’s effect upon violent crime will be permanent, and the most hard-hitting social curbs seen in 2020/21 are unlikely to be seen long-term. Therefore, it is best to be cautious and treat pandemic data as anomalous.

## Identifying and removing missing data items

The next data-cleaning step taken was to identify and remove missing data items in both samples.

Firstly, code generating a missing data heatmap was executed upon both spreadsheets, generating the following result:

A picture containing calendar

Description automatically generated

As displayed above, both samples had unnamed columns containing no data, which were later removed using code.

It is important to remove missing data because its presence renders parts of the dataset unavailable for analysis, therefore reducing an investigation’s statistical power and reliability (Kwak and Kim, 2017). However, removing missing data can remove a large proportion of the dataset, which could reduce an investigation’s accuracy and weigh the results more heavily toward outliers (Acock, 2005). Nonetheless, that was not a significant issue in this dataset because only two columns were missing in this case. No rows were missing, therefore negating any adverse effects.

## Dropping irrelevant data

The next step taken was to drop irrelevant data from the dataset using code. With the investigation focusing solely on violence against the person, it was decided that only “Homicide”, “Violence with injury”, “Violence without injury”, “Stalking and harassment”, and “Death or serious injury – unlawful driving” would be kept. All these crimes fall under the umbrella of violence against the person; therefore, they were relevant to the investigation.

Removing irrelevant columns was crucial because it made data processing more efficient without affecting results (Greiner, Grove and Kogan, 1997). However, removing irrelevant columns may have been unnecessary because it was possible to simply exclude them from the investigation (Li and Liu, 2017). Nonetheless, removing these columns led to more efficient data processing.

## Cleaning outliers

The next step taken was to identify and clean outliers in both samples.

Firstly, rows denoting amalgamated locations were removed because keeping them would have counted data items twice and skewed the relationships.

To identify outliers, boxplots were created for all columns in each dataset using code. Any anomalous rows found were altered using code (Singh, 2019) so that any outliers were replaced with the median value.

Boxplots before and after cleaning that were generated for each sample, showing the effect of outlier cleaning, can be found in Appendices 1 and 2.

It is important to detect and update outliers because their presence can reduce an investigation’s accuracy (Smiti, 2020). Since this dataset comes from the ONS, however, the data is unlikely to be erroneous. Therefore, outliers could have been kept, as they may have carried interesting information (Aguinis, Gottfredson and Joo, 2013). Nevertheless, this investigation focuses on overall relationships as opposed to isolated cases, so updating outliers was essential because their presence would have reduced the accuracy of the results.

Outliers were updated to the median value rather than removed because some rows only had anomalous data in some categories. Therefore, they were still useful to the investigation overall. The median was chosen over the mean because it is less affected by outliers and noise in the dataset; therefore, the confidence in its validity is higher (Pham-Gia, 2001). However, the median is also affected by bias because it may not be accurate if the dataset exhibits strong kurtosis or skewness (Pham-Gia, 2001). Nonetheless, the level of bias affecting the median is overall considerably lower for this dataset.

When the original outliers were updated, new outliers sometimes appeared due to the resulting change in standard deviation. It was chosen to keep these new outliers in place because significant alteration of data can make it hard to draw accurate conclusions (Alimohammadi and Nancy Chen, 2022). Some may argue that any outlier, whether original or found later, is still an erroneous data item that can affect the results of an investigation (Zhang, Meratnia and Havinga, 2010). Nevertheless, excessive cleaning can also adversely affect the results of an investigation, particularly when outliers found were not originally outliers.

## Merging the spreadsheets

Finally, the two cleaned datasets were each written into a new CSV before being merged. Both functions were executed using code.

The two datasets were merged to allow for the concurrent observation of more relevant data, which will always strengthen a dataset’s statistical power (Liu *et al.*, 2020). However, merging different datasets can cause issues because factors such as bias may differ between datasets (Rosenman *et al.*, 2022). The time period differed between datasets, but the remaining data was the same. Also, merging the two datasets was important in this case because it allowed for wider analysis.

# Dataset Analysis

## Mean, Median and Mode

Code was used to calculate the mean, median and mode of each category, producing the following results:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Violence without injury had the highest value for all three measures (Mean 10,670, Median 9,841, Mode 9,782 and 10,804). Therefore, violence without injury is likely to be the most prevalent violent crime within the population.

It should be noted that the median and/or mode of most categories is significantly lower than the mean. This would infer that there is skewness towards the right of the dataset in most categories (Dean and Illowsky, 2018). However, skewness is not always caused by the position of the mean, median and mode, particularly given that the mean can be strongly influenced by outliers (Leys *et al.*, 2013). Nonetheless, a recent study found that the positions of these measures were often a considerable indicator of skewness (Hartwig *et al.*, 2020).

## Standard Deviation and Variance

The standard deviation (SD) and variance of each category was calculated using code, producing the following results:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Violence without injury had the highest SD and variance (SD 5,577, Variance 31,100,572). Therefore, it can be inferred that the rates of violence without injury vary the most. Overall, all categories had very high SDs relative to the mean. Some categories, such as unlawful driving, had SDs close to 80% of the mean, and all categories had SDs over 50% of the mean. A ratio of SD to the mean can be used to measure variability; therefore, it can be ascertained that all categories in the dataset have high variability (Lin *et al.*, 2020). However, SD and variance may not be the most accurate measures to use since they can exaggerate variability, with standard error of mean (SE) potentially being a more accurate method (Barde and Barde, 2012). Nonetheless, SE is more commonly used for calculating confidence intervals, which were unnecessary for this investigation.

## Correlation

To show relationships within the dataset, heatmaps showing correlation and covariance were created using code (Zach, 2020), producing the following results:

A picture containing bar chart

Description automatically generated

Graphical user interface

Description automatically generated

The heatmaps portrayed that all categories were positively correlated with one another and had positive covariances, but some correlations and relationships were weaker than others.

Violence with injury and violence without injury appeared to share the strongest relationship. They shared a strong positive correlation of around 0.8, as well as a high positive covariance, thus indicating that they are likely to share a strong relationship (Akoglu, 2018). Therefore, it is likely to be the case that when violence with injury increases or decreases in prevalence, violence without injury will increase or decrease in prevalence alongside it. However, correlation does not always equal causation. The assumption that relationships exist based purely on a correlation is often unfounded (Nunes *et al.*, 2019). However, this dataset primarily looks at crime, so it is likely that this correlation could lead to a relationship, particularly seeing as the two crimes also shared a positive covariance. Therefore, the potential relationship between violence with injury and violence without injury warrants further investigation.

# Hypothesis Testing

A hypothesis test was conducted to test whether a correlation exists between violence with injury and violence without injury. This hypothesis was chosen because the strong positive correlation and high positive covariance found during dataset analysis indicates that a strong linear relationship between the two exists (Fisher, 1921). While a given correlation value assumes that the dataset is distributed equally (Onwuegbuzie and Daniel, 1999), which cannot be guaranteed here, this does not matter here since the hypothesis merely states the presence of a correlation.

The hypotheses identified for this test were as follows:

Graphical user interface, application

Description automatically generated

To test these hypotheses, a Pearson correlation coefficient test was conducted using code (Brownlee, 2018). This test was chosen because the relationship between the two variables is likely to be linear and the data in question is numerical (Isaac and Chikweru, 2018). One alternative test that could have been used was a Spearman’s correlation coefficient test. This could have worked well with the potential skewness of some categories. However, skewness could not be guaranteed, as a small number of extreme values existed. Also, Spearman’s coefficient is designed for use on datasets containing categorical data and non-linear relationships (Rebekić *et al.*, 2015), so it did not suit this dataset and Pearson’s coefficient was overall deemed preferable.

When conducted, the test spawned the following result:

Graphical user interface, text, application

Description automatically generated

This portrays that violence with injury and violence without injury are extremely likely to be correlated. With a highly statistically significant p-value of <0.001, the null hypothesis could be rejected. Therefore, the alternative hypothesis of a correlation existing is extremely likely to be true.

# Predictive Models

## Regression

To explore the hypothesis further, two regression models determining the effect of violence with injury upon violence without injury were implemented.

Firstly, a scatter plot to portray the relationship between the two crimes was created using code, producing the following result:

Chart, scatter chart

Description automatically generated

This evidences that there is likely to be an overall positive linear relationship between the two variables, even if some outliers exist.

Therefore, a linear regression model was selected because it uses a linear prediction model, so fitted the linear relationship shown (Su, Yan and Tsai, 2012). This model was created, fitted to the dataset, and executed using code (Stojiljkovic, 2022), producing the following result:

Chart, scatter chart

Description automatically generated

Graphical user interface, text, application

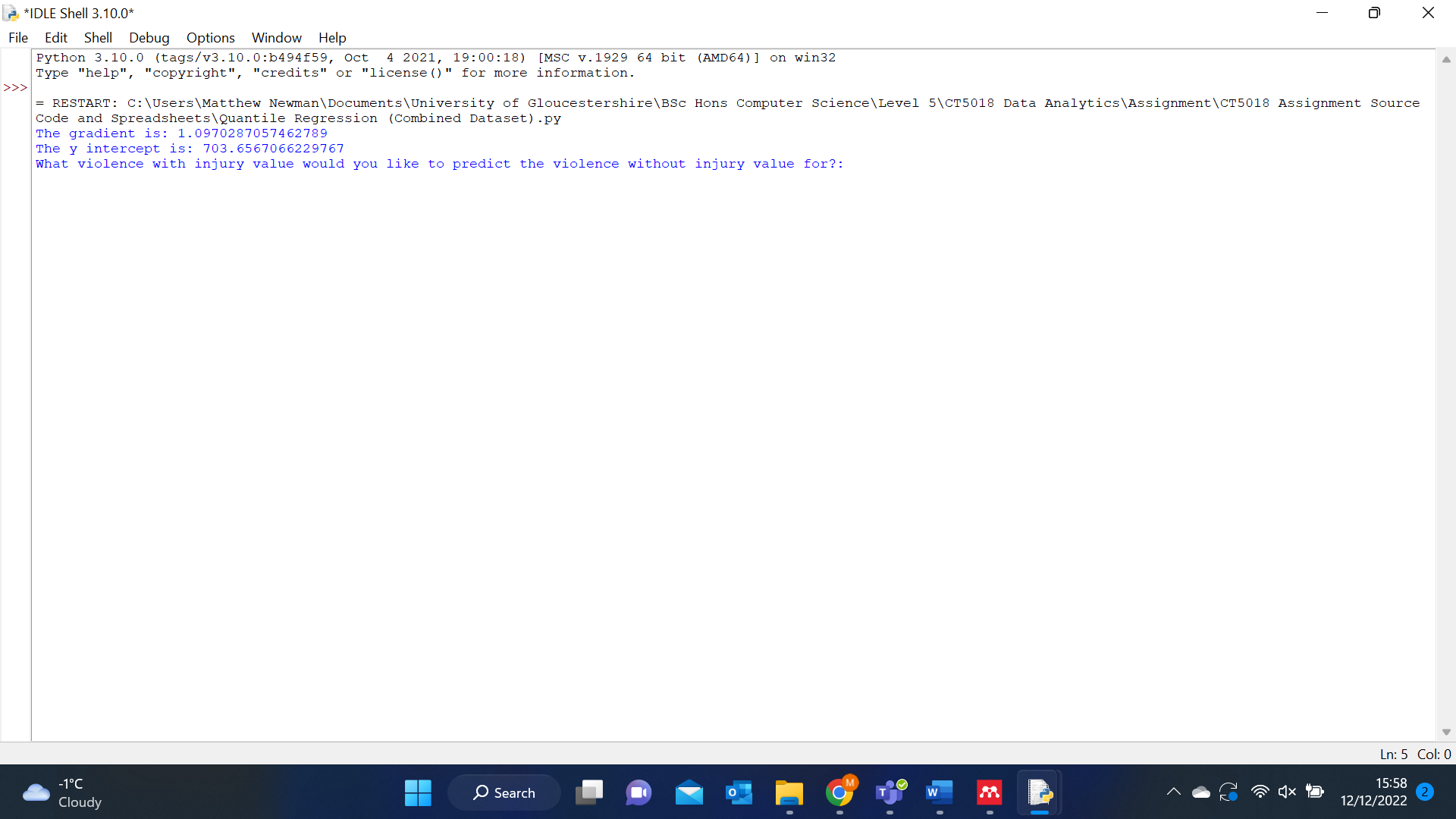
Description automatically generated

This model suggests that violence with injury and violence without injury share a linear relationship, as it offers an accurate average prediction for most of the dataset. The model implies that when violence with injury is 0, violence without injury will be approximately 2,302 on average. It is also implied that as violence with injury increases by 1, violence without injury increases by approximately 0.92 on average, which infers that it would eventually be overtaken by violence with injury.

However, the model appears inaccurate for edge cases. This could be due to the high variability that exists, particularly towards the higher end of the dataset, that could be skewing the high and low values.

As such, a second regression algorithm was applied. A quantile regression model calculating the median prediction was used because using this rather than the mean prediction results in less weighting towards extreme values (Li, 2015). This model was created using code (bhuwanesh, 2022), and produced the following results:





This portrays that when violence with injury increases by 1, violence without injury increases by approximately 1.1 on average. It also infers that when violence with injury is 0, violence without injury will be approximately 704 on average. This model fits much of the dataset, thus inferring that a strong linear relationship exists, and its accuracy also appears higher for edge cases. Therefore, it is likely to be an accurate fit for the dataset.

### Comparison of Techniques

Graphical user interface, application

Description automatically generated

(Worster, Fan and Ismaila, 2007; Tripepi *et al.*, 2008; Su, Yan and Tsai, 2012; Li, 2015; Waldmann, 2018; Ray, 2019)

## Association Rules

To test whether violence with injury and violence without injury share the strongest relationship within the dataset, two association rules algorithms were used to test which crimes most commonly fell within “average” range together. “Average” was chosen over “high” or “low” because more of the dataset will fall within “average” range, so it offered greater scope for accurate association rules to be generated. An “average” value was defined as any value falling within one standard deviation of the mean.

As association rules algorithms do not work on numerical data (Varol Altay and Alatas, 2020), the numerical data within the dataset needed to be converted into categorical data for the models to work. The steps taken to do this are shown in Appendix 5.

A model using the Apriori algorithm was firstly generated. The Apriori algorithm was used because it is a simple and effective way to derive association rules from frequent item sets (Kumbhare and Chobe, 2014).

This model was created using code, producing the following result:

Graphical user interface, text, application

Description automatically generated

This model infers that violence with injury and violence without injury could share the strongest relationship within the dataset, as the top two rules both contain violence with injury as an antecedent and violence without injury as a consequent. The lift of both rules is approximately 1.4, comfortably exceeding the benchmark lift of 1 and providing high confidence in these relationships. However, it should be noted that some outliers were replaced early in the investigation, which could have influenced the results. Nevertheless, these outliers comprised a very low percentage of the dataset, so the effect is unlikely to be significant.

A second association rules model using the FP-growth algorithm was also implemented. The FP-growth algorithm was used because it uses tree structures to form rules rather than the full database scans that the Apriori algorithm uses, so it is more efficient (Kumar and Rukmani, 2010).

The model was implemented using code (Alam, 2022), producing the following results:

Graphical user interface, text, application

Description automatically generated

The FP-growth algorithm produced slightly different results to the Apriori algorithm. It suggested that other crimes, such as homicide, may have a stronger relationship with violence without injury. With the top rule generated having a high lift of 1.4, this could suggest that stronger relationships exist. However, these results may be less accurate than those generated by the Apriori algorithm because the FP-growth algorithm scans the dataset only twice, while the Apriori algorithm scans it many times (Kavitha and Tamil Selvi, 2016).

### Comparison of Techniques

Graphical user interface, application, table

Description automatically generated with medium confidence

(Kumar and Rukmani, 2010; Al-Maolegi and Arkok, 2014; Kumbhare and Chobe, 2014; Kavitha and Tamil Selvi, 2016; Yang *et al.*, 2016)

# Conclusion

In conclusion, this investigation found that violence with injury and violence without injury are likely to be the most strongly related violent crimes. The two crimes shared the strongest positive correlation and covariance when dataset analysis was performed, and a correlation test found that a relationship was extremely likely to exist between the two. This was further reinforced when regression models were applied, which suggested that the two crimes grow together in a linear fashion. An association rules model using the Apriori algorithm reinforced that their relationship is likely to be the strongest in the dataset, as the two strongest rules generated had violence with injury as an antecedent and violence without injury as a consequent. While a model using the FP-growth algorithm generated different results, this algorithm is likely to be less accurate than the Apriori algorithm, so its results may not be reliable.

# References

Acock, A.C. (2005) ‘Working with missing values’, *Journal of Marriage and Family*, 67(4), pp. 1012–1028. Available at: https://doi.org/10.1111/j.1741-3737.2005.00191.x.

Aguinis, H., Gottfredson, R.K. and Joo, H. (2013) ‘Best-Practice Recommendations for Defining, Identifying, and Handling Outliers’, *Organizational Research Methods*. SAGE Publications Inc., pp. 270–301. Available at: https://doi.org/10.1177/1094428112470848.

Akoglu, H. (2018) ‘User’s guide to correlation coefficients’, *Turkish Journal of Emergency Medicine*, 18(3), pp. 91–93. Available at: https://doi.org/10.1016/j.tjem.2018.08.001.

Alam, B. (2022) *Implementation of FP-growth algorithm using Python*, *hands-on.cloud*. Available at: hands-on.cloud/implementation-of-fp-growth-algorithm-using-python (Accessed: 13 December 2022).

Alimohammadi, H. and Nancy Chen, S. (2022) ‘Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis’, *Expert Systems with Applications*, 191. Available at: https://doi.org/10.1016/j.eswa.2021.116371.

Al-Maolegi, M. and Arkok, B. (2014) ‘An Improved Apriori Algorithm For Association Rules’, *International Journal on Natural Language Computing*, 3(1), pp. 21–29. Available at: https://doi.org/10.5121/ijnlc.2014.3103.

Barde, P. and Barde, M. (2012) ‘What to use to express the variability of data: Standard deviation or standard error of mean?’, *Perspectives in Clinical Research*, 3(3), p. 113. Available at: https://doi.org/10.4103/2229-3485.100662.

bhuwanesh (2022) *How to Perform Quantile Regression in Python*, *GeeksForGeeks*. Available at: geeksforgeeks.org/how-to-perform-quantile-regression-in-python (Accessed: 13 December 2022).

Brownlee, J. (2018) *17 Statistical Hypothesis Tests in Python (Cheat Sheet)*, *Machine Learning Mastery*. Available at: machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet (Accessed: 24 November 2022).

Dean, S. and Illowsky, B. (2018) *Descriptive Statistics: skewness and the mean, median and mode*, *Connexions*. Available at: d1wqtxts1xzle7.cloudfront.net/40293986/descriptive-statistics-skewness-and-the-mean-median-and-mode-9-with-cover-page-v2.pdf (Accessed: 7 November 2022).

Fisher, R.A. (1921) *014: On the ‘probable error’ of a correlation coefficient deduced from a small sample*.

Greiner, R., Grove, A.J. and Kogan, A. (1997) *Knowing what doesn’t matter: exploiting the omission of irrelevant data ’*, *Artificial Intelligence*.

Hartwig, F.P. *et al.* (2020) ‘The median and the mode as robust meta-analysis estimators in the presence of small-study effects and outliers’, *Research Synthesis Methods*, 11(3), pp. 397–412. Available at: https://doi.org/10.1002/jrsm.1402.

Isaac, E. and Chikweru, E. (2018) ‘Test for Significance of Pearson’s Correlation Coefficient (r)’, *International Journal of Innovative Mathematics, Statistics & Energy Policies*, 6(1), pp. 11–23. Available at: www.seahipaj.org.

Kavitha, M.M. and Tamil Selvi, S.T. (2016) ‘Comparative Study on Apriori Algorithm and Fp Growth Algorithm with Pros and Cons’, *International Journal of Computer Science Trends and Technology (IJCS T)*, 4(4). Available at: www.ijcstjournal.org.

Kumar, S.B. and Rukmani, K. v (2010) ‘Implementation of web usage mining using APRIORI and FP growth algorithms’, *International Journal of Advanced Networking and Applications*, 1(6), pp. 400–404. Available at: https://www.researchgate.net/publication/266012367.

Kumbhare, T.A. and Chobe, S. v (2014) ‘An Overview of Association Rule Mining Algorithms’, *International Journal of Computer Science and Information Technologies*, 5(1), pp. 927–930. Available at: www.ijcsit.com.

Kwak, S.K. and Kim, J.H. (2017) ‘Statistical data preparation: Management of missing values and outliers’, *Korean Journal of Anesthesiology*, 70(4), pp. 407–411. Available at: https://doi.org/10.4097/kjae.2017.70.4.407.

Leys, C. *et al.* (2013) ‘Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median’, *Journal of Experimental Social Psychology*, 49(4), pp. 764–766. Available at: https://doi.org/10.1016/j.jesp.2013.03.013.

Li, J. and Liu, H. (2017) ‘Challenges of Feature Selection for Big Data Analytics’, *IEEE Intelligent Systems*, 32(2), pp. 9–15. Available at: https://doi.org/10.1109/MIS.2017.38.

Li, M. (2015) ‘Moving Beyond the Linear Regression Model: Advantages of the Quantile Regression Model’, *Journal of Management*, 41(1), pp. 71–98. Available at: https://doi.org/10.1177/0149206314551963.

Lin, C. *et al.* (2020) ‘Application of a combined standard deviation and mean based approach to MOPITT CO column data, and resulting improved representation of biomass burning and urban air pollution sources’, *Remote Sensing of Environment*, 241. Available at: https://doi.org/10.1016/j.rse.2020.111720.

Liu, C.Y. *et al.* (2021) ‘Rapid Review of Social Contact Patterns during the COVID-19 Pandemic’, *Epidemiology*, 32(6), pp. 781–791. Available at: https://doi.org/10.1097/EDE.0000000000001412.

