BIRMINGHAM CITY
University

Assessment Report: **Calderdale Accident Casualty**

Author: **Habban Islam (20109816)**

Date: **13 /01/2022**

Wordcount: **2212 Words**

Page count: **12 Pages**

Calderdale Accident Casualty 2014/2015

Contents

Cover Page 1

Introduction 2

Data Wrangling 3

Removing 5

Outliers 6

Data Exploration 8

Regression 11

Conclusion 12

Introduction

This report consists of data wrangling, cleaning, and manipulation of Calderdale council's road traffic collisions. The data which will be used for analysis are produced by Calderdale council available on their website - https://dataworks.calderdale.gov.uk/dataset/calderdale-accident-data-. Breakdown of all exploration and cleaning of the will be reported in here while the main R code for analysis, cleaning, regression etc. can be found in "20109816_assessment_2.R" file. All the used data cleaned data and regression data can be found in "20109816_assessment_2" folder with file name "Df".

Data Wrangling

Examining columns in the dataset

Accident dataset has 2069 rows and 14 columns. Columns like 'Number of Vehicles', 'Accident Date', 'Time 24hr', 'Type of vehicles' and '1st Road Class' are the factors which come into play during the accident; these variables will help us to classify type and number of vehicles which are involved the most with accidents as well as notifying us about the timing and surroundings of the accident.

Some other columns such as 'Road surface', 'Lighting Conditions', 'Daylight/Dark', 'Weather Conditions' allows us to acknowledge about the road conditions and environmental factors behind the accidents.

The column 'Local Authority' tells us the accidents occurred in Calderdale area.

Then 'Casualty Class', 'Casualty Severity', 'Sex of Casualty', 'Age of Casualty' columns helps us to understand the behavior and characteristics of individuals who are involved with the accident.

Examining missing data

```
> colSums(is.na(data))
  Number.of.Vehicles        Accident.Date          Time..24hr.      X1st.Road.Class          Road.Surface  Lighting.Conditions 
                   0                    0                    0                    0                    0                    0 
      Daylight.Dark  Weather.Conditions      Local.Authority      Type.of.Vehicle      Casualty.Class     Casualty.Severity 
                   0                    0                    0                    0                    0                    0 
    Sex.of.Casualty      Age.of.Casualty 
                   0                   19 
```

There are 19 missing values in the dataset and they all from "age of casualty" column**.** Down below I have listed all the columns which are related with 'age of casualty' missing values.

	Number.of.Vehicles	Accident.Date	Time..24hr.	X1st.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
6	1	15/01/2017	1659	U	Wet/Damp	4	Dark	1	Calderdale	9	3	3	1	NA
19	2	27/01/2017	1835	U	Wet/Damp	4	Dark	1	Calderdale	9	1	3	1	NA
31	1	04/02/2017	1730	A646	Dry	4	Dark	1	Calderdale	9	3	3	1	NA
73	2	26/03/2017	1353	U	Dry	1	Daylight	1	Calderdale	2	1	2	1	NA
104	1	01/05/2017	1454	U	Dry	1	Daylight	1	Calderdale	9	3	2	2	NA
137	2	20/06/2017	752	A58	Dry	1	Daylight	1	Calderdale	4	1	3	1	NA
181	1	24/07/2017	2328	U	Dry	4	Dark	1	Calderdale	9	1	3	2	NA
250	1	03/10/2017	850	A58	Wet/Damp	1	Daylight	1	Calderdale	9	3	3	1	NA
270	1	18/10/2017	1007	U	Dry	1	Daylight	1	Calderdale	9	3	3	1	NA
300	2	20/11/2017	1930	U	Wet/Damp	4	Dark	1	Calderdale	9	1	3	2	NA
407	2	26/02/2016	1340	6	1	1	Daylight	1	Calderdale	9	1	3	1	NA
582	1	21/05/2016	210	3	2	4	Dark	2	Calderdale	9	2	2	1	NA
806	4	27/10/2016	1735	1	1	1	Daylight	1	Calderdale	9	2	3	1	NA
918	2	23/01/2015	1825	6	2	4	Dark	5	Calderdale	9	2	3	2	NA
1219	2	03/08/2015	1403	3	1	1	Daylight	1	Calderdale	4	1	3	1	NA
1354	1	07/11/2015	2230	6	2	4	Dark	1	Calderdale	9	3	3	2	NA
1402	2	30/11/2015	1700	3	2	4	Dark	2	Calderdale	5	1	2	1	NA
1571	2	14/01/2014	1350	3	2	1	Daylight	1	Calderdale	1	1	3	1	NA
1662	1	09/05/2014	824	3	2	1	Daylight	2	Calderdale	9	3	2	1	NA

