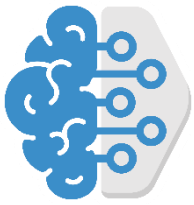


ASSESSMENT 2 REPORT
CALDERDALE ACCIDENT INFO

Ryan Needham

19156892

Max word count: 1350/1500/1650

Word count(not including cover page, contents, and code): 1628

Contents

BIRMINGHAM CITY University

Birmingham City University
Computer and Data Science

Introduction

The following report is based on the examination and manipulation of a dataset concerning road traffic collisions(RTCs) that were based within the Calderdale area. This report was sourced from the Calderdale Council and is available on 'https://dataworks.calderdale.gov.uk/dataset/calderdale-accident-data-'. The exploration/wrangling code will be labelled in a file named '19156892_assessment2.R', along with the CSV files of cleaned data and the regression predictions denoted as 'regression.csv' and 'cleanedSet.csv'.

Data Wrangling

Examining the columns

In this dataset there are 14 columns relating to RTCs in the Calderdale area. Some examples being 'Road Surface', 'Weather Conditions' and 'Lighting Conditions'; it can be acknowledged that these variables pertain to the conditions in which an accident happen. Similarly, we have variables such as 'Casualty Severity' and 'Age of Casualty' that relate to the characteristics of the people involved.

Missing Data

From looking at this dataset and using a little bit of code, I managed to find 37 missing values. All the missing values are within two different columns, 'age of casualty' and 'daylight/dark'. There are 19 missing values within 'age of casualty' and 18 missing values in 'daylight/dark'. The first step to identifying all the missing values was loading in the dataset appropriately with the following piece of code:

```
accident <- read.csv("data/accidents.csv",na.strings=c("","  ",NA))
```

This line of code allowed the dataset to be read in a manner that turned all possible missing values into 'NA'. Without this piece of code, it would only show 19 missing values in the 'age of casualty' column. When examining the count of the missing values I would then use these few lines of code:

```
na_data <- accident[!complete.cases(accident),]"
"count(na_data)"
```

If we were to look at the missing values that occurred in the 'age of casualty' column we can draw some conclusions on possible MNARs(Missing Not at Random) and MARs(Missing at Random).

One reason for a MNAR may be that a driver does not want to disclose their age. This could be due to them being elderly and not wanting to be associated with the typical stereotype linked with elderly drivers.

If we look at the 806th row of the dataset there are 4 vehicles involved in an accident on the motorway. This can be noted as a major road traffic collision and seeing as that it may be a chaotic scene, it is feasible that some of the drivers may have left before services could record their age. This would be an example of data being MAR.

Anomalies & Inconsistencies

The dataset given contained numerical values and characters values in both the 'Road Surface' and '1st road class' columns. To solve this issue, I began by writing some lines of code that converted the

numerical values into the correct character value. The following bit of code identified if there were any values emitted that were not a part of the 'Guidance.csv':

```
accident %>%
  group_by(Road.Surface) %>%
  summarise(.groups = 'drop')
#group the data by road service and find all the categorical types
```

Similarly, this works for '1st Road Class' column

```
accident %>%
  group_by(X1st.Road.Class) %>%
  summarise(.groups = 'drop')
```

Removing unnecessary columns and Rows

When looking through the variables(columns) within the dataset there is one column that is not necessary to keep. This would be the column 'Local Authority', which simply tells you where the incident is occurring. However, since we are looking at data within the same area(Calderdale), every observation(RTC) will have the value 'Calderdale' shown here:

```
accident %>%
  group_by(Local.Authority) %>%
  summarise(.groups = 'drop')
```

The output of this code states that there is only one distinct value throughout the whole column.

```
# A tibble: 1 x 1
  Local.Authority
  <chr>
1 Calderdale
```

Furthermore, I would then need to remove unnecessary rows, such as duplicated rows. For this I used the distinct function and found 2053 distinct observations. This would mean that I had removed 16 duplicated observations from the original 2069 observations. The code is shown below:

```
count(distinct(removed))
cleaner_data <-
distinct(removed)
#get all the distinct
variables from the removed
data and put them into the
cleaned dataset
```