Liu, R. *et al.* (2020) ‘Discovering and merging related analytic datasets’, *Information Systems*, 91. Available at: https://doi.org/10.1016/j.is.2020.101495.

Nivette, A.E. *et al.* (2021) ‘A global analysis of the impact of COVID-19 stay-at-home restrictions on crime’, *Nature Human Behaviour*, 5(7), pp. 868–877. Available at: https://doi.org/10.1038/s41562-021-01139-z.

Nunes, K.L. *et al.* (2019) ‘“I Know Correlation Doesn’t Prove Causation, but..”: Are We Jumping to Unfounded Conclusions About the Causes of Sexual Offending?’, *Sexual Abuse: Journal of Research and Treatment*, 31(2), pp. 220–236. Available at: https://doi.org/10.1177/1079063217729156.

Onwuegbuzie, A.J. and Daniel, L.G. (1999) ‘Uses and Misuses of the Correlation Coefficient’, in *Annual Meeting of the Mid-South Educational Research Association*.

Pham-Gia, T. (2001) ‘The Mean and Median Absolute Deviations’, *Mathematical and Computer Modelling*, 34, pp. 921–936. Available at: www.elsevier.nl/locate/mcm.

Ray, S. (2019) ‘A Quick Review of Machine Learning Algorithms’, in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing*. Faridabad, India: IEEE, pp. 35–39.

Rebekić, A. *et al.* (2015) ‘Pearson’s or Spearman’s correlation coefficient - which one to use?’, *Poljoprivreda*, 21(2), pp. 47–54. Available at: https://doi.org/10.18047/poljo.21.2.8.

Rosenman, E.T.R. *et al.* (2022) ‘Propensity score methods for merging observational and experimental datasets’, *Statistics in Medicine*, 41(1), pp. 65–86. Available at: https://doi.org/10.1002/sim.9223.

Singh, D. (2019) *Cleaning up Data from Outliers*, *PluralSight*. Available at: pluralsight.com/guides/cleaning-up-data-from-outliers (Accessed: 27 October 2022).

Smiti, A. (2020) ‘A critical overview of outlier detection methods’, *Computer Science Review*, 38. Available at: https://doi.org/10.1016/j.cosrev.2020.100306.

Stojiljkovic, M. (2022) *Linear Regression in Python*, *RealPython*. Available at: realpython.com/linear-regression-in-python/ (Accessed: 22 November 2022).

Stripe, N. (2021) *The nature of violent crime in England and Wales (Year ending March 2020)*, *Office for National Statistics*. Available at: ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/articles/thenatureofviolentcrimeinenglandandwales/yearendingmarch2020 (Accessed: 27 October 2022).

Su, X., Yan, X. and Tsai, C.L. (2012) ‘Linear regression’, *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3), pp. 275–294. Available at: https://doi.org/10.1002/wics.1198.

Tripepi, G. *et al.* (2008) ‘Linear and logistic regression analysis’, *Kidney International*, 73(7), pp. 806–810. Available at: https://doi.org/10.1038/sj.ki.5002787.

Varol Altay, E. and Alatas, B. (2020) ‘Intelligent optimization algorithms for the problem of mining numerical association rules’, *Physica A: Statistical Mechanics and its Applications*, 540. Available at: https://doi.org/10.1016/j.physa.2019.123142.

Waldmann, E. (2018) ‘Quantile regression: A short story on how and why’, *Statistical Modelling*, 18(3–4), pp. 203–218. Available at: https://doi.org/10.1177/1471082X18759142.

Worster, A., Fan, J. and Ismaila, A. (2007) ‘Understanding linear and logistic regression analyses’, *Canadian Journal of Emergency Medicine*, 9(2), pp. 111–113. Available at: https://doi.org/10.1017/S1481803500014883.

Yang, T. *et al.* (2016) ‘Improve the Prediction Accuracy of Naïve Bayes Classifier with Association Rule Mining’, in *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, pp. 129–133. Available at: https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.38.

Zach (2020) *How to Create a Covariance Matrix in Python*, *Statology*. Available at: statology.org/covariance-matrix-python (Accessed: 24 November 2022).

Zhang, Y., Meratnia, N. and Havinga, P. (2010) ‘Outlier detection techniques for wireless sensor networks: A survey’, *IEEE Communications Surveys and Tutorials*, 12(2), pp. 159–170. Available at: https://doi.org/10.1109/SURV.2010.021510.00088.

