LEEDS BECKETT UNIVERSITY
United Kingdom

Data Analytics and Visualization (COMP757)

Great Britain Decade of Action for Road Safety.

From 1999 to 2021

MSc Data Science
c7324621
Wijesinghe Mudiyanselage Dilendra Sajini

Abstract

In 2010, the United Nations Decade of Action for Road Safety was launched in order to reduce the rising trend in traffic deaths. The United Kingdom is one of the member countries with a road safety plan from 1999 to 2021, and the aim is to ensure that Britain remains a world leader in road safety. According to the actions taken by the government, the number of road deaths has been declining over the past few years and has a good record of reducing casualties. Therefore, motorcyclist, car driver, pedal cyclist, and pedestrian deaths have decreased over the years.

This research study will conduct descriptive and inferential analysis at many levels and show the success of the UK government's investment in road safety from 1999 to 2021 based on a data set from the Department of Transportation's traffic statistics. Additionally, machine learning models are supplied for assessing degrees of road fatality risk based on a variety of criteria. When the dataset is analysed, four levels of causality risk classification (low risk, medium risk, high risk, and extremely high risk) can be performed based on road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, highway). As a conclusion, constant development should be necessary while increasing traffic and population trends in the future to ensure a safer environment on the road. With the foregoing research, the report identifies distinct priorities and areas for improvement.

Keywords: Greate Britain Road Safety, Descriptive Statistics, Inferential Statistics, Machine Learning, Deep Learning, T-test, Anova, NN, Regression, KNN, Decision Tree, K-means, Neural Network, Supervised, Unsupervised

Table of content

1. Introduction

Roads are essential to our everyday life for driving, riding, walking or travelling. The rising number of traffic fatalities and injuries is a major public health issue. As a road user there is always a risk of being killed or injured. Great Britain has the fifth-lowest rate of road causality in the world. The number of road deaths is over three times in the United States compared with Britain. The road deaths in Britain have been reducing over the past years because of the actions taken by the government such as safer infrastructure, new vehicle technologies and shifting social attitude (Department of Transport, 2015).

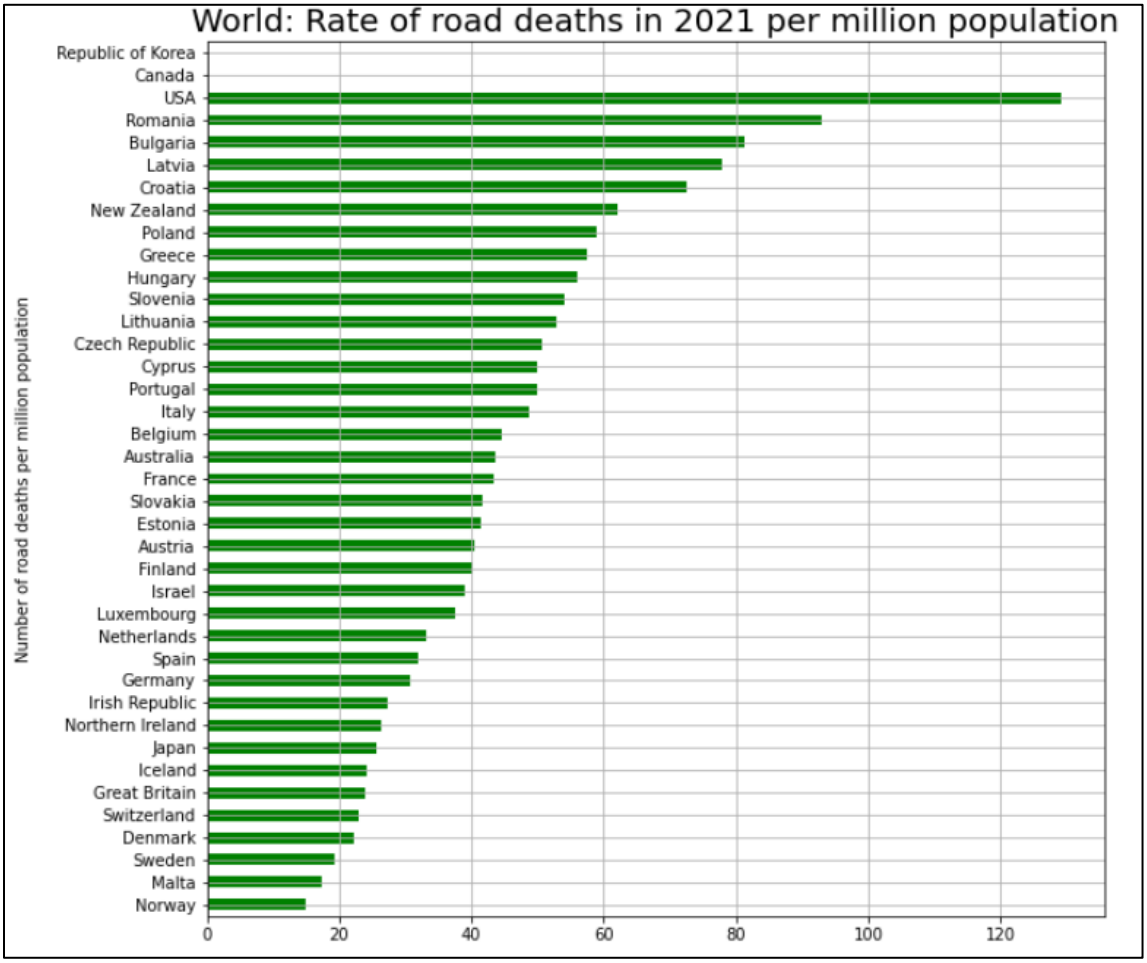


Figure 1.1 World: Road death rates in 2021 per million populations Department of Transport (2015).

1.1 Problem Statement

Examine the success of the United Kingdom government's investment in national road safety by analyzing and monitoring the road causalities during the period of 1999 -2021. The importance of identifying current and future trends that leads to find distinct priorities and areas for improvement.

2.2 Scope

The scope of this research is limited to the analysis of reported road causalities in Great Britain between 1999 and 2021. The research data is downloaded from the customise tool and that provided by Department for Transport traffic statistics.

- RAS0404: International comparisons (GOV.UK, 2021)
- RAS0202: Gender and age group (GOV.UK, 2021)
- RAS0501: Drivers involved by gender, age and road user type (GOV.UK, 2021)
- Reported road casualties, Great Britain (roadtraffic.dft.gov.uk., n.d.)
- Motor vehicle traffic (vehicle miles) by local authority in Great Britain (GOV.UK, 2018)
- United Nations - World Population Prospects (MacroTrends, n.d.)
- Department for Transport statistics (GOV.UK, 2021)

1.3 Rationale

The UK government invests in national road safety for three main reasons. The resulting reduction in society's losses due to traffic accidents will relieve pressure on the NHS and emergency services, keep traffic moving, and hence keep continue on economy expanding. Road traffic collisions are estimated to cost the UK economy more than £16.3 billion each year. It results in significant economic losses for injured people, their families, and the country as a whole. The cost of treatment, as well as the loss of job contribution for individuals and their family members who need to care for the injured, collectively make up this cost. Overall, there is a huge budget allocated for preventing road collisions and fatalities. But it is important to compare the value of the benefits of road improvements and road safety measures to their costs and select the priority decisions that provide the best value for money.

1.4 Aim

The aim of the study is to conduct a statistical assessment of road fatalities in the United Kingdom between 1999 and 2021. This study uses a secondary datasets from the Department for Transport and evaluates the effects of the government's pledge and subsequent efforts to improve national road safety.

1.5 Research Objectives

Broad Objectives

1. To conduct a literature review and investigate how the United Kingdom's road fatality rates compare to other countries around the world.
2. To perform a literature review and investigate Great Britain's efforts to reduce road-related injuries and deaths.
3. To identify and visualise the impact of the actions taken by the government on road casualties.
4. To create predictive machine learning models for the level of causality risk based on various factors.

To achieve the above-mentioned broad objectives, the report categories those into four narrow objective stages based on the types of analysis to be used. In each stage, try to answer different business questions based on a selected dataset. See Appendix 1 for the research questions addressed by each level.

1.5.1 Level 01: Descriptive statistics analysis

In Level 01 analysis, the research is going to answer the questions by exploring data with graphs and numerical summaries such as mean and interquartile range. It exploring relationship and the patterns but does not draw conclusions beyond what the data already shows.

1.5.2 Level 02: Inferential Statistics Analysis

Level 02 analyses the research that is going to answer the questions by exploring relationships and patterns between interesting features of the dataset. Inferential statistics allow conclusions to be drawn that go beyond the data available to the population from the sample. In this study, rather than using sample data, the entire population will be analyzed, yielding direct population summaries. There are two main types of inferential statistical methods: hypothesis testing and regression analysis. The chosen test types may vary with the variable type and number of samples used for the analysis.

2.5.3 Level 03: Machine Learning

In Level 3, the research creates machine learning algorithms or models based on historical data and makes forecasts. Based on labeled outputs, supervised learning algorithms are divided into two types: classification and regression. Unsupervised learning does not provide labeled output and instead seeks patterns in incoming data such as clustering.

2.5.4 Level 04: Deep Learning

In Level 04 analyses the research going to design deep learning algorithms or models based on given historical data and predicts the behavior. Classical machine learning approaches require the model to be trained on inputs with classified features and labeled output. However, deep learning automates feature extraction and eliminates the need for input for classified features.

2. Literature Review

2.1 Business Understanding

2.1.1 Importance of Road Safety

According to the World Health Organization, around 1.3 million people die each year as a result of road accidents. Between 20 and 50 million people suffer non-fatal injuries, while others become disabled. WHO described the trend of road traffic accidents on road safety in 2013 (World Health Organization, 2013), and continued to publish a status report in subsequent years (World Health Organization, 2015). According to that by 2030 road traffic deaths will become the fifth leading cause of death unless urgent action is taken. Most often road accidents and road causalities increase as economic development increases in a country. These injuries cost around 3% of the economies of low and middle-income countries. It results in significant economic losses for injured people, their families, and the country as a whole. The cost of treatment, as well as the loss of job contribution for individuals and their family members who need to care for the injured collectively make this cost (*Road safety*, n.d. ; Wegman et al., 2000).

2.1.2 Decade of Action for Road Safety

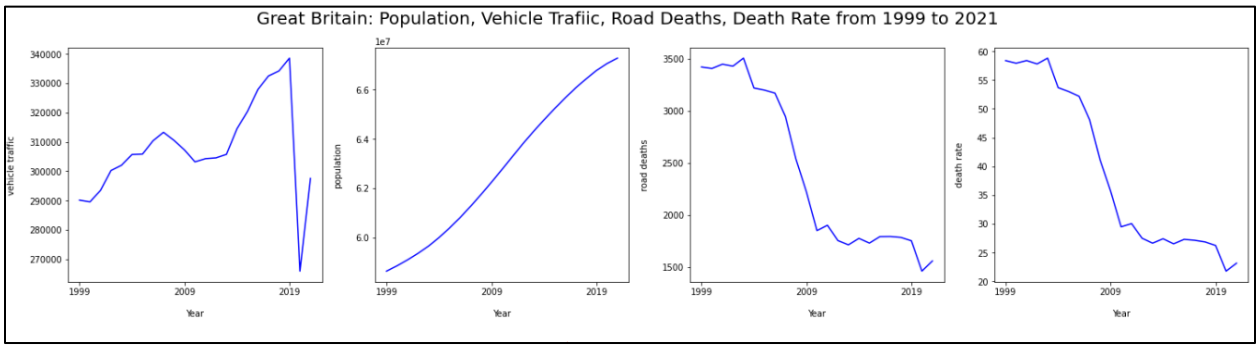
The United Nations General Assembly published a plan, named Decade of Action for Road Safety, in 2010 to achieve road safety goals from 2011 to 2020 (Nations, n.d.). United Kingdom (Great Britain) is one of the member states that follow Decade of Action for Road Safety. The vision is to ensure that Britain remains a world leader on road safety. The plan is published in government web site (Department of Transport, 2011; Department of Transport, 2013; International Traffic Safety Data and Analysis Group, 2021). Collision avoidance is an example of a concept that has the ability to protect all road users when things go wrong. The Government of the United Kingdom evaluates the framework action plan and the progress update up to September 2019 (Department of Transport, 2020). According to that most significant factors that drive road causalities are mentioned bellow,

- The distance people travel.
- The transport mode the people use.
- Behaviour of drivers, riders and pedestrians uses.
- The groups of people using the road. Ex: newly qualified or old drivers, age groups of drivers
- External factors such as weather or road type.

- The area type that people use to travel. Ex: urban or rural areas.
- Drivers drive speed

2.1.3 Great Britain: Current Status

The fundamental reason for this increase is that traffic levels rise in parallel with the economy. More incidents occur as traffic levels and the economy increase. This will continue until a key point in economic development is reached. Better training, vehicle standards, enforcement, and engineering all begin to dominate at that time to mitigate the effect of more traffic. While a result, bellow Figure 2.1 depicts even as traffic levels (vehicle miles) continue to rise, the number of incidents and fatalities begins to fall.



Great Britain: Population, Vehicle Traffic, Road Deaths, and Death Rate from 1999 to 2021

Here is the United Kingdom government's approach to road safety (Department of Transport, 2015):

1. Safer learning and road behaviours
2. Better testing and licensing
3. Increased road user awareness
4. Safer vehicles and equipment
5. Fairer and more responsive insurance
6. More intelligent and effective enforcement
7. Incentivising and Supporting Others

2.1.4 Machine Learning in Road Safety

There has a lengthy history of investigating numerous contributing elements in accidents. These factors are typically associated with traffic and road features, drivers and other road users, vehicles, and the environment. The traffic and road features that may influence road accidents are traffic flow and speed, as well as road geometry. Road users, their behavior, such as seat belt use, alcohol consumption, age, and the impact of passengers on drivers. Also, different road users may predict different accident

severity outcomes; for example, a male may be more vulnerable to accidents than a female due to risk taking attitudes (Wang et al., 2013). Machine learning technology has recently improved and is now widely used in road safety. Some of the studies used to provide crash prediction models (CPM) (Silva et al., 2020) and simulation models (Young et al., 2014) using different machine learning approaches such as nearest neighbour classification, decision trees, genetic programming, support-vector machines, and artificial neural networks. This analysis demonstrates the clear superiority of ML-based models over statistical ones.

2.2 Standardized Process

According to the research, there has been a lot of work and effort put into developing standardized data mining (DM) applications. KDD (Knowledge Discovery in Databases), SEMMA (Sample, Explore, Modify, Model, and Assess), and CRISP-DM (Cross-Industry Standard Process for Data Mining) are three of the most popular ones. The KDD technique is typically used to extract knowledge from database data after preprocessing, subsampling, and modification. It is an iterative process that necessitates a thorough understanding of the application domain, applicable past knowledge, and end objectives. DM is a stage in the nine-stage KDD process (Fayyad et al., 1996). The SEMMA is a simple, five-stage method that is easy to understand. It is separate from the DM tool of choice and is coupled to the SAS Enterprise Miner software. Despite the fact that the CRISP-DM technique is independent of the DM's preferred tool, it is tied to the SPSS Clementine software (Azevedo & Santos, 2008; Clark, 2018; Wirth & Hipp, 2000).

For this study, KDD was chosen as the typical procedure for implementation. Because others necessitate specific software, it will not provide significant improvement on small-scale projects and will require significant effort, cost, and time (Wirth & Hipp, 2000). The literature related to the KDD process is described in Section 3 while describing the current study content.

3. Methodology

3.1 The KDD Process

Over the last two decades, the discipline of DM has achieved tremendous success, both in terms of broad-ranging application results and scientific growth and knowledge. The computerised process of collecting previously undiscovered and vital actionable information and knowledge from a database (DB) that may then be used to make critical decisions is known as data mining. As a result, DM is also known as knowledge discovery in databases (KDD).

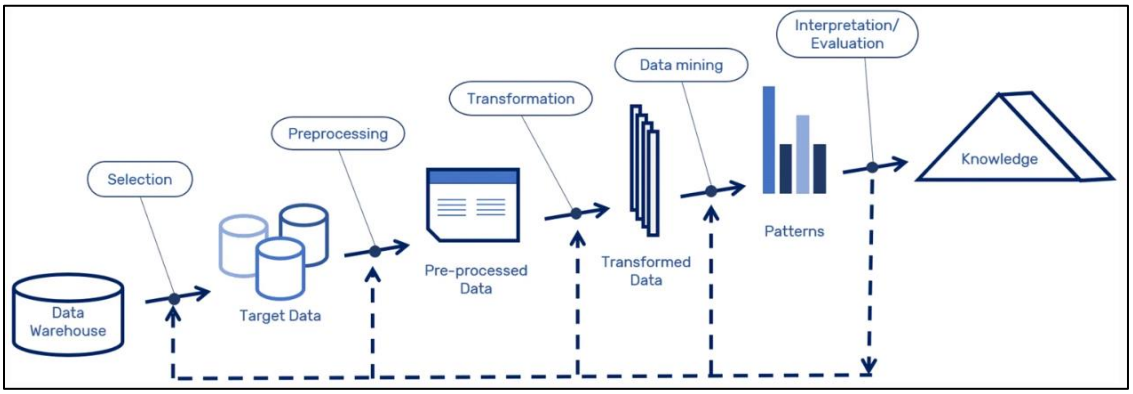


Figure 3.1: KDD process (Rotondo & Quilligan, 2020)

This report follows the knowledge discovery process that is iterative and interactive with nine steps. (Brachman & Anand, 1996; Fayyad et al., 1996).

Application Domain

According to the authors interest the analysis is done on Great Britain Road Safety domain. As the final goal the research need to fulfill the aim and research objectives in section 2.4 and section 2.5 respectively. As mentioned in Section 2.2, the scope of the research is limited to the analysis of reported road causalities in Great Britain between 2004 and 2021.

Selecting Target Data Set

According to the research, a number of secondary datasets can be found for the statistical assessment of road deaths in Great Britain. For this study, the author chose the Department of Transportation as the most trustworthy source. It was able to filter necessary knowledge from the data source using a customized tool given by the Department of Transportation's Bureau of Traffic Statistics. The dataset

was not totally row and has been preprocessed to some extent and provides a summary view for future study. Furthermore, for in-depth examination at the second and third levels of analysis, extra data is used. Overall, five data sets were employed, and due to the nature of some of the data sets, it was difficult to integrate them together. Therefore, treat them as separate datasets and continue with the below analysis.

Cleanse and Preprocess Data

As mentioned in Section 3.1.2, the data were already cleaned, and no further cleaning was required for the analysis. There were some missing data points, but they are irrelevant for the purposes of this analysis, so ignore them. Data column names and alters for meaningful names. Simplify the data sets by removing unwanted variables and grouping based on different variables.

Data Reduction and Transformation

Machine learning models require all input and output variables to be numeric. Therefore before fit into the model categorical data need encoding. Ordinal Encoder converts features, whereas Label Encoder converts target variables. Therefore data transformation done for several steps using sklearn preprocessing module because the dataset contains many categorical variables. The study does not use any dimensionality reduction because the number of attributes in the data set is limited and useful.

Data Mining Methods

When choosing a data mining method to answer a specific business question, it depends on three main factors. The first consideration is whether the data is categorical (qualitative, non-parametric) or numerical (quantitative). Second, the number of test samples used and the number of measurements taken in each sample. The third is the purpose of the question. It can be a test against the hypothesized value, comparing two statistics, or looking for a relationship. To assess the strength of a relationship between two categorical features, chi-square tests and two numerical features with correlation are preferred. Correlation is a parametric test, and chi-square is a non-parametric test. The chi-squared test and regression are similar because both are used for identifying relationships between two variables. Hypothesis testing is also required to investigate relationships among multiple attributes. The t-test is used for categorical and numerical features, and an ANOVA is used for more than two features. Supervised learning, unsupervised learning, and deep learning algorithms are used to answer questions about future predictions (Douglas, 2019).

As described in Section 2.5, the study continues with four levels of analysis. Each level describes which data mining method is suitable for answering the question. Descriptive analysis is used to make raw data more meaningful by visualising it with graphs and numerical summaries. And also, it is important to identify outliers and data distribution.

Data Mining Algorithms

As describe in 3.1.5 choosing data mining algorithms based on variable types are more important. Deciding algorithm for hypothesis analysis or statistical analysis is depending on how many categorical or numerical variables is use for the analysis. Supervised learning, unsupervised learning, and reinforcement learning are three machine learning paradigms. Supervised learning algorithms are classified into two types based on labeled outputs: classification and regression. Classification is used to assign data to different categories. Sample classification techniques are K-nearest neighbor, decision trees, support vector machines and random forest etc. Regression techniques are used to understand the relationship between two variables. Linear regression, logistic regression and polynomial regression are the most popular regression techniques. Unsupervised learning does not have labeled output and only aims to find patterns based on input data. There are three main types of unsupervised learning models, clustering, association, and dimensionality reduction. K-means grouping similar data point depend on their similarity. Market basket analysis uses to find relationships between variables depend

on different rules. Dimensionality reduction is the process of reducing the number of features in a dataset while keeping its integrity intact. It is commonly employed in data pre-processing phases (Alzubi et al., 2018; Douglas, 2019).

For machine learning K-Nearest Neighbors classification, linear regression, multiple linear regression techniques are used for modeling and prediction. K-means clustering can use for classifying unlabeled data and attempt to learn structures inside the data and create clusters based on data structure similarities.

Neural network use as a deep learning algorithm and its capable of providing high accuracy models for predicting different variables using other set of variables. Deep learning, or "deep neural networks," can be supervised or unsupervised. Deep learning neural network models have more than three layers and are classified into three types: artificial neural networks (ANN), convolutional neural networks (CNN) (Arel et al., 2020), and recurrent neural networks (RNN) (RNN). For restricted dataset analysis, ANN is dominating and may be utilized with tabular, picture, and text data. It is also known as feed-forward NN and multi-layer perceptron because the inputs are only processed forward (MLP). NN is the most basic variant. CNN is mostly used for image and video processing, and a large amount of training data is necessary for efficient operation. RNN is used for text, audio, and time series prediction. Back propagation is used to improve prediction (Jara, 2022).

Evaluation

Not all machine learning algorithms are designed to solve every problem. Therefore, it is important to find the best algorithm for extracting the data for different questions. During Section 4, it will summarize the evaluation for every solution based on different evaluation matrices. In Section 5, reports provide an overall evaluation with respect to the research objectives. This phase summarizes the overall analysis's ability to achieve business objectives. This step identifies areas for improvement and solutions if they are lacking. Using a feature selection algorithm or increasing the dataset size may improve model accuracy and reduce loss and error.

Patterns of Interest

At the initial point descriptive analysis depicts the patterns in pictorial view and inferential statistics identifies the relationship between the features. The uncovered patterns from these two levels can identify using machine learning techniques such as classification, regression, trees and clustering techniques. And based on identified patterns it is possible to predict future trends by designing predictive models.

Discovered Knowledge

When considering Great Britain causality data, provide conclusions about United Kingdom causality trends and patterns, as well as forecast some future possibilities. Related knowledge is described as "summery highlights" in each level of analysis and future recommendations during section 6.

4. Results and Discussion

4.1 Level 01: Descriptive statistics analysis

Data Set 1:

Title: RAS0404: International comparisons

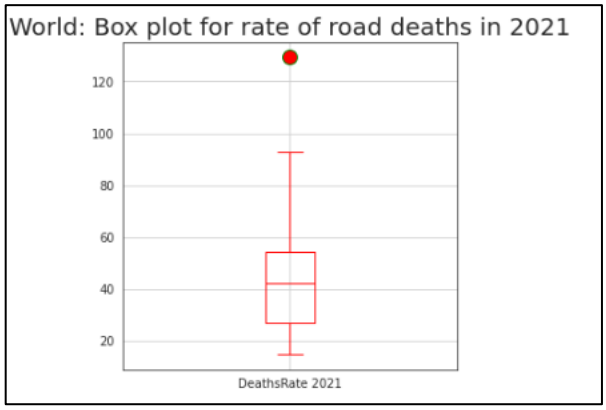
	Country	1999	2000	2001	2002	2003	2004	2005	2006	2007	...	2013	2014	2015	2016	2017	2018	2019	2020	2021	DeathsRate 2021
0	Great Britain	3423	3409	3450	3431	3508	3221	3201	3172	2946	...	1713	1775	1730.0	1792.0	1793.0	1784	1752	1460	1558.0	23.804771
1	Northern Ireland	141	171	148	150	150	147	135	126	113	...	57	79	74.0	68.0	63.0	55	56	56	50.0	26.293219
2	United Kingdom	3564	3580	3598	3581	3658	3368	3336	3298	3059	...	1770	1854	1804.0	1860.0	1856.0	1839	1808	1516	1608.0	23.875032
3	Austria	1079	976	958	956	931	878	768	730	691	...	455	430	475.0	432.0	413.0	409	416	344	362.0	40.440059
4	Belgium	1397	1470	1486	1306	1214	1162	1089	1069	1067	...	723	727	755.0	637.0	620.0	604	644	499	516.0	44.665212

Figure 4.1 World: Rate of road deaths in 2021 per million populations

See **Appendix 2** for more details.

DATA CLEANING AND PRE-PROCESSING:

Some cells contain no information. However, empty data is irrelevant for the purposes of this analysis therefore ignore them. The data set includes information for the United Kingdom, Northern Ireland, and Great Britain. The United Kingdom is formed by the union of Great Britain and Northern Ireland. As a result, the data for the United Kingdom has been removed. No dimensionality reduction applied. Further analysis based on Data Set 1 clearly displays the death rate in 2021, with the USA becoming an outlier because of its high death rate. It may be because of the high population in the USA (331.9 million) compared to Great Britain's (67.2 million). As a result, it contains valuable information and should not be removed.



	DeathsRate 2021
count	37.000000
mean	45.574082
std	23.279649
min	14.792024
25%	27.337343
50%	41.528918
75%	54.105183
max	129.303431

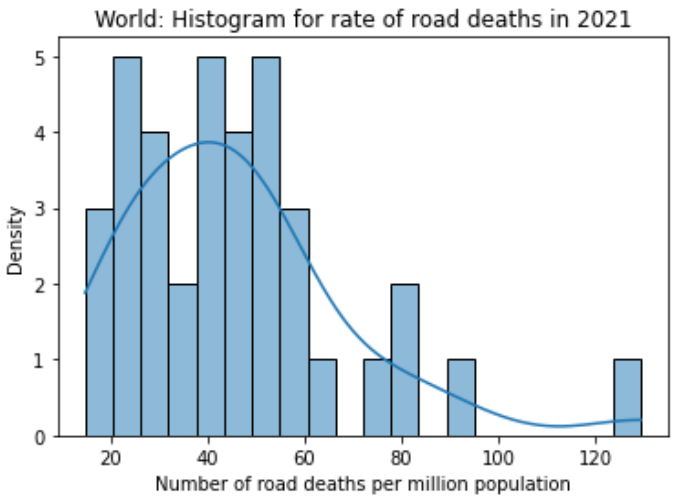


Figure 4.2 World: Box plot, summery and histogram for rate of road deaths in 2021

The world rate of deaths in 2021 has a positive skew distribution and a positive kurtosis value (a skew value of 1.53 and a kurtosis value of 3.5). Here, the median value of the death rate (41 road deaths per million people) will be a typical death rate value and give a better indication of the location of the distribution. Because the outlier has no effect on the median.

RESEARCH QUESTIONS AND RESULT EVALUATION:

1. Where does the United Kingdom rank in the world in terms of road causalities in 2021?

The United Kingdom has some of the finest road safety records in Europe and the globe. It has the fifth-lowest rate of road deaths in 2021 (Figure 1.1) and there has been a falling trend in recent years due to the government's safety initiatives (Figure 4.3). Otherwise, the expected truth is that as the population and vehicle traffic increase, so will the number of road fatalities. The actions taken by the government, such as safer infrastructure, new vehicle technologies, and a shifting social attitude has affect for this trend.

2. What is the trend in the death rate in the United Kingdom between 1999 and 2021?

Figure 4.3 Great Britain: Number of road deaths from 1999 to 2021

Since 2003, the long-term trend in the number of casualties in reported road accidents has been declining. Between 1999 (3423) and 2021 (1558), the number of fatalities decreased by 54%. In 2011, there was a 2.7 percent increase in fatalities compared with 2010 (1850), and this is the first increase since 2003. It was recognized from literature that sustained periods of snow and ice in the first and fourth quarters of 2010 contributed to the highest ever annual fall in 2010. Comparable periods of bad weather were not seen in 2011, and this is a main factor in the increase in road fatalities between 2010 and 2011 (International Traffic Safety Data and Analysis Group, 2021). 2012 seems to be back on track. In 2014, there was a rise in road deaths compared to 2013. The trend is broadly flat from 2016 to 2019.

Because of the COVID pandemic, is showing some interesting deviations from the continuing pattern in 2021. In 2021 (1558), there was a 6.7% increase in fatalities compared with 2020 (1460). In 2021, road casualties increased compared to 2020, when casualty figures were low, owing partly to periods of lockdown, which reduced vehicle traffic. Because the first half of 2021 was also affected by a lockdown, the overall results for 2021 remain lower than pre-pandemic levels.

Data Set 2:

Title: Great Britain: Population, Vehicle Traffic, and Road Deaths Great Britain from 1999 to 2021

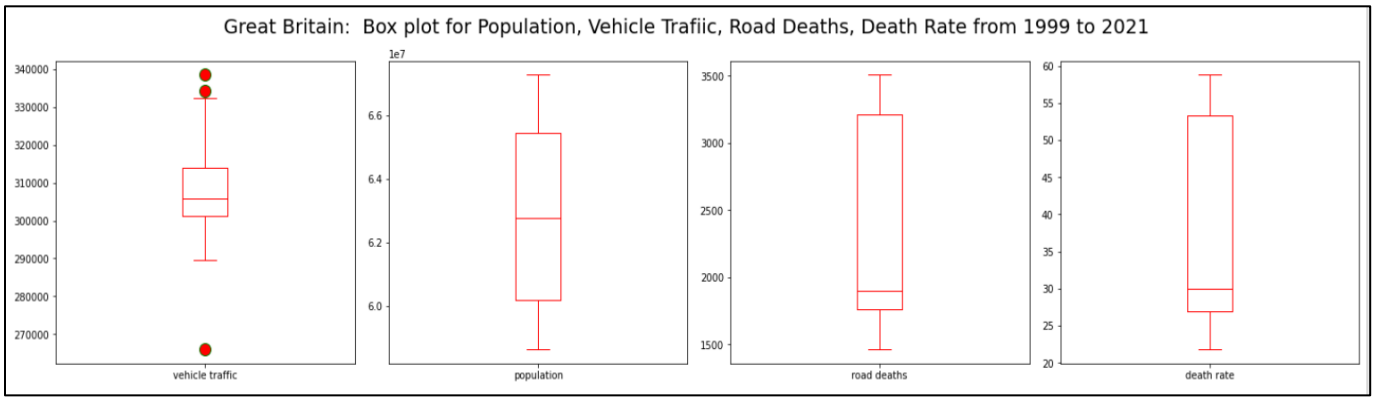
	Great Britain	1999	2000	2001	2002	2003	2004	2005	2006	2007	...	2012	2013	2014
0	vehicle traffic	2.901550e+05	2.895510e+05	2.934590e+05	3.002370e+05	3.020410e+05	3.057060e+05	3.058440e+05	3.103730e+05	3.132940e+05	...	3.045590e+05	3.057590e+05	3.144900e+05
1	population	5.863520e+07	5.885004e+07	5.909202e+07	5.935569e+07	5.964980e+07	5.999585e+07	6.038374e+07	6.080370e+07	6.126068e+07	...	6.380873e+07	6.430230e+07	6.477350e+07
2	road deaths	3.423000e+03	3.409000e+03	3.450000e+03	3.431000e+03	3.508000e+03	3.221000e+03	3.201000e+03	3.172000e+03	2.946000e+03	...	1.754000e+03	1.713000e+03	1.775000e+03
3	death rate	5.837790e+01	5.792689e+01	5.838352e+01	5.780406e+01	5.880992e+01	5.368705e+01	5.301096e+01	5.216788e+01	4.808958e+01	...	2.748840e+01	2.663979e+01	2.740318e+01

Figure 4.4 Great Britain: Population, Vehicle Trafiic, Road Deaths, and Death Rate from 1999 to 2021

See **Appendix 2** for more details.

