# Milestone 3: Blood alcohol limit changes upon crashes

Emma Wang

## 1 Goal

The goal of this project is to analyze if the two changes in drink-driving limits in 2011/2014 affected the crashing events across New Zealand; if so, the effects will be quantified by the model building process.

## 2 Data Source

The primary data sources Crashing Analysis System are published and actively maintained by NZ Transport Agency. The data (*lastly updated Apr 8, 2022*) recorded the crashing cases in New Zealand from 2000 to 2021 with 72 variables. From the field description, many variables may correlate with one another, where variable selection would be necessary to avoid over-fitting.

## 3 Data Processing

The first step was to check the input of each variable based on common sense. The data were pre-cleaned before publication as no suspicious values were found.
The main tasks before fitting the models were to remove irrelevant variables, and to create new variables that would assist with solving the question of interest.

Referring to the field description, the time when each crash took place was recorded in years. The lack of detailed information will cause uncertainty in the categorization of the cases. For example, for an incident observed in 2011, it was unclear whether it happened after the first change or not.
For the two changes in alcohol limits, the cases were categorized into 3 phases. Because both changes took place near the end of the year, the crashes in 2011/2014 were classified to the previous period. Thus, the variable `Change` was derived from `crashYear` as shown in Table 1.. It would be the key variable for investigating the effects of the changed policies.

Table 1: The categorization of the variable `Change`.

| Period | Label | Change |
|---|---|---|
| 2000-2011 | Baseline | 0 |
| 2012-2014 | Post-First Change | 1 |
| 2015-2021 | Post-Second Change | 2 |

```
pacman::p_load(tidyverse, data.table, here)
data_raw <- fread(here("Crash_Analysis_System_(CAS)_data.csv"))
# Create variable 'Change'
data <- data_raw %>%
  mutate(Change = as.factor(case_when(crashYear <= 2011 ~ 0,
                                      crashYear %in% 2012:2014 ~ 1,
                                      crashYear > 2014 ~ 2)))
```

The variable `caseSeverity` recorded the most severe injury in the crashes in four different levels: non-injury, minor, serious and fatal.
Other variables recorded the exact numbers of injuries with certain level of severity in an accident(`minorInjuryCount`, `seriousInjuryCount` and `fatalCount`). They are the potential candidates for the response variables.
For the complete data processing, please see the Appendix.

## 4 Data Exploration

In this section, the two proposed directions of analysis will be explored. Based on the missingness of each variable, and correlation between variables, it would be possible to perform variable selection and identify the important covariates in this section.

## 4.1 Most severe injury in the crashes and changes in alcohol limits

```
# Visualization: average cases per year
annual_case.df <- data %>%
  select(Change, crashSeverity) %>%
  group_by(Change, crashSeverity) %>%
  summarise(total_count = n()) %>%
  mutate(total_years = case_when(Change == 0 ~ 12,
                                 Change == 1 ~ 3,
                                 Change == 2 ~ 7),
         average_annual_count = total_count/total_years)

annual_case.df %>%
  ggplot +
  geom_bar(aes(x = crashSeverity, y = average_annual_count, fill = Change),
           stat = "identity", pos = "dodge") +
  scale_x_discrete(labels = c("Non-injury", "Mild Injury",
                              "Severe Injury", "Fatal")) +
  scale_y_continuous(limits = c(0, 28000), expand = c(0, 0)) +
  labs(x = "Severity", y = "Average Annual Count") +
  scale_fill_discrete(labels = c("Baseline", "Post-First Change", "Post-Second Change")) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```

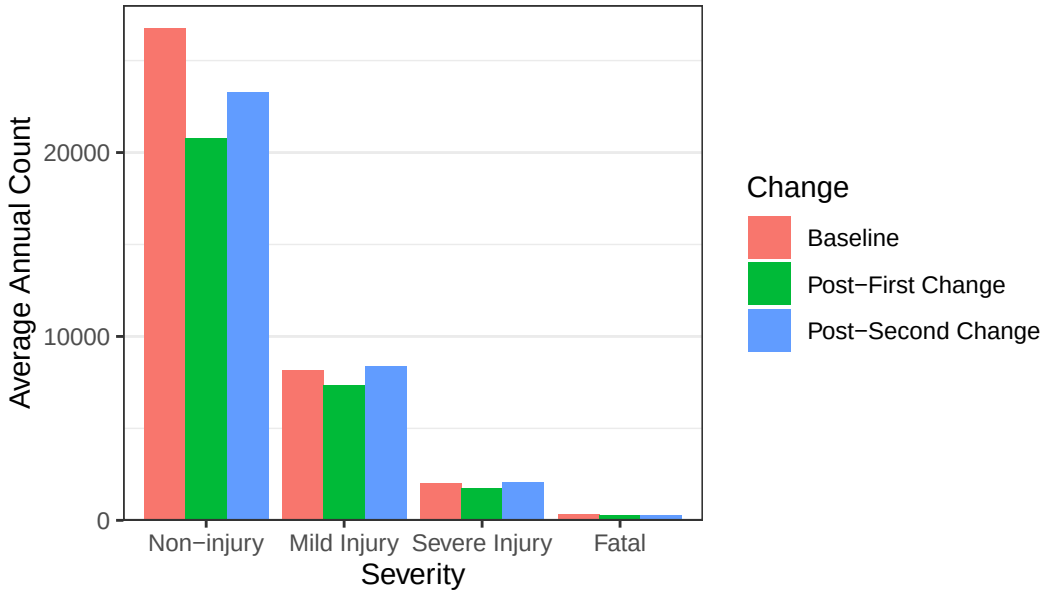


Fig 1. Distribution of the average annual counts of the most severe cases in the crashes across 3 periods of alcohol-limit changes. The average counts during the Post-First-change period were lower than that in the other period.

The counts were adjusted by the number of years in each period, as specified by `total_years`. After the first change in drinking driving policies, there were significant reduction in crashing numbers. An interesting observation was that the average counts increased after the second reduction in alcohol.

## 4.2 Changes in jnjuries counts across the years

```
data %>% select(crashYear, fatalCount, minorInjuryCount,
                seriousInjuryCount, Change) %>%
  pivot_longer(cols = 2:4, names_to = "Severity", values_to = "Count") %>%
  mutate(Severity = gsub("^([a-z]+)([I|C].+)", "\\1", Severity)) %>%
  group_by(crashYear, Severity) %>%
  summarise(annual_count = sum(Count, na.rm = T),
            num_cases = n()) %>%
  ggplot(aes(x = crashYear, y = 10 * annual_count/num_cases, colour = Severity)) +
  geom_point()+
```

