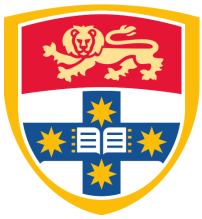


THE UNIVERSITY OF SYDNEY

STAT3022 Applied Linear Models

Assignment 2

Lecturer: Linh Nghiem

Authors: Matt Shu, Mason Wong, Zhuolin Jiang and Eva Yin

School of Mathematics and Statistics

The University of Sydney

Semester 1, 2022

Contents

```
library(GGally)
library(dplyr)
library(visdat)
library(summarytools)
library(tidyr)
library(ggplot2)
library(naniar)
library(DT)
library(usmap)
library(caret)
library(glmnet)
library(kableExtra)
library(patchwork)
library(MASS)
library(latex2exp)
library(formula.tools)
library(corrplot)
library(RColorBrewer)
```

Data Description

```
text_tbl <- data.frame(
  Variables = c("state", "year", "miles", "fatalities", "seatbelt", "speed65", "speed70", "drinkage", "alcoho
  "Description of Variables" = c(
    "factor indicating US state (abbreviation)",
    "factor indicating year",
    "millions of traffic miles per year",
    "number of fatalities per million of traffic miles (absolute frequencies of fatalities = fatalities times
    "seat belt usage rate, as self-reported by state population surveyed",
    "factor. Is there a 65 mile per hour speed limit?",
    "factor. Is there a 70 (or higher) mile per hour speed limit?",
    "factor. Is there a minimum drinking age of 21 years?",
    "factor. Is there a maximum of 0.08 blood alcohol content?",
    "median per capita income (in current US dollar)",
    "mean age",
    'factor indicating seat belt law enforcement ("no", "primary", "secondary")'
  ),
  Characteristics = c("", "", "", "outcome variable", "missing data", "", "", "", "", "", "", "")
)

kbl(text_tbl) %>%
  kable_paper(full_width = F) %>%
  column_spec(1, bold = T, border_right = T, width = "5em") %>%
  column_spec(2, width = "40em") %>%
  column_spec(3, border_left = T, width = "8em") %>%
  row_spec(4, background = "yellow")
```

Background (Literature)

The `Seat-Belt-Laws` dataset analysed in this report was sourced from a study done by Cohen & Einav (2003) which assessed the influence seat belt laws had on usage rates and by extension, fatality. The original data is an amalgamation of multiple sources from between 1983 to 1997 on 50 US States and the District of Columbia - our version appears to be limited to 35.

```
df = read.csv('https://raw.githubusercontent.com/mattshu0410/STAT3022-MLR-Seat-Belt-Laws/master/data/seatbelt

df %>% dplyr::select(state) %>% unique() %>% count() %>% pull()
```

Variables	Description.of.Variables	Characteristics
state	factor indicating US state (abbreviation)	
year	factor indicating year	
miles	millions of traffic miles per year	
fatalities	number of fatalities per million of traffic miles (absolute frequencies of fatalities = fatalities times miles)	outcome variable
seatbelt	seat belt usage rate, as self-reported by state population surveyed	missing data
speed65	factor. Is there a 65 mile per hour speed limit?	
speed70	factor. Is there a 70 (or higher) mile per hour speed limit?	
drinkage	factor. Is there a minimum drinking age of 21 years?	
alcohol	factor. Is there a maximum of 0.08 blood alcohol content?	
income	median per capita income (in current US dollar)	
age	mean age	
enforce	factor indicating seat belt law enforcement ("no", "primary", "secondary")	

```
## [1] 35
```

```
data_nona = df %>% drop_na()
```

Before proceeding we have encoded year as a factor rather than a numeric as macro-effects such as horrible weather or ramped-up traffic safety campaigning differ year-to-year are unlikely to be a linear trend between years. We considered grouping states together by geo-spatial proximity however we decided against it as it would require us to make unreasonable assumptions about the inherent similarity between adjacent states. For instance, two nearby states may have very different road laws.

```
df = df %>%
  mutate(year = as.factor(year))
```

In context of the study by Cohen & Einav (2003) which focused primarily on the impact of seat belt usage on fatalities, `year` and `state` were particularly important blocking variables. In our case, this is an indication we should likely include these factors as fixed effects in our model. This is further supported by contextual information which tells us that each state phased in seat belt laws at different years.

