How does the media's justification of racial brutality within the US police match up to raw data?

A data-science project aimed to address a worldwide phenomenon

Elliot Cooke
Paulo Lisboa (supervisor)
16 April 2021

Contents

On May 25th 2020, the US police killing of George P. Floyd Jr. marked a worldwide phenomenon, following an uproar against the prevalent issue of racism across the globe. The US police were targeted with allegations regarding brutality towards specific ethnic groups. Manipulation of information within the media to make headlines is often observed, raising queries about the extent of the issues broadcasted- but to what extent has there been an underlying issue of police brutality towards specific ethnic minorities within the US? A more rigorous analysis of shootings-related data can help answer this question and either reinforce the allegations of racism against the police, or unveil a side to the story that the media failed to broadcast.

Combining the passion for both this topic and data science provided an opportunity to apply related skills and knowledge of the field to gain meaningful answers.

1. Literature Review

The project was kickstarted via a detailed analysis of papers/publishes that had utilised the primary dataset for this project; it was found that five papers, in the form of Kaggle kernels, had been published. In order to conduct research on each of these, an excel spreadsheet containing the following variables was formed:

1. Research questions

The literature review highlighted common research questions within the papers:

- Are African-Americans disproportionately killed? *(Tkt, N., 2020)*
- Are police shooting deaths increasing? (*Tkt, N., 2020)*
- What is the geographical distribution of victims? *(Zarin, H., 2020)*
- Was there a difference between armed and unarmed victims? *(Almog, G., 2020)*

The proportion of African-American victims combined with the geographical distribution of victims and the social economic factors associated within their areas provided key insights regarding racial bias. Knowing whether there was a trend in fatalities within various races over the years highlighted those races for further analysis. Differences between armed and unarmed victims signalled the legitimacy of the police shooting at the time. These questions seemed vital to address in this project to test the reproducibility of results and gain key insights from the raw data.

2. Data used

Within all the five reviewed papers, the common dataset used was 'Fatal Police Shootings in the US (2015-2020)'; this was observed to be the main dataset, whilst supplementary data included:

- Crime rate in the US by state (2018) (*Almog, G., 2020)*
- US Census Demographic (common within two papers) *(Tkt, N., 2020)*
- US State Populations (2018, common within two papers) *(Tkt, N., 2020), (Zarin, H., 2020), (Thompson, C., 2020)*

All papers used more than one dataset (the main dataset with supplementary datasets). It seemed necessary to take this approach within this project as the main dataset alone had limitations to conclusions that could be derived from it. The reviewed papers used, on average, around two to three datasets; this was set as the minimum number of datasets to use for this project.

3. Data representation

All papers included the following main data types:

- Boolean (signs of mental illness, body camera, image mask)
- Int64 (Id)
- Float64 (age, census dataset, gender/race ratios, social-economic factors etc.)
- Object (name, date, manner of death, gender, race, armed, city, state, threat etc.)

Float64 (float) was used for some variables rather than Int64 (integer) due to its decimal place feature providing a higher degree of accuracy. Object proved to be the most common data type within the main dataset (A) as many of its variables were categorical. Int64 was used for 'victim ID' within the main dataset and frequency counts for variables within certain data frames. Boolean (binary) data was utilised for 'mental illness signs' and 'presence of a police body camera'. The range of data types within the reviewed papers indicated that a substantial variety of methods and techniques would be required in order to analyse and interpret the data sufficiently.

4. Null values

Null values, corresponding to missing values within the data, were observed to be present within the following variables:

- Age
- Race
- Flee
- Armed
- Gender

It was unknown as to how null values were dealt with in the majority of papers but one (*Almog, G., 2020).* A systematic approach utilising 3 techniques was applied to the these null values*:*

1. For the **continuous variables (‘age’), a function was used to fill nulls with the median, based on city and state.** This metric was chosen over the mean due to the presence of large outliers within this variable.

2. For the **categorical variables: ’race’, ‘flee’ and ‘armed’, the nulls were filled with a ‘Unknown’** (string).

3. Nulls in the final **categorical variable, ‘gender’, were filled with ‘male’.** As the vast majority of victims were male, this was seen the best course of action.

5. Techniques/ Methods used

Dealing with null values as a preliminary step opened the gateway to the next series of statistical techniques. These would form the structure of the exploratory analysis and modelling (described below):

- Data loading/cleaning (*Almog, G., 2020)*
- Dataframe creations (*Almog, G., 2020)*
- Univariate histograms (Marking the beginning of the exploratory analysis) (*Almog, G., 2020), (Kerneler, K., 2020)*

- Scatterplots (*Almog, G., 2020)*
- Bar charts, frequency line graphs *(Thompson, C., 2020)*
- Chi-squared tests *(Thompson, C., 2020)*
- Correlation Matrices (*Almog, G., 2020)*

Data cleaning addressed consistency in headings within different datasets. This proved key when creating and merging data frames to visualise important features and variable relationships.

In **all** the papers, it was observed that the exploratory data analysis section seemed to take up a large proportion of the project. This section included visualisation of variable's data distributions, through histograms and bar plots, whilst also entailing correlations between social-economic factors and number of victims. These correlations were visualised by **correlation matrices** and **scatterplots**. **Chi-squared tests** were used to **measure correlations of categorical variables**.

6. Conclusions

The key conclusions found from the review were:

- African-American deaths were disproportionate to their percentage population. (*Tkt, N., 2020)*

- No significant difference in the number of fatal shootings over the last 5 years. (*Tkt, N., 2020)*

- Alaska and New Mexico contained the highest number of fatalities. (*Tkt, N., 2020),* (*Almog, G., 2020), (Zarin, H., 2020)*

- Most victims were **male, shot, armed with a gun if armed, white, observed to show no mental illness signs, not fleeing, posing an 'attack' threat-level, and shot without the presence of a police body camera**. *(Zarin, H., 2020),* (*Almog, G., 2020)*

Whilst a desired element of uniqueness was presented throughout this project, it was important to try and reproduce the results and conclusions seen within the papers. These conclusions gave scope to further exploration, and therefore, new insights.

7. Evaluation of methods

Within (*Almog, G., 2020),* methods were evaluated via probability plots, MSE and STD statistics. All the other papers relied on visual comparison of graphs in the exploratory analysis as an evaluation technique. The evaluation of the methods within this project would, therefore, utilise visual comparison of graphs and some model performance statistics. Evaluation of modelling methods would depend entirely on the type of modelling technique(s) used.

2. Project research questions

After conducting the literature review, this project's research questions were devised:

A. What kind of people were killed?
B. How were they killed?
C. Were they armed? What with?
D. Were the victims posing a threat?
E. Were the fatalities recorded on police video record?
F. What were the correlations between variables?
G. What states showed the largest increases in/ most fatalities?
H. What insights could be found regarding social-economic factors relating to the victim's states?

Questions A-G formed the exploratory data analysis section; they were directly linked to the common questions observed within the literature review papers and would comprise of a large proportion of this project. Question H related solely to application of modelling methods, following on from key insights found regarding the social-economic factors of victim's states. These questions combined would realise an answer to the primary query of this project: **'To what extent does the media's justification of racial brutality within the US police match up to raw data?'.**

3. Methods (review of techniques/ methods to be used)

Histograms *(Histogram - Wikipedia, 2021):*

Histograms were utilised as an approximate representation of the discrete (numerical) data distribution. To construct a histogram, the range of values are 'binned' (dividing this range into an interval series) and the values that fall into each of these intervals are counted. These bins are specified as consecutive, non-overlapping intervals of a certain variable and must be adjacent. Histograms provide an approximate idea of the density of the distribution of data analysed, with a purpose to examine the probability distribution of a given variable. This is achieved by portraying the frequencies of observations occurring within a certain range of values.

Bar plots:

These specifically visualise categorical data with rectangular bars of heights proportional to their represented values. These charts provide easy comparison between these discrete categories.

Frequency-time graphs

These graphs derive from line charts, giving clear visualisation of changes in frequencies of variables over a time period.

Scatterplots *(Scatter plot - Wikipedia, 2021):*

Scatterplots use cartesian coordinates to exhibit values for, typically, two continuous variables within a dataset. These are used when just one of the variables is under control of the experimenter and the other depends on it, or when both variables are independent. Various types of correlations between variables entailing a specific confidence interval can be derived from these plots; these can be positive (increasing), negative (decreasing) or null (uncorrelated). Overall, the respective purpose is to highlight the type of relationship (if any) between two quantitive variables.

Cramer's V correlations (*Cramér's V - Wikipedia, 2021):*

The Cramer's V test, based on the Pearson's Chi-Squared statistic, was used to **measure the association between two nominal variables**, outputting a binary value (between 0 and +1). The theory behind this test statistic is shown below:

$$V = \sqrt{\frac{\varphi^2}{min(k-1, r-1)}} = \sqrt{\frac{\chi^2/n}{min(k-1, r-1)}}$$

Where:

- φ is the phi coefficient
- χ^2 is derived from the Pearson's chi-squared test
- n is the grand total of observations
- k is the number of columns
- r is the number of rows

The significance of the Cramer's V test result is measured by the **p-value.**

Logging *(Logarithmic scale - Wikipedia, 2021):*

Utilising the logarithmic scale is a technique of exhibiting numerical data over a wide range of values in a compact form. This scale is **non-linear.** In statistics, log transformations on data is used to turn highly-skewed distributions less skewed, providing value for easily interpreting patterns within the data whilst meeting assumptions of inferential statistics (regarding predictions).

Shapiro-Wilk test *(Shapiro–Wilk test - Wikipedia, 2021)*:

This is a normality test (verifies whether a data distribution resembles normality). The Shapiro-Wilk test examines the null hypothesis that a sample $x_1, \ldots, x_n$ is derived from a normally-distributed population. The test statistic is documented below:

$$W = \frac{(\sum_{i=1}^{n} a_i x(i))^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2},$$

Where,

- $x(i) = i^{th}$ order statistic (i^{th} smallest number within sample)
- $\bar{x} = (x_1 + \ldots + x_n)/n$ is equivalent to the sample mean

The coefficients a_i are given by:

$$(a_1, \ldots, a_n) = \frac{m^T V^{-1}}{C},$$

Where C is a vector norm (a function from a real/ complex vector space to the non-negative real numbers that behave in specific ways. Commutes with scaling, obeys a form of triangle inequality and is only zero at the origin):

$$C = ||V^{-1}m|| = (m^T V^{-1} V^{-1} m)^{1/2}$$

And the vector m,

$$m = (m_1, \ldots, m_n)^T$$

Is comprised of the expected values of the order of statistics of independently distributed random variables sampled from the normal distribution.

Finally, V is the covariance matrix of the normal order statistics.

Jarque-Bera Test *(Jarque–Bera test - Wikipedia, 2021):*

This test is a second normality test examining whether a data sample contains skewness and kurtosis (whether the tails of a distribution contain extreme values) matching a normal distribution. The test statistic is always non-negative- if it is far from zero, it highlights that the data does **not** posses a normal distribution.

The test statistic is provided below:

$$J = \frac{n}{6}(S^2 + \frac{1}{4}(K-3)^2)$$

Where n is the number of observations.

S, sample skewness, is described as:

$$S = \frac{\hat{u}_3}{\hat{\sigma}^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^{3/2}},$$

And K, sample kurtosis, is:

$$S = \frac{\hat{u}_4}{\hat{\sigma}^4} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}$$

Where $\hat{u}_3$ and $\hat{u}_4$ are approximations of the third and fourth central movements (a moment of a probability distribution of a random variable about its mean).

If the data derives from a normal distribution, this test statistic asymptotically entails a chi-squared distribution containing two degrees of freedom. Therefore, the statistic can be used to test the hypothesis that the data sample resembles normality.

Heatmaps (*Heat map - Wikipedia, 2021):*

Used as a data visualisation technique, this shows the magnitude of phenomenon as colour within two dimensions. Colour variation may be by hue or intensity, giving clear visual discrimination between clusters/ varies within a space. The cluster heat map (used in this project), entails a matrix of fixed-sized cells, whereby its rows and columns are discrete phenomena/ categories. The heat map chart specifically shows a two-dimensional correlation matrix table between two discrete dimensions, and the correlation values range from -1 to +1. Values closer to zero signalled **no linear trend between the two variables.** The closer to +/-1 the values where, the **more positively correlated the variables were.**

K-means clustering (*k-means clustering - Wikipedia, 2021):*

Clustering is applied as a form of unsupervised learning, whereby the goal is to discover hidden patterns in the data that can realise valuable results. The unique aspect of unsupervised learning is the absence of any labels associated with the learning algorithm (in this case, the K-means algorithm) when finding the structure of its input variables. Clustering relates to the grouping of certain data points into unique 'clusters', whereby a label is attached to each data point within a set. The data points that are observed to be closer together are assigned equal labels. Within a cluster, inter-point distances are small compared to distances outside the cluster. In order to compute these distances, the clustering algorithms rely on a distance metric between the data points; two common distance metrics include the euclidean distance (length of a line segment connecting y and x) and the city block distance (absolute value distance).

The K-means clustering algorithm was specifically chosen as the unsupervised clustering algorithm in this project for various reasons *(Google Developers, 2021)*:

A. Relatively simple to implement **efficiently**- this proved as a good selection for starting out in the foreign topic of clustering
B. Scaled to **large datasets** (the S.E.F dataset was high-dimensional)
C. Guaranteed convergence to a centroid within a cluster

D. Generalised to clusters of different shapes and sizes (such as elliptical clusters)
E. Works only for numerical data (this could be seen as a drawback, but the dataset solely comprised of numerical data)

Despite the suitability of this algorithm for the task, there are some disadvantages of using K-means. These included having a strong sensitivity to outliers/noise and a relatively low capability to pass the local optimum.

The **overall goal of this algorithm is to partition the dataset into a number (K) of clusters**; the optimum number of clusters is discovered using a complimentary analysis technique (silhouette).

The setup of the k-means algorithm is described below:

- Obtain the dataset of N observations (in this case, 1002060 data points)
- $\mu_k(k = 1,...k; k \leq N) =$ prototype associated with k^{th} cluster. (μ_k represents cluster centres)
- Each observation (vector x_n) assigned to **only one** cluster
- $\gamma_{nk} \sum 0,1$ describes which of the k clusters the data point x_n is assigned to
- If x_n assigned to cluster k, $\gamma_{nk} = 1$ and $\gamma_{nj} = 0$ for $j \neq k$
- Objective function: $J = \sum_{N=1}^{N} \sum_{k=1}^{k} \gamma_{nk} || x_n - \gamma_k ||^2$
- Obj. function relates to sum of squares of distances of each point to its assigned vector μ_k
- Goal: find γ_{nk} and μ_k so that it minimises J

The iterative procedure to assign the data points to their closest cluster centres is also stated below:

1. Choose the initial values for μ_k
2. Minimise J with respect to γ_{nk}, keeping μ_k **fixed**
3. Assign the n^{th} data point to the **closest cluster centre**
4. Minimise J with respect to μ_k, keeping γ_{nk} fixed

The equation for μ_k is stated below:

$$\mu_k = \frac{\sum_{N=1}^{N} \gamma_{nk} X_n}{\sum_{N=1}^{N} \gamma_{nk}}$$

The denominator of the equation equates the number of data points assigned to cluster, K.

μ_k is equal to the **mean of all data points X_n assigned to cluster, K.**

Silhouette (*Silhouette (clustering) - Wikipedia):*

To compliment the k-means method, silhouette analysis was computed in order to find the optimum number for K (number of clusters). This analysis method supplies a graphical representation of how well each item has been clustered. The result was a measure of how similar an item was to its assigned cluster (cohesion) in contrast to other clusters (separation). This value ranged from -1 to +1, whereby a higher value indicated that the object was better matched to its own cluster and poorly matched to neighbouring clusters. If the majority of objects realise a high value, the clustering configuration is deemed appropriate (many points with a low value highlight that there may be a wrong number of clusters. Whilst the silhouette can be calculated with any distance metric, the default 'euclidean distance' metric was used (it is a safe option as the choice of distance metric has a big influence on the clustering results).