These 19 values could be missing because of absence of the casualty during the data collection, or they did not want to disclose their age. Data can be missing because of Missing at Random (MAR) or missing not at Random (MNAR). Example of MNAR can be found that casualty was a young driver and did not want to disclose his age because of shame as he does not want other people to judge his driving skills. MNAR can be that the casualty who was involved in accident, but no one was there for data gathering and he was not asked about his age.

Examine and fix anomalies

If we look closely at 'Road surface' and '1st Road class', we can see that there is inconsistency through the columns. Data type of the columns differ as some of them are text/char and some are numerical.

U	Wet/Damp
U	Dry
A58	Frost/Ice
U	Dry
3	2
6	2
6	1
3	2
3	2

But the data type should only be numerical as they are categorized in 'Guidance.csv' file. For example: Road surface Dry has been categorized as 1 so throughout the column it should only appear as 1 not 'Dry'. These anomalies must be fixed for us to do exploration on that data. For us to fix these inconsistencies we will have to convert these real-life conditions to numerical value using the guideline provided in 'Guidance.csv' file.

Let's fix the Road surface column first, we can use this R code below to show us all the different variables in the Road surface column.

```
> unique(data$Road.Surface)
 [1] "Wet/Damp"   "Dry"        "Frost/Ice"  "Ice"        "Snow"       "Wet"        "Wet ¨ Damp" "2"          "1"          "3"
[11] "4"          "5"
```

Then we can use pipe to mutate each variable to its matching numerical values which are taken from 'Guidance.csv' file.

```
> # Changing surfaces to its matching number from "Guidance"
> data <- data %>% mutate(Road.Surface = sub("Wet/Damp", "2", Road.Surface)) %>%
+       mutate(Road.Surface = sub("Wet ¨ Damp", "2", Road.Surface)) %>%
+       mutate(Road.Surface = sub("Wet", "2", Road.Surface)) %>%
+       mutate(Road.Surface = sub("Snow", "3", Road.Surface)) %>%
+       mutate(Road.Surface = sub("Frost/Ice", "4", Road.Surface)) %>%
+       mutate(Road.Surface = sub("Ice", "4", Road.Surface)) %>%
+       mutate(Road.Surface = sub("Dry", "1", Road.Surface))
> # Checking if the anomalies are fixed
> unique(data$Road.Surface)
[1] "2" "1" "4" "3" "5"
```

This above code shows that all the text or character have been changed to numerical value. To check if all the values been changed, we used 'unique (data$Road.Surface)' and it displayed only numerical value suggesting anomalies have been fixed successfully.

