Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2020, 8, 134-157
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Prediction of Accident Severity Using Artificial Neural Network: A Comparison of Analytical Capabilities between Python and R

Imran Chowdhury Dipto[1], Md Ashiqur Rahman[1], Tanzila Islam[2], H M Mostafizur Rahman[3]

[1]Department of Electronics, Computing and Mathematics, University of Derby, Derby, UK
[2]Department of Computer Science and Intelligent Systems, Iwate University, Morioka, Japan
[3]BAC International Study Centre, Dhaka, Bangladesh
Email: dipto.imranchowdhury@gmail.com, ashiq.rahman6868@gmail.com, tanzilamohita@gmail.com, mostafizur@bacbd.org

How to cite this paper: Dipto, I.C., Rahman, Md.A., Islam, T. and Rahman H.M.M. (2020) Prediction of Accident Severity Using Artificial Neural Network: A Comparison of Analytical Capabilities between Python and R. *Journal of Data Analysis and Information Processing*, **8**, 134-157.
https://doi.org/10.4236/jdaip.2020.83008

Received: February 3, 2020
Accepted: August 4, 2020
Published: August 7, 2020

Abstract

Large amount of data has been generated by Organizations. Different Analytical Tools are being used to handle such kind of data by Data Scientists. There are many tools available for Data processing, Visualisations, Predictive Analytics and so on. It is important to select a suitable Analytic Tool or Programming Language to carry out the tasks. In this research, two of the most commonly used Programming Languages have been compared and contrasted which are Python and R. To carry out the experiment two data sets have been collected from Kaggle and combined into a single Dataset. This study visualizes the data to generate some useful insights and prepare data for training on Artificial Neural Network by using Python and R language. The scope of this paper is to compare the analytical capabilities of Python and R. An Artificial Neural Network with Multilayer Perceptron has been implemented to predict the severity of accidents. Furthermore, the results have been used to compare and tried to point out which programming language is better for data visualization, data processing, Predictive Analytics, etc.

Keywords

Artificial Neural Network, Accident Severity, Machine Learning, Python, R

1. Introduction

The government of UK collects and publishes information regarding traffic accidents across the country usually every year. This information contains aspects such as the severity of accidents, number of casualties, road conditions and so on, making the data sets quite interesting and comprehensive for analysis and

research [1].

1.1. The Dataset

The datasets used are publicly available from the Open Data website of the UK government (http://data.gov.uk/) where the data have been published by the Department of Transport. The dataset consists of two files in a CSV format. The name of the first dataset is "Accident_Information.csv" where each line of the data is identified by Accident_Index columns and the data ranges from 2005-2017. The second file "Vehicle_Information.csv" also contains the unique identifier and data of different passenger properties as columns. The data ranges from 2004-2016.

For this research, the Datasets have been collected from Kaggle which is a platform that nurtures, train and challenge data scientist from around the world to solve Data Science, Machine Learning and Predictive Analytics problems [2]. The idea of merging the Datasets has been taken from the work of Battendorf [3] published in Kaggle and the merged file will be modified by adding some extra features.

1.2. Business Context

According to Corrigan [4], despite collecting large quantities of traffic data, Transportation Departments of all levels are unable to use such data to good effect. Founded in 2015, a startup called ODN could predict when and where accidents are most likely to happen. Officials could use such information to direct safety efforts at the stretches of road where the impacts could be the biggest. In the context of this research, the UK government could use the information generated from a prediction system with a Neural Network predicting the accident severity and use this information to enhance the laws to build safer roads for the future.

1.3. The Chosen Analytics Technique

ANN is selected because ANN is an adaptive system that can change its structure based on external or internal information flowing through the Network [5]. From several studies, it is found that Neural Network proved to be a better predictor of accident severity. The choice is further justified from a research work of [6] where the researchers used ANN and DT to find the injury severity resulting from traffic accidents. Actual data from the National Automotive Sampling System (NASS)-General Estimates System (GES) were used. They used three Machine Learning models including ANN, DT and a Hybrid ANN, the results showed that the Hybrid model was the best and ANN also performed better than DT. Besides, most of the similar research work involved the use of ANN as found from the works of Abdelwahab (2001) [7], Zeng (2014) [8], Delen (2016) [9] and so on. From these works, it could be concluded that the choice of the Analytic Technique is appropriate and it could be used for strong research.

1.4. Programming Languages

In a recent worldwide survey, it was found that 83% of 24,000 data professionals used Python. Data Scientists consider Python because it is a general-purpose and dynamic programming language. In comparison to R, Python is found to work better than R with iterations less than 1000 and some consider it better than R for data manipulation tasks. It contains ideal packages for Machine Learning tasks and is inherently an Object-Oriented Programming Language [10]. In contrast, millions of analysts and data researchers use R Programming Language to handle their most difficult issues in the fields running from Computer Science to extensive marketing. R has become an important tool for organisations driven by Data Analytics. Some examples of companies using R include; Google, Facebook and Linkedin. R is considered as the best mechanism for Statistics, data analysis and Machine Learning. It is more than a statistical package as it allows for the creation of objects, functions and packages [11].

1.5. Project Aim and Objectives

This project aims to compare the analytical capabilities of Python and R. To reach the aim, two large data sets will be merged to form a single data set. Then data analysis will be performed on the data set to generate useful insights from the data. An Artificial Neural Network with Multilayer Perceptron will be implemented to predict the severity of accidents and finally, the results will be used to compare and contrast between the two chosen Programming Languages.

2. Artificial Neural Network

Machine Learning is an evolving branch of a set of Algorithms designed to copy the intelligence level of human beings by learning from their surroundings. These Algorithms are regarded as the key components of this era of Big Data. Techniques based on Machine Learning have been applied successfully in various fields ranging from computer vision, spacecraft engineering, entertainment and so on [12]. Machine Learning is required for tasks that are quite complicated for humans to code directly. Therefore the large sources of datasets or Big Data are provided to the Machine Learning Algorithm. It could then explore the data and complete tasks which it is programmed to perform [13].

Machine Learning is of three types which are supervised, unsupervised and reinforcement learning. In Supervised Learning, the Algorithms are provided with data that contains labelled examples. Then Algorithms are trained on these datasets. After the completion of the training process, these Algorithms could predict the outcomes when given new unseen data. Applications of Supervised Learning Algorithms include face recognition, Spam Classification and predicting Advertisement Popularity. In Unsupervised Learning Algorithms are provided with unlabelled data and such algorithms analyse the data to group them in such a way that a human being could make sense of this newly organised data. Applications of Unsupervised Algorithms are Recommendation Systems, gene-

rating Buying Habits and so on. On the other hand, in Reinforcement Learning an agent learns from an environment and is provided feedback depending on how correct the agent has learned from the data. Applications of Reinforcement Learning are Video Games, Industrial Simulations and so on [14].

2.1. How Artificial Neural Network Works

Inspired by the capabilities of the human brain for its incredible processing capabilities due to interconnected neurons. Artificial Neural Networks (ANN) are designed by processing units known as Perceptrons. These Perceptrons consists of one layer and they can solve linearly separable problems. However to solve non-linear problems Multilayer-Perceptrons are used which contains usually three layers which are; an input layer, one or more hidden layers and an output layer [15].

The building block of an Artificial Neural Network is a neuron. These are simple computational units consisting of weighed input signals that produce an output signal by an activation function. Each neuron has a bias which can be thought of as an input that always has the value 1 or 0 and it must also be weighed. Weights are often initialised to small random values, ranging between 0 to 0.3, however, more complex initialisation schemes could be used. The weighted inputs are added and passed through an activation function also known as the transfer function. The activation function is a mapping of the summed input to the output of the neuron. There are different types of activation functions used for specific purposes [16].

From the research work of Abdelwahab [7], the concept of Multilayer-Perceptron is illustrated in **Figure 1**.

Figure 1 consists of one input layer, one hidden layer and an output layer. In the figure displayed the input layer has K nodes and a bias node denoted as (Node 0). The hidden layer has J nodes and a bias node (Node 0). The output layer is denoted as I nodes with no bias node. The connections in the entire network are of feedforward type. This means that the connections are allowed from a layer of a certain index to layers of a higher index. It is also seen from the Figure that no connections are permitted from a layer of a certain index to a layer of lower index (*i.e.*, feedback connections). In addition, no connections are allowed among the nodes belonging to the same layer. The MLP network can operate in the training and testing phase. In the training phase, given a set of training data $\{x(1),d(1)\},\cdots,\{x(p),d(p)\},\cdots,\{x(PT),d(PT)\}$. Target is to map $\{x(1)\}$ to $\{d(1)\}$. The Backpropagation Algorithm is used to train the MLP. A simple representation of the Algorithm is illustrated in **Figure 2**.