The theory behind silhouette is described below:

- **Assuming the data has been clustered (via k-means for example) into k clusters…**

  For the data point $i \in C_i$ (data point i in cluster C_i), let:

$$a(i) = \frac{1}{|C_i - 1|} \sum_{j \in C_i, i \neq j} d(i, j)$$

  …be the **mean distance** between i and all other data points within the same cluster.

  - $d(i, j) =$ distance between data points i and j in cluster C_i
  - $a(i) =$ measure of how well i suits its assigned cluster (smaller value = more suited)

- **For each data point $i \in C_i$, we then define…**

$$b(i) = min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

  … to be the **smallest mean distance** of i to all points in any other cluster where i is not a member.

  - Cluster with **smallest mean dissimilarity** = neighbouring cluster of i.

- **We can then define a silhouette (value) of a data point, i…**

$$s(i) = \frac{b(i) - a(i)}{max[a(i), b(i)]} \quad \text{if } |C_i| > 1$$

- $s(i) = 0$, if $|C_i| = 1$
- Therefore, $-1 \leq s(i) \leq 1$

The mean, $s(i)$, across all data points of a cluster measures how closely grouped the points within a cluster are. The mean over **the entire dataset** is a **measure of how well the data has been clustered.** If there are a too many or too few clusters (bad choice of K), then some of them may result in much narrower silhouettes than the rest. These silhouette graphs and means are used to establish the optimum number of clusters for the dataset.

4. Application of methods

A search of data from Kaggle resulted in the selection of 4 datasets to be used for this project:

A. Fatal Police Shootings (main dataset) *(kaggle.com, 2020)*
B. Census tract 2017 *(factfinder.census.gov, 2017)*
C. Violent Crime Rates US 2017 *(FBI.gov. 2017)*
D. Percent over 25 that completed high school (education dataset) *(factfinder.census.gov, 2015)*

These datasets were selected based on insights from the literature review, where (A) and (B) were common, and aims of this project. Datasets (C) and (D) helped create paths to deep analysis in this topic.

IDE setup

All results were produced using python code, due to its power regarding data science applications. This project posed as an excellent opportunity in adapting a basic data science role in a real-world scenario, given that most data science roles utilise Python. Jupyter notebook was used as the IDE due to its interactive benefits when updating the project leader on the coding progress made. The set-up in this IDE included importing numerous libraries and packages for data loading, plotting, linear algebra computations and more.

Data handling

The 4 datasets were loaded into Python before undergoing brief analysis; this realised an understanding of how the data was formatted. A lack of consistency in regards to column names within the four datasets was observed- in order to prepare the data for manipulation later, code was executed *(Welch, A., n.d., undated)* to make these column headings consistent with one another. Secondly, 'Puerto Rico' was dropped from the Census dataset as this project is based on **US** data. Within the literature review, it was seen that data-frame creations, data merging and handling of null values were necessary actions in preparing the data for analysis. These steps are specified below:

1. Creation of a 'Total Population' ,'Race Ratios' and a 'Socio-Economic' data frame
2. Merging these data-frames together with the main dataset (A) to make dataset (E)
3. Checking for null values
4. Handling null values within the poverty dataset and overall dataset (E).

The data frame for ‘total population’ for each state included variables such as the number of men and women and gender ratios. Gender ratios were key as they provided the proportion of men to women in each state; these were calculated using division functions within Python.

A ‘race ratios’ data-frame was then created, showing the percentage of the population of each state that fell into six race categories; white, black, hispanic, asian and pacific. The computation of these ratios presented the opportunity to discover if any states signalled possible prevalence of racism towards its victims.

The ‘social-economic’ data-frame entailed **social-economic rates** of variables such as poverty, carpool and income per capita. Some of these variables seemed unrelated to the shootings at first, but would later provide insightful correlations between social-economic factors and the number of victims.

These data frames were combined with the main dataset (A) to create the primary dataset for this project. Having all the data in one location made accessing data throughout this project fluid.

Null values (part of data handling)

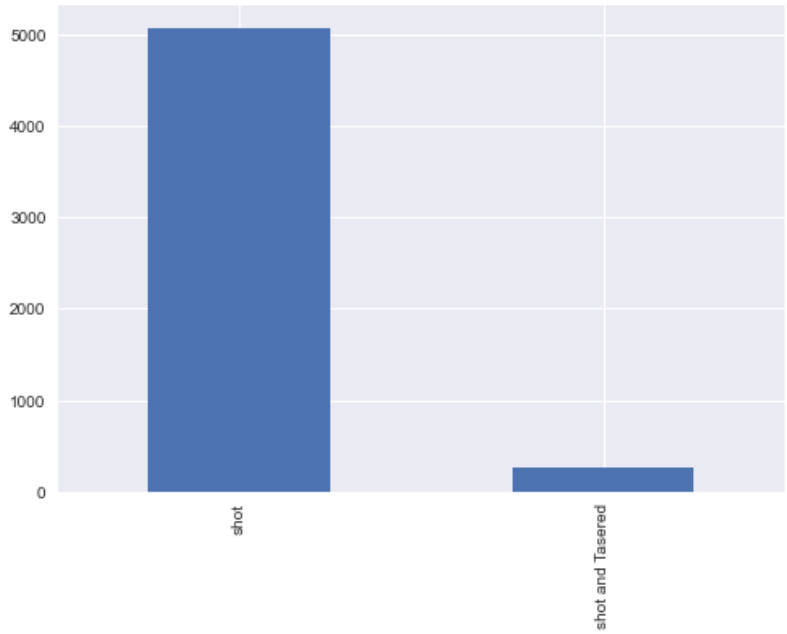
There were a number of null values found within the data; these had to be dealt with in order to avoid misleading conclusions from the analysis and modelling. When searching for nulls within the overall dataset (E), it was found that there were 1346 missing entries out of 5332 (25%). This percentage was excessive; a systematic approach for filling the null values was applied. This approach was similar to the system observed within the literature paper (*Almog, G., 2020):*

- **Age:** 249 null values found, which were filled with the **‘median’ based on city and state.** This technique was used within the literature papers as the **presence of extreme outliers within the age** data highlighted that the **median would be a better measurement** to use in contrast to the mean.

- **Race, Flee and Armed:** These variables contained a combined total of 1097 null values, which were filled with **‘unknown’**. These could not be filled with a mean or median as they were **categorical variables.**

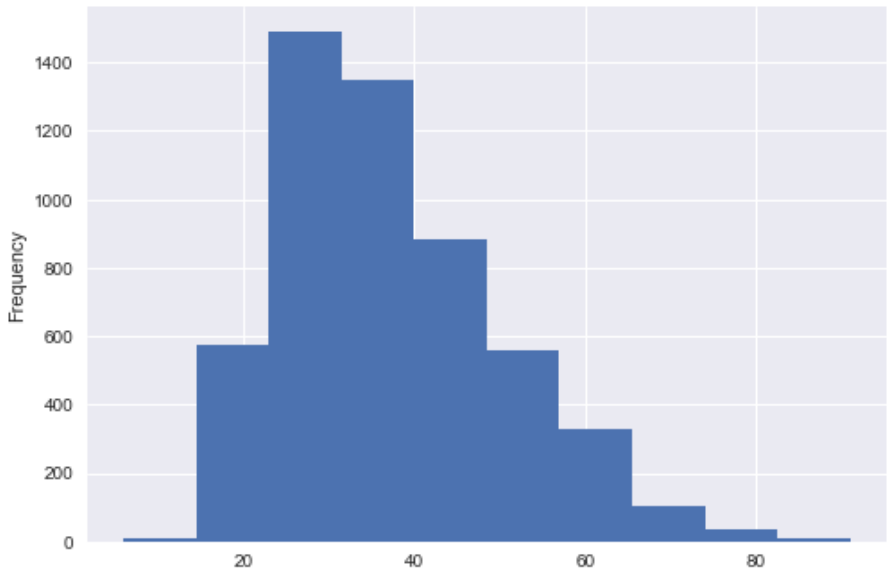
- **Gender:** There were only 2 null values found. Due to the **large majority of male victims** and a significantly small number of missing entries, these were filled with **‘male’.**

Exploratory data analysis

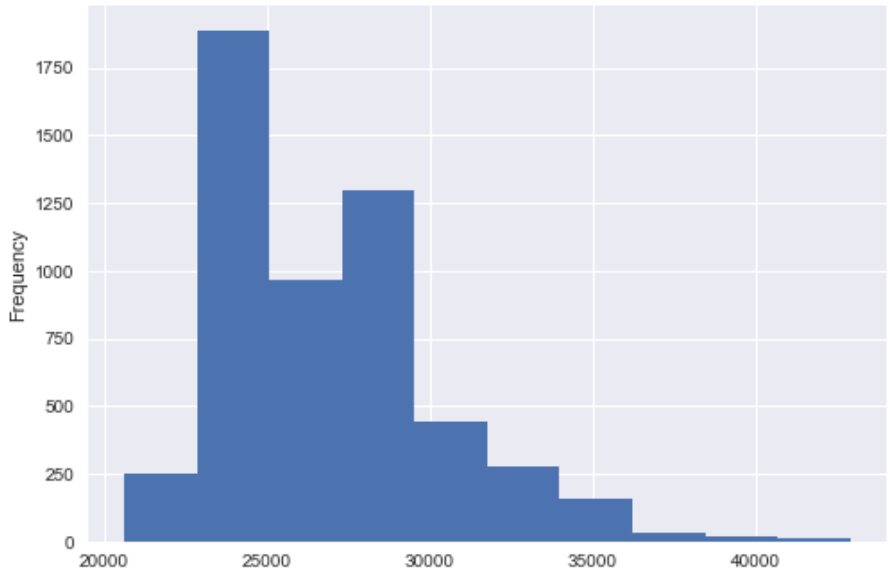


1. Histograms

The first step entailed creating univariate histograms *(Thompson, C., 2020)* of the relevant discrete numerical variables within the data (discrete data only took certain values). Univariate histograms were common plots within **all** literature papers, and were used to visualise the distribution of data. The plots obtained are shown below:



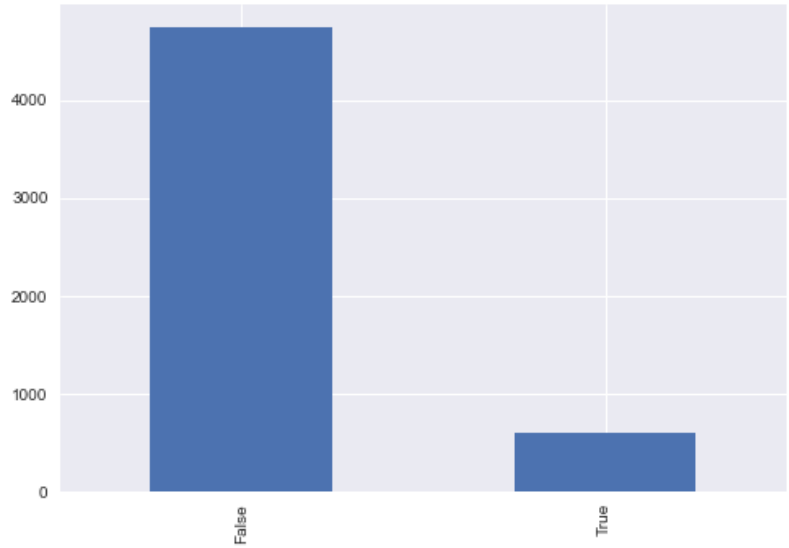
The age histogram highlighted a positive skew in the data, indicating that the majority of victims fell into the younger to middle-aged category. There was a **large range** spotted (5-95 years old).



Income per capita measured the average income earned **per person in a given area** within a specified year. This metric was calculated by division of the area's total income by its total population. The distribution of this data visualised a positive skew in general, with an unexpected jump at around $28000 breaking the decreasing trend. The highest peak was at around $23000, and the higher end of income had a smaller frequency of victims.

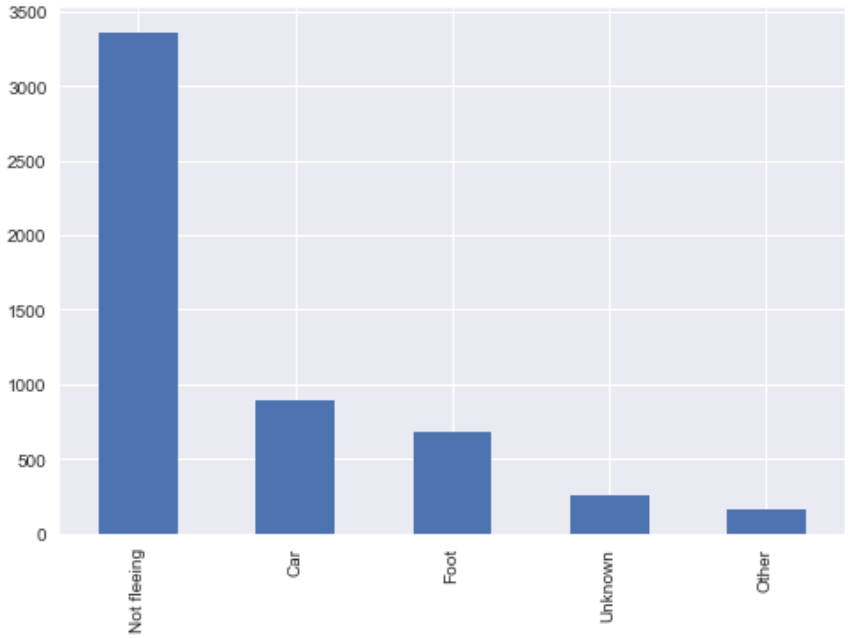
2. Bar plots

The distributions of categorical variables were visualised using ‘bar’ charts *(Thompson, C., 2020)*- a key technique when plotting ‘non-numerical’ data that could only be distinguished in certain categories. These results are documented below:

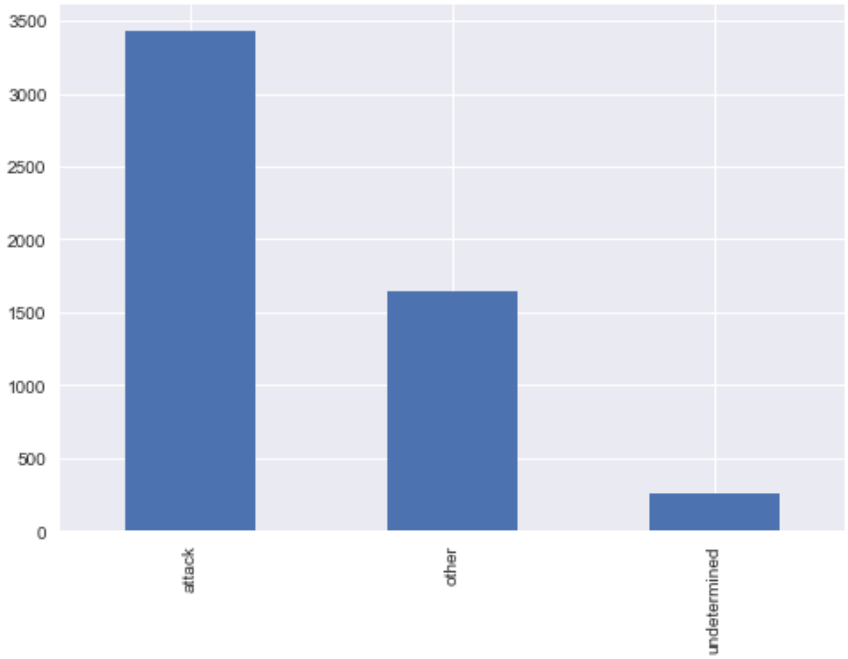


The l**arge majority of shootings occurred without a police body camera** present.