```
  geom_line() +
  geom_vline(xintercept = 2011, linetype = "dashed", colour = "orange") +
  geom_vline(xintercept = 2014, linetype = "dashed") +
  labs(x = "Year", y = "Count of Injuries per 10 Case",
       title ="Annual total injuries in the crashs") +
  theme_bw() +
  annotate("text", label = c("First Change", "Second Change"),
           x = c(2007, 2018.5), y = 2, family = "mono", size = 3.5,
          fontface = "bold", colour = c("orange", "black"))
```

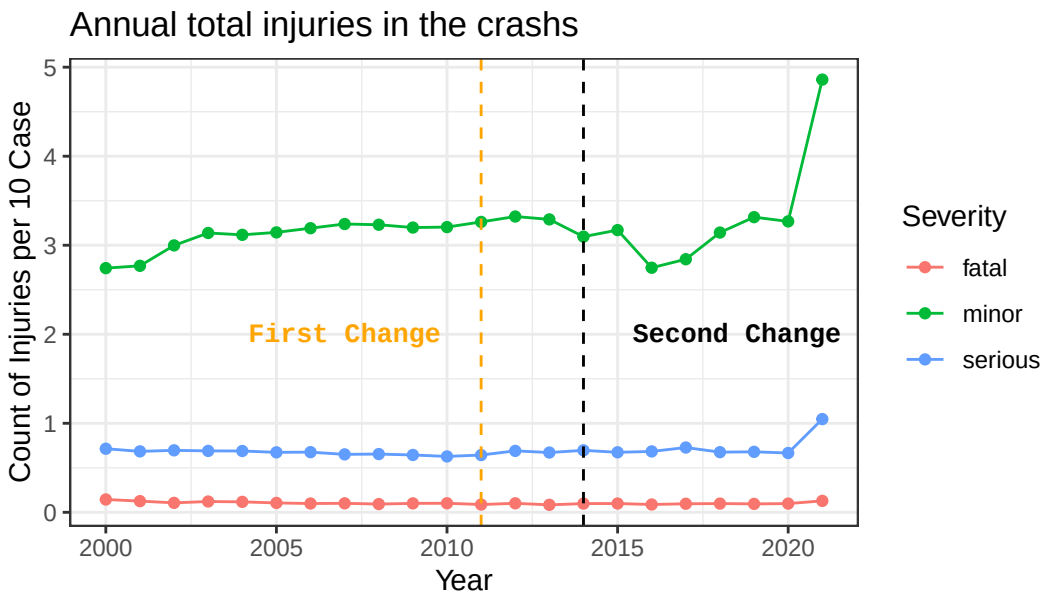


Fig 2. Changes in average injury counts per 10 cases of various severity from 2000 to 2021. The two changes in drinking-driving regulations (2011, 2014) were marked by the dashed vertical lines. Minor injury was the leading type across the time.

In Fig 2., the minor-injury counts of the injuries per 10 cases increased before 2012. The decremented pattern started from 2012 might result from the reducing alcohol limit in 2011.
After second change in alcoholic limit (2014-2016), we saw a dramatically reduction in the minor injury counts. Unexpectedly, the minor injuries increased significantly after 2016. We would build a model to see whether the changing policies played a role in pattern decribed above.

## 4.3 Variable Selection for the Models

### 4.3.1 Missing values analysis

Here the variables with more than 50% missingness were identified. Variables with more than 90% missingness will be discarded because they had no predictive ability.
The variables included in the Multiple Imputation would be assessed based on
**1.** correlation with the outcome variables AND;
**2.** causal relationship with the response.
Please see the next 2 subsections for the selection of key predictors for the models.

```
# Variables with more than 50% missing values
var_50_miss <- miss_var_summary(data) %>%
  filter(pct_miss > 50) %>%
  select(variable) %>%
  unlist %>% unname
gg_miss_var(data %>% select(all_of(var_50_miss)), show_pct = T)+
  labs(title="Percentages of missing values") +
  theme(axis.text.y = element_text(size = 7))
```


Fig 3. Variables with more than 50% missing values.

### 4.3.2 Correlation Analysis

The code and plot of correlation analysis were shown here in the Appendix. According to Fig 5. There were no strong correlations between the outcome variables with the covariates from the plot. `crashSeverity` only showed a noticeable positive correlation with the outcome variables `minorInjuryCount` and `fatalCount`. And it had a moderate negative correlation with `urban`.

## 4.4 Other variables to be included in models

Using common sense and field descriptions, there were a list of variables that may contribute to the severity of the car crashes.

```
road_conditions <- c("roadSurface", "NumberOfLanes", "speedLimit", "light",
                     "roadLane", "streetLight", "trafficControl", "flatHill",
                     "roadworks")
location <- c("urban", "region")

other_vehicles <- c("bicycle", "train", "slipOrFlood", "vehicle", "schoolBus",
                    "truck", "vanOrUtility", "bus")

time <- c("Change", "crashYear")
weather <- c("weatherA", "weatherB")
other_objects <- c("strayAnimal","bridge", "waterRiver", "cliffBank")
```

Those variables would also be imputed, if they had missing values.

## 4.5 Multiple Imputation

As shown in the Missing values analysis, there was large amount of missing values in the original data, especially in the key predictors.

```
miss_var_summary(data) %>% filter(pct_miss > 0) %>% select(variable) %>%
  unlist %>% length # 47 variables with missing values out of 75

## [1] 47

pct_complete_case(data) # There were no complete cases in the original data

## [1] 0
```

It was insensible to remove all the 47 incomplete variables. Therefore, Multiple Imputation was needed to make the most use of the data. Firstly, the irrelevant, duplicated variables were removed.

```
# Variables without predictive powers
var_set_up <- c("X", "Y", "OBJECTID")
# Duplicated variables that represent information already in the data
var_dup <- c("crashFinancialYear", "tlaId")
# Variables with more than 90% missingness
var_high_miss <- miss_var_summary(data) %>%
  filter(pct_miss > 90) %>%
  select(variable) %>%
  unlist %>% unname
var_relevant <- setdiff(names(data), c(var_dup, var_set_up, var_high_miss))
data_clean <- data %>% select(all_of(var_relevant))
```

For further variable selection, please see Variables included in the Multiple Imputation. The key criteria were listed here:
**1.** Variables with at least 0.2 correlation with the outcome variables;
**2.** Key variables affecting the seveity of crash crashes, using common sense;
**3.** Variables that were complete in the original data.
The complete Multiple Imputation was included in the Appendix. For simplicity, the results were directly attached here.

```
# Imputation mids object with 10 imputations
load("imp_all.Rdata")
# The long data set
load("comp_long.Rdata")
```

# 5 Analytical Plan

From the exploration, the counts and severity of crashes after the first change in policies (2011) were less than the in the other two periods. The underlying reasons might be the changed policies – we need to construct models and adjust for other variables before concluding.
The analysis would be a supervised learning problem aiming to analyze the relationship between the severity of cases and policy changes in drinking-driving regulations.

## 5.1 The relationship between `Change` and probability of fatal or serious crash `crashSeverity`

One option would be to analyze the relationship between the most severe injury in the crashes and changes in alcohol limits. Due to the small number of fatal cases in the data, we would analyze them with serious cases together.

Based on the workflow of imputed data[1], the general plan to obtain the final models are listed here:
**1.** Fit the unadjusted model on each copy of 10 copies of the imputed data, which only included `Change` as the explanatory variable;
**2.** Fit the full adjusted model on each copy of imputed data, which included all the key predictors in the data set;
**3.** Do variable selection for the each of full adjusted models using model selection techniques(e.g. LASSO and Ridge);
**4.** Select the key variables using the majority rule (mimic the stepwise model selection in chapter 5.4 of Stef's book[1]);
**5.** Fit the model with the selected key variables in the step 4, together with (`Change`. Pool the estimates together.

The response `crashSeverity` fell into four categories. A binary variable would be created, which is 1 if the severity was fatal or serious, otherwise 0. A logistic regression would be fitted by `glm()` function.

### 5.2 The relationship between `crashYear` and minor injury counts per case `minorInjuryCount`

According to Fig 2., the counts of minor causalities per case varied significantly across different periods on a linear scale. Instead of using `Change`, we would fit a piece-wise linear regression on `crashYear` to demonstrate the effects of time. It were assumed (and observed in the plot) that turning points occurred in year 2011, 2014 and 2016. Piece-wise linear changes were built accordingly:

Table 2: The segments of `crashYear` in the piece-wise linear regression.

| Period | Note | Code in the Model |
|---|---|---|
| 2011-2014 | Post-First Change | `pmin(pmax(crashYear - 2011, 0), 3)` |
| 2014-2016 | Post-Second Change | `pmin(pmax(crashYear - 2014, 0), 2)` |
| 2016-2021 | The period of incremental pattern since 2016 | `pmax(crashYear - 2016, 0)` |

If, after adjustment, the coefficients for either Post-First or Post-Second piece-wise linear terms were significant, the reductions in alcoholic limit was effective in reducing the injury counts.
Theoretically, the efficiency of the linear regression can be improved by using `biglm()`[3] function, especially for large data sets. The computation of the least square line took less memory, thus fewer time to execute.
During model fitting, the execution time of the following 3 methods would be carefully compared by `system.time()`:
**1.** `lm()`;
**2.** `biglm()` that fits all data at once;
**3.** `biglm()` with chunks of data to be updated by `update()`. The final method would be chosen according to efficiency and the amount of useful information that the method can deliver.

## 6 Results

### 6.1 The relationship between `Change` and probability of fatal or serious crashes `crashSeverity`

#### 6.1.1 Unadjusted model

The unadjusted model only included `Change` as the explanatory variable.