From the figure it could be considered that output of MLP is equal to $x(p)$ applied across the input layer of MLP. To ensure that the output of MLP is same as $d(p)$ an error function is constructed which could be written as follow;

$$E(W)=\sum_{p=1}^{PT}\sum_{i=1}^{I}\left[d_i^2(p)-y_i^2(p)\right]^2 \tag{1}$$

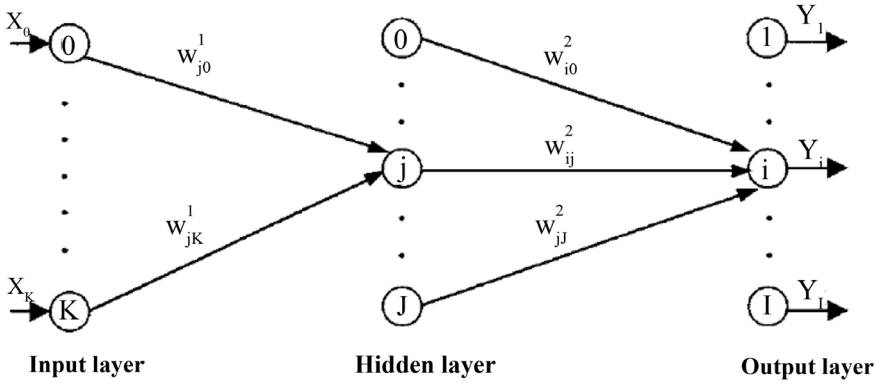

Figure 1. Structure of multilayer-perceptron [7].

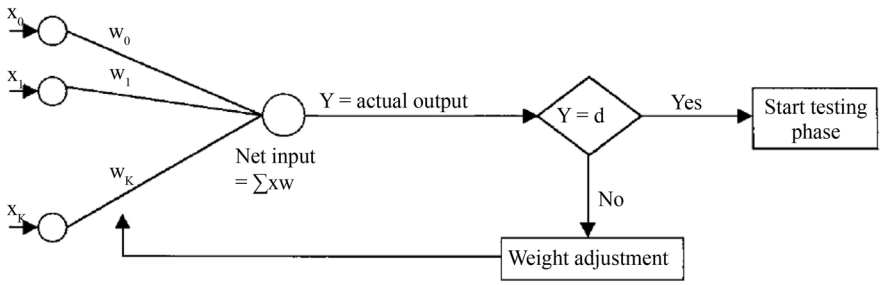

Figure 2. How backpropagation works [7].

Here:

- $E(W)$ = Error function to be minimised,
- W = Weight vector,
- PT = Number of training patterns,
- I = Number of output nodes,
- $d_i^2(p)$ = Desired output of node i when pattern p is introduced to the MLP,
- $y_i^2(p)$ = Actual output of node i when pattern p is introduced to the MLP.

The aim is to change the weight vector W so that the function shown above is minimised. By reducing the error function the actual output is taken closer to the desired output where it is assumed that Equation (1) can be differentiated, renowned optimization techniques could be applied to perform the minimisation task. One such technique is the gradient descent technique that changes W by an amount that is proportional to the negative gradient given by:

$$\Delta W = -\eta \nabla E(W) \tag{2}$$

where:

- ΔW = change of weight vector,
- η = learning parameter and
- $\nabla E(W)$ = gradient vector $E(W)$ with respect to weight vector W.

The equations that describe the change of weights are given as follow:

$$\Delta W_{ij}^2 = \eta \delta_i^2(p)(y)_j(p) \tag{3}$$

$$\Delta W_{jk} = \eta \delta_j(p) x_k(p) \tag{4}$$

where:

- ΔW_{ij}^2 = weight converging to the output layer,
- ΔW_{jk} = weight converging to the hidden layer,
- $\delta_i^2(p)$ = error term associated with output node i due to presentation of input pattern p,
- $\delta_j(p)$ = error term associated with hidden node j due to presentation of input pattern p,
- $y_j(p)$ = output of hidden node j due to presentation of input pattern p, and
- $x_k(p)$ = input component of index k in input pattern p.

2.2. Advantages and Limitations of Artificial Neural Network

Neural Networks require limited formal statistical training to be developed. And it could be developed by newcomers given that they are provided with appropriate datasets and software. Datasets on which it is possible to apply other techniques such as Logistic Regression, Artificial Neural Network could also be applied to the same dataset. Also, Neural Networks could be trained using inputs and outputs that are both categorical and continuous. Vast amounts of Neural Network Software packages are available ranging from user-friendly, the backpropagation only packages to complex packages with different training Algorithms. It is also possible to build a Neural Network model according to user's needs and there are vast resources available for implementing an Artificial Neural Network. Although the Network architecture and the mathematical calculations in the network are quite complicated, developers of a Neural Network require only some basic knowledge about the model's parameters to implement such a network. Therefore understandings of the calculations of the Backpropagation Algorithm and optimiser Algorithms used to train such a model is not required by developers. Neural Networks have the ability to detect complex non-linear relationships between dependent and independent variables. Given a dataset where there is a high amount of complex non-linear relationships, a Neural Network model automatically adjusts it's weight vectors. Furthermore, due to the use of hidden layers, Neural Networks possess the powers to detect all possible interactions between all the predictor variables. Neural Network models could be trained using a variety of training Algorithms other then Backpropagation Algorithms. In this field, researchers are continuously developing new and improved learning Algorithms which could be used to generate better results [17].

On the other hand, there are some drawbacks of Artificial Neural Networks. Firstly, Artificial Neural Networks are dependent on hardware. These networks need processors with parallel processing capabilities. As a result, a mid to high-end Central Processing Unit is desirable for implementing such a technique. Another problem of a Neural Network is that the behaviour of such a network is unexplained. When an Artificial Neural Network generates a probing result, no indication or reasons are generated resulting in a lack of trust in implementing such a technique. It is also quite difficult to determine the ideal structure of a network.

There are no set of rules which could be used to find the best structure. When building a Neural Network experience is required, for novices a painstaking process of trial and error is inevitable. Appropriate data preprocessing is required as Neural Networks are sensitive to feature scaling. This means the training and testing data of feature matrices have to be transferred to a range of values before training the Neural Network. In addition, the network is reduced to a certain error value when training finishes. The produced value does not provide optimum results [18].

2.3. Use of Artificial Neural Network in Industries

The selected Advanced Analytics Technique has a broad range of applications in real-world business problems. Currently, it has been successful in a number of industries. Neural Networks are ideal for identifying patterns or trends in data. Hence, it is suited for prediction or forecasting requirements that include: Sales forecasting, Prediction Control in industries, Customer Research and so on [19].

Neural Networks have been applied with success in Banking and Finance to solve problems like derivative security pricing and hedging, future price and exchange rate forecasting and stock performance. Currently, Neural Networks are being used as the underlying technology for decision making. It is an ongoing research area in Medicine and it is believed that Neural Networks will be used in a wide-scale in biomedical systems in the near future. Nowadays, the research is mostly conducted modelling body parts of human beings and recognising diseases from different types of scans [20].

Neural Networks are used in research to simulate the human cardiovascular systems which are used to compare real-time physiological measurements taken from patients. If such a process is carried out on a timely basis then potential harmful medical conditions could be detected at an early stage thus making the process of curing the disease much easier. ANNs are currently used experimentally to build electronic noses as they have the potential to be used in several applications in telemedicine. As the sense of smell is important for surgeons, collection and generation of patients sense of smell could be made possible by Neural Networks. Neural Networks are also used in Credit Evaluation as well. A company named HNC has developed several applications using Neural Network one of them is a Credit Scoring System used to increase the profitability of the current system by 27% [19].

Neural Networks could be used in Marketing where the tasks involve market segmentation where the market is divided into different groups of customers with different consumer behaviours. In these cases, a Neural Network could be trained to perform this task. Furthermore, Neural Networks are also used in Retail and Sales. As they can consider different variables at the same time such as market demand for products, customer's income, product price and so on. Supermarkets could use a Neural Network to predict future sales [20].

2.4. Designed ANN for the Experiment

The designed ANN architecture consists of three layers; an input layer, a hidden layer and an output layer as shown in Figure 3.