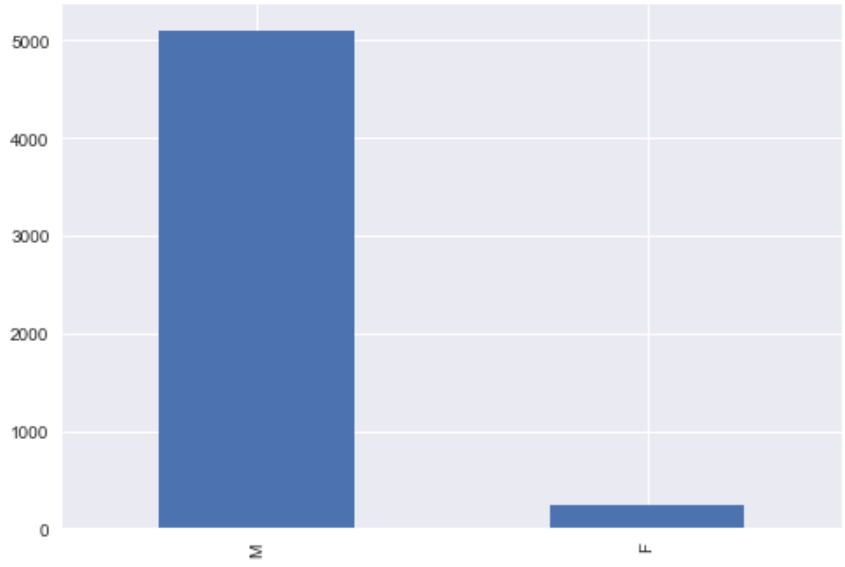
A **huge proportion of victims were solely shot**, as opposed to shot and tasered. Questions are raised here about the extent of police brutality in these cases; should all victims be tasered before being shot to potentially avoid a fatality?



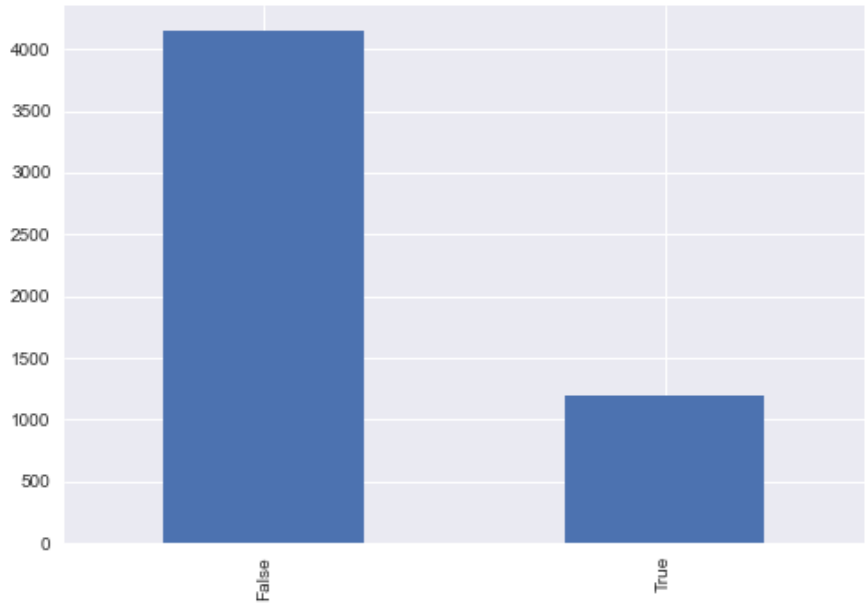
The bar chart showing whether the victims fled the scene or not underlined a mixed distribution in data. However, **a significantly large proportion of victims were recorded as 'not fleeing'**.



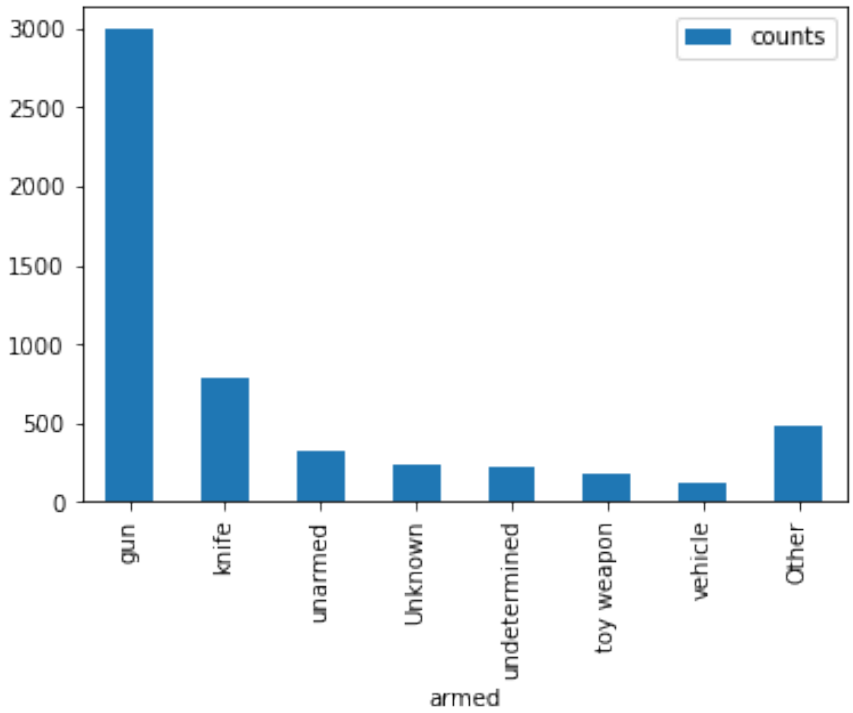
Following the previous question, the chart above signalled that **most victims showed determination to attack/ were attacking** when being confronted. However, a significant proportion took up the unspecified 'other' category. What was defined as 'other'?



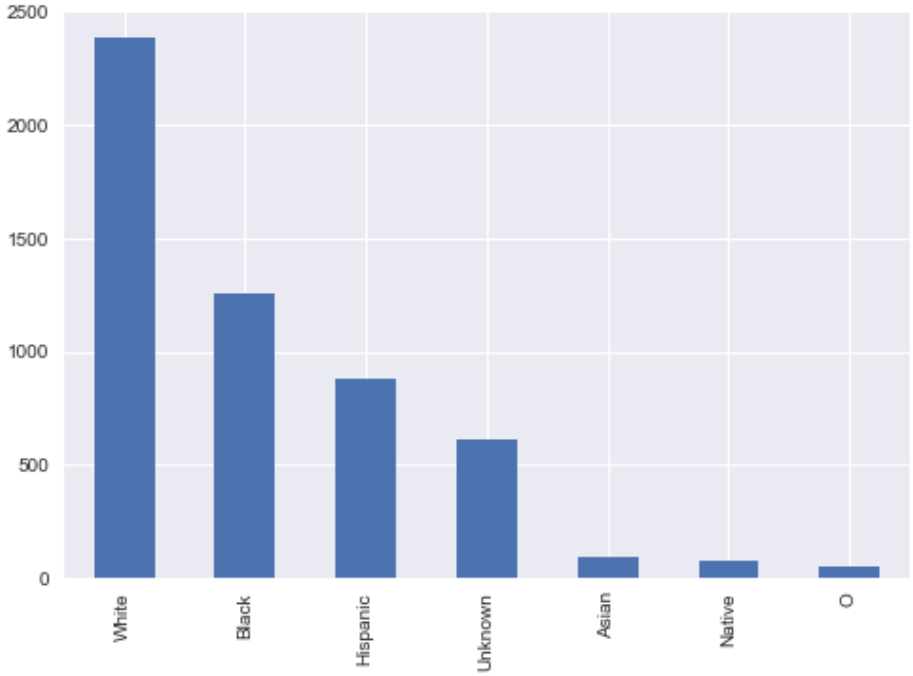
It was seen from the gender bar plot above that the **vast majority of victims were male.**



Another huge proportion of victims seemed to show no mental illness signs.



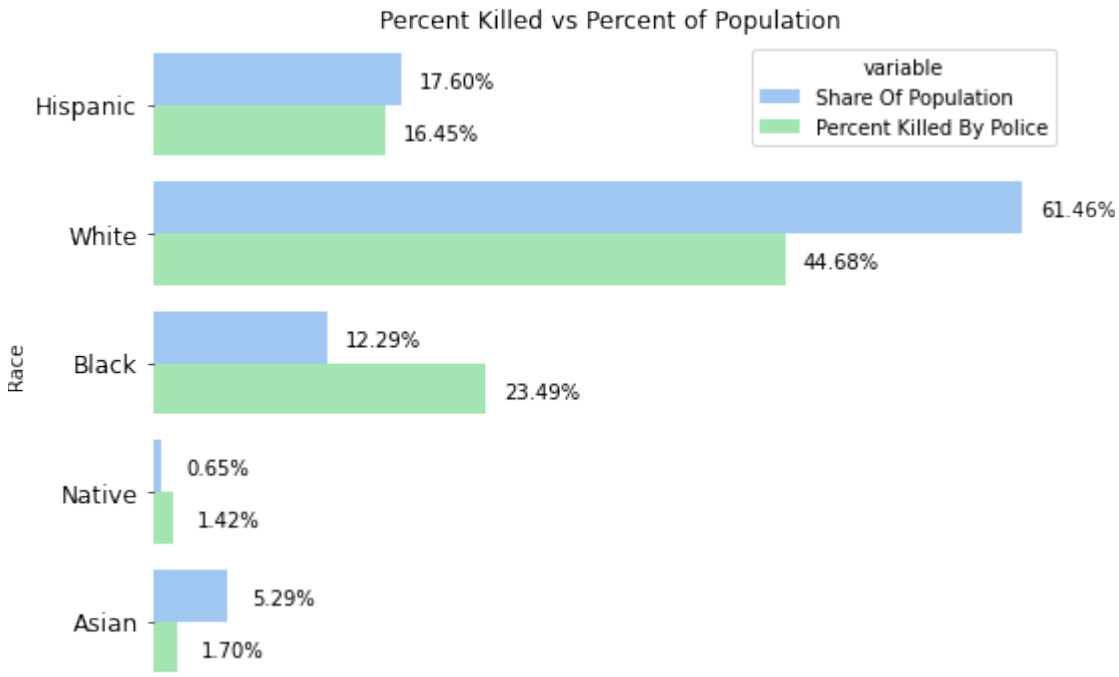
Most victims were armed with a gun, and other weaponry generally had lower frequencies. This variable contained a huge number of categories; these were grouped into an 'other' category using python functions.



The final bar chart, showing the distribution of race, shows that whilst **white victims were killed the most, Asian, native and other race frequencies were much lower.**

3. Stacked bar plot

Whilst the race plot gave some interesting insights, a stacked bar plot was produced (*Almog, G., 2020)* in attempt to highlight races that were killed disproportionately to their



Stacked bar chart

percentage share of the US population. Data was used from a dataset seen within (*Almog, G., 2020),* containing the share of race by city *(factfinder.census.gov, 2017).*

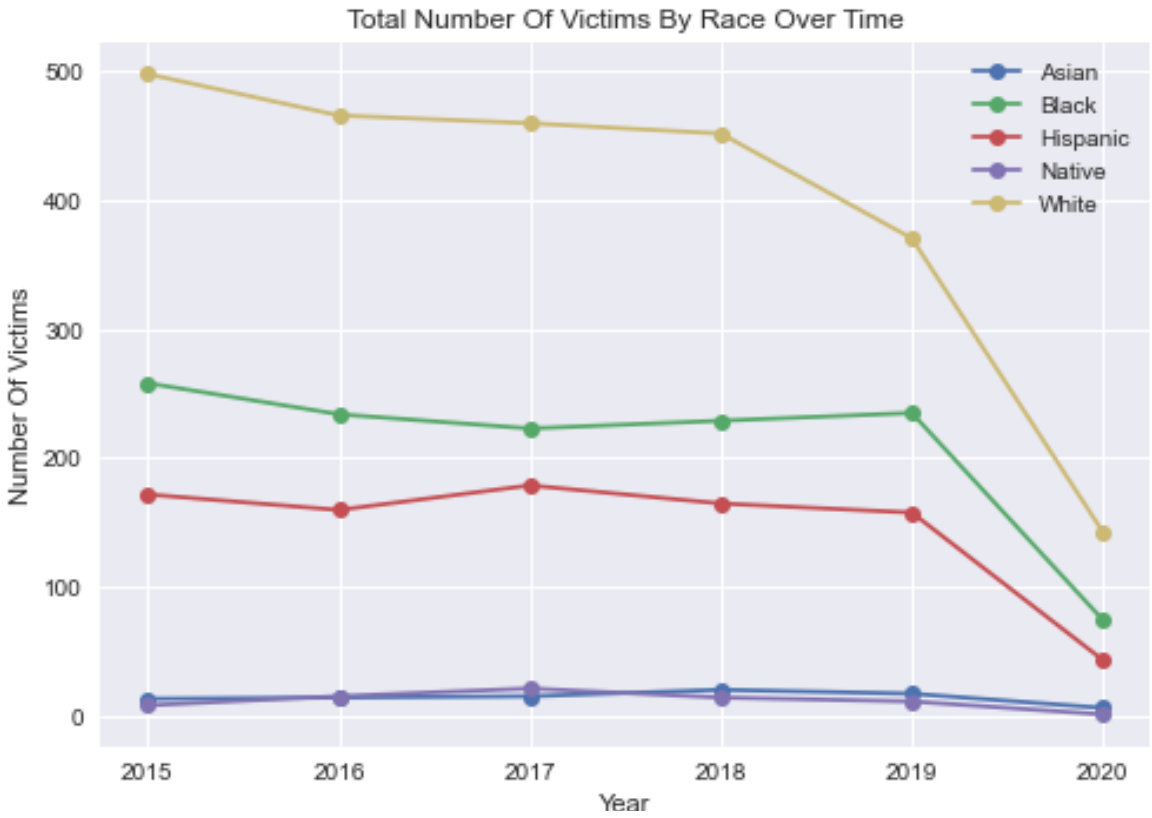
The plot highlighted that whilst the percent of white victims was noticeably higher than the percentage share of whites within the US population, the opposite was true for blacks and natives. These ethnic groups seemed to be killed disproportionally to their percentage share of the population so that there was a higher percent killed than share of population. In order to compare these two ethnic groups, the percentage difference between share of population and percent killed by police was calculated:

- Difference for black: 48%
- Difference for native: 54%

It was highlighted that whilst both differences were similar, native was higher by 6%. Both ethnic groups seemed to contain **twice** the amount of victims as their US population.