```
# Build the model
expr1 <- expression(glm(I(crashSeverity %in% c("F", "S")) ~ Change,
                        family = "binomial"))
# Apply the expression to each of the 10 copies of the imputed data
## fit_unadj <- with(imp_all, expr1)
## save(fit_unadj, file = "fit_unadj.Rdata")
load("fit_unadj.Rdata")

# Pool the estimates
sum_unadj <- pool(fit_unadj) %>% summary(conf.int = TRUE)
sum_unadj %>% mutate_if(is.numeric, function(x)(round(x,4))) %>%
  kbl(align = c("l", rep("c",7)), booktabs=T)
```

| term | estimate | std.error | statistic | df | p.value | 2.5 % | 97.5 % |
|---|---|---|---|---|---|---|---|
| (Intercept) | -2.6885 | 0.0061 | -439.3599 | 776125.4 | 0.0000 | -2.7005 | -2.6766 |
| Change1 | 0.0360 | 0.0147 | 2.4401 | 776125.4 | 0.0147 | 0.0071 | 0.0649 |
| Change2 | 0.0920 | 0.0101 | 9.0880 | 776125.4 | 0.0000 | 0.0721 | 0.1118 |

```
# Extract the confidence intervals
confint_unadj <- sum_unadj %>% select(`2.5 %`, `97.5 %`)
cbind(term = sum_unadj[,"term"], 100*(exp(confint_unadj)-1)) %>% kbl(booktabs=T)
```

| term | 2.5 % | 97.5 % |
|---|---|---|
| (Intercept) | -93.2830629 | -93.12000 |
| Change1 | 0.7105809 | 6.70354 |
| Change2 | 7.4799527 | 11.82926 |

As shown in the unadjusted model, the coefficients of `Change1` and `Change2` were both highly significant ($p$-value close to 0).
The model equation was:

$$\text{logit}(p_i) = \beta_0 + \beta_1 \times \text{Change1}_i + \beta_2 \times \text{Change2}_i$$

$$Y_i \sim \text{Binomial}(n_i, p_i)$$

where for the $i^{\text{th}}$ crash in the data, $p_i$ denotes the probability of fatal or serious crash. $\text{Change1}_i$ was a dummy variable that was 1 if the crash took place in the post-first change period (2012-2014), otherwise 0. $\text{Change2}_i$ was a dummy variable that was 1 if the crash was in the post-second change period (2015-2021), otherwise 0.
For the unadjusted model, the probabilities of serious or fatal cases increased after the 2 changes. Let's look at the adjusted model.

### 6.1.2 Adjusted Model

The selection of variables included in the adjusted model involved parallel computation[4] and LASSO with 1-SE rule(`glmnet` package[2]). For more details, please see the section Adjusted Model. The final adjusted model was shown here:

```
expr2 <- expression(glm(I(crashSeverity %in% c("F", "S")) ~
                          Change + bicycle + cliffBank + motorcycle
                        + NumberOfLanes + roadLane + speedLimit
                        + streetLight + train + truck + waterRiver
                        + weatherA, family = "binomial"))
## fit_adj <- with(imp_all, expr2)
## save(fit_adj, file = "fit_adj.Rdata")
load("fit_adj.Rdata")

sum_adj <- pool(fit_adj) %>% summary(conf.int = TRUE)
sum_adj %>% mutate_if(is.numeric, function(x)(round(x,4))) %>%
  kbl(align = c("l", rep("c",7)), booktabs = T, digits = 3)
```

| term | estimate | std.error | statistic | df | p.value | 2.5 % | 97.5 % |
|---|---|---|---|---|---|---|---|
| (Intercept) | -4.833 | 0.041 | -117.664 | 23829.546 | 0.000 | -4.914 | -4.753 |
| Change1 | -0.027 | 0.015 | -1.750 | 739118.446 | 0.080 | -0.057 | 0.003 |
| Change2 | 0.054 | 0.011 | 5.073 | 40334.152 | 0.000 | 0.033 | 0.075 |
| bicycle | 1.589 | 0.019 | 82.858 | 48157.041 | 0.000 | 1.552 | 1.627 |
| cliffBank | 0.109 | 0.021 | 5.079 | 35.982 | 0.000 | 0.066 | 0.153 |
| motorcycle | 2.003 | 0.015 | 135.223 | 227283.707 | 0.000 | 1.974 | 2.032 |
| NumberOfLanes | -0.130 | 0.006 | -20.098 | 10997.001 | 0.000 | -0.143 | -0.117 |
| roadLane2-way | 1.081 | 0.024 | 44.228 | 18422.447 | 0.000 | 1.033 | 1.129 |
| roadLaneNull | 0.826 | 0.248 | 3.326 | 629656.885 | 0.001 | 0.339 | 1.312 |
| roadLaneOff road | 1.678 | 0.046 | 36.724 | 67986.188 | 0.000 | 1.588 | 1.767 |
| speedLimit | 0.020 | 0.000 | 77.246 | 1840.222 | 0.000 | 0.019 | 0.020 |
| streetLightNull | -0.234 | 0.013 | -17.544 | 163149.772 | 0.000 | -0.260 | -0.208 |
| streetLightOff | -0.468 | 0.018 | -26.731 | 35861.969 | 0.000 | -0.502 | -0.433 |
| streetLightOn | 0.015 | 0.017 | 0.887 | 223074.756 | 0.375 | -0.018 | 0.049 |
| train | 1.123 | 0.151 | 7.438 | 23.084 | 0.000 | 0.811 | 1.435 |
| truck | 0.337 | 0.015 | 22.495 | 421056.271 | 0.000 | 0.308 | 0.366 |
| waterRiver | 0.626 | 0.082 | 7.616 | 15.506 | 0.000 | 0.451 | 0.801 |
| weatherAHail or Sleet | 0.174 | 0.328 | 0.532 | 771964.408 | 0.595 | -0.468 | 0.817 |
| weatherAHeavy rain | -0.218 | 0.025 | -8.685 | 746897.086 | 0.000 | -0.267 | -0.169 |
| weatherALight rain | -0.224 | 0.014 | -15.671 | 607242.994 | 0.000 | -0.252 | -0.196 |
| weatherAMist or Fog | -0.148 | 0.037 | -3.962 | 745445.467 | 0.000 | -0.220 | -0.075 |
| weatherANull | -1.054 | 0.064 | -16.334 | 773904.260 | 0.000 | -1.181 | -0.928 |
| weatherASnow | -0.282 | 0.103 | -2.742 | 687901.572 | 0.006 | -0.483 | -0.080 |

```
# Extract the confidence intervals
confint_adj <- sum_adj %>%
  filter(term %in% c("Change1", "Change2")) %>% select(`2.5 %`, `97.5 %`)
```

```
cbind(term=c("Change1", "Change2"),100*(exp(confint_adj)-1)) %>%
  kbl(booktabs = T, digits = 3)
```

| term | 2.5 % | 97.5 % |
|---|---|---|
| Change1 | -5.544 | 0.324 |
| Change2 | 3.386 | 7.814 |

After being adjusted by other variables, the coefficient of `Change2` was significant, but not for `Change1`.

Comparing the the baseline period, there was 95% confidence that the odds of fatal or serious cases after the first reduction of alcohol limit in 2014 increased by somewhere between 3.39% to 7.81%.
No significant changes were seen after the first reduction in 2011, comparing to the baseline.

In this case, only the first change has led to the reduction of the odds of fatal or serious crash.

## 6.2 The relationship between `crashYear` and minor injury counts per case `minorInjuryCount`

For the methods for linear regression proposed in last section, there were no significant differences in speed for direct `lm()` or `biglm()` functions.
The method of updating `biglm()` by chunks of data was slower, due to the time and memory used to save the subsets of the data.
Between `lm()` and `biglm()`, the former was more desirable because it conveyed relevant information for model fitness, such as `R-sqaured`.

### 6.2.1 Unadjusted Piece-wise Linear Model

For the unadjusted model, only `crashYear` and the corresponding piece-wise linear terms were included.