The input layer consists of the number of neurons that are equal to the number of input features of the dataset where "I" denotes the input layer neuron with "I_1" the first input feature where "n" is the last neuron. "H" denotes the first input neuron of the hidden layer and the same illustration is used to denote the input neurons. The output layer contains three neurons as the Network will predict three different outcomes. The neurons are shown as "O" with an underscore followed by the number of the neuron.

The neurons in the input and the hidden layers will be activated using the ReLu Function. The ReLu function is expressed as shown in Equation (5). From the shown equation it is seen that this function gives an output of "x" given that "x" is positive and 0 otherwise. To carry out this experiment, ReLu is considered among other Activation Functions such as Sigmoid and Tanh Functions because it is computationally less expensive as it involves less complicated calculations. ReLu function is also useful in cases where a Network has weights randomly initialised and about 50% of Network results in 0 activation due to the features of ReLu. It means that the lower number of Neurons would be activated resulting in a lighter Network [21]. Equation (3) shows the ReLu function.

$$A(x) = \max(0, x) \tag{5}$$

The Neurons of the output layer will be activated using the Softmax function.

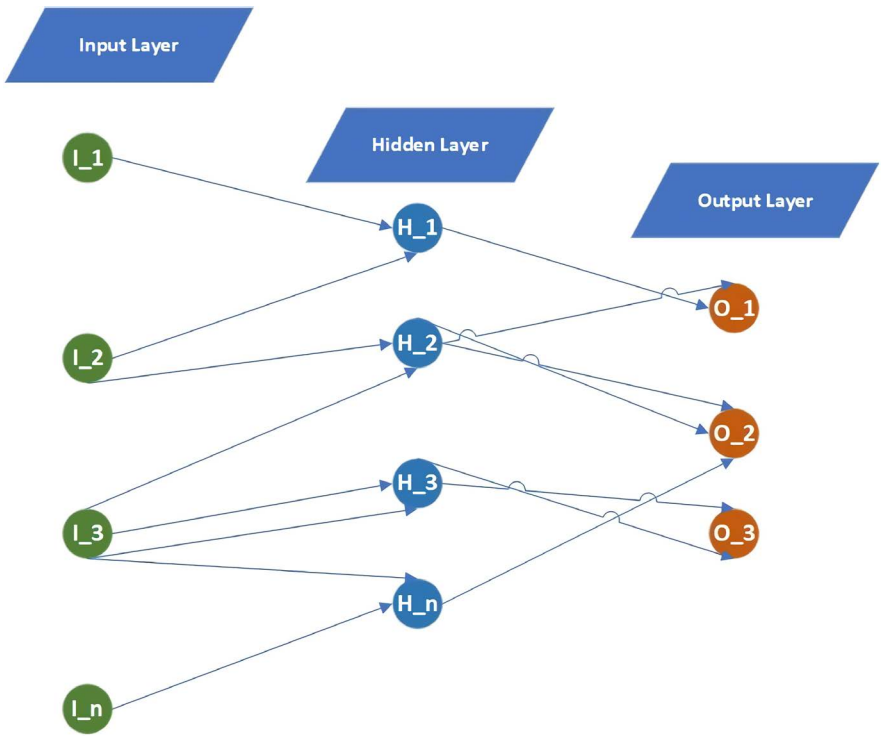

Figure 3. Design of artificial neural network for the experiment.

The Softmax function converts numbers (logits) into probabilities that add up to one. This function produces a vector showing the probability distributions of a list that are of a list of possible outcomes. Equation (6) shows that function where y_i is each element in the vector *y*. The ideal cost function to be used with the Softmax function is the Cross-Entropy Loss which measures the similarity of the predictions to the actual value [22].

$$S\left(y_i\right) = \frac{e^{y_i}}{\sum_i e^{y_j}} \tag{6}$$

The choice of a loss function and optimiser could be vital for a Neural Network to generate better results. Optimisation functions are used to modify the weights of a Neural Network. For this research, the Adaptive Moment Estimation or Adam, in short, will be implemented. In practice, Adam Optimiser outperforms other optimiser Algorithms such as Stochastic Gradient Descent and Adagrad [23].

2.5. Design of ANN for the Experiment

After creating and saving the dataset, the selected Programming Languages will then be used to find descriptive statistics about the dataset, data visualisation and training of Advanced Analytic Technique. Then the data will be preprocessed for implementing the Artificial Neural Network.

For Data Analysis and processing in Python, Numpy and Pandas will be used as Numpy is a package used for Data Analysis and Pandas library provides high-level data structures and methods to ease the data analysis process. The Matplotlib library is chosen as it provides quality plotting functionalities [24]. In addition, the Seaborn library is will be used which is built on top of Matplotlib providing attractive plotting options of statistical graphs [25]. For the Data Analysis and Visualisation part in R, "gglot2" library is chosen as it is over 10 years old and used by many users and organisations to generate millions of plots. It is a library that has methods to generate high-quality graphs in R [26]. To efficiently manipulate the used dataset the "dplyr" library is used as it provides a collection of tools for efficient data processing. It increases efficiency and it provides methods such as filter, group_by and so on [27].

For conversion of Categorical Features into numbers, suitable functionalities of Python and R have been used. For Python Pandas library is used whereas R in-built functions such as "data.matrix" and "as.numeric" functions have been used. As it is found from the research that one of ANNs drawbacks is that it requires scaled features as input data. To overcome this issue the most commonly used scaler called Standard Scaler is used. This method assumes that the data is normally distributed within each feature and will scale them such that the distribution gets centred around 0 with a standard deviation of 1 [28]. The formula is shown in Equation (7) where *x* is the input value, μ is the mean of column values and σ is Standard Deviation for a particular column [29].

$$z = \frac{x - \mu}{\sigma} \tag{7}$$

The dataset will be divided into a training set and a test set with training set being 80% and the rest for the test set. In Python, the data splitting and scaling has been performed using the scikit-learn library. The library provides the implementations of all the well-known Machine Learning Algorithms and steps needed for data processing and performance evaluation of Algorithms [30]. This library is used to split, process and compute the confusion matrix of the implemented Advanced Analytics Technique. In R external libraries will not be required to split the data as R has built-in matrix handling options. For scaling the data in R the "scale" method is used which implements the task of standard scaler [31]. For implementing the Neural Network the Keras library will be used. Keras is a high-level Neural Network API written in Python capable of running on top of other libraries such as TensorFlow, Theano or CNTK and it focuses on fast experimentation [32]. The same library is also used in R as the package provides an interface to Keras from within R [33].

Performance Evaluation Methods

To evaluate the performance of the trained ANN, the confusion matrix will be used which is a performance measuring tool for classification problems. It is a table with four combinations of predicted and actual values [34] [35]. The Confusion Matrix is shown in **Table 1**.

From the metric, the prediction accuracy will be calculated for the ANN trained using both Programming Languages as shown in the Equation (8).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{8}$$

As mentioned previously there are many libraries required in Python to perform different tasks such as data pre-processing and Machine Learning. To ensure that all these libraries are of the same version and in a fixed directory, the Anaconda Software will be used which is free software [36]. Installation of Anaconda provides all the required libraries for Data Science and even some IDE are also provided in the distribution. For the IDE the Scientific Python Development Environment (Spyder in short) is used for Python. This IDE is provided for free with Anaconda distribution and it is a strong environment for Scientific computing which is also written in the Python Language designed for scientists and engineers. It provides all the standard functionalities of IDEs such as debugging, interactive interface and so on. But it has unique features for

Table 1. Confusion matrix.

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Data Science such as a variable explorer which allows users to view the stored variables in memory, excellent data visualisation features and data exploration makes it a comprehensive development tool [37].

For the tasks performed in R the RStudio is chosen as the IDE as it allows developers to implement Machine Learning Algorithms [36]. RStudio is an ideal Integrated Development Environment designed for R. It is available as both open source and for professionals and it runs on platforms such as Windows, Mac and Linux. It has standard features such as syntax highlighting, code completion and so on. For Data Science it has data viewer, it integrates the tools used in R with a single environment, provides stunning Data Visualisation capabilities and so on [38]. The code of Python and R are provided in Appendix G-K and the Computer Configuration used for the experiment is provided in Appendix L in the Appendices Section.

2.6. Potential Contribution of ANN in the Chosen Context

With the use of Artificial Neural Network, it could be possible for the authorities working for the Government of the UK to get information on the type of accident that could occur in future. The trained Neural Network could be used as part of a software implementation where users could be able to input certain values such as day of the week, time, road condition, age of drivers, type of vehicles and so on. Then the software system could be able to generate future outcomes. Based on the results produced, the Government could enforce or update existing laws such as setting up new speed limits and so on.