4. Frequency-Time Graph

An interesting aspect explored within one of the reviewed papers *(Almog, G., 2020)* was the **total number of victims by race over time**. This was also explored in this



Frequency-Time Graph

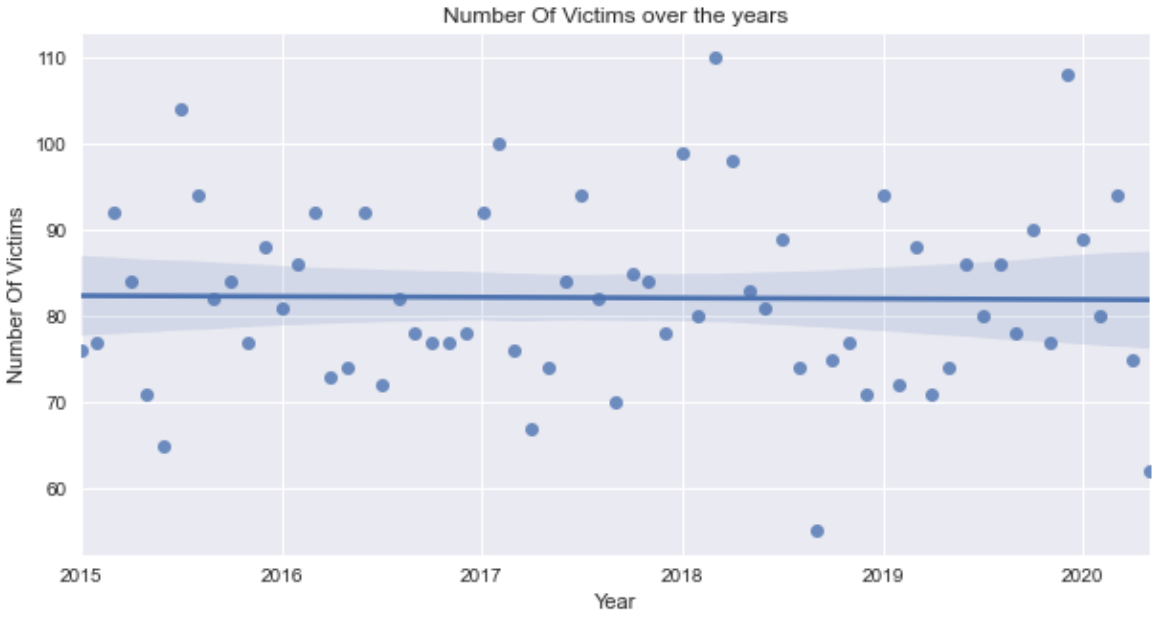
project, by merging race data with time data (year) and plotting:

The frequency-time graph was used to display the trend in number of victims over the past 5 years by race. Whilst the graph showed that numbers for native and Asian victims had remained fairly constant, the numbers for Hispanic and black had shown fluctuations (increases and decreases). The number of white victims highlighted a decreasing trend over the years, with a large fall from 2018-2019. It was important to note here that the **decreasing trend from 2019-2020 for all races was due to the fact that the recorded data for this year was for only half of 2020,** and therefore it was **not valid to say that numbers had shown a decrease for the full 2020 annum.**

The native data seemed to show slightly less of a decrease for the half year of 2020.

5. Scatterplot

In order to find a trend in the **overall number of victims** over the years, a scatterplot *(Almog, G., 2020)* was constructed:



Scatterplot

Despite apparent fluctuations within the frequency-time graph, the distribution of data displayed here showed no skew and a flat gradient.

6. Correlations

Exploring the distributions only provided a basic understanding of the data and some of its characteristics. In order to understand what factors influenced the shootings, and to derive relationships between different variables, correlations were explored. There were two methods in exploring these:

1. Cramer's V Test (for categorical variables)
2. Choropleth Heatmap (for the social-economic factors a.k.a continuous numerical variables)

6.1. Cramer's V test

The Cramer's V test was computed in Python by defining a function to return the correlation nature as 'negligible', 'small', 'medium' or 'large'. This was achieved so that

the user could clearly interpret the size of the correlation generated from the Cramer's V value. Then, the Cramer's V test was executed, using the 'chi-squared' test. An example of the output is documented below:

Out[61]:

	Chi-square test	results
0	Pearson Chi-square (50.0) =	84.2990
1	p-value =	0.0017
2	Cramer's V =	0.1257

```
In [62]: V = res.iloc[2,1]
         print(V)
         int(V)

         0.1257

Out[62]: 'small'
```

Cramer's V Example

The results from the test were outputted as a 3x2 table containing the p-value and Cramer's V statistic. In the case above, the V value came to **0.1257,** whilst the correlation-nature function highlighted only a small correlation between the two variables in comparison.

The correlations between these categorical variables are documented below:

- Threat level vs flee (0.118- small)
- Armed vs signs of mental illness (0.2571- small)
- Armed vs race (0.1257- small)
- Armed vs age (0.1086- small)
- Age vs race (0.1844- small)
- Manner of death vs race (0.0406- negligible)
- Manner of death vs state (0.1257- small)

The highest correlations were capped at 'small'.

6.2.1 Preliminary steps to heat map

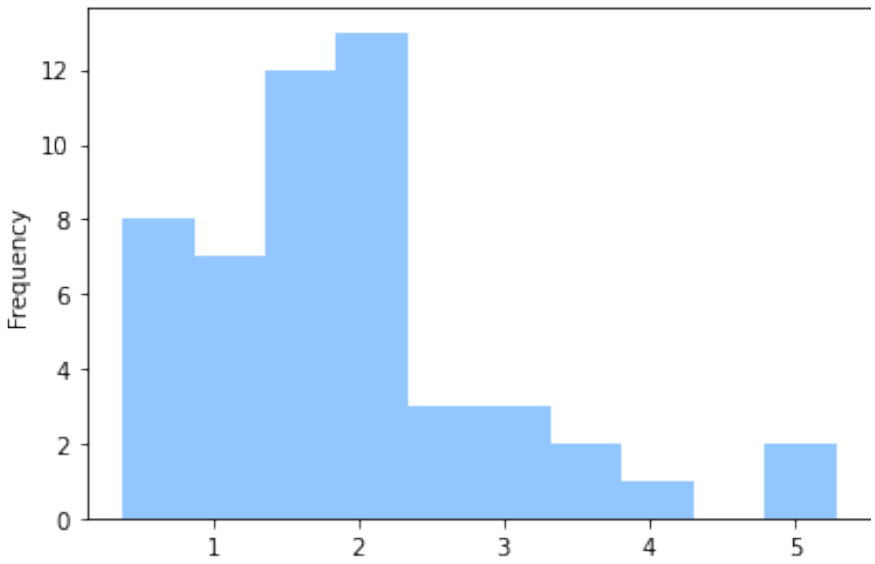
Correlations for the social-economic factors (S.E.F) were computed via a data-frame creation of each state's demographic and social-economic factors. Data from the population, race ratios and S.E.F datasets created previously was merged on 'state'. A column containing the number of victims per 100k citizens was then created; this was observed as a good measure of population and was used within the literature review paper (*Almog, G., 2020).* 100k citizens was a good number to use as 100,000 people comprises of a substantial amount of data that would realise sufficient conclusions.

Two more columns were added to this dataset:

- **Violent crime rate** (taken from the violent crime rate dataset *(FBI.gov, 2017)*)

- **Percent completed high school** (taken from the education dataset *(factfinder.census.gov, 2015)*)

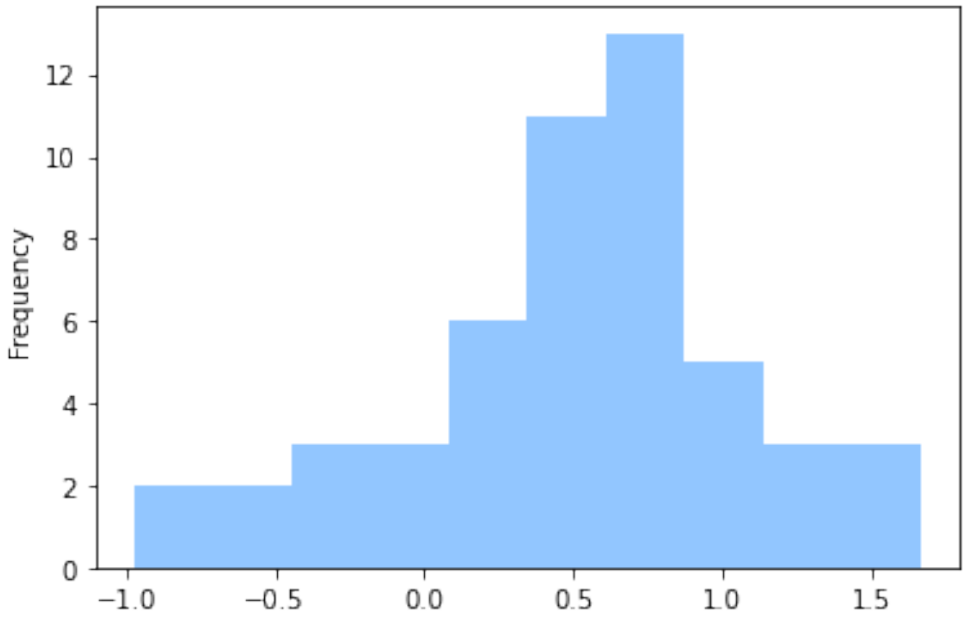
Having constructed the new data-frame, a histogram was plotted to gain insight of the distribution of data within:



Victims per 100k citizens

In order to respond to the highlighted skewness of data shown from this histogram, the data was **logged.** It was important to later verify whether the data was definitely normal when exploring correlations between the variables. The resulting 'logged' histogram is shown below:

The distribution of data became more clear, but normality was verified using stats as well as graphs:



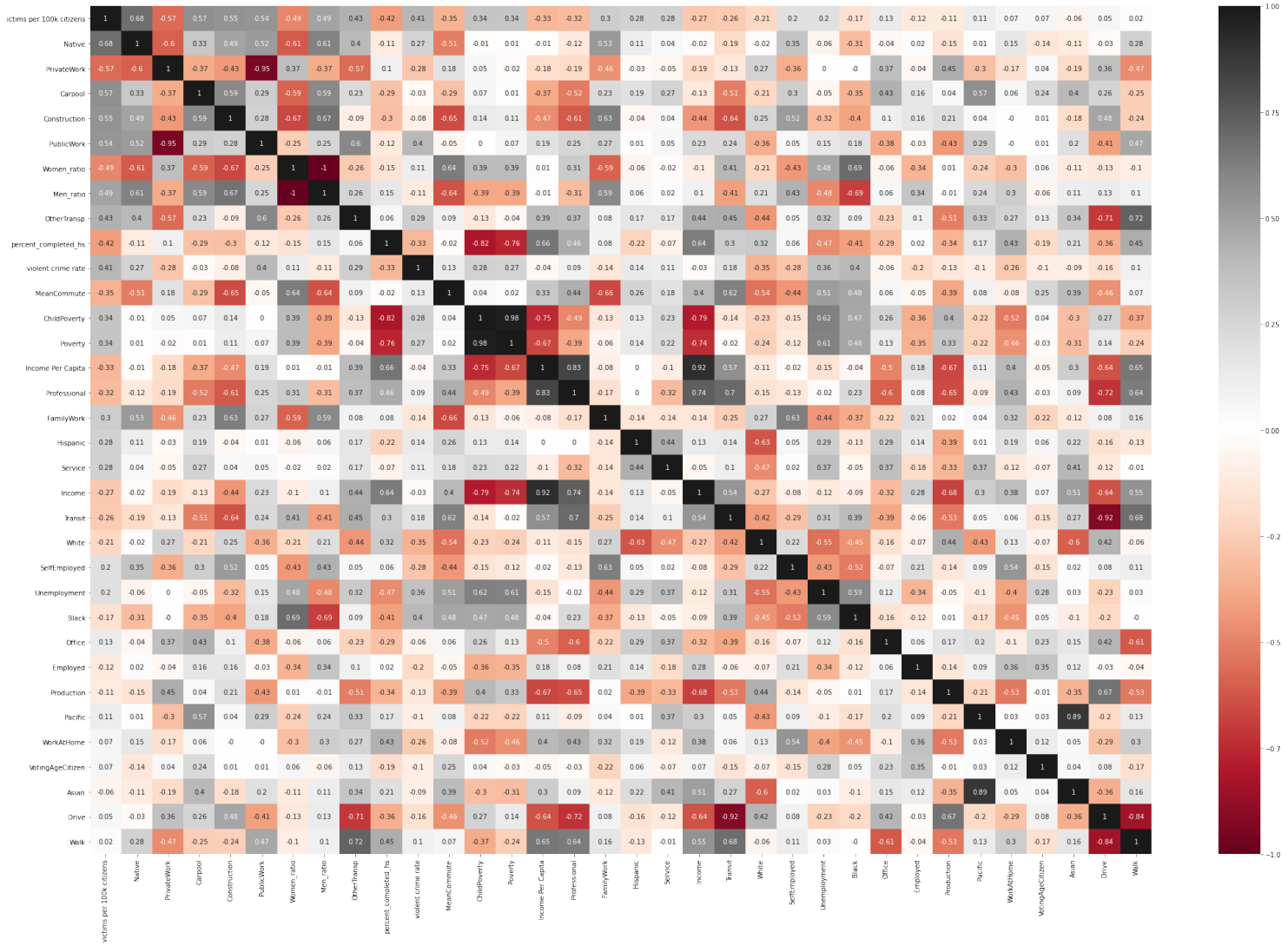
Logged victims per 100k citizens

- **Shapiro-Walk Test:** An outputted a p-value of 0.327 indicated that the data **was normal**, as the p value was greater than 0.05 (at the 5% significance level).

- **Jarque-Bera Test:** this test was used to verify whether the data was definitely not normal, as it was discovered that datasets with 2000 entries or more made the Shapiro-Wilk test unreliable to use. However, the Jarque-Bera test still concluded that this data **was normal.**

6.2.2 Heatmap correlation matrix

This discovery directly related to the next stage of the analysis: **the correlation heat-map of the social-economic factors.** From previously finding that the data resembled normality, the test used for this heat-map to determine the correlation values was the 'Spearman's Rank Test'. Using a series of Python functions *(Almog, G., 2020)*, including the 'heatmap' function from the 'seaborn' package, the correlation matrix table was constructed:



Heatmap of social-economic factors

Insights regarding 'victims per 100k citizens':

- **Native (0.68), Carpool (0.57) and Construction (0.55)** seemed to be the **most positively correlated variables** to the **number of victims (per 100k citizens).**

- **Private work (-0.57)** seemed to be the **most negatively correlated variable to victims.** All other negatively correlated variables showed low correlation.

Insights regarding 'violent crime rate':

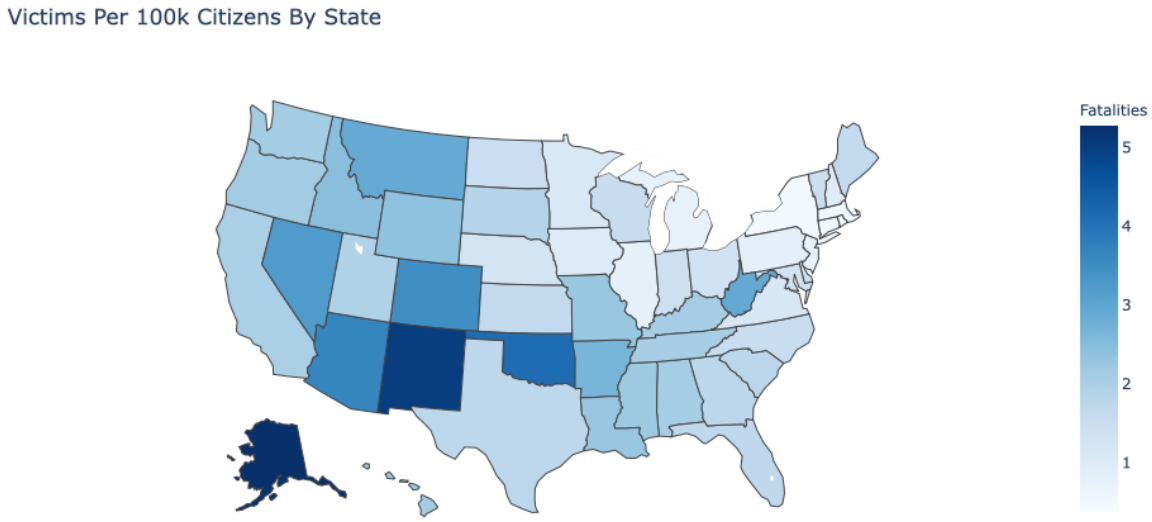
- **Moderate** correlation to **number of victims** (0.41)
- **Moderate** association to **'black'** (0.40)

Other insights:

- **Private work** and **public work** were **highly negatively correlated** (-0.95)
- **Poverty** and **child poverty** were **highly positively correlated** (0.95)
- **Income** and **income per capita** were **highly positively correlated** (0.92)
- **Transit** and **drive** were **highly negatively correlated** (0.92)
- **Income per capita** and **professional** were **highly positively correlated** (0.83)
- **Percent completed high school** and **poverty** were **highly negatively correlated** (-0.76)

7. Chorpoleth heat map map of USA

This map was coded *(Thompson, C., 2020)* to display the **number of victims per 100k citizens by state:**



US map showing no. victims per state

It was seen that **Alaska, New Mexico and Oklahoma** were the **top three states** for **highest** number of victims respectively. The t**op three states with the lowest number** of victims were **New York, Massachusetts and Connecticut.**

Having discovered this information, the mean rate of the 'black' and median rate of the 'native' column within the S.E.F dataset was calculated (median was used for 'native' due to the presence of outliers, which were discovered in a box plot of this variable).