Summary Statistics & Visualisations

Missing Values Based on the visualization below, most of the missing values appear to arise from the seatbelt usage rate variable.

```
vis_dat(df)
```

Based on the table below, it also appears there are two entries missing for the state of New York for the years of 1996 and 1997.

```
df %>%
  dplyr::select(state, year) %>%
  filter(state == "NY") %>%
  arrange(year) %>%
  datatable()
```

Upon closer inspection it appears that different states have varying degrees of sparsity

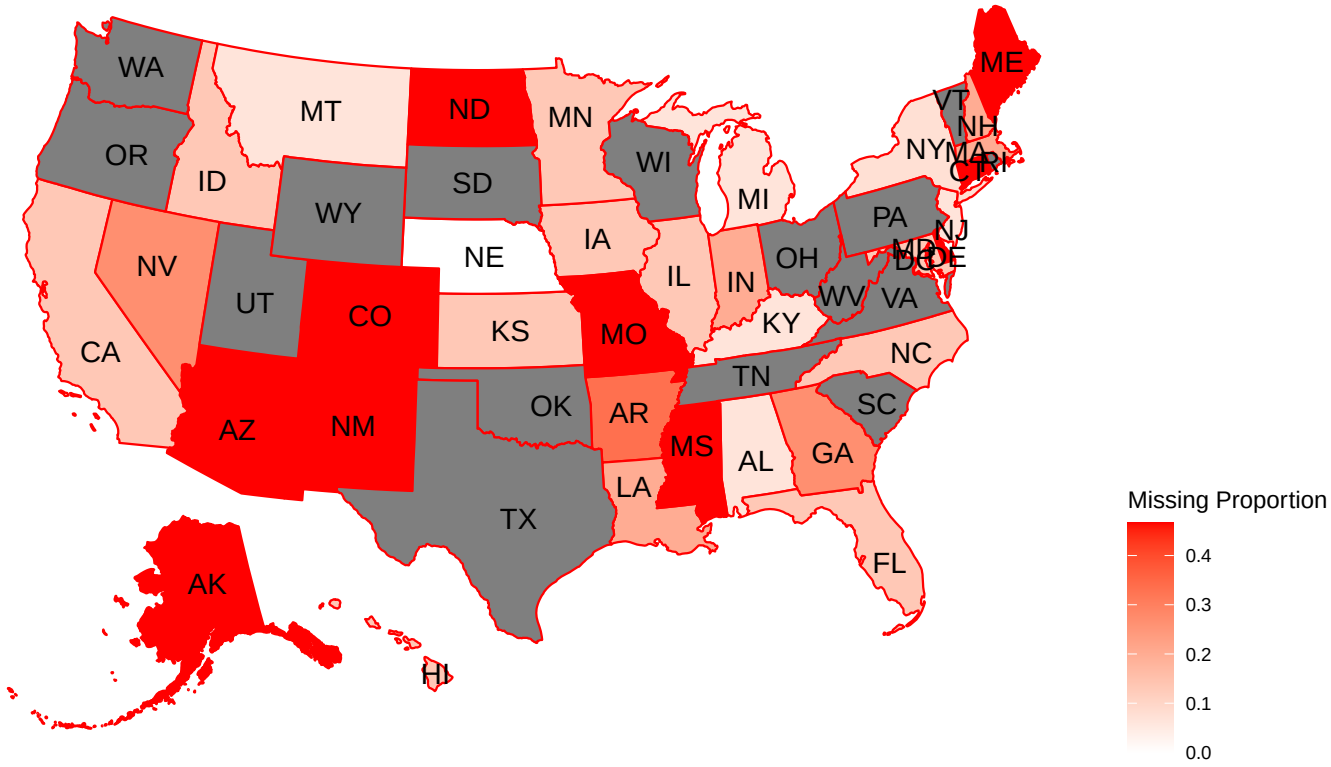
From the chloropleth plot we can see that there is a clump of southern states such as Arkansas, Arizona and New Mexico as well as states in the northeast such as Maine, Connecticut and Delaware which have a high number of missing values.

Furthermore it appears that the missing values follow a pattern whereby the earlier the year, the more missing values there are.