2.7. Framework for the Evaluation of Python and R

1) ***Cost***: Cost is an important factor for any Organisation wanting to develop software products for deployment. It would be ideal for companies to use Analytical Tools that are less expensive or free so that they could save monthly or annual costs for paying subscription fees.

2) ***Ease of Use***: Having a user-friendly interface and writing fewer lines of codes could be beneficial. It would increase productivity and will allow developers to produce a system or analyze data much faster.

3) ***Available Libraries***: A variety of libraries is a must for any Programming Languages. Having access to suitable libraries would enable analysts, scientists and developers perform different tasks according to business needs.

4) ***Visualisation***: In order to make data meaningful for high-level authorities such as CEOs and Managers and so on the proper visualisation of data, is crucial. The visuals must be clear enough for the Management of a corporation to reach a business decision.

5) ***Community Support***: It is not always possible for Business Analysts, Data Scientists and Engineers to perform their tasks instantly. They could make coding errors or inexperienced employees could require to learn new concepts. In such cases, it is ideal to have good community support for Analytics Tools to

resolve such issues.

6) ***Language Unity***: Programming Language versions are constantly being updated. It is often found that major changes are seen in a change in method names for instance. A language that does not change the names of class or methods could be useful as users do not have to go through the hassle of going through new documentations.

7) ***Statistical Correctness***: As Statistics is important in making any business decision, having tools that would generate correct statistical results is a must for businesses.

8) ***Memory Consumption and Speed***: An Analytical Tool with less memory consumption is important because a less RAM consuming tool could lead to the generation of faster results and leave enough memory for the computer to perform other tasks.

9) ***Execution Time***: A faster tool would help produce results leading to increased productivity for organisations.

10) ***Machine Learning***: Since the main focus of this research is to implement and train an ANN to Predict the Severity of Accidents, hence a programming language should generate the most accurate results after training a Machine Learning model using the large source(s) of Big Data. Also, the language must have proper data processing functionalities.

3. Evaluation of Results

In this section, the results generated from Python and R will be evaluated based on the Analytical Framework developed in the previous section.

3.1. Cost

In this experiment two most commonly used Programming Languages for Data Science have been used which are Python and R. It has been already mentioned that both are free. For both languages, suitable IDEs were downloaded which were free editions and there are Professional versions available. Organizations could use either one of these Analytical Tools when Cost is a consideration.

3.2. Ease of Use

From a personal perspective, In terms of ease of use, it is found that the Syntax of both the Languages was quite simple which often seemed as writing English sentences. However, it could also be stated that users who are familiar with at least one other Programming Language such as C, C++ or Java will find it easy to learn Python and R but for beginner Python's. The syntax would be much easier and the claim is further justified by a post of [39].

3.3. Available Libraries

The libraries used for Python are Numpy and Pandas for Data Handling, scikit-learn for data pre-processing and displaying the results of the Confusion

Matrix and Matplotlib and Seaborn for Data Visualisation. Whereas, data handling libraries are not required in R and for visualisation, the ggplot2 library was used along with deplyr for additional features such as filtering, grouping data and so on. For implementing ANN Keras was used for both Languages. It could be concluded that both Languages have adequate libraries. This could is further confirmed by a report published by [40] in TechRepublic where the author mentioned that it is a tie between Python and R when it comes to available libraries.

3.4. Statistical Correctness

It is safe to say that R is better as fewer lines of codes are required to generate useful statistics as shown in **Figure A1** and **Figure A2**. Although the results generated in Python looks worse compared to R however, the results could be improved by using different IDEs rather than Spyder and it could be matched with R's results but it would take stronger logic and more lines of code making R the better choice for generating statistics. Also, it could be further assured from a post of [41] where it is found that fewer lines of code are required in R to generate Statistics.

3.5. Visualisations

To represent the data different visualisation techniques have been used in this research. The main goal is to implement a Machine Learning model which in this case is an Artificial Neural Network. The task of this model is to predict the Accident Severity depending on certain features such as Year, Weather Conditions, Road Conditions and so on. Therefore, the visualisations have been generated by showing the distribution of Accident Severity which are of three types; "Slight", "Serious" and "Fatal". Firstly the Pie Charts of the distribution of Accident Types in the data set and accidents distribution in different areas are shown in **Figure A3** and **Figure A4**. Then Bar Charts have been generated shown in **Figure A5** and **Figure A6**. Finally Line graphs are shown in **Figure A7** and **Figure A8** in **Appendices** Section produced by Python and R. From the results produced it is illustrated that results of Python represent better graphics than R. This is due to the seaborn library of Python.

3.6. Machine Learning

The target variable Accident Severity has three values. During model training the target values of Severity classes "Fatal", "Severe" and "Slight" were converted to "0", "1" and "2". From the result of the Confusion Matrix produced in Python as shown in **Figure A11** the total number of observation is 340,828 among which correct predictions for "Fatal", "Serious" and "Slight" classes are; 1767 and 292,220 making a total number of correct predictions equal to 292,988. Hence giving an Accuracy of 85.96% whereas in R shown the total correct observation is 292,944 and the total number of predictions are 340,890 thus the Accuracy is 85.94%. The Confusion Matrix produced by R is shown in **Figure A13** of **Ap-**

pendices.

3.7. Community Support

Considering R, it has large community support that comes from mailing-lists, documentation contributed by users, active Stackoverflow members and so on. On the other hand, Python also has strong community support found from the mentioned list of supporting sources that are available in R [41].

3.8. Execution Time and Memory Consumption

To carry out the tests two separate Script files were created using the Languages. One script was created to perform Data Visualisation and on the other script code for ANN implementation is written. The results show that for Data Visualisation R is faster whereas Python is faster for training the ANN. In contrast, R has a lower memory consumption compared to Python when comes to both Data Visualisation and training the Machine Learning Algorithm. The results are shown in **Figure A9**, **Figure A10**, **Figure A12**, and **Figure A14** in the Appendices Section.

4. Conclusions and Future Recommendations

In this research, a comparison has been made between Python and R which are the most commonly used Programming Languages used by Data Scientists. For this research, the context has been set to predict the Severity of Accidents from the data published by the Government of the UK using ANN. Two large sources of Big Data have been used and combined into a single Dataset. After carrying out the tests it could be concluded that Python was marginally better at Data Visualisation and was more accurate at predicting the outcomes with faster execution time. On the other hand, it was easy to generate Descriptive Statistics in R and overall memory consumption of R was lower than Python. In terms of Costs and Community Support both were on par, however, when it comes to ease of use and Machine Learning tasks Python shines as beginners should find Python easy to learn and due to the fantastic Library in Python called scikit-learn data processing, implementing algorithms and assessing model performance are much easier in Python than R. Hence, from this research, Python is slightly ahead in most of the tasks performed in this research project.

The results of Data Visualisation show that in this dataset there are values which are "Data Missing or Out of Range", for a more conclusive research, a larger dataset with added features such as "Name of Road", "Street Number", "Vehicle Model" and more is required with appropriate values in the columns. For an extension to the current research, instead of using a Machine Learning model, it could be more reasonable to use a Deep Learning model instead to test the Analytical Capabilities of both Python and R.

Acknowledgements

I would like to express my sincere gratitude towards several people because

without their help it would not have been possible to complete this work. First I would like to thank Mr. Liton Hossain for selling and setting up the entire Computer System used for this research work. Then I would like to thank Thanasis for publishing the datasets in Kaggle and Britta Bettendorf who published her work in Kaggle from which I have borrowed the concepts of combining the Datasets, the Government of the United Kingdom for making the data publicly available. Finally, I would like to thank Mrs. Tanzila Islam for her guidance for conducting this research and my Supervisor Mr. H M Mostafizur Rahman.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Thanasis (2017) UK Road Safety: Traffic Accidents and Vehicles. Kaggle.com. https://www.kaggle.com/tsiaras/uk-road-safety-accidents-and-vehicles

[2] Hassan, Z. (2019) What Is Kaggle, Why I Participate, What Is the Impact? Kaggle.com. https://www.kaggle.com/getting-started/44916

[3] Battendorf, B. (2019) Predicting Accident_Severity with RF + SMOTE. Kaggle.com. https://www.kaggle.com/brittabettendorf/predicting-accident-severity-with-rf-smote

[4] Corrigan, J. (2019) How Data Analytics Could Help the Government Reduce Traffic Deaths. Nextgov.com. https://www.nextgov.com/analytics-data/2019/03/how-data-analytics-could-help-government-reduce-traffic-deaths/155323

[5] Alkheder, S., Taamneh, M. and Taamneh, S. (2016) Severity Prediction of Traffic Accident Using an Artificial Neural Network. *Journal of Forecasting*, **36**, 100-108. https://doi.org/10.1002/for.2425

[6] Chong, M., Abraham, A. and Paprzycki, M. (2004) Traffic Accident Data Mining Using Machine Learning Paradigms. *Fourth International Conference on Intelligent Systems Design and Applications*, Hungary, 415-420.