DATA CLEANING AND PRE-PROCESSING:

Great Britain	vehicle traffic	population	road deaths	death rate
count	23.000000	2.300000e+01	23.000000	23.000000
mean	307731.869565	6.284520e+07	2407.956522	38.916778
std	16190.374956	2.933112e+06	772.454675	14.175958
min	265894.000000	5.863520e+07	1460.000000	21.771719
25%	301139.000000	6.018980e+07	1764.500000	26.997072
50%	305759.000000	6.276004e+07	1901.000000	30.038067
75%	313892.000000	6.543978e+07	3211.000000	53.349002
max	338596.000000	6.728104e+07	3508.000000	58.809922



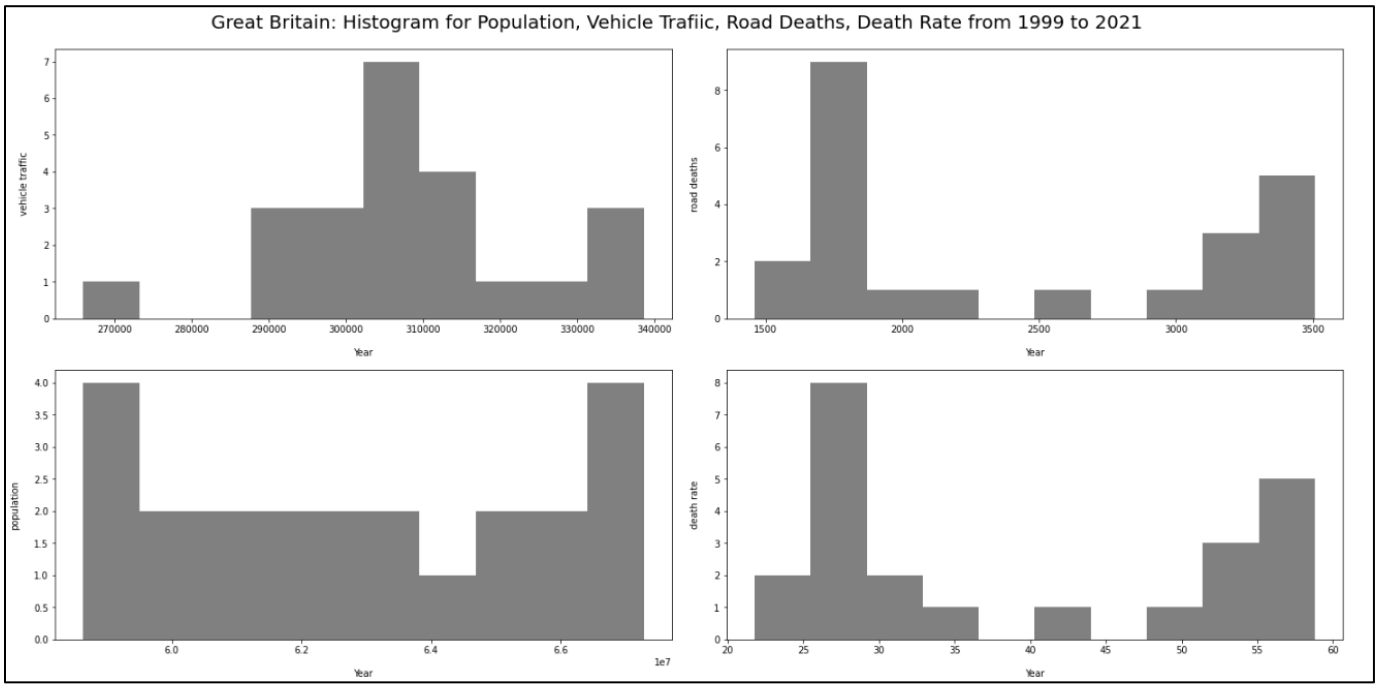


Figure 4.5: Great Britain: Box plot, summery and

histogram for Population, Vehicle Trafiic, Road Deaths, Death Rate from 1999 to 2021

In Figure 2.1, the population, vehicle traffic, road deaths, and death rate from 1999 to 2021 are described in Great Britain. Figure 4.5 depicts the data distribution, and according to that, there are three outliers shown (2018, 2019, and 2020). 2019 and 2018 are expected to be the peak years for vehicle traffic due to economic growth. 2020 will be the lowest record for vehicle traffic because of the COVID pandemic lockdown, which reduced vehicle traffic. As a result, no preprocessing is required because outliers contain critical information.

3. What are the summery highlights in Population, Vehicle Trafiic, Road Deaths, and Death Rate in the United Kingdom in from 1999 to 2020?

   1. The United Kingdom has some of the finest road safety records in Europe and the globe. It has the fifth-lowest rate of road deaths in 2021.

   2. The number of road deaths has been decreasing since 2003, despite rising population and vehicle traffic.

Data Set 3:

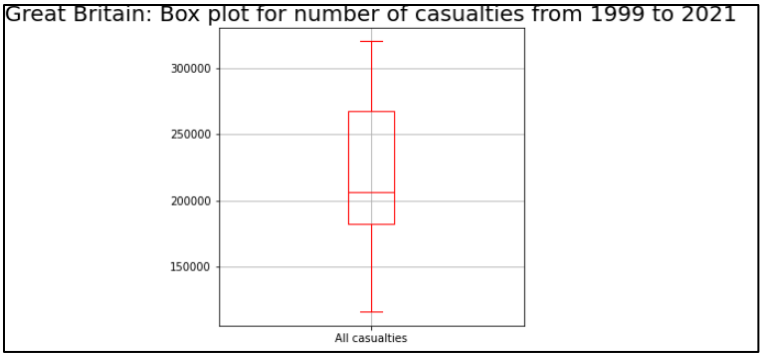
Title: Reported road casualties, Great Britain

	Accident year	Road user	Casualty sex	Urban rural	Built up roads	All casualties
0	2000	Pedestrian	Unknown	Urban	Built up road	43
1	2000	Pedestrian	Unknown	Rural	Built up road	5
2	2000	Pedestrian	Unknown	Rural	Non built up road	2
3	2000	Pedestrian	Unknown	Unallocated	Non built up road	1
4	2000	Pedestrian	Male	Urban	Motorway	15

Figure 4.6 Great Britain: Number of casualties from 1999 to 2021

See **Appendix 2** for more details.

DATA CLEANING AND PRE-PROCESSING:



count	22.000000
mean	219112.636364
std	59576.793051
min	115584.000000
25%	181955.500000
50%	206299.000000
75%	267863.750000
max	320283.000000

Figure 4.7: Great Britain: Box plot for number of casualties from 1999 to 2021

There are no considerable outliers or null values and data cleaning step not required.

RESEARCH QUESTIONS AND RESULT EVALUATION:

4. What is the yearly trend in road causalities in the United Kingdom?

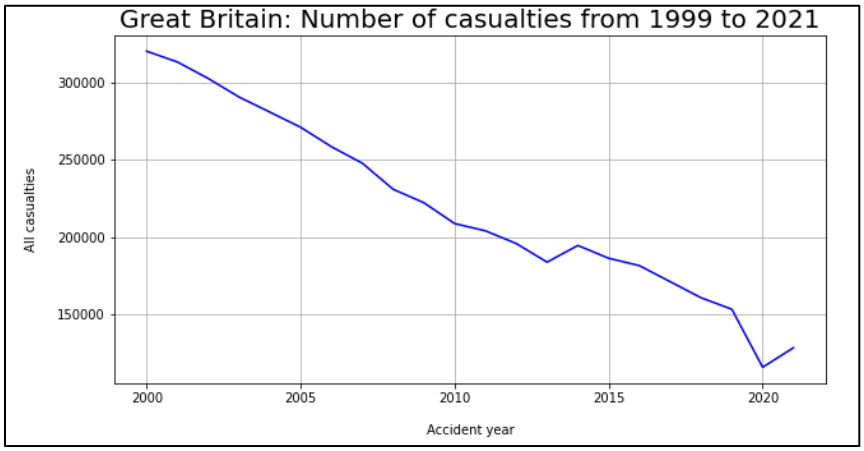


Figure 4.8 Great Britain: Number of casualties from 1999 to 2021

The number of road causalities has been declining over the last five years. In 2014, there was a rise (194477) in road deaths compared to 2013 (183670). 2014 seems to be back on track. Then the declining trend will continue through 2020. In 2021 (12809), there is a rise in road casualties compared to 2020 (115584). This trend is similar to the previous question's death rate in the United Kingdom.

5. Which area (rural or urban) has the highest rate of road causalities?

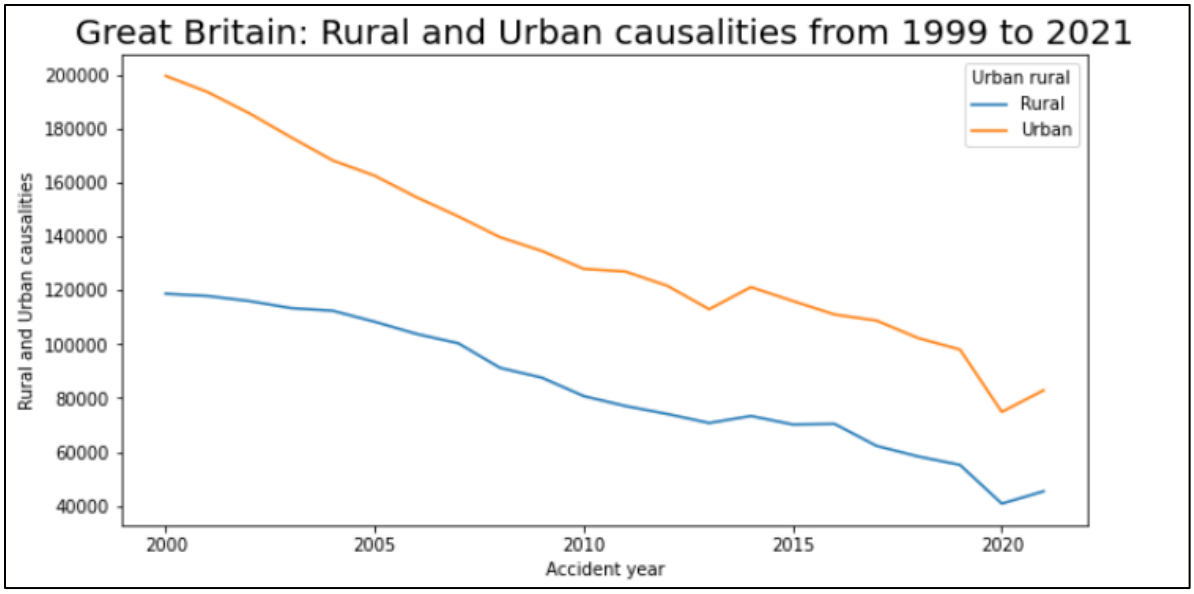
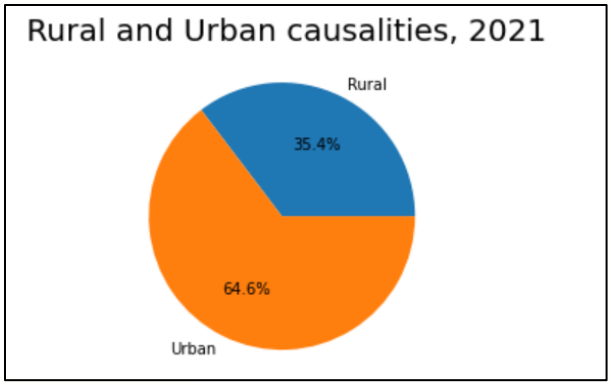


Figure 4.9 Great Britain: Rural and Urban causalities from 1999 to 2021



Urban rural	Rural	Urban
count	22.000000	22.000000
mean	83976.272727	134887.181818
std	24420.172417	35018.771933
min	40745.000000	74822.000000
25%	70211.500000	111487.000000
50%	78847.500000	127451.500000
75%	107181.500000	160576.750000
max	118725.000000	199659.000000

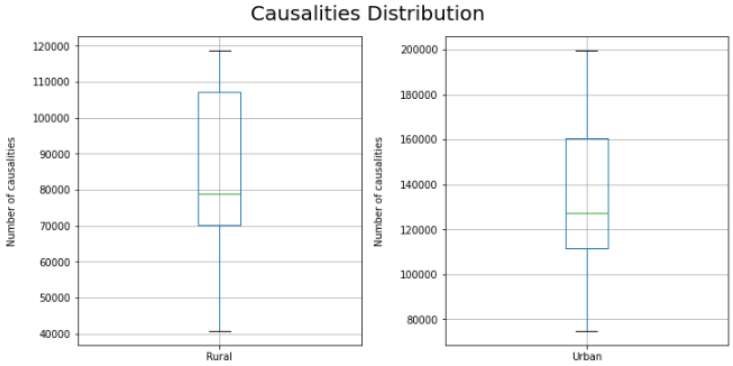


Figure 4.10 Great Britain: Pie chart, summaries and boxplot

Rural and Urban causalities from 1999 to 2021

Urban roads are those within an area with a population of 10,000 or more. Roads outside these areas will be classified as "rural." The majority of casualties occurred on urban roads, and it's double when compared to rural areas. When comparing the spread of the causalities for rural and urban areas, the

standard deviation of the urban causality distribution (35018) is higher than the rural distribution (24420). Therefore, the urban causalities are more spread out. By considering inter-quartile range or range, the same information can be extracted.

6. Who is the most vulnerable road user group in UK in terms of road causalities?

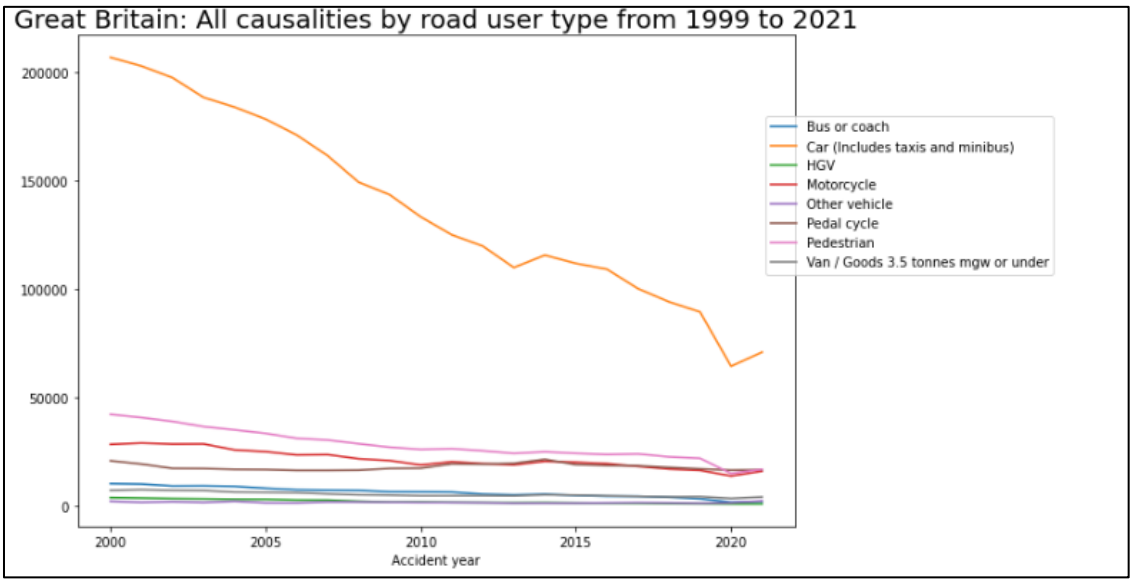


Figure 4.11 Great Britain: All causalities by road user type from 1999 to 2021

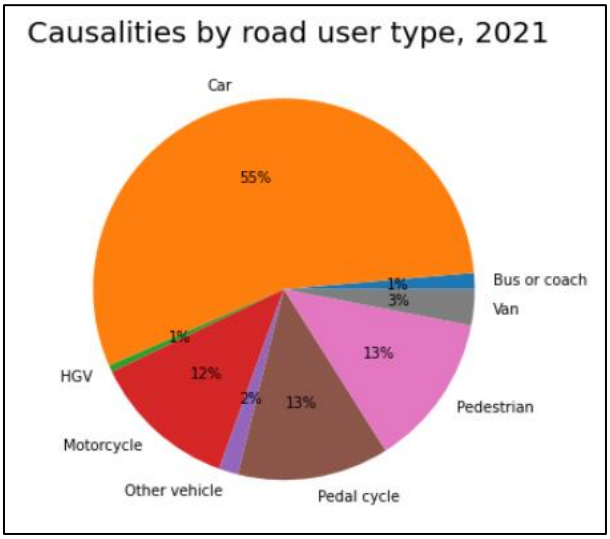


Figure 4.12 Great Britain: All causalities by road user type from 1999 to 2021

Car occupants (including car drivers and car passengers) were the road user group with the greatest number of fatalities (55% of total fatalities in 2021). However, this is unsurprising as cars account for the majority of the traffic on Britain's roads. According to the research, although car occupants account for

the largest number of fatalities, motorcycle riders have the highest fatality rate. The fatality rate is calculated per billion kilometers driven. Motorcyclists do not drive as many kilometers compared to cars, but they are still facing fatalities very often (International Traffic Safety Data and Analysis Group, 2021). It's identified as a good area for future analysis.

7. What is the most vulnerable gender group in the UK in terms of road causalities?

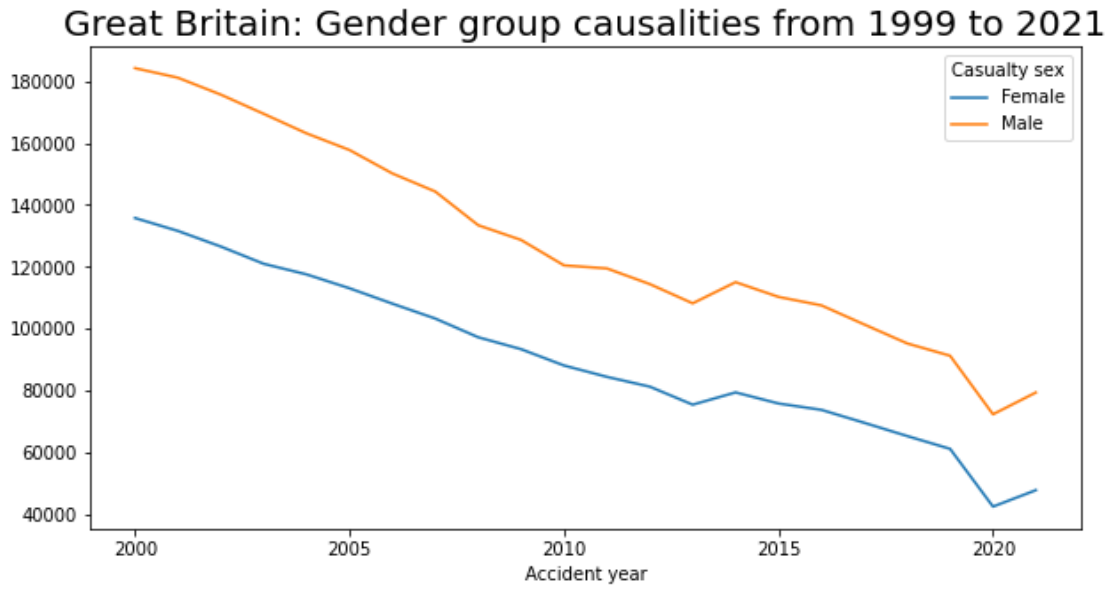


Figure 4.13 Great Britain: Gender group causalities from 1999 to 2021

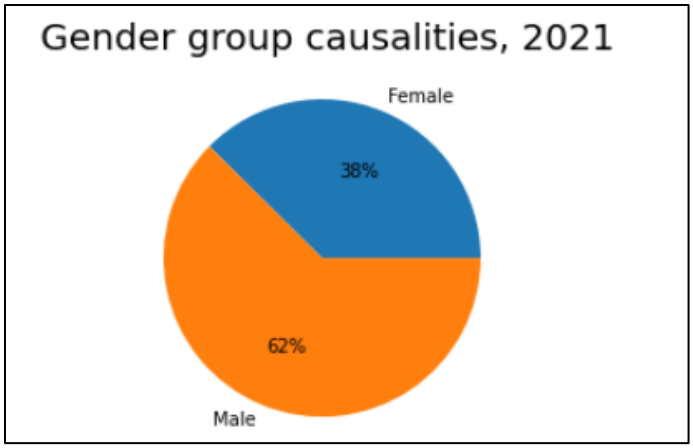


Figure 4.14 Great Britain: Gender group causalities from 1999 to 2021

Males are the most vulnerable gender group in the UK in terms of road fatalities compared to females. Males may be more vulnerable to accidents than females due to their risk-taking attitudes (Wang et al., 2013).

8. What is the most vulnerable road user age group in the UK in terms of road causalities?

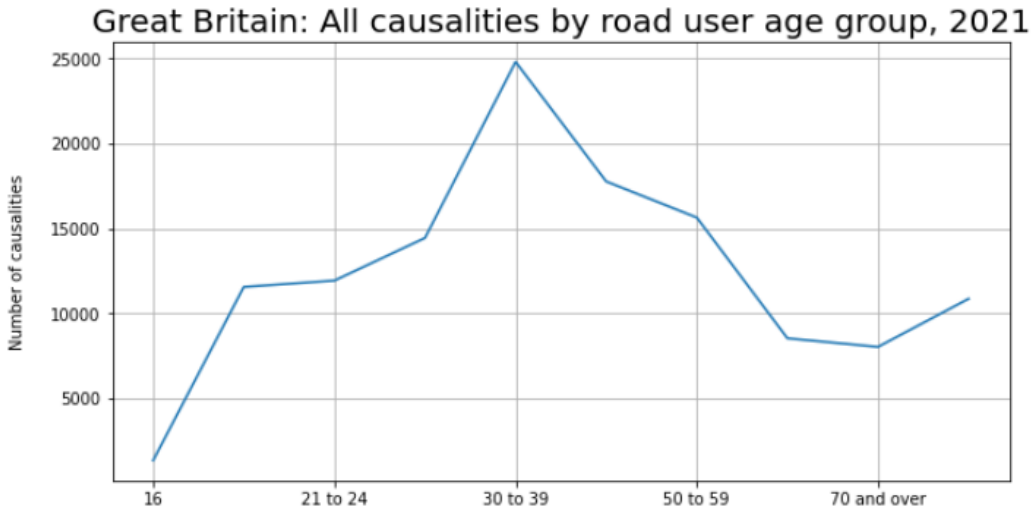


Figure 4.15 Great Britain: All causalities by road user age group, 2021

The reduction in fatalities from 1999 to 2021 has benefited all age groups. The 30-39 age groups are the most vulnerable age group in the UK in terms of road causalities considering the latest records in 2021. In bellow Data Set 4 analysis it shows that the same age groups are the car users.

9. What is the most dangerous type of road in the United Kingdom?

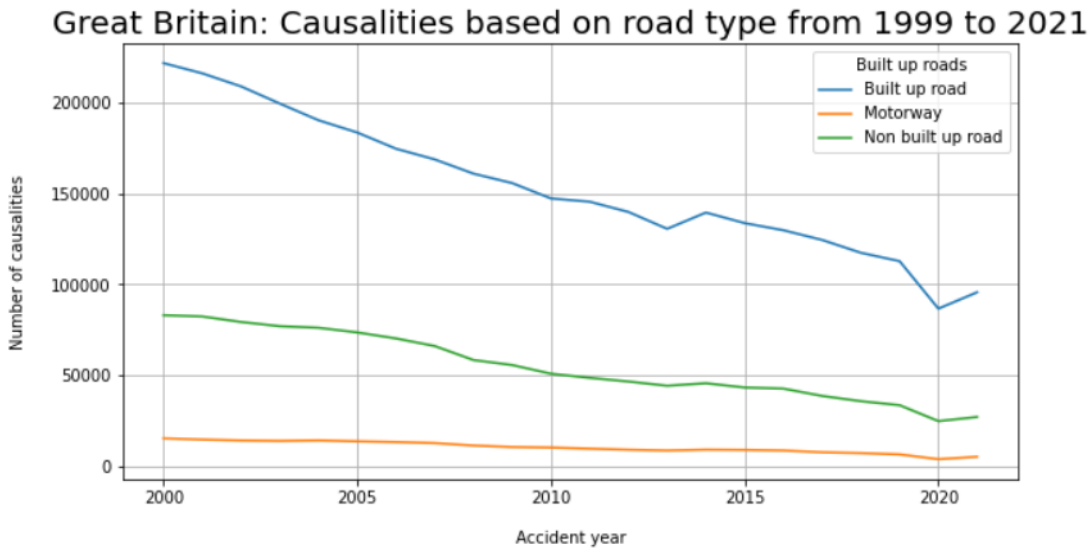


Figure 4.16 Great Britain: Causalities based on road type from 1999 to 2021

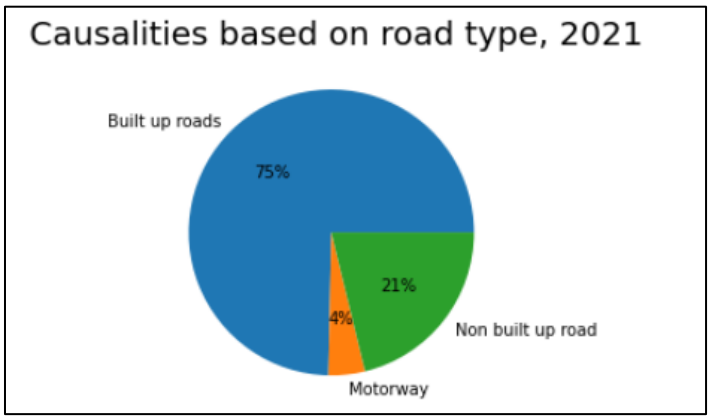


Figure 4.17 Great Britain: Causalities based on road type from 1999 to 2021

Built-up roads are roads with speed limits of 40mph or less (ignoring temporary limits), and non-built-up roads are roads with speed limit of over 40mph or more, excluding motorways. The majority of causalities occur in built up areas.

Further classification is done by road user type and severity data in the Inferential Statistics Analysis section.

10. What are the summery highlights in road casualities in terms of road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, highway) in the United Kingdom from 1999 to 2021?

```
{'High Risk': 0, 'Low Risk': 1, 'Medium Risk': 2}
{'Built up road': 0, 'Motorway': 1, 'Non built up road': 2}
{'Rural': 0, 'Unallocated': 1, 'Urban': 2}
{'Female': 0, 'Male': 1, 'Unknown': 2}
{'Bus or coach': 0, 'Car (Includes taxis and minibus)': 1, 'HGV': 2,
'Motorcycle': 3, 'Other vehicle': 4, 'Pedal cycle': 5, 'Pedestrian': 6, 'Van / Goods 3.5 tonnes mgw or under': 7}
```

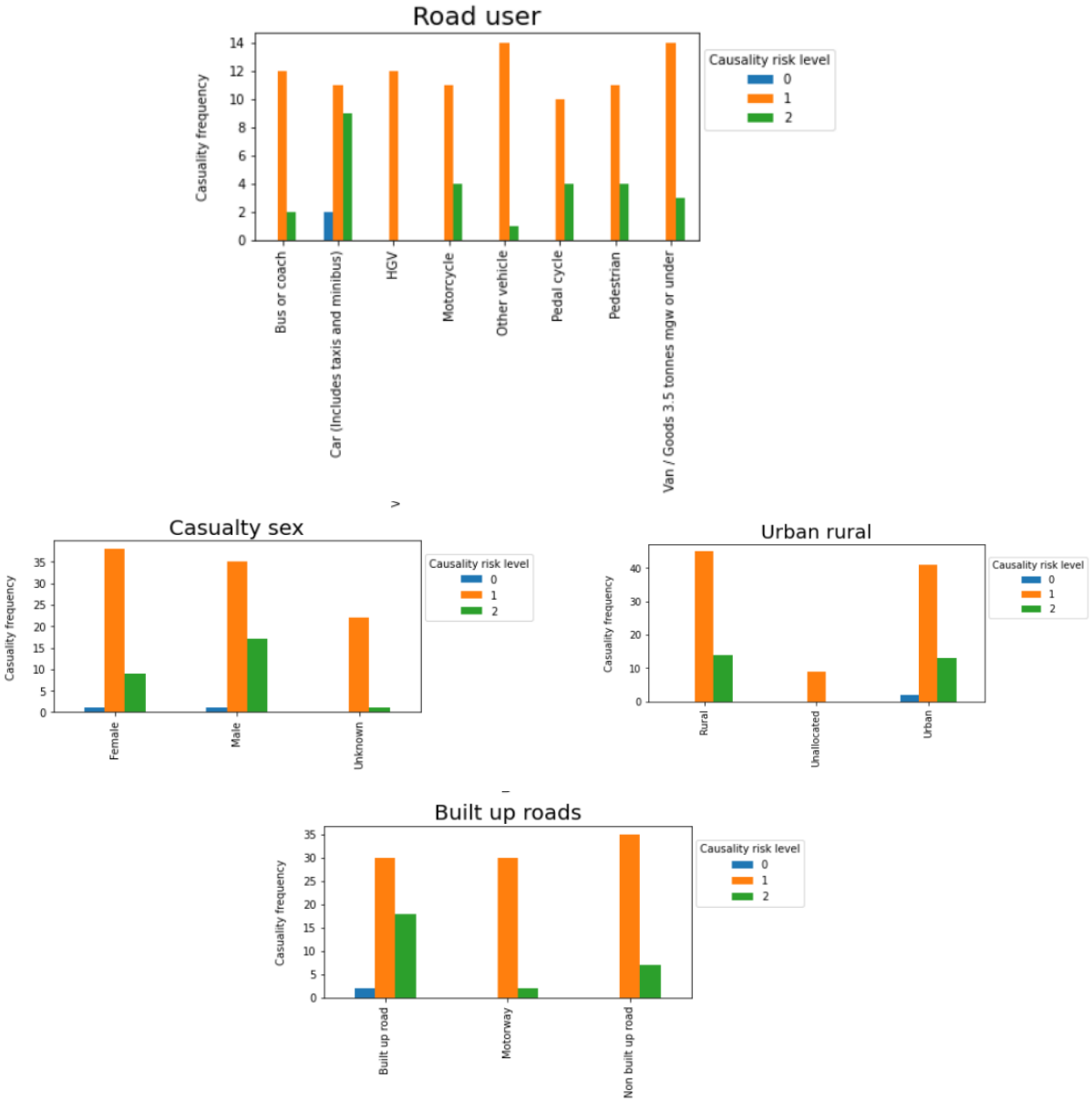


Figure 4.18 Great Britain: Causalities based on road user type, gender types, area and road type from 1999 to 2021

Above Figure 4.18 summarize the highlights.

1. When it comes to road user types (pedestrian, bicycle, motorcycle, car, bus, coach, van, HGV, and other vehicle), the car is the most vulnerable.
2. Among the different gender types (female and male), both are at high risk, but males are the most vulnerable.
3. Considering area (urban or rural), urban areas are the most vulnerable and high-risk category.

4. Considering road type (built-up road, non-built-up road, highway) build up roads are the most vulnerable high-risk category.

Data Set 4:

Title: RAS0202: Gender and age group

	Road user type	Sex	Age group	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	Pedestrians	Male	Under 16	4162	3861	3888	3688	3645	3486	3207	3105	2038	2507
1	Pedestrians	Male	16	289	280	256	232	237	269	220	189	146	159
2	Pedestrians	Male	17 to 20	1174	1072	982	957	861	945	798	802	449	569
3	Pedestrians	Male	21 to 24	1082	928	1002	874	875	843	736	727	443	494
4	Pedestrians	Male	25 to 29	1113	1040	1073	1052	970	1000	888	865	575	608

Figure 4.19 Great Britain: Number of causalities with age groups

See **Appendix 2** for more details.

Data Set 5:

Title: RAS0501: Drivers involved by gender, age and road user type

	Driver type	Sex	Age group	Severity	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	Pedal cyclists	Female	Under 16	All collisions	380	306	289	265	262	274	221	229	222	244
1	Pedal cyclists	Female	16	All collisions	37	33	48	49	29	37	24	37	45	28
2	Pedal cyclists	Female	17-20	All collisions	222	208	220	214	202	182	154	154	149	126
3	Pedal cyclists	Female	21-24	All collisions	343	367	415	363	339	302	297	252	264	250
4	Pedal cyclists	Female	25-29	All collisions	584	582	659	586	563	548	575	545	466	466

Figure 4.20 Great Britain: All collisions by age group, 2021

See **Appendix 2** for more details.

RESEARCH QUESTIONS AND RESULT EVALUATION:

11. What is the most vulnerable driver age group in the UK for road collisions?

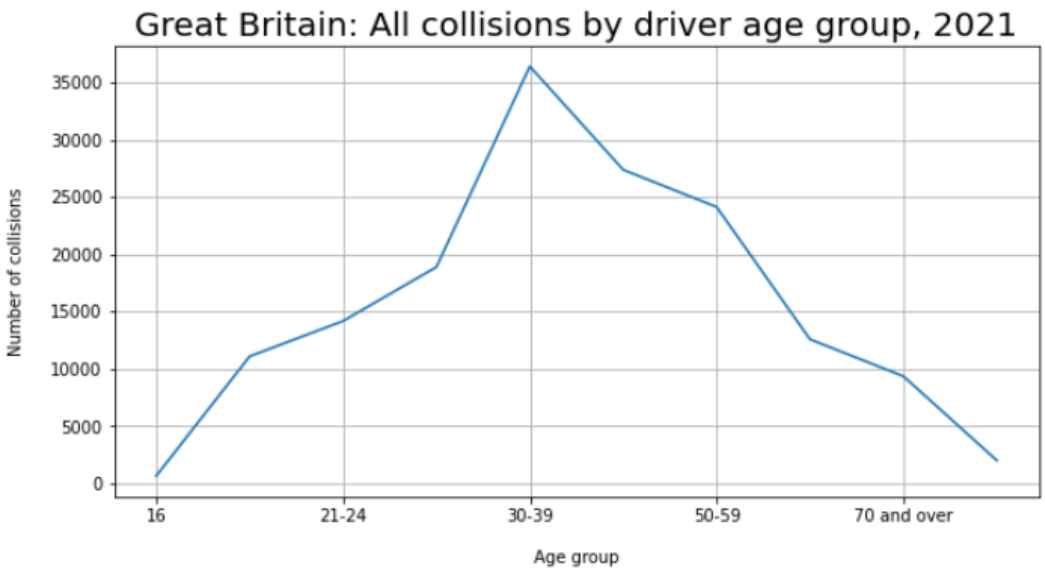


Figure 4.21 Great Britain: All collisions by driver age group, 2021

The 30-39 age groups are the most vulnerable age group in the UK in terms of road collisions considering the latest records in 2021. Data Set 3 analysis shows that the same age group are the mostvulnerable for road causalities.

