Calderdale Accident Casualty Info 2020/2021

Gregory Sampson - S19156605

Contents

Introduction

The data we have been given includes statistics concerning road traffic collisions within the Calderdale area. Calderdale Council made this available at 'https://dataworks.calderdale.gov.uk/dataset/calderdale-accident-data-'. While the best version of the report & code is in this report, a raw form of all R code is included in file "19156605_assessment2.R" and also the CSV files of the cleaned data and the regression predictions are provided as "regression.csv" and "cleanedSet.csv".

Data Wrangling

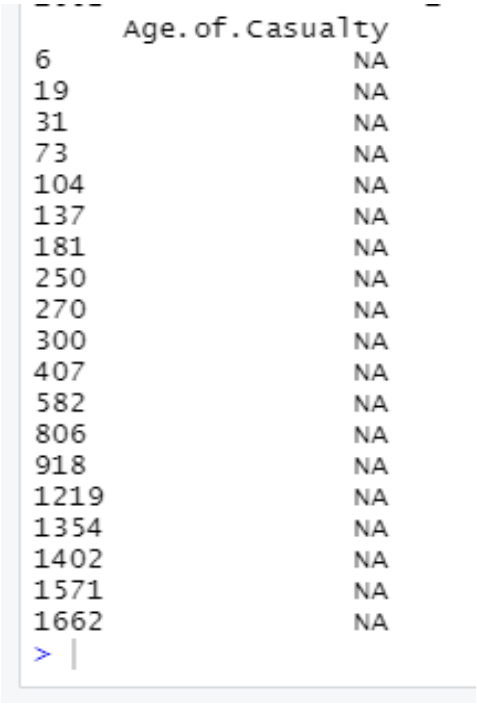
Examining the columns in the data

The data set has 14 columns, some such as 'Road Surface', 'Lighting Conditions' and 'Weather Conditions' convey the conditions during the accident and some such as 'Casualty Severity' and 'Age of Casualty' help display any characteristics of the people involved.

Missing data

In this dataset there are in total 19 missing values:

Now these values can all be found in the same column, age. There are 19 cases where the age of the casualty involved was either not discovered or disclosed by the person involved. You can see these 19 cases here:



	Age.of.Casualty
6	NA
19	NA
31	NA
73	NA
104	NA
137	NA
181	NA
250	NA
270	NA
300	NA
407	NA
582	NA
806	NA
918	NA
1219	NA
1354	NA
1402	NA
1571	NA
1662	NA

>

One example of MNAR(Missing Not At Random) could be due to the candidate not wanting to provide their age as they may be younger and did not want to be seen as a reckless driver due to their age and existing stereotypes around younger (mostly male) drivers being more reckless and driving quickly, this could be the case for row 6.

Number of Accident	Date	Time (24hr)	1st Road Class	Road Surface	Lighting Conditions	Daylight/Dark	Weather Conditions	Local Authority	Type of Vehicle	Casualty Class	Casualty Severity	Sex of Casualty	Age of Casualty
1	15/01/2017	1659	U	Wet/Damp	4	Dark	1	Calderdale	9	3	3	1	

An example of MAR (Missing At Random) can be found by looking at row 806, there is a total of 4 vehicles that are in the accident with a low casualty score which could lead to us believing that the culprit had left the sight of the accident before any responders could arrive which would leave the age blank as no age could be taken.

4	27/10/2016	1735	1	1	1	Daylight	1	Calderdale	9	2	3	1	

Anomalies & Inconsistencies

In the ‘Guidance.csv’ there are different tables that translate what some of the values in the data mean in real world terms such as 1 meaning the road type ‘motorway’, within this table are many different tables and such examples are:

	A	B	C	D	E
1	1st Road Class		1st Road Class Desc		
2		1	Motorway		
3		2	A(M)		
4		3	A		
5		4	B		
6		5	C		
7		6	Unclassified		
8					
9	Road Surface		Road Surface Desc		
10		1	Dry		
11		2	Wet / Damp		
12		3	Snow		
13		4	Frost / Ice		
14		5	Flood (surface water over 3cm deep)		
15					
16	Lighting Conditions		Lighting Conditions Desc		
17		1	Daylight: street lights present		
18		2	Daylight: no street lighting		
19		3	Daylight: street lighting unknown		
20		4	Darkness: street lights present and lit		
21		5	Darkness: street lights present but unlit		
22		6	Darkness: no street lighting		
23		7	Darkness: street lighting unknown		

By, looking at the different values that have been entered for this we can see that not every record has the equivalent text for the numeric value and instead uses the number value:

When looking through all the entries in the dataset, we can see that 1733 of the rows have the road surface and type entered as a numeric value.

```
> sum(accidents$FirstRoadClass==1|accidents$FirstRoadClass==2|accidents$FirstRoadClass==3|accidents$FirstRoadClass==4|accidents$FirstRoadClass==5|accidents$FirstRoadClass==6)
[1] 1733
> sum(accidents$RoadSurface==1|accidents$RoadSurface==2|accidents$RoadSurface==3|accidents$RoadSurface==4|accidents$RoadSurface==5|accidents$RoadSurface==6)
[1] 1733
>
```

Now that we are aware of these issues, we can fix them as they still hold valid data. To do this we will replace the numeric value with the correct correlating text value. Here is my example of this process being done on the column ‘1stRoadClass’ and ‘RoadSurface’ also changing the `WeatherConditions` for later use in the graphsx:

3	28/12/2017	2020	A58	Frost/Ice		4	Dark	1	Calderdale	9	1	2	1	24
2	31/12/2017	1331	U	Dry		1	Daylight	1	Calderdale	9	1	3	1	28
1	01/01/2016	52	3	2		4	Dark	1	Calderdale	8	3	3	1	27
1	01/01/2016	1204	6	2		1	Daylight	1	Calderdale	9	1	3	2	87
2	01/01/2016	1400	6	1		1	Daylight	1	Calderdale	9	1	3	1	23

```
accidents$FirstRoadClass <- replace(accidents$FirstRoadClass, accidents$FirstRoadClass == 1, "motorway")
accidents$FirstRoadClass <- replace(accidents$FirstRoadClass, accidents$FirstRoadClass == 2, "A(m)")
accidents$FirstRoadClass <- replace(accidents$FirstRoadClass, accidents$FirstRoadClass == 3, "A")
accidents$FirstRoadClass <- replace(accidents$FirstRoadClass, accidents$FirstRoadClass == 4, "B")
accidents$FirstRoadClass <- replace(accidents$FirstRoadClass, accidents$FirstRoadClass == 5, "C")
accidents$FirstRoadClass <- replace(accidents$FirstRoadClass, accidents$FirstRoadClass == 6, "Unclassified")


accidents$RoadSurface <- replace(accidents$RoadSurface, accidents$RoadSurface == 1, "dry")
accidents$RoadSurface <- replace(accidents$RoadSurface, accidents$RoadSurface == 2, "Wet/Damp")
accidents$RoadSurface <- replace(accidents$RoadSurface, accidents$RoadSurface == "wet", "Wet/Damp")
accidents$RoadSurface <- replace(accidents$RoadSurface, accidents$RoadSurface == 3, "Snow")
accidents$RoadSurface <- replace(accidents$RoadSurface, accidents$RoadSurface == 4, "Frost/Ice")
accidents$RoadSurface <- replace(accidents$RoadSurface, accidents$RoadSurface == 5, "Flood(surface water over 3cm deep)")


accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 1, "Fine without high winds")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 2, "Raining without high winds")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 3, "Snowing without high winds")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 4, "Fine with high winds")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 5, "Raining with high winds")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 6, "Snowing with high winds")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 7, "Fog or mist – if hazard")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 8, "Other")
accidents$WeatherConditions <- replace(accidents$WeatherConditions, accidents$WeatherConditions == 9, "Unknown")
```

I then test the document to see if my changes have been made:

```
> sum(accidents$FirstRoadClass==1|accidents$FirstRoadClass==2|accidents$FirstRoadClass==3|accidents$FirstRoadClass==4|accidents$FirstRoadClass==5|accidents$FirstRoadClass==6)
[1] 0
> sum(accidents$RoadSurface==1|accidents$RoadSurface==2|accidents$RoadSurface==3|accidents$RoadSurface==4|accidents$RoadSurface==5|accidents$RoadSurface==6)
[1] 0
```