We can use these same commands to fix '1st Road Class' anomalies. First, we will use the unique command to check all the different variables then we will use pipe mutate command to change them.

```
> # Check for Unique 1st.Road.Class
> unique(data$X1st.Road.Class)
 [1] "U"         "A58"       "A646"      "B6138"     "A629"      "A641"      "A672"      "A6033"     "A6139"     "A644"      "A62"       "B6114"
[13] "A6319"     "B6112"     "M62"       "A681"      "B6113"     "A629(M)"   "A643"      "A6036"     "A6025"     "A647"      "A6026(M)"  "A649"
[25] "A6026"     "3"         "6"         "1"         "4"         "2"
>
```

We will be using Expression language to put down each of these variables for the mutation.

```
> # Changing 1st.Road.Class to its matching number from "Guidance" USING EXPRESSION LANGUAGE
> data <- data %>% mutate(X1st.Road.Class = sub("M\\d{1,9}", "1", X1st.Road.Class)) %>%
+       mutate(X1st.Road.Class = sub("A\\d{1,9}\\(M)", "2", X1st.Road.Class)) %>%
+       mutate(X1st.Road.Class = sub("A\\d{1,9}", "3", X1st.Road.Class)) %>%
+       mutate(X1st.Road.Class = sub("B\\d{1,9}", "4", X1st.Road.Class)) %>%
+       mutate(X1st.Road.Class = sub("U", "6", X1st.Road.Class))
> # Checking if the anomalies are fixed
> unique(data$X1st.Road.Class)
[1] "6" "3" "4" "1" "2"
```

'sub("A\\d{1,9}\\(M)", "2", X1st.Road.Class)' This is expression language. In this code we are telling the function to match anything which starts with 'A' then has digits (\\d) from 1 to 9 ({1, 9}) and ends with (M). Then change that to '2' in the 'X1st.Road.Class' column.

We called the unique command again to check if all the characters have been changed to numbers and indeed it has as we can see on the above screenshot.

Removing unnecessary columns

```
 [ reached 'max' / getOption("max.print") -- omitted 1986 rows ]
> updated <- data%>%select(-c("Daylight.Dark","Local.Authority"))
>
```

If we look at the 'Accident.csv' database, we can identify two columns which are not needed. These two columns are 'Local Authority' and 'Daylight/Dark' columns. Local Authority is constant column with non-changing value, all the rows in the column repeats the name of the local authority 'Calderdale' which is the council where all these accidents taken place. So, we have decided to take the column away, instead of the full column, a title can be used with the 'Calderdale' to address that all accidents occurred in that area. 'Local authority' is non changing variable which is why it has been deleted.

'Daylight/Dark' is not needed to be present on the data base as we already have a column with 'lighting conditions' which provides more detailed information about the surrounding and lighting of the road during the accident. 'Daylight/Dark' is a duplicated data because of this, it will also be erased.

Check and delete outliers from 'Age of Casualty'

Outliers are data variables which differ from other observed data on the dataset. Outliers can occur for several reasons such as wrong measurement, missing data gathering. These outliers can have significant affect on data analysis as these values are wrong data and should be erased from the data set. There are 3 main methods we can use to identify outliers, they are: 3 sigma rule, Hampel identifier and box plot. 3 sigma rule uses mean to look for outliers. 3 sigma rule follows a pattern with which values gather near the mean; having almost equal data below and above the mean. This is the code I used for 3 sigma rule:

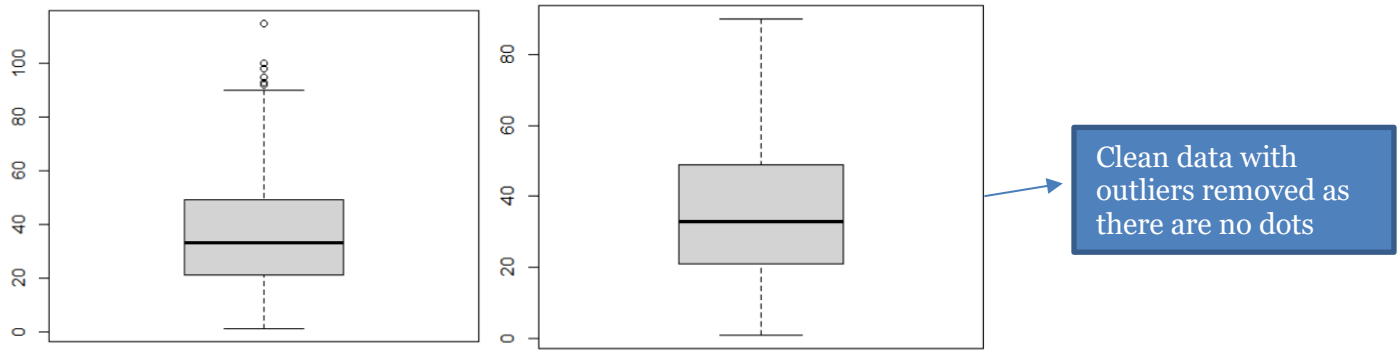
```
# 3sigma Rule
# Standard deviation for age of casualty
sd_value <- sd(updated$Age.of.Casualty, na.rm = "TRUE") # na.rm skips NA value
sd_value
] 19.55346
# Mean for age of casualty
mean_Value <- mean(updated$Age.of.Casualty, na.rm = "TRUE")
mean_Value
] 36.21366
# calculate upper and lower bounds
upper_bound <- mean_Value + 3*sd_value
lower_bound <- mean_Value - 3*sd_value
upper_bound
] 94.87405
lower_bound
] -22.44673
# Extract outliers
outliers_sigma <-updated %>% filter((Age.of.Casualty > upper_bound)| (Age.of.Casualty < lower_bound))
outliers_sigma
Number.of.Vehicles Accident.Date Time..24hr. X1st.Road.Class Road.Surface Lighting.Conditions Weather.Conditions Type.of.Vehicle
                 2    31/01/2017        1840              6            2                   5                  2               9
                 1    16/06/2016        1215              6            1                   1                  1               9
                 1    05/02/2014        1715              6            2                   4                  1               8
                 2    08/06/2014        1650              3            1                   1                  1               9
Casualty.Class Casualty.Severity Sex.of.Casualty Age.of.Casualty
             1                 2                2             115
             3                 3                1             100
             3                 2                1              95
             2                 3                2              98
```

In the above screenshot, we can see that I have calculated standard deviation which tells us how dispersed our data is compared to mean. I had to remove NULL value as mean can not be calculated with Null value present in the data. 3 sigma rule managed to find 4 outliers. Next outlier we will be checking is Hampel Identifier. In Hampel Identifier we use median and MAD value instead of mean and standard deviation. Below is the code I used for Hampel Identifier:

```
# Hampel Identifier
# Calculate median and MAD
median_value <- median(updated$Age.of.Casualty, na.rm = "TRUE")
MAD_value <- mad (updated$Age.of.Casualty, na.rm = "TRUE")
median_value
] 33
MAD_value
] 19.2738
# Calculate upper and lower bounds
upper_bound <- median_value + 3*MAD_value
lower_bound <- median_value - 3*MAD_value
upper_bound
] 90.8214
lower_bound
] -24.8214
# Extract outliers found be Hampel identifier
outliers_hampel <- updated %>% filter((Age.of.Casualty > upper_bound)| (Age.of.Casualty < lower_bound))
outliers_hampel
Number.of.Vehicles Accident.Date Time..24hr. X1st.Road.Class Road.Surface Lighting.Conditions Weather.Conditions Type.of.Vehicle
                 2    31/01/2017        1840              6            2                   5                  2               9
                 1    24/06/2017        1200              3            1                   1                  1               9
                 1    19/03/2016        1902              6            1                   1                  1               9
                 1    16/06/2016        1215              6            1                   1                  1               9
                 1    28/06/2016        1627              3            1                   1                  1               9
                 1    05/02/2014        1715              6            2                   4                  1               8
                 2    08/06/2014        1650              3            1                   1                  1               9
Casualty.Class Casualty.Severity Sex.of.Casualty Age.of.Casualty
             1                 2                2             115
             3                 2                1              93
             3                 2                1              93
             3                 3                1             100
             3                 2                2              92
             3                 2                1              95
             2                 3                2              98
```