Identifying outliers in the 'Age of Casualty' column

In identifying outliers there are three main methods, the 'boxplot', '3 sigma

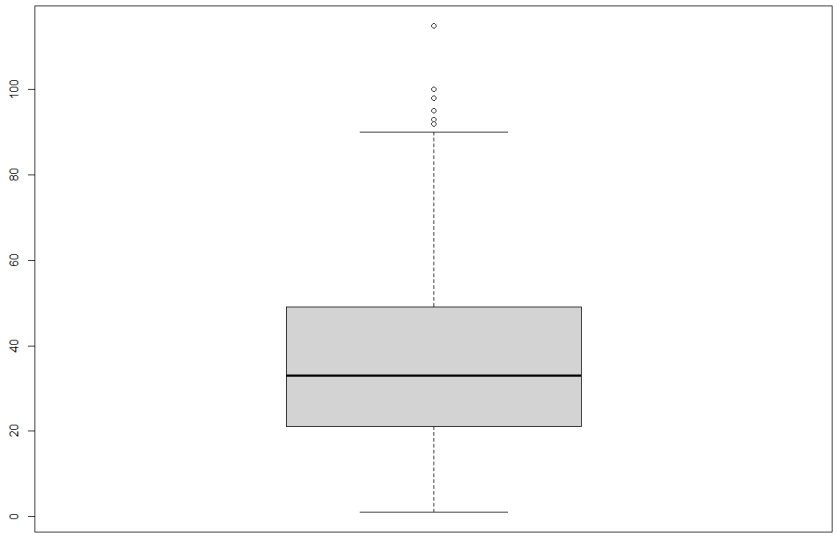


Figure 1

rule' and the 'Hampel Identifier'. The boxplot is applied in this project as shown below:

```
boxplot.stats(cleaner_data$Age.of.Casualty)$out
```

The output here shows that there are 7 outliers:
115,93,93,100,92,95,98

Another method to find outliers is the '3 sigma rule' Below is the following code to find outliers using this method:

```
#obtain standard deviation and mean
sd <- sd(cleaner_data$Age.of.Casualty, na.rm = TRUE)
mean <- mean(cleaner_data$Age.of.Casualty, na.rm = TRUE)
#get the upper and lower bound
upper_bound <- mean + (3*sd)
lower_bound <- mean - (3*sd)

outliers_sigma <-cleaner_data %>%
  filter((Age.of.Casualty > upper_bound)|
           (Age.of.Casualty < lower_bound))
#display the outliers
outliers_sigma$Age.of.Casualty
```

This method yielded 3 outliers that were:
115,100,98

The Hampel Identifier is next, here is the code below:

```
median_value <- median(cleaner_data$Age.of.Casualty, na.rm = TRUE)
MAD_value <- mad (cleaner_data$Age.of.Casualty, na.rm = TRUE)

upper_bound2 <- median_value + (3*MAD_value)
lower_bound2 <- median_value - (3*MAD_value)

outliers_sigma2 <-cleaner_data %>%
  filter((Age.of.Casualty > upper_bound2)|
           (Age.of.Casualty < lower_bound2))

outliers_sigma2$Age.of.Casualty
```

This method yielded 7 outliers the exact same as the boxplot method:
115,93,93,100,92,95,98

After looking at all the methods I have decided to go with the Hampel Identifier because it shares the same result as the boxplot method. In addition, unlike the 3-sigma rule it does not use the mean; this is beneficial because if you include the mean in the equation it also calculated the outliers themselves. I have now cleaned the dataset further by removing the outliers.

Data Exploration

Weather conditions and their effects on drivers of different genders

The question that is posed to us is: 'Are there any weather conditions where male drivers/riders have more accidents than female drivers?'.