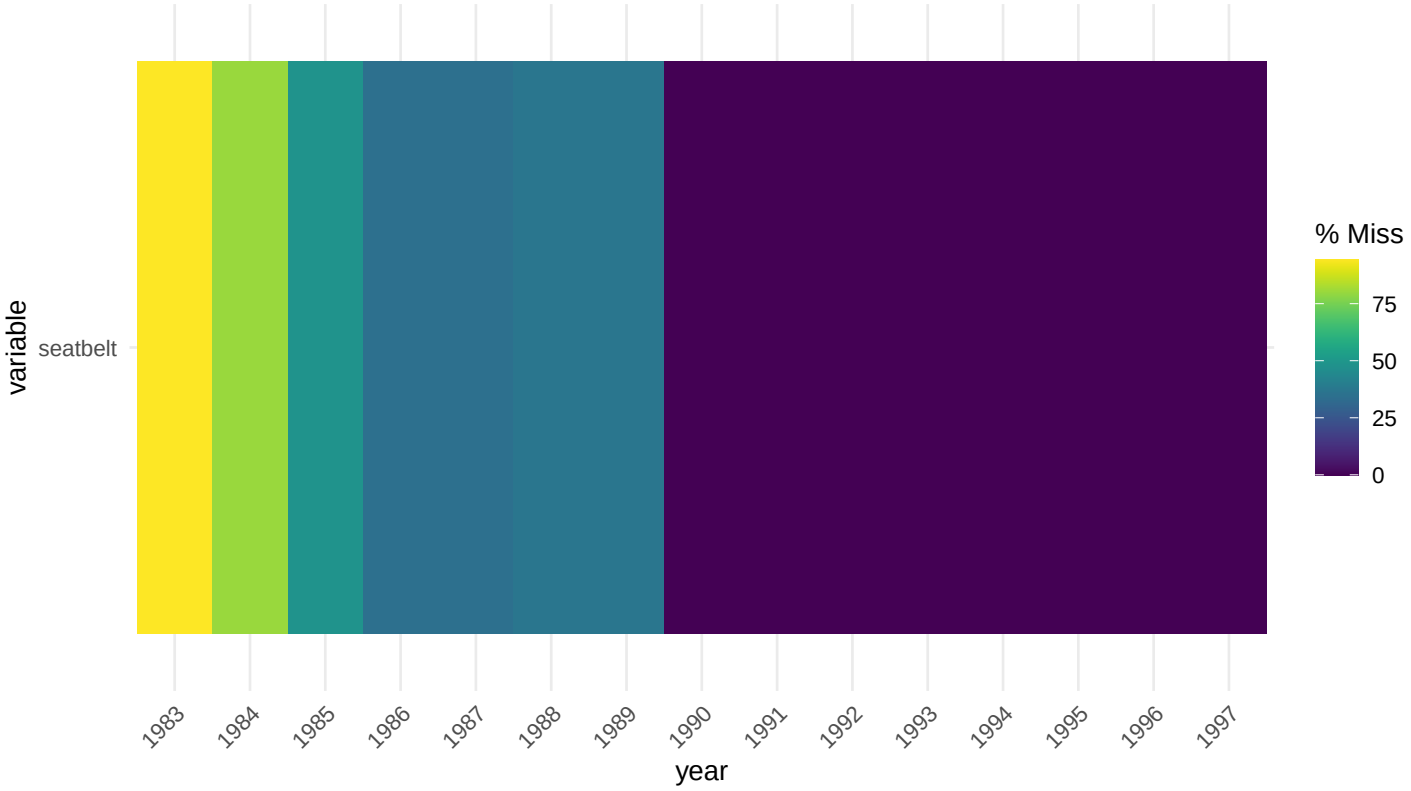
Both observations are consistent with study Cohen & Einav (2003) which has corroborated a consistent, complete National Highway Traffic Safety Administration (NHTSA source) between 1990-1999 with an incomplete source from the BRFSS between 1984-1997 that progressively added more states each year.

```
missing_df = df %>%
  dplyr::select(state, seatbelt) %>%
  group_by(state) %>%
  summarise(values = sum(is.na(seatbelt))/n()) %>%
  mutate(state = as.character(state))

# Missingness by State
plot_usmap(data = missing_df, color = "red", labels = TRUE) +
  scale_fill_continuous(
    low = "white",
    high = "red",
    name = "Missing Proportion"
  ) +
```

```
  theme(legend.position = "right") +
  labs(
    title = ""
  )
```




```
# Missingness by Year
df %>%
  dplyr::select(year, seatbelt) %>%
  gg_miss_fct(., fct = year)
```