Removing unnecessary columns

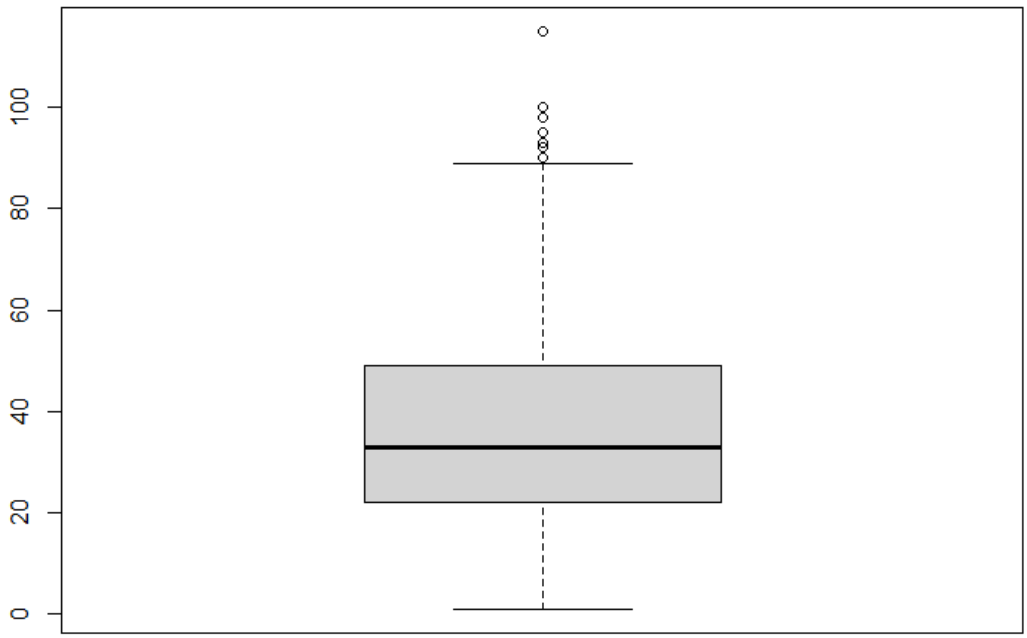
Looking through all the columns, we can see that the 'Local.Authority' has a repeated variable throughout it. This is because all the tests were taken within a local area that this authority has domain over, each value is therefore set as Calderdale Council. As evidenced here you can see the code used to remove it and the proof of removal as it was previously between weather conditions and type of vehicle:

```
34
35 accidents<-select(accidents, -c(LocalAuthority))
36 accidents
37
```

```
Daylight.Dark WeatherConditions TypeOfVehicle CasualtyClass CasualtySeverity
```

Identifying outliers in the 'Age of Casualty' column

There are three main methods in order identify outliers in a set of data. The first one is to mark the data out on a boxplot., this method shows me 7 outliers:



The outliers values are as follows:

```
38  boxplot(accidents$AgeOfCasualty)
39  outliers <- boxplot(accidents$AgeOfCasualty, plot=FALSE)$out
40  outliers
41
```

41:1 (Top Level)

```
Console C:/Users/fredr/OneDrive/HS NOW/Assessment_2/
79                  2   05/04/2017 1913           A672          Dry
     1                  3                1             34
80                  1   07/04/2017   22             M62          Dry
     1                  3                1             52
81                  1   07/04/2017 1317               U          Dry
     2                  3                2              9
82                  2   07/04/2017    0           B6113          Dry
     1                  3                1             68
 [ reached 'max' / getOption("max.print") -- omitted 1940 rows ]
> boxplot(accidents$AgeOfCasualty)
> outliers <- boxplot(accidents$AgeOfCasualty, plot=FALSE)$out
> outliers
 [1] 115  93  90  93 100  90  92  95  98  90
>
```

We can also find outliers using the 3 Sigma rule. Here is the code I used to find the outliers along with the output of the 3 outliers found:

```
43
44  sd_value <- sd(accidents$AgeOfCasualty, na.rm = TRUE)
45  mean_value <- mean(accidents$AgeOfCasualty, na.rm = TRUE)
46  upper_bound <- mean_value+3*sd_value
47  lower_bound <- mean_value-3*sd_value
48
49  outliers_sigma <-accidents %>% filter((AgeOfCasualty > upper_bound)| (AgeOfCasualty < lower_bound))
50  outliers_sigma
51
52  median_value <- median(accidents$AgeOfCasualty, na.rm = TRUE)
53  MAD_value <- mad (accidents$AgeOfCasualty, na.rm = TRUE)
54
55  upper_bound <- median_value+3*MAD_value
56  lower_bound <- median_value-3*MAD_value
57
58  outliers_hampel <- accidents %>% filter((AgeOfCasualty > upper_bound)| (AgeOfCasualty < lower_bound))
59  outliers_hampel
60
61
```

52:1 (Top Level)

```
Console C:/Users/fredr/OneDrive/HS NOW/Assessment_2/
4                 3             1           100
5                 2             2            92
6                 2             1            95
7                 3             2            98
> outliers_sigma
  NumberOfVehicles AccidentDate Time FirstRoadClass RoadSurface LightingConditions Daylight.Dark WeatherConditions TypeOfVehicle CasualtyClass
1                2   31/01/2017 1840              U    Wet/Damp                  5          Dark                  2             9            1
2                1   16/06/2016 1215   Unclassified         dry                  1      Daylight                  1             9            3
3                2   08/06/2014 1650              A         dry                  1      Daylight                  1             9            2
  CasualtySeverity SexOfCasualty AgeOfCasualty
1                2             2           115
2                3             1           100
3                3             2            98
>
```

Finally, I will be trying the Hampel method, here is the code I have used to find the outliers with this method:

```
51
52 median_value <- median(accidents$AgeOfCasualty, na.rm = TRUE)
53 MAD_value <- mad (accidents$AgeOfCasualty, na.rm = TRUE)
54
55 upper_bound <- median_value+3*MAD_value
56 lower_bound <- median_value-3*MAD_value
57
58 outliers_hampel <- accidents %>% filter((AgeOfCasualty > upper_bound)| (AgeOfCasualty < lower_bound))
59 outliers_hampel
60
61
```