To visualize this, I made a bar graph showing the percentages of all collisions by weather condition and gender:

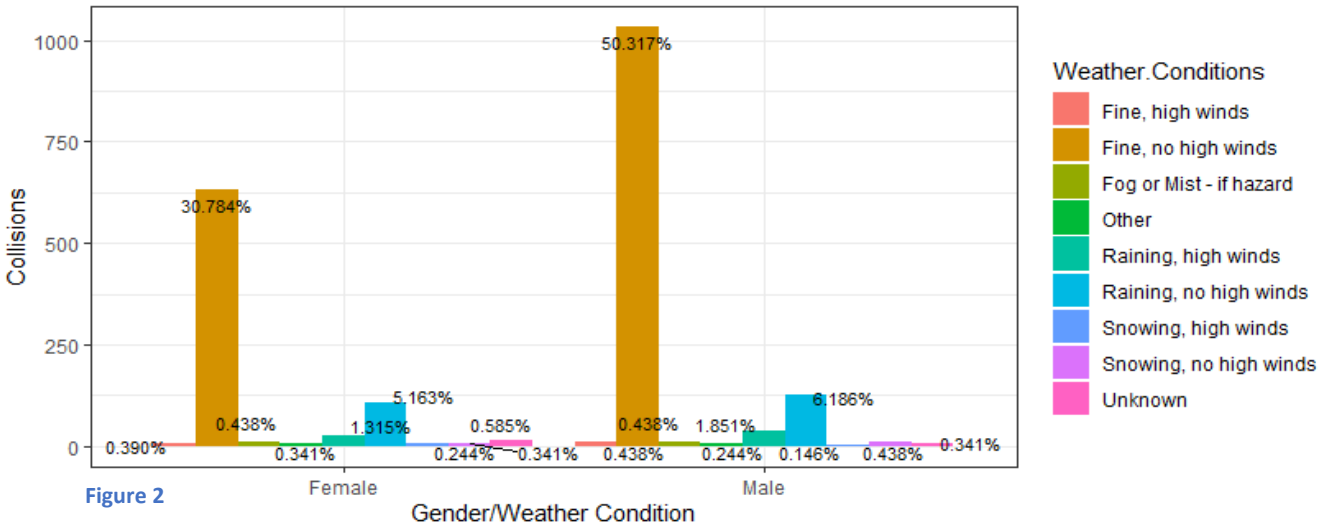


Figure 2

Percentages of all collisions by weather condition and gender.

Gender	Fine, High Winds	Fine, No High Winds	Fog or Mist if Hazard	Other	Raining, High Winds	Raining, No High Winds	Snowing, High Winds	Snowing, No High Winds	Unknown
Male	0.4%	50.3%	0.4%	0.2%	1.9%	6.2%	0.2%	0.4%	0.3%
Female	0.4%	30.8%	0.4%	0.3%	1.3%	5.2%	0.2%	0.3%	0.6%

From the observation of the outputs in the table and figure 1, we can say that male drivers have a higher percentage of accidents in 4 different weather conditions whereas women have a higher percentage in 2 different weather conditions, the remaining weather conditions are the same.

Casualty Numbers on a year-by-year basis

The dataset that we are using does not give an exact number of casualties for each RTC, so to solve this problem we will assume that each observation(accident) will have one casualty. To create the correct graph and use the 'Accident Date' column, I would need to change the data type in 'Accident Date' from as factor to a date data type.

```
newdate <- accident %>%
  select(Accident.Date) %>%
  mutate(as.Date(Accident.Date,format="%d/%m/%Y"))
#mutate all the cells with dates into a date type
```

To visualise the data, I used a line graph that would show the casualties of each year:

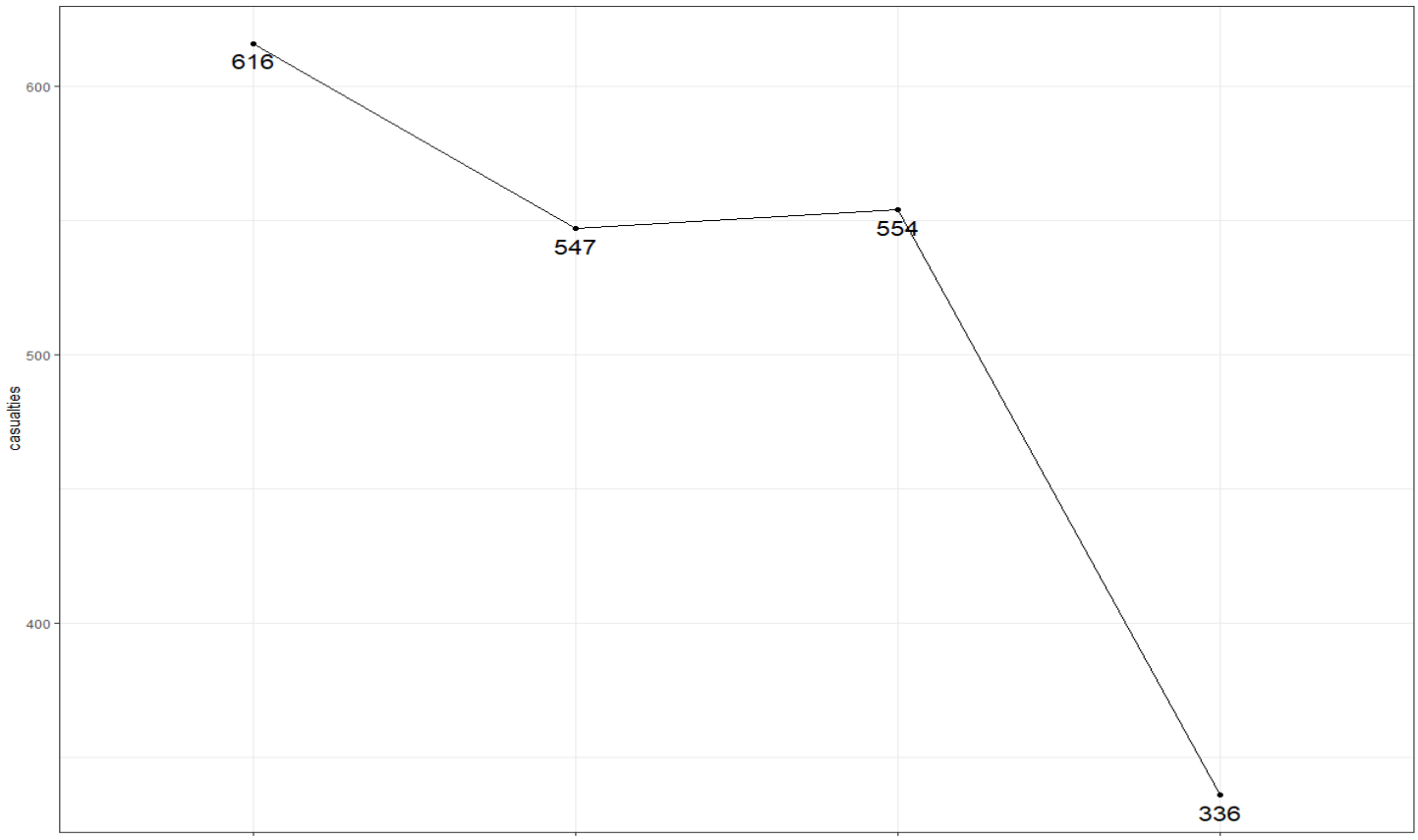


Figure 3

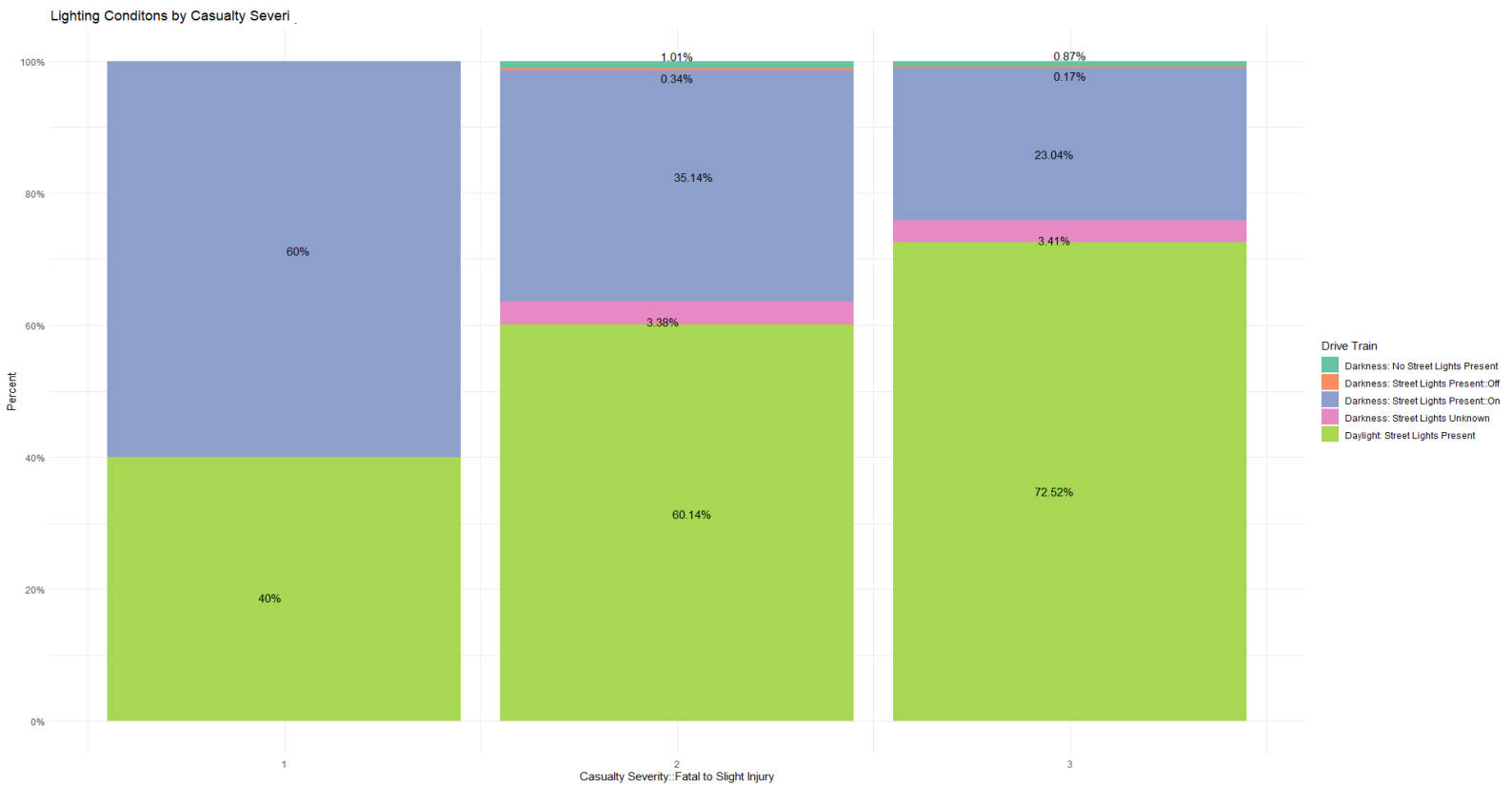
From this chart we can identify that there is a decrease in casualties through most of the years. The year with the highest number of accidents is '2014' with 616 accidents. This data shows that '2014' accounts for ~30% of the accidents in the dataset.

Exploring the relationship between light condition and the severity of casualty's condition

To find the relationship between light condition and severity of casualty, I decided that an ideal graph would be a stacked bar chart, that shows the percentage of accidents within each light condition and is also sorted by casualty severity.