Whilst rates for 'black' were high, normal (mean) and low for NY, CT and MA respectively, **the rates for native were below average in all these states.**

When analysing the top three states with the highest number of fatalities in the same manner, it was found that there were below average rates for 'black' and way-above average rates for 'native' in **all these top states**.

To further explore these ideas, the S.E.F dataset was targeted for modelling.

8. **Modelling**

The modelling stage of the project would follow on from the heat map correlation analysis of the social-economic factors dataset. The aim was to conduct a key modelling method plus two complimentary analysis techniques in order to characterise the dataset and gain a deeper insight into how the social-economic factors were related to one another.

The techniques utilised were:

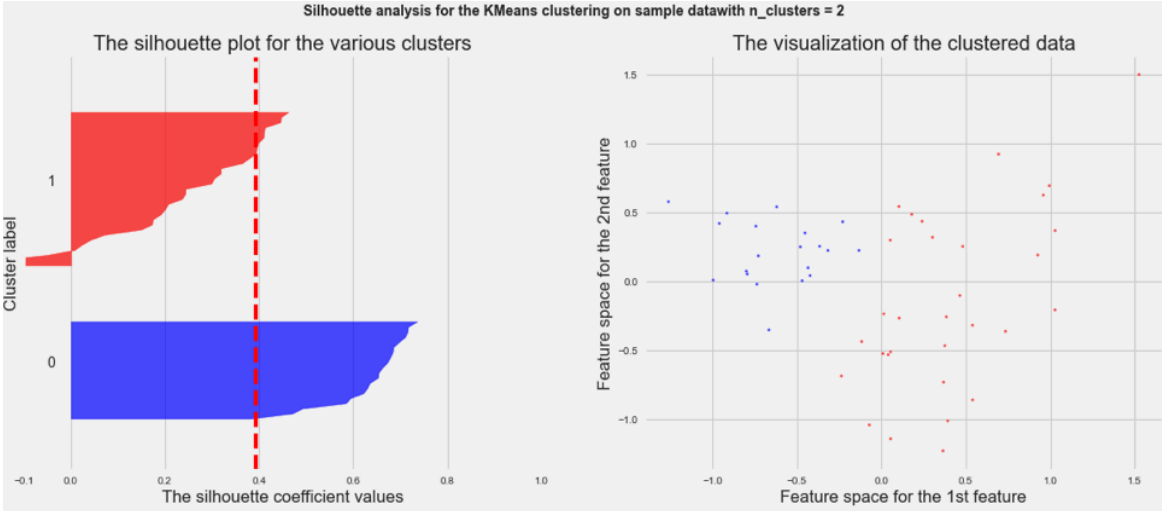
1. K-means clustering
2. Silhouette analysis
3. Boxplots (showing distribution of clusters within each S.E.F variable)

8.1 & 8.2. K-means clustering and silhouette

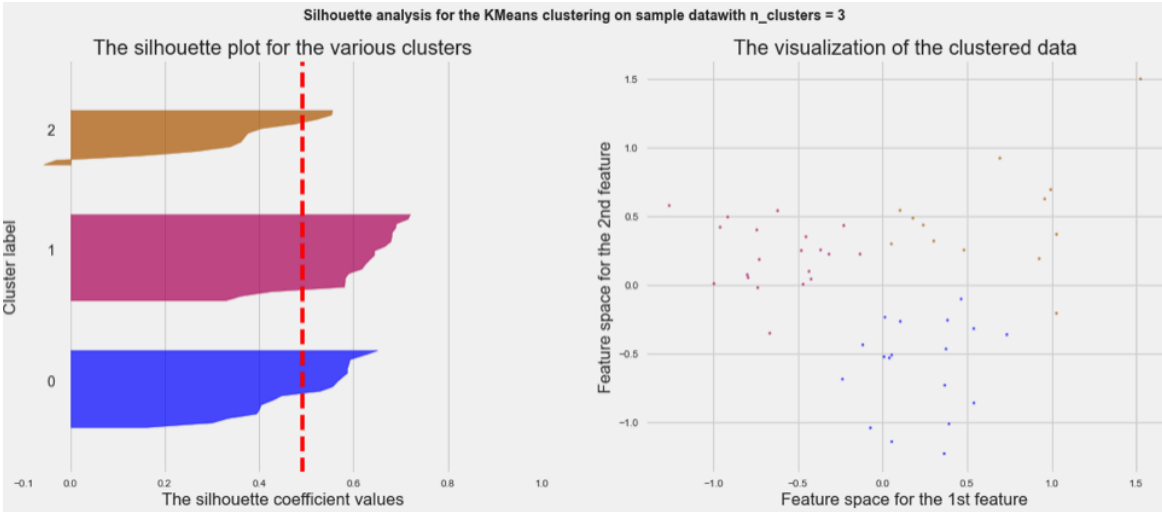
A K-means algorithm was built *(scikit-learn.org, 2021)* and computed within Python. This aimed to characterise the data by grouping S.E variables into a number of clusters for analysis.

To complement the k-means method, silhouette analysis was computed in order to find the optimum number for K (number of clusters). Whilst the silhouette could be calculated with any distance metric, the default 'euclidean distance' metric was used (it is a safe option as the choice of distance metric would have had a strong influence on the clustering results).

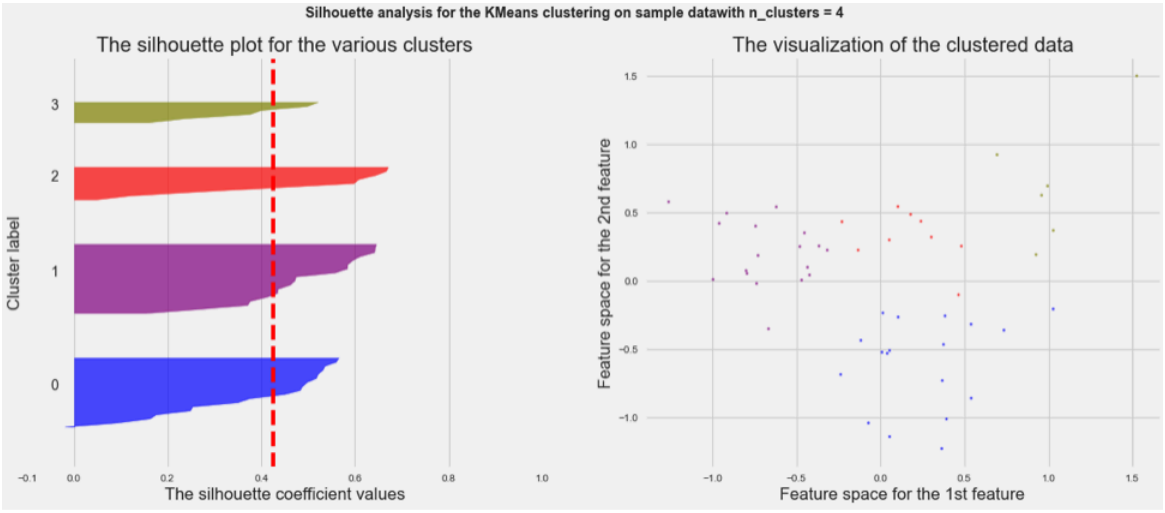
In order to prepare the S.E.F dataset for clustering, a ‘pipeline’ processor was built in python containing a ‘MinMaxScaler’ (to scale the data) and a PCA function (to reduce the high number of dimensions to **just two**). Once this pipeline had been applied, the transformed data was clustered using the ‘KMeans’ cluster package from ‘sklearn’, and a range in K from 2 to 6 clusters was used. In addition, silhouette scores were calculated for each value of K, to provide a perspective into the density and separation of the formed clusters. The silhouette and cluster plots were constructed using a series of graphical functions operating within a loop (iterating through K value). For each K, complimentary statistics, such as the average silhouette score and sizes of each cluster, were displayed. The graphical outputs are shown below:



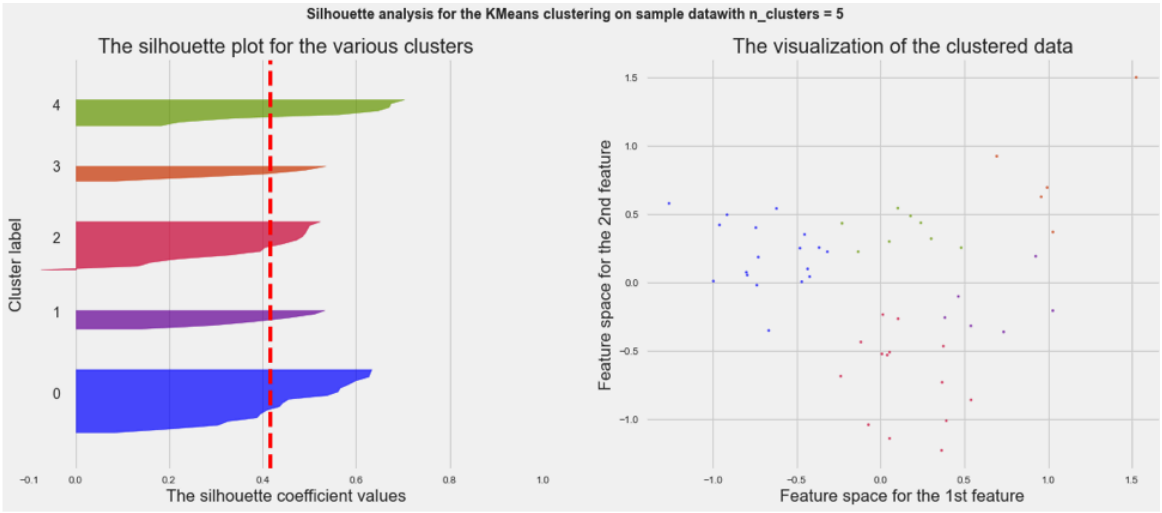
K = 2 clusters



K = 3 clusters



K = 4 clusters



K = 5 clusters

K = 6 clusters

From analysing these plots, the following results were found:

- A general trend was highlighted; as the **number of clusters increased**, the **fluctuations in cluster size decreased**.

- **No overlapping** of clusters in any of the plots.

- No plots contained clusters below the average silhouette score (red-dotted line)

In compliment to these plots, average silhouette scores *(Scikit-learn.org, 2021)* were produced:

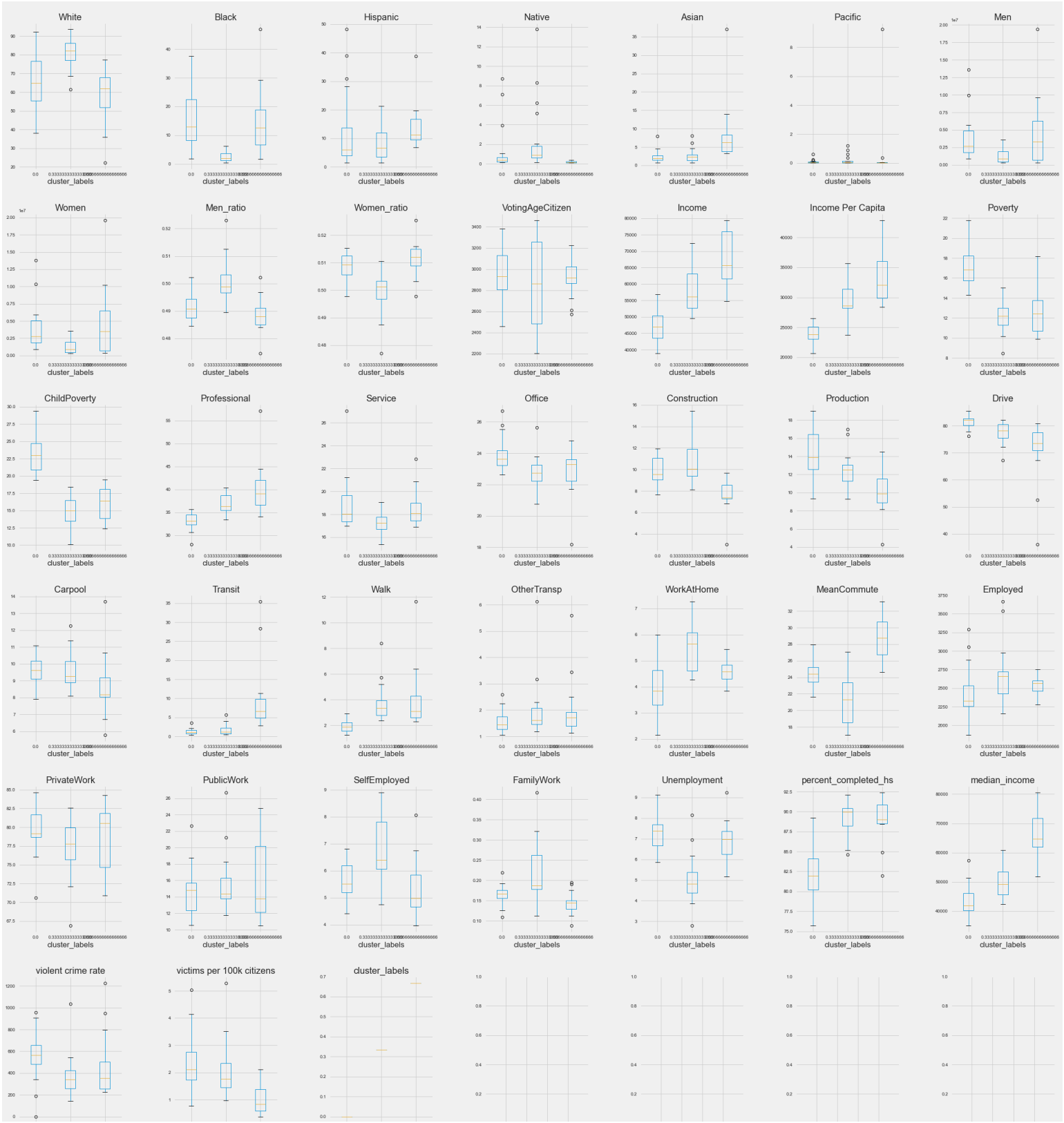
```
For n_clusters =  2 The average silhouette_score is:  0.39220219575935705
size of cluster i =  20
size of cluster i =  31
For n_clusters =  3 The average silhouette_score is:  0.4924593150307898
size of cluster i =  18
size of cluster i =  20
size of cluster i =  13
For n_clusters =  4 The average silhouette_score is:  0.4258032538632927
size of cluster i =  18
size of cluster i =  18
size of cluster i =  9
size of cluster i =  6
For n_clusters =  5 The average silhouette_score is:  0.4158880859956034
size of cluster i =  18
size of cluster i =  6
size of cluster i =  14
size of cluster i =  5
size of cluster i =  8
For n_clusters =  6 The average silhouette_score is:  0.4374244225963091
size of cluster i =  7
size of cluster i =  18
size of cluster i =  5
size of cluster i =  6
size of cluster i =  7
size of cluster i =  8
```

Average silhouette scores per K value

These stats signalled which K values realised samples far away from neighbouring clusters. A score closer to +1 indicated clear separation, whilst a score closer to -1 showed that the samples may have been assigned to the **wrong** cluster. A value of 0 highlighted that sample was on/ very close to the decision boundary between two neighbouring clusters. It was clear that the highest score (linking the optimal K value) was the score for **K = 3 clusters.**

8.3. Boxplots

The next stage was to construct box-plots *(seaborn.pydata.org, undated)* of each of the three clusters for each variable used within the S.E.F dataset. This was achieved so that the clusters that grouped certain variables could be visualised. The figure is shown overleaf:



Boxplots (K= 3 clusters) for each S.E.F variable

The analysis system applied to this figure is documented below:

- Start from cluster 1 (for each variable) and determine its position in proportion to its other 2 cluster box plots (low, medium or high).