52:1 (Top Level)

Console C:/Users/fredr/OneDrive/HS NOW/Assessment_2/

	NumberOfVehicles	AccidentDate	Time	FirstRoadClass	RoadSurface	LightingConditions	Daylight.Dark	WeatherConditions	TypeOfVehicle	CasualtyClass
1	2	31/01/2017	1840	U	Wet/Damp	5	Dark	2	9	1
2	1	24/06/2017	1200	A646	Dry	1	Daylight	1	9	3
3	1	19/03/2016	1902	Unclassified	dry	1	Daylight	1	9	3
4	1	16/06/2016	1215	Unclassified	dry	1	Daylight	1	9	3
5	1	28/06/2016	1627	A	dry	1	Daylight	1	9	3
6	1	05/02/2014	1715	Unclassified	Wet/Damp	4	Dark	1	8	3
7	2	08/06/2014	1650	A	dry	1	Daylight	1	9	2

	CasualtySeverity	SexOfCasualty	AgeOfCasualty
1	2	2	115
2	2	1	93
3	2	1	93
4	3	1	100
5	2	2	92
6	2	1	95

I have decided to use the Hampel method as it produces the same result as the box plot. Once I remove the outliers the data can be considered as “cleaned” giving us a dataset to analyze and report on in the next section of the report.