[7] Abdelwahab, H.T. and Abdel-Aty, M.A. (2001) Development of Artificial Neural Network Models to Predict Driver Injury Severity in Traffic Accidents at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*, **1746**, 6-13. https://doi.org/10.3141/1746-02

[8] Zeng, Q., Huang, H.-L., Xu, P.-P. and Ma, M. (2014) Developing an Optimized Artificial Neural Network to Predict Traffic Crash Injury Severity. 14*th COTA International Conference of Transportation Professionals*, Changsha, 4-7 July 2014, 2396-2407. https://doi.org/10.1061/9780784413623.229

[9] Delen, D., Sharda, R. and Bessonov, M. (2006) Identifying Significant Predictors of Injury Severity in Traffic Accidents Using a Series of Artificial Neural Networks. *Accident Analysis & Prevention*, **38**, 434-444. https://doi.org/10.1016/j.aap.2005.06.024

[10] Mitchell, M. (2019) Programming Languages for Data Scientists. Medium. https://towardsdatascience.com/programming-languages-for-data-scientists-afde2eaf5cc5

[11] Agrawal, V. (2016) Applications of R Programming in Real World. e-Learning In-

dustry. https://elearningindustry.com/applications-r-programming-r-eal-world

[12] Naqa, I.E. and Murphy, M.J. (2015) What Is Machine Learning? In: *Machine Learning in Radiation Oncology*, Springer, Berlin, 3-11. https://doi.org/10.1007/978-3-319-18305-3_1

[13] Le, J. (2018) The 10 Neural Network Architectures Machine Learning Researchers Need to Learn. Medium. https://medium.com/cracking-the-data-science-interview/a-gentle-introduction-to-neural-networks-for-machine-learning-d5f3f8987786

[14] Heidenreich, H. (2018) What Are the Types of Machine Learning? Medium. https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f

[15] Karaylan, T. and Klç, O. (2017) Prediction of Heart Disease Using Neural Network. 2017 *International Conference on Computer Science and Engineering* (*UBMK*), Antalya, 5-8 October 2017, 719-723. https://doi.org/10.1109/UBMK.2017.8093512

[16] Brownlee, J. (2016) Crash Course on Multi-Layer Perceptron Neural Networks. Machine Learning Mastery. https://machinelearningmastery.com/neural-networks-crash-course

[17] Tu, J.V. (1996) Advantages and Disadvantages of Using Artificial Neural Networks versus Logistic Regression for Predicting Medical Outcomes. *Journal of Clinical Epidemiology*, **49**, 1225-1231. https://doi.org/10.1016/S0895-4356(96)00002-9

[18] Mijwell, M.M. (2018) Artificial Neural Networks Advantages and Disadvantages. 18.

[19] Mihajlovic, I. (2019) Artificial Neural Networks in Practice. Medium. https://towardsdatascience.com/artificial-neural-networks-in-practice-c950c4be47ee

[20] Shah, J. (2017) Neural Networks for Beginners: Popular Types and Applications. Medium. https://blog.statsbot.co/neural-networks-for-beginners-d99f2235efca

[21] Sharma V, A. (2017) Understanding Activation Functions in Neural Networks. Medium. https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

[22] Uniqtech (2018) Understand the Softmax Function in Minutes. Medium. https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d

[23] Agrawal, A. (2017) Loss Functions and Optimization Algorithms. Demystified. Medium. https://medium.com/data-science-group-iitr/loss-functions-and-optimization-algorithms-demystified-bb92daff331c

[24] McKinney, W. (2012) Python for Data Analysis. OReilly Media, Inc., Sebastopol.

[25] Sharma, M. (2018) Data Visualization Using Seaborn. Medium. https://towardsdatascience.com/data-visualization-using-seaborn-fc24db95a850

[26] Wickham, H. (2019) Ggplot2 Package. R Documentation. Rdocumentation.org. https://www.rdocumentation.org/packages/ggplot2/versions/3.2.1

[27] Wickham, H. (2014) Introducing dplyr. RStudio Blog. Rstudio.com. https://blog.rstudio.com/2014/01/17/introducing-dplyr/#:~:targetText=dplyr%20is%20a%20new%20package,focussing%20on%20only%20data%20frames.targetText=With%20dplyr%20%2C%20anything%20you%20can,to%20a%20remote%20database%20table

[28] Boris, E. (2019) Demystifying Feature Scaling. Medium. https://becominghuman.ai/demystifying-feature-scaling-baff53e9b3fd

[29] Scikit-learn.org (2019) sklearn.preprocessing.StandardScaler-scikit-learn 0.23.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[30] Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F. and Mueller, A. (2011) Scikit-Learn. GetMobile: Mobile Computing and Communications. *Journal of Machine Learning Research*, **12**, 2825-2830.

[31] Crawley, M.J. (2013) The R Book. Wiley, Chichester.

[32] François, C. (2018) Keras: The Python Deep Learning Library. Astrophysics Source Code Library.

[33] Arnold, T.B. (2017) kerasR: R Interface to the Keras Deep Learning Library. *The Journal of Open Source Software*, **2**, 296. https://doi.org/10.21105/joss.00296

[34] Narkhede, S. (2018) Understanding Confusion Matrix. Medium. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[35] Ferreira, H. (2018) Confusion Matrix and Other Metrics in Machine Learning. Medium. https://medium.com/hugo-ferreiras-blog/confusion-matrix-and-other-metrics-in-machine-learning-894688cb1c0a

[36] Mueller, J.P. and Massaron, L. (2016) Machine Learning for Dummies. John Wiley et Sons, Hoboken.

[37] SPYDER (2018) SPYDER. Spyder-ide.org. https://www.spyder-ide.org

[38] RStudio (2019) RStudio IDE Features. Rstudio.com. https://rstudio.com/products/rstudio/features/

[39] Sayantini (2019) R vs Python. Best Programming Language for Data Science and Analysis. Codementor.io. https://www.codementor.io/@sayantinideb/r-vs-python-best-programming-language-for-data-science-and-analysis-te05xgx98

[40] Rayome, A.D. (2019) R vs. Python: Which Is a Better Programming Language for Data Science? TechRepublic. https://www.techrepublic.com/article/r-vs-python-which-is-a-better-programming-language-for-data-science

[41] Willems, K. (2015) Choosing R or Python for Data Analysis? An Infographic. DataCamp Community. https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

Appendices

Results Produced in Python and R

```
  Accident_Severity  Daytime  ...  Engine_Capacity_.CC.           Vehicle_Manoeuvre
0            Slight        3  ...                8268.0         Slowing or stopping
1            Slight        5  ...                8300.0  Going ahead right-hand bend
2            Slight        2  ...                1769.0           Going ahead other
3            Slight        4  ...                  85.0           Going ahead other
4            Slight        2  ...                2976.0                Moving off

[5 rows x 16 columns]
             Daytime   Speed_limit  ...  Age_of_Vehicle  Engine_Capacity_.CC.
count  1.704138e+06  1.704138e+06  ...    1.704138e+06          1.704138e+06
mean   2.574147e+00  3.982655e+01  ...    7.113812e+00          2.034436e+03
std    1.116715e+00  1.460575e+01  ...    4.706821e+00          1.909882e+03
min    1.000000e+00  1.000000e+01  ...    1.000000e+00          1.000000e+00
25%    2.000000e+00  3.000000e+01  ...    3.000000e+00          1.299000e+03
50%    3.000000e+00  3.000000e+01  ...    7.000000e+00          1.598000e+03
75%    3.000000e+00  5.000000e+01  ...    1.000000e+01          1.997000e+03
max    5.000000e+00  7.000000e+01  ...    1.110000e+02          9.600000e+04

[8 rows x 6 columns]
```

Figure A1. Descriptive statistics generated in Python.