# Appendices

## Appendix 1: March 2018 Boxplots Before and After Outlier Cleaning

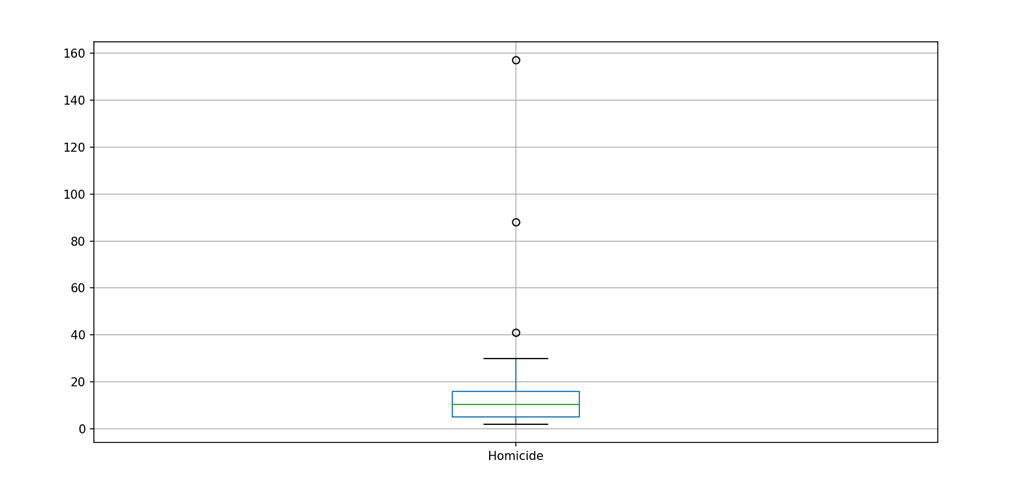


Figure : Homicide before outlier cleaning

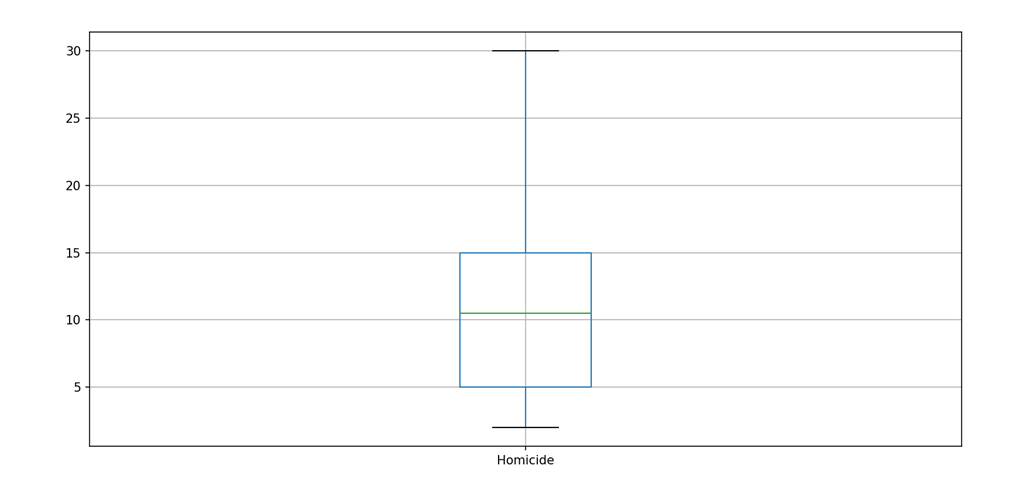


Figure : Homicide after outlier cleaning

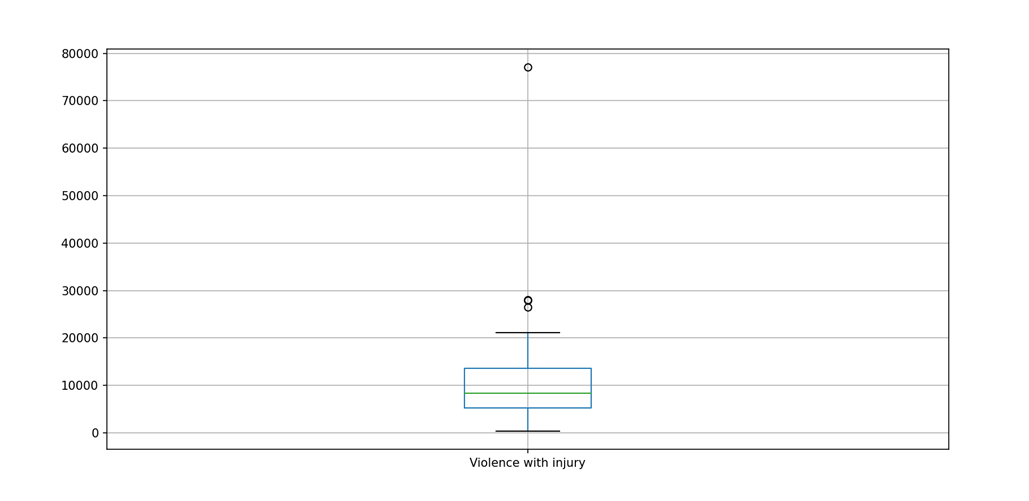


Figure : Violence with injury before outlier cleaning

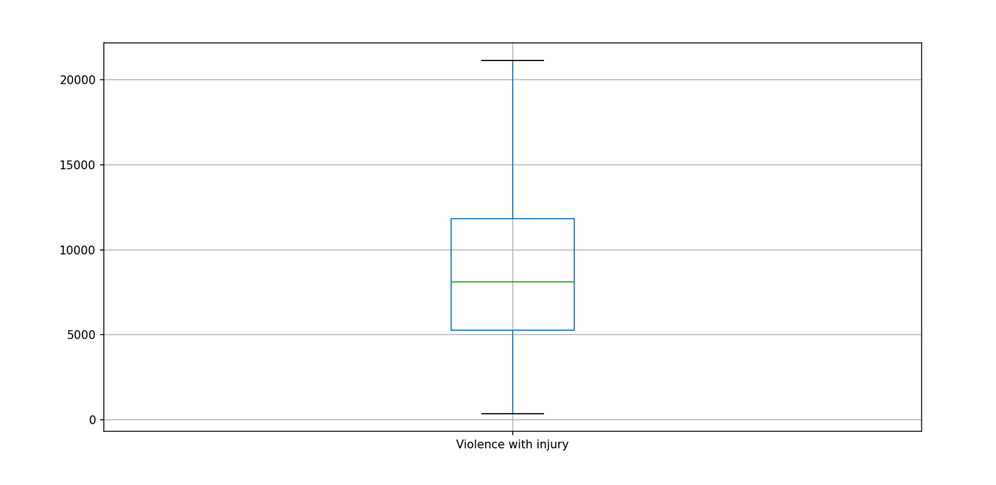


Figure : Violence with injury after outlier cleaning

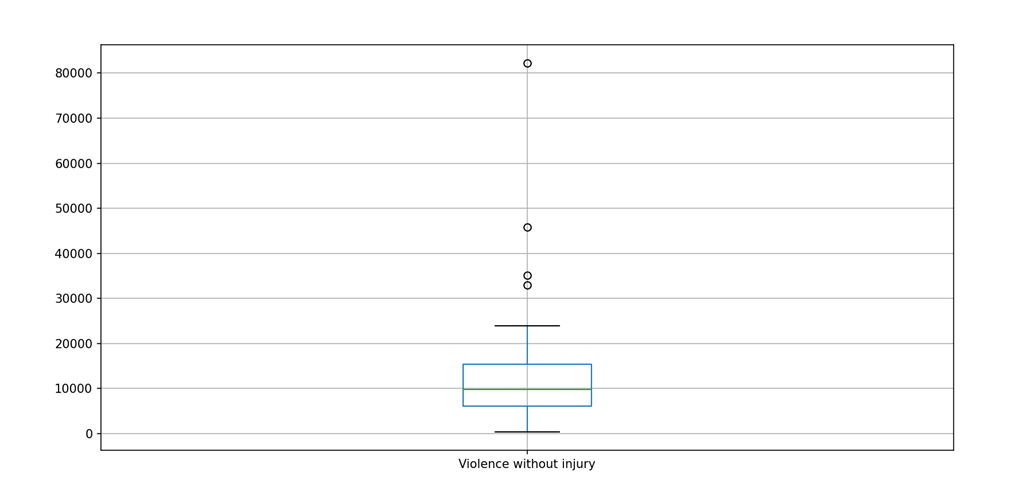


Figure : Violence without injury before outlier cleaning

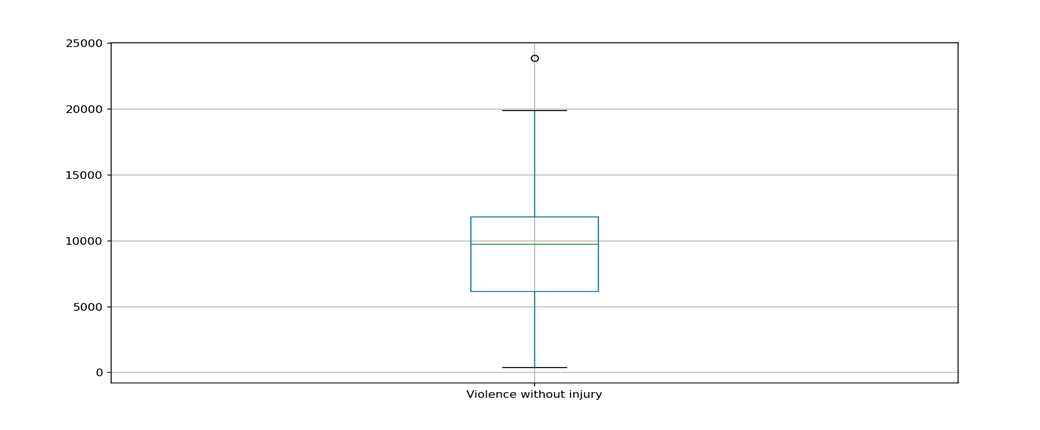


Figure : Violence without injury after outlier cleaning

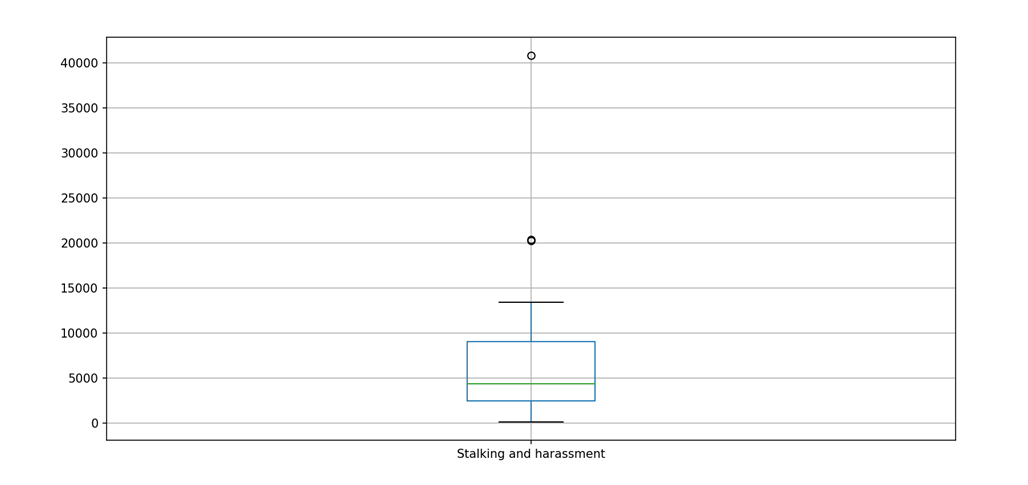


Figure : Stalking and harassment before outlier cleaning

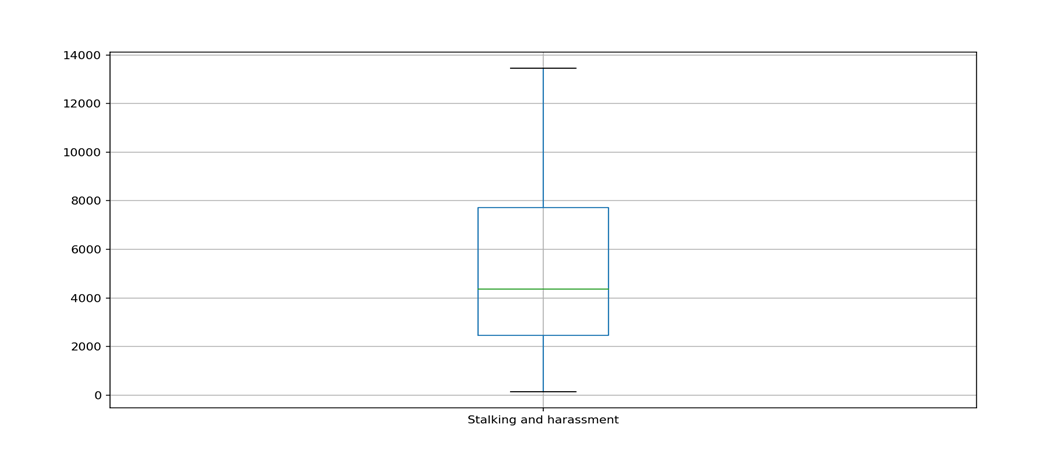


Figure : Stalking and harassment after outlier cleaning

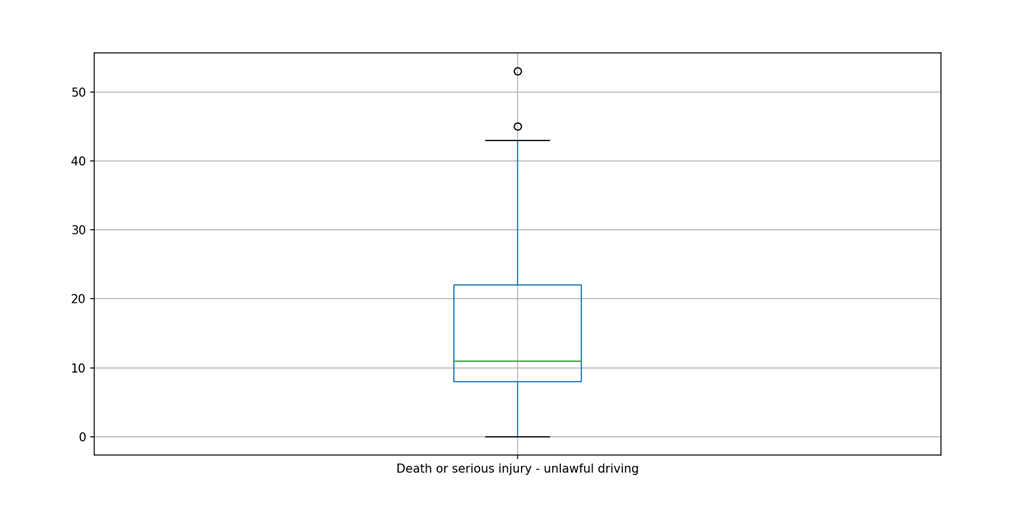


Figure : Unlawful driving before outlier cleaning

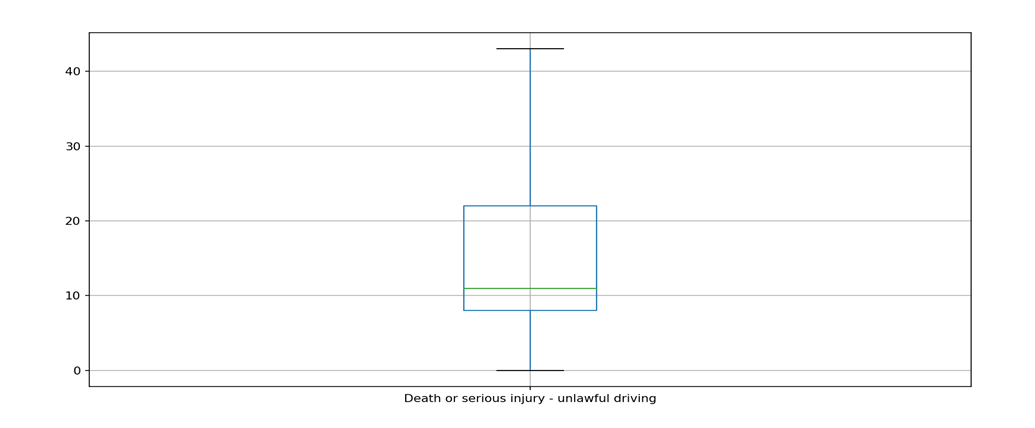


Figure : Unlawful driving after outlier cleaning

## Appendix 2: March 2019 Boxplots Before and After Data Cleaning

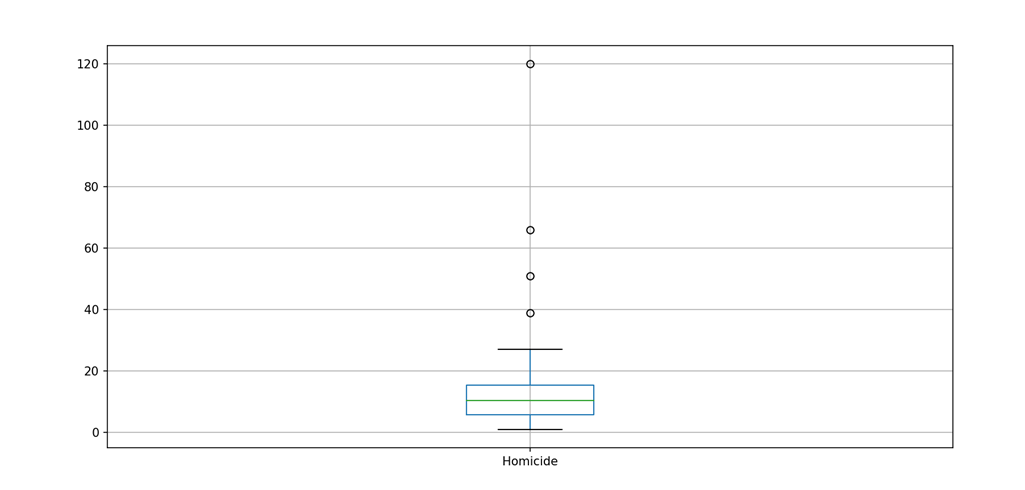


Figure : Homicide before outlier cleaning

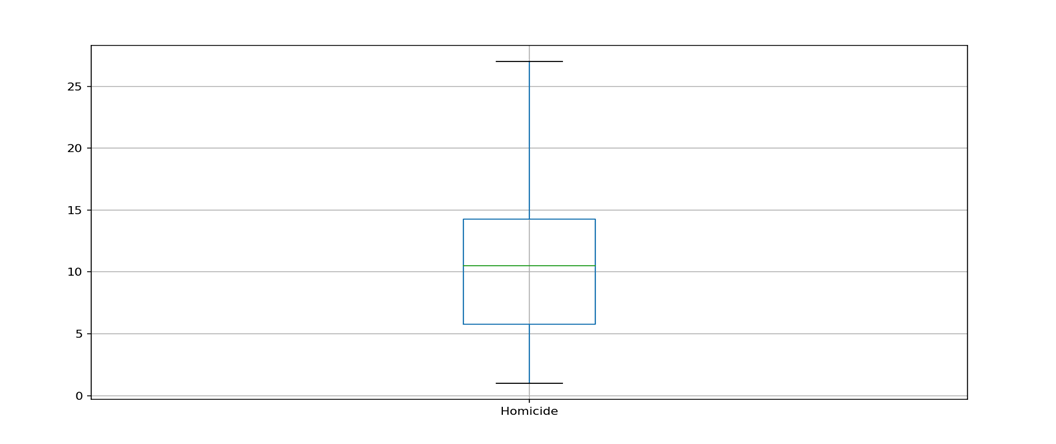


Figure : Homicide after outlier cleaning

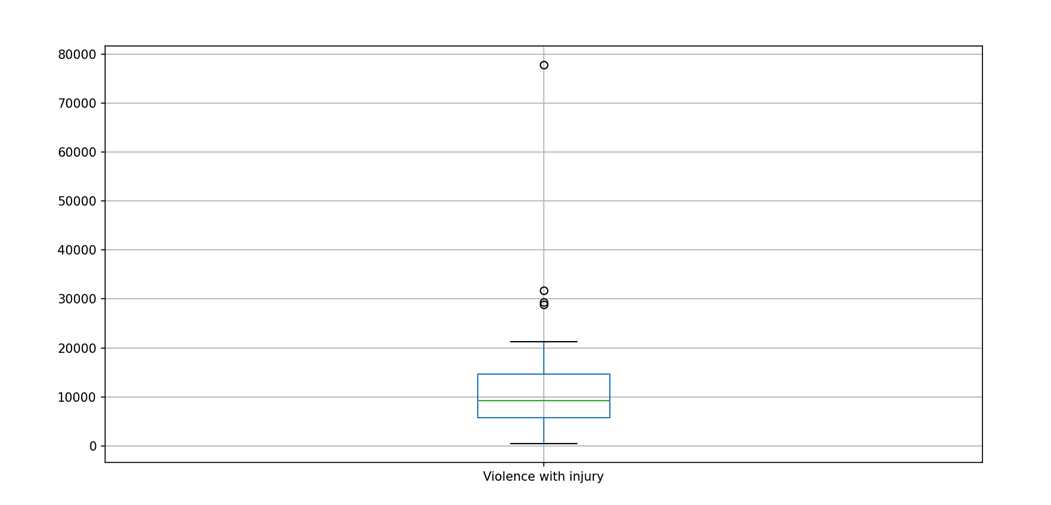


Figure : Violence with injury before outlier cleaning

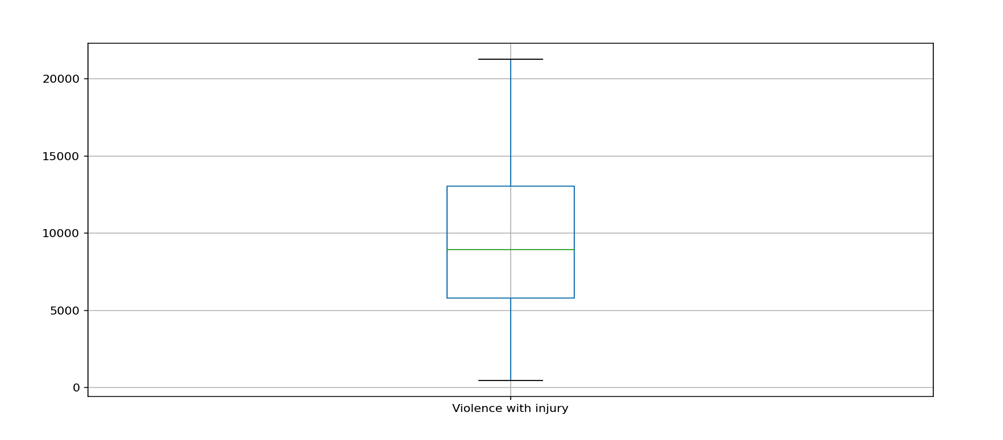


Figure : Violence with injury after outlier cleaning

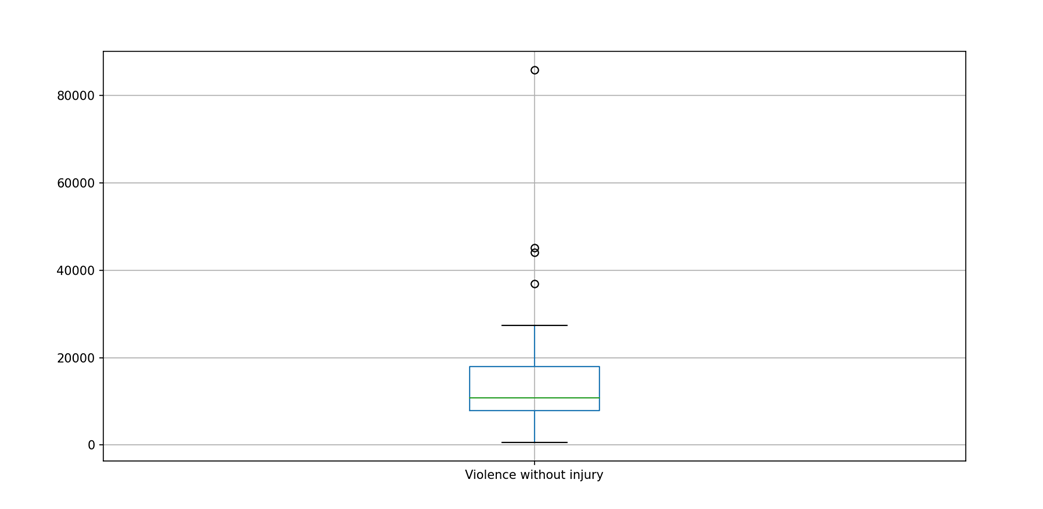


Figure : Violence without injury before outlier cleaning

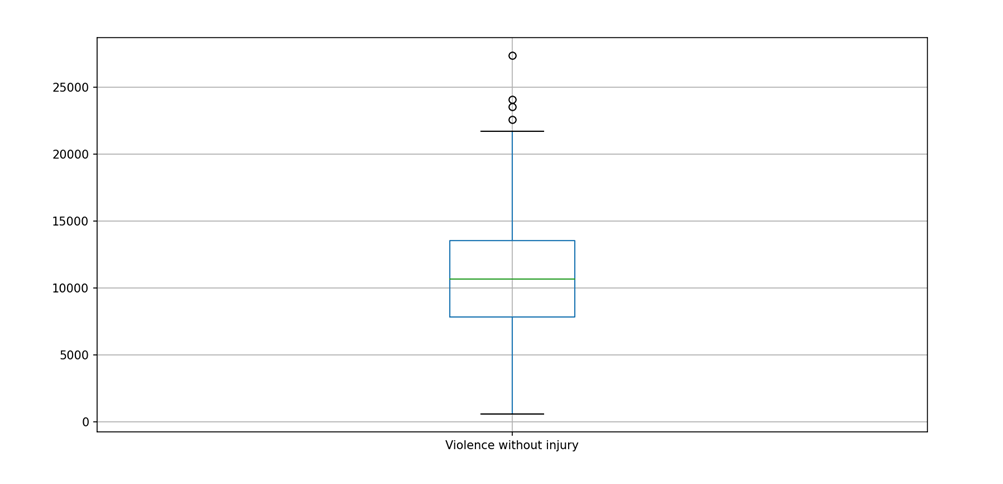


Figure : Violence without injury after outlier cleaning

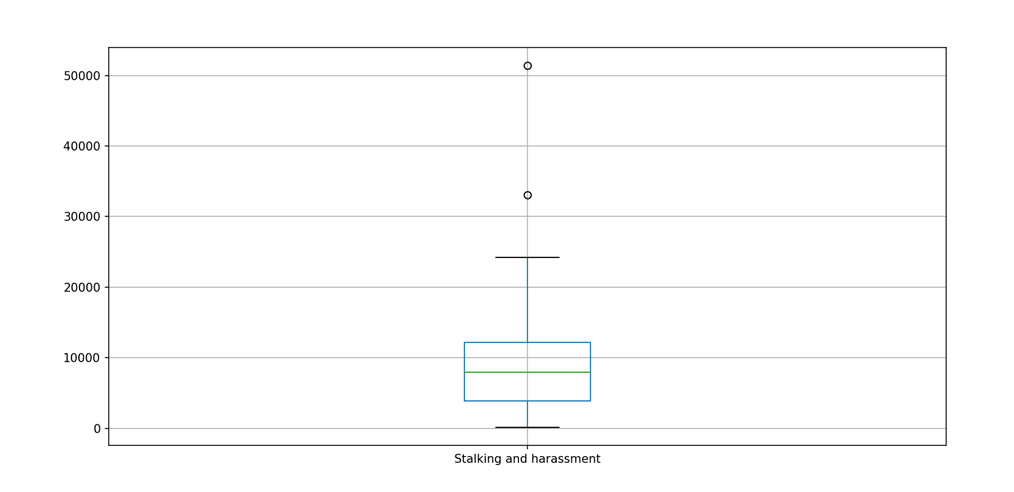


Figure : Stalking and harassment before outlier cleaning

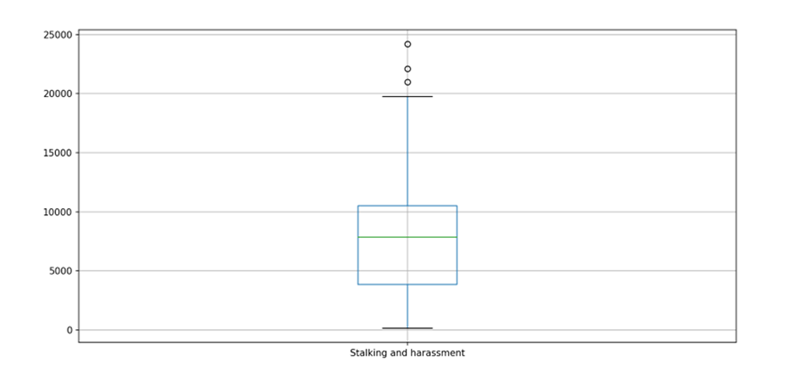


Figure : Stalking and harassment after outlier cleaning

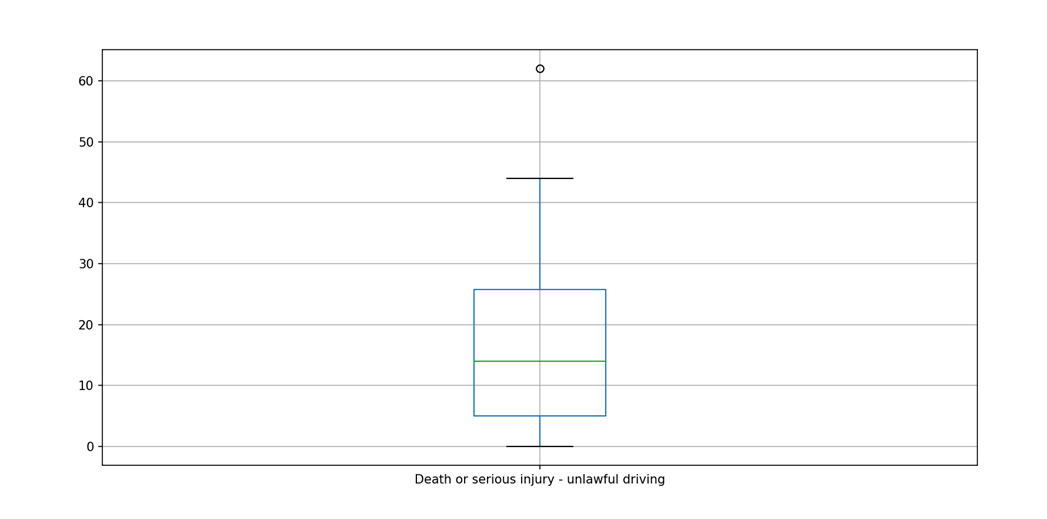


Figure : Unlawful driving before outlier cleaning

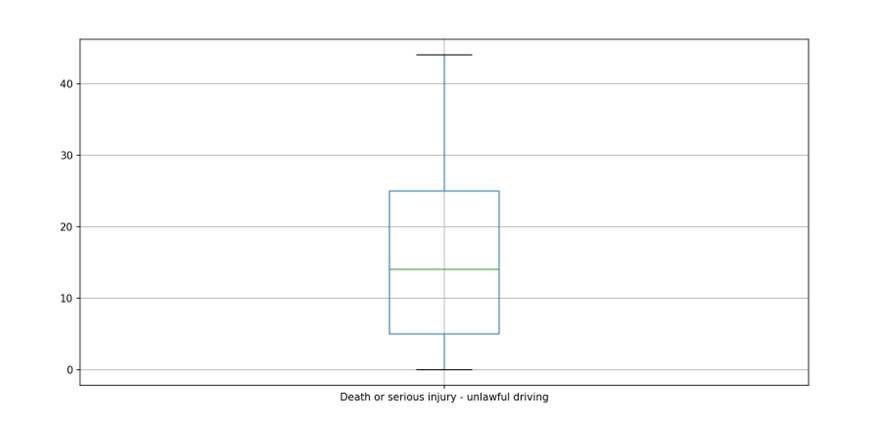


Figure : Unlawful driving after outlier cleaning

## Appendix 3: Test of Linear Regression Model Upon Mean Value of Violence With Injury

When the linear regression model was tested upon the mean violence with injury value of 9,116, the result was as follows:

Graphical user interface, text, application

Description automatically generated

The violence without injury value generated by the model matched that of the dataset to the nearest whole number (both values are 10,670), so the model is likely to be very accurate for much of the dataset.

## Appendix 4: Test of Quantile Regression Model Upon Mean Value of Violence With Injury

When the quantile regression model was tested upon the mean violence with injury value of 9,116, the result was as follows:

Graphical user interface, text, application

Description automatically generated

The violence without injury value generated matched the mean value (10,670) to three significant figures. The value generated was 10,704, therefore both values are approximately 10,700 to three significant figures. This would suggest that the quantile regression model is a fairly accurate match for the dataset.

## Appendix 5: Steps Taken to Convert Numerical Data into Categorical Data for Association Rules Models

To convert the numerical data into categorical data, a “high” and “low” value for each was determined. As an “average” value was denoted as any value that falls within one standard deviation of the mean, a “high” value was denoted as any value more than one standard deviation above the mean, and a “low” value was denoted as any value more than one standard deviation below the mean. The minimum “high” value and the maximum “low” value for each category were generated using code, producing the following result:

Graphical user interface, text, application

Description automatically generated

As such:

* An “average” value for homicide was denoted as any value sitting between 4 and 17.
* An “average” value for violence with injury was denoted as any value sitting between 4,420 and 13,811.
* An “average” value for violence without injury was denoted as any value sitting between 5,093 and 16,246.
* An “average” value for stalking and harassment was denoted as any value sitting between 1,728 and 11,971.
* An “average” value for unlawful driving was denoted as any value sitting between 4 and 27.

With these values in mind, the dataset was saved into a separate CSV file (to protect the existing numerical data) before being converted into binary, with an “average” value being denoted as 1 and a “high” or “low” value being denoted as 0.

## Appendix 6: Apriori Association Rules Model

#Importing modules

import numpy as np

import pandas as pd

from mlxtend.frequent\_patterns import apriori, association\_rules

#Reading CSV

DataFrame = pd.read\_csv('CT5018 Association Rules Dataset.csv')

#Building the association rules model

FrequentItems = apriori(DataFrame, min\_support = 0.1, use\_colnames = True)

#Collecting the association rules

AssociationRules = association\_rules(FrequentItems, metric ="lift", min\_threshold = 1)

AssociationRules = AssociationRules.sort\_values(['confidence', 'lift'], ascending =[False, False])

#Filtering the association rules so that only those with a confidence of more than 60% and a lift of more than 1 show

FilteredRules = AssociationRules[(AssociationRules['confidence']>0.60) & (AssociationRules['lift']>1)]

#Ensuring that all columns are shown when association rules are printed

pd.pandas.set\_option('display.max\_columns', None)

#Cleaning the results

print(FilteredRules[['antecedents', 'consequents', 'lift']])

## Appendix 7: Correlation Heatmap

#Importing modules

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from pandas.plotting import scatter\_matrix

import seaborn as sns

#Reading CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Constructing heatmap

sns.heatmap(DataFrame.corr(), cmap="RdYlGn", linewidths=0.30)

#Showing heatmap

plt.show()

## Appendix 8: Covariance Heatmap

#Importing modules

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from pandas.plotting import scatter\_matrix

import seaborn as sns

#Reading CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Calculating covariance between data categories

Covariance = DataFrame.cov()

#Defining labels

Labels = ["Homicide", "Violence with injury", "Violence without injury", "Stalking and harassment", "Death and serious injury - Unlawful driving"]

#Constructing heatmap

sns.heatmap(Covariance, annot=True, fmt="g", xticklabels=Labels, yticklabels=Labels)

#Showing heatmap

plt.show()

## Appendix 9: Data Cleaning for March 2018 Sample

#Importing modules

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import matplotlib.mlab as mlab

import matplotlib

from matplotlib.pyplot import figure

#Defining missing values

MissingValues = ["n/a", "na", "?", ".."]

#Opening and reading CSV

DataFrame = pd.read\_csv("policeforceareatablesyearendingmarch2018v2.csv", na\_values=MissingValues)

#Displaying number of rows and columns within CSV

print("Shape: ",DataFrame.shape)

#Specifying empty columns to drop

ColumnsToDrop = ["Unnamed: 3","Unnamed: 22"]

#Dropping empty columns

DataFrame = DataFrame.drop(ColumnsToDrop, axis=1)

#Displaying number of rows and columns within CSV after cleaning for confirmation of successful missing data deletion

print("Shape: ",DataFrame.shape)

#Identifying amalgamated rows for wider areas to be dropped, as keeping these would make each area's data be counted twice and skew the spread of data

RowsToDrop = [0,1,2,6,12,17,23,28,35,38,44,50]