12. Who are the most susceptible road drivers in the United Kingdom in terms of road causalities /collisions?

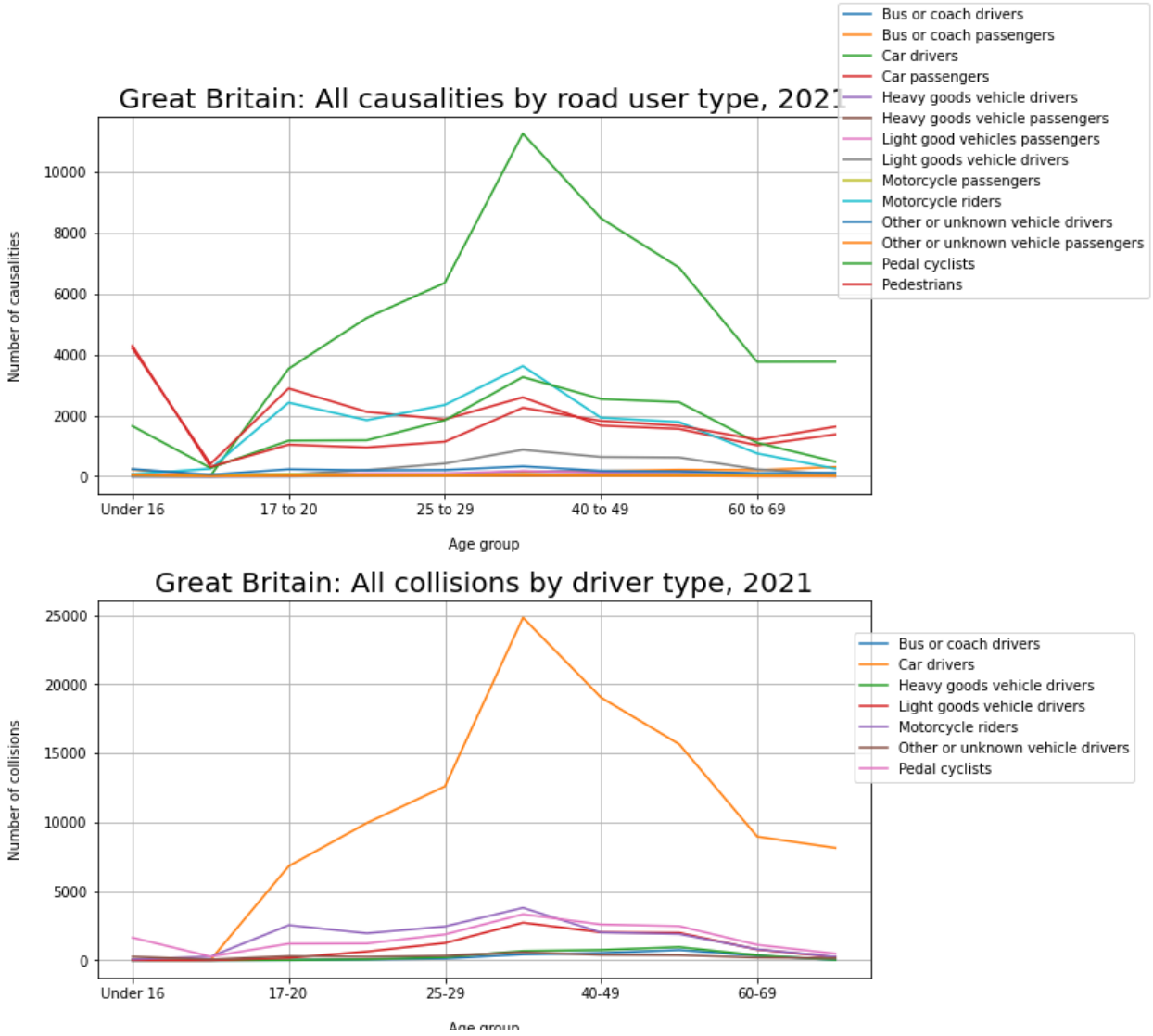


Figure 4.22 Great Britain: All causalities by road user type/ all collisions by driver type, 2021

Car drivers are the most vulnerable driver group in the UK in terms of road causalities and collisions considering all ages. But there are clear deviations when considering each age group. The Under 16 age group, most vulnerable driver type is Pedal cyclists because pedal cycles are most popular among the group.

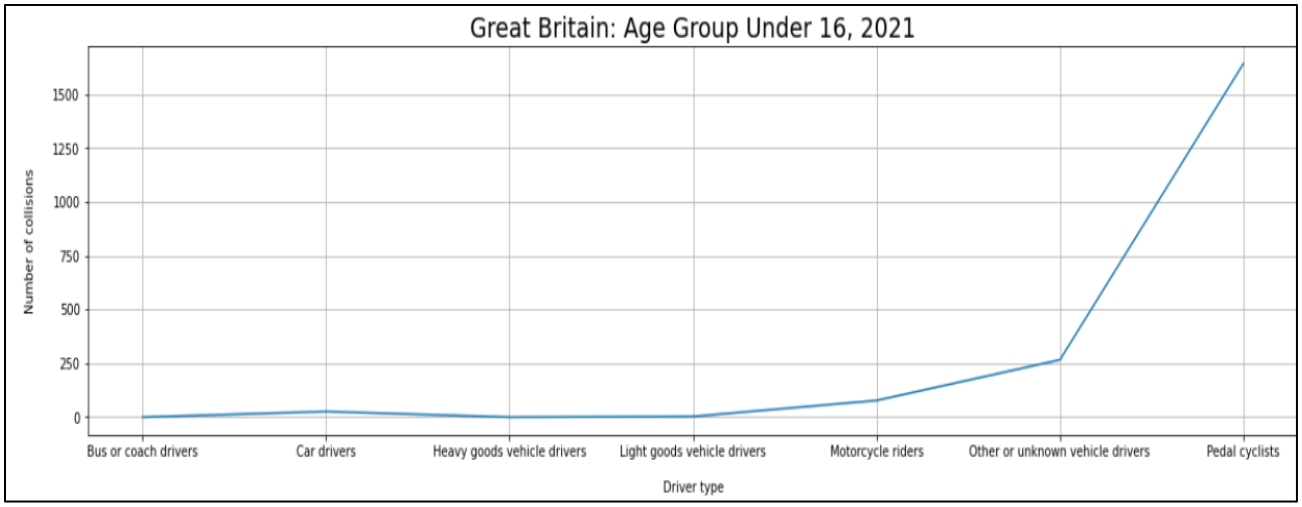


Figure 4.23 Great Britain: Collisions by driver type, age group Under 16, 2021

The 16 age group, most vulnerable driver group is motorcycle riders because motorcycles are most popular among the group.

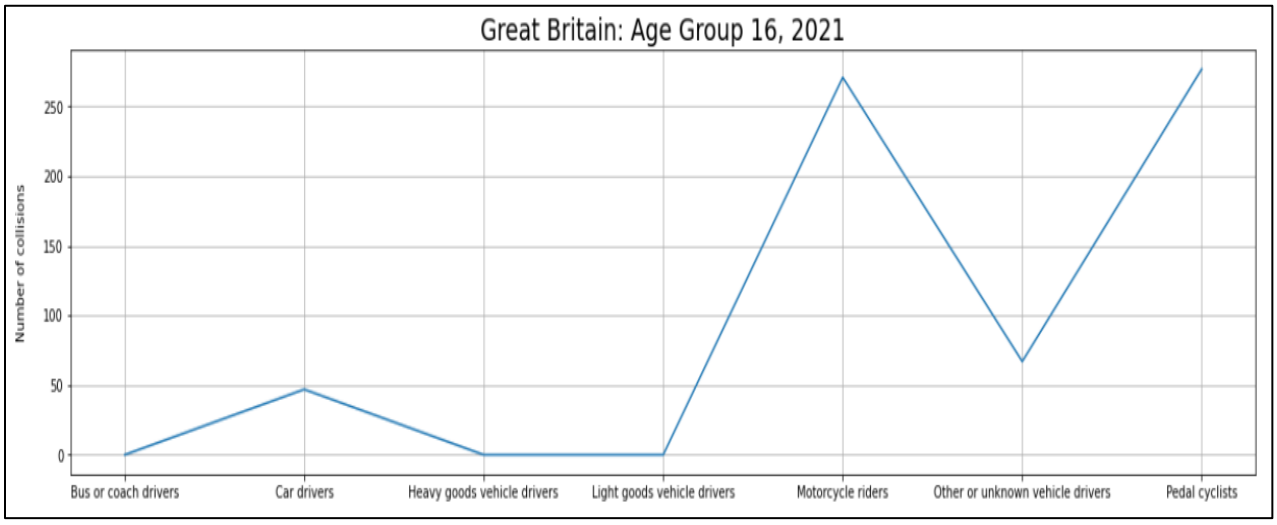


Figure 4.24 Great Britain: Collisions by driver type, age group 16, 2021

For other age groups, most vulnerable driver group is car driver's because cars are most popular among the group.

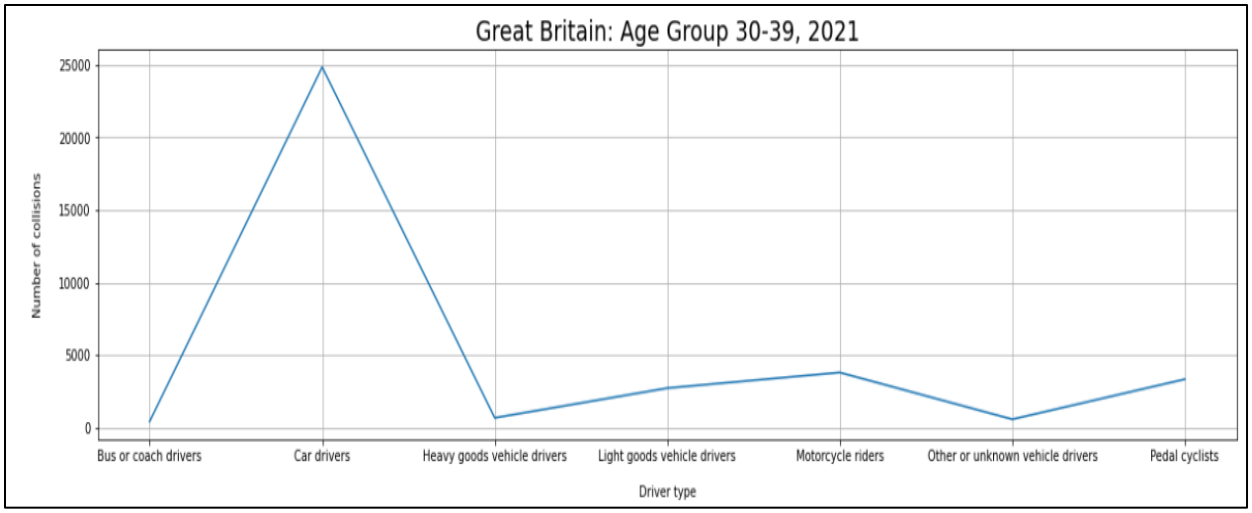


Figure 4.25 Great Britain: Collisions by driver type, age group 30-39, 2021

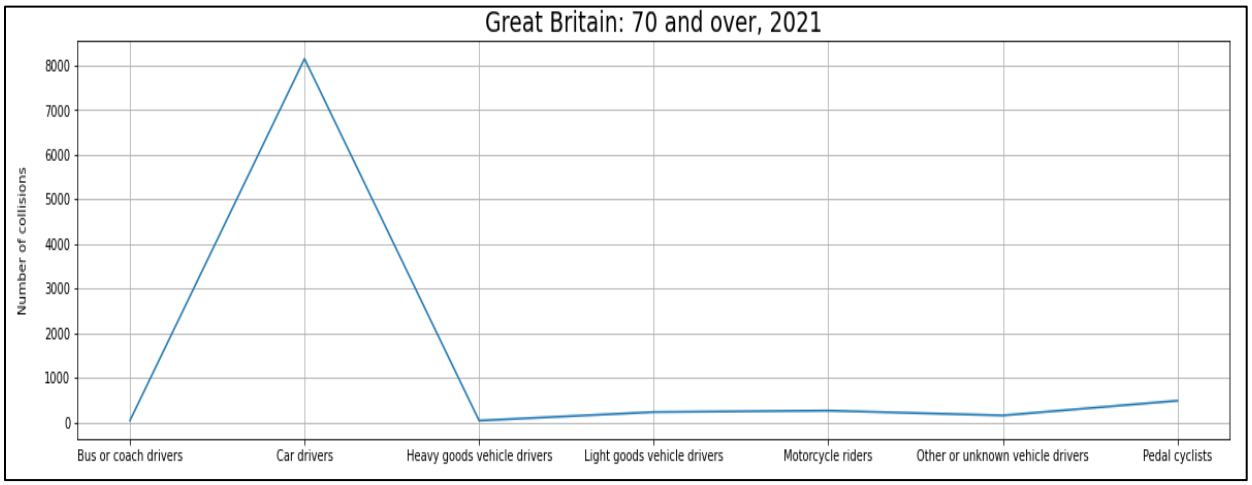


Figure 4.26 Great Britain: Collisions by driver type, age group 70 and over, 2021

13. What are the summery highlights in road collisions in terms of driver type (Bus or coach drivers, Car drivers, Heavy goods vehicle drivers, Light goods vehicle drivers, Motorcycle riders, Other or unknown vehicle drivers, Pedal cyclists) in the United Kingdom in 2021?
    1. The number of road causalities has been declining over the last five years.
    2. The 30-39 year old age groups are the most vulnerable age group in the UK in terms of road collisions and fatalities, considering the latest records in 2021.
    3. Car drivers are the most vulnerable group of drivers in the UK in terms of road fatalities and collisions, regardless of age.
    4. Pedal cyclists are the most vulnerable driver type among the under-16 age group because they are the most popular among the group.

5. Motorcycle riders, the most vulnerable driver group, are 16-year-olds because motorcycles are the most popular among the group.

2.5.2 Level 02: Inferential Statistics Analysis

T-Test and Anova

14. Is the average number of causalities in 2021 greater than zero?

Accident year	All casualties
2000	320283
2001	313309
2002	302605
2003	290607
2004	280840
2005	271017
2006	258404
2007	247780
2008	230905
2009	222146
2010	208648
2011	203950
2012	195723

```
data_AllCausalities_AccidentYear['All casualties'].describe()

count        22.000000
mean     219112.636364
std       59576.793051
min      115584.000000
25%      181955.500000
50%      206299.000000
75%      267863.750000
max      320283.000000
Name: All casualties, dtype: float64
```

```
from scipy import stats
stats.ttest_1samp(data_AllCausalities_AccidentYear['All casualties'], 0)

Ttest_1sampResult(statistic=17.250498224981282, pvalue=7.057630401562082e-14)
```

Figure 4.27 Great Britain: Causality statistics

Use hypothesis testing,

α= 0.05

H0: μ <= 0

H1: μ > 0, trying to prove

Right tail T test

P value < 1-α

$7.05 \times e^{-14} < 0.95$

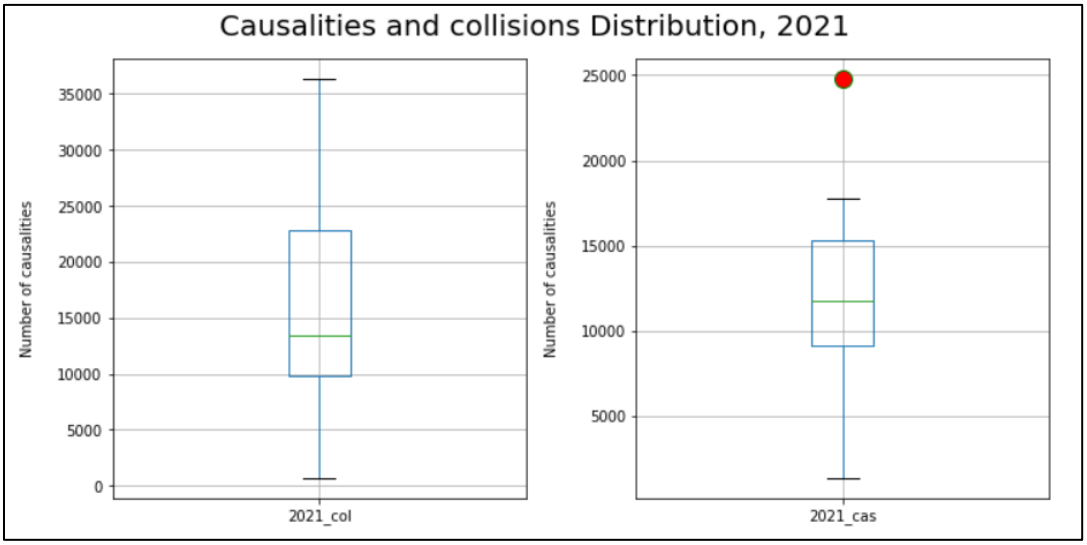
P is lower than the alpha value so H0 is rejected and conclude that mean number of causalities in 2021 is more than zero.

15. Does this suggest that the number of causalities in different age groups will be equal to zero in 2021?

Age group	2021_col	2021_cas
16	2680	1369
17 to 20	44776	11575
21 to 24	57144	11946
25 to 29	76190	14458
30 to 39	149098	24816
40 to 49	110668	17779
50 to 59	97190	15655
60 to 69	50588	8552
70 and over	37634	8048
Under 16	8074	10880

```
df_collisions_casualties_All_Ages_merge. describe()
```

	2021_col	2021_cas
count	10.000000	10.000000
mean	63404.200000	12507.800000
std	45759.842584	6293.135481
min	2680.000000	1369.000000
25%	39419.500000	9134.000000
50%	53866.000000	11760.500000
75%	91940.000000	15355.750000
max	149098.000000	24816.000000



```
from scipy import stats
stats.ttest_1samp(df_collisions_casualties_All_Ages_merge['2021_cas'], 0)

Ttest_1sampResult(statistic=6.2851239480685335, pvalue=0.00014349934404667546)
```

Figure 4.28 Great Britain: Causality/collisions statistics

Use hypothesis testing,

α= 0.05

H0: μ = 0

H1: μ != 0

Two tail T test

P value < α

0.00014 < 0.025

P is lower than the alpha value so H0 is rejected and conclude that mean number of causalities in in different age groups in 2021 is not equals to zero.

16. Is it true that the number of causalities in men is bigger than the number of causalities in women?

Casualty sex	Female	Male
Accident year		
2000	135803	184259
2001	131631	181167
2002	126583	175706
2003	121001	169492
2004	117573	163173
2005	113087	157797
2006	108111	150212
2007	103292	144363
2008	97250	133478
2009	93390	128711
2010	88117	120490
2011	84445	119498
2012	81277	114439
2013	75446	108213

```
data_AllCausalities_Gender_Group_pivot.describe()
```

Casualty sex	Female	Male
count	22.000000	22.000000
mean	90562.318182	128342.636364
std	26529.272634	33164.753974
min	42488.000000	72335.000000
25%	74193.500000	107720.250000
50%	86281.000000	119994.000000
75%	111843.000000	155900.750000
max	135803.000000	184259.000000

```
stats.ttest_ind(data_AllCausalities_Gender_Group_pivot['Male'], data_AllCausalities_Gender_Group_pivot['Female'])

Ttest_indResult(statistic=4.172481427056453, pvalue=0.00014798491110957495)
```

Figure 4.29 Great Britain: Causality gender statistics

Use hypothesis testing, Two sample T test

$\alpha = 0.05$

H0: $0 => \mu_{male} - \mu_{female}$

H1: $0 < \mu_{male} - \mu_{female}$

Right tail T test

P value < 1-α

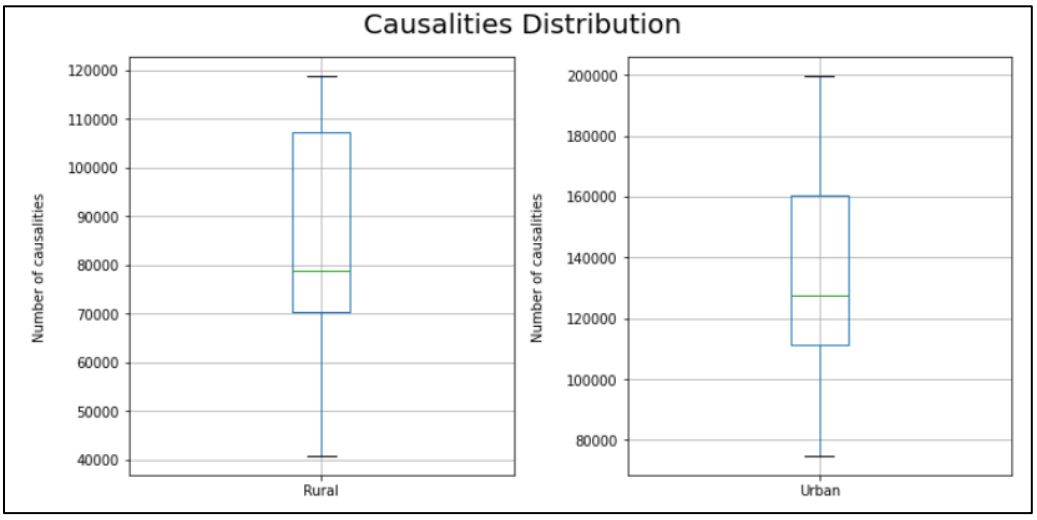
0.00014 < 0.95

P is lower than the alpha value so H0 is rejected and conclude that number of causalities in males is greater than the number of causalities in females.

17. Is it true that the number of causalities in urban regions exceeds the number of causalities in rural areas?

Urban rural	Accident year	Rural	Urban
0	2000	118725.0	199659.0
1	2001	117875.0	193677.0
2	2002	116007.0	185779.0
3	2003	113377.0	176835.0
4	2004	112385.0	168216.0
5	2005	108309.0	162578.0
6	2006	103799.0	154573.0
7	2007	100320.0	147444.0
8	2008	91176.0	139716.0
9	2009	87533.0	134613.0
10	2010	80692.0	127956.0
11	2011	77003.0	126947.0
12	2012	74043.0	121680.0
13	2013	70707.0	112963.0

Urban rural	Rural	Urban
count	22.000000	22.000000
mean	83976.272727	134887.181818
std	24420.172417	35018.771933
min	40745.000000	74822.000000
25%	70211.500000	111487.000000
50%	78847.500000	127451.500000
75%	107181.500000	160576.750000
max	118725.000000	199659.000000




```
t2, p = stats.ttest_ind(data_AllCausalities_Urban_Rural_T["Urban"], data_AllCausalities_Urban_Rural_T["Rural"])
print("p value = {:g}".format(p))
print("t value = {:g}". format(t2))
```

```
p value = 1.52426e-06
t value = 5.59332
```

Figure 4.30 Great Britain: Causality area type statistics

Use hypothesis testing, Two sample t test

α= 0.05

H0: $0 >= \mu_{urban} - \mu_{rural}$

H1: $0 < \mu_{urban} - \mu_{rural}$

Right tail T test

P value < 1- α

$1.5 \times e^{-06} < 0.95$

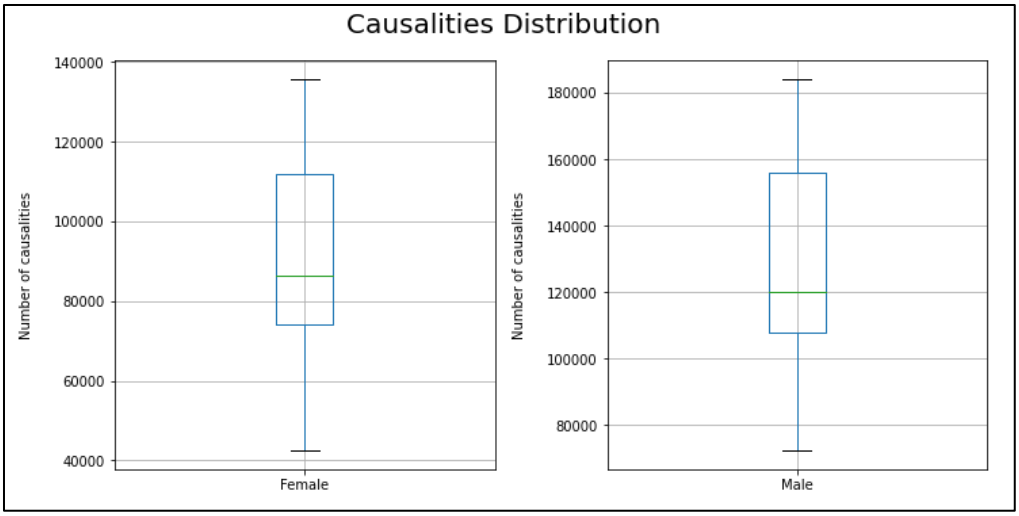
P is lower than the alpha value so H0 is rejected number of causalities in urban areas are greater than the number of causalities in rural areas.

18. Is there a difference in the average number of causalities and collisions between age groups?

Casualty sex	Female	Male
Accident year		
2000	135803	184259
2001	131631	181167
2002	126583	175706
2003	121001	169492
2004	117573	163173
2005	113087	157797
2006	108111	150212
2007	103292	144363
2008	97250	133478
2009	93390	128711
2010	88117	120490
2011	84445	119498
2012	81277	114439
2013	75446	108213

```
data_AllCausalities_Gender_Group_pivot.describe()
```

Casualty sex	Female	Male
count	22.000000	22.000000
mean	90562.318182	128342.636364
std	26529.272634	33164.753974
min	42488.000000	72335.000000
25%	74193.500000	107720.250000
50%	86281.000000	119994.000000
75%	111843.000000	155900.750000
max	135803.000000	184259.000000




```
stats.ttest_ind(data_AllCausalities_Gender_Group_pivot['Female'], data_AllCausalities_Gender_Group_pivot['Male'])

Ttest_indResult(statistic=-4.172481427056453, pvalue=0.00014798491110957495)
```

Figure 4.31 Great Britain: Causality gender type statistics

Use hypothesis testing, Two sample t test

α= 0.05

H0: $\mu_{female} = \mu_{male}$

H1: $\mu_{female} != \mu_{male}$

Two tail T test

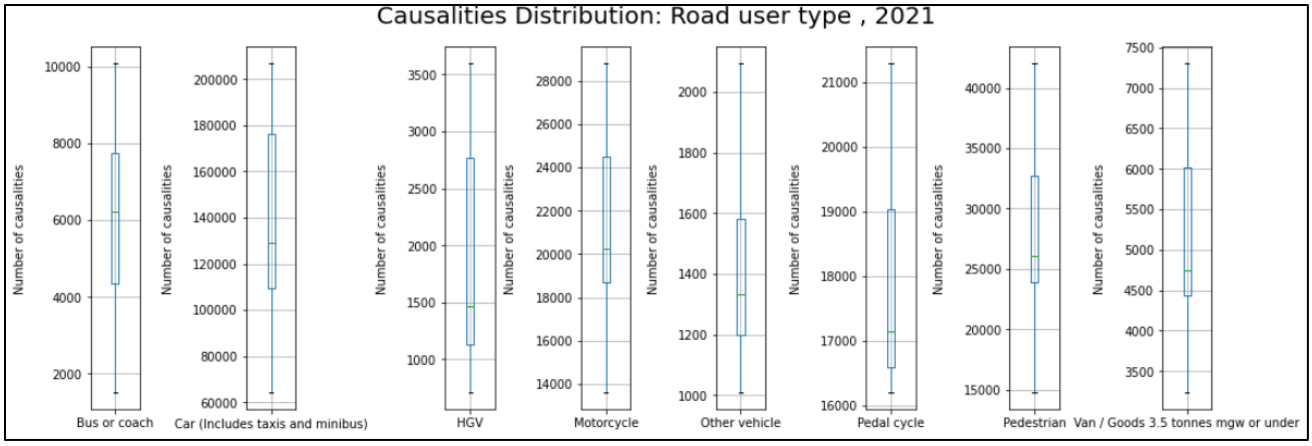
P value < α

0.00014 < 0.025

P is lower than the alpha value so H0 is rejected and conclude that mean number of causalities in in different gender groups in 2021 is not equal.

19. Is it true that all road users have the same average number of fatalities?

Road user	Accident year	Bus or coach	Car (Includes taxis and minibus)	HGV	Motorcycle	Other vehicle	Pedal cycle	Pedestrian	Van / Goods 3.5 tonnes mgw or under
0	2000	10088	206799	3597	28212	1935	20612	42033	7007
1	2001	9884	202802	3388	28810	1430	19114	40577	7304
2	2002	9005	197425	3178	28353	1746	17107	38784	7007
3	2003	9068	188342	3061	28411	1390	17033	36405	6897
4	2004	8820	183858	2883	25641	1943	16648	34881	6166
5	2005	7920	178302	2843	24824	1238	16561	33281	6048
6	2006	7253	171000	2530	23326	1203	16196	30982	5914
7	2007	7079	161433	2476	23459	1607	16195	30191	5340
8	2008	6929	149188	1930	21550	1616	16297	28482	4913
9	2009	6317	143412	1519	20703	1501	17064	26887	4743
10	2010	6268	133205	1578	18686	1387	17185	25845	4494
11	2011	6177	124924	1415	20150	1372	19215	26198	4499
12	2012	5234	119708	1339	19310	1290	19091	25218	4533
13	2013	4873	109787	1296	18752	1065	19438	24033	4426
14	2014	5198	115530	1353	20366	1080	21287	24748	4915
15	2015	4626	111707	1203	19918	1080	18844	24061	4750
16	2016	4246	109046	1105	19297	1199	18477	23550	4464
17	2017	4236	100082	1038	18042	1295	18321	23805	4174
18	2018	3801	93979	880	16818	1192	17550	22432	3945
19	2019	3085	89331	786	16224	1009	16884	21770	4069
20	2020	1506	64255	710	13604	1230	16294	14750	3235
21	2021	1762	70755	735	15838	2094	16458	16654	3913




```
data_AllCausalities_Road_User_T.set_index('Accident year').describe()
```

Road user	Bus or coach	Car (Includes taxis and minibus)	HGV	Motorcycle	Other vehicle	Pedal cycle	Pedestrian	Van / Goods 3.5 tonnes mgw or under
count	22.000000	22.000000	22.000000	22.00000	22.000000	22.000000	22.000000	22.000000
mean	6062.500000	137494.090909	1856.500000	21377.00000	1404.636364	17812.318182	27980.318182	5125.272727
std	2465.565979	43411.652729	951.116074	4449.38183	305.057457	1499.055524	7242.094306	1164.368301
min	1506.000000	64255.000000	710.000000	13604.00000	1009.000000	16195.000000	14750.000000	3235.000000
25%	4341.000000	109231.250000	1129.500000	18702.50000	1200.000000	16582.750000	23862.000000	4435.500000
50%	6222.500000	129064.500000	1467.000000	20258.00000	1333.500000	17146.000000	26021.500000	4746.500000
75%	7753.250000	176476.500000	2764.750000	24482.75000	1580.500000	19029.250000	32706.250000	6014.500000
max	10088.000000	206799.000000	3597.000000	28810.00000	2094.000000	21287.000000	42033.000000	7304.000000

```
import scipy.stats as stats
fvalue, pvalue  = stats.f_oneway(data_AllCausalities_Road_User_T["Bus or coach"],data_AllCausalities_Road_User_T["Car (Includes taxis and minibus)"],data_AllCausalities_Road_User_T
print("F Value  = {:g} ".format(fvalue))
print("P Value  = {:g} ".format(pvalue))

alpha = 0.05

if pvalue < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected")
else:
    print("The null hypothesis is accepted")
```

```
F Value  = 185.667
P Value  = 1.28941e-75
The null hypothesis can be rejected
```

Figure 4.32 Great Britain: Causality road user type type statistics

Use hypothesis testing, Anova

α= 0.05

H0: μ_{bus} or coach = μ_{car}= μ_{HGV}= $\mu_{motorcycle}$= $\mu_{pedalcycle}$ = $\mu_{pedestrain}$= μ_{van}= μ_{other}

H1: μ_{bus} or coach != μ_{car}= μ_{HGV} !=$\mu_{motorcycle}$!=$\mu_{pedalcycle}$!=$\mu_{pedestrain}$= μ_{van}!=μ_{other}

One way Anova

P value < α

$1.28 \times e^{-75} < 0.05$

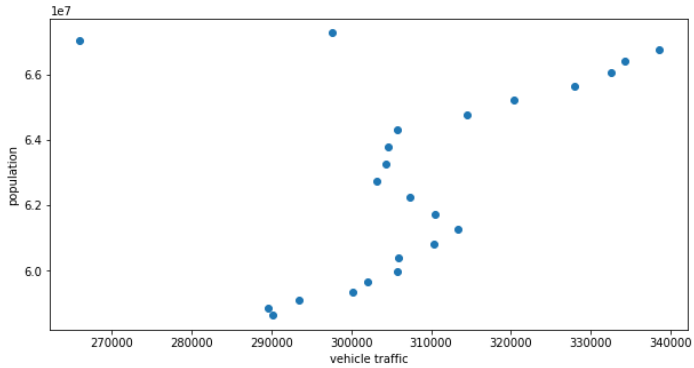
P is lower than the alpha value so H0 is rejected and conclude that no road user types have similar mean numbers of causalities

Correlation and Chi Square

20. Find the correlation between population and vehicle traffic over the years in Great Britain Use Pearson correlation

Great Britain	vehicle traffic	population	road deaths	death rate
1999	290155.0	58635202.0	3423.0	58.377901
2000	289551.0	58850043.0	3409.0	57.926891
2001	293459.0	59092016.0	3450.0	58.383522
2002	300237.0	59355690.0	3431.0	57.804062
2003	302041.0	59649799.0	3508.0	58.809922
2004	305706.0	59995851.0	3221.0	53.687046
2005	305844.0	60383741.0	3201.0	53.010959
2006	310373.0	60803700.0	3172.0	52.167878
2007	313294.0	61260676.0	2946.0	48.089577
2008	310530.0	61742151.0	2538.0	41.106440
2009	307270.0	62243378.0	2222.0	35.698577
2010	303198.0	62760039.0	1850.0	29.477356
2011	304287.0	63286362.0	1901.0	30.038067
2012	304559.0	63808727.0	1754.0	27.488403
2013	305759.0	64302297.0	1713.0	26.639795
2014	314490.0	64773504.0	1775.0	27.403180

Correlation between vehicle traffic and population, Great Britain



```
correlation = dftrain_stats_T.corr(method='pearson')
correlation
```

Great Britain	vehicle traffic	population	road deaths	death rate
vehicle traffic	1.000000	0.388131	-0.304557	-0.325512
population	0.388131	1.000000	-0.941479	-0.949827
road deaths	-0.304557	-0.941479	1.000000	0.999398
death rate	-0.325512	-0.949827	0.999398	1.000000

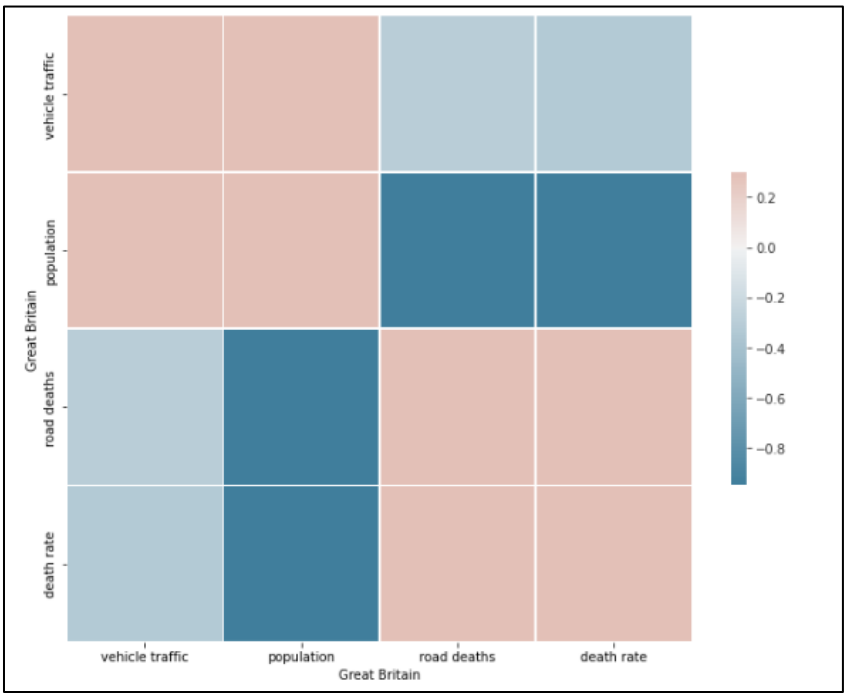


Figure 4.33 Great Britain: Death rate, road deaths, population, vehicle traffic statistics