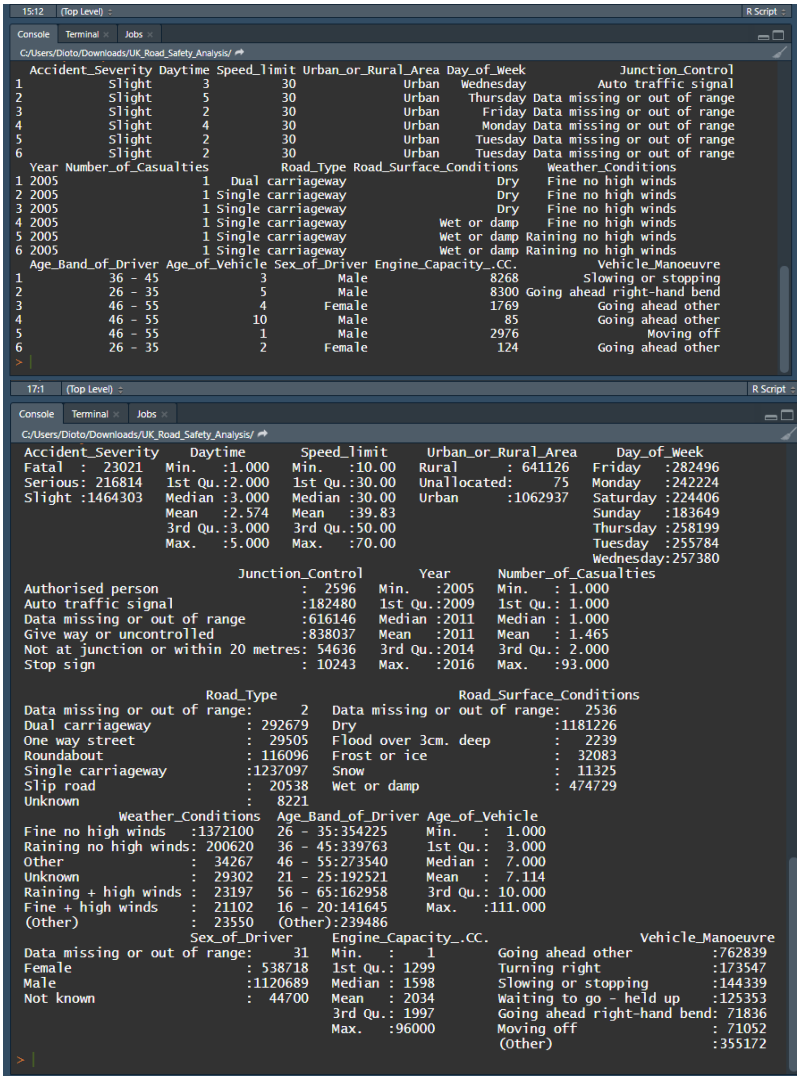

Figure A2. Descriptive statistics generated in R.

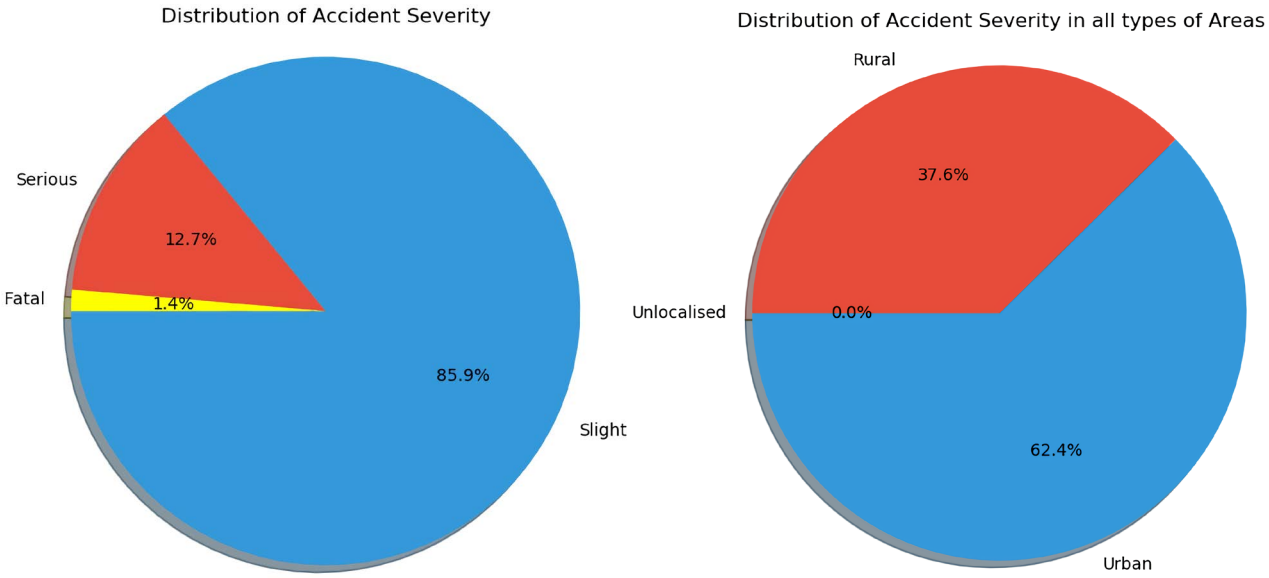

Figure A3. Pie charts generated in Python.

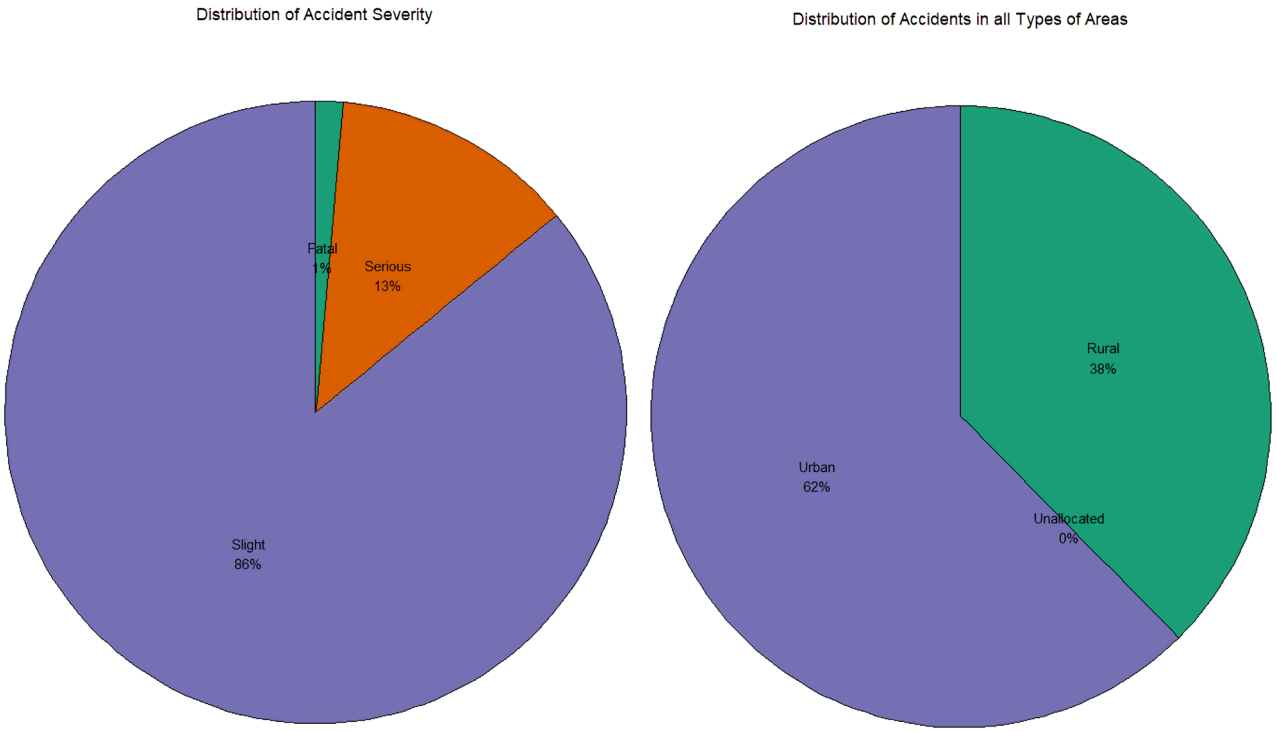

Figure A4. Pie charts generated in R.