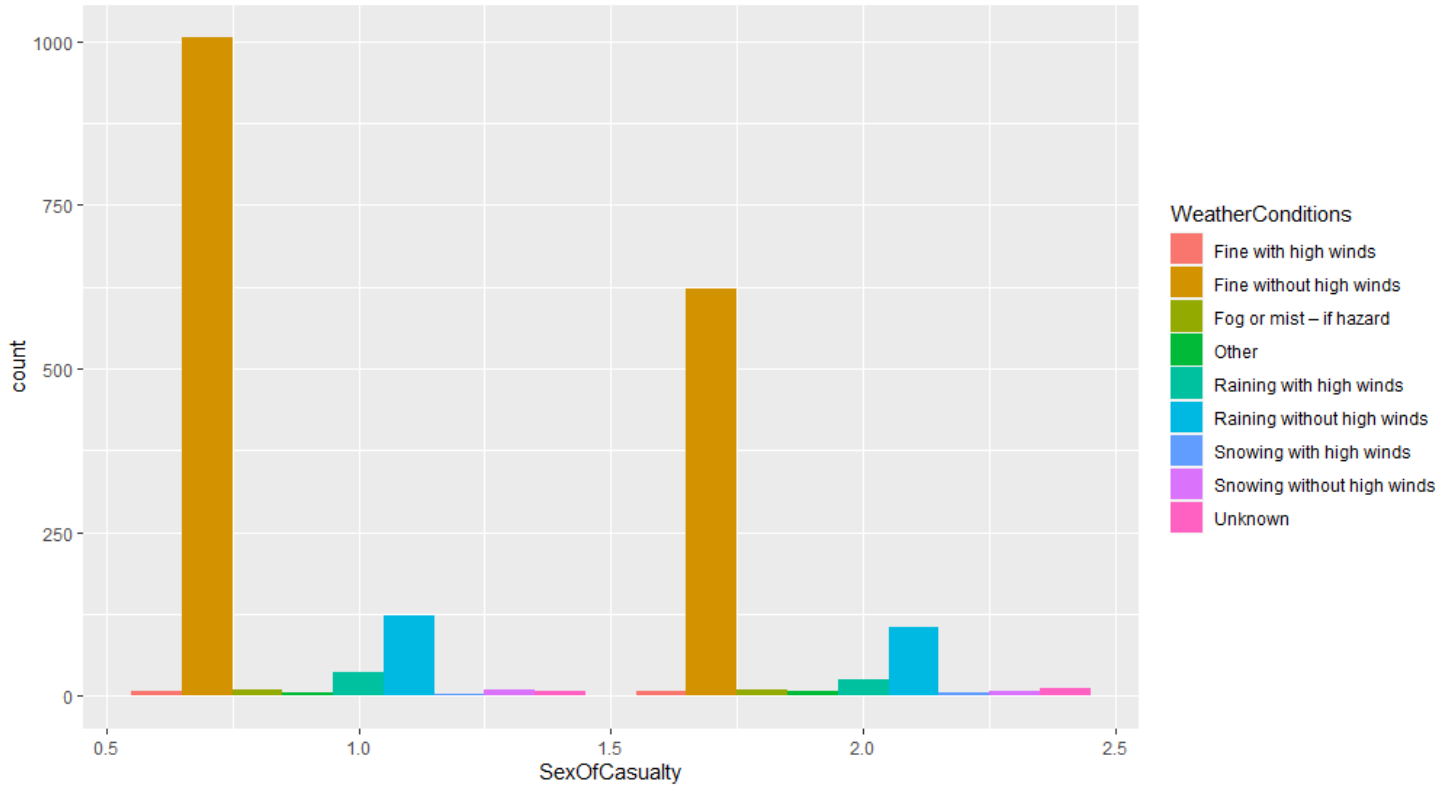
Data Exploration

Weather conditions and their effects on drivers of different genders

The question given is: **"Is there any weather condition where male drivers/riders have more accidents than female drivers?".**

I have created a bar graph to help visualize and differences in the amount of accidents caused by a specific gender:

```
ggplot(accidents_new, aes(SexOfCasualty, fill=WeatherConditions )) +
  geom_bar(position="dodge")
```



As it is hard to see the real values here is a numeric comparison between the genders (I am unable to remove the large spaces between currently:

```
> print(output1)
[1] ""
> print(output2)
[1] "Males have "                    "17"                    " more accidents in the class 'Raining without high winds' "
> print(output3)
[1] "Males have "                    "2"                     " more accidents in the class 'Snowing without high winds' "
> print(output4)
[1] "Males have "                    "17"                    " more accidents in the class 'Fine with high winds' "
> print(output5)
[1] "Males have "                    "17"                    " more accidents in the class 'Raining with high winds' "
> print(output6)
[1] "Males have "                    "17"                    " more accidents in the class 'Snowing with high winds' "
> print(output7)
[1] "Males have "                    "17"                    " more accidents in the class 'Fog or mist - if hazard' "
> print(output8)
[1] "Males have "                    "17"                    " more accidents in the class 'Other' "
> print(output9)
[1] "Males have "                    "17"                    " more accidents in the class 'Unknown' "
>
```

Casualty Numbers on a year by year basis

Looking at the data there are no columns on number of casualties so we will say it is only one casualty per accident. We now need to group our data by year. I have converted the dates into just a year using code below and assigned 4 different data frames with number of dates that are in our dataset e.g. "2017 has 318 casualties"

```
159
160 x <-format(as.Date(accidents_new$AccidentDate, format="%d/%m/%Y"),"%Y")
161 y <- 0
162 z <- 0
163 n <- 0
164 f <- 0
165 other <- 0
166 for (i in x ){
167
168   if ( i == 2017) {
169     y <- y + 1
170
171   } else if ( i == 2016) {
172     z <- z + 1
173
174   }else if ( i == 2015) {
175     n <- n + 1
176
177   }else if ( i == 2014) {
178     f <- f + 1
179   }else{
180     other <- other + 1
181   }}
182 print(c("2017: ",y))
183 print(c("2016: ",z))
184 print(c("2015: ",n))
185 print(c("2014: ",f))
186 print(c("other: ",other))
187
188
189
```

85:21 (Top Level)

Console C:/Users/fredr/OneDrive/HS NOW/Assessment_2/

```
   }else if ( i == 2015) {
     n <- n + 1

   }else if ( i == 2014) {
     f <- f + 1
   }else{
     other <- other + 1
   }}
 print(c("2017: ",y))
1] "2017: " "318"
 print(c("2016: ",z))
1] "2016: " "544"
 print(c("2015: ",n))
1] "2015: " "537"
 print(c("2014: ",f))
1] "2014: " "607"
 print(c("other: ",other))
1] "other: " "0"
```