0.5> ($r_{\text{population - vehicle traffic}}$ = 0.388) > 0.1 hence having week positive relationship between vehicle traffic and the population

21. Find the correlation between population and the road deaths over the years in Great Britain. Use Pearson correlation

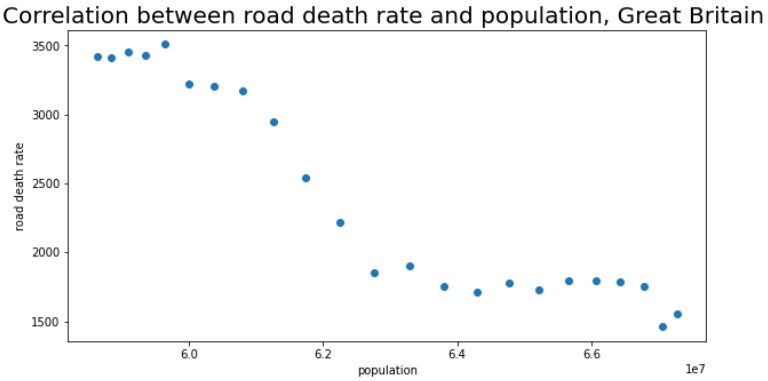


Figure 4.34 Correlation between road deaths and population

-1> ($r_{population\text{-}deaths}$= -0.94) > -0.5 hence having strong negative relationship between population and the road deaths

22. Find the correlation between the road death rate and the number of deaths.

    Use Pearson correlation

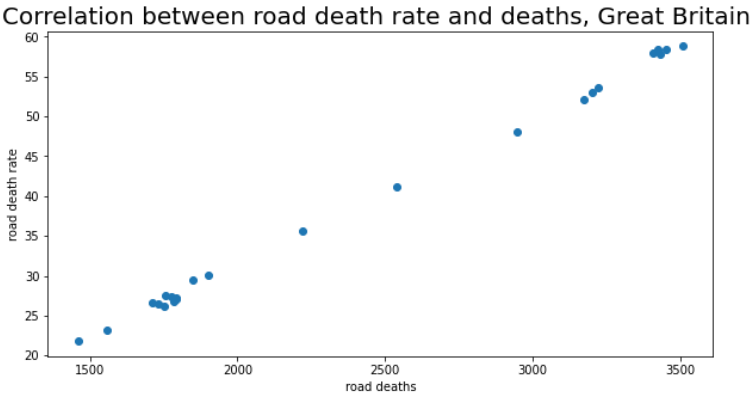
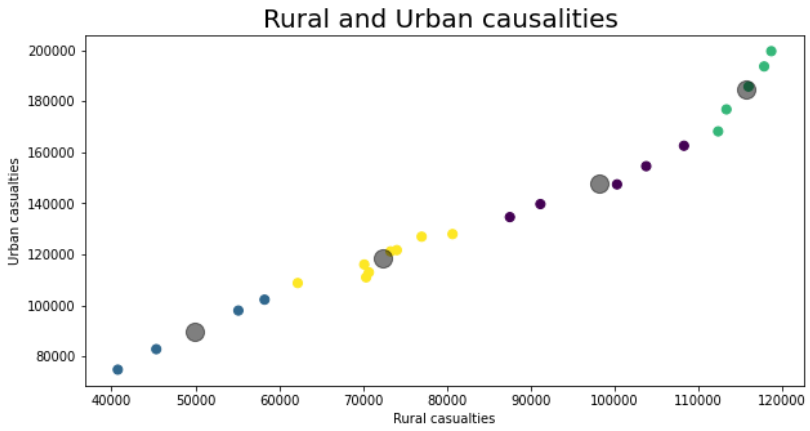


Figure 4.35 Correlation between road death rate and road deaths

1> ($r_{deaths\text{-}death\ rate}$= 0.99) > 0.5 hence having strong positive relationship between road deaths and the road deaths

23. Does the type of area (rural or urban) have a relationship with the causality risk level?




```
crosstab = pd.crosstab(data_risk_group["Urban rural"], data_risk_group["Rating"])
w, x, y, z = stats.chi2_contingency(crosstab)
print("The Chi Square value is:", w)
print("The pvalue is:", x)
print("The value for degree of freedom is :", y)
print("Expected cell counts is:", z)
print("\n")

alpha = 0.01 #alpha is 0.01 or level of confidence is 99%
if x < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected, Urban rural and Rating have a relationship. Knowing the value of one variable does help you predict the value of another variable.")
else:
    print("The null hypothesis is accepted, Urban rural and Rating have no relationship. It does not help you predict the value of another variable")
```

```
The Chi Square value is: 384.0022616339667
The pvalue is: 3.538627696604785e-77
The value for degree of freedom is : 9
Expected cell counts is: [[2.47450689e+02 7.17379978e+01 4.53583923e+02 4.47227391e+02]
 [5.86174172e+01 1.69936732e+01 1.07447339e+02 1.05941571e+02]
 [4.05656866e-01 1.17603275e-01 7.43580201e-01 7.33159658e-01]
 [2.38526237e+02 6.91507257e+01 4.37225158e+02 4.31097879e+02]]
```

Figure 4.36 Chi square statistics between area type and causality risk

Use hypothesis testing, 2 categorical variables hypothesis testing chi square

$\alpha = 0.01$

H0: Urban rural and risk level have no relationship

H1: Urban rural and risk level have a relationship

P value < α

$3.5 \times e^{-77} < 0.01$

P is lower than the alpha value so H0 is rejected and conclude that urban rural and risk level have a relationship.

24. What are the summery highlights identifying relationship between different features considering road causalities in the United Kingdom in 2021?

1. There is a positive relationship between vehicle traffic and the population.
2. There is a strong negative relationship between population and road deaths.
3. There is a strong positive relationship between road deaths and death rate.

4. Urban, rural, and risk level have a relationship in terms of the number of causalities.

2.5.3 Level 03: Machine Learning

Unsupervised Learning

K-means Clustering

25. Use K-means clustering to identify risk and causality levels.

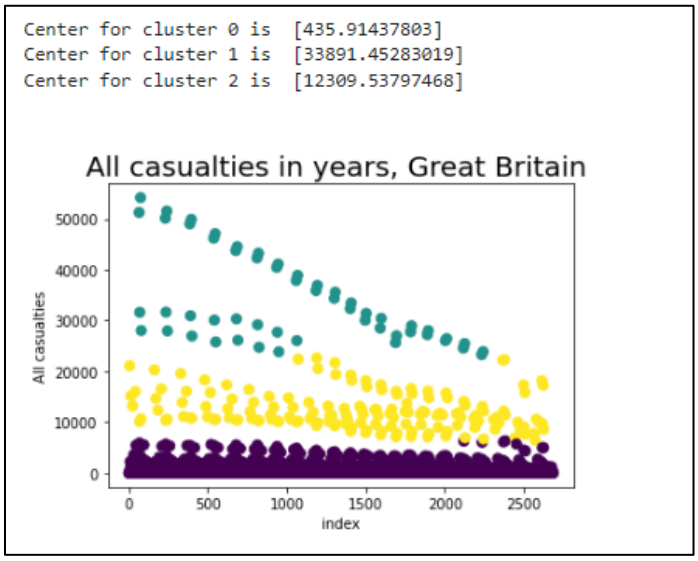


Figure 4.37 K-means clustering risk and causality levels

There are 3 clusters that can classify the causality risk level.

Number of causalities <= 435: 'Low Risk'

435 < Number of causalities <= 12309': 'Medium Risk'

12309 < Number of causalities <= 33891: 'High Risk'

Number of causalities > 33891: 'Extremely High'

Supervised Learning

26. Provide Nearest Neighbour classification models for predicting the risk causality level based on road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, highway), and rate the modules on accuracy.

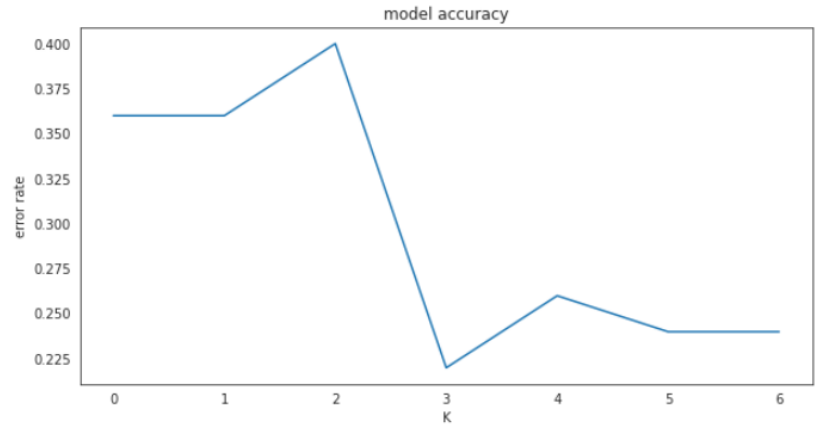
Nearest Neighbor:

	Road user	Casualty sex	Urban rural	Built up roads	Rating
2563	Pedestrian	Unknown	Urban	Built up road	1
2564	Pedestrian	Unknown	Rural	Built up road	1
2565	Pedestrian	Male	Urban	Motorway	1
2566	Pedestrian	Male	Urban	Built up road	2
2567	Pedestrian	Male	Urban	Non built up road	1
...	...	...	...	...	...
2682	Other vehicle	Female	Urban	Built up road	1
2683	Other vehicle	Female	Urban	Non built up road	1
2684	Other vehicle	Female	Rural	Motorway	1
2685	Other vehicle	Female	Rural	Built up road	1
2686	Other vehicle	Female	Rural	Non built up road	1

Before fit in to the model need to use encoding. Encoding maps categorical values to numerical as bellows,

```
{'High Risk': 0, 'Low Risk': 1, 'Medium Risk': 2}
{'Built up road': 0, 'Motorway': 1, 'Non built up road': 2}
{'Rural': 0, 'Unallocated': 1, 'Urban': 2}
{'Female': 0, 'Male': 1, 'Unknown': 2}
{'Bus or coach': 0, 'Car (Includes taxis and minibus)': 1, 'HGV': 2,
'Motorcycle': 3, 'Other vehicle': 4, 'Pedal cycle': 5, 'Pedestrian': 6, 'Van / Goods 3.5 tonnes mgw or under': 7}
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.96      0.59      0.73        39
           2       0.36      0.90      0.51        10

    accuracy                           0.64        50
   macro avg       0.44      0.50      0.41        50
weighted avg       0.82      0.64      0.67        50

[[ 0  0  1]
 [ 1 23 15]
 [ 0  1  9]]
```




```
Predicted classification is [1]
Predicted probabilities are [[0. 0. 1.]]
```

Figure 4.38 Nearest neighbor classification matrices for predicting the risk causality

The accuracy of the model is 0.64, and in this case, the error rate of the model is decreasing while k increases. With the precision and recall values, the model prediction for "high risk" and "medium risk" causality categories are not satisfactory. The data set is high bias with "low risk" data model face with over fitting. Therefore, the nearest neighbor algorithm needs more data to obtain a good a predictive model for the data set.

27. Use different regression modules to predict the death rate from deaths in Great Britain and assess the model based on the following factors:

    Confusion matrix

    AUC/ROC

    Accuracy

    Precision

    Recall

    R-squared (and Adjusted R-squared)

    RMSE

Liner Regression

According to the linear regression model,

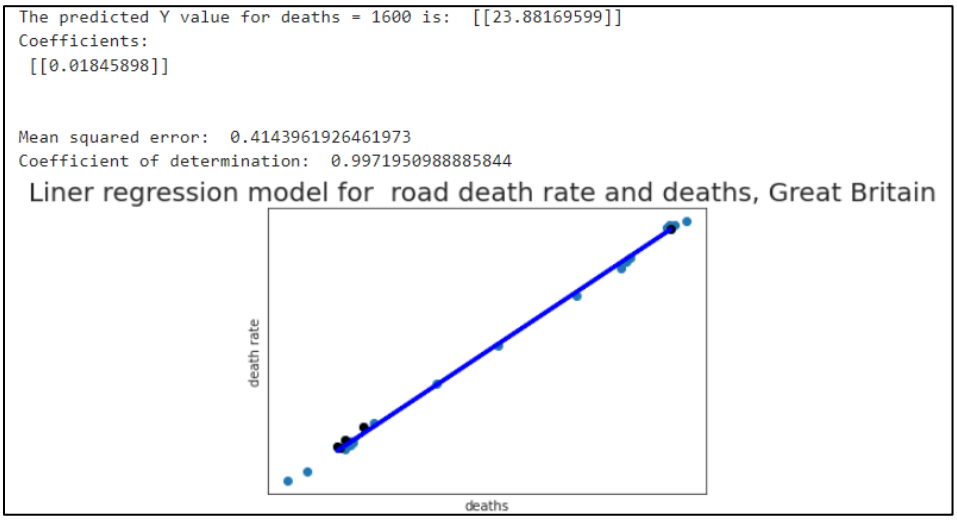


Figure 4.39 Linear regression matrices for predicting road death rate

Coefficient of determination is 0.999. Therefore the road deaths and death rate has very strong positive liner relationship. Mean square error is very low value and prediction accuracy is high. According to that model predicts road deaths = 1600 can have death rate of 23.88.

Multiple Liner Regression

28. Use multiple liner regression to predict the death rate from road deaths, population and vehicle traffic.

    Multiple Linear Regressions:

```
Predicted Y value:  [[22.47659126]]


Coefficients:
 [[-1.34231393e-05 -2.93000438e-07  1.71728372e-02]]


Mean squared error:  0.07694577970565147
Coefficient of determination:  0.9994767618552619
                            OLS Regression Results
==============================================================================
Dep. Variable:             death rate   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 2.169e+04
Date:                Fri, 16 Dec 2022   Prob (F-statistic):           9.51e-34
Time:                        13:26:37   Log-Likelihood:                0.48759
No. Observations:                  23   AIC:                             7.025
Df Residuals:                      19   BIC:                             11.57
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const             21.4400      4.022      5.331      0.000      13.022      29.858
vehicle traffic -1.413e-05    3.8e-06     -3.721      0.001   -2.21e-05   -6.18e-06
population      -3.096e-07   5.92e-08     -5.226      0.000   -4.34e-07   -1.86e-07
road deaths         0.0171      0.000     78.772      0.000       0.017       0.018
==============================================================================
Omnibus:                        0.242   Durbin-Watson:                   0.403
Prob(Omnibus):                  0.886   Jarque-Bera (JB):                0.204
Skew:                          -0.189   Prob(JB):                        0.903
Kurtosis:                       2.734   Cond. No.                     4.66e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.66e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figure 4.40 multiple linear regression matrices for predicting road death rate

The model

Based on Stats model

Y = 21.44 -0.00001413* vehicle traffic -0.0000003096* population+ 0.0171*road deaths

R-squared value is 1

Adjusted R-squared value is 1

F-Statistics

P value < 0.05 (95% confidence level) and the specified model are significantly different from the base model.

Predict death rate for

Vehicle traffic = 340000.0, Population = 70000000.0, Road deaths = 1600.0

Death rate from the model

Y = 21.44 - 340000 * 0.00001413 - 0.000000309 * 70000000 + 0.0171*1600

= 21.44 - 4.804 - 21.6 + 27.36

= 22.39 ≈ 22.47 (predicted value from python model)

Mean square error = 0.076 and coefficient of determination is 0.99. Therefore can conclude that the model is high accurate. But the contribution from vehicle traffic and population values for the model is low compare to road deaths.

29. Can we predict the road causalities from road collisions in 2021 using liner regression?

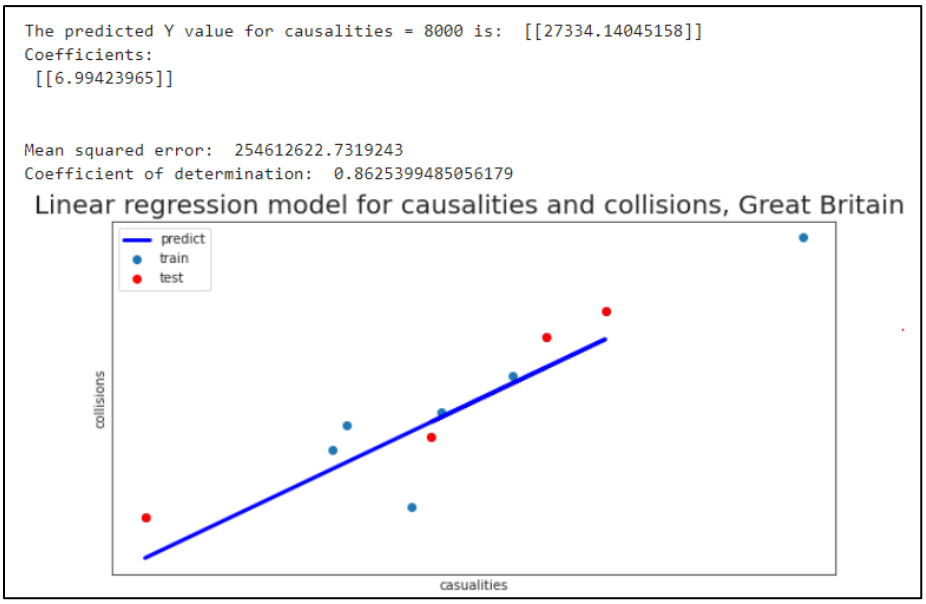
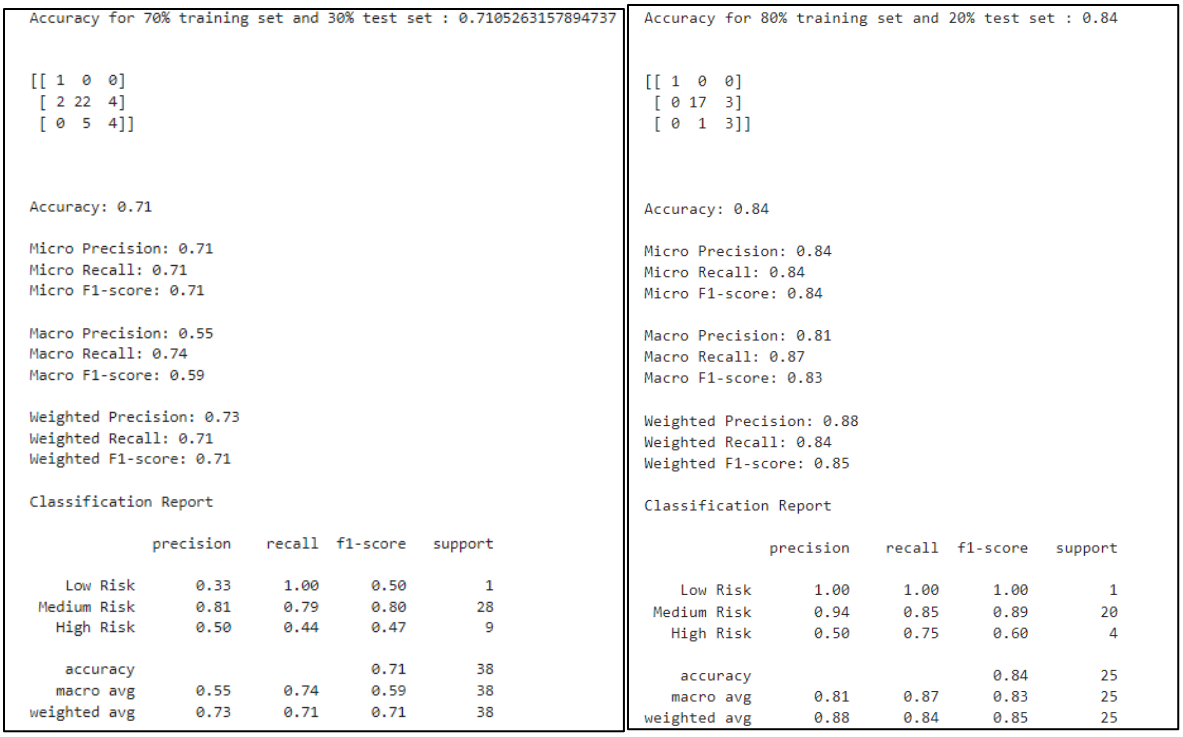


Figure 4.41 Linear regression matrices for predicting road causalities from road collisions

The red colour scatters are the actual test data and blue colour line is the predicted regression model. The squared difference (mean squared error) is high. The liner regression model having high mean squared error therefore the module needs to be improved with sufficient amount of data.

Decision Tree

30. Use a decision tree for data classification and predict road causality levels.



```
Accuracy for 70% training set and 30% test set : 0.7105263157894737


[[ 1  0  0]
 [ 2 22  4]
 [ 0  5  4]]



Accuracy: 0.71

Micro Precision: 0.71
Micro Recall: 0.71
Micro F1-score: 0.71

Macro Precision: 0.55
Macro Recall: 0.74
Macro F1-score: 0.59

Weighted Precision: 0.73
Weighted Recall: 0.71
Weighted F1-score: 0.71

Classification Report

              precision    recall  f1-score   support

    Low Risk       0.33      1.00      0.50         1
 Medium Risk       0.81      0.79      0.80        28
   High Risk       0.50      0.44      0.47         9

    accuracy                           0.71        38
   macro avg       0.55      0.74      0.59        38
weighted avg       0.73      0.71      0.71        38
```

```
Accuracy for 80% training set and 20% test set : 0.84


[[ 1  0  0]
 [ 0 17  3]
 [ 0  1  3]]



Accuracy: 0.84

Micro Precision: 0.84
Micro Recall: 0.84
Micro F1-score: 0.84

Macro Precision: 0.81
Macro Recall: 0.87
Macro F1-score: 0.83

Weighted Precision: 0.88
Weighted Recall: 0.84
Weighted F1-score: 0.85

Classification Report

              precision    recall  f1-score   support

    Low Risk       1.00      1.00      1.00         1
 Medium Risk       0.94      0.85      0.89        20
   High Risk       0.50      0.75      0.60         4

    accuracy                           0.84        25
   macro avg       0.81      0.87      0.83        25
weighted avg       0.88      0.84      0.85        25
```

Model 1

Training set and 70% test set: 30%

Model2

Training set and 80% test set: 20%

Figure 4.42 Decision tree models for predicting road causality risk levels

The model 1 accuracy is 71.0%. Considering the precision and recall values, the model predictions for "high risk" and "medium risk" causality categories are not satisfactory. But improving the choice of training set was able to reach (model 2) good accuracy (84%) and recall and precision values.

31. How useful are the different factors (road user, casualty sex, urban-rural, built-up roads) when deciding the causality risk levels?

	Accident year	Road user	Casualty sex	Urban rural	Built up roads	All casualties	Rating
0	2000	Pedestrian	Unknown	Urban	Built up road	43	Low Risk
1	2000	Pedestrian	Unknown	Rural	Built up road	5	Low Risk
2	2000	Pedestrian	Unknown	Rural	Non built up road	2	Low Risk
3	2000	Pedestrian	Unknown	Unallocated	Non built up road	1	Low Risk
4	2000	Pedestrian	Male	Urban	Motorway	15	Low Risk
...	...	...	...	...	...	...	...
2682	2021	Other vehicle	Female	Urban	Built up road	384	Low Risk
2683	2021	Other vehicle	Female	Urban	Non built up road	2	Low Risk
2684	2021	Other vehicle	Female	Rural	Motorway	4	Low Risk
2685	2021	Other vehicle	Female	Rural	Built up road	73	Low Risk
2686	2021	Other vehicle	Female	Rural	Non built up road	57	Low Risk

2687 rows × 7 columns

```
Importance of features:
[0.49788766 0.18343901 0.15392864 0.16474469]
```

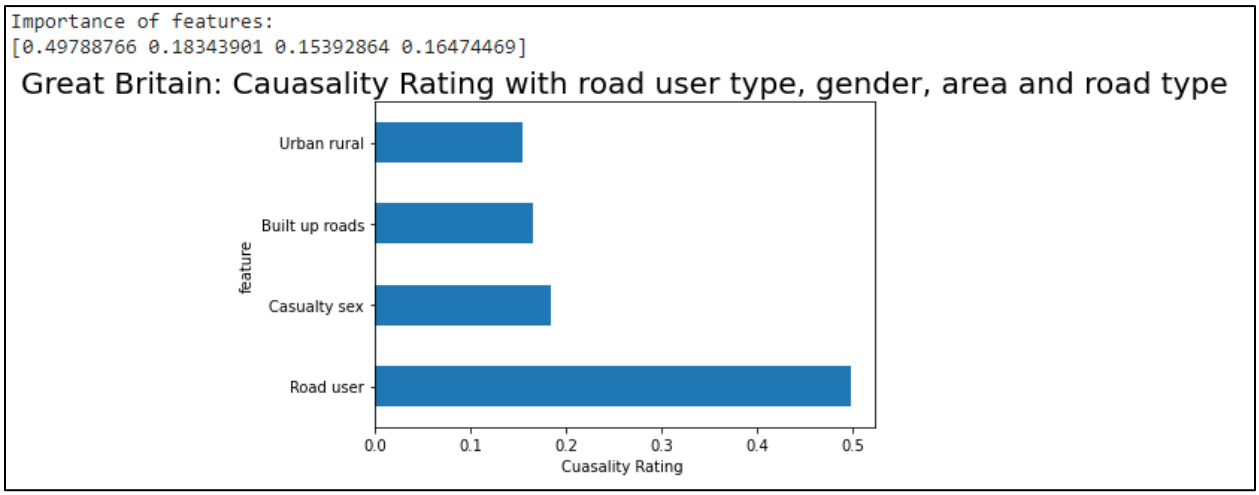


Figure 4.43 road user, casualty sex, urban-rural, built-up roads feature importance for predicting road causality risk levels

Using ExtraTreesClassifier, Road User is the most importance factor for deciding causality risk level. Secondly the causality gender and other two factors are in similar level of importance.

2.5.4 Level 04: Deep Learning

32. Use neural network algorithms to predict the risk level (low risk, medium risk, high risk, or extremely high risk) based on 4 different factors: road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, and highway). Evaluate the model based on performance.

Neural Network:

	Road user	Casualty sex	Urban rural	Built up roads	Rating
2563	Pedestrian	Unknown	Urban	Built up road	1
2564	Pedestrian	Unknown	Rural	Built up road	1
2565	Pedestrian	Male	Urban	Motorway	1
2566	Pedestrian	Male	Urban	Built up road	2
2567	Pedestrian	Male	Urban	Non built up road	1
...	...	...	...	...	...
2682	Other vehicle	Female	Urban	Built up road	1
2683	Other vehicle	Female	Urban	Non built up road	1
2684	Other vehicle	Female	Rural	Motorway	1
2685	Other vehicle	Female	Rural	Built up road	1
2686	Other vehicle	Female	Rural	Non built up road	1

124 rows × 5 columns

```
Model Summary
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_28 (Dense)            (None, 4)                 20

 dense_29 (Dense)            (None, 40)                200

 dense_30 (Dense)            (None, 40)                1640

 dense_31 (Dense)            (None, 4)                 164

=================================================================
Total params: 2,024
Trainable params: 2,024
Non-trainable params: 0
_________________________________________________________________
```

```
Epoch 100/100
2/2 [==============================] - 0s 32ms/step - loss: 0.4344 - accuracy: 0.8243 - val_loss: 0.8077 - val_accuracy: 0.6842
1/1 [==============================] - 0s 90ms/step
```

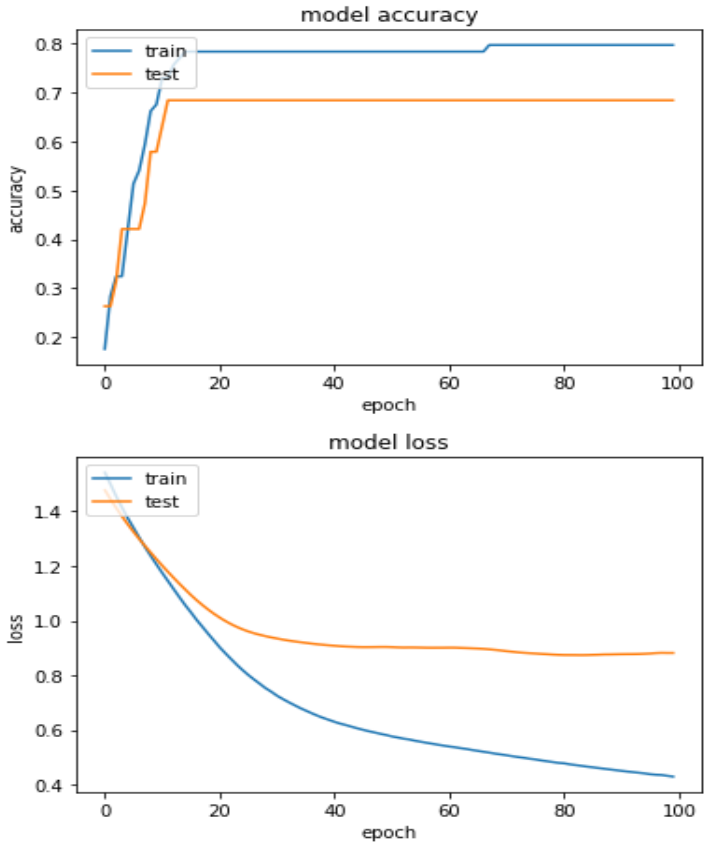


Figure 4.44 Neural Network model matrices for predicting road causality risk levels

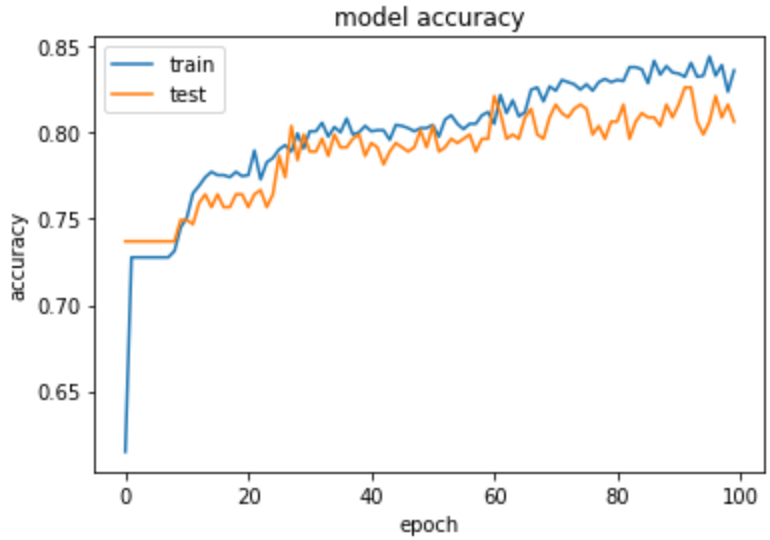
The ANN model (training set and 25% test set: 75%) with 50 batch size and 100 epochs.

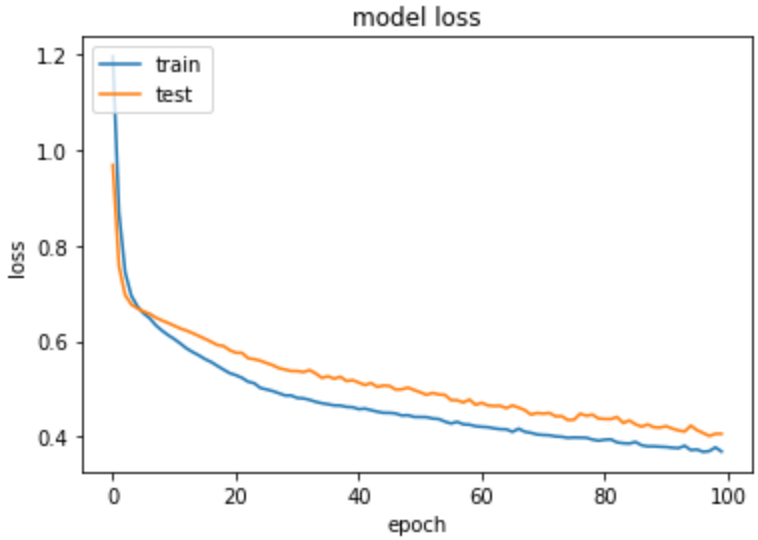
The target variable is causality risk level (Low Risk, Medium Risk, High Risk, and Extremely High).

The accuracy of the model, 0.8243 and loss is 0.4344 so model is good enough for prediction. But considering the model loss it seems training dataset is over fit for the model. The main reason should be the low size (124 rows) of the dataset and model, which can be improved by adding more data for the analysis.

Increased data set size (2687 rows) improved the model as bellow,