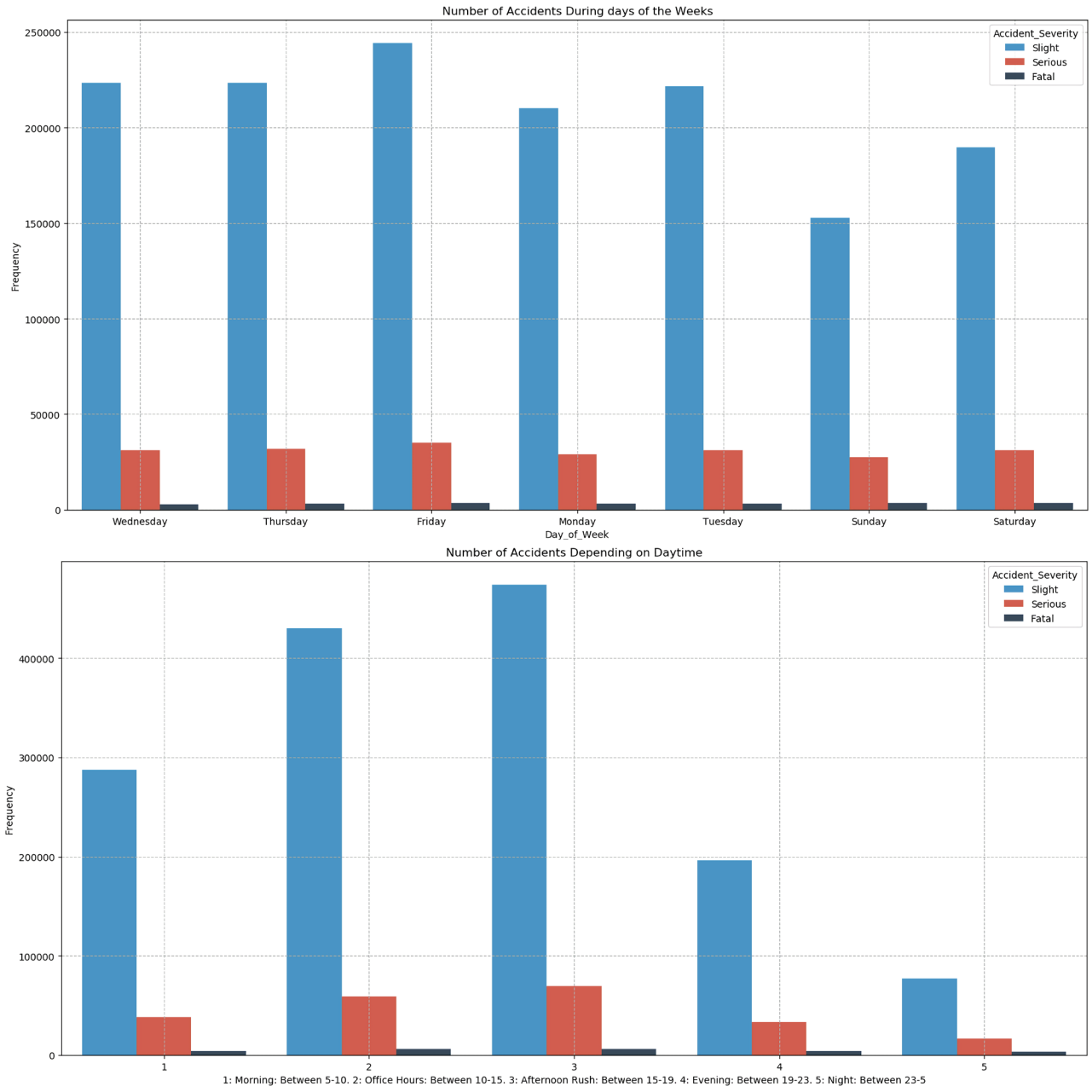

Figure A5. Bar graphs produced using Python.

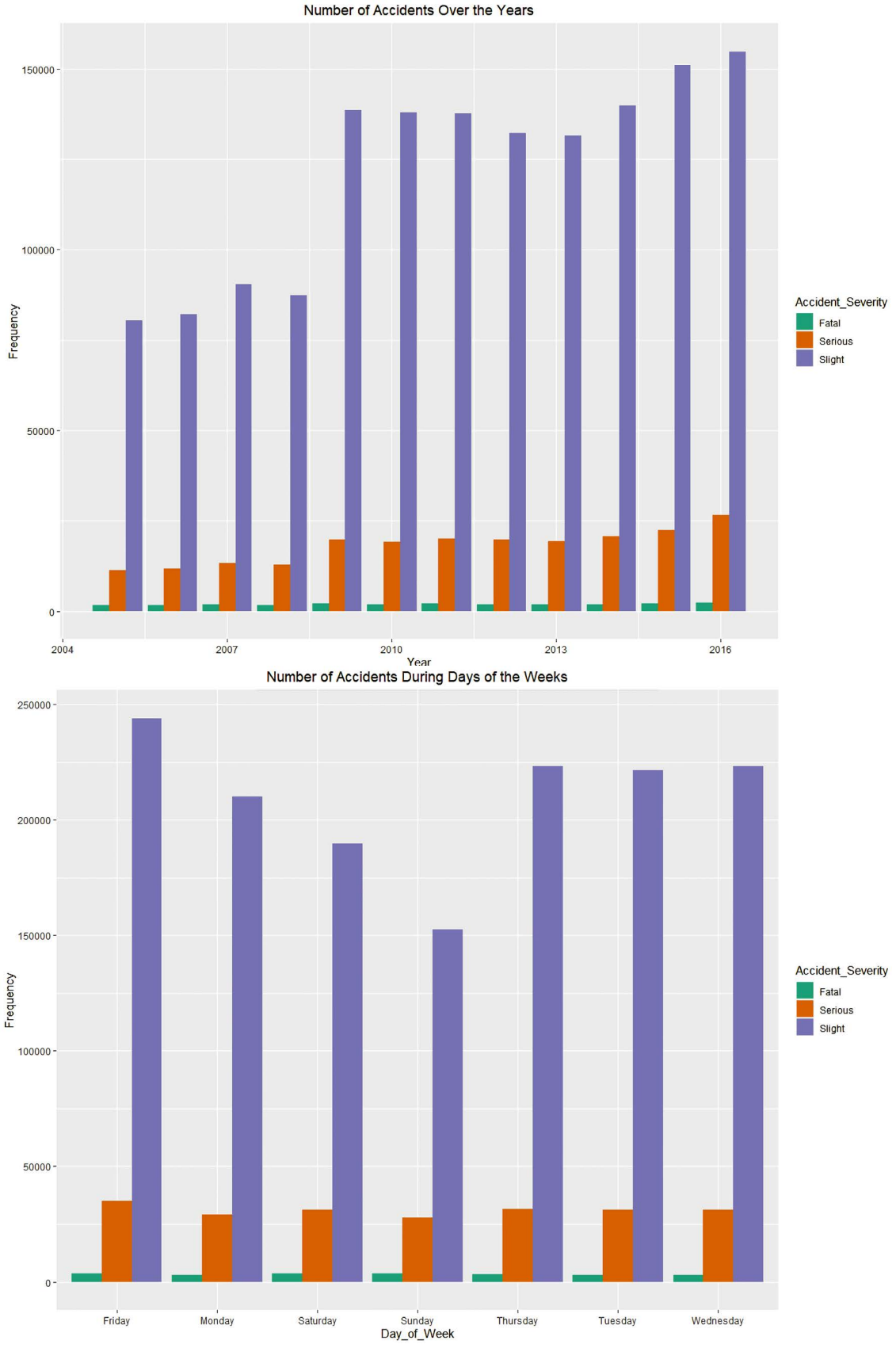

Figure A6. Bar graphs generated using R.