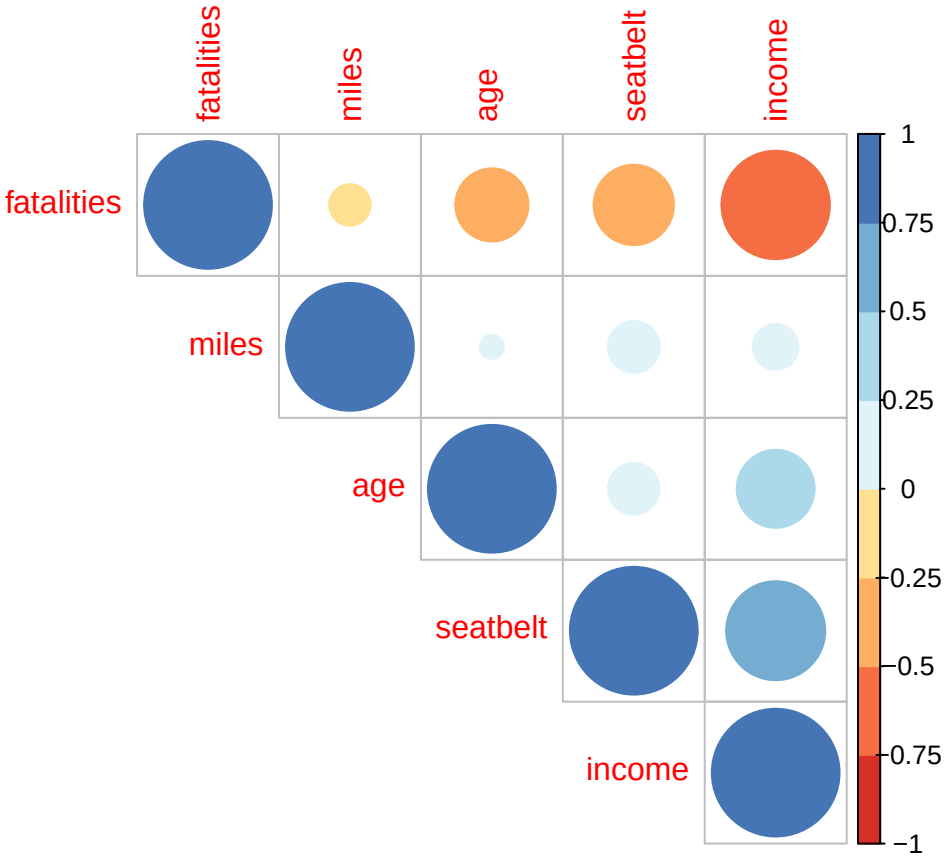


We chose to drop the rows with NA instead of deleting the whole `seatbelt` variable because of its significance in the original study (Cohen & Einav, 2003)

Collinearity From the correlation matrix, we can see that median per-capita income and seatbelt usage are highly correlated (r=0.60). The mean age and median per-capita income are also moderately correlated (r=0.40). There is strong limitation to this analysis is that this only captures the relationships between the quantitative variables.