```
Epoch 100/100
33/33 [==============================] - 0s 4ms/step - loss: 0.3685 - accuracy: 0.8362 - val_loss: 0.4056 - val_accuracy: 0.8065
```





33. What are the summery highlights designing predictive models in the United Kingdom in 2021?

1. The linear regression model provides a high-accuracy model for predicting the death rate from road deaths.
2. A multiple liner regression model provides a high-accuracy model for predicting death rates using vehicle traffic, population, and road deaths.
3. The linear regression model provides a low-accuracy model for predicting road collisions from road causalities. The model needs to be improved with a sufficient amount of data.
4. KNN does not predict causality risk levels well because data is high biased with "low risk" data and model face with over fitting. Need to improve the model with more data size.
5. A decision tree provides a good accuracy model for predicting road causality risk levels using road user type, gender, area, and road type.

6. ExtraTreesClassifier is useful for determining the importance of various factors in determining causality risk level.
7. ANN provides good accuracy in the model for predicting road causality risk levels using road user type, gender, area, and road type. However, the model has an over fitting problem.

5. Evaluation

Different algorithms can be evaluated based on the different matrices (classification accuracy, loss, confusion matrix, area under the curve, mean absolute error, mean squared error). Section 4 describes the evaluation for each test. Overall, the data set is biased and results in over fitting; we need to increase the data size to handle the over fitting. If increasing the data size is not an option, must deal with the issue by balancing the dataset using oversampling or under sampling methods. The oversampling method creates new samples by duplicating existing minority class data, whereas the under sampling method deletes majority class data to balance the number of instances in both classes. The careful consideration needed when using oversampling is because it can increase the chance of over fitting since it duplicates the minority-class examples exactly. The SMOTE technique or any other algorithmic-level techniques (such as the threshold method, one-class learning, and cost-sensitive learning) can be used for the same purpose, but further research needs to be done in this area (Kotsiantis et al, 2006).

6. Recommendations

Although lots of research has been done on this field of road safety monitoring and predictions, there was no research found on the area of predicting causality risk level based on road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, highway) in the United Kingdom. The first level employs descriptive and inferential analyses to determine the impact of each factor on road causality. Car occupants (including car drivers and car passengers) are always the most vulnerable road users, and car drivers are the most vulnerable driver group. The main reason for this is that people are mostly using cars for commuting because of the low quality of service provided by public transport. Current economic policies that reduce public investment in public transport are leading to a decline in the quality of these services and promoting the growth of private car use. Addressing vulnerable road users' road safety appears to be a key determinant of whether more sustainable and healthier modes of transportation can increase or maintain their share of total transport. For example, promoting non-motorized transport such as cycling and walking. To achieve this, the government needs to focus on increasing the safety and appeal of cycling.

The age group under 16 is the most vulnerable for road deaths, and there is a higher risk for cyclists and pedestrians to be killed or sustain severe injuries in road crashes compared with car occupants. As a result, parents are concerned about traffic risks on their child's way to school, and this fear greatly influences whether youngsters walk to school. 75% of the traffic fatalities happened in densely populated areas, which emphasises the necessity of putting in place road safety measures in urban areas where diverse road users are more likely to interact.

The gender difference is especially pronounced among road casualties, and males represent 62% of the total number of victims. This has been related to a combination of differences in exposure and risk-taking attitudes. Men had more average access to motor vehicles than women, particularly those with the highest fatality rates, such as motorcycles. Furthermore, they are more likely to participate in dangerous behaviours, such as speeding and driving under the influence of alcohol, which increases both the likelihood and severity of crashes. It is really important to encourage safer learning and road behaviours and provide education and training, focusing on young male and inexperienced drivers.

7. Conclusion

This report conducted a different level of analysis on road fatalities based on yearly (1999–2021) data in Great Britain and examines the success of the United Kingdom government's investment in national road safety.

The report conducted a literature and statistical analysis on how the United Kingdom's road fatality statistics compare to those of other countries throughout the world, as well as the United Kingdom's efforts to reduce traffic-related injuries and deaths. The United Kingdom has some of the finest road safety records in Europe and the globe. The number of road deaths has been decreasing since 2003, despite rising population and vehicle traffic. That is a good trend, and it needs to continue with the current economy expanding. Therefore, it is important to identify the focus area for further improvement, and this analysis tries to answer major societal concerns about trends in road fatality risk levels.

The report examines a secondary dataset from the Department of Transportation in four ways. It explores the data with graphs and numerical summaries at the first level of analysis and identifies the most important four high-level factors (road user type, gender, area, and road type) for deciding causality risk level. Then ExtraTreesClassifier helps identify different levels of importance among different factors. Regression models, decision trees, and neural networks can be used to design well-accurate predictive models to decide the risk causality levels based on road user type, gender, area, and road type and rate.

8. Future Work

Before considering further analysis the first point need to handle the data imbalance problem. Data is high biased with "low risk" data and predictive model face with over fitting. With the constrained time frame, the report could not cover time series analysis and forecasting. Hence, important future work needs to be carried out to uncover the seasonality trends and insights in the dataset. According to the report, there is a downward trend in the number of casualties within the considered timeframe. The analysis shows some interesting deviations in 2021 because of the COVID pandemic, and it would be a great future research area to analyze. The leanings will provide future techniques for reducing road fatalities. The majority of casualties occurred on urban roads compared to rural roads and highways. Built-up roads have higher causality rates than non-built-up roads and motorways. This fact proves that urbanization is increasing the causality risk level considerably. The report focused on analyzing the number of fatalities to identify the most vulnerable road user group, but it is really important to compare the fatality rate to increase the accuracy of the predictive models because the fatality rate depends on the distance people travel.

It's a very significant factor that drives the causality risk level. The mode of transportation that people use for commuting is also an important factor in determining the level of causality risk. The mode of transportation is always changing depending on the age group. With under-16 age group, pedal cycles are the most popular way of commuting. The motorcycles are mostly used by young drivers aged 16 or older. For other age groups, cars are the most preferred choice. Therefore, the age group needs to be considered when determining the causality risk level. In the future, the causality age group needs to be considered as a factor for the predictive model. Because males have greater access to various modes of transportation than women, the causality risk varies with causal gender. Men, for example, have always preferred motorcycles over other modes of transportation, but women do not. Men are also more likely to participate in dangerous behaviors, such as speeding and driving under the influence of alcohol.

Driving and rider behaviour, environmental factors such as weather or road type, and driver experience all have an indirect impact on assessing causation risk levels and are vital to consider.

9. Bibliography

1. Alzubi, J., Nayyar, A. and Kumar, A., 2018, November. Machine learning from theory to algorithms: an overview. In *Journal of physics: conference series* (Vol. 1142, No. 1, p. 012012). IOP Publishing.
2. Arel, I., Rose, D.C. and Karnowski, T.P. (2010) Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *IEEE computational intelligence magazine*, *5*(4), pp.13-18.
3. Azevedo, A. and Santos, M.F. (2008) KDD, SEMMA and CRISP-DM: a parallel overview. *IADS-DM*.
4. Brachman, R.J. and Anand, T., 1994, July. The Process of Knowledge Discovery in Databases: A First Sketch. In *KDD workshop* (Vol. 3, pp. 1-12).
5. Clark, A. (2018) The machine learning audit-CRISP-DM Framework. *Isaca Journal*, *1*, pp.42-47.
6. Department for Transport (2022) *Road traffic estimates in Great Britain: 2021.* [Online]. *GOV.UK*. Available from: https://www.gov.uk/government/statistics/road-traffic-estimates-in-great-britain-2021 [Accessed: December 17, 2022].
7. Department of Transport (2011) *Strategic framework for road safety*, *GOV.UK*. GOV.UK. Available at: https://www.gov.uk/government/publications/strategic-framework-for-road-safety (Accessed: January 2, 2023).
8. Department of Transport (2013) *Strategic framework for road safety Action Plan – Final Progress Update 2013.* GOV.UK. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/246071/2013-roads-strategic-framework-progress.pdf.(Accessed: January 1, 2023).
9. Department of Transport (2015) *'Working Together to Build a Safer Road System British Road Safety Statement'*. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/487949/british_road_safety_statement_web.pdf.(Accessed: January 1, 2023).
10. Department of Transport (2020) *Reported Road Casualties Great Britain: 2019 Annual Report Moving Britain Ahead*, *GOV.UK*. GOV.UK. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/928205/reported-road-casualties-gb-annual-report-2019.pdf (Accessed: January 2, 2023).
11. Douglas, G. (2019) Open Access proceedings Journal of Physics: Conference series.
12. Fayyad, U.M., Piatetsky-Shapiro, G. and Smyth, P. (1996) August. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *KDD* (Vol. 96, pp. 82-88).

13. GOV.UK (2021) *Reported road collisions, vehicles and casualties tables for Great Britain.*[Online]. Available from: at: https://www.gov.uk/government/statistical-data-sets/reported-road-accidents-vehicles-and-casualties-tables-for-great-britain#factors-contributing-to-collisions-and-casualties-ras07 [Accessed: December 17, 2022].
14. GOV.UK. (2018). *Road traffic statistics (TRA)*. [online] Available at: https://www.gov.uk/government/statistical-data-sets/road-traffic-statistics-tra. (Accessed: December 31, 2022).
15. International Traffic Safety Data and Analysis Group (2021) *Road Safety Annual Report 2013*. International Transport Forum. Available at: https://www.itf-oecd.org/sites/default/files/united-kingdom-road-safety.pdf.(Accessed: January 1, 2023).
16. Jara, A. (2022) *How neural networks can be used for data mining*, *GetSmarter Blog*. Available at: https://www.getsmarter.com/blog/career-advice/how-artificial-neural-networks-can-be-used-for-data-mining/ (Accessed: December 31, 2022).
17. MacroTrends (n.d.) *U.K. population 1950-2022*. [Online]. Available from: https://www.macrotrends.net/countries/GBR/united-kingdom/population#:~:text=The%20population%20of%20U.K.%20in,a%200.52%25%20increase%20from%202018. [Accessed: December 17, 2022].
18. Nations, U. (n.d.). *Road Safety. [online] United Nations. Available at: https://www.un.org/en/safety-and-security/road-safety.*
19. *Road safety* (n.d.) *World Health Organization*. World Health Organization. Available at: https://www.who.int/health-topics/road-safety#tab=tab_1 (Accessed: January 1, 2023).
20. roadtraffic.dft.gov.uk. (n.d.) Road traffic statistics - About. [Online]. Available from: https://roadtraffic.dft.gov.uk/custom-downloads/road-accidents/reports/f062b60e-77bc-48f4-a534-f2e506ca151c [Accessed 19 Dec. 2022].
21. Silva, P.B., Andrade, M. and Ferreira, S. (2020) Machine learning applied to road safety modeling: A systematic literature review. *Journal of traffic and transportation engineering (English edition)*, *7*(6), pp.775-790.
22. Wang, C., Quddus, M.A. and Ison, S.G. (2013) The effect of traffic and road characteristics on road safety: A review and future research direction. *Safety science*, *57*, pp.264-275.
23. Wegman, F., Lynam, D. and Nilsson, G. (2002) SUNflower: a comparative study of the developments of road safety in Sweden, the United Kingdom, and the Netherlands. *SWOV, Leidschendam*, pp.1-147.

24. Wirth, R. and Hipp, J. (2000) April. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (Vol. 1, pp. 29-39).

25. World Health Organization. (2013). *WHO global status report on road safety 2013: supporting a decade of action. World Health Organization*. Available at: https://apps.who.int/iris/handle/10665/78256 (Accessed: January 1, 2023).

26. World Health Organization. (2015). Global status report on road safety 2015. World Health Organization. Available at: https://apps.who.int/iris/handle/10665/189242 (Accessed: January 1, 2023).

27. Young, W., Sobhani, A., Lenné, M.G. and Sarvi, M. (2014) Simulation of safety: A review of the state of the art in road safety simulation modelling. *Accident Analysis & Prevention*, *66*, pp.89-103.

28. Kotsiantis, S., Kanellopoulos, D. and Pintelas, P. (2006) Handling imbalanced datasets: A review. GESTS international transactions on computer science and engineering, 30(1), pp.25-36.

Appendix 1: Narrow Objectives

1.5.1 Level 01: Descriptive statistics analysis

1. Where does the United Kingdom rank in the world in terms of road causalities in 2021?
2. What is the trend in the death rate in the United Kingdom between 1999 and 2021?
3. What is the yearly trend in road causalities in the United Kingdom?
4. Which area (rural or urban) has the highest rate of road causalities?
5. Who is the most vulnerable road user group in UK in terms of road causalities?
6. What is the most vulnerable gender group in the UK in terms of road causalities?
7. What is the most vulnerable road user age group in the UK in terms of road causalities?
8. What is the most dangerous type of road in the United Kingdom?
9. What is the summery highlights in road causalities in terms of road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, highway) in the United Kingdom in 2021?
10. What is the most vulnerable driver age group in the UK for road collisions?
11. Who are the most susceptible road users/drivers in the United Kingdom in terms of road causalities /collisions?

1.5.2 Level 02: Inferential Statistics Analysis

Hypothesis tests for deciding population parameters:

12. Is the average number of causalities in 2021 greater than zero?
13. Does this suggest that the number of causalities in different age groups will be equal to zero?
14. Is it true that the number of causalities in men is higher than the number of causalities in women?
15. Is it true that the number of causalities in urban regions exceeds the number of causalities in rural areas?
16. Is there a difference in the average number of causalities and collisions between age groups?
17. Is it true that all road users have the same average number of fatalities?

Measure the relationship between factors:

18. Find the correlation between vehicle traffic and the population over the years in Great Britain.

19. Find the correlation between population and the number of road deaths over the years in Great Britain.
20. Find the correlation between the road death rate and the number of deaths.
21. Does the type of area (rural or urban) have a relationship with the causality risk level?

2.5.3 Level 03: Machine Learning

Supervised Learning

22. Provide nearest neighbour classification models for predicting the risk causality level based on road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, highway), and rate the modules on accuracy.
23. Use different regression models to predict the death rate from deaths in Great Britain and assess the model based on the evaluation matrices.
24. Use multiple linear regressions to predict the death rate from road deaths, population and vehicle traffic.
25. Can we predict the road causalities from road collisions in 2021 using liner regression?
26. Use a decision tree for data classification and predict road risk causality levels.
27. How useful are the different factors (road user, casualty sex, urban-rural, built-up roads) when deciding the causality risk levels?

Unsupervised Learning

28. Use K-means clustering to group the dataset based on risk and causality levels.

2.5.4 Level 04: Deep Learning

29. Use neural network algorithms to predict the risk level (low risk, medium risk, high risk, or extremely high risk) based on 4 different factors: road user type (pedestrian, pedal cycle, motorcycle, car, bus, coach, van, HGV, other vehicle), gender (female, male), area (urban or rural), and road type (built-up road, non-built-up road, and highway). Evaluate the model based on its performance.

Appendix 2: Data Description

Data Set 1:

Title: RAS0404: International comparisons

Description: International comparisons of road deaths, number and rates for different road users, by selected countries

Source: (GOV.UK, 2021)

Size: 40 rows 25 columns

International Comparison.xlsx

	Country	1999	2000	2001	2002	2003	2004	2005	2006	2007	...	2013	2014	2015	2016	2017	2018	2019	2020	2021	DeathsRate 2021
0	Great Britain	3423	3409	3450	3431	3508	3221	3201	3172	2946	...	1713	1775	1730.0	1792.0	1793.0	1784	1752	1460	1558.0	23.804771
1	Northern Ireland	141	171	148	150	150	147	135	126	113	...	57	79	74.0	68.0	63.0	55	56	56	50.0	26.293219
2	United Kingdom	3564	3580	3598	3581	3658	3368	3336	3298	3059	...	1770	1854	1804.0	1860.0	1856.0	1839	1808	1516	1608.0	23.875032
3	Austria	1079	976	958	956	931	878	768	730	691	...	455	430	475.0	432.0	413.0	409	416	344	362.0	40.440059
4	Belgium	1397	1470	1486	1306	1214	1162	1089	1069	1067	...	723	727	755.0	637.0	620.0	604	644	499	516.0	44.665212

Figure: World: Rate of road deaths in 2021 per million populations

Data Dictionary:

Variable	Variable description?	Variable format: string, numeric, etc.	Type of the variables: categorical, continuous, and discrete
Country	Country that data belongs to.	String	categorical
1999 -2021	Road fatality as one being due to a road accident where death occurs within 30 days of the accident.	Numeric	discrete
DeathsRate 2021	Rate of road deaths in 2021 per million population	Numeric	continuous

Table: Data dictionary for dataset 1

Data Set 2:

Title: Great Britain: Population, Vehicle Traffic, and Road Deaths Great Britain from 1999 to 2021

Description: This is a customized dataset to analyses population vs. vehicle traffic vs. road deaths in Great Britain from 1999 to 2021

Aggregated dataset with below sources:

U.K. Population

United Nations - World Population Prospects

(MacroTrends, n.d.)

U.K. Road Deaths

Department for Transport statistics

RAS0404: International comparisons of road deaths, number and rates for different road users, by selected countries

(GOV.UK, 2021)

U.K. Vehicle Traffic

Motor vehicle traffic (vehicle miles) by local authority in Great Britain, annual from 1993

(Department for Transport, 2022; GOV.UK., 2018)

Size: 4 rows 24 columns

Population_Vehicle_t
rafiic_Road_deaths_(

	Great Britain	1999	2000	2001	2002	2003	2004	2005	2006	2007	...	2012	2013	2014
0	vehicle traffic	2.901550e+05	2.895510e+05	2.934590e+05	3.002370e+05	3.020410e+05	3.057060e+05	3.058440e+05	3.103730e+05	3.132940e+05	...	3.045590e+05	3.057590e+05	3.144900e+05
1	population	5.863520e+07	5.885004e+07	5.909202e+07	5.935569e+07	5.964980e+07	5.999585e+07	6.038374e+07	6.080370e+07	6.126068e+07	...	6.380873e+07	6.430230e+07	6.477350e+07
2	road deaths	3.423000e+03	3.409000e+03	3.450000e+03	3.431000e+03	3.508000e+03	3.221000e+03	3.201000e+03	3.172000e+03	2.946000e+03	...	1.754000e+03	1.713000e+03	1.775000e+03
3	death rate	5.837790e+01	5.792689e+01	5.838352e+01	5.780406e+01	5.880992e+01	5.368705e+01	5.301096e+01	5.216788e+01	4.808958e+01	...	2.748840e+01	2.663979e+01	2.740318e+01

Figure: Great Britain: Population, Vehicle Trafiic, Road Deaths, and Death Rate from 1999 to 2021

Data Dictionary:

Variable	Variable description?	Variable format: string, numeric, etc.	Type of the variables: categorical, continuous, and discrete
vehicle traffic	Motor vehicle traffic (vehicle miles) by local authority in Great Britain	String	discrete
population	Number of humans in Great Britain	String	discrete

road deaths	Number of road deaths	String	discrete
death rate	Number of road deaths /Number of humans in Great Britain	Numeric	continuous

Table: Data dictionary for dataset 2

Data Set 3:

Title: Reported road casualties, Great Britain

Title: Reported road casualties, Great Britain

Description: Reported road casualties in Great Britain. Provided by severity, geography, year, and additional filters.

Source: Custom data report (roadtraffic.dft.gov.uk., no date)

Size: 2687 rows 6 columns

Road-Casualties_Great Britain_Reigon.xlsx

	Accident year	Road user	Casualty sex	Urban rural	Built up roads	All casualties
0	2000	Pedestrian	Unknown	Urban	Built up road	43
1	2000	Pedestrian	Unknown	Rural	Built up road	5
2	2000	Pedestrian	Unknown	Rural	Non built up road	2
3	2000	Pedestrian	Unknown	Unallocated	Non built up road	1
4	2000	Pedestrian	Male	Urban	Motorway	15

Figure: Great Britain: Number of casualties from 1999 to 2021

Data Set 4:

Title: RAS0202: Gender and age group

Description: Reported road casualties by road user type, gender and age, Great Britain, ten years up to 2021

Source: (GOV.UK, 2021)

Size: 462 rows 13 columns

Casualties_Demographics.csv

	Road user type	Sex	Age group	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	Pedestrians	Male	Under 16	4162	3861	3888	3688	3645	3486	3207	3105	2038	2507
1	Pedestrians	Male	16	289	280	256	232	237	269	220	189	146	159
2	Pedestrians	Male	17 to 20	1174	1072	982	957	861	945	798	802	449	569
3	Pedestrians	Male	21 to 24	1082	928	1002	874	875	843	736	727	443	494
4	Pedestrians	Male	25 to 29	1113	1040	1073	1052	970	1000	888	865	575	608

Figure: Great Britain: Number of causalities with age groups

Data Dictionary:

Variable	Variable description?	Variable format: string, numeric, etc.	Type of the variables: categorical, continuous, and discrete
Road user type	Bus or coach drivers, Bus or coach passengers, Car drivers, Car passengers, Heavy goods vehicle drivers, Heavy goods vehicle passengers, Light good vehicles passengers, Light goods vehicle drivers, Motorcycle passengers, Motorcycle riders, Other or unknown vehicle drivers, Other or unknown vehicle passengers, Pedal cyclists, Pedestrian	String	categorical
Gender	Female, Male	String	categorical
Age group	'Under 16','16','17 to 20','21 to 24','25 to 29','30 to 39','40 to 49','50 to 59','60 to 69','70 and over'	String	categorical
1999 -2021	Number of road casualties	Numeric	discrete

Table: Data dictionary for dataset 4

Data Set 5:

Title: RAS0501: Drivers involved by gender, age and road user type

Description: Drivers in reported fatal or serious collisions (FSC) by gender, road user type and age, Great Britain, ten years up to 2021

Source: (GOV.UK, 2021)

Size: 264 rows 14 columns

Collisions_Demographics.csv

	Driver type	Sex	Age group	Severity	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	Pedal cyclists	Female	Under 16	All collisions	380	306	289	265	262	274	221	229	222	244
1	Pedal cyclists	Female	16	All collisions	37	33	48	49	29	37	24	37	45	28
2	Pedal cyclists	Female	17-20	All collisions	222	208	220	214	202	182	154	154	149	126
3	Pedal cyclists	Female	21-24	All collisions	343	367	415	363	339	302	297	252	264	250
4	Pedal cyclists	Female	25-29	All collisions	584	582	659	586	563	548	575	545	466	466

Figure: Great Britain: All collisions by age group, 2021

Data Dictionary:

Variable	Variable description?	Variable format: string, numeric, etc.	Type of the variables: categorical, continuous, and discrete
Driver type	Pedal cyclists, Motorcycle riders, Car drivers, Bus or coach drivers, Light goods vehicle drivers, Heavy goods vehicle drivers, Other or unknown vehicle drivers	String	categorical
Gender	Female, Male	String	categorical
Age group	'Under 16','16','17 to 20','21 to 24','25 to 29','30 to 39','40 to 49','50 to 59','60 to 69','70 and over'	String	categorical
1999 - 2021	Number of road casualties	Numeric	discrete

Table: Data dictionary for dataset 5

Appendix 3: Python Scripts


```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

Data Set 1:

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from scipy import stats
import numpy as np

df = pd.read_excel('/content/gdrive/MyDrive/Colab Notebooks/Report/Final/International Comparison.xlsx')
df.head()
df.tail()
```

	Country	1999	2000	2001	2002	2003	2004	2005	2006	2007	...	2013	2(
35	Canada	2972	2903	2756	2921	2779	2731	2898	2884	2761	...	1923	18
36	Japan	10372	10403	10060	9575	8877	8492	7931	7272	6639	...	5152	48
37	New Zealand	509	462	455	404	461	436	405	391	422	...	253	2
38	Republic of Korea	10756	10236	8097	7222	7212	6563	6376	6327	6166	...	5092	47
39	USA	41717	41945	42116	42815	42643	42636	43443	42708	41259		32719	32(

```
df['Country'] = df['Country'].astype('string')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country         40 non-null     string
 1   1999            40 non-null     int64
 2   2000            40 non-null     int64
 3   2001            40 non-null     int64
 4   2002            40 non-null     int64
 5   2003            40 non-null     int64
 6   2004            40 non-null     int64
 7   2005            40 non-null     int64
 8   2006            40 non-null     int64
 9   2007            40 non-null     int64
 10  2008            40 non-null     int64
 11  2009            40 non-null     int64
 12  2010            40 non-null     int64
 13  2011            40 non-null     int64
 14  2012            40 non-null     int64
 15  2013            40 non-null     int64
 16  2014            40 non-null     int64
 17  2015            39 non-null     float64
 18  2016            38 non-null     float64
 19  2017            39 non-null     float64
 20  2018            40 non-null     int64
 21  2019            40 non-null     int64
 22  2020            40 non-null     int64
 23  2021            38 non-null     float64
 24  DeathsRate 2021  38 non-null    float64
dtypes: float64(5), int64(19), string(1)
memory usage: 7.9 KB
```

```
df.isna().any()
```

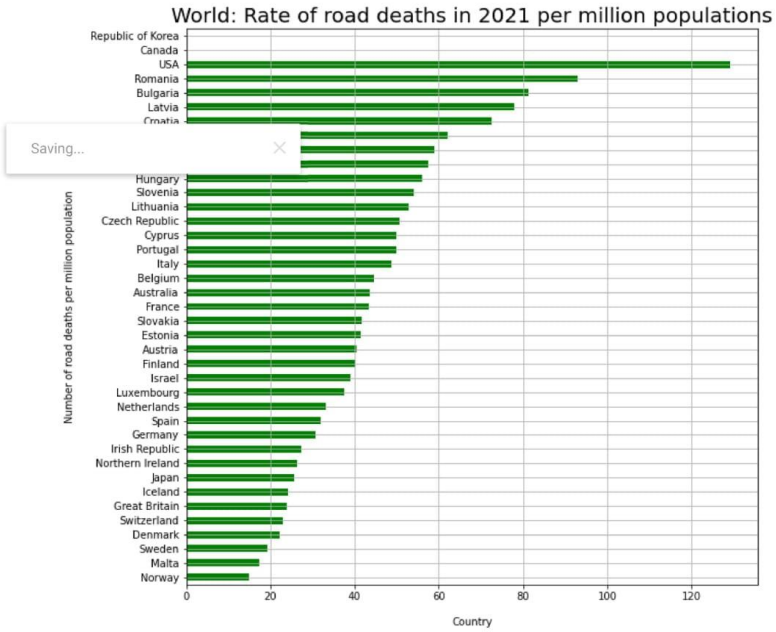
```
Country           False
1999              False
2000              False
2001              False
2002              False
2003              False
2004              False
```

```
2005               False
2006               False
2007               False
2008               False
2009               False
2010               False
2011               False
2012               False
2013               False
2014               False
2015                True
2016                True
2017                True
2018               False
2019               False
2020               False
2021                True
DeathsRate 2021     True
dtype: bool
```

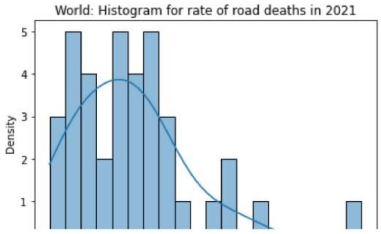
```
df.shape
```

```
(40, 25)
```

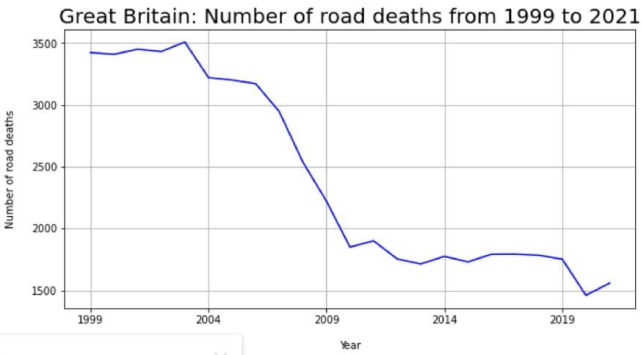
```
df = df.drop(df.index[[2]]) # Remove data for United Kingdom, hence Northern Ireland,Great Britain is having the data.
df_rate = df[["Country","DeathsRate 2021"]].set_index("Country").sort_values(by=['DeathsRate 2021'])
df_rate.plot(kind='barh',figsize=(10, 10), legend=False, color='green', rot=0);
plt.title("World: Rate of road deaths in 2021 per million populations", fontsize=20)
plt.xlabel("Country", labelpad=15)
plt.ylabel("Number of road deaths per million population", labelpad=15)
plt.grid()
plt.show()
```




```
sns.histplot(df_rate['DeathsRate 2021'] , bins=20, kde=True)
plt.title("World: Histogram for rate of road deaths in 2021")
plt.xlabel("Number of road deaths per million population")
plt.ylabel('Density')
plt.rcParams["figure.figsize"] = [10,5]
plt.show()
```

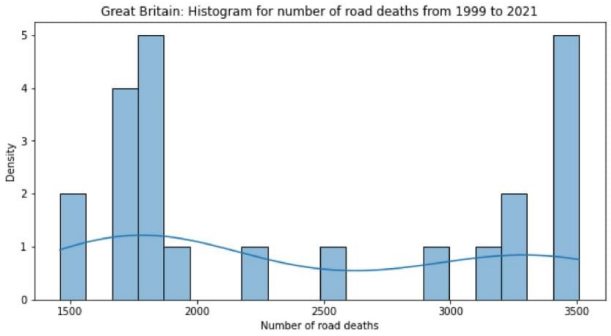



```
df_Great_Britain = df.iloc[[0]].set_index("Country")
df_Great_Britain = df_Great_Britain.drop(columns=["DeathsRate 2021"])
df_Great_Britain.T.plot(kind='line',figsize=(10, 5), legend=False, color='blue', rot=0);
plt.title("Great Britain: Number of road deaths from 1999 to 2021", fontsize=20)
plt.xlabel("Year", labelpad=15)
plt.ylabel("Number of road deaths", labelpad=15)
plt.grid()
plt.show()
```

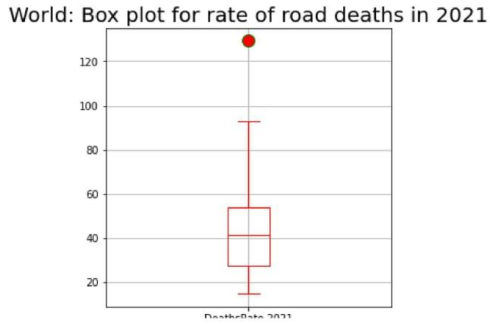


Saving...

```
df_Great_Britain_T = df_Great_Britain.T
sns.histplot(df_Great_Britain_T['Great Britain'] , bins=20, kde=True)
plt.title("Great Britain: Histogram for number of road deaths from 1999 to 2021")
plt.xlabel("Number of road deaths")
plt.ylabel('Density')
plt.rcParams["figure.figsize"] = [10,5]
plt.show()
```




```
flierprops = dict(marker='o', markerfacecolor='r', markersize=12,
                  linestyle='none', markeredgecolor='g')
df_rate.plot(kind='box',figsize=(5, 5), flierprops=flierprops, color='red');
plt.title("World: Box plot for rate of road deaths in 2021", fontsize=20)
plt.grid()
plt.show()
```




```
df_rate.describe()
```

	DeathsRate 2021
count	37.000000
mean	45.574082
std	23.279649
min	14.792024
25%	27.337343
50%	41.528918
75%	54.105183
max	129.303431

```
df_rate.median()
```

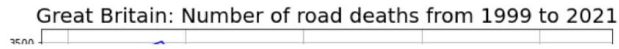
```
DeathsRate 2021    41.528918
```

Saving...

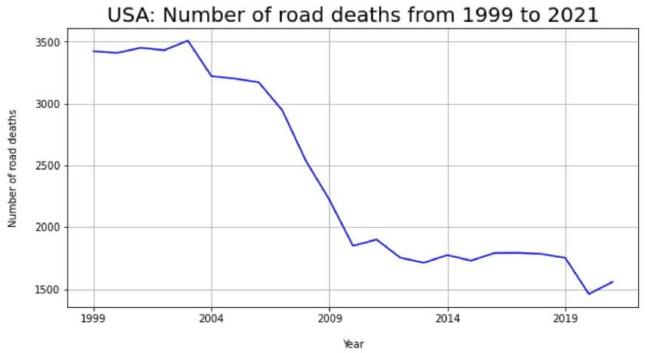
```
skewValue = df_rate.skew(axis=0)
print(skewValue)
kurt = df_rate.kurt(axis=0 )
print(kurt)
```

```
DeathsRate 2021    1.535462
dtype: float64
DeathsRate 2021    3.563912
dtype: float64
```

```
df_Great_Britain = df.iloc[[0]].set_index("Country")
df_Great_Britain = df_Great_Britain.drop(columns=["DeathsRate 2021"])
df_Great_Britain.T.plot(kind='line',figsize=(10, 5), legend=False, color='blue', rot=0);
plt.title("Great Britain: Number of road deaths from 1999 to 2021", fontsize=20)
plt.xlabel("Year", labelpad=15)
plt.ylabel("Number of road deaths", labelpad=15)
plt.grid()
plt.show()
```




```
df_USA = df.iloc[[0]]
df_USA = df_USA.set_index("Country").drop(columns=["DeathsRate 2021"])
df_USA.T.plot(kind='line',figsize=(10, 5), legend=False, color='blue', rot=0);
plt.title("USA: Number of road deaths from 1999 to 2021", fontsize=20)
plt.xlabel("Year", labelpad=15)
plt.ylabel("Number of road deaths", labelpad=15)
plt.grid()
plt.show()
```



Data Set 2:

```
import matplotlib.pyplot as plt
import pandas as pd

df_stats = pd.read_excel('/content/gdrive/MyDrive/Colab Notebooks/Report/Final/Population_Vehicle_trafiic_Road_deaths_Great Britain.xlsx')
```

Saving...

	Great Britain	1999	2000	2001	2002	2003	2004	2005	2006	2007
0	vehicle traffic	2.901550e+05	2.895510e+05	2.934590e+05	3.002370e+05	3.020410e+05	3.057060e+05	3.058440e+05	3.103730e+05	3.132940e+05
1	population	5.863520e+07	5.885004e+07	5.909202e+07	5.935569e+07	5.964980e+07	5.999585e+07	6.038374e+07	6.080370e+07	6.126068e+07
2	road deaths	3.423000e+03	3.409000e+03	3.450000e+03	3.431000e+03	3.508000e+03	3.221000e+03	3.201000e+03	3.172000e+03	2.946000e+03
3	death rate	5.837790e+01	5.792689e+01	5.838352e+01	5.780406e+01	5.880992e+01	5.368705e+01	5.301096e+01	5.216788e+01	4.808958e+01

4 rows × 24 columns

```
df_stats.shape
```

```
(4, 24)
```

```
df_stats.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 24 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Great Britain  4 non-null      object
 1   1999           4 non-null      float64
 2   2000           4 non-null      float64
 3   2001           4 non-null      float64
 4   2002           4 non-null      float64
 5   2003           4 non-null      float64
 6   2004           4 non-null      float64
```

```
 7   2005          4 non-null      float64
 8   2006          4 non-null      float64
 9   2007          4 non-null      float64
 10  2008          4 non-null      float64
 11  2009          4 non-null      float64
 12  2010          4 non-null      float64
 13  2011          4 non-null      float64
 14  2012          4 non-null      float64
 15  2013          4 non-null      float64
 16  2014          4 non-null      float64
 17  2015          4 non-null      float64
 18  2016          4 non-null      float64
 19  2017          4 non-null      float64
 20  2018          4 non-null      float64
 21  2019          4 non-null      float64
 22  2020          4 non-null      float64
 23  2021          4 non-null      float64
dtypes: float64(23), object(1)
memory usage: 896.0+ bytes
```

```
df_stats.isna().any()
```

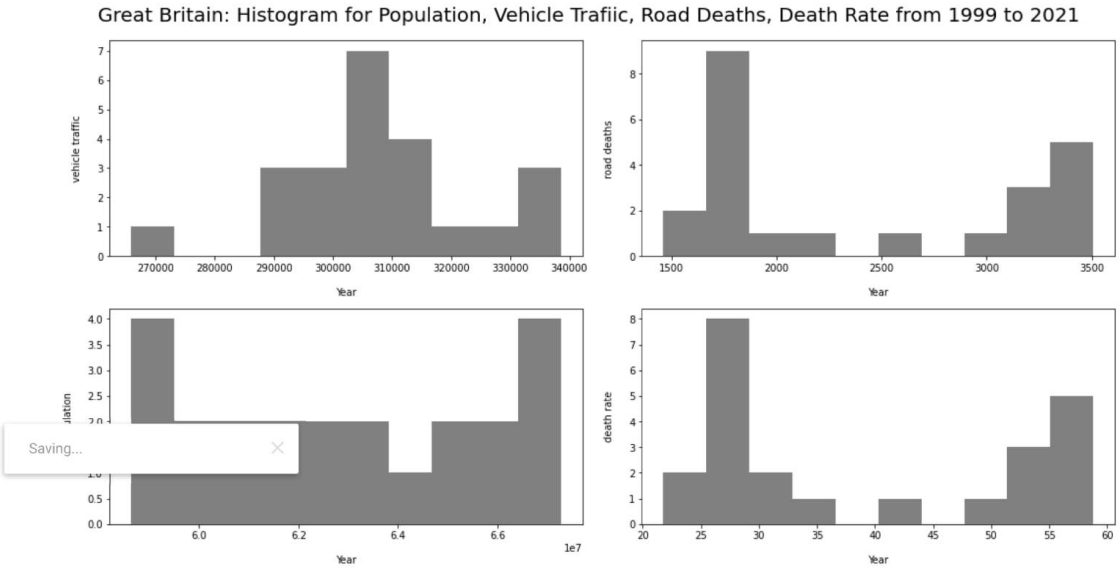
```
Great Britain    False
1999             False
2000             False
2001             False
2002             False
2003             False
2004             False
2005             False
2006             False
2007             False
2008             False
2009             False
2010             False
2011             False
2012             False
2013             False
2014             False
2015             False
2016             False
2017             False
2018             False
2019             False
2020             False
2021             False
dtype: bool
```

Saving...