#Dropping amalgamated rows

DataFrame = DataFrame.drop(RowsToDrop, axis=0)

#Displaying number of rows and columns within CSV after cleaning for confirmation of successful amalgamated row deletion

print("Shape: ",DataFrame.shape)

#Creating boxplot of homicide

DataFrame.boxplot(column=["Homicide"])

#Showing boxplot

plt.show()

#Creating boxplot of violence with injury

DataFrame.boxplot(column=["Violence with injury"])

#Showing boxplot

plt.show()

#Creating boxplot of violence without injury

DataFrame.boxplot(column=["Violence without injury"])

#Showing boxplot

plt.show()

#Creating boxplot of stalking and harassment

DataFrame.boxplot(column=["Stalking and harassment"])

#Showing boxplot

plt.show()

#Creating boxplot of death and injury via unlawful driving

DataFrame.boxplot(column=["Death or serious injury - unlawful driving"])

#Showing boxplot

plt.show()

#Showing median value of homicide

print(DataFrame["Homicide"].quantile(0.50))

#Replacing outliers with median value of homicide

DataFrame["Homicide"] = np.where(DataFrame["Homicide"] > 40, 11, DataFrame["Homicide"])

#Creating new boxplot of homicide to confirm removal of original outliers

DataFrame.boxplot(column=["Homicide"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Violence with injury"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Violence with injury"] = np.where(DataFrame["Violence with injury"] > 26525, 8339, DataFrame["Violence with injury"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Violence with injury"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Violence without injury"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Violence without injury"] = np.where(DataFrame["Violence without injury"] > 33001, 9782, DataFrame["Violence without injury"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Violence without injury"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Stalking and harassment"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Stalking and harassment"] = np.where(DataFrame["Stalking and harassment"] > 20232, 4369, DataFrame["Stalking and harassment"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Stalking and harassment"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Death or serious injury - unlawful driving"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Death or serious injury - unlawful driving"] = np.where(DataFrame["Death or serious injury - unlawful driving"] > 44, 11, DataFrame["Death or serious injury - unlawful driving"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Death or serious injury - unlawful driving"])

#Showing new boxplot

plt.show()

#Confirming rows and columns before dropping irrelevant data

print("Shape: ",DataFrame.shape)

#Identifying irrelevant columns to drop

IrrelevantColumns = ["Area Code", "Area Name", "Unnamed: 2", "Violence against the person", "Sexual offences", "Robbery", "Theft offences", "Burglary", "Residential burglary", "Non-residential burglary", "Vehicle offences", "Theft from the person", "Bicycle theft", "Shoplifting", "Other theft offences", "Criminal damage and arson", "Drug offences", "Possession of weapons offences", "Public order offences", "Miscellaneous crimes"]

#Dropping irrelevant columns

DataFrame = DataFrame.drop(IrrelevantColumns, axis=1)

#Confirming rows and columns to confirm that irrelevant data has been removed

print("Shape: ",DataFrame.shape)

#Writing cleaned dataset to new CSV

DataFrame.to\_csv("March 2018 Cleaned Dataset.csv")

## Appendix 10: Data Cleaning for March 2019 Sample

#Importing modules

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import matplotlib.mlab as mlab

import matplotlib

from matplotlib.pyplot import figure

#Defining missing values

MissingValues = ["n/a", "na", "?", ".."]