```
p1 <- pmin(pmax(data$crashYear - 2011, 0), 3)
p2 <- pmin(pmax(data$crashYear - 2014, 0), 2)
p3 <- pmax(data$crashYear - 2016, 0)

# Loop around all the 10 imputations
fit_lm_unadj <- list(10)
for(i in 1:10){
  comp <- comp_long %>% filter(`.imp` == i)
  fit_lm_unadj[[i]] <- lm(minorInjuryCount ~ crashYear
                          + p1 + p2 + p3, data = comp)
}

mira_lm_unadj <- fit_lm_unadj %>% as.mira
```

```
# Pool the estimates for confidence intervals
pool(mira_lm_unadj) %>%
  summary(conf.int = TRUE) %>%
  kbl(booktabs = T, digits = 3)
```

| term | estimate | std.error | statistic | df | p.value | 2.5 % | 97.5 % |
|---|---|---|---|---|---|---|---|
| (Intercept) | -8.049 | 0.552 | -14.589 | 776123.4 | 0 | -9.130 | -6.967 |
| crashYear | 0.004 | 0.000 | 15.154 | 776123.4 | 0 | 0.004 | 0.005 |
| p1 | -0.007 | 0.001 | -5.489 | 776123.4 | 0 | -0.010 | -0.005 |
| p2 | -0.035 | 0.002 | -18.458 | 776123.4 | 0 | -0.039 | -0.032 |
| p3 | 0.024 | 0.001 | 27.293 | 776123.4 | 0 | 0.023 | 0.026 |

All the piece-wise linear terms were highly significant.
The model equation was:

$$\mu_i = \beta_0 + \beta_1 \times \text{crashYear}_i + \beta_2 \times \text{p1}_i + \beta_3 \times \text{p2}_i + \beta_4 \times \text{p3}_i$$

$$Y_i \sim \text{Normal}(\mu_i, \sigma^2)$$

where for the $i^{\text{th}}$ crash in the data, $\mu_i$ was the average number of minor injuries per crash. $\text{p1}_i$ denoted the period between 2011–2014 where we assumed a gradient change since 2011. $\text{p2}_i$ and $\text{p3}_i$ were defined in similar ways that represented 2014–2016 and 2016–2021, respectively.
The gradient changes at 2011 and 2014 were significant, which may or may not result from policy changes. In addition, the R_squared showed the model only explained around 0.1% variability in the response. It would be necessary to adjust the model by other covariates, as shown in the next section.

### 6.2.2 Adjusted Piece-wise Linear Model

The variables included in the adjusted model were selected using LASSO, and the detailed process was listed here. Besides the three piece-wise-linear terms, the variables `bicycle`, `cliffBank`, `motorcycle`, `roadLane`, `speedLimit`, `streetLight` and `weatherA` would be included in the final model.
The model was fitted to each copy of data, and the estimates were pooled together.

```
# Fit model for 10 copies of data
fit_lm_adj <- list(10)
for(i in 1:10){
  comp <- comp_long %>% filter(`.imp` == i) %>%
    select(all_of(var_pred_lm), minorInjuryCount)
  fit_lm_adj[[i]] <- lm(minorInjuryCount ~ crashYear
                        + p1 + p2 + p3
                        + bicycle + cliffBank + motorcycle + roadLane
                        + speedLimit + streetLight + truck + weatherA,
                          data = comp)
}
mira_lm_adj <- fit_lm_adj %>% as.mira
```

```
# Pool the estimates and show confidence intervals
pool(mira_lm_adj) %>% summary(conf.int = TRUE) %>% kbl(booktabs = T, digits = 3)
```

| term | estimate | std.error | statistic | df | p.value | 2.5 % | 97.5 % |
|---|---|---|---|---|---|---|---|
| (Intercept) | -5.898 | 0.548 | -10.764 | 268114.202 | 0.000 | -6.972 | -4.824 |
| crashYear | 0.003 | 0.000 | 10.765 | 260433.789 | 0.000 | 0.002 | 0.003 |
| p1 | -0.007 | 0.001 | -5.485 | 378106.297 | 0.000 | -0.010 | -0.005 |
| p2 | -0.032 | 0.002 | -16.859 | 442440.011 | 0.000 | -0.036 | -0.028 |
| p3 | 0.030 | 0.001 | 32.251 | 279160.739 | 0.000 | 0.028 | 0.032 |
| bicycle | 0.377 | 0.004 | 89.215 | 503979.580 | 0.000 | 0.369 | 0.385 |
| cliffBank | 0.066 | 0.005 | 12.305 | 16.072 | 0.000 | 0.055 | 0.077 |
| motorcycle | 0.193 | 0.004 | 51.461 | 529468.785 | 0.000 | 0.186 | 0.201 |
| roadLane2-way | 0.093 | 0.003 | 34.461 | 1992.333 | 0.000 | 0.088 | 0.098 |
| roadLaneNull | 0.009 | 0.029 | 0.301 | 630450.653 | 0.763 | -0.047 | 0.065 |
| roadLaneOff road | 0.129 | 0.007 | 19.140 | 18352.039 | 0.000 | 0.116 | 0.142 |
| speedLimit | 0.003 | 0.000 | 73.644 | 294.086 | 0.000 | 0.003 | 0.003 |
| streetLightNull | 0.020 | 0.002 | 7.905 | 11774.310 | 0.000 | 0.015 | 0.025 |
| streetLightOff | -0.013 | 0.003 | -4.724 | 1637.667 | 0.000 | -0.019 | -0.008 |
| streetLightOn | 0.000 | 0.003 | 0.128 | 11934.699 | 0.898 | -0.005 | 0.006 |
| truck | -0.052 | 0.003 | -20.437 | 598064.628 | 0.000 | -0.057 | -0.047 |
| weatherAHail or Sleet | -0.125 | 0.067 | -1.876 | 757904.447 | 0.061 | -0.256 | 0.006 |
| weatherAHeavy rain | -0.005 | 0.004 | -1.308 | 704714.055 | 0.191 | -0.012 | 0.002 |
| weatherALight rain | -0.006 | 0.002 | -3.058 | 511269.102 | 0.002 | -0.010 | -0.002 |
| weatherAMist or Fog | -0.008 | 0.006 | -1.351 | 765050.781 | 0.177 | -0.021 | 0.004 |
| weatherANull | -0.213 | 0.006 | -37.874 | 716221.602 | 0.000 | -0.224 | -0.202 |
| weatherASnow | -0.022 | 0.016 | -1.340 | 762023.366 | 0.180 | -0.054 | 0.010 |

After adjustment, the three terms `p1`, `p2` and `p3` were still highly significant. The coefficients were slightly smaller than the unadjusted model.

Our goal was to find out the underlying linkage between alcoholic changes and car crashes. Thus, the key in this model was the interpretation of the piece-wise linear terms. We assumed, if the restricted drinking driving regulations were effective, the minor injuries should show significant negative trend from 2011 and 2014.

The term `p1` showed that, on top of the general trend before 2011, there was a significant gradient change since then. With 95% confidence, there was a further reduction of somewhere between 0.5 to 1 minor injuries per 100 cases. Similarly, the coefficient of `p2` showed that there was greater reduction in the gradient between 2014–2016, comparing the Post-first change period. Compared to the general trend before 2014, we have 95% confidence that there was a further reduction of somewhere between 2.8 to 3.2 minor injuries per 100 cases.

The `R-squared` of this model was as low as 3% (see Adjusted Model). The covariates in the model were not sufficient to explain the fluctuations in the minor injury counts. The results, however, were as far as I could achieve using a combination of linear regression and LASSO. The further improvements would be proposed in the Discussion section.

# 7 Discussion

Drinking driving was a long-standing controversial topic. Followed by years of frequent drinking-related deaths on road, the government decided to reduce the legal alcohol limits. The first change in 2011 restricted zero alcohol intake for those under 20 years old, while the second change in 2014 further reduced the upper limits for those over 20. The goal of our project was to analyse the crash data and determine whether the changes in policies alleviated the accidents on road.
The data published by NZTA had an integrated structure, however, high degrees of missingness. This saved the efforts to tidy up the values present in the data. But dealing with the `NA`'s required much more efforts than I have expected. In the meantime, the data contained more than 700 thousands rows that covered the period 2000–2021. This was beneficial to analyse the effects of the two changes on a larger timescale. On the other hand, each modification or computation around the data set would take a lot of RAM, and special techniques were required to handle the large data. To make the most use of the variables, Multiple Imputations were implemented. To save computational powers, only certain variables were imputed: those significantly correlated or had significant contributions to the response variables. From the diagnostics, there were no significant issues with the imputation data (10 copies). To explain the relationships between the policy changes and crash data, supervised learning method was implemented. The `lm()` and `glm()` methods were chosen because it would be possible to obtain the coefficients and interpret the effects. In contrast, other black-box methods (e.g. tree, random forest) focused more on the predictive accuracy of the model, which did not match our purpose.
For model selection, a combination of LASSO and parallel computation was used. For the probabilities of fatal cases, we concluded that the two reductions in drinking alcohol limits negatively affected the probabilities of fatal cases in New Zealand. Furthermore, the effect of the second reduction seemed to be less effective than the first one.
In terms of minor injury counts per 100 accidents, we also saw significant reductions from 2011–2014 and 2014–2016. The second reduction would be A major drawback of the adjusted model, however, was the low `R-squared` around 3%. For future improvements, LASSO with 1-SE rule might be too harsh on model selection and more variables would be needed to explain the variability in the response.

There were, however, more limitations in the analysis:
1. The categorization of the variable `Change` was inaccurate because the time of the case was only up to year level. This would cause bias in analysis;

2. The large number of missing values in the data would cause bias. The Multiple Imputation method was an approximation of the complete data;

3. For improvements, the models should be adjusted by other variables. For example, the number of cars on the road each year would contribute to the fluctuations in the number/severity of the car accidents. Also, on a larger time scale, the Financial Crisis started in 2008, and the global COVID pandemic might also affect vehicle usage and crash data.

4. There were no variables that directly link to alcoholic states of the drivers involving in the crash. Therefore, we had to analyse the effects of policy changes by gradients of changes of injuries on a longer time periods. Any conclusion drawn from there would not prove the direct contributions of the restricted alcohol limits.

We may conclude that, on a larger timescale, there were reductions in minor injuries counts beyond 2011 and 2014, but not for the probabilities of serious or fatal cases. Restrictions on drinking driving policies may or may not directly played a role in the process (as discussed above). We hope they were effective, or at least, they had raised the awareness of the public on drinking driving.

# 8 Appendix

## 8.1 Data Importation and Transformation