- `income` and `seatbelt` (r=0.60)
- `age` and `income` (r=0.40)

```
#df %>%
#  dplyr::select(-seatbelt, -state) %>%
#  ggpairs()
#
#df


M = cor(df[,sapply(df, is.numeric)] %>% drop_na())
corrplot(M, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```

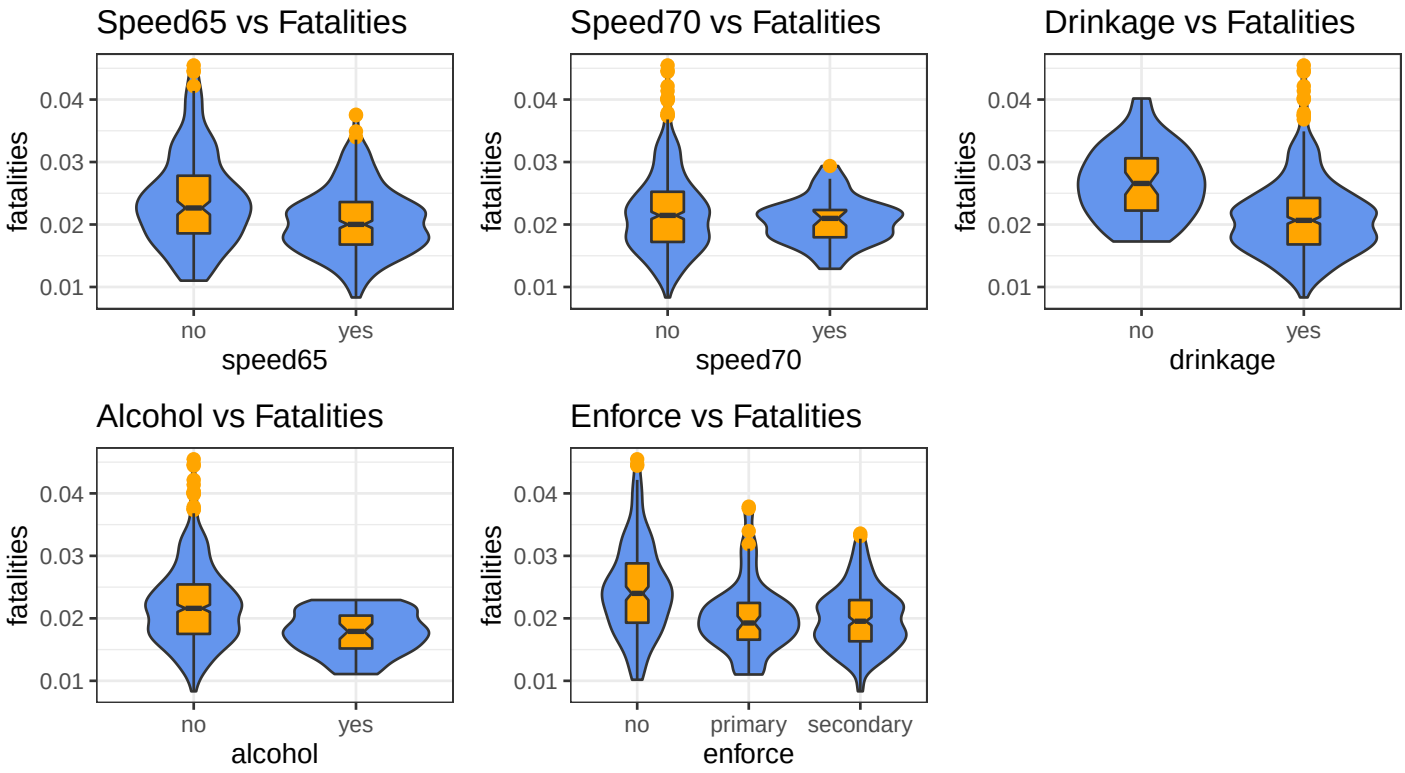