Finally, the last outlier is Box plot. Box plot looks at the Minimum, 1st quartile, median, 3rd quartile and maximum of the dataset. Any data which are above minimum, and maximum is classed as outliers. Below is the code I have used for box plot analysis:

```
# Boxplot
boxplot(updated$Age.of.Casualty)
outlier <- boxplot(updated$Age.of.Casualty, plot=TRUE)$out
outlier
L] 115  93  93 100  92  95  98
# Removing outliers from the dataset
# Rows with outliers
updated[which(updated$Age.of.Casualty %in% outlier),]
    Number.of.Vehicles Accident.Date Time..24hr. X1st.Road.Class Road.Surface Lighting.Conditions Weather.Conditions Type.of.Vehicle
1                    2    31/01/2017        1840              6            2                   5                  2                9
46                   1    24/06/2017        1200              3            1                   1                  1                9
58                   1    19/03/2016        1902              6            1                   1                  1                9
L7                   1    16/06/2016        1215              6            1                   1                  1                9
35                   1    28/06/2016        1627              3            1                   1                  1                9
462                  1    05/02/2014        1715              6            2                   4                  1                8
722                  2    08/06/2014        1650              3            1                   1                  1                9
    Casualty.Class Casualty.Severity Sex.of.Casualty Age.of.Casualty
1                1                 2                2             115
46               3                 2                1              93
58               3                 2                1              93
L7               3                 3                1             100
35               3                 2                2              92
462              3                 2                1              95
722              2                 3                2              98
# Rows containing the outliers
clean <- updated[-which(updated$ Age.of.Casualty %in% outlier),]
# Checking boxplot to see if outliers are gone
boxplot(clean$Age.of.Casualty)
```



If we look carefully, we can see that box plot also provides us with same 7 outliers as Hampel Identifier. Box plot also visualizes the outliers with dots in the box plots, which is why I have decided to use Box plot to remove all the outlier values.

Save clean data

Now that we removed all the outliers from our database, it can be classed as clean dataset. Ready for next part of the report which is analysis, exploration, and regression.

```
> # save
> write.csv(clean, "clean_accident.csv")
>
```

Data Exploration

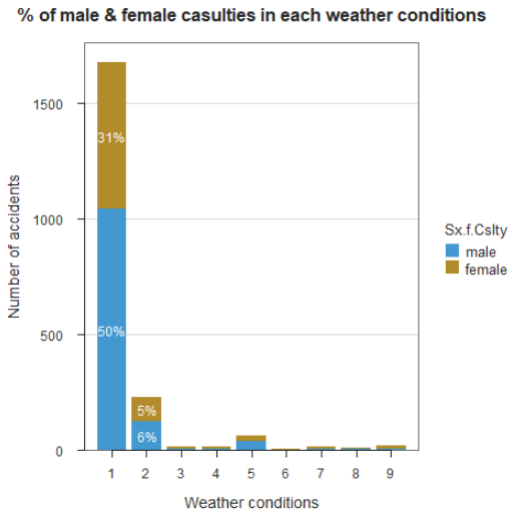
Exploratory data analysis (EDA) helps us processing data to summarise the main characteristics of a dataset. Outcome of EDA can be represented with visual or non-visual methods.

Accident effects on both gender under all weather conditions

We will be comparing male and female drivers and see who were involved with accidents the most and in what weather conditions.

```
# Comparing male & female casualties with weather Conditions
# Filter Weather.Conditions and Sex.of.Casualty
wsdata <- clean[,c("Weather.Conditions", "Sex.of.Casualty")]
# Number of male & female casualties in each weather conditions and their %
# Table showing Number of Accidents caused by male(1) & female(2) in each weather conditions
table(wsdata$Sex.of.Casualty, wsdata$Weather.Conditions)

     1    2    3    4    5    6    7    8    9
1 1041  127    9    9   38    3    9    5    7
2  634  105    7    8   27    5    9    7   12
# % of male & female casualties in each weather conditions
BarChart(data = clean, Weather.Conditions, by = Sex.of.Casualty,
         main = "% of male & female casulties in each weather conditions",
         legend_labels = c("male", "female"),
         xlab = "Weather conditions", ylab = "Number of accidents")
```