Graph:

Figure 4



By observation 5 out of the possible 7 lighting conditions are shown as these were the only lighting conditions prevalent in the dataset. As for the relationship we can see that the most prominent lighting condition was 'Daylight: Street Lights Present', this is most likely due to most roads having streetlights present. In addition, the more fatal a severity the less lighting conditions are graphed, this is due to the more fatal accidents being a rarer occurrence. Furthermore, it can be said that if an injury is more fatal, it is more likely to have occurred during darkness, as ~60% of fatal injuries occurred then.

Exploring the relationship between weather conditions and the number of vehicles involved in a RTC

To find a relationship between number of vehicles and weather conditions I was stuck between two different graphs, one shows the average number of vehicles against the weather conditions, and the other shows the frequency of accidents, with weather conditions on the x-axis and number of vehicles on the y-axis in a bubble chart. Both graphs will be included in a separate file but for this report I will show the average number of vehicles in a plot as it will help pull out a more accurate result.

Graph:

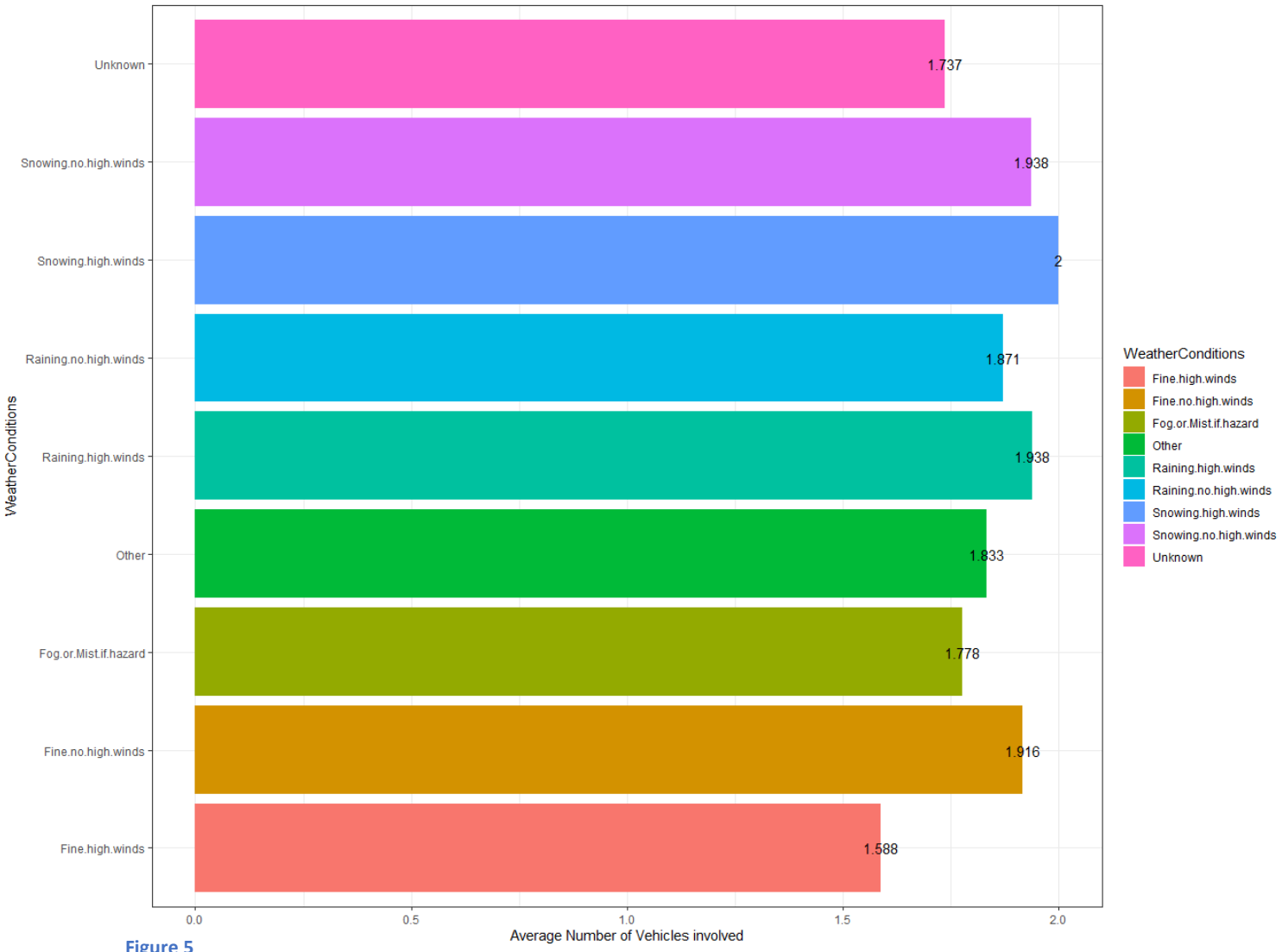


Figure 5

As you can see the weather condition with the highest average number of vehicles involved in an accident is 'Snowing, high winds', and joint second is 'Snowing, no high winds' with 'Raining, high winds. This can echo that more dangerous weather conditions will cause more vehicles to be involved in a single accident.

Regression

Training and imputing new values using linear regression

In this case we will be training the columns 'casualty class', 'casualty severity', 'type of vehicles' and 'weather condition'. We will be using linear regression and these variables to take the missing values from the 'age of casualty' column and impute them with the predicted output from the linear regression model. Firstly, we will split the data into a training and test set. Here's the code:

```
regression <- cleaner_data
Casualty.Class <- cleaner_data$Casualty.Class
Casualty.Severity<- cleaner_data$Casualty.Severity
Type.of.Vehicle <- cleaner_data$Type.of.Vehicle
Weather.Conditions <- regression$Weather.Conditions
Age.of.Casualty <- cleaner_data$Age.of.Casualty
```

```