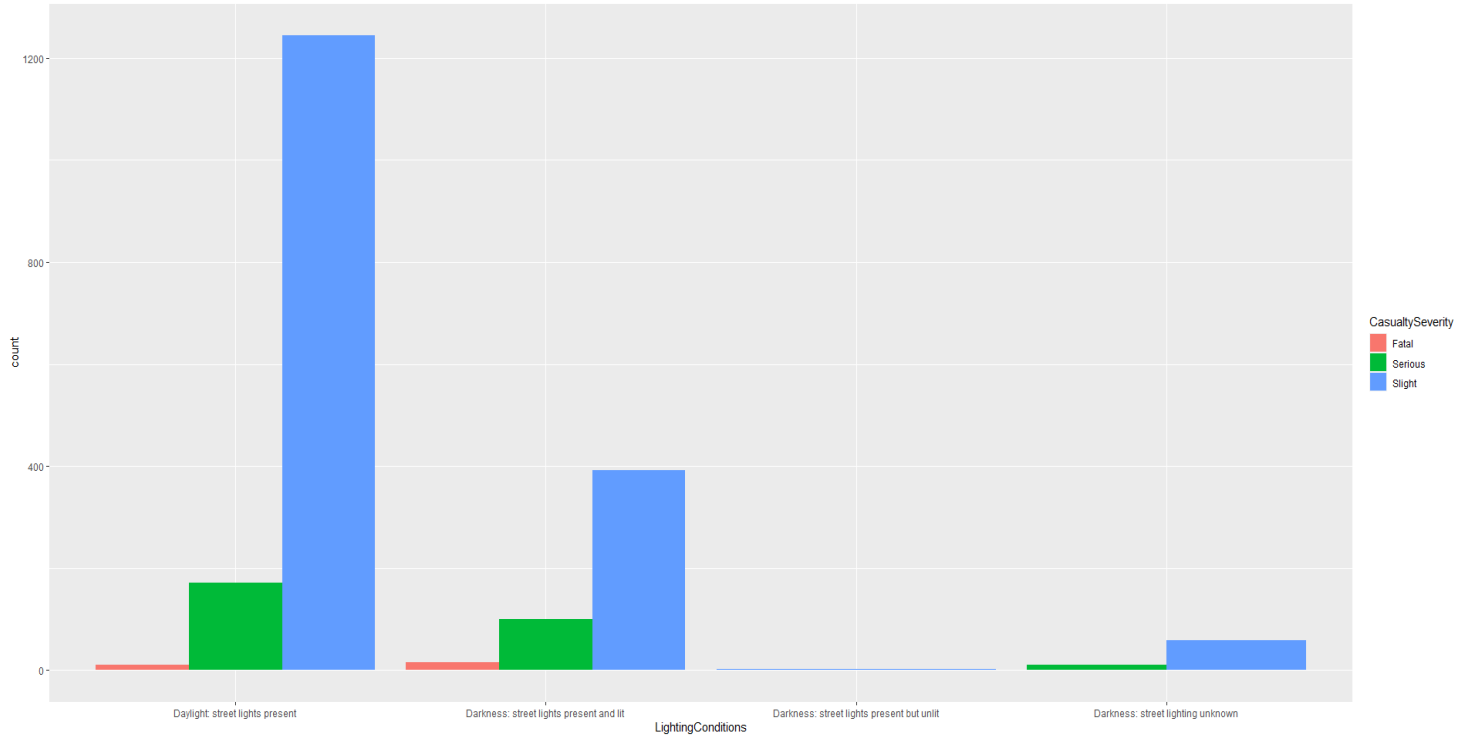
We can see that the number of accidents occurring is decreasing each year. 2014 has the highest casualties with a value of 607 accidents.

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

EDA, also known as exploratory data analysis, allows us to use many graphs to help find relationships between features in data. I have decided a bar chart is the easiest way to represent this severity as it allows us to easily see a comparison of the number of crashes between each severity class against each individual lighting condition after which I should be able to see how each condition affects the number of crashes as well as the severity of those crashes and which is most common.

Using GGplot I can create a visually pleasing bar chart for my work, the code used to make this is simple and will be shown below, I had to manipulate the data slightly to get it to produce the graph I wanted:

```
accidents_new$LightingConditions<-as.factor
accidents_new$LightingConditions<-factor(accidents_new$LightingConditions,
                                         levels=c(1,2,3,4,5,6,7),
                                         labels=c(
                                           "Daylight: street lights present",
                                           "Daylight: no street lighting",
                                           "Daylight: street lighting unknown",
                                           "Darkness: street lights present and lit",
                                           "Darkness: street lights present but unlit",
                                           "Darkness: no street lighting",
                                           "Darkness: street lighting unknown" ))
accidents_new$CasualtySeverity <-as.factor(accidents_new$CasualtySeverity )

accidents_new$CasualtySeverity<-factor(accidents_new$CasualtySeverity,
                                       levels=c(1,2,3),
                                       labels=c("Fatal",
                                                "Serious",
                                                "Slight" ))


ggplot(accidents_new, aes(LightingConditions, fill=CasualtySeverity )) +
  geom_bar(position="dodge")
```