In the code above we can see that I have only selected weather condition and sex of casualty columns as we only need these 2 columns for the required analysis. I have also created a table to display number of accidents both male and female were involved in and their weather conditions (1-9). Bar chart also been used to visualize this data easily:



I have also looked at individual weather conditions to see which gender had most accidents in which weather condition, below I have shown analysis for one weather condition:

```
# Compare the rate of accidents by each sex in various weather conditions
# Weather conditions: (1) Fine without high winds
male <- count(subset(wsdata, Sex.of.Casualty == 1 & Weather.Conditions == 1))
male
   n
1041
female <- count(subset(wsdata, Sex.of.Casualty == 2 & Weather.Conditions == 1))
female
  n
634
difference <- as.integer( male - female)
difference # male 407 more
```

From this we can see that male drivers have 407 accidents more than female drivers in ‘ (1) Fine without high winds’ weather conditions. Like this I have done analysis for all weather conditions on the R file.

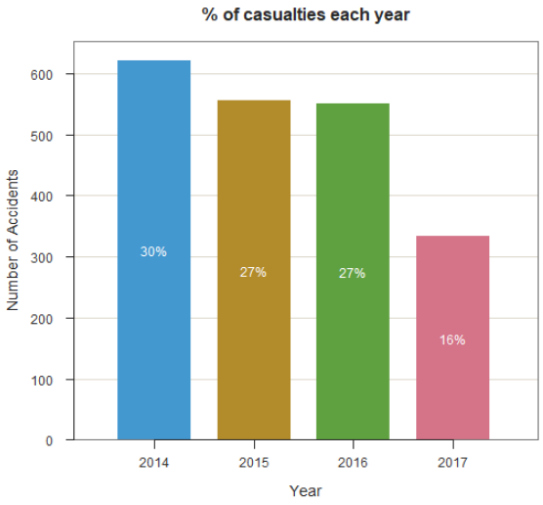
Number of casualties on yearly basis

Since our dataset does not clarify about the number of casualties involved in accident, I will be assuming there is one casualty per accident. The date column in our dataset are not grouped by year so we will have to group it by year in order to get the analysis done.

```
# Parsing date to be used as year & add only year in column instead of whole date
date <- as.Date(clean$Accident.Date, "%d/%m/%Y")
date_df$year <- strftime(date, "%Y")
date_df
```

This code will create a column with the name 'year' where each year of accidents will be displayed instead of the whole date. Then I have created a table and bar chart displaying number of accidents which occurred in yearly bases:

```
> tabyl(date_df, year) %>% adorn_pct_formatting(digits = 1)
 year   n percent
 2014 621   30.1%
 2015 556   27.0%
 2016 551   26.7%
 2017 334   16.2%
> # Representing as bar chart
> BarChart(data = date_df, year,
+          main = "% of casualties each year",
+          xlab = "Year",  ylab = "Number of Accidents")
```



This bar chart suggests that accident rates have been decreasing over the years, as we can see it came down to 16% in 2017 from 30% which was in 2014. 2014 is the year with highest casualties.

Relationship between light conditions and severity of casualty

First, I have created a relational table which represents the relationship between lighting condition and severity which shows us Number of accidents which occurred in each condition (1-7) and their severity.