```
df_stats_T = df_stats.set_index("Great Britain").T
ncols = 4
nrows = 1

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
for col, ax in zip(df_stats_T.columns, axs.T.ravel()):
    df_stats_T[[col]].plot(ax=ax, kind='line',figsize=(10, 5), legend=False, color='blue', rot=0);
    ax.set_xlabel('Year',labelpad=15)
    ax.set_ylabel(col,labelpad=15)

fig.suptitle('Great Britain: Population, Vehicle Trafiic, Road Deaths, Death Rate from 1999 to 2021', fontsize=20)
fig.set_size_inches(20, 5)
```

Great Britain: Population, Vehicle Trafiic, Road Deaths, Death Rate from 1999 to 2021

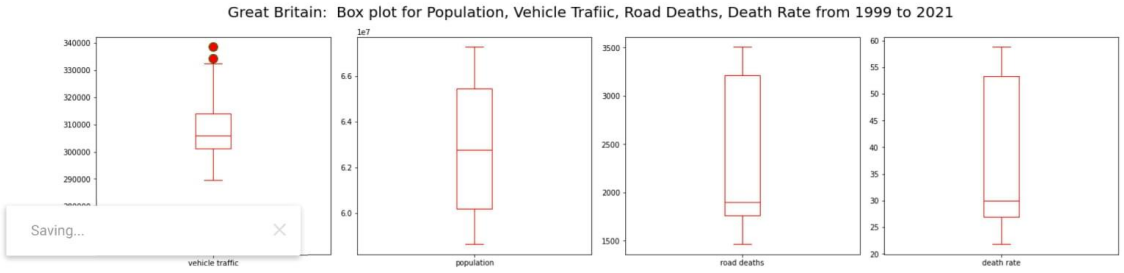
```
ncols = 2
nrows = 2

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
for col, ax in zip(df_stats_T.columns, axs.T.ravel()):
   df_stats_T[[col]].plot(ax=ax, kind='hist',figsize=(10, 5), legend=False, color='gray', rot=0);
   ax.set_xlabel('Year',labelpad=15)
   ax.set_ylabel(col,labelpad=15)

fig.suptitle('Great Britain: Histogram for Population, Vehicle Trafiic, Road Deaths, Death Rate from 1999 to 2021', fontsize=20)
fig.set_size_inches(15, 8)
```




```
df_stats_T
```

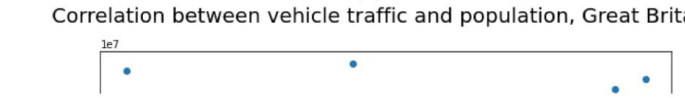
Great Britain	vehicle traffic	population	road deaths	death rate
1999	290155.0	58635202.0	3423.0	58.377901
2000	289551.0	58850043.0	3409.0	57.926891
2001	293459.0	59092016.0	3450.0	58.383522
2002	300237.0	59355690.0	3431.0	57.804062
2003	302041.0	59649799.0	3508.0	58.809922
2004	305706.0	59995851.0	3221.0	53.687046
2005	305844.0	60383741.0	3201.0	53.010959
2006	310373.0	60803700.0	3172.0	52.167878

```
df_stats_T = df_stats.set_index("Great Britain").T
ncols = 4
nrows = 1

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
flierprops = dict(marker='o', markerfacecolor='r', markersize=12,
                  linestyle='none', markeredgecolor='g')
for col, ax in zip(df_stats_T.columns, axs.T.ravel()):
    df_stats_T[[col]].plot(ax=ax, kind='box',figsize=(5, 5), flierprops=flierprops, color='red');

fig.suptitle('Great Britain:  Box plot for Population, Vehicle Trafiic, Road Deaths, Death Rate from 1999 to 2021', fontsize=20)
fig.set_size_inches(20, 5)
```



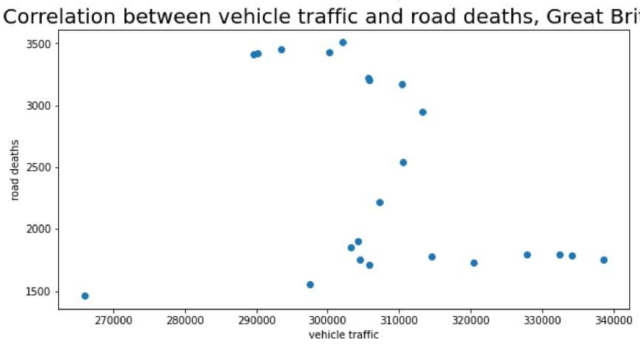

```
df_stats_T = df_stats.set_index('Great Britain').T
column_0 = df_stats_T['vehicle traffic']
column_1 = df_stats_T['population']

correlation_1 = column_0.corr(column_1)
print('Correlation between vehicle traffic and population, Great Britain :',correlation_1)
column_2 = df_stats_T['road deaths']
plt.scatter(column_0, column_1)
plt.ylabel('population', fontsize=10)
plt.xlabel('vehicle traffic', fontsize=10)
plt.title('Correlation between vehicle traffic and population, Great Britain \n', fontsize=20)
plt.show()
```

```
Correlation between vehicle traffic and population, Great Britain : 0.38813085174390255
```




```
column_2 = df_stats_T['road deaths']

correlation_2 = column_0.corr(column_2)
print('Correlation between vehicle traffic and road deaths, Great Britain :',correlation_2)
plt.scatter(column_0, column_2)
plt.ylabel('road deaths', fontsize=10)
plt.xlabel('vehicle traffic', fontsize=10)
plt.title('Correlation between vehicle traffic and road deaths, Great Britain ', fontsize=20)
plt.show()
```

```
Correlation between vehicle traffic and road deaths, Great Britain : -0.30455743603997415
```

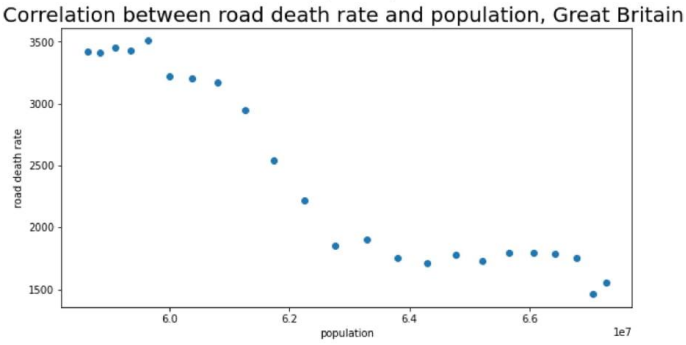



```
correlation_3 = column_1.corr(column_2)
print('Correlation between road death rate and population, Great Britain :',correlation_3)
```



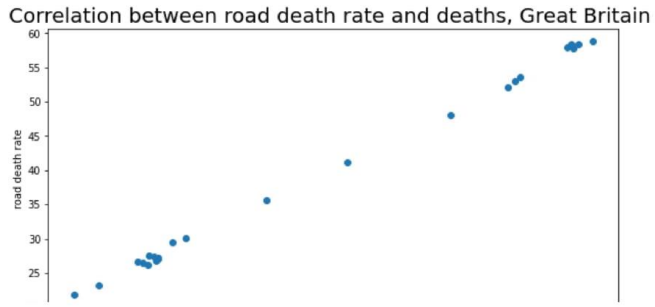
```
plt.ylabel('road death rate', fontsize=10)
plt.xlabel('population', fontsize=10)
plt.title('Correlation between road death rate and population, Great Britain ', fontsize=20)
plt.show()
```

```
Correlation between road death rate and population, Great Britain : -0.9414794184721537
```




```
correlation_4 = column_3.corr(column_2)
print('Correlation between road deaths and death rate, Great Britain :',correlation_4)
plt.scatter(column_2, column_3)
plt.ylabel('road death rate', fontsize=10)
plt.xlabel('road deaths', fontsize=10)
plt.title('Correlation between road death rate and deaths, Great Britain ', fontsize=20)
plt.show()
```

```
Correlation between road deaths and death rate, Great Britain : 0.9993977451204253
```




```
correlation = df_stats_T.corr(method='pearson')
correlation
```

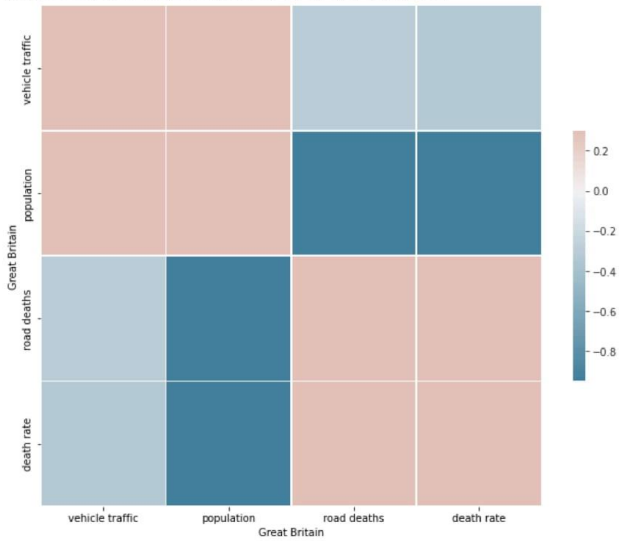
Great Britain	vehicle traffic	population	road deaths	death rate
Great Britain				
vehicle traffic	1.000000	0.388131	-0.304557	-0.325512
population	0.388131	1.000000	-0.941479	-0.949827
road deaths	-0.304557	-0.941479	1.000000	0.999398
death rate	-0.325512	-0.949827	0.999398	1.000000

```
#Set up matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

#Set up a seaborn heatmap
#seaborne aesthetics https://seaborn.pydata.org/tutorial/aesthetics.html
sns.set_style("white")
```

Saving...

```
ap
, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(correlation, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f45a957bf10>
```



```
df_stats_T.describe()
```

Great Britain	vehicle traffic	population	road deaths	death rate
count	23.000000	2.300000e+01	23.000000	23.000000
mean	307731.869565	6.284520e+07	2407.956522	38.916778
std	16190.374956	2.933112e+06	772.454675	14.175958
min	265894.000000	5.863520e+07	1460.000000	21.771719
25%	301139.000000	6.018980e+07	1764.500000	26.997072
50%	305759.000000	6.276004e+07	1901.000000	30.038067
75%	313892.000000	6.543978e+07	3211.000000	53.349002
max	338596.000000	6.728104e+07	3508.000000	58.809922

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import datasets, linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

#Use only 1 feature - AGE to predict the target Y
deaths_x = df_stats_T[['road deaths']]
death_rate_y = df_stats_T[['death rate']]

#The scatterplot evidence that seemingly there is a trend
plt.scatter(deaths_x, death_rate_y)

#Split the dataset into training and testing sets (80%:20%)
x_train,x_test,y_train,y_test=train_test_split(deaths_x, death_rate_y,test_size=0.2)
print(x_train)
print("\n")
print(y_train)
```

Saving...

```
#Create linear regression object
regr = LinearRegression()

# Train the model using the training sets and reshape 1D arrays
regr.fit(x_train.to_numpy(), y_train.to_numpy())

# Make predictions using the testing set
y_pred = regr.predict(x_test.to_numpy())
y_pred2 = regr.predict([[1600]])
print("\n")
print("The predicted Y value for deaths = 1600 is: ", y_pred2)

# The coefficients
print('Coefficients: \n', regr.coef_)
print("\n")

# The mean squared error
print('Mean squared error: ', mean_squared_error(y_test, y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: ', r2_score(y_test, y_pred))

 # Plot outputs
plt.scatter(x_test, y_test,  color='black')
plt.plot(x_test, y_pred, color='blue', linewidth=3)
plt.ylabel('death rate', fontsize=10)
plt.xlabel('deaths', fontsize=10)
plt.title('Liner regression model for  road death rate and deaths, Great Britain ', fontsize=20)

plt.xticks(())
plt.yticks(())

plt.show()
```

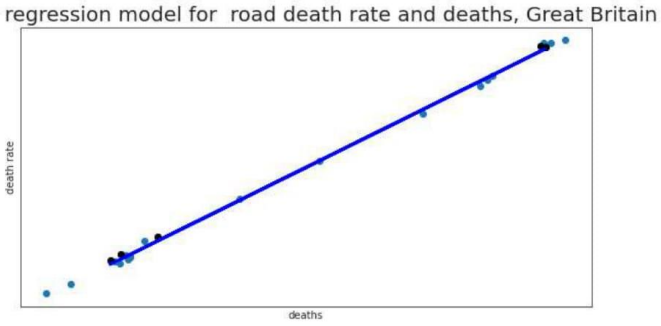
```
Great Britain  road deaths
2008                2538.0
2020                1460.0
2021                1558.0
2015                1730.0
2009                2222.0
2010                1850.0
2005                3201.0
2017                1793.0
2001                3450.0
2007                2946.0
2003                3508.0
2004                3221.0
2014                1775.0
2006                3172.0
2019                1752.0
2016                1792.0
2018                1784.0
1999                3423.0
```

```
Great Britain  death rate
2008            41.106440
2020            21.771719
2021            23.156598
2015            26.523831
2009            35.698577
2010            29.477356
2005            53.010959
2017            27.140018
2001            58.383522
2007            48.089577
2003            58.809922
2004            53.687046
2014            27.403180
2006            52.167878
2019            26.235927
2016            27.294105
2018            26.854127
1999            58.377901
```


```
2012                1754.0
2002                3431.0
2011                1901.0
```

```
The predicted Y value for deaths = 1600 is:  [[23.96557429]]
Coefficients:
 [[0.01835188]]
```

```
Mean squared error:  0.3568743835315488
Coefficient of determination:  0.9983364534181608
```




```
import statsmodels.api as sm

#Use more than 1 feature - AGE to predict the target Y
_x = df_stats_T[['vehicle traffic', 'population',   'road deaths']]
_y = df_stats_T[['death rate']]
```

```
#Split the dataset into training and testing sets (80%:20%)
x_train,x_test,y_train,y_test=train_test_split(_x, _y,test_size=0.2)
print(x_train)
print("\n")
print(y_train)
print("\n")
print(x_test)
print("\n")

#Create linear regression object
regr = LinearRegression()

# Train the model using the training sets
regr.fit(x_train.to_numpy(), y_train.to_numpy())

# Make predictions using the testing set
y_pred = regr.predict(x_test.to_numpy())
print("Y test: ")
print(y_test)
print("\n")
print("Y Predicted: ")
print(y_pred)
#Make prediction for the following:
VT = 340000.0
PO = 70000000.0
RD = 1600.0
print("\n")
print ('Predicted Y value: ', regr.predict([[VT, PO, RD]]))
print("\n")

# The coefficients
print('Coefficients: \n', regr.coef_)
print("\n")

# The mean squared error
print('Mean squared error: ', mean_squared_error(y_test, y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: ', r2_score(y_test, y_pred))
```

Saving...

```
#use statsmodel for model summary
_x = sm.add_constant(_x)
model = sm.OLS(_y, _x).fit()
predictions = model.predict(_x)
print_model = model.summary()
print(print_model)
```

Great Britain	vehicle traffic	population	road deaths
2020	265894.0	67059474.0	1460.0
2005	305844.0	60383741.0	3201.0
2000	289551.0	58850043.0	3409.0
2019	338596.0	66778659.0	1752.0
2008	310530.0	61742151.0	2538.0
2001	293459.0	59092016.0	3450.0
2012	304559.0	63808727.0	1754.0
1999	290155.0	58635202.0	3423.0
2007	313294.0	61260676.0	2946.0
2011	304287.0	63286362.0	1901.0
2014	314490.0	64773504.0	1775.0
2013	305759.0	64302297.0	1713.0
2002	300237.0	59355690.0	3431.0
2006	310373.0	60803700.0	3172.0
2004	305706.0	59995851.0	3221.0
2016	327926.0	65655203.0	1792.0
2003	302041.0	59649799.0	3508.0
2009	307270.0	62243378.0	2222.0

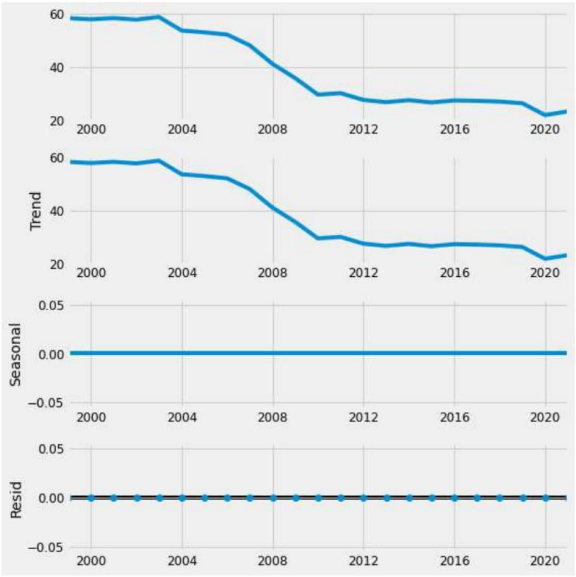
Great Britain	death rate
2020	21.771719
2005	53.010959
2000	57.926891
2019	26.235927
2008	41.106440
2001	58.383522
2012	27.488403
1999	58.377901
2007	48.089577

```
2011            30.038067
2014            27.403180
2013            26.639795
2002            57.804062
2006            52.167878
2004            53.687046
2016            27.294105
2003            58.809922
2009            35.698577


Great Britain  vehicle traffic  population  road deaths
2017                 332504.0  66064804.0       1793.0
2021                 297559.0  67281039.0       1558.0
2010                 303198.0  62760039.0       1850.0
2015                 320388.0  65224364.0       1730.0
2018                 334213.0  66432993.0       1784.0


Y test:
Great Britain  death rate
2017            27.140018
2021            23.156598
2010            29.477356
```

```
df_stats_T = df_stats.set_index("Great Britain").T
df_stats_T.index = pd.DatetimeIndex(df_stats_T.index)
df_stats_T = df_stats_T.drop(columns=['vehicle traffic','population','road deaths'])
df_stats_T
```

Great Britain	death rate
1999-01-01	58.377901
2000-01-01	57.926891
2001-01-01	58.383522
2002-01-01	57.804062
2003-01-01	58.809922
2005-01-01	53.010959
2006-01-01	52.167878
2007-01-01	48.089577
2008-01-01	41.106440
2009-01-01	35.698577
2010-01-01	29.477356
2011-01-01	30.038067
2012-01-01	27.488403
2013-01-01	26.639795
2014-01-01	27.403180
2015-01-01	26.523831
2016-01-01	27.294105
2017-01-01	27.140018
2018-01-01	26.854127
2019-01-01	26.235927
2020-01-01	21.771719
2021-01-01	23.156598


```
from pylab import rcParams
import statsmodels.api as sm
rcParams['figure.figsize'] = 8, 8
decomposition = sm.tsa.seasonal_decompose(df_stats_T, model='additive')
fig = decomposition.plot()
plt.show()
```




```
import itertools
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d, q))]
print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```

Saving...

```
Examples of parameter combinations for Seasonal ARIMA...
SARIMAX: (0, 0, 1) x (0, 0, 1, 12)
SARIMAX: (0, 0, 1) x (0, 1, 0, 12)
SARIMAX: (0, 1, 0) x (0, 1, 1, 12)
SARIMAX: (0, 1, 0) x (1, 0, 0, 12)
```

```
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
import pandas as pd
import statsmodels.api as sm
import matplotlib
matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 12
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(df_stats_T,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)
            results = mod.fit()
            print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
        except:
            continue
mod = sm.tsa.statespace.SARIMAX(df_stats_T,
                                order=(1, 1, 1),
                                seasonal_order=(1, 1, 0, 12),
                                enforce_invertibility=False)
results = mod.fit()
```

```
print(results.summary().tables[1])
```

```
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:227.14466545467698
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:97.61695454236362
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:95.76908712079823
ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:4.0
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:53.04156260702363
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:51.3907117310967
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:4.0
ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:6.0
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:193.72269424527178
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:80.99348715362946
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:77.1377673983206
ARIMA(0, 0, 1)x(0, 1, 1, 12)12 - AIC:6.0
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:53.36866198971255
ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:49.65734448701473
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:6.0
ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:8.0
ARIMA(0, 1, 0)x(0, 0, 0, 12)12 - AIC:108.29979988959201
ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:38.73725348574501
ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:49.80716553419713
ARIMA(0, 1, 0)x(0, 1, 1, 12)12 - AIC:4.0
ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:42.524292032598524
ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:39.367217019795405
ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:4.0
ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:6.0
ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:103.05623935219924
ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:37.23396912448007
ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:47.162186185597314
ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:6.0
ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:40.4139382190768
ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:32.73701028412476
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:6.0
ARIMA(0, 1, 1)x(1, 1, 1, 12)12 - AIC:8.0
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:106.12298146887144
ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:43.93658150202694
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:54.601953415272604
ARIMA(1, 0, 0)x(0, 1, 1, 12)12 - AIC:6.0
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:43.708387342403036
ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:45.70838733711476
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:6.0
ARIMA(1, 0, 0)x(1, 1, 1, 12)12 - AIC:8.0
                                 - AIC:103.15340231905468
                                 - AIC:41.20945449740106
                                 - AIC:50.97995824758725
ARIMA(1, 0, 1)x(0, 1, 1, 12)12 - AIC:8.0
ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:41.790824142905606
ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:37.30510658114885
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:8.0
ARIMA(1, 0, 1)x(1, 1, 1, 12)12 - AIC:10.0
ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:104.75812118143705
ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:40.61546934090201
ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:51.35465237976139
ARIMA(1, 1, 0)x(0, 1, 1, 12)12 - AIC:6.0
ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:37.2811380376359
ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:39.28113803660446
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:6.0
ARIMA(1, 1, 0)x(1, 1, 1, 12)12 - AIC:8.0
ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:100.99389476058703
ARIMA(1, 1, 1)x(0, 0, 1, 12)12 - AIC:39.1709703829743
```

Saving...

Data Set 3:

```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_excel('/content/gdrive/MyDrive/Colab Notebooks/Report/Final/Road-Casualties_Great Britain_Reigon.xlsx')
data.head()
data.tail()
```

	Accident year	Road user	Casualty sex	Urban rural	Built up roads	All casualties
2682	2021	Other vehicle	Female	Urban	Built up road	384
2683	2021	Other vehicle	Female	Urban	Non built up road	2
2684	2021	Other vehicle	Female	Rural	Motorway	4

```
data.isna().any()
```

```
Accident year     False
Road user         False
Casualty sex      False
Urban rural       False
Built up roads    False
All casualties    False
dtype: bool
```

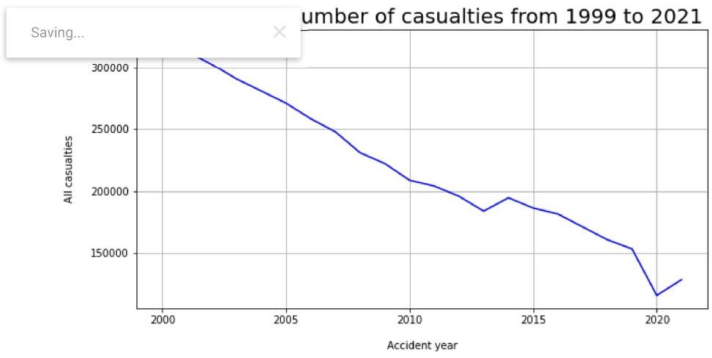
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2687 entries, 0 to 2686
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Accident year   2687 non-null   int64 
 1   Road user       2687 non-null   object
 2   Casualty sex    2687 non-null   object
 3   Urban rural     2687 non-null   object
 4   Built up roads  2687 non-null   object
 5   All casualties  2687 non-null   int64 
dtypes: int64(2), object(4)
memory usage: 126.1+ KB
```

```
data.shape
```

```
(2687, 6)
```

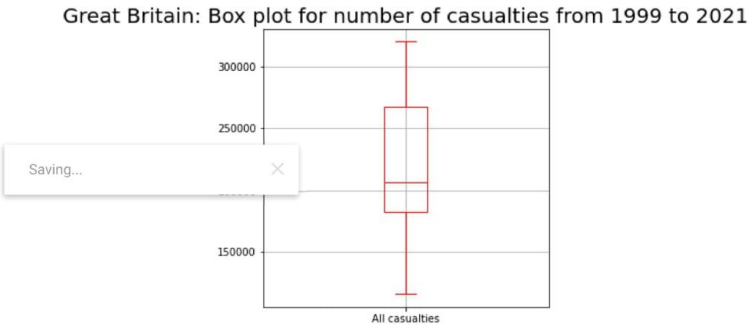
```
data_AllCausalities_AccidentYear = data.groupby(['Accident year']).sum()
data_AllCausalities_AccidentYear.plot(kind='line',figsize=(10, 5), legend=False, color='blue', rot=0);
plt.title("Great Britain: Number of casualties from 1999 to 2021", fontsize=20)
plt.xlabel("Accident year", labelpad=15)
plt.ylabel("All casualties", labelpad=15)
plt.grid()
plt.show()
```




```
data_AllCausalities_AccidentYear
```

Accident year	All casualties
2000	320283
2001	313309
2002	302605
2003	290607
2004	280840
2005	271017
2006	258404
2007	247780
2008	230905
2009	222146
2010	208648
2011	203950

```
flierprops = dict(marker='o', markerfacecolor='r', markersize=12,
                  linestyle='none', markeredgecolor='g')
data_AllCausalities_AccidentYear.plot(kind='box',figsize=(5, 5), flierprops=flierprops, color='red');
plt.title("Great Britain: Box plot for number of casualties from 1999 to 2021", fontsize=20)
plt.grid()
plt.show()
```




```
from scipy import stats
stats.ttest_1samp(data_AllCausalities_AccidentYear['All casualties'], 0)
```

```
Ttest_1sampResult(statistic=17.250498224981282, pvalue=7.057630401562082e-14)
```

```
data_AllCausalities_AccidentYear['All casualties'].describe()
```

```
count        22.000000
mean     219112.636364
std       59576.793051
min      115584.000000
25%      181955.500000
50%      206299.000000
75%      267863.750000
max      320283.000000
Name: All casualties, dtype: float64
```

```
from scipy import stats
stats.ttest_1samp(data_AllCausalities_AccidentYear['All casualties'], 220000)
```

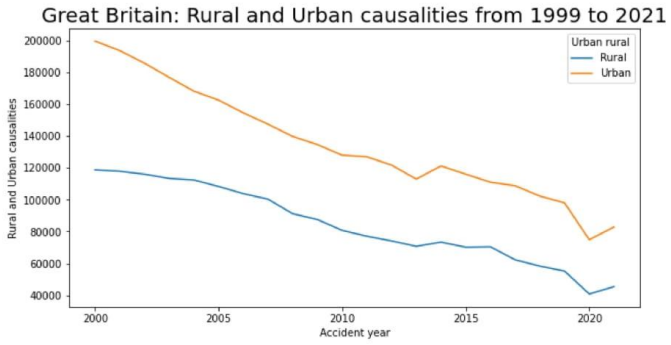
```
Ttest_1sampResult(statistic=-0.06986116861192783, pvalue=0.9449651553287055)
```

```
data_AllCausalities_AccidentYear
```

Accident year	All casualties
2000	320283
2001	313309
2002	302605
2003	290607
2004	280840
2005	271017
2006	258404
2007	247780
2008	230905
2009	222146
2010	208648
2011	203950
2012	195723
2013	183670
2014	194477
2015	186189
2016	181384
2017	170993
2018	160597
2019	153158
2020	115584

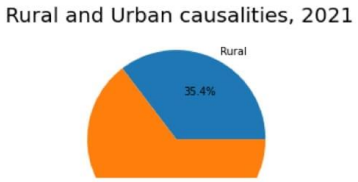
Saving...

```
ta.groupby(['Accident year','Urban rural']).sum().reset_index()
data_AllCausalities_Urban_Rural.pivot(index='Accident year', columns='Urban rural', values='All casualtie
data_AllCausalities_Urban_Rural_T =data_AllCausalities_Urban_Rural_T.drop(data_AllCausalities_Urban_Rural_T.index[[1,2]]).T.reset_index()
data_AllCausalities_Urban_Rural_T.plot(x="Accident year", y=["Rural", "Urban"], figsize=(10, 5))
plt.ylabel('Rural and Urban causalities', fontsize=10)
plt.xlabel('Accident year', fontsize=10)
plt.title('Great Britain: Rural and Urban causalities from 1999 to 2021', fontsize=20)
plt.show()
```




```
labels = ["Rural", "Urban"]
values = data_AllCausalities_Urban_Rural_T.iloc[[21]].drop(columns=['Accident year']).values.ravel()
plt.pie(values, labels = labels, autopct='%1.1f%%')
plt.title("\n Rural and Urban causalities, 2021", fontsize=20)
```

```
Text(0.5, 1.0, '\n Rural and Urban causalities, 2021')
```

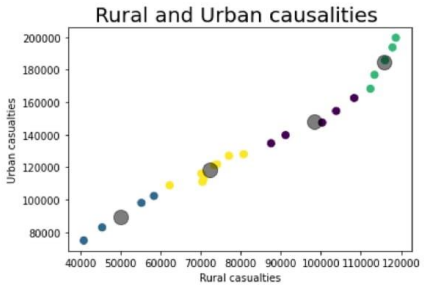



```
from sklearn import metrics
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
#Select columns for X
X = data_AllCausalities_Urban_Rural_T[["Rural","Urban"]]
#Perform Clustering
kmeans = KMeans(n_clusters=4, random_state=0).fit(X)
cluster_class = kmeans.predict(X)
print(cluster_class)
print("\n")
#Print off the labels of kmeans model
print(kmeans.cluster_centers_)
print("\n")
#Plot the graphs
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
plt.scatter(X["Rural"], X["Urban"], c=y_kmeans, s=50, cmap='viridis')
plt.ylabel('Urban casualties', fontsize=10)
plt.xlabel('Rural casualties', fontsize=10)
plt.title('Rural and Urban causalities ', fontsize=20)
centers = kmeans.cluster_centers_
print("Center for cluster 0 is ", centers[0])
print("Center for cluster 1 is ", centers[1])
print("Center for cluster 2 is ", centers[2])
print("Center for cluster 3 is ", centers[3])
print("\n")
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```

Saving...

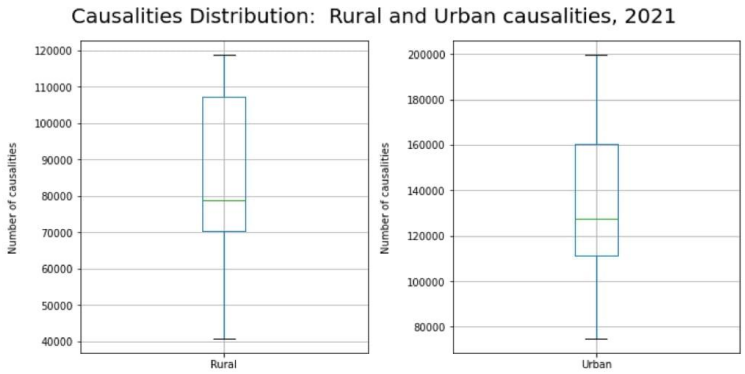
```
3 3 3 1 1 1 1]
```

```
[[ 98227.4   147784.8  ]
 [ 49869.5    89477.75 ]
 [115673.8   184833.2  ]
 [ 72311.75  118314.625]]


Center for cluster 0 is  [ 98227.4 147784.8]
Center for cluster 1 is  [49869.5  89477.75]
Center for cluster 2 is  [115673.8 184833.2]
Center for cluster 3 is  [ 72311.75  118314.625]
```




```
ncols = 2
nrows = 1

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
for col, ax in zip(data_AllCausalities_Urban_Rural_T.drop(columns=["Accident year"]).columns, axs.T.ravel()):
    flierprops = dict(marker='o', markerfacecolor='r', markersize=12,
```

```
                linestyle='none', markeredgecolor='g')
    data_AllCausalities_Urban_Rural_T[[col]].boxplot(ax=ax, flierprops=flierprops);
    ax.set_ylabel('Number of causalities',labelpad=15)

fig.suptitle('Causalities Distribution:  Rural and Urban causalities, 2021', fontsize=20)
fig.set_size_inches(10, 5)
```




```
skewValue = data_AllCausalities_Urban_Rural_T.set_index("Accident year").skew(axis=0)
print(skewValue)
kurt = data_AllCausalities_Urban_Rural_T.set_index("Accident year").kurt(axis=0 )
print(kurt)
```

```
Urban rural
Rural   -0.038665
Urban    0.306767
dtype: float64
Urban rural
Rural   -1.161705
Urban   -0.702687
```

Saving...