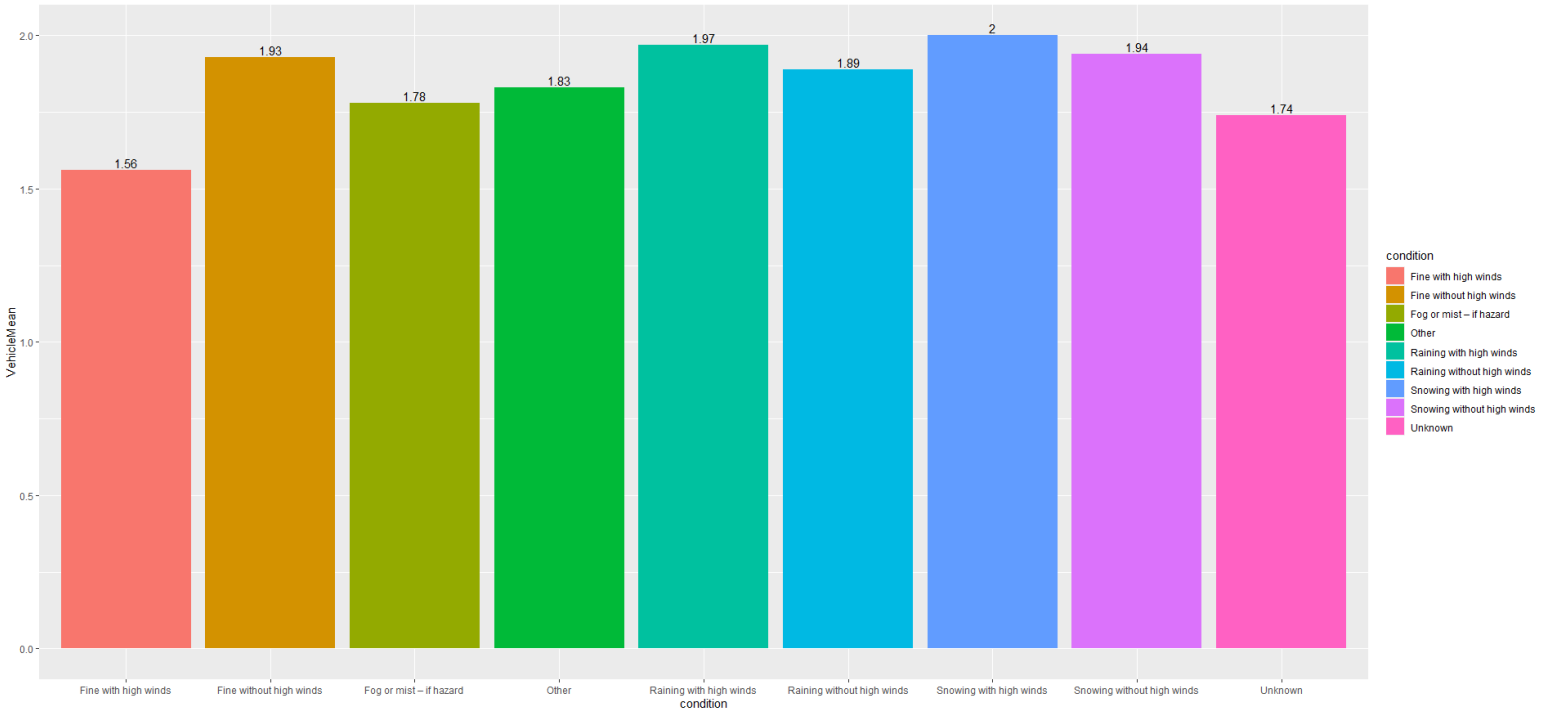
From this plot we can see how relationship between casualty severity and the lighting conditions. Within each lighting condition we are able to see the number of accidents within the level of severity occurred, the some conditions don't contain specific levels meaning they didn't have any casualties at this severity.

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

To explore this relationship I've decided again to use a bar chart, however this time I have provided the mean number on top of each bar to allow us to compare them in an easier way. This also helps to represent the mean number for the amount of cars involved in the crashes related to each condition.

This code required a lot more work as I had to work out the mean for each weather condition and then place them all into a new data frame to make it easier to use inside of the GGplot command. see below for the code to create the plot as well as the plot:

```
weather1 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions == "Fine without high winds")

weather2 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions == "Raining without high winds")

weather3 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions ==  "Snowing without high winds")

weather4 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions ==     "Fine with high winds")

weather5 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions ==         "Raining with high winds")

weather6 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions =="Snowing with high winds")

weather7 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions == "Fog or mist - if hazard")

weather8 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions == "Other")

weather9 <- accidents_new %>%
  group_by(NumberOfVehicles, WeatherConditions) %>%
  filter(WeatherConditions ==  "Unknown")


weather_1_m <-round((mean(weather1[["NumberOfVehicles"]])), digits = 2)
weather_2_m <-round(( mean(weather2[["NumberOfVehicles"]])), digits = 2)
weather_3_m <-round(( mean(weather3[["NumberOfVehicles"]])), digits = 2)
weather_4_m <-round(( mean(weather4[["NumberOfVehicles"]])), digits = 2)
weather_5_m <-round(( mean(weather5[["NumberOfVehicles"]])), digits = 2)
weather_6_m <-round(( mean(weather6[["NumberOfVehicles"]])), digits = 2)
weather_7_m <-round(( mean(weather7[["NumberOfVehicles"]])), digits = 2)
weather_8_m <-round(( mean(weather8[["NumberOfVehicles"]])), digits = 2)
weather_9_m <-round(( mean(weather9[["NumberOfVehicles"]])), digits = 2)
```




```
condition <-c("Fine without high winds",
              "Raining without high winds",
              "Snowing without high winds",
              "Fine with high winds",
              "Raining with high winds",
              "Snowing with high winds",
              "Fog or mist - if hazard",
              "Other",
              "Unknown")

vehicleMean <- c(weather_1_m,weather_2_m,weather_3_m,weather_4_m,weather_5_m,weather_6_m,weather_7_m,weather_8_m,weather_9_m)
WBC <- data.frame(condition,vehicleMean)

ggplot(data=WBC, aes(x=condition, y=vehicleMean, fill=condition)) +
  geom_bar(position = 'dodge', stat='identity') +
  geom_text(aes(label=vehicleMean), position=position_dodge(width=0.9), vjust=-0.25)
```