```
# Data Importation
pacman::p_load(tidyverse, magrittr, data.table, conflicted, here, naniar, mice,
        ragg, ggcorrplot, dagitty, glmnet, parallel, doParallel, knitr,
        kableExtra, biglm, DBI)
data_raw <- fread(here("Crash_Analysis_System_(CAS)_data.csv"))
dim(data_raw)

# Resolve conflicts of functions
conflict_prefer("select", "dplyr")
conflict_prefer("filter", "dplyr")
conflict_prefer("extract", "tidyr")

# Transformation
data <- data_raw %>%
  transform(weatherA = as.factor(weatherA),
            weatherB = as.factor(weatherB),
            urban = as.factor(urban),
            trafficControl = as.factor(trafficControl),
            streetLight = as.factor(streetLight),
            roadSurface = as.factor(roadSurface),
            roadLane = as.factor(roadLane),
            roadCharacter = as.factor(roadCharacter),
            light = as.factor(light),
            holiday = as.factor(holiday),
            flatHill = as.factor(flatHill),
            crashSHDescription = as.factor(crashSHDescription),
            crashDirectionDescription = as.factor(crashDirectionDescription),
            crashSeverity = ordered(levels = c('N','M','S','F'),
              case_when(crashSeverity == 'Non-Injury Crash' ~ 'N',
                        crashSeverity == 'Minor Crash' ~ 'M',
                        crashSeverity == 'Serious Crash' ~ 'S',
                        crashSeverity == 'Fatal Crash' ~ 'F')),
            crashLocation1 = as.factor(crashLocation1),
            crashLocation2 = as.factor(crashLocation2),
            directionRoleDescription = as.factor(directionRoleDescription),
            tlaName = as.factor(tlaName),
            region = as.factor(region)) %>%
  mutate(baseline = ifelse(crashYear <= 2011,1,0),
         firstChange = ifelse(crashYear %in% 2012:2014,1,0),
         secondChange = ifelse(crashYear > 2014,1,0),
         Change = as.factor(case_when(crashYear <= 2011 ~ 0,
                                      crashYear %in% 2012:2014 ~ 1,
                                      crashYear > 2014 ~ 2)))
```

To deal with the `NA`'s in the injury counts, I referred to the outcome variable `crashSeverity` that recorded the level of the most severe case in the crash. the main concept is:

- If `crashSeverity = "Non-Injury Crash"`, all the injury counts were 0.
  That is, `minorInjuryCount = seriousInjuryCount = fatalInjuryCount = 0`.
- If `crashSeverity = "Minor Crash"`, then all the other injury counts (`seriousInjuryCount = fatalInjuryCount = 0`);
- If `crashSeverity = "Serious Crash"`, the fatal counts was 0 (`fatalInjuryCount = 0`).

```
# Deal with missing values in injury counts
complete_data <- data %>% select(crashSeverity, seriousInjuryCount, fatalCount,
                                 minorInjuryCount) %>% complete.cases()
# Replace NA's with 0
data[!complete_data,] <- data[!complete_data,] %>%
```

```
  mutate(minorInjuryCount = if_else(crashSeverity == "N", 0,
                                    as.double(minorInjuryCount)),
         seriousInjuryCount = if_else(crashSeverity %in% c("N", "M"), 0,
                                      as.double(seriousInjuryCount)),
         fatalCount = if_else(crashSeverity %in% c("N", "M", "S"), 0,
                              as.double(fatalCount)))

data %>% select(crashSeverity, seriousInjuryCount, fatalCount,
                                  minorInjuryCount) %>% miss_var_summary()
```

After the modification, nearly all the `NA`'s in the injury counts have been resolved. The single missing value in `minorInjuryCount` would be imputed in Multiple Imputation.

## 8.2 Correltion Analysis

```
# Create dataset to detect multicollinearity
## collin <- data %>% lapply(as.numeric) %>% data.frame()

# Correlations between variables in pairs, after removing missing values
## cor_collin <- collin %>% cor(use='pairwise.complete.obs')
## save(cor_collin, file= "cor_collin_all.Rdata")

load("cor_collin_all.Rdata")
# Inspect correlations for all
ggcorrplot(cor_collin, title = "Correlation plot of the all relevant variables") +
  theme(axis.text.x = element_text(size = 5),
        axis.text.y = element_text(size = 5))
```

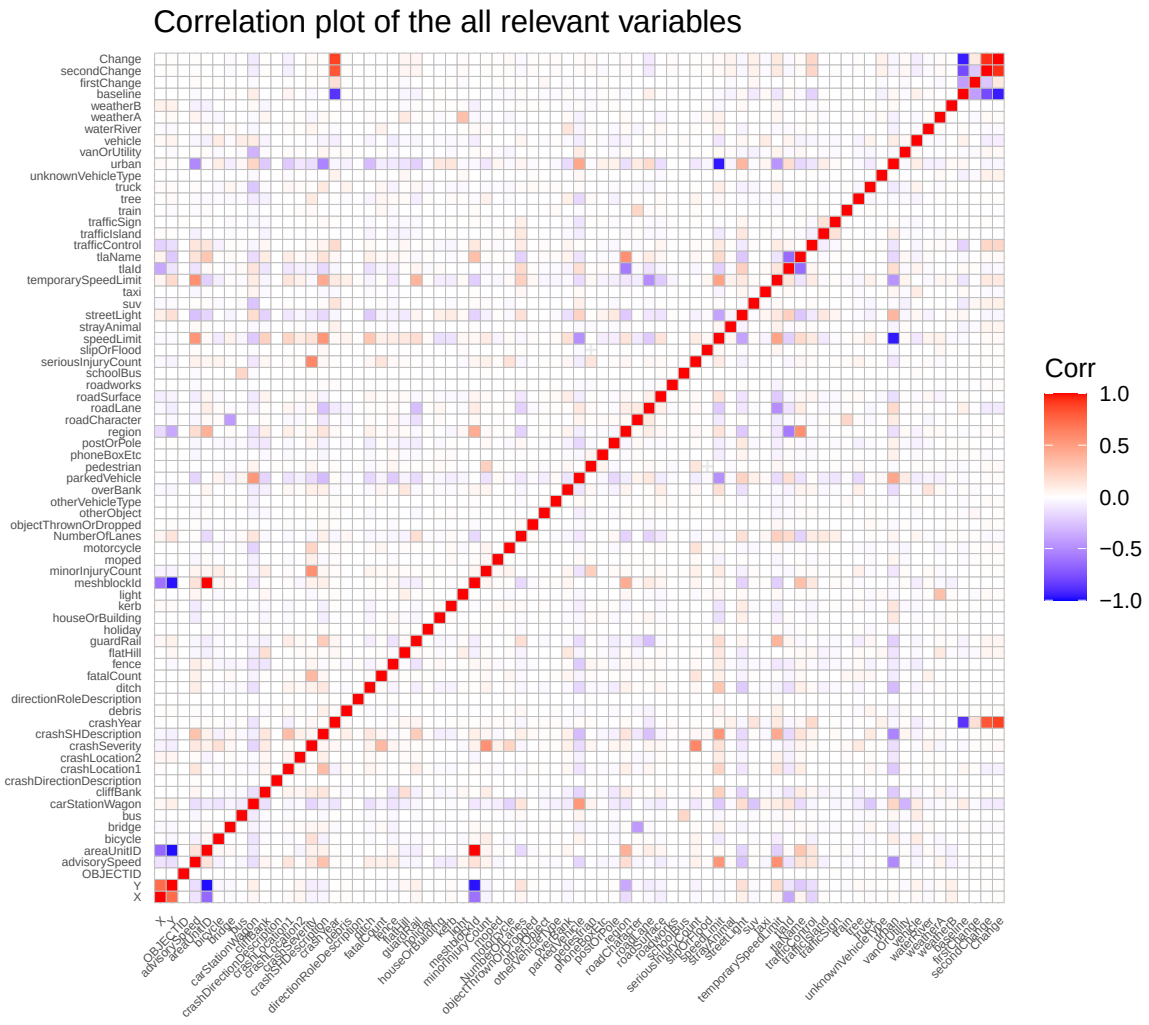


Fig 4. Correlation matrix of all variables in the data set.

## 8.3 Multiple Imputation with *mice* [5] package

### 8.3.1 Variables included in the Multiple Imputation