```
#df[,sapply(df, is.numeric)] %>%
#  qtlcharts::iplotCorr()
```

Qualitative Variables The violin box-plots show the distribution of fatalities per million miles of traffic `fatalities` for different levels of our categorical variables. We use a heuristic where any case of non-overlapping notches signify strong evidence at a 95% confidence level, that the medians of the compared levels differ. From the plots you can see that the implementation of a 65-mile speed limit `speed65`, a minimum drinking age `drinkage`, maximum BAC `alcohol` and any form of seat belt law enforcement `enforce` produced a strongly differentiated fatality rate.

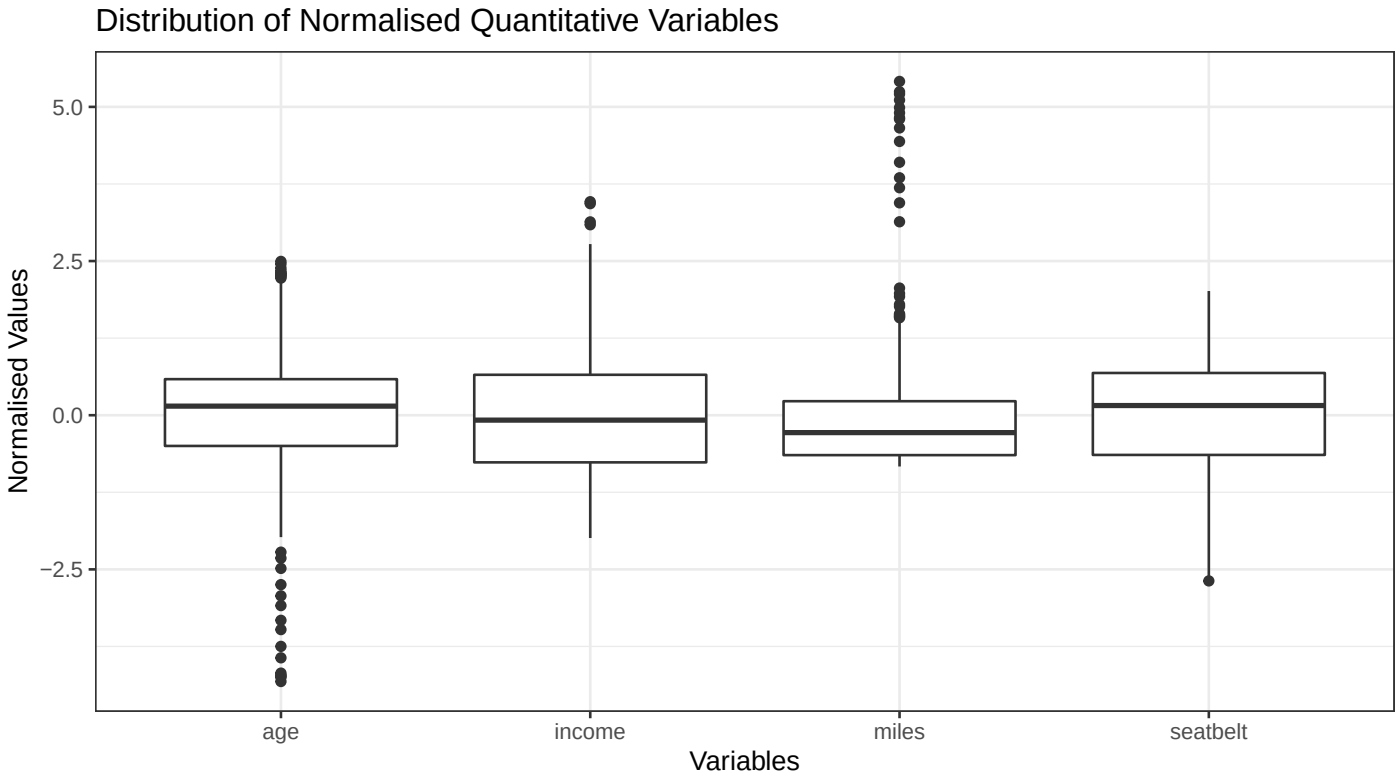
```
g6 <- ggplot(df,
       aes(x = speed65,
           y = fatalities)) +
  geom_violin(fill = "cornflowerblue") +
  geom_boxplot(notch = TRUE,
               width = .2,
               fill = "orange",
               outlier.color = "orange",
               outlier.size = 2) +
  labs(title = "Speed65 vs Fatalities") +
  theme_bw()

g7 <- ggplot(df,
       aes(x = speed70,
           y = fatalities)) +
  geom_violin(fill = "cornflowerblue") +
  geom_boxplot(notch = TRUE,
               width = .2,
               fill = "orange",
               outlier.color = "orange",
```

```
               outlier.size = 2) +
  labs(title = "Speed70 vs Fatalities") +
  theme_bw()

g8 <- ggplot(df,
       aes(x = drinkage,
           y = fatalities)) +
  geom_violin(fill = "cornflowerblue") +
  geom_boxplot(notch = TRUE,
               width = .2,
               fill = "orange",
               outlier.color = "orange",
               outlier.size = 2) +
  labs(title = "Drinkage vs Fatalities") +
  theme_bw()

g9 <- ggplot(df,
       aes(x = alcohol,
           y = fatalities)) +
  geom_violin(fill = "cornflowerblue") +
  geom_boxplot(notch = TRUE,
               width = .2,
               fill = "orange",
               outlier.color = "orange",
               outlier.size = 2) +
  labs(title = "Alcohol vs Fatalities") +
  theme_bw()

g10 <- ggplot(df,
       aes(x = enforce,
           y = fatalities)) +
  geom_violin(fill = "cornflowerblue") +
  geom_boxplot(notch = TRUE,
               width = .2,
               fill = "orange",
               outlier.color = "orange",
               outlier.size = 2) +
  labs(title = "Enforce vs Fatalities") +
  theme_bw()


g6+g7+g8+g9+g10+plot_layout(ncol = 3)
```



Quantitative Variables From the boxplot visualisation, we can see that age is heavily left skewed, miles is heavily right skewed. `income` & `seatbelt` appear to be fairly normally distributed.