#Opening and reading CSV

DataFrame = pd.read\_csv("policeforceareatablesyeendingmarch2019.csv", na\_values=MissingValues)

#Displaying number of rows and columns within CSV

print("Shape: ",DataFrame.shape)

#Specifying empty columns to drop

ColumnsToDrop = ["Unnamed: 3","Unnamed: 22"]

#Dropping empty columns

DataFrame = DataFrame.drop(ColumnsToDrop, axis=1)

#Displaying number of rows and columns within CSV after cleaning for confirmation of successful missing data deletion

print("Shape: ",DataFrame.shape)

#Identifying amalgamated rows for wider areas to be dropped, as keeping these would make each area's data be counted twice and skew the spread of data

RowsToDrop = [0,1,2,6,12,17,23,28,35,38,44,50]

#Dropping amalgamated rows

DataFrame = DataFrame.drop(RowsToDrop, axis=0)

#Displaying number of rows and columns within CSV after cleaning for confirmation of successful amalgamated row deletion

print("Shape: ",DataFrame.shape)

#Creating boxplot of homicide

DataFrame.boxplot(column=["Homicide"])

#Showing boxplot

plt.show()

#Creating boxplot of violence with injury

DataFrame.boxplot(column=["Violence with injury"])

#Showing boxplot

plt.show()

#Creating boxplot of violence without injury

DataFrame.boxplot(column=["Violence without injury"])

#Showing boxplot

plt.show()

#Creating boxplot of stalking and harassment

DataFrame.boxplot(column=["Stalking and harassment"])

#Showing boxplot

plt.show()

#Creating boxplot of death and injury via unlawful driving

DataFrame.boxplot(column=["Death or serious injury - unlawful driving"])

#Showing boxplot

plt.show()

#Showing median value of homicide

print(DataFrame["Homicide"].quantile(0.50))

#Replacing outliers with median value of homicide

DataFrame["Homicide"] = np.where(DataFrame["Homicide"] > 38, 11, DataFrame["Homicide"])

#Creating new boxplot of homicide to confirm removal of original outliers

DataFrame.boxplot(column=["Homicide"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Violence with injury"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Violence with injury"] = np.where(DataFrame["Violence with injury"] > 28828, 9207, DataFrame["Violence with injury"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Violence with injury"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Violence without injury"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Violence without injury"] = np.where(DataFrame["Violence without injury"] > 36917, 10804, DataFrame["Violence without injury"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Violence without injury"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Stalking and harassment"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Stalking and harassment"] = np.where(DataFrame["Stalking and harassment"] > 33095, 7925, DataFrame["Stalking and harassment"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Stalking and harassment"])

#Showing new boxplot

plt.show()

#Showing median value of violence with injury

print(DataFrame["Death or serious injury - unlawful driving"].quantile(0.50))

#Replacing outliers with median value of violence with injury

DataFrame["Death or serious injury - unlawful driving"] = np.where(DataFrame["Death or serious injury - unlawful driving"] > 61, 14, DataFrame["Death or serious injury - unlawful driving"])

#Creating new boxplot of violence with injury to confirm removal of original outliers

DataFrame.boxplot(column=["Death or serious injury - unlawful driving"])

#Showing new boxplot

plt.show()

#Confirming rows and columns before dropping irrelevant data

print("Shape: ",DataFrame.shape)

#Identifying irrelevant columns to drop

IrrelevantColumns = ["Area Code", "Area Name", "Unnamed: 2", "Violence against the person", "Sexual offences", "Robbery", "Theft offences", "Burglary", "Residential burglary", "Non-residential burglary", "Vehicle offences", "Theft from the person", "Bicycle theft", "Shoplifting", "Other theft offences", "Criminal damage and arson", "Drug offences", "Possession of weapons offences", "Public order offences", "Miscellaneous crimes"]

#Dropping irrelevant columns

DataFrame = DataFrame.drop(IrrelevantColumns, axis=1)

#Confirming rows and columns to confirm that irrelevant data has been removed

print("Shape: ",DataFrame.shape)

#Writing cleaned dataset to new CSV

DataFrame.to\_csv("March 2019 Cleaned Dataset.csv")

## Appendix 11: FP-growth Association Rules Model

#Importing modules

import numpy as np

import pandas as pd

from mlxtend.frequent\_patterns import fpgrowth, association\_rules

#Reading CSV

DataFrame = pd.read\_csv('CT5018 Association Rules Dataset.csv')

#Building the association rules model

FrequentItems = fpgrowth(DataFrame, min\_support = 0.1, use\_colnames = True)

#Collecting the association rules

AssociationRules = association\_rules(FrequentItems, metric ="lift", min\_threshold = 1)

AssociationRules = AssociationRules.sort\_values(['confidence', 'lift'], ascending =[False, False])

#Filtering the association rules so that only those with a confidence of more than 60% and a lift of more than 1 show

FilteredRules = AssociationRules[(AssociationRules['confidence']>0.60) & (AssociationRules['lift']>1)]

#Ensuring that all columns are shown when association rules are printed

pd.pandas.set\_option('display.max\_columns', None)

#Cleaning the results

print(FilteredRules[['antecedents', 'consequents', 'lift']])

## Appendix 12: High and Low Values

#Importing libraries

import numpy as np

import pandas as pd

#Reading CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Determining and printing homicide high value

HomicideHigh = (DataFrame["Homicide"].mean() + DataFrame["Homicide"].std())

print("High value of homicide:",HomicideHigh)

#Determining and printing homicide low value

HomicideLow = (DataFrame["Homicide"].mean() - DataFrame["Homicide"].std())

print("Low value of homicide:",HomicideLow)

#Determining and printing violence with injury high value

ViolenceWithInjuryHigh = (DataFrame["Violence with injury"].mean() + DataFrame["Violence with injury"].std())

print("High value of violence with injury:",ViolenceWithInjuryHigh)

#Determining and printing violence with injury low value

ViolenceWithInjuryLow = (DataFrame["Violence with injury"].mean() - DataFrame["Violence with injury"].std())

print("Low value of violence with injury:",ViolenceWithInjuryLow)

#Determining and printing violence without injury high value

ViolenceWithoutInjuryHigh = (DataFrame["Violence without injury"].mean() + DataFrame["Violence without injury"].std())

print("High value of violence without injury:",ViolenceWithoutInjuryHigh)

#Determining and printing violence without injury low value

ViolenceWithoutInjuryLow = (DataFrame["Violence without injury"].mean() - DataFrame["Violence without injury"].std())

print("Low value of violence without injury:",ViolenceWithoutInjuryLow)

#Determining and printing stalking and harassment high value

StalkingHigh = (DataFrame["Stalking and harassment"].mean() + DataFrame["Stalking and harassment"].std())

print("High value of stalking and harassment:",StalkingHigh)

#Determining and printing stalking and harassment low value

StalkingLow = (DataFrame["Stalking and harassment"].mean() - DataFrame["Stalking and harassment"].std())

print("Low value of stalking and harassment:",StalkingLow)

#Determining and printing unlawful driving high value

UnlawfulDrivingHigh = (DataFrame["Death or serious injury - unlawful driving"].mean() + DataFrame["Death or serious injury - unlawful driving"].std())

print("High value of unlawful driving:",UnlawfulDrivingHigh)

#Determining and printing unlawful driving low value

UnlawfulDrivingLow = (DataFrame["Death or serious injury - unlawful driving"].mean() - DataFrame["Death or serious injury - unlawful driving"].std())

print("Low value of unlawful driving:",UnlawfulDrivingLow)

## Appendix 13: Linear Regression Model

#Importing modules

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import LinearRegression

from scipy.stats import linregress

#Reading the CSV

DataFrame = pd.read\_csv('CT5018 Combined Assignment Dataset.csv', index\_col=0)

#Plotting a scatter plot, with Violence with injury on the x axis and Violence without injury on the y axis

DataFrame.plot(kind='scatter', x='Violence with injury', y='Violence without injury')

#Showing scatter plot

plt.show()

#Defining x axis of linear regression model and reshaping to fit model

ViolenceWithInjury = DataFrame["Violence with injury"].values.reshape(-1,1)

#Defining y axis of linear regression model

ViolenceWithoutInjury = DataFrame["Violence without injury"]

#Creating fitted linear regression model

LinearRegressionModel = LinearRegression().fit(ViolenceWithInjury,ViolenceWithoutInjury)

#Printing gradient of linear regression line

Gradient = LinearRegressionModel.coef\_

print("The gradient is:",Gradient)

#Printing y intercept of linear regression line

YIntercept = LinearRegressionModel.intercept\_

print("The y intercept is:",YIntercept)

#Generating a MatPlotLib figure to visualise the quantile regression model

Figure, Axes = plt.subplots(figsize=(10,8))

#Fitting the regression line onto the figure

YLine = lambda a, b: a + ViolenceWithInjury

y = YLine(YIntercept, Gradient)

#Adding the dataset values for violence with injury and violence without injury onto the figure

Axes.plot(ViolenceWithInjury, y, color="black")

Axes.scatter(ViolenceWithInjury, ViolenceWithoutInjury, alpha=.3)

Axes.set\_xlabel("Violence with injury", fontsize=20)

Axes.set\_ylabel("Violence without injury", fontsize=20)

#Showing the figure

plt.show()

#Taking violence with injury value to predict from user

UserValue = int(input("What violence with injury value would you like to predict the violence without injury value for?: "))

#Predicting value of violence without injury for given user value

Prediction = (Gradient\*UserValue)+YIntercept

#Printing prediction

print("For a violence with injury value of",UserValue,", the expected violence without injury value is",Prediction)

## Appendix 14: Means

#Importing Pandas

import pandas as pd

#Reading CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Determining mean value of homicide

HomicideMean = DataFrame["Homicide"].mean()

#Printing mean value of homicide

print("Mean of homicide:",HomicideMean)

#Determining mean value of violence with injury

ViolenceWithInjuryMean = DataFrame["Violence with injury"].mean()

#Printing mean value of violence with injury

print("Mean of violence with injury: ",ViolenceWithInjuryMean)

#Determining mean value of violence without injury

ViolenceWithoutInjuryMean = DataFrame["Violence without injury"].mean()

#Printing mean value of violence without injury

print("Mean of violence without injury: ",ViolenceWithoutInjuryMean)