- Repeat this process for clusters 2 and 3.

- To determine whether a box is low, medium or high, analyse factors such as **box plot medians, ranges and upper/lower quartiles**. These were compared throughout each box plot within a variable.

The results were documented in a table showing a colour coded scheme regarding **low, medium and high prevalences of variables within each cluster** (please see on next page):

High prevalence
Medium prevalence
Low prevalence

1	2	3
Black	White	Hispanic
Unemployment	Native	Asian
Violent crime rate	Men-ratio	Income
Victims per 100k citizens	HS over 25	Professional
Poverty	Self-employed	Service
Office	Work at home	Transit
Production	Family work	Walk
Service	Walk	Mean commute
Drive	Hispanic	White
White	Asian	Black
Hispanic	Income	Men-ratio
Native	Victims per 100k citizens	Violent crime rate
Men-ratio	Construction	Unemployment
Construction	Production	HS over 25
Carpool	Office	Office
Mean commute	Drive	Work at home
Family work	Carpool	Native
Private work	Black	Poverty
Self-employed	Poverty	Victims per 100k citizens
Income	Violent crime rate	Self-employed
Professional	Unemployment	Construction
Transit	Service	Production
Walk	Transit	Family work
Work at home	Mean commute	Drive
HS over 25	Private work	Carpool
Asian		

Boxplot results tabulated

5. Interpretation of results/ discussion

Interpretation of exploratory analysis results:

Race Bar plot

The results did not indicate a bias in killing white people over black, despite most victims being white. Proportions of race had to be considered.

Stacked bar chart

The rate at which black and native Americans are killed is significantly higher than that of other ethnic groups, suggesting racial bias. However, the states in which fatalities were most and least prevalent needed to be investigated before making any conclusions.

Frequency time graph

The smaller downward trend in native fatalities compared to other groups points to racial bias towards this group.

Black fatalities were the only fatalities to increase in 2019, signalling a **growing racial bias towards this group.**

Scatterplot

A straight line through the centre indicated that numbers of victims shows no fluctuation over the years.

Correlation heatmap matrix

'Native' had a high correlation to number of victims as it was observed this was also correlated highly to 'men-ratio'. Given that the vast majority of victims were male, this was not surprising. However, there may be other factors influencing this as the correlation was **surprisingly high**.

There were questions raised as to why correlations regarding carpool and construction to number of victims was fairly high.

The correlations between crime rate and blacks and victims per 100k citizens, separately were surprising. A higher positive correlation with number of victims was expected, but it signalled that other factors contributed to the fatalities. The moderate correlation to the black race suggested violent crime was more prevalent within the black community than others.

The reasons for the other S.E.F correlations were clear, the variables correlated were either very similar or a direct impact upon each other. When modelling, only one of the variables within the stated correlated pairs (‘other correlations’) were selected due to this.

Choropleth Heatmap USA

The ‘low’ results for ‘native’ highlighted that in the three states where shootings were **least prevalent,** there were few native people. There were mixed frequencies of black people within these states.

These results combined with the interpreted results from the ‘stacked bar plot’ highlighted two insights:

1. The prevalence of ‘native’ groups, compared to its overall prevalence within the USA, is extremely high within the top three states for number of victims. Hence the correlation between native and number of victims.

2. Within the top three states for fatalities, the prevalence of black people was low compared to their overall prevalence within the USA. This was ‘medium’ within the bottom three states, so **it is justifiable to think that US police may show racial bias towards black ethnic groups.**

K Means & Silhouette

The choice of K = 3 clusters for optimum modelling made sense, as there was no presence of silhouettes below the red-dotted line (av. silhouette score) and no overlapping of clusters. However, this did combat the trend where fluctuation in silhouette sizes **decreased** as K increased, as K = 3 was a low value out of 6.

Boxplots

Interpretation of significant table results:

Cluster 1:

- The variables in green signalled that these had a higher prevalence within cluster one than the other variables.

- This highlighted some similar characteristics between black communities, high unemployment rates and higher violent crime rates leading to a higher number of victims per 100k citizens.

- High poverty levels were grouped with these variables, understandable given high unemployment. This indicates the prevalence of poverty within black communities.

- Overall, this cluster indicated that **black communities may have a higher prevalence of unemployment, poverty, violent crime rate and victims per 100k citizens than other ethnic groups.**

- **These factors did not seem to be as prevalent within native communities.**

Cluster 2:

- The variables more prevalent within this cluster included white, native, and men-ratio. The heat map shows that men-ratio was negatively correlated to black communities (that there are more women to men in black communities on average).

- Native and men ratio showed a 0.61 correlation from the heat map, showing that most natives were male. This, together with prevalence of native groups in states where fatalities were highest, further indicated **less racial bias towards natives than blacks (although racial brutality towards natives was a factor).**

Cluster 3

- Within this cluster, asian, income, hispanic, service and professional showed high prevalence.

- Asian communities contained **low poverty, unemployment and violent crime rate. This supported the insight that these factors had a significant impact on the number of victims.**

6. Conclusion

This project has revealed a number of insights concerning racial brutality fromUS police towards specific ethnic groups. Basic exploratory analysis techniques highlighted higher numbers of white victims, but deeper analysis signalled the disproportionately high rates of killing of black **and** native minorities.

This anomaly was explored by examining the prevalences of these groups within certain US states; it was concluded that whilst a problem regarding disproportionate killing of native Americans was present, racial bias was not necessarily the case. However, when studying black groups in the same manner, a racial bias clearer.

The correlation of social-economic factors highlighted why this may have been the case, but ultimately, k-means clustering and box plot analysis reinforce the bias argument.

The media's coverage of the racial issue regarding black people is some-what justifiable- **however,** one of the most important insights from this project was that **poverty, unemployment and the violent crime rate were more prevalent within black communities than other ethnic groups.** Therefore, the **police may not have an overt bias towards black people**, but may unconsciously judge a black individual as a threat based on social issues within the communities they are policing. This, nonetheless, is a form of racism and shows that the **media's justification of police racial brutality towards black people, leading to the BLM protests is strongly supported by the available data.**

Word count: 6952

7. References

Almog, G., 2020. Analisys and Predictions of US Fatal Encounters. [online] Kaggle.com. Available at: <https://www.kaggle.com/guyalmog/analisys-and-predictions-of-us-fatal-encounters> [Accessed 14 April 2021].

Kerneler, K., 2020. Starter: Fatal Police Shootings in the 747af843-d. [online] Kaggle.com. Available at: <https://www.kaggle.com/kerneler/starter-fatal-police-shootings-in-the-747af843-d> [Accessed 14 April 2021].

Thompson, C., 2020. [Shootings] Understanding US Police Shootings. [online] Kaggle.com. Available at: <https://www.kaggle.com/cwthompson/shootings-understanding-us-police-shootings> [Accessed 14 April 2021].

Tkt, N., 2020. Seaborn practice - EDA. [online] Kaggle.com. Available at: <https://www.kaggle.com/nicholastkt02/seaborn-practice-eda> [Accessed 14 April 2021].

Zarin, H., 2020. EDA: Fatal Police Shootings. [online] Kaggle.com. Available at: <https://www.kaggle.com/zarinhelena/eda-fatal-police-shootings> [Accessed 14 April 2021].

En.wikipedia.org. 2021. Logarithmic scale - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Logarithmic_scale#Motivation> [Accessed 14 April 2021].

En.wikipedia.org. 2021. Scatter plot - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Scatter_plot> [Accessed 14 April 2021].

En.wikipedia.org. 2021. Shapiro–Wilk test - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test> [Accessed 14 April 2021].

En.wikipedia.org. 2021. Heat map - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Heat_map> [Accessed 14 April 2021].

En.wikipedia.org. 2021. Histogram - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Histogram> [Accessed 15 April 2021].

En.wikipedia.org. 2021. *Cramér's V - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Cramér%27s_V> [Accessed 15 April 2021].

En.wikipedia.org. 2021. Jarque–Bera test - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Jarque–Bera_test> [Accessed 15 April 2021].

En.wikipedia.org. 2021. *k-means clustering - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/K-means_clustering> [Accessed 15 April 2021].

En.wikipedia.org. 2021. *Silhouette (clustering) - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Silhouette_(clustering)> [Accessed 15 April 2021].

Factfinder.census.gov. 2015. PercentOver25CompletedHighSchool. [online] Available at: <https://factfinder.census.gov/faces/nav/jsf/pages/community_facts.xhtml> [Accessed 14 April 2021].

Factfinder.census.gov. 2017. acs2017_census_tract_data. [online] Available at: <https://factfinder.census.gov/faces/nav/jsf/pages/community_facts.xhtml> [Accessed 14 April 2021].

Factfinder.census.gov. 2017. n.d. ShareRaceByCity. [online] Available at: <https://factfinder.census.gov/faces/nav/jsf/pages/community_facts.xhtml> [Accessed 14 April 2021].

FBI.gov. 2017. Missouri. [online] Available at: <https://ucr.fbi.gov/crime-in-the-u.s/2017/crime-in-the-u.s.-2017/tables/table-8/table-8-state-cuts/missouri.xls> [Accessed 14 April 2021].

Google Developers. 2021. k-Means Advantages and Disadvantages | Clustering in Machine Learning. [online] Available at: <https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages> [Accessed 14 April 2021].

Pandas.pydata.org. 2021. pandas.DataFrame.drop — pandas 1.2.4 documentation. [online] Available at: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html> [Accessed 14 April 2021].

Scikit-learn.org. 2021. Selecting the number of clusters with silhouette analysis on KMeans clustering — scikit-learn 0.24.1 documentation. [online] Available at: <https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html> [Accessed 14 April 2021].

Scikit-learn.org. 2021. sklearn.metrics.silhouette_score — scikit-learn 0.24.1 documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html> [Accessed 14 April 2021].

Welch, A., n.d. How to Rename Columns in the Pandas Python Library. [online] Chartio. Available at: <https://chartio.com/resources/tutorials/how-to-rename-columns-in-the-pandas-python-library/> [Accessed 14 April 2021].

8. Bibliography

algorithms?, W., Pattern, G., QUIT--Anony-Mousse, H. and Coggins, D., 2021. Why do we use k-means instead of other algorithms?. [online] Cross Validated. Available at: <https://stats.stackexchange.com/questions/58855/why-do-we-use-k-means-instead-of-other-algorithms> [Accessed 1 February 2021].

Anh.cs.luc.edu. 2020. 3.1. If Statements — Hands-on Python Tutorial for Python 3. [online] Available at: <http://anh.cs.luc.edu/handsonPythonTutorial/ifstatements.html> [Accessed 14 April 2021].

Beniger, J. and L., D., 1978. "Quantitative Graphics in Statistics: A Brief History", The American Statistician, Taylor & Francis, Ltd., 32 (1): 1–11. [online] Wikipedia. Available at: <https://en.wikipedia.org/wiki/Bar_chart#:~:text=A%20bar%20graph%20shows%20comparisons,more%20than%20one%20measured%20variable.> [Accessed 14 April 2021].

Bhatt, P., 2019. 21 Pandas operations for absolute beginners. [online] Medium. Available at: <https://towardsdatascience.com/21-pandas-operations-for-absolute-beginners-5653e54f4cda#:~:text=A%20few%20key%20points%3Aa,column%20is%20really%20an%20index.> [Accessed 14 April 2021].

Brownlee, J., 2020. How to Use StandardScaler and MinMaxScaler Transforms in Python. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/> [Accessed 14 April 2021].

Clagett, M., 1968. Nicole Oresme and the Medieval Geometry of Qualities and Motions, Madison: Univ. of Wisconsin Press, pp. 85–99. [online] Wikipedia. Available at: <https://en.wikipedia.org/wiki/Bar_chart#:~:text=A%20bar%20graph%20shows%20comparisons,more%20than%20one%20measured%20variable.> [Accessed 14 April 2021].

colors, P., Tsukanov, S. and Asahi, D., 2019. Python side by side matplotlib boxplots with colors. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/52536052/python-side-by-side-matplotlib-boxplots-with-colors> [Accessed 14 April 2021].

DeZyre. n.d. How to append output of a for loop in a python dataframe? - #hackerday-dezyre. [online] Available at: <https://www.dezyre.com/recipes/append-output-of-for-loop-python-dataframe> [Accessed 14 April 2021].

Kassambara, A., 2021. Clustering Distance Measures - Datanovia. [online] Datanovia. Available at: <https://www.datanovia.com/en/lessons/clustering-distance-measures/> [Accessed 14 April 2021].

Kaufman, L. and Rousseew, P., 2021. Finding groups in data : Leonard Kaufman : Free Download, Borrow, and Streaming : Internet Archive. [online] Internet Archive. Available at: <https://archive.org/details/findinggroupsind00kauf/page/87/mode/2up> [Accessed 14 April 2021].

Matplotlib.org. 2021. matplotlib.pyplot.figure — Matplotlib 3.4.1 documentation. [online] Available at: <https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.figure.html> [Accessed 6 April 2021].

Omz-software.com. 2021. Tight Layout guide — Matplotlib 1.3.1 documentation. [online] Available at: <http://omz-software.com/pythonista/matplotlib/users/tight_layout_guide.html> [Accessed 7 April 2021].

output, m. and Celeste, K., 2016. modify pandas boxplot output. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/40125528/modify-pandas-boxplot-output/40126110> [Accessed 14 April 2021].

Overleaf.com. 2021. List of Greek letters and math symbols - Overleaf, Online LaTeX Editor. [online] Available at: <https://www.overleaf.com/learn/latex/List_of_Greek_letters_and_math_symbols> [Accessed 14 April 2021].

Pandas.pydata.org. 2021. pandas.DataFrame.groupby — pandas 1.2.4 documentation. [online] Available at: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html> [Accessed 14 April 2021].

python?, S. and patel, p., 2021. Show mean in the box plot in python?. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/29777017/show-mean-in-the-box-plot-in-python> [Accessed 7 April 2021].

Seaborn.pydata.org. n.d. seaborn.boxplot — seaborn 0.11.1 documentation. [online] Available at: <https://seaborn.pydata.org/generated/seaborn.boxplot.html> [Accessed 14 April 2021].

Statistics Solutions. 2021. Correlation (Pearson, Kendall, Spearman) - Statistics Solutions. [online] Available at: <https://www.statisticssolutions.com/correlation-pearson-kendall-spearman/> [Accessed 14 April 2021].

Thispointer.com. 2019. How to get & check data types of Dataframe columns in Python Pandas – thispointer.com. [online] Available at: <https://thispointer.com/how-to-get-check-data-types-of-dataframe-columns-in-python-pandas/> [Accessed 14 April 2021].

Thispointer.com. 2019. Python Pandas : How to display full Dataframe i.e. print all rows & columns without truncation – thispointer.com. [online] Available at: <https://thispointer.com/python-pandas-how-to-display-full-dataframe-i-e-print-all-rows-columns-without-truncation/> [Accessed 14 April 2021].

value, h. and prashanth, v., 2018. how to divide the sum of a groupby value with the count the another value. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/49514302/how-to-divide-the-sum-of-a-groupby-value-with-the-count-the-another-value> [Accessed 14 April 2021].

VanderPlas, J., 2021. Customizing Matplotlib: Configurations and Stylesheets | Python Data Science Handbook. [online] Jakevdp.github.io. Available at: <https://jakevdp.github.io/PythonDataScienceHandbook/04.11-settings-and-stylesheets.html> [Accessed 14 April 2021].

Xinapse.com. 2021. Image Masking. [online] Available at: <http://www.xinapse.com/Manual/masking.html> [Accessed 14 April 2021].