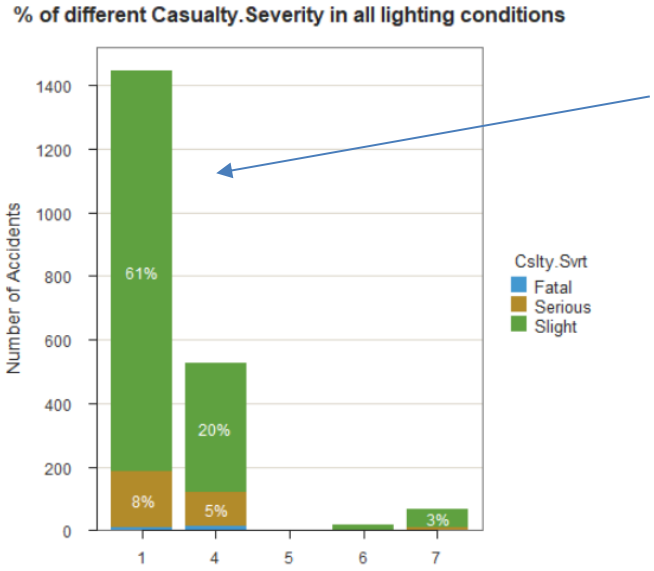
```
> # Number of accidents compared with severity(1-3) and lighting condition(1-7) in table
> table(clean$Casualty.Severity, clean$Lighting.Conditions)

    1   4   5   6   7
1  10  15   0   0   0
2 175 106   0   3  10
3 1261 405   3  15  59
```

(1, 2, 3) represents severity

These values are the Number of accidents which occurred

(1 to 7) represents lighting conditions

To explore the relationship between these two variables I have also done bar chart which shows number and percentage of each casualty severity (blue = fatal, green = slight, red = serious) in each lighting conditions.



This shows most accident occurred in weather condition 1 (Daylight: streetlights present) followed by 4

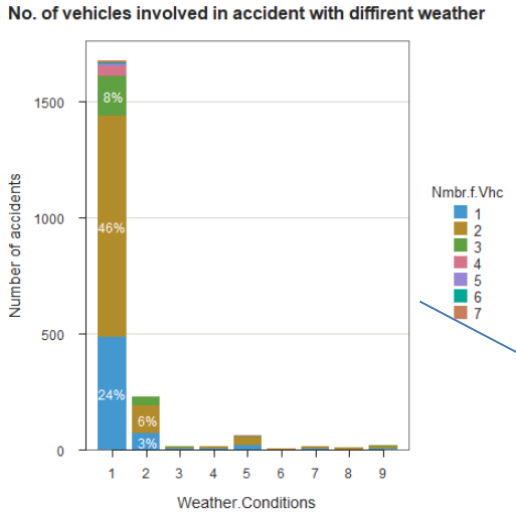
Bar chart above is mostly green indicating that most casualty severities are slight, followed by serious and fatal.

Relationship between weather conditions and the Number of vehicles involved

For this comparison I have also displayed all the information in a table for clear visualization. Number of vehicles (1-7) and weather conditions (1-9)

```
> # Weather condition and number of vehicles involved
> # Number of vehicles (1-7) involved in accident in different Weather conditions (1-9)
> table(clean$Number.of.Vehicles, clean$Weather.Conditions)

     1   2   3   4   5   6   7   8   9
  1 489  71   6   7  19   2   6   4   6
  2 949 122   6  10  35   4  10   6  12
  3 170  37   3   0   7   2   2   2   1
  4  44   2   1   0   4   0   0   0   0
  5   8   0   0   0   0   0   0   0   0
  6   5   0   0   0   0   0   0   0   0
  7  10   0   0   0   0   0   0   0   0
```

Bar chart also have been done for this relationship.




```
# Bar chart displaying % of each Casualty.Severity for all (1-7) lighting conditions
BarChart(data = clean, Weather.Conditions, by = Number.of.Vehicles,
         main = "No. of vehicles involved in accident with diffirent weather",
         xlab = "Weather.Conditions", ylab = "Number of accidents")
```

From this Bar chart we can analyse that mostly covered brown colour indicates most accidents occurred when two vehicles were involved in the collusion as has been coded as brown and the bar chart mostly covered with brown.
We can also see that most accident occurred in 1st weather condition (Fine without high winds) followed by 2nd.