```
# Boxplots for quantitative variables
df %>%
  dplyr::select(age, income, miles, seatbelt) %>%
  apply(., 2, scale) %>%
  data.frame() %>%
  pivot_longer(cols = 1:4,
               names_to = "variable",
               values_to = "value") %>%
  ggplot() +
  geom_boxplot() +
  aes(x = variable,
      y = value) +
  theme_bw() +
  labs(
    title = "Distribution of Normalised Quantitative Variables",
    x = "Variables",
    y = "Normalised Values"
  )
```



Model Building

Overview

This is a summary of our model building procedure.

Variable Selection

To build our models, we consider the data frame `data_nona` where we remove the rows with the `NA` observations. Doing this is an example of listwise deletion. We consider two sets of models:

- The first set of models use the AIC critera to step through the covariates and
- The second set of models uses the BIC criteria to step through the covariates

Because we want to produce a parsimonious model, where we balance the accuracy of our model with the minimal amount of covariates, this is why we employ both the AIC and BIC criterion. As the AIC criterion tends to produce models with more covariates and the BIC criterion tends to produce models with less covariates (the penalty of the cost function is larger for the BIC) we aim to balance accuracy and number of covariates in this way.

With these two sets of models, we aim to arrive at two models

- With the first set of models (using the AIC criteria to step through) we employ forward, backward and bidirectional approaches. We then pick the best out of the three based on adjusted R^2 value
- With the second set of models (using the BIC criteria to step through we employ forward, backward and bidirectional approaches. We then pick the best out of the three based on adjusted R^2 value