#Determining mean value of stalking and harassment

StalkingMean = DataFrame["Stalking and harassment"].mean()

#Printing mean value of stalking and harassment

print("Mean of stalking and harassment: ",StalkingMean)

#Determining mean value of unlawful driving

UnlawfulDrivingMean = DataFrame["Death or serious injury - unlawful driving"].mean()

#Printing mean value of unlawful driving

print("Mean of unlawful driving: ",UnlawfulDrivingMean)

## Appendix 15: Medians

#Importing Pandas

import pandas as pd

#Opening CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Reading median of homicide

HomicideMedian = DataFrame["Homicide"].median()

#Printing median of homicide

print("Median of homicide:",HomicideMedian)

#Reading median of violence with injury

ViolenceWithInjuryMedian = DataFrame["Violence with injury"].median()

#Printing median of violence with injury

print("Median of violence with injury: ",ViolenceWithInjuryMedian)

#Reading median of violence without injury

ViolenceWithoutInjuryMedian = DataFrame["Violence without injury"].median()

#Printing median of violence without injury

print("Median of violence without injury: ",ViolenceWithoutInjuryMedian)

#Reading median of stalking and harassment

StalkingMedian = DataFrame["Stalking and harassment"].median()

#Printing median of stalking and harassment

print("Median of stalking and harassment: ",StalkingMedian)

#Reading median of death or serious injury through unlawful driving

UnlawfulDrivingMedian = DataFrame["Death or serious injury - unlawful driving"].median()

#Printing median of death or serious injury through unlawful driving

print("Median of unlawful driving: ",UnlawfulDrivingMedian)

## Appendix 16: Merging March 2018 and March 2019 Samples

#Importing modules

import pandas as pd

#Specifying datasets to be merged

Dataset1 = pd.read\_csv("March 2018 Cleaned Dataset.csv")

Dataset2 = pd.read\_csv("March 2019 Cleaned Dataset.csv")

#Merging datasets together

CombinedDataset = pd.concat([Dataset1,Dataset2])

#Dropping unnamed column at the beginning

CombinedDataset.drop(["Unnamed: 0"], inplace=True, axis=1)

#Writing merged dataset to new CSV

CombinedDataset.to\_csv("CT5018 Combined Assignment Dataset.csv")

## Appendix 17: Missing Values Heatmap for March 2018 Sample

#Importing modules

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import matplotlib.mlab as mlab

import matplotlib

from matplotlib.pyplot import figure

#Declaring missing values

MissingValues = ["n/a", "na", "?", ".."]

#Opening CSV

DataFrame = pd.read\_csv("policeforceareatablesyearendingmarch2018v2.csv", na\_values=MissingValues)

#Declaring columns to be examined

Columns = DataFrame.columns[:27]

#Creating heatmap with colours denoting available data and missing data

Colours = ["#000099", "#ffff00"]

sns.heatmap(DataFrame[Columns].isnull(), cmap=sns.color\_palette(Colours))

#Showing heatmap

plt.show()

## Appendix 18: Missing Values Heatmap for March 2019 Sample

#Importing modules

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import matplotlib.mlab as mlab

import matplotlib

from matplotlib.pyplot import figure

#Declaring missing values

MissingValues = ["n/a", "na", "?", ".."]

#Opening CSV

DataFrame = pd.read\_csv("policeforceareatablesyeendingmarch2019.csv", na\_values=MissingValues)

#Declaring columns to be examined

Columns = DataFrame.columns[:27]

#Creating heatmap with colours denoting available data and missing data

Colours = ["#000099", "#ffff00"]

sns.heatmap(DataFrame[Columns].isnull(), cmap=sns.color\_palette(Colours))

#Showing heatmap

plt.show()

## Appendix 19: Modes

#Importing Pandas

import pandas as pd

#Opening CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Reading mode of homicide

HomicideMode = DataFrame["Homicide"].mode()

#Printing mode of homicide

print("Mode of homicide:",HomicideMode)

#Reading mode of violence with injury

ViolenceWithInjuryMode = DataFrame["Violence with injury"].mode()

#Printing mode of violence with injury

print("Mode of violence with injury: ",ViolenceWithInjuryMode)

#Reading mode of violence without injury

ViolenceWithoutInjuryMode = DataFrame["Violence without injury"].mode()

#Printing mode of violence without injury

print("Mode of violence without injury: ",ViolenceWithoutInjuryMode)

#Reading mode of stalking and harassment

StalkingMode = DataFrame["Stalking and harassment"].mode()

#Printing mode of stalking and harassment

print("Mode of stalking and harassment: ",StalkingMode)

#Reading mode of death or serious injury through unlawful driving

UnlawfulDrivingMode = DataFrame["Death or serious injury - unlawful driving"].mode()

#Printing mode of death or serious injury through unlawful driving

print("Mode of unlawful driving: ",UnlawfulDrivingMode)

## Appendix 20: Pearson Correlation Coefficient Test for Violence with Injury and Violence without Injury

#Importing modules

from scipy.stats import pearsonr

import pandas as pd

#Opening CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Reading data for violence with injury

WithInjury = DataFrame["Violence with injury"]

#Reading data for violence without injury

WithoutInjury = DataFrame["Violence without injury"]

#Conducting test upon data

TestStatistic, PValue = pearsonr(WithInjury, WithoutInjury)

#Outputting test statistic and p-value

print("Test statistic is: %.3f, P-value is: %.3f"%(TestStatistic, PValue))

#Outcome if confidence in alternative hypothesis is 90-95%

if 0.05 < PValue < 0.1:

#Outputting that variables are somewhat likely to be correlated

print("Marginally significant evidence to reject null hypothesis; variables are somewhat likely to be correlated.")

#Outcome if confidence in alternative hypothesis is 95-99%

elif 0.01 < PValue < 0.05:

#Outputting that variables are likely to be correlated

print("Significant evidence to reject null hypothesis; variables are likely to be correlated.")

#Outcome if confidence in alternative hypothesis is above 99%

elif PValue < 0.01:

#Outputting that variables are very likely to be correlated

print("Extremely significant evidence to reject null hypothesis; variables are very likely to be correlated.")

#Outcome if confidence in alternative hypothesis is below 90%

else:

#Outputting that variables are unlikely to be correlated

print("Insufficient evidence to reject null hypothesis; variables are unlikely to be correlated.")

## Appendix 21: Quantile Regression Model

#Importing modules

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

import statsmodels.api as sm

import statsmodels.formula.api as smf

#Reading the CSV

DataFrame = pd.read\_csv('CT5018 Combined Assignment Dataset.csv', index\_col=0)

#Plotting a scatter plot, with Violence with injury on the x axis and Violence without injury on the y axis

DataFrame.plot(kind='scatter', x='Violence with injury', y='Violence without injury')

#Showing scatter plot

plt.show()

#Defining x axis of linear regression model and reshaping to fit model

ViolenceWithInjury = DataFrame["Violence with injury"].values.reshape(-1,1)

#Defining y axis of linear regression model

ViolenceWithoutInjury = DataFrame["Violence without injury"]

#Creating fitted linear regression model

QuantileRegressionModel=smf.quantreg("ViolenceWithoutInjury ~ ViolenceWithInjury", DataFrame).fit(q=0.5)

#Printing gradient of quantile regression line

Gradient = QuantileRegressionModel.params["ViolenceWithInjury"]

print("The gradient is:",Gradient)

#Printing y intercept of quantile regression line

YIntercept = QuantileRegressionModel.params["Intercept"]

print("The y intercept is:",YIntercept)

#Generating a MatPlotLib figure to visualise the quantile regression model

Figure, Axes = plt.subplots(figsize=(10,8))

#Fitting the regression line onto the figure

YLine = lambda a, b: a + ViolenceWithInjury

y = YLine(QuantileRegressionModel.params["Intercept"], QuantileRegressionModel.params["ViolenceWithInjury"])

#Adding the dataset values for violence with injury and violence without injury onto the figure

Axes.plot(ViolenceWithInjury, y, color="black")

Axes.scatter(ViolenceWithInjury, ViolenceWithoutInjury, alpha=.3)

Axes.set\_xlabel("Violence with injury", fontsize=20)

Axes.set\_ylabel("Violence without injury", fontsize=20)

#Showing the figure

plt.show()

#Taking violence with injury value to predict from user

UserValue = int(input("What violence with injury value would you like to predict the violence without injury value for?: "))

#Predicting value of violence without injury for given user value

Prediction = (Gradient\*UserValue)+YIntercept

#Printing prediction

print("For a violence with injury value of",UserValue,", the expected violence without injury value is",Prediction)

## Appendix 22: Standard Deviations

#Importing libraries

import pandas as pd

#Opening CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Reading standard deviation of homicide

HomicideSDev = DataFrame["Homicide"].std()

#Printing standard deviation of homicide

print("Standard deviation of homicide:",HomicideSDev)

#Reading standard deviation of violence with injury

ViolenceWithInjurySDev = DataFrame["Violence with injury"].std()

#Printing standard deviation of violence with injury

print("Standard deviation of violence with injury: ",ViolenceWithInjurySDev)

#Reading standard deviation of violence without injury

ViolenceWithoutInjurySDev = DataFrame["Violence without injury"].std()

#Printing standard deviation of violence without injury

print("Standard deviation of violence without injury: ",ViolenceWithoutInjurySDev)

#Reading standard deviation of stalking and harassment

StalkingSDev = DataFrame["Stalking and harassment"].std()

#Printing standard deviation of stalking and harassment

print("Standard deviation of stalking and harassment: ",StalkingSDev)

#Reading standard deviation of death or serious injury through unlawful driving

UnlawfulDrivingSDev = DataFrame["Death or serious injury - unlawful driving"].std()

#Printing standard deviation of death or serious injury through unlawful driving

print("Standard deviation of unlawful driving: ",UnlawfulDrivingSDev)

## Appendix 23: Variances

#Importing Pandas

import pandas as pd

#Opening CSV

DataFrame = pd.read\_csv("CT5018 Combined Assignment Dataset.csv")

#Reading variance of homicide

HomicideVariance = DataFrame["Homicide"].var()

#Printing variance of homicide

print("Variance of homicide:",HomicideVariance)

#Reading variance of violence with injury

ViolenceWithInjuryVariance = DataFrame["Violence with injury"].var()

#Printing variance of violence with injury

print("Variance of violence with injury: ",ViolenceWithInjuryVariance)

#Reading variance of violence without injury

ViolenceWithoutInjuryVariance = DataFrame["Violence without injury"].var()

#Printing variance of violence without injury

print("Variance of violence without injury: ",ViolenceWithoutInjuryVariance)

#Reading variance of stalking and harassment

StalkingVariance = DataFrame["Stalking and harassment"].var()

#Printing variance of stalking and harassment

print("Variance of stalking and harassment: ",StalkingVariance)

#Reading variance of death or serious injury through unlawful driving

UnlawfulDrivingVariance = DataFrame["Death or serious injury - unlawful driving"].var()

#Printing variance of death or serious injury through unlawful driving

print("Variance of unlawful driving: ",UnlawfulDrivingVariance)