Regression

Training and imputing new values using linear regression

Linear regression follows a uniform path from which we can predict values of based on one or more other values. In simple a line of best fit is used to help visualize the relationship between different variables, this is also a good example of linear regression and how it is used to predict values. Firstly, we split the data into two sets, Training set and a Test set. The training set will not contain any missing values unlike the test set which will be missing values for the area that we will attempt to predict. The code below will show the necessary steps made, alongside the comments to explain each step:

```
# Linear Regression ----

Data_Test <- read.csv("accidents.csv", na.strings = c("","NA"))

# The following function takes a vector as an argument and returns a binary vector
# of 0 corresponding the missing value in the argument vector and 1 if there isn't one.
Data_Test
missDummy <- function(t)
{
  x <- dim(length(t))
  x[which(!is.na(t))] = 1
  x[which(is.na(t))] = 0
  return(x)
}
# Now we will use this function to create a 'dummy' variable that will indicate
# missing values by assigning the value '0', otherwise will take the value '1'.

Data_Test$missing <- missDummy(Data_Test$AgeOfCasualty)
Data_Test
# this will pick the instances with values as training data
TrainData<- Data_Test[Data_Test['missing']==1,]
# this will pick the instances with unknown '(NA)' value as testing data
TestData<- Data_Test[Data_Test['missing']==0,]

# This will then fit a linear model with AgeOfCasualty as dependent variable and CasualtyClass, CasualtySeverity,
# TypeOfVehicle, WeatherConditions as independent variables.

model<- lm(AgeOfCasualty~CasualtyClass+CasualtySeverity+TypeOfVehicle+WeatherConditions, TrainData)

# This will predict missing values based on the model
pred<- predict(model, TestData)
pred
# Next we insert it back in the original,m the first part will show us where the NAs are.
#the second will replace the NA with the predicted variables
Data_Test$AgeOfCasualty[is.na(Data_Test$AgeOfCasualty)]
Data_Test$AgeOfCasualty[is.na(Data_Test$AgeOfCasualty)]<- pred
```

```
> pred<- predict(model, TestData)
> pred
       6       19       31       73      104      137      181      250      270      300      407      582      806      918     1219     1354
   1402     1571
31.60305 37.81372 31.60305 39.43595 34.35592 37.00612 37.81372 31.60305 31.60305 37.81372 37.81372 36.97621 34.70838 32.76823 37.00612 31.60305 3
9.43547 36.52156
    1662
33.87088
>
> Data_Test$AgeOfCasualty[is.na(Data_Test$AgeOfCasualty)]
 [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
> Data_Test$AgeOfCasualty[is.na(Data_Test$AgeOfCasualty)]<- pred
>
> summary(model)

Call:
lm(formula = AgeOfCasualty ~ CasualtyClass + CasualtySeverity +
    TypeOfVehicle + WeatherConditions, data = TrainData)

Residuals:
    Min      1Q  Median      3Q     Max
-48.544 -14.814  -3.814  12.397  74.918

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      48.20902    3.25801  14.797  < 2e-16 ***
CasualtyClass    -3.10533    0.55707  -5.574 2.81e-08 ***
CasualtySeverity -2.75287    1.06224  -2.592  0.00962 **
TypeOfVehicle     0.16152    0.06447   2.505  0.01231 *
WeatherConditions -0.48504    0.32030  -1.514  0.13010
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.37 on 2045 degrees of freedom
Multiple R-squared:  0.02012,   Adjusted R-squared:  0.01821
F-statistic:  10.5 on 4 and 2045 DF,  p-value: 2.025e-08

> |
```

Conclusion

The new dataset allows us to draw some conclusions. Firstly, crashes within Calderdale are on a decline each year. Secondly, we see that slight injuries are most common along with normal or simply windy conditions. Next, within the different weather conditions, snowy and with high winds have the highest average number of vehicles involved within the crashes. Finally, we can see that that women are involved in less crashes, allowing us to conclude that they are less likely to crash than men while driving in: "snowy with high winds", "unknown" or "other" weather conditions.