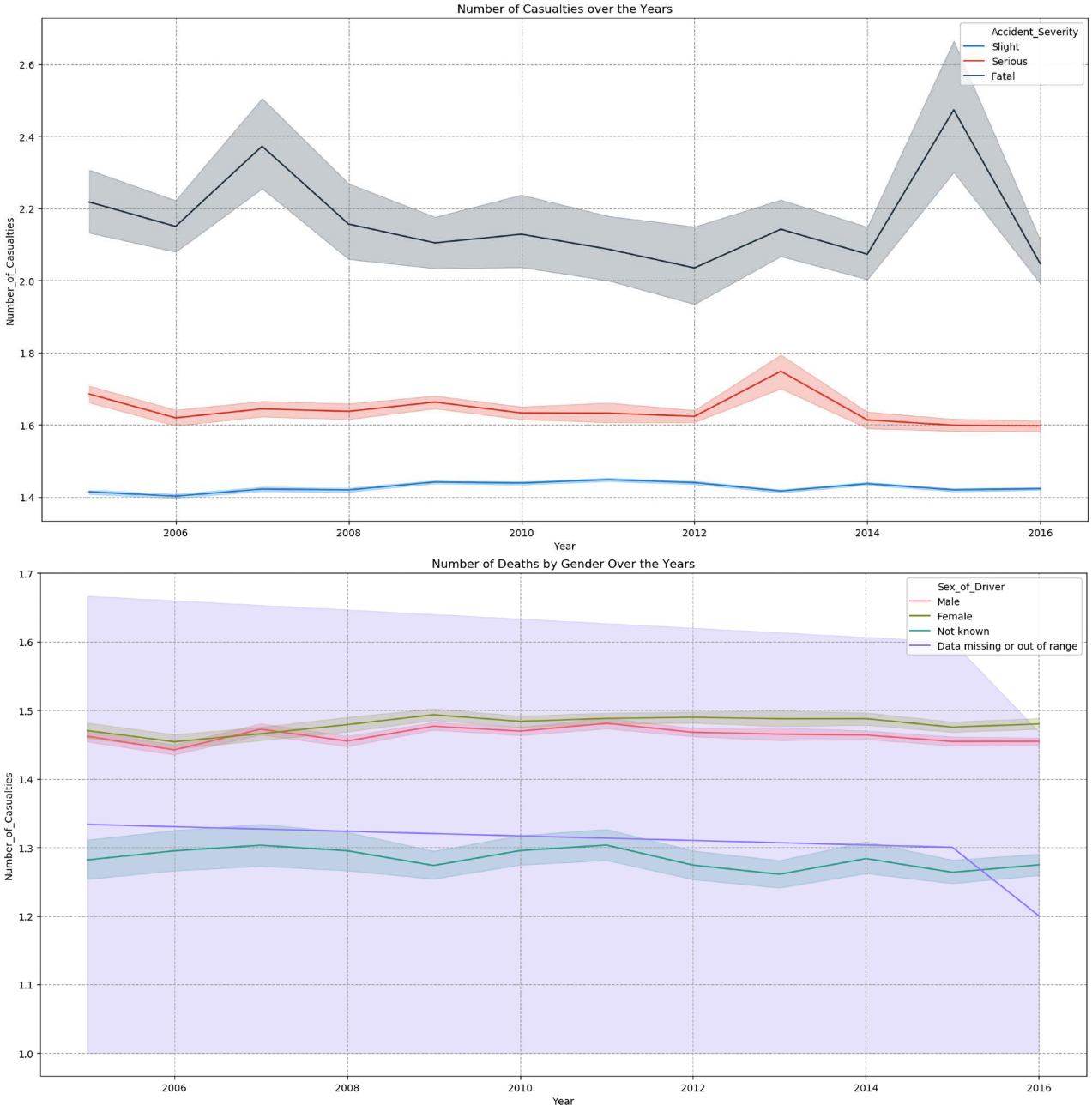

Figure A7. Line graphs generated using Python.

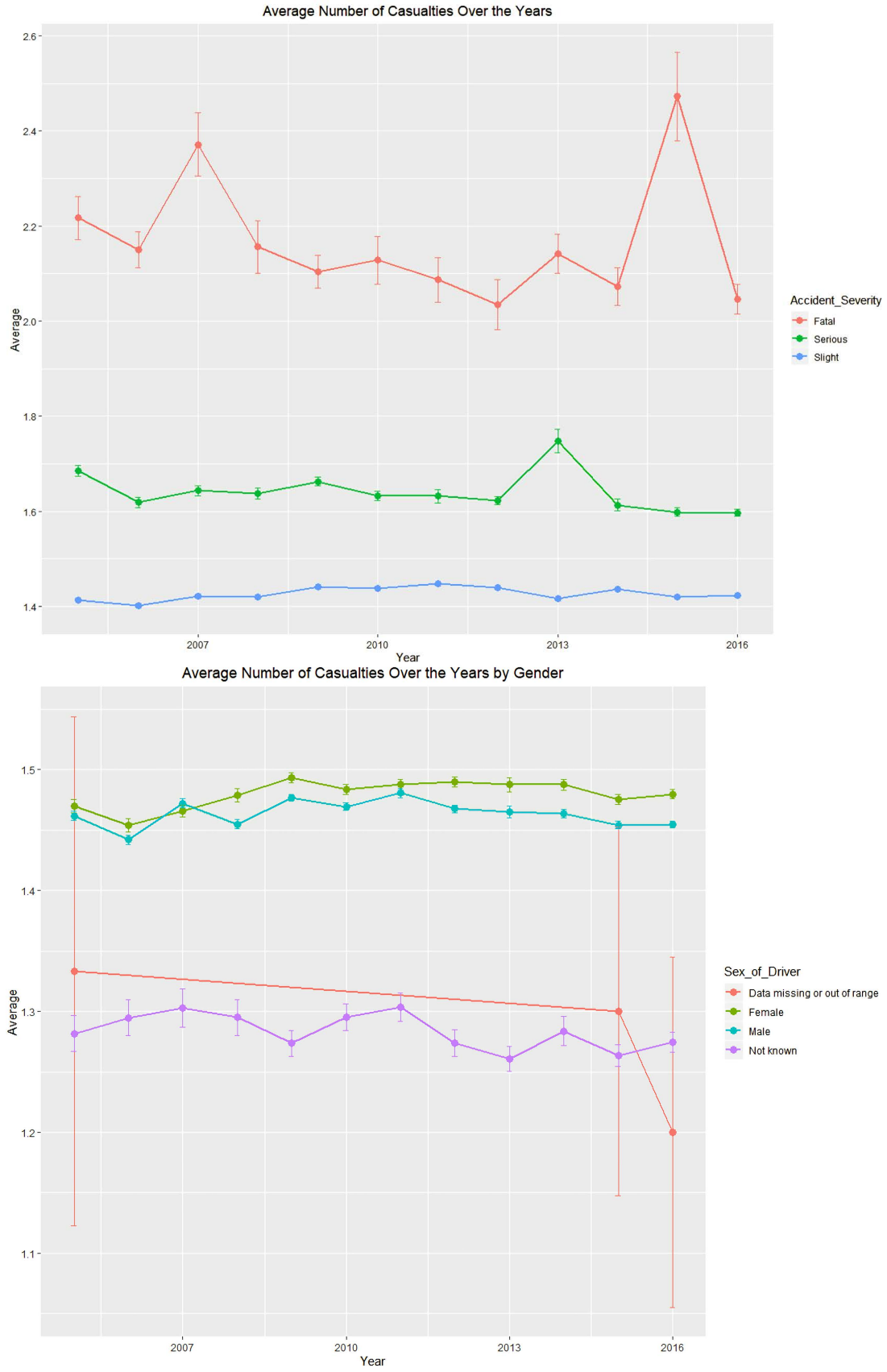

Figure A8. Line graphs generated using R.

```
64.24610209465027 seconds
svmem(total=8520081408,
available=4340695040,
percent=49.1, used=4179386368,
free=4340695040)
```

Figure A9. Time taken and memory used to perform the visualisations in Python.

```
   user  system elapsed
  44.79    2.21   48.17
         used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  982385  52.5    2904288 155.2  3500004 187.0
Vcells 35485354 270.8   88898803 678.3 88898803 678.3
```

Figure A10. Time Taken and memory used to perform the visualisations in R.

Index	Predicted: 0	Predicted: 1	Predicted: 2
Actual: 0	1	241	4253
Actual: 1	6	767	42639
Actual: 2	2	699	292220

Figure A11. Prediction results produced by ANN in Python.

```
34.635552167892456 seconds
Memory Consumption
 svmem(total=8520081408,
available=2357723136,
percent=72.3, used=6162358272,
free=2357723136)
```

Figure A12. Time taken and memory consumed in Python to train and produce the results of ANN.

```
          Actual
Predicted      0      1      2
        0      1      1      2
        1     72    150    144
        2   4424  43303 292793
```

Figure A13. Prediction results produced by ANN in R.

```
   user  system elapsed
  52.78    2.92   59.14
         used  (Mb) gc trigger   (Mb)  max used   (Mb)
Ncells 2115646 113.0    5854660  312.7   5854660  312.7
Vcells 110559567 843.6  267081536 2037.7 266521203 2033.4
```

Figure A14. Time taken and memory consumed in R to train and produce results by ANN.