Yawintutor.com. 2021. SyntaxError: unexpected character after line continuation character – Yawin Tutor. [online] Available at: <https://www.yawintutor.com/syntaxerror-unexpected-character-after-line-continuation-character/> [Accessed 14 April 2021].

9. Appendix

	Study	Research questions	Data used	Data representation	How were missing values dealt with?	Was data categorized?	Techniques/ Methods used	Conclusions	Evaluation of methods
2	Analysis and Predictions of US Fatal Police Encounters	• What is the relationship between victims and their proportion of total population in terms of race? • Has the number of victims been on the rise for the past few years like many of the protestors/ media suggest? • How are other categorical factors in the data distributed? • Is there any difference between armed and unarmed victims? • Is there any difference between the number of victims per 100k people for each state? • Predict number of victims per 100k citizens with high accuracy...	• Fatal Police Shootings in the US (2015-2020) • Fatal Police Shootings in the US • Crime rate in the United States in 2018, by state	• Boolean • Float64 • Int64 • Object	• Nulls in 'Age' filled with median based on city & state • Nulls in 'Gender' filled with 'male' as majority male • Nulls in other categories filled with 'unknown'	✓	• Various ML classification techniques • Loading data • Cleaning data • Exploring data • Handling null values • Feature engineering • Model comparison • Model selection • Model tuning • Lasso • Random forests • XGB, LR, SVR • Enet • Light GBM • Bayes • GB • Boxplots • Percentage bar charts • Frequency line graphs • Scatter plots • Univariate histograms	• % white victims lower than their proportion of the population • Numbers seen to increase at first for white victims in 2019, but deeper analysis showed no change increase/ decrease, as opposed to what many people say • Most of the victims were not fleeing • > 70% of victims didn't have signs of mental illness • > 60% of victims are labeled as 'attackers' • > 90% are males • No difference in victim's age for specific race • Race- both armed and unarmed data are similar • Age- both armed and unarmed victim's age distribution are peaked between late teens and mid 30s • Weapon- most victims who were armed had a gun, whilst most unarmed labeled as 'Unknown' • Alaska and New Mexico are leading with number of victims via police shootings (reasons explored and concluded) • 'Gradien Booster' (model) generated best results	• Probability plots • MSE, STD, differences
3	Seaborn practice- EDA	• Are African-Americans disproportionately killed? • Are police shooting deaths increasing? • How many deaths by US states per 100,000 people? • Number of victims armed compared to unarmed? • Who were the unarmed victims?	• Fatal Police Shootings in the US (2015-2020) • US Census Demographic Data • US state populations- 2018	• Boolean (for image mask) • Boolean • Float64 • Int64 • Object	• Unknown	✓	• Regression line, grouping of data • 'WordCloud' for displaying names of victims • 'Image mask' for USA map	• African-American deaths found to be disproportionate to % population • No significant difference in number of fatal police shootings over last 5 years • USA map comprised of unarmed victim's names • New Mexico and Alaska leading with number of victims	• Visual comparison of graphs
4	Understanding US Police Shootings	• Who are the police killing? • Are African-Americans disproportionately killed? • Where do the shootings happen? • Are police shooting deaths increasing? • Are the victims armed or unarmed? • Who are the unarmed victims?	• Fatal Police Shootings in the US (2015-2020) • US Census Demographic Data • US state populations- 2018	• Boolean (for image mask) • Boolean • Float64 • Int64 • Object	• Unknown	✓	• Bar plots in exploratory data analysis • Regression line, grouping of data • Scatterplots • Splitting of dataset for armed/ unarmed analysis • Chi-squared test for if there is some difference between distribution of races/ ages of armed/ unarmed victims • 'WordCloud' for displaying names of victims	• Vast majority of victims male, peak at ages 15-25 • Both black and native people are disproportionately killed • Both white and asian people have much lower percentages than their overall percentages of the population • New Mexico and Alaska leading with number of victims • Oklahoma and Arizona are quite high ^^^ • No major change in number of shootings per month	• Chi-squared test (0.05%)
5	EDA: Fatal Police Shootings	• What is the racial distribution of victims? • ....geographical distribution of victims?	• Fatal Police Shootings in the US (2015-2020) • US state populations- 2018	• Boolean • Float64 • Int64 • Object	• Unknown	✓	• Reading data • New column 'year' introduced as data will be examined during years at some point • Exploring data by gender • DATA VISUALIZATION NOTEBOOK ONLY • Data info (data types, null count) • Percentage bar charts • Stacked bar charts • Stacked bar chart for states instead of	• Alaska, then New mexico in leading states (number of victims) • Vast majority of victims male	• Visual comparison of graphs
6	Starter: Fatal Police Shootings...	• To achieve exploratory data analysis using python	• Fatal Police Shootings in the US (2015-2020)	• Boolean • Float64 • Int64 • Object	• Unknown	✓	• Exploratory analysis using matplotlib • Histograms to display distributions of variables • Correlation matrix • Scatter and density plots • Bar charts for clumn distributions	• Most victims shot, not shot and tasered • Most victims who were armed had a gun • Most victims were male • Majority of dataset comprises of white people • Most victims had no signs of mental illness • Majority of threat levels were 'attack' • Most victims did not flee • Most events did NOT occur with a body camera	• Visual comparison of graphs
7	Final Project	**• Are the US police killing races disproportionately?** **• What US states have seen the largest increases in fatal police shootings over the years?** **• Has the number of victims been on the rise over the past few years like the media suggests?** **• Relationships between variables**	**• Fatal Police Shootings in the US (2015-2020)**	**• Boolean** **• Float64** **• Int64** **• Object**	**• Median thresholds**	✓	**• Exploratory data analysis (histograms, normality tests, correlation maps, prevalences etc)** **• Cleaning data** **• Loading data into Python** **• Model comparisons** **• Model selection** **• Model tuning** **• AUROC scores** **• Clustering** **• Random Forests (Ensemble)** **• Neural Networks** **• Transfer Learning**		**• AUROC scores**

Literature review spreadsheet

Code for execution of this project:

```
# Importing libraries and defining functions for
plotting the data
from datetime import datetime
from sklearn.preprocessing import StandardScaler,
LabelEncoder, OneHotEncoder
import matplotlib.pyplot as plt                    #
for plotting
import pandas as pd                                #
for data processing & reading
import numpy as np                                 #
for linear algebra computations
import seaborn as sns
import plotly.graph_objects as pgo
import researchpy
In [ ]:
# Reading data from excel into jupyter
sh = pd.read_csv('fatal-police-shootings-data.csv',
encoding = "ISO-8859-1")
sh
In [ ]:
# Census dataset
ce = pd.read_csv('acs2017_census_tract_data.csv',
encoding = "ISO-8859-1")
ce.rename(columns = {'State' : 'state'}, inplace =
True)
ce
In [ ]:
# Education dataset
ed =
pd.read_csv('PercentOver25CompletedHighSchool.csv',
encoding = "ISO-8859-1")
ed.rename(columns = {'Geographic Area' : 'state'},
inplace = True)
ed
In [ ]:
# Crime rate dataset
cr =
pd.read_csv('crimerate_mod.csv',encoding="ISO-8859-1")
cr.rename(columns={'State':'state'},inplace=True)
In [ ]:
# Changing state names in 'crime rate' dataset to their
abbreviations, for consistency
```

Caption

```
cr2
In [ ]:
£
cr3 = cr2.groupby(by='state',as_index=False).sum()
In [ ]:
cr3
In [ ]:
cr4 =
cr3.rename(columns={'Violent\ncrime':'violentcrimes'})
cr4
In [ ]:
cr4['violent crime rate'] = cr4.violentcrimes/
cr4.Population*100000
cr4
In [ ]:
vcr = cr4.drop(columns=['Population','violentcrimes'])
vcr.dtypes
In [ ]:
# Changing state names in 'census' dataset to their
abbreviations, for consistency
ce.state.replace({'California' : 'CA', 'Texas' : 'TX',
'Florida' : 'FL', 'New York' : 'NY', 'Pennsylvania' :
'PA',
       'Illinois' : 'IL', 'Ohio' : 'OH', 'Georgia' :
'GA', 'North Carolina' : 'NC', 'Michigan' : 'MI',
       'New Jersey' : 'NJ', 'Virginia' : 'VA',
'Washington' : 'WA', 'Arizona' : 'AZ',
'Massachusetts' : 'MA',
       'Tennessee' : 'TN', 'Indiana' : 'IN', 'Missouri'
: 'MO', 'Maryland' : 'MD', 'Wisconsin' : 'WI',
       'Colorado' : 'CO', 'Minnesota' : 'MN', 'South
Carolina' : 'SC', 'Alabama' : 'AL', 'Louisiana' : 'LA',
       'Kentucky' : 'KY', 'Oregon' : 'OR', 'Oklahoma' :
'OK', 'Connecticut' : 'CT', 'Iowa' : 'IA', 'Utah' :
'UT',
       'Nevada' : 'NV', 'Arkansas' : 'AR',
'Mississippi' : 'MS', 'Kansas' : 'KS', 'New Mexico' :
'NM',
       'Nebraska' : 'NE', 'West Virginia' : 'WV',
'Idaho' : 'ID', 'Hawaii' : 'HI', 'New Hampshire' :
'NH',
       'Maine' : 'ME', 'Montana' : 'MT', 'Rhode Island'
: 'RI', 'Delaware' : 'DE', 'South Dakota' : 'SD',
```

Caption

```
       'North Dakota' : 'ND', 'Alaska' : 'AK',
'District of Columbia' : 'DC', 'Vermont' : 'VT',
       'Wyoming' : 'WY'}, inplace = True)
In [ ]:
# Displaying information on null values within the
education dataset
ed.isnull().sum()
In [ ]:
# Discarding null values from the education (ed)
dataset
ed.percent_completed_hs=ed.percent_completed_hs.apply(p
d.to_numeric, errors='coerce')
ed=ed.dropna()
ed.percent_completed_hs =
ed.percent_completed_hs.astype(float)
In [ ]:
ed
In [ ]:
# Displaying information on columns within the census
dataset
ce.columns
In [ ]:
# Discarding 'Puerto Rico' as it does not appear within
the other datasets (consistency)
ce.drop(ce[ce.state == 'Puerto Rico'].index, inplace =
True)
In [ ]:
# Creating a population dataset (Total population by
state and men/women ratio)
pop = pd.DataFrame(ce.groupby(by='state')
[('TotalPop','Men','Women')].sum()).reset_index()
pop['Men_ratio']=pop.Men/pop.TotalPop
pop['Women_ratio']=pop.Women/pop.TotalPop
pop
In [ ]:
# Creating a dataframe for ratios of race as a percent
of the total population by state
race_ratios = pd.DataFrame({'state':[x for x in
ce.state.unique()]})
In [ ]:
# Defining function to obtain percentages of total
population by state, in terms of race
def get_share(race):
```

Caption

```
    countries = [x for x in race_ratios.state]
    share = []
    for country in countries:

share.append(((ce[race].loc[ce['state']==country]*ce.To
talPop.loc[ce['state']==country])/

(ce.TotalPop.loc[ce['state']==country]).sum()).sum())
    race_ratios[race] = share

In [ ]:
# Calling function for each race
get_share('White')
get_share('Black')
get_share('Hispanic')
get_share('Native')
get_share('Asian')
get_share('Pacific')
race_ratios
In [ ]:
# Creating a dataframe for socio-economic factor rates
for each state, using the median for voting age
citizens,
# income an d income per capita
sef = pd.DataFrame(ce.groupby(by = 'state')
[('VotingAgeCitizen', 'Income',
'IncomePerCap')].median()).reset_index()
In [ ]:
# Defining a function to return socio-economic rates
def get_rates(data):
    countries = [x for x in sef.state]
    columns = ['Poverty',
       'ChildPoverty', 'Professional', 'Service',
'Office', 'Construction',
       'Production', 'Drive', 'Carpool', 'Transit',
'Walk', 'OtherTransp',
       'WorkAtHome', 'MeanCommute', 'Employed',
'PrivateWork', 'PublicWork',
       'SelfEmployed', 'FamilyWork', 'Unemployment']
    for column in (columns):
        rate = []
        for country in (countries):
```

Caption

```
In [ ]:
# Defining function to fill nulls in age based on city
& state
def null_fill(data, col, col2, col3):
    index_nan = list(data[col]
[data[col].isnull()].index)
    for i in index_nan:
        null_fill = data[col][((data[col2] ==
data.loc[i][col2]))].median()
        med_fill = data[col][((data[col2] ==
data.loc[i][col2]) & (data[col3] == data.loc[i]
[col3]))].median()
        if not np.isnan(med_fill):
            data[col].loc[i] = med_fill
        else: data[col].loc[i] = null_fill
In [ ]:
# Calling function to fill nulls...
null_fill(df, 'age', 'state', 'city')
In [ ]:
# Filling nulls
df.race.fillna(value = 'Unknown', inplace = True)
df.flee.fillna(value = 'Unknown', inplace = True)
df.armed.fillna(value = 'Unknown', inplace = True)
df.gender.fillna(value = 'M', inplace = True)
In [ ]:
# Checking for existence of any null values after
update
df.isnull().sum()
In [ ]:
###### Exploratory Data Analysis ######
df

# 1) Plotting univariate histograms of relevant
discrete numerical data variables
# Displaying all columns within the data frame
df.columns
In [ ]:
df['age'].plot.hist()
In [ ]:
df['Income'].plot.hist()
In [ ]:
df['IncomePerCap'].plot.hist()
In [ ]:
```

Caption

```
# 2) Plotting bar graphs of categorical data variables
(excluding state, city, name and date)
df['manner_of_death'].value_counts().plot.bar()
In [ ]:
df['armed'].value_counts().plot.bar()
In [ ]:
# Improving visualisation of the plot....
dfa = pd.DataFrame(df.armed)
cdfa =
pd.DataFrame(df.armed.value_counts().reset_index().valu
es, columns=['armed', 'counts'])
Other_total= cdfa.iloc[7:93].sum()
new_row = {'armed':'Other','counts': 480}
cdfa = cdfa.append(new_row,
ignore_index=True).drop(cdfa.index[7:93])
cdfa.plot(kind='bar',x='armed',y='counts')
plt.show()
In [ ]:
df['gender'].value_counts().plot.bar()
```

df['race'].value_counts().plot.bar()

```
In [ ]:
df['signs_of_mental_illness'].value_counts().plot.bar()
In [ ]:
df['threat_level'].value_counts().plot.bar()
In [ ]:
df['flee'].value_counts().plot.bar()
In [ ]:
df['body_camera'].value_counts().plot.bar()
In [ ]:
races = pd.DataFrame({'race': ['Hispanic', 'White',
'Black', 'Native', 'Asian']})
In [ ]:
popu = ce['TotalPop'].sum()
In [ ]:
races['Share Of Population'] =
races['race'].apply(lambda x: ce.apply(lambda y:
y[x]*y['TotalPop']/popu,axis=1).sum())
In [ ]:
races['Percent Killed By Police'] =
races['race'].apply(lambda x: len(df[df['race']==x])/
len(df))*100
In [ ]:
```

Caption

```
races
In [ ]:
plotraces = races.melt(id_vars='race')
In [ ]:
plt.style.use('seaborn-pastel')
fig, ax = plt.subplots(1,1,figsize = (8,5))
sns.barplot('value', 'race',hue='variable',
data=plotraces, ax=ax )
for i in ax.patches:
    width = i.get_width()
    plt.text(4+i.get_width(), i.get_y()
+0.55*i.get_height(),
             '{:1.2f}%'.format(width),
             ha='center', va='center')
ax.tick_params(axis='both', labelsize=12)
for spine in ax.spines.values():
    spine.set_visible(False)
ax.set_title('Percent Killed vs Percent of Population')
ax.set_xlabel('')
ax.set_ylabel('Race')
ax.set_xticks([])
plt.tight_layout()
# divid epercywnt killed by share oif piouylaiton
In [ ]:
# Black diff: 191%
# Native diff: 218%
In [ ]:
df
In [ ]:
# Creating a year column in shootings database (main).
df.date = pd.to_datetime(df.date)
df['year'] = df['date'].apply(lambda x: x.year)
In [ ]:
# Plotting number of victims by race over time
g = df[(df['race']!='Unknown') & (df['race']!
='O')].pivot_table('id','year','race',aggfunc='count').
plot(marker='o')
plt.xlabel('Year')
plt.ylabel('Number Of Victims')
t = plt.title('Total Number Of Victims By Race Over
Time')
f = plt.legend()
In [ ]:
```