```
## 1. Variables that have at least 0.2 correlation with the key outcomes
var_miss <- miss_var_summary(data_clean) %>%
  filter(pct_miss > 0 & pct_miss < 70) %>%
  select(variable) %>%
  unlist %>% unname

# Correlation
collin_mid <- data_clean %>%
  select(all_of(var_miss), crashSeverity, minorInjuryCount, seriousInjuryCount,
         fatalCount) %>%
  lapply(as.numeric) %>%
  data.frame()

# Correlations between variables in pairs, after removing missing values
# cor_collin_mid <- collin_mid %>% cor(use='pairwise.complete.obs')
# save(cor_collin_mid, file= "cor_collin_mid.Rdata")
load("cor_collin_mid.Rdata")

cor_outcome <- cor_collin_mid[c("crashSeverity","minorInjuryCount",
                                "seriousInjuryCount", "fatalCount"),]
ind <- apply(cor_outcome, 2, function(x) any(x > .2))
var_high_cor <- colnames(cor_outcome)[ind]

## 2. Key variables identified using common sense
road_conditions <- c("roadSurface", "NumberOfLanes", "speedLimit", "light",
                     "roadLane", "streetLight", "trafficControl", "flatHill",
                     "roadworks")
location <- c("urban", "region")

other_vehicles <- c("bicycle", "train", "slipOrFlood", "vehicle", "schoolBus",
                    "truck", "vanOrUtility", "bus")

time <- c("Change", "crashYear")
weather <- c("weatherA", "weatherB")
other_objects <- c("strayAnimal","bridge", "waterRiver", "cliffBank")

# Key predictors
var_key_pred <- c(road_conditions, location, other_vehicles, time, weather,
                  other_objects)


## 3. Complete variables in the original data set
var_complete <- miss_var_summary(data_clean) %>%
  filter(pct_miss == 0) %>%
  select(variable) %>%
  unlist %>% unname

## 4. Outcome variables
var_outcome <- c("crashSeverity", "fatalCount", "seriousInjuryCount", "minorInjuryCount")

# Variables to be included in the model
var_model <- c(var_outcome, var_high_cor, var_key_pred, var_complete) %>%
  unique()
# Irrelevant variables
var_irrelevant <- setdiff(names(data_clean), var_model)
```

### 8.3.2 Dry Run & Modification of methods, prediction matrix and post-processing

A dry run was performed a dry run to check for the initial settings in the Multiple Imputation. From there, it was possible to modify and supply the methods, prediction matrix and post-processing in the

real imputation.

```
# Dry run
## dry_run <- mice(data_clean, maxit = 0)
## save(dry_run, file = "dry_run.Rdata")
load("dry_run.Rdata")

## pred_all <- quickpred(data_clean)
## save(pred_all, file= "pred_all.Rdata")
load("pred_all.Rdata")
```

### Method

The default methods were used for the relevant variables, as selected by `mice()` function.
In addition, `mice` could not handle variables with more than 50 unique values. Those variables would be be imputed, by setting `methods` to empty.

```
# mice package cannot handle variables with more than 50 unique values
var_name <- names(data_clean)
## Do not impute variables with more than 50 unique values
var_len_uniq <- data_clean %>%
  sapply(function(x) length(unique(x)))

# Method
meth_all <- dry_run$method
meth_all[all_of(var_irrelevant)] <- ""
meth_all[var_name[var_len_uniq > 50]] <- ""
```

### Prediction Matrix

A few modifications were made to the prediction matrix:

1. Not predict irrelevant variables or use them to predict anything;
2. Not predict variables with more than 50 unique values, or use them to predict anything;
3. Do not impute variables derived from `Change` because of the high multi-collinearity with `Change`. Otherwise, this would cause errors in Multiple Imputation;
4. Set the diagonal to 0 such that the variables would not predict themselves.

```
## pred_all <- quickpred(data_clean)
## save(pred_all, file= "pred_all.Rdata")
load("pred_all.Rdata")

# Not impute irrelevant variables or use them as predictors
pred_all[all_of(var_irrelevant), ] <- 0
pred_all[,all_of(var_irrelevant)] <- 0


# Do not impute variables with more than 50 unique values or predict anything
pred_all[var_name[var_len_uniq > 50],] <- 0
pred_all[,var_name[var_len_uniq > 50]] <- 0

# Not use variables derived from 'Change' due to high collinearity with Change
pred_all[,c("baseline", "firstChange", "secondChange")] <- 0

# Not use crashYear to predict due to high collinearity with Change
pred_all[,"crashYear"] <- 0
pred_all["crashYear",] <- 0

# Remove predictions that do not make sense
```

```r
pred_all[c("waterRiver", "cliffBank"), "bicycle"] <- 0
# ------------------------------------------------------------------------------
# Make the diagonal elements 0
diag(pred_all) <- 0
```

### Post-processing

The post-processing was done for the numeric variables, to ensure that the predicted values stayed in the realistic bound.

```r
# Variables to constrain the upper and lower limit
numeric_vars <- data_clean %>%
  select(where(is.numeric)) %>%
  names()
integer_vars <- data_clean %>%
  select(where(is.integer)) %>%
  names()
num_int_vars <- c(numeric_vars, integer_vars)

# Extract post
post <- dry_run$post

# Function takes variable name and creates a new squeezed variable,
# if it is already present in meth_all and is numeric

squeeze_post <- function(var_name) {
  if (meth_all[var_name] != '' & var_name %in% num_int_vars) {
    post_value <- paste0("imp[[j]][, i] <- squeeze(imp[[j]][, i], c(",
                         min(data_clean[[var_name]], na.rm=TRUE), # Realistic lower bound
                         ", ",
                         2 * max(data_clean[[var_name]], na.rm=TRUE), # Realistic upper bound
                         "))")
    return(post_value)
  } else {
    return('')
  }
}

# Add squeezes to post-processing variable, to pass to mice()
for (i in 1:length(post)) {
  post[names(post)[i]] <- squeeze_post(names(post)[i])
}
```

## 8.4 Run Multiple Imputation

There were 20 iterations in each of the 10 imputations. The imputed data were extracted and stored.