Regression

Training and assigning new values using linear regression

Linear regression is a regression model which estimates relationship between independent and dependent variable using a constant path. In this data set out dependent variable is 'age of casualties' as we are trying to find an estimate of the missing values, so to find them we will use independent variables like 'casualty class', 'casulty severity', 'Type of vehicle' and 'weather conditions.

To start the regression, we will have to split the data into training(independent) and test data(dependent). Training data will have full set of data without any missing values we will be using that data to predict the missing values for testing data.

```
> # Using the training data to create a multilinear regression to model the relationship
> model <- lm(Age.of.Casualty ~ Casualty.Class + Casualty.Severity + Type.of.Vehicle + Weather.Conditions, data=trainData)
> # Predict missing values in the df and replace them
> predictions <- model %>% predict (testData)
> predictions
       6       19       31       73      104      137      181      250      270      300      407      582      806      918     1219
31.03531 37.88713 31.03531 38.79651 33.11843 37.04875 37.88713 31.03531 31.03531 37.88713 37.88713 36.07858 34.46122 32.59815 37.04875
    1354     1402     1571     1662
31.03531 38.83377 36.54571 32.65267
> selected_rd$Age.of.Casualty[is.na(selected_rd$Age.of.Casualty)]<-as.integer (predictions)
```

First, we start with a linear model, which describes the relationship between variables in the regression. Then we use R built in predict command for predictions. Once we have the predictions, we replace all the NA values in age of casualty with predicted values.

```
> summary <- summary(model)
> summary

Call:
lm(formula = Age.of.Casualty ~ Casualty.Class + Casualty.Severity +
    Type.of.Vehicle + Weather.Conditions, data = trainData)

Residuals:
    Min      1Q  Median      3Q     Max
-48.126 -14.887  -3.887  12.454  57.288

Coefficients:
                  Estimate Std. Error t value             Pr(>|t|)
(Intercept)       46.51908    3.21421  14.473 < 0.0000000000000002 ***
Casualty.Class    -3.42591    0.54972  -6.232       0.000000000558 ***
Casualty.Severity -2.08312    1.04918  -1.985              0.04722 *
Type.of.Vehicle    0.16768    0.06337   2.646              0.00821 **
Weather.Conditions -0.46577    0.31487  -1.479              0.13923
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.04 on 2038 degrees of freedom
Multiple R-squared:  0.02268,   Adjusted R-squared:  0.02076
F-statistic: 11.82 on 4 and 2038 DF,  p-value: 0.000000001703
```

This above figure shows us the summary of the model. It shows us the residuals, coefficients, t values and r square. Residuals tells us the difference between observed and predicted value meaning showing us the error. Smaller residuals are better as it means the median is closer to '0' so less error. We can also see that the magnitude of 1st and 3rd quartile are similar, and the min and max which suggests that data is evenly distributed.

Coefficients shows us the estimate, t values and p values. Estimates are used to predict the value of response variable. Std. error tells us about the average amount by which estimate varies from the actual value. In here we can see we have smaller std.error value meaning our predictions are quite good. T value measure how many standard deviations there are between the estimate and zero. Our t values are small meaning there is significant evidence to reject the H0. We also have smaller p value meaning that we can accept the H1 hypothesis which clarifies that there is a good similarity between the independent and dependent variables. Our dataset is also not overfitting a we can see that the difference between multiple r2 and adjusted r2 is not significant.

Conclusion

From the analysis, we can draw a conclusion that overall accidents in Calderdale is decreasing and most of the accidents which occur have slight injury and most of them takes place in windy and rainy weather conditions. Also, most of the accidents occur in normal daylight condition and generally 2 cars are involved in accident. The data also has proven that men are more likely to be involved with accidents than female.