Caption

```
# Grouping data by month....
by_month =
df.groupby(pd.Grouper(key='date' ,freq='M')).count().re
set_index()[['date', 'id']]
by_month['date_ordinal'] =
by_month['date'].apply(lambda x: x.toordinal())
In [ ]:

In [ ]:
# Plotting this...
years = df.year.unique()
fig, ax = plt.subplots(1,1,figsize = (10,5))
sns.regplot(by_month['date_ordinal'],
by_month['id'],ci=95, ax=ax)
labels = [by_month['date_ordinal'].min() + (x * 365)
for x in range(6)]
ax.set_xticks(labels)
labels = ax.set_xticklabels(years)
ax.set_xlabel('Year')
l = ax.set_ylabel('Number Of Victims')
t = plt.title('Number Of Victims over the years')
In [ ]:
###### Correlations ######

## Cramer's V Statistic for discrete data (categorical
variables)
# Correlation between 'armed' & 'mental illness'?
# Displaying counts table...
countTable = pd.crosstab(df['armed'],
df['signs_of_mental_illness'])
countTable
In [ ]:
# Activate function for Cramer's V...
crosstab, res = researchpy.crosstab(df['armed'],
df['signs_of_mental_illness'], test = "chi-square")
crosstab
res
In [ ]:
deg = min(countTable.shape[0], countTable.shape[1]) - 1
In [ ]:
# Defining function to return nature of correlation in
4 measurements:
def int(V):
```

Caption

```
    if deg == 1:
        if V < 0.10:
            return 'negligible'
        elif V < 0.30:
            return 'small'
        elif V < 0.50:
            return 'medium'
        else:
            return 'large'
    elif deg == 2:
        if V < 0.07:
            return 'negligible'
        if V < 0.21:
            return 'small'
        if V < 0.35:
            return 'medium'
        else:
            return 'large'
    elif deg == 3:
        if V < 0.06:
            return 'negligible'
        elif V < 0.17:
            return 'small'
        elif V < 0.29:
            return 'medium'
        else:
            return 'large'
```

```
In [ ]:
V = res.iloc[2,1]
print(V)
int(V)
In [ ]:
# Cramer's V for correlation between threat level +
fleeing
crosstab, res = researchpy.crosstab(df['threat_level'],
df['flee'], test = "chi-square")
crosstab
res
In [ ]:
V = res.iloc[2,1]
print(V)
int(V)
In [ ]:
```

Caption

```
crosstab, res =
researchpy.crosstab(df['manner_of_death'], df['state'],
test = "chi-square")
crosstab
res
In [ ]:
V = res.iloc[2,1]
print(V)
int(V)
In [ ]:
df
In [ ]:
ed2 = ed.groupby('state')
['percent_completed_hs'].agg('mean').reset_index()
In [ ]:
####### Correlations between SEF #######

# Creating a dataframe containing each state's
demographic and SEF.
sdata = df.groupby('state')['id'].count().reset_index()
sdata =
sdata.merge(race_ratios,on='state').merge(pop,on='state
').merge(sef,on='state').merge(ed2,on='state').merge(vc
r,on='state')
sdata.rename(columns={'id':'Number Of
Victims','IncomePerCap':'Income Per
Capita'},inplace=True)
sdata['victims per 100k citizens']=sdata['Number Of
Victims']/(sdata['TotalPop']/100000)
In [ ]:
sdata.head()
In [ ]:
sdata.head()
In [ ]:
sdata.columns
In [ ]:
# Displaying number of victims per state.
sfig = pgo.Figure(data = pgo.Choropleth(
    locations = sdata['state'],
    z = sdata['victims per 100k citizens'],
    locationmode = 'USA-states',
    colorscale = 'Blues',
    colorbar_title = 'Fatalities'
```

Caption

```
pd.set_option('display.max_rows',None)
sdata
In [ ]:
print(sdata["Black"].mean())
print(sdata["Native"].median())
In [ ]:
# Creating a heatmap to show correlations between
socio-economic factors.
sdata_hm = sdata.drop(columns=['Men','Women','Number Of
Victims','TotalPop'])
hcorrs = abs(sdata_hm.corr()).nlargest(40, 'victims per
100k citizens').index
plt.figure(figsize = (30,20))
plt.rcParams["figure.figsize"] = (40,30)
crmap =
np.round(sdata_hm[hcorrs].corr(method='pearson'),2)
htmap = sns.heatmap(crmap, cbar=True, annot=True, cmap
= 'RdGy', yticklabels = hcorrs.values, xticklabels =
hcorrs.values)
plt.tight_layout()
In [ ]:
####### Modelling #######
### K-Means
In [ ]:
sdata.columns
In [ ]:
from sklearn.cluster import KMeans
sdata.columns;
In [ ]:
# Dropping unwanted features within the data
newdata = sdata.drop(columns=['state','Number Of
Victims','TotalPop'])
newdata
In [ ]:
# Isolating the features columns.
features = list(newdata.columns)
In [ ]:
# Obtaining the features data.
featuresdata = newdata[features]
featuresdata
In [ ]:
from sklearn.decomposition import PCA
In [ ]:
```

Caption

```
pd.set_option('display.max_rows',None)
sdata
In [ ]:
print(sdata["Black"].mean())
print(sdata["Native"].median())
In [ ]:
# Creating a heatmap to show correlations between
socio-economic factors.
sdata_hm = sdata.drop(columns=['Men','Women','Number Of
Victims','TotalPop'])
hcorrs = abs(sdata_hm.corr()).nlargest(40, 'victims per
100k citizens').index
plt.figure(figsize = (30,20))
plt.rcParams["figure.figsize"] = (40,30)
crmap =
np.round(sdata_hm[hcorrs].corr(method='pearson'),2)
htmap = sns.heatmap(crmap, cbar=True, annot=True, cmap
= 'RdGy', yticklabels = hcorrs.values, xticklabels =
hcorrs.values)
plt.tight_layout()
In [ ]:
####### Modelling #######
### K-Means
In [ ]:
sdata.columns
In [ ]:
from sklearn.cluster import KMeans
sdata.columns;
In [ ]:
# Dropping unwanted features within the data
newdata = sdata.drop(columns=['state','Number Of
Victims','TotalPop'])
newdata
In [ ]:
# Isolating the features columns.
features = list(newdata.columns)
In [ ]:
# Obtaining the features data.
featuresdata = newdata[features]
featuresdata
In [ ]:
from sklearn.decomposition import PCA
In [ ]:
```

Caption

```
))

sfig.update_layout(
     title_text = 'Victims Per 100k Citizens By State',
     geo_scope = 'usa'
)

sfig.show()
In [ ]:
sdata;
In [ ]:
# Checking for outliers within 'black' and 'native'
sns.boxplot(data=sdata['Black'])
In [ ]:
sns.boxplot(data=sdata['Native'])
In [ ]:
# Plotting histogram...
vhist = sdata['victims per 100k citizens'].plot.hist()
In [ ]:
# Plotting log histogram of 'Victims Per 100k Citizens'
to show distribution of data.
log_victims = np.log(sdata['victims per 100k
citizens'])
log_victims.plot.hist()
In [ ]:
# Checking to see if normal...
from scipy.stats import shapiro
stat, p = shapiro(log_victims)
print('stat=%.3f, p=%.3f\n' % (stat,p))
if p > 0.05:
    print('Probably Normal')
else:
    print('Probably Not Normal')
In [ ]:
# Using a better method
from scipy.stats import jarque_bera
statistic, pval = jarque_bera(log_victims)
print('statistic=%.3f, p=%.3f\n' % (statistic,pval))
if pval > 0.05:
    print('Probably Normal')
else:
    print('Probably Not Normal')
In [ ]:
```

Caption

```
pd.set_option('display.max_rows',None)
sdata
In [ ]:
print(sdata["Black"].mean())
print(sdata["Native"].median())
In [ ]:
# Creating a heatmap to show correlations between
socio-economic factors.
sdata_hm = sdata.drop(columns=['Men','Women','Number Of
Victims','TotalPop'])
hcorrs = abs(sdata_hm.corr()).nlargest(40, 'victims per
100k citizens').index
plt.figure(figsize = (30,20))
plt.rcParams["figure.figsize"] = (40,30)
crmap =
np.round(sdata_hm[hcorrs].corr(method='pearson'),2)
htmap = sns.heatmap(crmap, cbar=True, annot=True, cmap
= 'RdGy', yticklabels = hcorrs.values, xticklabels =
hcorrs.values)
plt.tight_layout()
In [ ]:
####### Modelling #######
### K-Means
In [ ]:
sdata.columns
In [ ]:
from sklearn.cluster import KMeans
sdata.columns;
In [ ]:
# Dropping unwanted features within the data
newdata = sdata.drop(columns=['state','Number Of
Victims','TotalPop'])
newdata
In [ ]:
# Isolating the features columns.
features = list(newdata.columns)
In [ ]:
# Obtaining the features data.
featuresdata = newdata[features]
featuresdata
In [ ]:
from sklearn.decomposition import PCA
In [ ]:
```

Caption

```
## Implementing a K-Means clustering pipeline (using
MinMaxScaler when we cannot assume that feature shapes
## follow a normal distribution).

from sklearn.metrics import adjusted_rand_score,
silhouette_score
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler

preprocessor = Pipeline(
    [
        ('scaler', MinMaxScaler()),
        ('pca', PCA(n_components=2,
random_state=None)),
    ]
)
In [ ]:
# Building the K-Means pipeline with user-defined
arguments.
clusterer = Pipeline(
[
    ('kmeans',
     KMeans(
     n_clusters=3,
     init='k-means++',      # k-means ++ ensures
centroids initialized with some distance betweem them.
     n_init=10,
     max_iter=300,
     random_state=None,),)

])
In [ ]:
# Extending the pipeline...
pipe = Pipeline(
[
    ('preprocessor', preprocessor),
    ('clusterer', clusterer)
])
In [ ]:
# Applying the pipeline to the scaled_features data
pipe.fit(featuresdata)
In [ ]:
```

Caption

```
# Evaluating performance by calculating the silhouette
coefficient.
preprocessed_data =
pipe['preprocessor'].transform(featuresdata)
predicted_labels = pipe['clusterer']['kmeans'].labels_
silhouette_score(preprocessed_data, predicted_labels)
In [ ]:
## Silhouette 2

from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples,
silhouette_score

import matplotlib.pyplot as plt
import matplotlib.cm as cm
import numpy as np
In [ ]:
print(__doc__)
In [ ]:
range_n_clusters = [2,3,4,5,6]
In [ ]:
for n_clusters in range_n_clusters:

    # Creating a subplot with 1 row, 2 columns.
    fig, (ax1,ax2) = plt.subplots(1,2)
    fig.set_size_inches(18,7)

    # 1st subplot = silhouette plot
    # Silhouette coefficient ranging from -0.1 to 1.
    ax1.set_xlim([-0.1,1])

    # (n_clusters+1)*10 for inserting space between
silhouette plots of each of the clusters.
    ax1.set_ylim([0, len(preprocessed_data) +
(n_clusters+1)*10])

    # Initialize the cluster with n_clusters value and
a random generator seed of 'None'.
    clusterer =
KMeans(n_clusters=n_clusters,random_state=None)
    cluster_labels =
clusterer.fit_predict(preprocessed_data)
```

Caption

```
    # Silhouette score gives average value for all
samples.
    # Provides a perspective into density and
separation of the formed clusters.
    silhouette_avg =
silhouette_score(preprocessed_data, cluster_labels)
    print("For n_clusters = ", n_clusters,
          "The average silhouette_score is: ",
silhouette_avg)

    # Compute silhouette scores for each sample.
    sample_silhouette_values =
silhouette_samples(preprocessed_data, cluster_labels)


    y_lower = 10
    for i in range(n_clusters):

        # Aggregate the silhouette scores for samples
belonging to cluster i, and sort them.
        ith_cluster_silhouette_values = \
        sample_silhouette_values[cluster_labels == i]

        ith_cluster_silhouette_values.sort()

        size_cluster_i =
ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i
        print("size of cluster i = ", size_cluster_i)

        color = cm.brg(float(i) / n_clusters)
        ax1.fill_betweenx(np.arange(y_lower, y_upper),
                          0,
ith_cluster_silhouette_values,
                          facecolor=color,
edgecolor=color, alpha=0.7)

        # Label the silhouette plots with their cluster
numbers at the middle,
        ax1.text(-0.05, y_lower + 0.5 * size_cluster_i,
str(i))
```

Caption

```
        # Compute the new y_lower for the next plot.
        y_lower = y_upper + 10      # 10 for the 0
samples.

    ax1.set_title("The silhouette plot for the various
clusters")
    ax1.set_xlabel("The silhouette coefficient values")
    ax1.set_ylabel("Cluster label")

    # The vertical line for the av. silhouette score of
all the values.
    ax1.axvline(x=silhouette_avg, color="red",
linestyle="--")

    ax1.set_yticks([])     # Clearing the y_axis labels
/ ticks.
    ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1.0])

    # 2nd plot showing actual clusters formed.
    colors = cm.brg(cluster_labels.astype(float) /
n_clusters)

ax2.scatter(preprocessed_data[:,0],preprocessed_data[:,
1],marker='.',s=30,lw=0,alpha=0.7,c=colors,edgecolor='k
')

    if range(0,3) == range(n_clusters):
        clu_dataframe =
pd.DataFrame(data=(cluster_labels.astype(float) /
n_clusters))
        clu_dataframe.columns=['cluster_labels']
        data_w_k3 = featuresdata.join(clu_dataframe)
        data_w_k3

    if range(0,4) == range(n_clusters):
        clu_dataframe_2 =
pd.DataFrame(data=(cluster_labels.astype(float) /
n_clusters))
        clu_dataframe_2.columns=['cluster_labels']
        data_w_k4 = featuresdata.join(clu_dataframe_2)
        data_w_k4
```

Caption

```
    ax2.set_title("The visualization of the clustered
data")
    ax2.set_xlabel("Feature space for the 1st feature")
    ax2.set_ylabel("Feature space for the 2nd feature")

    plt.suptitle(("Silhouette analysis for the KMeans
clustering on sample data"
                  "with n_clusters = %d"
%n_clusters), fontsize=14,fontweight='bold')

plt.show()
In [ ]:
## Interpretation of results from silhouette analysis:
# If there is a presence of clusters with below average
silhouette scores, or wide fluctuation in size of
# silhouette plots.

# Plot shows no below average silhouette scores.
# Tend to be wider fluctuations in size of silhouettes
as n_clusters decrease.
# Av. silhouette score shows quality of clustering;
# higher score to 1 = better.
# Optimal: n = 3 clusters (ss = 0.4951...)
# Worst: n = 4 clusters (ss = 0.4443...) + wide
fluctuation in size of the two silhouettes.

# 2 cluster solutions identified: k = 3, 4
In [ ]:
data_w_k3
In [ ]:
sa = list(data_w_k3.columns.values)
sa
In [ ]:
## k = 3, boxplots
import matplotlib.pyplot as plt

fig,axes = plt.subplots(6,7)
plt.rcParams["figure.figsize"] = (40,40)

for i,el in enumerate(list(data_w_k4.columns.values)
[:]):
    a = data_w_k3.boxplot(el, by="cluster_labels", ax =
axes.flatten()[i])
```

Caption

```
plt.subplots_adjust(bottom=0.1,right=0.9,top=0.95,hspac
e=0.3,wspace=0.5)
plt.show()
```

Caption