```r
# Parameters
maxit_n <- 20 # maximal iterations
seed <- 765
m <- 10 # Number of imputation

# Run the imputation
## imp_all <- mice(data_clean, pred = pred_all, method = meth_all,
##                 post = post, m = m, maxit = maxit_n, seed = seed)
## save(imp_all, file = "imp_all.Rdata")

# Save data in long format
## comp_long <- complete(imp_all, action = "long")
## save(comp_long, file = "comp_long.Rdata")

# Load the imputation data
```

```
load("imp_all.Rdata")
load("comp_long.Rdata")
```

## 8.5 Diagnosis of Multiple Imputation

### Missing Values in the Imputed Variables

```
# Variables imputed in mice
var_imp <- row.names(pred_all)[rowSums(pred_all) > 0]
gg_miss_var(comp_long %>% select(all_of(var_imp)), show_pct = T)+
  labs(title="Percentages of missing values") +
  theme(axis.text.y = element_text(size = 7))
```


Fig 5. The number of missing values in the imputed variables in the post-imputation data.

As shown in Fig 6., all the imputed variables were complete.

### Convergence

The convergence of the imputation was analyzed by plotting the 10 streams in the plot. Each was independent without any trend. Therefore, there was no particular concern about the convergence. Here was an example of the first four imputed variables:

## 8.6 Model for the Policy Change and Probability of Fatal or Serious Crash

### 8.6.1 Unadjusted Model

```
# Build the model
expr1 <- expression(glm(I(crashSeverity %in% c("F", "S")) ~ Change, family = "binomial"))
# Apply the expression to each of the 10 copies of the imputed data
## fit_unadj <- with(imp_all, expr1)
## save(fit_unadj, file = "fit_unadj.Rdata")
load("fit_unadj.Rdata")
```

```
summary(pool(fit_unadj))

# Confidence intervals
```

```
confint_adj <- pool(fit_adj) %>% summary(conf.int = TRUE) %>%
  select(`2.5 %`, `97.5 %`)
cbind(term = sum_unadj[,"term"], 100*(exp(confint_unadj)-1)) %>%
  kbl(booktabs = T))

## Error:  <text>:7:20:  unexpected ')'
## 6:  cbind(term = sum_unadj[,"term"], 100*(exp(confint_unadj)-1)) %>%
## 7:    kbl(booktabs = T))
##                         ^
```

### 8.6.2 Adjusted Model

Here we identified a list of key predictors to be included in the adjusted model.

**Model Selection**
The long format of the imputed data included more than 7 million rows. For each copy of data, LASSO would be used to select the best set of predictors and Using 1-SE rule, `motorcycle` and `speedLimit` should be included in the model. The other key variables would also be included here.

```
# Key predictors
var_pred <- setdiff(c(var_imp, var_complete),
                    c(var_outcome, "baseline", "firstChange", "secondChange",
                      "crashYear"))

# Function to be passed into parallel computation
# Model selection using LASSO
select_model <- function(...){
  library(glmnet)
  library(Matrix)
  library(tidyverse)
  i <- (...)
  data_model <- comp_long %>%
    filter(`.imp` == i) %>%
    select(all_of(var_pred), crashSeverity)
  X <- data_model %>%
    select(all_of(var_pred)) %>%
    sapply(as.numeric) %>%
    Matrix()
  y <- data_model %>%
    mutate(Severity_FS = if_else(crashSeverity %in% c("F", "S"), 1, 0)) %>%
    select(Severity_FS) %>% unlist %>% unname
  fit <- glmnet(X, y, family = "binomial")
  xval <- cv.glmnet(X, y)
  return(coef(fit, s = xval$lambda.1se))}


# Use Parallel Computation for model selection using LASSO

# Make new cluster
## cl <- makeCluster(5)

# Export the objects to each cluster
## clusterExport(cl, c('select_model', 'comp_long', "var_pred"))

# Do the computation
# par_output <- parLapply(cl, 1:m, fun = select_model)

# Stop the cluster
## stopCluster(cl)

## save(par_output, file = "model_selection_fit.Rdata")


# The outcome of LASSO was returned as a list
```

```
load("model_selection_fit.Rdata")
```

The occurrences of each variable in the 10 models were listed in the table. The majority method for pooling the variables was described in Section 5.4 of the `mice` book[1]. In another word, the more frequent a variable was present in the models, the more important it would be.

```
# Turn the list as a mira object
fit_all <- par_output %>% as.mira

# Find the names of variables where the coefficients were not 0
terms <- lapply(fit_all$analyses, function(x) row.names(x)[which(x != 0)])
# Use majority rule
votes <- unlist(terms)
table(votes)
```

```
## votes
##   (Intercept)       bicycle     cliffBank    motorcycle NumberOfLanes      roadLane    speedLimit
##            10            10             7            10            10            10            10
```

The variables `bicycle`, `cliffbank`, `motorcycle`, `NumberOfLanes`, `roadLane`, `speedLimit`, `streetLight`, `train`, `truck`, `waterRiver` and `weatherA` would be included in the final model, together with the key variable `Change`.

### Final Adjusted Model

```
expr2 <- expression(glm(I(crashSeverity %in% c("F", "S")) ~
                          Change + bicycle + cliffBank + motorcycle
                        + NumberOfLanes + roadLane + speedLimit
                        + streetLight + train + truck + waterRiver
                        + weatherA, family = "binomial"))
## fit_adj <- with(imp_all, expr2)
## save(fit_adj, file = "fit_adj.Rdata")
load("fit_adj.Rdata")
```

```
summary(pool(fit_adj))
```

```
##                    term    estimate    std.error    statistic           df      p.value
## 1           (Intercept) -4.83318497 0.0410763303 -117.6635043  23829.54643 0.000000e+00
## 2               Change1 -0.02690148 0.0153740380   -1.7497991 739118.44579 8.015341e-02
## 3               Change2  0.05426699 0.0106974788    5.0728764  40334.15253 3.935817e-07
## 4               bicycle  1.58949626 0.0191832976   82.8583434  48157.04131 0.000000e+00
## 5             cliffBank  0.10937965 0.0215348154    5.0792008     35.98169 1.180631e-05
## 6            motorcycle  2.00334503 0.0148150727  135.2234349 227283.70671 0.000000e+00
## 7          NumberOfLanes -0.13000996 0.0064689497  -20.0975384  10997.00103 0.000000e+00
## 8          roadLane2-way  1.08081771 0.0244374407   44.2279421  18422.44694 0.000000e+00
## 9           roadLaneNull  0.82576314 0.2482450720    3.3264030 629656.88488 8.797967e-04
## 10      roadLaneOff road  1.67754095 0.0456793923   36.7242396  67986.18815 0.000000e+00
## 11            speedLimit  0.01986797 0.0002572042   77.2459165   1840.22183 0.000000e+00
## 12       streetLightNull -0.23434224 0.0133575906  -17.5437506 163149.77176 0.000000e+00
## 13        streetLightOff -0.46759189 0.0174925856  -26.7308621  35861.96917 0.000000e+00
## 14         streetLightOn  0.01518109 0.0171078311    0.8873769 223074.75567 3.748770e-01
## 15                train  1.12304663 0.1509919975    7.4377891     23.08423 1.429401e-07
## 16                truck  0.33693694 0.0149786028   22.4945506 421056.27114 0.000000e+00
## 17           waterRiver  0.62602739 0.0821944881    7.6164157     15.50567 1.272521e-06
## 18 weatherAHail or Sleet  0.17441121 0.3279228855    0.5318665 771964.40822 5.948186e-01
## 19    weatherAHeavy rain -0.21800137 0.0250998322   -8.6853716 746897.08582 0.000000e+00
## 20    weatherALight rain -0.22382457 0.0142824250  -15.6713284 607242.99397 0.000000e+00
## 21   weatherAMist or Fog -0.14754216 0.0372366854   -3.9622796 745445.46732 7.424444e-05
## 22          weatherANull -1.05416958 0.0645372993  -16.3342685 773904.25970 0.000000e+00
## 23          weatherASnow -0.28165539 0.1027338414   -2.7416028 687901.57198 6.114180e-03
```

```r
# Confidence intervals
confint_adj <- pool(fit_adj) %>% summary(conf.int = TRUE) %>%
  filter(term %in% c("Change1", "Change2")) %>%
  select(`2.5 %`, `97.5 %`)
cbind(term = c("Change1", "Change2"), 100*(exp(confint_adj)-1)) %>%
  kbl(booktabs = T)
```

| term | 2.5 % | 97.5 % |
|---|---|---|
| Change1 | -5.543813 | 0.3236359 |
| Change2 | 3.386033 | 7.8136718 |

## 8.7 Model for the Policy Change and Counts of Minor Injuries per case , piece-wise linear model

### 8.7.1 Unadjusted Model

```r
# Loop around all the 10 imputations
fit_lm_unadj <- list(10)
for(i in 1:10){
  comp <- comp_long %>% filter(`.imp` == i)
  fit_lm_unadj[[i]] <- lm(minorInjuryCount ~ crashYear
                          + p1 + p2 + p3, data = comp)
}

mira_lm_unadj <- fit_lm_unadj %>% as.mira


# Pool the estimates for confidence intervals
pool(mira_lm_unadj) %>%
  summary(conf.int = TRUE) %>%
  kbl(booktabs = T, digits = 3)
```