```
correlation = data_AllCausalities_Urban_Rural_T.set_index('Accident year').corr(method='pearson')
correlation
```

Urban rural	Rural	Urban
Urban rural		
Rural	1.00000	0.98384
Urban	0.98384	1.00000

```
data_AllCausalities_Urban_Rural_T.drop(columns=["Accident year"]).describe()
```

Urban rural	Rural	Urban
count	22.000000	22.000000
mean	83976.272727	134887.181818
std	24420.172417	35018.771933
min	40745.000000	74822.000000
25%	70211.500000	111487.000000
50%	78847.500000	127451.500000
75%	107181.500000	160576.750000
max	118725.000000	199659.000000

```
data_AllCausalities_Urban_Rural_T
```

Urban rural	Accident year	Rural	Urban
0	2000	118725.0	199659.0
1	2001	117875.0	193677.0
2	2002	116007.0	185779.0
3	2003	113377.0	176835.0
4	2004	112385.0	168216.0
5	2005	108309.0	162578.0
6	2006	103799.0	154573.0
7	2007	100320.0	147444.0
8	2008	91176.0	139716.0
9	2009	87533.0	134613.0
10	2010	80692.0	127956.0
11	2011	77003.0	126947.0
12	2012	74043.0	121680.0
13	2013	70707.0	112963.0
14	2014	73298.0	121179.0
15	2015	70156.0	116033.0
16	2016	70378.0	110995.0
17	2017	62217.0	108764.0
18	2018	58261.0	102256.0
19	2019	55135.0	97981.0

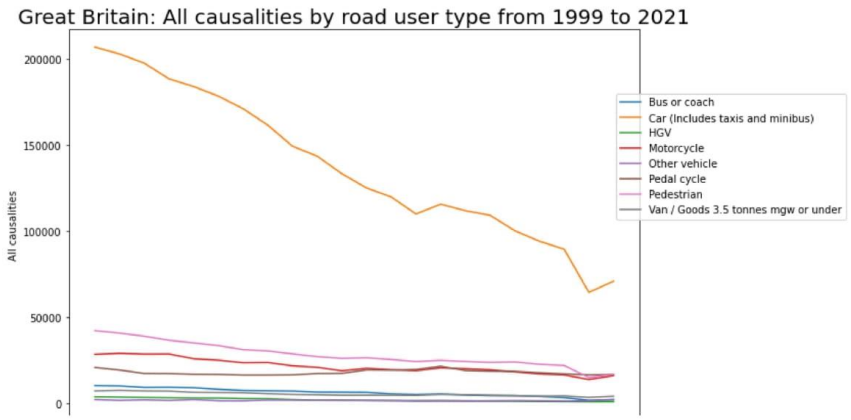
```
t2, p = stats.ttest_ind(data_AllCausalities_Urban_Rural_T["Urban"], data_AllCausalities_Urban_Rural_T["Rural"])
print("p value = {:g}".format(p))
print("t value = {:g}". format(t2))
```

Saving...

```
pearsonCoeff_rvalue, p_value = stats.pearsonr(data_AllCausalities_Urban_Rural_T["Rural"], data_AllCausalities_Urban_Rural_T["Urban"])
#define the columns to perform calculations on
print("Pearson Correlation Coefficient r value : ", pearsonCoeff_rvalue.round(decimals=3), "and a P-value of:", p_value.round(decimals =3)) #
print("\n")

#Conduct Correlation Coefficient Hypothesis Testing
#Use 2 tail test
#Confidence level is 95%, alpha is 0.05 and alpha/2 is 0.025
alpha = 0.05
alpha_half = 0.025

if p_value < alpha_half:  # null hypothesis: x comes from a normal distribution
    print("Conclusion drawn: The null hypothesis can be rejected")
else:
    print("Conclusion drawn: The null hypothesis is accepted")
```

```
 Pearson Correlation Coefficient r value :  0.984 and a P-value of: 0.0


 Conclusion drawn: The null hypothesis can be rejected
```

```
data_AllCausalities_Road_User = data.groupby(['Accident year','Road user']).sum().reset_index()
data_AllCausalities_Road_User_T = data_AllCausalities_Road_User.pivot(index='Accident year', columns='Road user', values='All casualties').T
data_AllCausalities_Road_User_T =data_AllCausalities_Road_User_T.T.reset_index()
data_AllCausalities_Road_User_T.plot(x="Accident year", y=["Bus or coach","Car (Includes taxis and minibus)","HGV","Motorcycle","Other vehicl
plt.title('Great Britain: All causalities by road user type from 1999 to 2021', fontsize=20)
plt.ylabel('All causalities', fontsize=10)
plt.xlabel('Accident year', fontsize=10)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.show()
```

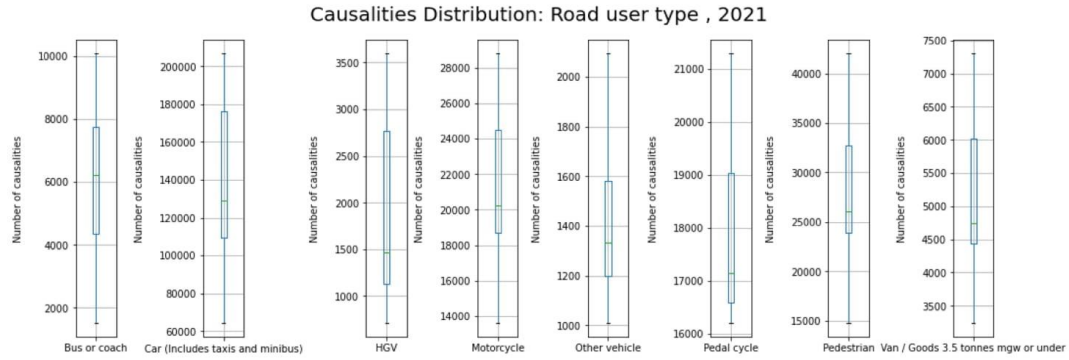


data_AllCausalities_Road_User_T

	Road user	Accident year	Bus or coach	Car (Includes taxis and minibus)	HGV	Motorcycle	Other vehicle	Pedal cycle	Pedestrian	Van / Goods 3.5 tonnes mgw or under
0		2000	10088	206799	3597	28212	1935	20612	42033	7007
1		2001	9884	202802	3388	28810	1430	19114	40577	7304
2		2002	9005	197425	3178	28353	1746	17107	38784	7007
3		2003	9068	188342	3061	28411	1390	17033	36405	6897
4		2004	8820	183858	2883	25641	1943	16648	34881	6166
5		2005	7920	178302	2843	24824	1238	16561	33281	6048
6		2006	7253	171000	2530	23326	1203	16196	30982	5914
				161433	2476	23459	1607	16195	30191	5340
				149188	1930	21550	1616	16297	28482	4913
9		2009	6317	143412	1519	20703	1501	17064	26887	4743
10		2010	6268	133205	1578	18686	1387	17185	25845	4494
11		2011	6177	124924	1415	20150	1372	19215	26198	4499
12		2012	5234	119708	1339	19310	1290	19091	25218	4533
13		2013	4873	109787	1296	18752	1065	19438	24033	4426
14		2014	5198	115530	1353	20366	1080	21287	24748	4915
15		2015	4626	111707	1203	19918	1080	18844	24061	4750
16		2016	4246	109046	1105	19297	1199	18477	23550	4464
17		2017	4236	100082	1038	18042	1295	18321	23805	4174
18		2018	3801	93979	880	16818	1192	17550	22432	3945
19		2019	3085	89331	786	16224	1009	16884	21770	4069
20		2020	1506	64255	710	13604	1230	16294	14750	3235
21		2021	1762	70755	735	15838	2094	16458	16654	3913

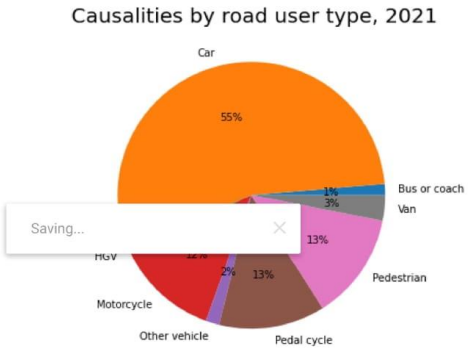
Saving...

```
ncols = 8
nrows = 1

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
for col, ax in zip(data_AllCausalities_Road_User_T.drop(columns=["Accident year"]).columns, axs.T.ravel()):
    flierprops = dict(marker='o', markerfacecolor='r', markersize=12,
                  linestyle='none', markeredgecolor='g')
    data_AllCausalities_Road_User_T[[col]].boxplot(ax=ax, flierprops=flierprops);
    ax.set_ylabel('Number of causalities',labelpad=15)

fig.suptitle('Causalities Distribution: Road user type , 2021', fontsize=20)
fig.set_size_inches(15, 5)
```




```
labels = ["Bus or coach", "Car", "HGV", "Motorcycle", "Other vehicle", "Pedal cycle", "Pedestrian", "Van"]
plt.figure(figsize = (6, 6))
values = data_AllCausalities_Road_User_T.iloc[[21]].drop(columns=['Accident year']).values.ravel()
plt.pie(values, labels = labels, autopct='%1.0f%%')
plt.title("\n Causalities by road user type, 2021", fontsize=20)
```

```
Text(0.5, 1.0, '\n Causalities by road user type, 2021')
```




```
import scipy.stats as stats
stats.f_oneway(data_AllCausalities_Road_User_T["Bus or coach"],data_AllCausalities_Road_User_T["Car (Includes taxis and minibus)"],data_AllCa
```

```
F_onewayResult(statistic=185.66680623359917, pvalue=1.2894104732119707e-75)
```

```
import scipy.stats as stats
fvalue, pvalue  = stats.f_oneway(data_AllCausalities_Road_User_T["Bus or coach"],data_AllCausalities_Road_User_T["Car (Includes taxis and min
print("F Value  = {:g} ".format(fvalue))
print("P Value  = {:g} ".format(pvalue))

alpha = 0.05

if pvalue < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected")
else:
    print("The null hypothesis is accepted")
```

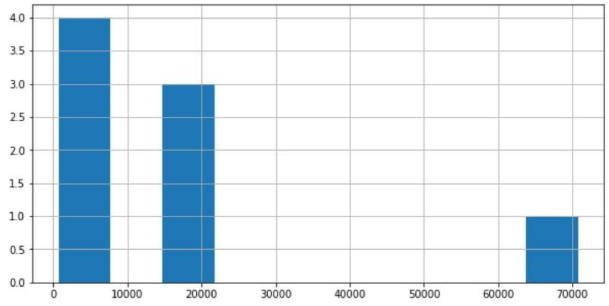
```
F Value  = 185.667
P Value  = 1.28941e-75
The null hypothesis can be rejected
```

```
data_AllCausalities_Road_User_T.set_index('Accident year').describe()
```

Road user	Bus or coach	Car (Includes taxis and minibus)	HGV	Motorcycle	Other vehicle	Pedal cycle	Pedestrian	Van / Goods 3.5 tonnes mgw or under
count	22.000000	22.000000	22.000000	22.00000	22.000000	22.000000	22.000000	22.000000
mean	6062.500000	137494.090909	1856.500000	21377.00000	1404.636364	17812.318182	27980.318182	5125.272727
std	2465.565979	43411.652729	951.116074	4449.38183	305.057457	1499.055524	7242.094306	1164.368301
min	1506.000000	64255.000000	710.000000	13604.00000	1009.000000	16195.000000	14750.000000	3235.000000
25%	4341.000000	109231.250000	1129.500000	18702.50000	1200.000000	16582.750000	23862.000000	4435.500000
50%	6222.500000	129064.500000	1467.000000	20258.00000	1333.500000	17146.000000	26021.500000	4746.500000
75%	7753.250000	176476.500000	2764.750000	24482.75000	1580.500000	19029.250000	32706.250000	6014.500000

```
data_AllCausalities_Road_User_2021 = data_AllCausalities_Road_User.loc[data_AllCausalities_Road_User['Accident year'] == 2021]
data_AllCausalities_Road_User_2021_cummulative = data_AllCausalities_Road_User_2021.drop(columns="Accident year").set_index("Road user")
data_AllCausalities_Road_User_2021_cummulative["All casualties"].hist(figsize=(10, 5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9b9475e430>
```

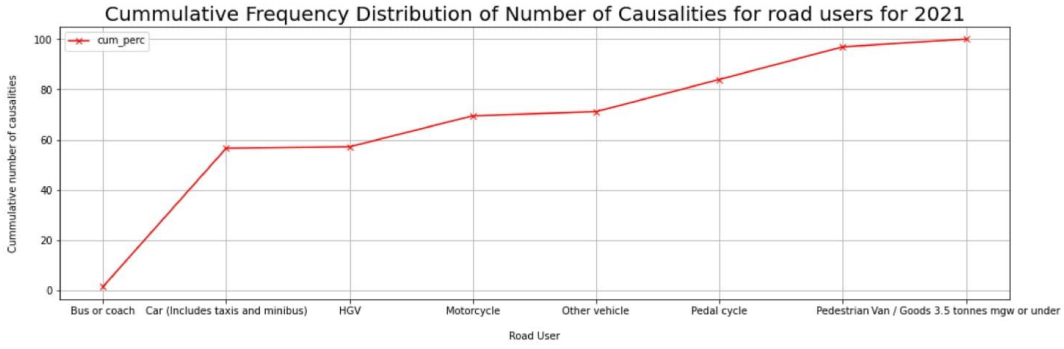



```
cas_2019 = data_AllCausalities_Road_User.loc[data_AllCausalities_Road_User['Accident year'] == 2019].set_index("Road user")['All casualties']
cas_2020 = data_AllCausalities_Road_User_2021['All casualties']
                                              s_2020)
```

Saving...

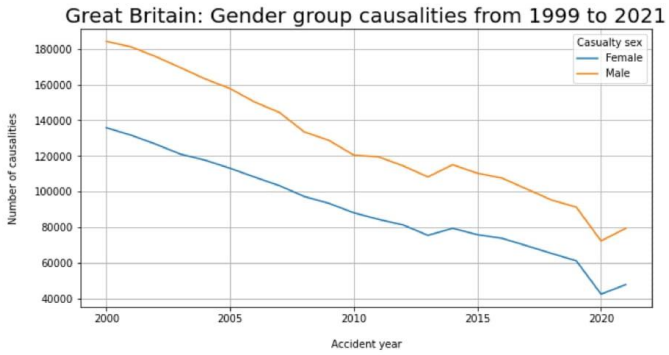
```
p value = 0.817697
t value = 0.234887
```

```
data_AllCausalities_Road_User_2021_cummulative['cum_sum'] = data_AllCausalities_Road_User_2021_cummulative.cumsum()
data_AllCausalities_Road_User_2021_cummulative['cum_perc'] = 100*data_AllCausalities_Road_User_2021_cummulative['cum_sum']/data_AllCausalitie
data_AllCausalities_Road_User_2021_cummulative.plot(kind='line',y='cum_perc', figsize=(17, 5), legend=True, marker= "x", color='red', rot=0);
plt.title("Cummulative Frequency Distribution of Number of Causalities for road users for 2021", fontsize=20)
plt.xlabel("Road User", labelpad=15)
plt.ylabel("Cummulative number of causalities", labelpad=15)
plt.grid()
plt.show()
```



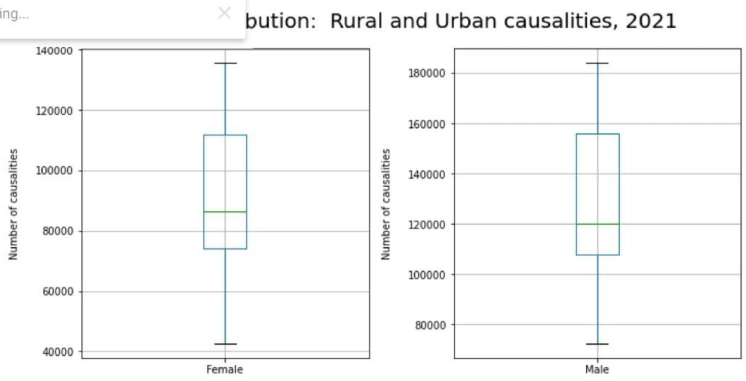

```
data_AllCausalities_Gender_Group = data.groupby(['Accident year','Casualty sex']).sum().reset_index()
data_AllCausalities_Gender_Group_pivot = data_AllCausalities_Gender_Group.pivot(index='Accident year', columns='Casualty sex', values='All ca
```

```
data_AllCausalities_Gender_Group_pivot.plot(x="Accident year", y=["Female","Male"], figsize=(10, 5))
plt.title('Great Britain: Gender group causalities from 1999 to 2021', fontsize=20)
plt.xlabel("Accident year", labelpad=15)
plt.ylabel("Number of causalities", labelpad=15)
plt.grid()
plt.show()
```



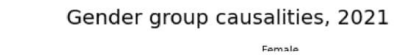

```
ncols = 2
nrows = 1

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
for col, ax in zip(data_AllCausalities_Gender_Group_pivot.drop(columns=["Accident year"]).columns, axs.T.ravel()):
    flierprops = dict(marker='o', markerfacecolor='r', markersize=12,
                      linestyle='none', markeredgecolor='g')
    data_AllCausalities_Gender_Group_pivot[[col]].boxplot(ax=ax, flierprops=flierprops);
    ax.set_ylabel('Number of causalities',labelpad=15)

fig.suptitle('Causalities Distribution:  Rural and Urban causalities, 2021', fontsize=20)
fig.set_size_inches(10, 5)
```

Saving...




```
labels = ["Female", "Male"]
values = data_AllCausalities_Gender_Group_pivot.iloc[[21]].drop(columns=['Accident year']).values.ravel()
plt.pie(values, labels = labels, autopct='%1.0f%%')
plt.title("\n Gender group causalities, 2021", fontsize=20)
```

```
Text(0.5, 1.0, '\n Gender group causalities, 2021')
```




```
skewValue = data_AllCausalities_Gender_Group_pivot.set_index("Accident year").skew(axis=0)
print(skewValue)
kurt = data_AllCausalities_Gender_Group_pivot.set_index("Accident year").kurt(axis=0 )
print(kurt)
```

```
Casualty sex
Female    0.061408
Male      0.217792
dtype: float64
Casualty sex
Female   -0.828875
Male     -0.980667
dtype: float64
```

```
data_AllCausalities_Gender_Group_pivot = data_AllCausalities_Gender_Group_pivot.set_index("Accident year")
correlation = data_AllCausalities_Gender_Group_pivot.corr(method='pearson')
correlation
```

Casualty sex	Female	Male
Casualty sex		
Female	1.000000	0.997561
Male	0.997561	1.000000

```
data_AllCausalities_Gender_Group_pivot
```

Casualty sex	Female	Male
Accident year		
2000	135803	184259
2003	121001	169492
2004	117573	163173
2005	113087	157797
2006	108111	150212
2007	103292	144363
2008	97250	133478
2009	93390	128711
2010	88117	120490
2011	84445	119498
2012	81277	114439
2013	75446	108213
2014	79413	115061
2015	75829	110299
2016	73776	107556
2017	69587	101379
2018	65305	95252
2019	61160	91265
2020	42488	72335
2021	47807	79393

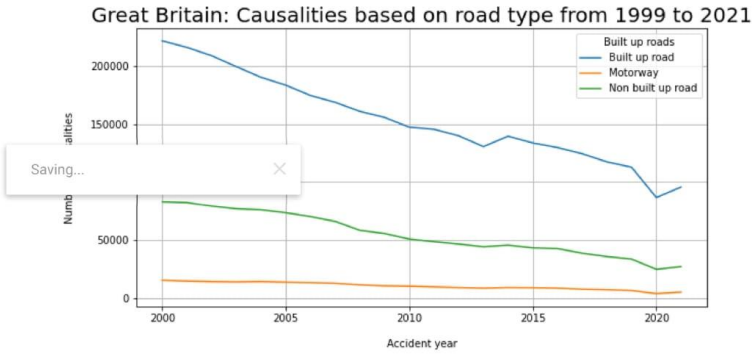

```
stats.ttest_ind(data_AllCausalities_Gender_Group_pivot['Male'], data_AllCausalities_Gender_Group_pivot['Female'])
```

```
Ttest_indResult(statistic=4.172481427056453, pvalue=0.00014798491110957495)
```

```
data_AllCausalities_Gender_Group_pivot.describe()
```

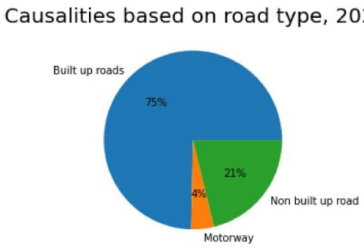
Casualty sex	Female	Male
count	22.000000	22.000000
mean	90562.318182	128342.636364
std	26529.272634	33164.753974
min	42488.000000	72335.000000
25%	74193.500000	107720.250000
50%	86281.000000	119994.000000
75%	111843.000000	155900.750000
max	135803.000000	184259.000000

```
data_AllCausalities_Road_Type = data.groupby(['Accident year','Built up roads']).sum().reset_index()
data_AllCausalities_Road_Type_T = data_AllCausalities_Road_Type.pivot(index='Accident year', columns='Built up roads', values='All casualties
data_AllCausalities_Road_Type_T =data_AllCausalities_Road_Type_T.drop(data_AllCausalities_Road_Type_T.index[[3]]).T.reset_index()
data_AllCausalities_Road_Type_T.plot(x="Accident year", y=["Built up road","Motorway","Non built up road"], figsize=(10, 5))
plt.title('Great Britain: Causalities based on road type from 1999 to 2021', fontsize=20)
plt.xlabel("Accident year", labelpad=15)
plt.ylabel("Number of causalities", labelpad=15)
plt.grid()
plt.show()
```




```
labels = ["Built up roads", "Motorway","Non built up road"]
values = data_AllCausalities_Road_Type_T.iloc[[21]].drop(columns=['Accident year']).values.ravel()
plt.pie(values, labels = labels, autopct='%1.0f%%')
plt.title("\n Causalities based on road type, 2021", fontsize=20)
```

```
Text(0.5, 1.0, '\n Causalities based on road type, 2021')
```

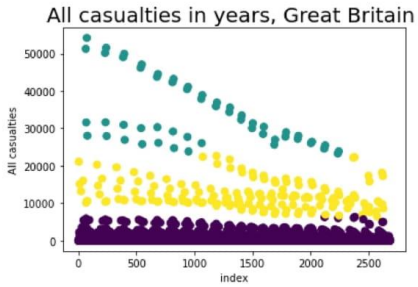



```
from sklearn import metrics
from sklearn.cluster import KMeans
from sklearn.preprocessing import LabelEncoder
```

```
#Select columns for X
X = data[["All casualties"]]
#Perform Clustering
kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
cluster_class = kmeans.predict(X)
print(cluster_class)
print("\n")
#Print off the labels of kmeans model
print('The kmeans labels are: ')
print(kmeans.labels_)
print("\n")
#Note there are two centroids and each centroid comprises the values for all the predictors
print(kmeans.cluster_centers_)
print("\n")
#Plot the graphs
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
plt.scatter(X.index, X["All casualties"], c=y_kmeans, s=50, cmap='viridis')
plt.ylabel('All casualties', fontsize=10)
plt.xlabel('index', fontsize=10)
plt.title('All casualties in years, Great Britain ', fontsize=20)
centers = kmeans.cluster_centers_
print("Center for cluster 0 is ", centers[0])
print("Center for cluster 1 is ", centers[1])
print("Center for cluster 2 is ", centers[2])
print("\n")
```

```
[0 0 0 ... 0 0 0]


The kmeans labels are:
[0 0 0 ... 0 0 0]


[[  435.91437803]
 [33891.45283019]
 [12309.53797468]]


Center for cluster 0 is  [435.91437803]
                         .45283019]
                         .53797468]
```

Saving...




```
data
```

	Accident year	Road user	Casualty sex	Urban rural	Built up roads	All casualties	Rating
0	2000	Pedestrian	Unknown	Urban	Built up road	43	Low Risk
1	2000	Pedestrian	Unknown	Rural	Built up road	5	Low Risk
2	2000	Pedestrian	Unknown	Rural	Non built up road	2	Low Risk

```
data_risk_group = data

# Add a new column named 'All casualties'
rating = []
for row in data['All casualties']:
    if row <= 435 :    rating.append('Low Risk')
    elif 435 < row <= 12309:   rating.append('Medium Risk')
    elif 12309 < row <= 33891:   rating.append('High Risk')
    elif row > 33891:  rating.append('Extremely High')
    else:           rating.append('Not_Rated')

data_risk_group['Rating'] = rating
data_risk_group
```

	Accident year	Road user	Casualty sex	Urban rural	Built up roads	All casualties	Rating
0	2000	Pedestrian	Unknown	Urban	Built up road	43	Low Risk
1	2000	Pedestrian	Unknown	Rural	Built up road	5	Low Risk
2	2000	Pedestrian	Unknown	Rural	Non built up road	2	Low Risk
3	2000	Pedestrian	Unknown	Unallocated	Non built up road	1	Low Risk
4	2000	Pedestrian	Male	Urban	Motorway	15	Low Risk
...	...	...	...	...	...	...	...
2682	2021	Other vehicle	Female	Urban	Built up road	384	Low Risk
2683	2021	Other vehicle	Female	Urban	Non built up road	2	Low Risk
2684	2021	Other vehicle	Female	Rural	Motorway	4	Low Risk
[illegible]	[illegible]	[illegible]	Female	Rural	Built up road	73	Low Risk
[illegible]	[illegible]	[illegible]	Female	Rural	Non built up road	57	Low Risk

Saving...

2687 rows × 7 columns

```
from sklearn.preprocessing import LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
data_risk_group_2021 = data_risk_group.loc[data_risk_group["Accident year"] == 2021]
data_risk_group_2021_updated = data_risk_group_2021.drop(columns=["All casualties","Accident year"])
le_class = LabelEncoder()
le_features = LabelEncoder()
data_risk_group_2021_updated['Rating'] = le_class.fit_transform(data_risk_group_2021_updated["Rating"])
X = data_risk_group_2021_updated[["Built up roads"]].apply(le_features.fit_transform)
y = data_risk_group_2021_updated["Rating"]

#Build and fit classififier
neigh = KNeighborsClassifier(n_neighbors=3)
model = neigh.fit(X.to_numpy(), y)
print(model)

#Use model for prediction
print("Predicted classification is", neigh.predict([[0]]))
print("Predicted probabilities are", neigh.predict_proba([[0.9]]))
```

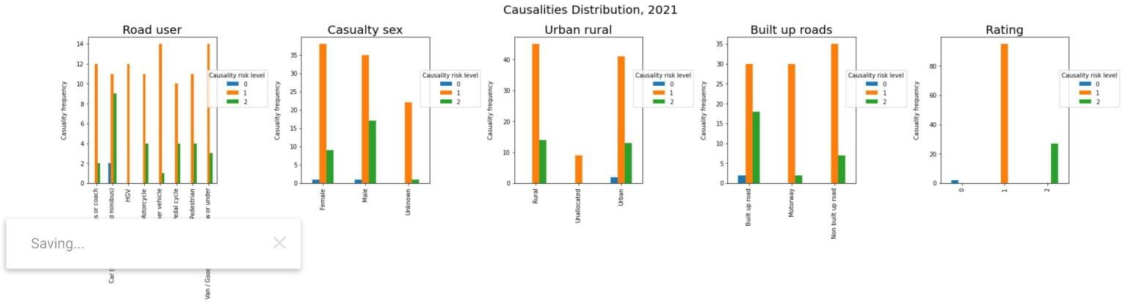
```
KNeighborsClassifier(n_neighbors=3)
Predicted classification is [1]
Predicted probabilities are [[0. 1. 0.]]
```

```
le_name_mapping = dict(zip(le_class.classes_, le_class.transform(le_class.classes_)))
print(le_name_mapping)
data_risk_group_2021_updated[["Built up roads"]].apply(le_features.fit_transform)
le_name_mapping = dict(zip(le_features.classes_, le_features.transform(le_features.classes_)))
print(le_name_mapping)
data_risk_group_2021_updated[["Urban rural"]].apply(le_features.fit_transform)
le_name_mapping = dict(zip(le_features.classes_, le_features.transform(le_features.classes_)))
print(le_name_mapping)
data_risk_group_2021_updated[["Casualty sex"]].apply(le_features.fit_transform)
```

```
le_name_mapping = dict(zip(le_features.classes_, le_features.transform(le_features.classes_)))
print(le_name_mapping)
data_risk_group_2021_updated[["Road user"]].apply(le_features.fit_transform)
le_name_mapping = dict(zip(le_features.classes_, le_features.transform(le_features.classes_)))
print(le_name_mapping)
```

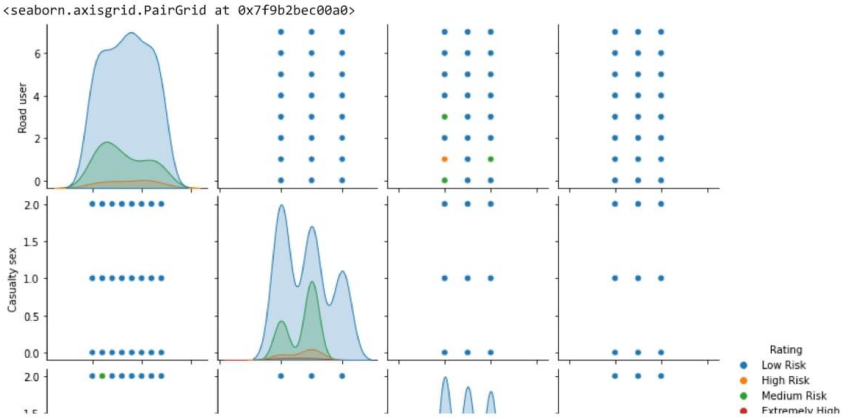
```
{'High Risk': 0, 'Low Risk': 1, 'Medium Risk': 2}
{'Built up road': 0, 'Motorway': 1, 'Non built up road': 2}
{'Rural': 0, 'Unallocated': 1, 'Urban': 2}
{'Female': 0, 'Male': 1, 'Unknown': 2}
{'Bus or coach': 0, 'Car (Includes taxis and minibus)': 1, 'HGV': 2, 'Motorcycle': 3, 'Other vehicle': 4, 'Pedal cycle': 5, 'Pedestrian
```

```
ncols = 5
nrows = 1

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
for col, ax in zip(data_risk_group_2021_updated.columns, axs.T.ravel()):
   CrosstabResult=pd.crosstab(index=data_risk_group_2021_updated[[col]][col].values,columns=y.values)
   CrosstabResult.plot(ax=ax, kind='bar', figsize=(17, 5), legend=True);
   ax.set_ylabel('Casuality frequency',labelpad=15)
   ax.set_xlabel('')
   ax.set_title(col, fontsize=20)
   ax.legend(title='Causality risk level',loc='lower center', bbox_to_anchor=(1.16, 0.5))
fig.suptitle('Causalities Distribution, 2021', fontsize=20)
fig.set_size_inches(25,7)
```




```
data_risk_group_2000 = data_risk_group.loc[data_risk_group["Accident year"] == 2000]
data_risk_group_2000_filtered = data_risk_group_2000.drop(columns=["All casualties","Accident year"])
le_features = LabelEncoder()
data_risk_group_2000_filtered.iloc[:,0 : 4] = data_risk_group_2000_filtered.iloc[:,0 : 4].apply(le_features.fit_transform)
sns.pairplot(data_risk_group_2000_filtered , hue='Rating')
```

```
<seaborn.axisgrid.PairGrid at 0x7f9b2bec00a0>
```



data_risk_group_2000_filtered

	Road user	Casualty sex	Urban rural	Built up roads	Rating
0	6	2	2	0	Low Risk
1	6	2	0	0	Low Risk
2	6	2	0	2	Low Risk
3	6	2	1	2	Low Risk
4	6	1	2	1	Low Risk
...	...	...	...	...	...
152	4	0	0	1	Low Risk
153	4	0	0	0	Low Risk
				2	Low Risk
				0	Low Risk
156	4	0	1	2	Low Risk

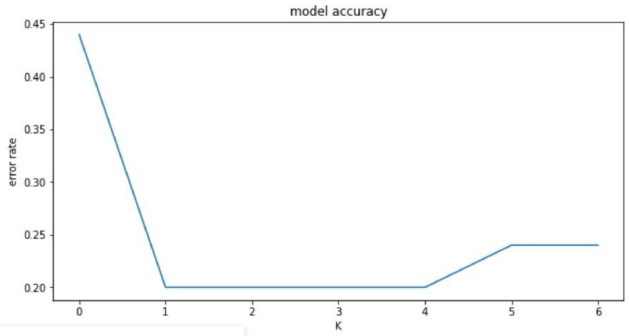
157 rows × 5 columns

Saving...

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
X = data_risk_group_2021_updated.drop(["Rating"], axis=1).apply(le_features.fit_transform)
#Split the data set into training data and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

#Train the model and make predictions
model = KNeighborsClassifier(n_neighbors = 8)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

#Performance measurement
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

#Selecting an optimal K value
error_rates = []
for i in np.arange(1, 8):
    new_model = KNeighborsClassifier(n_neighbors = i)
    new_model.fit(X_train, y_train)
    new_predictions = new_model.predict(X_test)
    error_rates.append(np.mean(new_predictions != y_test))

plt.figure(figsize=(10,5))
plt.plot(error_rates)
plt.title('model accuracy')
plt.ylabel('error rate')
plt.xlabel('K')
plt.show()
```

```
#Use model for prediction
print("Predicted classification is", model.predict([[6,2,2,0]]))
print("Predicted probabilities are", model.predict_proba([[0.9,0.1,0.3,0.4]]))
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-de
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-de
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-de
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.76      1.00      0.86        19
           2       0.00      0.00      0.00         5

    accuracy                           0.76        25
   macro avg       0.25      0.33      0.29        25
weighted avg       0.58      0.76      0.66        25

[[ 0  1  0]
 [ 0 19  0]
 [ 0  5  0]]
```