Weather.Conditions <- as.numeric(Weather.Conditions)

#The function will take a vector as an argument and returns a number
#It will return 0 if it finds a missing value and 1 if it finds a
known value.
regression <-
data.frame(Casualty.Class,Age.of.Casualty,Weather.Conditions,Type.of
.Vehicle,Casualty.Severity)
missDummy <- function(t)
{
x <- dim(length(t))
x[which(!is.na(t))] = 1
x[which(is.na(t))] = 0
return(x)
}
regression$dummy <- missDummy(regression$Age.of.Casualty)
# Choose the values with known y values as training data
TrainData<- regression [regression ['dummy']==1,]
# Choose the missing values with(NA) y values as testing data
TestData<- regression [regression ['dummy']==0,]
#create the linear model with the formula to predict 'age of
casualty'
model<-
lm(Age.of.Casualty~Casualty.Severity+Casualty.Class+Weather.Conditio
ns+Type.of.Vehicle, TrainData)
#put values in an object
pred<- predict(model, TestData)
#show predicted values
pred

#round the decimal
pred <- round(pred)
regression$Age.of.Casualty[is.na(Age.of.Casualty)]
#input values into regression model
regression$Age.of.Casualty[is.na(Age.of.Casualty)]<- pred
#delete dummy column from the regression dataset
regression <- regression[-c(6)]
write.csv(regression,'regression.csv')
```

The model was able to use the linear regression formula to predict missing values in the 'Age of Casualty' column, some problems that I ran into originally were that the values in other variables being used to predict the missing values were not numeric. Using a linear regression model with categorical variables is a very difficult task so I simply went and changed the values to a numeric one.

Conclusion

In conclusion we can first identify that all the crashes in Calderdale were steadily on the decline, we could also say that male drivers were typically the cause of most of the crashes, especially during fine, no winds. Secondly most of the accidents were only slight injuries, with more fatal injuries being occurred in darkness ~60% of the time. In terms of weather conditions, the average number of total vehicles involved in an accident would rise if it occurred during snowy or raining weather. Even though the initial dataset was included with missing values and an inconsistent character or numeric value entered some observations, it was able to be cleaned and later have imputed values into the missing cells.