```r
r2_unadj <- sapply(mira_lm_unadj$analyses, function(mod) summary(mod)$r.squared)
r2_unadj

##  [1] 0.001846814 0.001845765 0.001845765 0.001845765 0.001846476 0.001845765 0.001845765 0.001845

# Pool the estimates for confidence intervals
pool(mira_lm_unadj) %>% summary(conf.int = TRUE)

##          term     estimate    std.error statistic       df      p.value        2.5 %       97.5 %
## 1 (Intercept) -8.048512221 0.5516774345 -14.58916 776123.4 0.000000e+00 -9.129781809 -6.967242632
## 2   crashYear  0.004168198 0.0002750544  15.15409 776123.4 0.000000e+00  0.003629101  0.004707296
## 3          p1 -0.007330558 0.0013354755  -5.48910 776123.4 4.041129e-08 -0.009948046 -0.004713070
## 4          p2 -0.035368167 0.0019161465 -18.45797 776123.4 0.000000e+00 -0.039123751 -0.031612583
## 5          p3  0.024428060 0.0008950239  27.29319 776123.4 0.000000e+00  0.022673842  0.026182277
```

### 8.7.2 Adjusted Model

Again, LASSO was used to identify the most important variables.

**Model Selection**

```r
# Piecewise Linear Terms
p1 <- pmin(pmax(data$crashYear - 2011, 0), 3)
p2 <- pmin(pmax(data$crashYear - 2014, 0), 2)
p3 <- pmax(data$crashYear - 2016, 0)

# Key predictors
var_pred_lm <- setdiff(c(var_imp, var_complete),
```

```
                    c(var_outcome, "baseline", "firstChange", "secondChange",
                      "Change"))

comp_sub <- comp_long %>% select(`.imp`, all_of(var_pred_lm), minorInjuryCount)

# Function to be passed into parallel computation
# Model selection using LASSO
select_model_lm <- function(...){
  library(glmnet)
  library(Matrix)
  library(tidyverse)
  i <- (...)
  set.seed(i)
  data_model <- comp_sub %>%
    filter(`.imp` == i)

  X <- data_model %>%
    select(-minorInjuryCount, -`.imp`) %>%
    sapply(as.numeric) %>%
    Matrix() %>%
  # Add the parallel terms
    cbind(p1, p2, p3)
  y <- data_model$minorInjuryCount
  # Set penalty factors to 0 to keep the 3 piece-wise linear terms
  fit <- glmnet(X, y, penalty.factor=c(rep(1, ncol(X)-3), 0, 0, 0))
  xval <- cv.glmnet(X, y,
                    penalty.factor=c(rep(1, ncol(X)-3), 0, 0, 0))
  return(coef(fit, s = xval$lambda.1se))
}

# Use Parallel Computation for model selection using LASSO

# Make new cluster
## cl <- makeCluster(5)

# Export the objects to each cluster
## clusterExport(cl, c('select_model_lm', 'comp_sub', "var_pred_lm", "p1", "p2", "p3"))

# Do the computation
# par_output_lm <- parLapply(cl, 1:m, fun = select_model_lm)

# Stop the cluster
## stopCluster(cl)

## save(par_output_lm, file = "model_selection_lm_fit.Rdata")
```

```
# The outcome of LASSO was returned as a list
load("model_selection_lm_fit.Rdata")
```

Similarly, the variables included in the adjusted model were the ones with major votes in the 10 models.

```
# Mira object for pooling
fit_lm_all <- par_output_lm %>% as.mira

# Names of variables where the coefficients were not 0
terms_lm <- lapply(fit_lm_all$analyses, function(x) row.names(x)[which(x != 0)])
# Use majority rule
votes_lm <- unlist(terms_lm)
table(votes_lm)

## votes_lm
##    (Intercept)        bicycle      cliffBank     motorcycle             p1             p2
##             10             10              9             10             10             10
```

```
##          truck       weatherA
##              3              3
```

The variables `bicycle`, `cliffBank`, `motorcycle`, `roadLane`, `speedLimit`, `streetLight`, `truck` and `weatherA` would be included in the final model, together with the piecewise linear terms.

### Final Adjusted Simple Linear Regression Model

```
fit_lm_adj <- list(10)
for(i in 1:10){
  comp <- comp_long %>%
    filter(`.imp` == i) %>%
    select(all_of(var_pred_lm), minorInjuryCount)
  fit_lm_adj[[i]] <- lm(minorInjuryCount ~ crashYear
                        + p1 + p2 + p3
                        + bicycle + cliffBank + motorcycle + roadLane
                        + speedLimit + streetLight + truck + weatherA,
                          data = comp)
}

mira_lm_adj <- fit_lm_adj %>% as.mira
```

```
# Compute r-square
r2_adj <- sapply(mira_lm_adj$analyses, function(mod) summary(mod)$r.squared)
r2_adj

##  [1] 0.03169330 0.03172437 0.03172114 0.03167892 0.03166423 0.03193123 0.03173510 0.03169875 0.03

pool(mira_lm_adj) %>% summary(conf.int = TRUE)

##                    term      estimate    std.error   statistic           df      p.value        
## 1           (Intercept) -5.8981786860 5.479508e-01 -10.7640654 268114.20212 0.000000e+00 -6.97214
## 2             crashYear  0.0029412589 2.732131e-04  10.7654394 260433.78940 0.000000e+00  0.00240
## 3                    p1 -0.0072341427 1.318824e-03  -5.4852981 378106.29746 4.130360e-08 -0.00981
## 4                    p2 -0.0321318292 1.905904e-03 -16.8590975 442440.01149 0.000000e+00 -0.03580
## 5                    p3  0.0298594317 9.258440e-04  32.2510407 279160.73915 0.000000e+00  0.02804
## 6               bicycle  0.3768776338 4.224353e-03  89.2154704 503979.57990 0.000000e+00  0.36859
## 7             cliffBank  0.0659208395 5.357304e-03  12.3048523     16.07247 1.353855e-09  0.05450
## 8            motorcycle  0.1933883547 3.757958e-03  51.4610224 529468.78504 0.000000e+00  0.18601
## 9         roadLane2-way  0.0931333368 2.702589e-03  34.4607874   1992.33343 0.000000e+00  0.08783
## 10         roadLaneNull  0.0086075051 2.856169e-02   0.3013654 630450.65311 7.631360e-01 -0.04737
## 11     roadLaneOff road  0.1292203682 6.751271e-03  19.1401555  18352.03902 0.000000e+00  0.11598
## 12          speedLimit  0.0031520127 4.280065e-05  73.6440421    294.08614 0.000000e+00  0.00306
## 13      streetLightNull  0.0197536523 2.498911e-03   7.9049040  11774.30976 2.886580e-15  0.01485
## 14       streetLightOff -0.0132137252 2.797120e-03  -4.7240460   1637.66656 2.509552e-06 -0.01870
## 15        streetLightOn  0.0003613386 2.823690e-03   0.1279668  11934.69921 8.981774e-01 -0.00517
## 16                truck -0.0520820610 2.548427e-03 -20.4369460 598064.62812 0.000000e+00 -0.05707
## 17 weatherAHail or Sleet -0.1249469976 6.661854e-02  -1.8755590 757904.44724 6.071625e-02 -0.25551
## 18    weatherAHeavy rain -0.0048103477 3.676822e-03  -1.3082896 704714.05517 1.907755e-01 -0.01201
## 19    weatherALight rain -0.0062267643 2.036489e-03  -3.0575980 511269.10182 2.231302e-03 -0.01021
## 20   weatherAMist or Fog -0.0084124224 6.225934e-03  -1.3511904 765050.78099 1.766348e-01 -0.02061
## 21          weatherANull -0.2125286127 5.611472e-03 -37.8739497 716221.60152 0.000000e+00 -0.22351
## 22          weatherASnow -0.0218579865 1.631374e-02  -1.3398510 762023.36566 1.802942e-01 -0.05383
```

# References


[1] Stef van Buuren. *Flexivle inputation of missing data*. Chapman & Hall/CRC, 2021.

[2] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.

[3] Thomas Lumley. *biglm: Bounded Memory Linear and Generalized Linear Models*, 2020.

[4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022.

[5] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011.