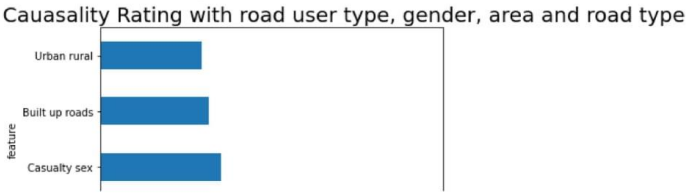

```
.    0.625 0.375]]
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but KNeighborsClassifier w
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but KNeighborsClassifier w
  warnings.warn(
```

```
a = data_risk_group_2021_updated['Rating'].unique()
print(sorted(a))
```

```
[0, 1, 2]
```

```
from sklearn.ensemble import ExtraTreesClassifier
#Build the model
model = ExtraTreesClassifier()
model.fit(X,y)

#Use inbuilt feature_importances of tree-based classifiers
print('Importance of features:')
print(model.feature_importances_)

#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title('Great Britain: Cauasality Rating with road user type, gender, area and road type', fontsize=20)
plt.ylabel('feature', fontsize=10)
plt.xlabel('Cuasality Rating ', fontsize=10)
plt.show()
```

```
Importance of features:
[0.49788766 0.18343901 0.15392864 0.16474469]
```




```
data_risk_group
```

	Accident year	Road user	Casualty sex	Urban rural	Built up roads	All casualties	Rating
0	2000	Pedestrian	Unknown	Urban	Built up road	43	Low Risk
1	2000	Pedestrian	Unknown	Rural	Built up road	5	Low Risk
2	2000	Pedestrian	Unknown	Rural	Non built up road	2	Low Risk
3	2000	Pedestrian	Unknown	Unallocated	Non built up road	1	Low Risk
4	2000	Pedestrian	Male	Urban	Motorway	15	Low Risk
...	...	...	...	...	...	...	...
2682	2021	Other vehicle	Female	Urban	Built up road	384	Low Risk
2683	2021	Other vehicle	Female	Urban	Non built up road	2	Low Risk
2684	2021	Other vehicle	Female	Rural	Motorway	4	Low Risk
2685	2021	Other vehicle	Female	Rural	Built up road	73	Low Risk
2686	2021	Other vehicle	Female	Rural	Non built up road	57	Low Risk

2687 rows × 7 columns

```
crosstab = pd.crosstab(data_risk_group["Built up roads"], data_risk_group["Rating"])
                                                  crosstab)
print("The value for degree of freedom is :", y)
print("Expected cell counts is:", z)
print("\n")

alpha = 0.01 #alpha is 0.01 or level of confidence is 99%
if x < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected, Built up roads and Rating have a relationship. Knowing the value of one variable does help yo
else:
    print("The null hypothesis is accepted, Built up roads and Rating have no relationship. It does not help you predict the value of another
```



```
The Chi Square value is: 194.89533459794126
The pvalue is: 3.8885579076107004e-37
The value for degree of freedom is : 9
Expected cell counts is: [[7.72608857e+00 3.59263119e+01 7.53679940e+02 2.40667659e+02]
 [5.19538519e+00 2.41585411e+01 5.06809825e+02 1.61836249e+02]
 [6.81801265e+00 3.17037588e+01 6.65097134e+02 2.12381094e+02]
 [2.60513584e-01 1.21138817e+00 2.54131001e+01 8.11499814e+00]]


The null hypothesis can be rejected, Built up roads and Rating have a relationship. Knowing the value of one variable does help you prec
```

```
a = data_risk_group["Built up roads"].unique()
print(sorted(a))
```

```
['Built up road', 'Motorway', 'Non built up road', 'Unknown']
```

```
b = data_risk_group["Rating"].unique()
print(sorted(b))
```

```
['Extremely High', 'High Risk', 'Low Risk', 'Medium Risk']
```

```
crosstab = pd.crosstab(data_risk_group["Urban rural"], data_risk_group["Rating"])
w, x, y, z = stats.chi2_contingency(crosstab)
```

```
print("The Chi Square value is:", w)
print("The pvalue is:", x)
print("The value for degree of freedom is :", y)
print("Expected cell counts is:", z)
print("\n")

alpha = 0.01 #alpha is 0.01 or level of confidence is 99%
if x < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected, Urban rural and Rating have a relationship. Knowing the value of one variable does help you p
else:
    print("The null hypothesis is accepted, Urban rural and Rating have no relationship. It does not help you predict the value of another va
```

```
The Chi Square value is: 154.97139099767622
The pvalue is: 8.219111039956285e-29
The value for degree of freedom is : 9
Expected cell counts is: [[9.08075921e+00 4.22255303e+01 8.85828061e+02 2.82865649e+02]
 [2.15109788e+00 1.00026051e+01 2.09839598e+02 6.70066989e+01]
 [1.48864905e-02 6.92221809e-02 1.45217715e+00 4.63714179e-01]
 [8.75325642e+00 4.07026424e+01 8.53880164e+02 2.72663937e+02]]

The null hypothesis can be rejected, Urban rural and Rating have a relationship. Knowing the value of one variable does help you predict
```

```
a = data_risk_group["Urban rural"].unique()
print(sorted(a))
```

```
['Rural', 'Unallocated', 'Unknown', 'Urban']
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers import Dense
import numpy
from numpy import array

le_features = LabelEncoder()
p(["Rating"], axis=1).apply(le_features.fit_transform)
ting"]
#Step 2 Split, train and test data
#Train, test and split the dataset. Random number generator, with popular integer see #numbers are 0 and 42
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
#Step 3 Preprocess the X training data by Scaling
sc = StandardScaler(with_mean=True, with_std=True)
sc.fit(X_train)

#Step3.1 Apply the scaler to the X training data
X_train_std = sc.transform(X_train)
#Step 3.2 Apply the SAME scaler to the X test data
X_test_std = sc.transform(X_test)

No_classes = 4
y_train = np_utils.to_categorical(y_train, No_classes)
y_test  = np_utils.to_categorical(y_test, No_classes)
# Step 5 create an deep NN model
#Output layer with 8 inputs
#1 hhidden layer with 64 neurons
#Output layer with 2 neurons for 2 classes (0,1)
model = Sequential()
model.add(Dense(4, input_dim=4, activation='relu'))
model.add(Dense(40, activation='relu'))
model.add(Dense(40, activation='relu'))
model.add(Dense(4, activation='sigmoid'))
print("\n")
print("Model Summary ")
print(model.summary())
print("\n")

#Step 6 Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

#Step 7 Fit Model
history = model.fit(X_train_std, y_train, batch_size=50, epochs=100,verbose=1, validation_split=0.20)

#Use model for Prediction but have to transform it first
```

```
Xnew1 = array([[4,1,0,0]])
#Xnew1_std = sc.transform(Xnew1)
Xnew1_std = pd.DataFrame(Xnew1,columns=['Road user','Casualty sex','Urban rural','Built up roads'])
# make a prediction

ynew1 = np.argmax(model.predict(Xnew1_std),axis=1)

# show the inputs and predicted outputs
print("\n")
print("The predicted classification for X1 \n", Xnew1_std)
print("\n Is :", ynew1)
print("\n")


#Evaluate the model
score = model.evaluate(X_test_std, y_test, verbose=1)
print("Names of the score metrics in model evaluation are: ")
print(model.metrics_names)
print("The score values are: Loss (Categorical Cross Entropy) and Accuracy (%)")
print(score)
print("\n")

# list all data in history
print(history.history.keys())
print("\n")

# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

Saving...

```
Model Summary
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_12 (Dense)            (None, 4)                 20

 dense_13 (Dense)            (None, 40)                200

 dense_14 (Dense)            (None, 40)                1640

 dense_15 (Dense)            (None, 4)                 164

=================================================================
Total params: 2,024
Trainable params: 2,024
Non-trainable params: 0
_________________________________________________________________
None
```

```
a = data_risk_group_2021_updated['Rating'].unique()
print(sorted(a)) # in 2021 dataset doesnot have 'Extremely High' category level
```

```
[0, 1, 2]
2/2 [==============================] - 0s 30ms/step - loss: 1.4140 - accuracy: 0.1757 - val_loss: 1.3773 - val_accuracy: 0.3158
```

```
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn import tree
from sklearn.tree import export_text

feature_cols = ['Road user', 'Casualty sex', 'Urban rural', 'Built up roads']
```

Saving...

```
ture_cols].apply(le_features.fit_transform)  # Features
Y = data_risk_group_2021_updated['Rating'] # Target variable
#Function for split dataset will have 3 parameters: features, target, test_set size
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=1) # 80% training and 20% test
#Build Decision Tree Model using Scikit Learn
# Create Decision Tree classifer object
clf = DecisionTreeClassifier()
# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

# Train Decision Tree Classifer

# r = export_text(clf)
# print(r)

#Predict the response for test dataset
y_pred = clf.predict(X_test)


#Evaluate the accuracy of the model (or classifier) for prediction
# Model Accuracy, how often is the classifier correct?
print("\n")
print("Accuracy for 80% training set and 20% test set :",
          metrics.accuracy_score(y_test, y_pred))

#How to improve the accuracy of the model? By tuning the number of features for the model

#Confusion matrix
print("\n")
print(confusion_matrix(y_test, y_pred))
print("\n")

cm = confusion_matrix(y_test, y_pred)


print('\nAccuracy: {:.2f}\n'.format(accuracy_score(y_test, y_pred)))
```

```
print('Micro Precision: {:.2f}'.format(precision_score(y_test, y_pred, average='micro')))
print('Micro Recall: {:.2f}'.format(recall_score(y_test, y_pred, average='micro')))
print('Micro F1-score: {:.2f}\n'.format(f1_score(y_test, y_pred, average='micro')))

print('Macro Precision: {:.2f}'.format(precision_score(y_test, y_pred, average='macro')))
print('Macro Recall: {:.2f}'.format(recall_score(y_test, y_pred, average='macro')))
print('Macro F1-score: {:.2f}\n'.format(f1_score(y_test, y_pred, average='macro')))

print('Weighted Precision: {:.2f}'.format(precision_score(y_test, y_pred, average='weighted')))
print('Weighted Recall: {:.2f}'.format(recall_score(y_test, y_pred, average='weighted')))
print('Weighted F1-score: {:.2f}'.format(f1_score(y_test, y_pred, average='weighted')))

print('\nClassification Report\n')
target_names=['Low Risk', 'Medium Risk', 'High Risk']
print(classification_report(y_test, y_pred, target_names=target_names))

#r = export_text(clf)
fig = plt.figure(figsize=(80,40))
tree.plot_tree(clf,
                   feature_names=feature_cols,
                   class_names=target_names,
                   filled=True)
```

Saving...

Data Set 4: Data Set 5:

Accuracy for 80% training set and 20% test set : 0.84

```
import matplotlib.pyplot as plt
import pandas as pd

df_casualties = pd.read_csv('/content/gdrive/MyDrive/Colab Notebooks/Report/Final/Casualties_Demographics.csv')
df_casualties = df_casualties.loc[:, ~df_casualties.columns.str.contains('^Unnamed')]
df_casualties.head()
df_casualties.tail()
```

	Road user type	Sex	Age group	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
457	Other or unknown vehicle passengers	All	40 to 49	53	41	40	36	44	38	44	30	25	24
458	Other or unknown vehicle passengers	All	50 to 59	35	24	19	25	33	38	27	26	22	17
459	Other or unknown vehicle passengers	All	60 to 69	27	15	14	17	16	18	21	14	7	5
460	Other or unknown vehicle passengers	All	70 and over	14	16	16	11	13	18	16	11	12	8
461	Other or unknown vehicle passengers	All	All ages	349	228	222	220	274	295	262	221	172	215

Epoch 86/100

```
df_casualties.shape
```

```
(462, 13)
```

2/2 [==============================] - 0s 29ms/step - loss: 0.5093 - accuracy: 0.7973 - val_loss: 0.9281 - val_accuracy: 0.6842

```
df_casualties_All_Ages = df_casualties.loc[(df_casualties["Age group"] != "All ages") & (df_casualties["Sex"] != "All")]
df_casualties_All_Ages_2021 = df_casualties_All_Ages.groupby(['Age group'])['2021'].sum().reset_index()
df_casualties_All_Ages_2021
```

	Age group	2021
0	16	1369
1	17 to 20	11575
2	21 to 24	11946
3	25 to 29	14458
4	30 to 39	17779
5	40 to 49	17779
6	50 to 59	15655
7	60 to 69	8552
8	70 and over	8048
9	Under 16	10880

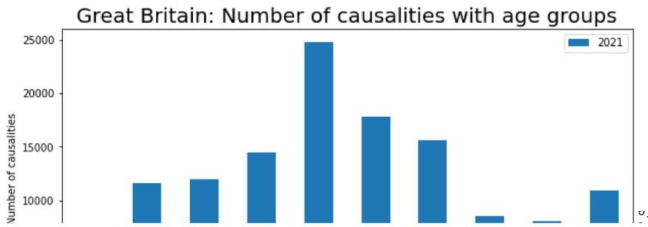
Saving...

```
skewValue = df_casualties_All_Ages_2021.set_index("Age group").skew(axis=0)
print(skewValue)
kurt = df_casualties_All_Ages_2021.set_index("Age group").kurt(axis=0 )
print(kurt)
```

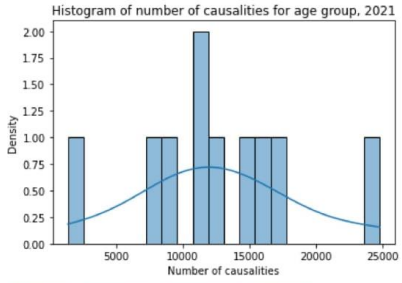
```
2021    0.285163
dtype: float64
2021    1.209706
dtype: float64
```

'loss': 'accuracy',

```
df_casualties_All_Ages_2021.plot(kind='bar', x="Age group", figsize=(10, 5))
plt.title('Great Britain: Number of causalities with age groups', fontsize=20)
plt.ylabel('Number of causalities', fontsize=10)
plt.xlabel('Age group ', fontsize=10)
plt.show()
```




```
sns.histplot(df_casualties_All_Ages_2021['2021'] , bins=20, kde=True)
plt.title("Histogram of number of causalities for age group, 2021")
plt.xlabel("Number of causalities")
plt.ylabel('Density')
plt.rcParams["figure.figsize"] = [10,5]
plt.show()
```




```
df_casualties_All_Ages_2021['2021']
```

```
0     1369
4    24816
5    17779
6    15655
7     8552
8     8048
9    10880
Name: 2021, dtype: int64
```

Saving...

```
import matplotlib.pyplot as plt
import pandas as pd

df_collisions = pd.read_csv('/content/gdrive/MyDrive/Colab Notebooks/Report/Final/Collisions_Demographics.csv')
df_collisions.head()
```

	Driver type	Sex	Age group	Severity	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
0	Pedal cyclists	Female	Under 16	All collisions	380	306	289	265	262	274	221	229	222	244
1	Pedal cyclists	Female	16	All collisions	37	33	48	49	29	37	24	37	45	28
2	Pedal cyclists	Female	17-20	All collisions	222	208	220	214	202	182	154	154	149	126
3	Pedal cyclists	Female	21-24	All collisions	343	367	415	363	339	302	297	252	264	250
4	Pedal cyclists	Female	25-29	All collisions	584	582	659	586	563	548	575	545	466	466

```
df_collisions.tail()
```

Driver type	Sex	Age group	Severity	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021

```
df_collisions.shape
```

```
(264, 14)
```

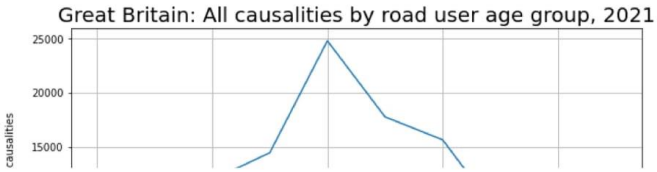
```
df_collisions_All_Ages = df_collisions.loc[(df_collisions["Age group"] != "All ages") & (df_collisions["Sex"] != "All") & (df_collisions["Dri
df_collisions_All_Ages_2021 = df_collisions_All_Ages.groupby(['Age group'])['2021'].sum().reset_index()
df_collisions_All_Ages_2021
```

	Age group	2021
0	16	662
1	17-20	11095
2	21-24	14176
3	25-29	18889
4	30-39	36395
5	40-49	27369
6	50-59	24131
7	60-69	12580
8	70 and over	9361
9	Under 16	2014

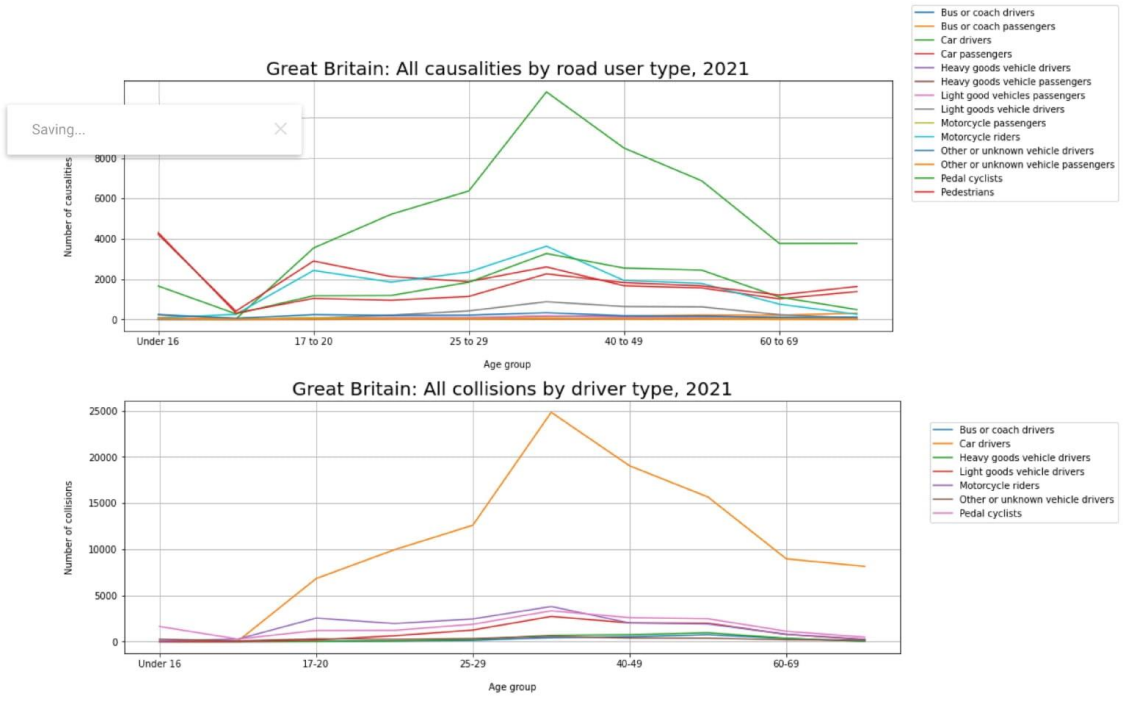
```
df_casualties_All_Ages_2021.plot(x="Age group", figsize=(10, 5))
plt.title('Great Britain: All causalities by road user age group, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.gca().legend_.remove()
plt.xlabel("Age group", labelpad=15)
plt.ylabel("Number of causalities", labelpad=15)
plt.grid()
```

Saving...

```
Age group", figsize=(10, 5))
ions by driver age group, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.gca().legend_.remove()
plt.xlabel("Age group", labelpad=15)
plt.ylabel("Number of collisions", labelpad=15)
plt.grid()
plt.show()
```



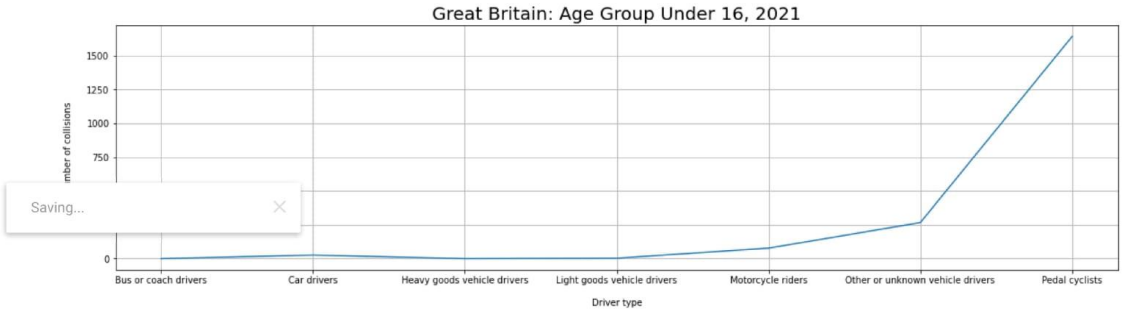

```
df_casualties_Road_User_2021 = df_casualties_All_Ages.groupby(['Age group','Road user type'])['2021'].sum().reset_index()
df_casualties_Road_User_2021 = df_casualties_Road_User_2021.pivot(index='Age group', columns='Road user type', values='2021')
df_casualties_Road_User_2021 = df_casualties_Road_User_2021.reindex(['Under 16','16','17 to 20','21 to 24','25 to 29','30 to 39','40 to 49','

df_collisions_Road_User_2021 = df_collisions_All_Ages.groupby(['Age group','Driver type'])['2021'].sum().reset_index()
df_collisions_Road_User_2021 = df_collisions_Road_User_2021.pivot(index='Age group', columns='Driver type', values='2021')
df_collisions_Road_User_2021 = df_collisions_Road_User_2021.reindex(['Under 16','16','17-20','21-24','25-29','30-39','40-49','50-59','60-69',

df_casualties_Road_User_2021.plot(x="Age group", figsize=(15, 5))
plt.title('Great Britain: All causalities by road user type, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.xlabel("Age group", labelpad=15)
plt.ylabel("Number of causalities", labelpad=15)
plt.grid()
plt.show()

df_collisions_Road_User_2021.plot(x="Age group", figsize=(15, 5))
plt.title('Great Britain: All collisions by driver type, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.xlabel("Age group", labelpad=15)
plt.ylabel("Number of collisions", labelpad=15)
plt.grid()
plt.show()
```




```
df_collisions_Road_User_2021
```

	Driver type	Age group	Bus or coach drivers	Car drivers	Heavy goods vehicle drivers	Light goods vehicle drivers	Motorcycle riders	Other or unknown vehicle drivers	Pedal cyclists
0		Under 16	0	26	0	3	77	266	1642
1		16	0	47	0	0	271	67	277
2		17-20	10	6828	21	174	2547	309	1206
3		21-24	70	9955	73	636	1960	260	1222
4		25-29	132	12608	229	1253	2454	337	1876
5		30-39	429	24838	680	2726	3803	580	3339
6		40-49	531	19045	753	2029	2023	392	2596
7		50-59	747	15666	962	1997	1906	375	2478
8		60-69	344	8968	364	782	800	198	1124

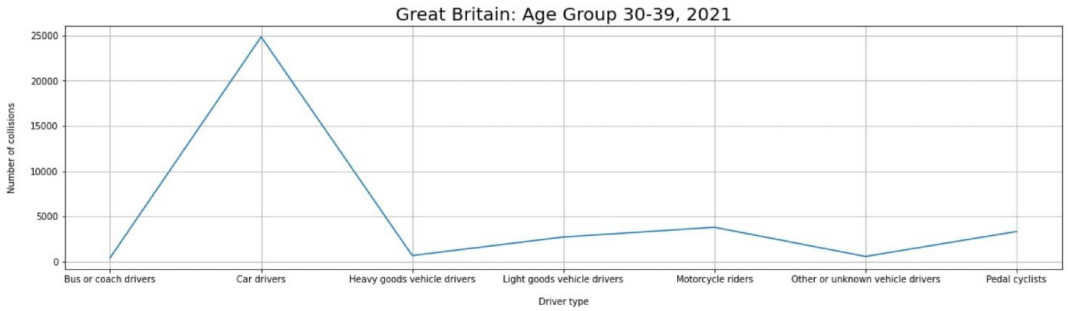
```
df_collisions_Under_16 = df_collisions_Road_User_2021.loc[(df_collisions_Road_User_2021["Age group"] == "Under 16")].drop(columns='Age group'
df_collisions_Under_16.T.plot(figsize=(20, 5))
plt.title('Great Britain: Age Group Under 16, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.xlabel("Driver type", labelpad=15)
plt.ylabel("Number of collisions", labelpad=15)
plt.grid()
plt.gca().legend_.remove()
plt.show()
```




```
df_collisions_Under_16 = df_collisions_Road_User_2021.loc[(df_collisions_Road_User_2021["Age group"] == "16")].drop(columns='Age group')
df_collisions_Under_16.T.plot(figsize=(20, 5))
plt.title('Great Britain: Age Group 16, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.xlabel("Driver type", labelpad=15)
plt.ylabel("Number of collisions", labelpad=15)
plt.grid()
plt.gca().legend_.remove()
plt.show()
```

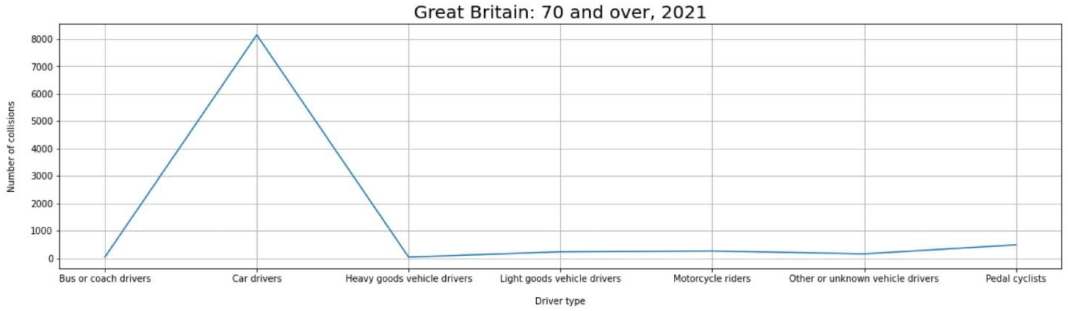
Great Britain: Age Group 16, 2021

```
df_collisions_Under_16 = df_collisions_Road_User_2021.loc[(df_collisions_Road_User_2021["Age group"] == "30-39")].drop(columns='Age group')
df_collisions_Under_16.T.plot(figsize=(20, 5))
plt.title('Great Britain: Age Group 30-39, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.xlabel("Driver type", labelpad=15)
plt.ylabel("Number of collisions", labelpad=15)
plt.grid()
plt.gca().legend_.remove()
plt.show()
```




```
df_collisions_Under_16 = df_collisions_Road_User_2021.loc[(df_collisions_Road_User_2021["Age group"] == "70 and over")].drop(columns='Age gro
df_collisions_Under_16.T.plot(figsize=(20, 5))
plt.title('Great Britain: 70 and over, 2021', fontsize=20)
plt.legend(loc='lower center', bbox_to_anchor=(1.16, 0.5))
plt.xlabel("Driver type", labelpad=15)
plt.ylabel("Number of collisions", labelpad=15)
plt.grid()
```

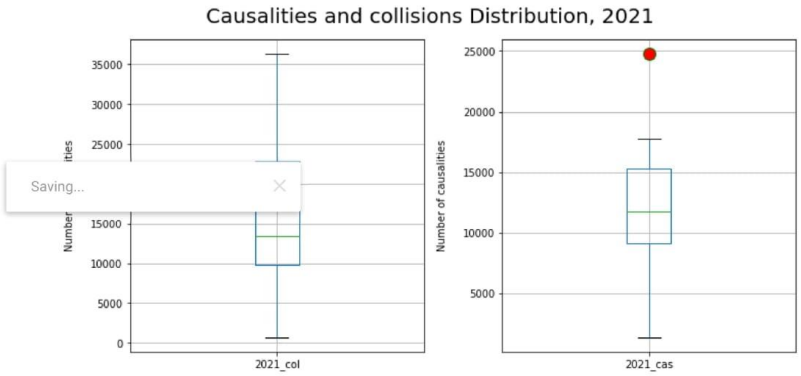
Saving...



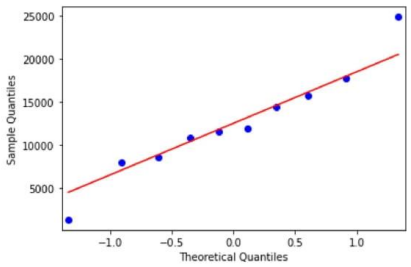

```
df_collisions_All_Ages_2021.loc[df_collisions_All_Ages_2021["Age group"] == "17-20", "Age group"] = "17 to 20"
df_collisions_All_Ages_2021.loc[df_collisions_All_Ages_2021["Age group"] == "21-24", "Age group"] = "21 to 24"
df_collisions_All_Ages_2021.loc[df_collisions_All_Ages_2021["Age group"] == "25-29", "Age group"] = "25 to 29"
df_collisions_All_Ages_2021.loc[df_collisions_All_Ages_2021["Age group"] == "30-39", "Age group"] = "30 to 39"
df_collisions_All_Ages_2021.loc[df_collisions_All_Ages_2021["Age group"] == "40-49", "Age group"] = "40 to 49"
df_collisions_All_Ages_2021.loc[df_collisions_All_Ages_2021["Age group"] == "50-59", "Age group"] = "50 to 59"
df_collisions_All_Ages_2021.loc[df_collisions_All_Ages_2021["Age group"] == "60-69", "Age group"] = "60 to 69"
df_collisions_All_Ages_2021
df_collisions_All_Ages_2021 = df_collisions_All_Ages_2021.rename(columns={'2021': '2021_col'})
df_casualties_All_Ages_2021 =  df_casualties_All_Ages_2021.rename(columns={'2021': '2021_cas'})
df_collisions_casualties_All_Ages_merge = pd.merge(df_collisions_All_Ages_2021, df_casualties_All_Ages_2021, on='Age group')
df_collisions_casualties_All_Ages_merge
```

	Age group	2021_col	2021_cas
0	16	662	1369
1	17 to 20	11095	11575
2	21 to 24	14176	11946
3	25 to 29	18889	14458
4	30 to 39	36395	24816
5	40 to 49	27369	17779
6	50 to 59	24131	15655
7	60 to 69	12580	8552
8	70 and over	9361	8048

```
ncols = 2
nrows = 1

fig, axs = plt.subplots(nrows, ncols, constrained_layout=True)
for col, ax in zip(df_collisions_casualties_All_Ages_merge.drop(columns=["Age group"]).columns, axs.T.ravel()):
    flierprops = dict(marker='o', markerfacecolor='r', markersize=12,
                  linestyle='none', markeredgecolor='g')
    df_collisions_casualties_All_Ages_merge[[col]].boxplot(ax=ax, flierprops=flierprops);
    ax.set_ylabel('Number of causalities',labelpad=15)

fig.suptitle('Causalities and collisions Distribution, 2021', fontsize=20)
fig.set_size_inches(10, 5)
```




```
from statsmodels.graphics.gofplots import qqplot
from matplotlib import pyplot as plt

#Load the dataset

# q-q plot for the age column, line = s is standardised line
qqplot(df_casualties_All_Ages_2021['2021_cas'], line='s')
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import datasets, linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from matplotlib.lines import Line2D

#Use only 1 feature - AGE to predict the target Y
cas_x = df_collisions_casualties_All_Ages_merge[['2021_cas']]
col_y = df_collisions_casualties_All_Ages_merge[['2021_col']]

#The scatterplot evidence that seemingly there is a trend
plt.scatter(cas_x, col_y)

#Split the dataset into training and testing sets (80%:20%)
x_train,x_test,y_train,y_test=train_test_split(cas_x, col_y,test_size=0.4)
print(x_train)
print("\n")
print(y_train)
print("\n")
print(x_test)


#Create linear regression object
regr = LinearRegression()

# Train the model using the training sets and reshape 1D arrays
regr.fit(x_train.to_numpy(), y_train.to_numpy())

# Make predictions using the testing set
y_pred = regr.predict(x_test.to_numpy())
y_pred2 = regr.predict([[8000]])
print("\n")
print("The predicted Y value for causalities = 8000 is: ", y_pred2)

# The coefficients
print('Coefficients: \n', regr.coef_)
```

Saving...

```
# The mean squared error
print('Mean squared error: ', mean_squared_error(y_test, y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: ', r2_score(y_test, y_pred))

 # Plot outputs
plt.scatter(x_test, y_test,  color='red')
plt.plot(x_test, y_pred, color='blue', linewidth=3)
plt.ylabel('collisions', fontsize=10)
plt.xlabel('casualities', fontsize=10)
plt.title('Linear regression model for causalities and collisions, Great Britain \n', fontsize=20)
plt.legend(["predict" , "train", "test"])
plt.xticks(())
plt.yticks(())

plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-143-2ce031ea58b1> in <module>
      9
     10 #Use only 1 feature - AGE to predict the target Y
---> 11 cas_x = df_collisions_casualties_All_Ages_merge[['2021_cas']]
     12 col_y = df_collisions_casualties_All_Ages_merge[['2021_col']]
     13

NameError: name 'df_collisions_casualties_All_Ages_merge' is not defined
```

SEARCH STACK OVERFLOW

```
df_collisions_casualties_All_Ages_merge = df_collisions_casualties_All_Ages_merge.set_index('Age group')
correlation = df_collisions_casualties_All_Ages_merge.corr(method='pearson')
correlation
```

	2021_col	2021_cas
2021_col	1.000000	0.919989

```
df_collisions_casualties_All_Ages_merge
```

Age group	2021_col	2021_cas
16	662	1369
17 to 20	11095	11575
21 to 24	14176	11946
25 to 29	18889	14458
30 to 39	36395	24816
40 to 49	27369	17779
50 to 59	24131	15655
60 to 69	12580	8552
70 and over	9361	8048
Under 16	2014	10880

```
df_collisions_casualties_All_Ages_merge. describe()
```

	2021_col	2021_cas
count	10.00000	10.000000
mean	15667.20000	12507.800000
std	11217.95818	6293.135481
min	662.00000	1369.000000
		0
		0
75%	22820.50000	15355.750000
max	36395.00000	24816.000000

Saving...

```
from scipy import stats
stats.ttest_1samp(df_collisions_casualties_All_Ages_merge['2021_cas'], 0)
```

```
Ttest_1sampResult(statistic=6.2851239480685335, pvalue=0.00014349934404667546)
```

```
col_2021 = df_collisions_casualties_All_Ages_merge['2021_col']
cas_2021 = df_collisions_casualties_All_Ages_merge['2021_cas']
print(col_2021.mean())
print(cas_2021.mean())
stats.ttest_ind(col_2021, cas_2021)
```

```
15667.2
12507.8
Ttest_indResult(statistic=0.776741115378785, pvalue=0.4473993713961305)
```