```
n = nrow(data_nona)
full_mod = lm(fatalities ~ ., data = data_nona)
null_mod = lm(fatalities ~ 1, data = data_nona)
# The AIC criterion models
forward_aic = stepAIC(null_mod,
                      scope = list(upper = formula(full_mod),
                                   lower = formula(null_mod)),
                      direction = 'forward',
                      k = 2)
backward_aic = stepAIC(full_mod,
```

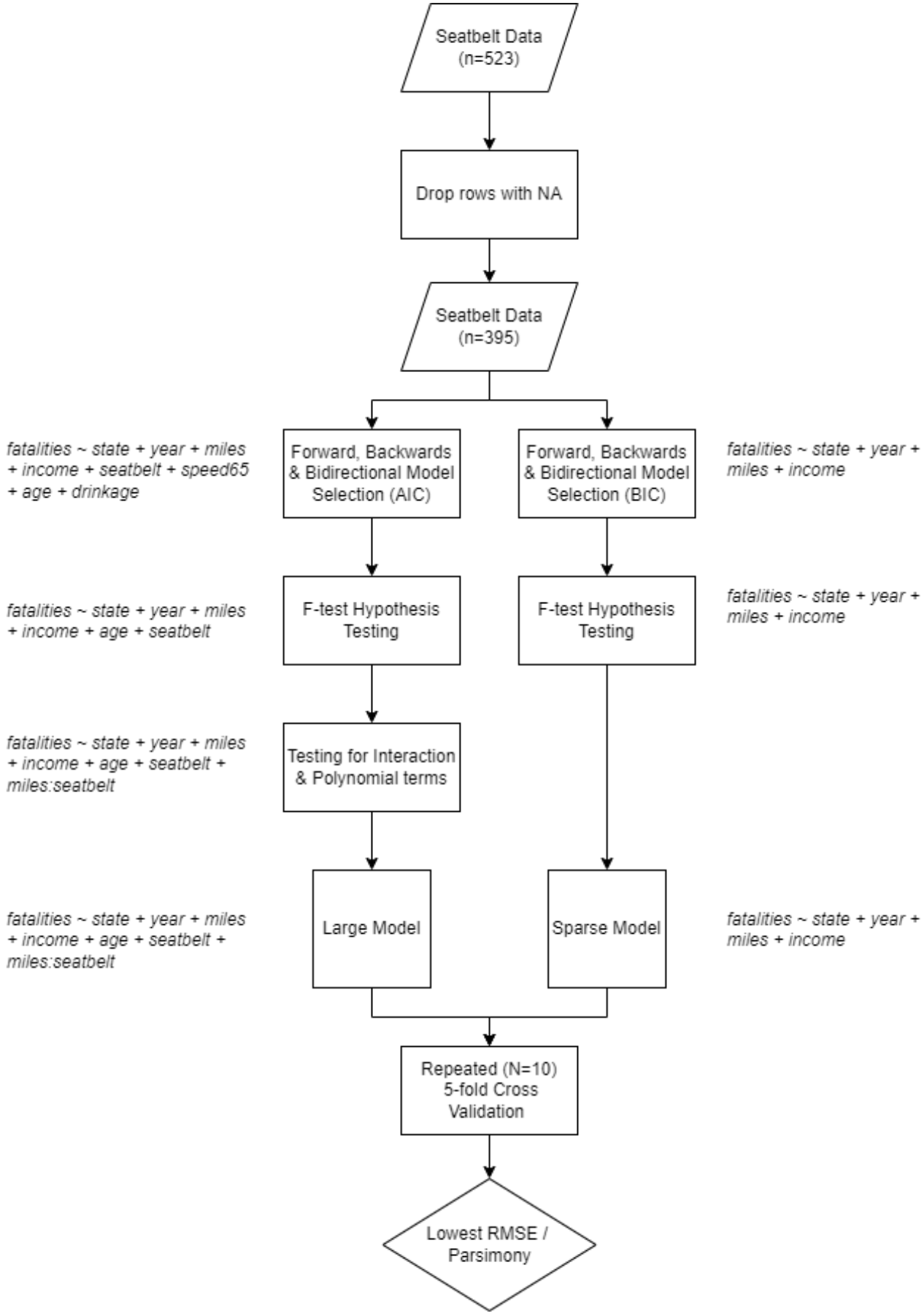


Figure 1: Model Selection Procedure

```
                        scope = list(upper = formula(full_mod),
                                     lower = formula(null_mod)),
                        direction = 'backward',
                        k = 2)
bidir_aic = stepAIC(full_mod,
                        scope = list(upper = formula(full_mod),
                                     lower = formula(null_mod)),
                        direction = 'both',
                        k = 2)
# The BIC criterion models
forward_bic = stepAIC(null_mod,
                        scope = list(upper = formula(full_mod),
                                     lower = formula(null_mod)),
                        direction = 'forward',
                        k = log(n))
backward_bic = stepAIC(full_mod,
                        scope = list(upper = formula(full_mod),
                                     lower = formula(null_mod)),
                        direction = 'backward',
                        k = log(n))
bidir_bic = stepAIC(full_mod,
                        scope = list(upper = formula(full_mod),
                                     lower = formula(null_mod)),
                        direction = 'both',
                        k = log(n))
```

For the AIC criterion we see the models:

- forward
  - adj R^2 : 0.8899443
  - formula: fatalities ~ state + year + miles + income + speed70 + age + seatbelt + enforce
- backward
  - adj R^2 : 0.8899443
  - formula: fatalities ~ state + year + miles + seatbelt + speed70 + income + age + enforce
- bidirectional
  - adj R^2 : 0.8899443
  - formula: fatalities ~ state + year + miles + seatbelt + speed70 + income + age + enforce

For the BIC criterion we see the models:

- forward
  - adj R^2 : 0.8894103
  - formula: fatalities ~ income + state + year + miles + speed70 + age + seatbelt
- backward
  - adj R^2 : 0.8894103
  - formula: fatalities ~ state + year + miles + seatbelt + speed70 + income + age
- bidirectional
  - adj R^2 : 0.8894103
  - formula: fatalities ~ state + year + miles + seatbelt + speed70 + income + age

Models we have so far

Thus we see that the models we have arrived at so far are:

- By the AIC criterion:

$$fatalities \sim state + year + age + income + miles + seatbelt + speed65 + drinkage$$

- By the BIC crtierion

$$fatalities \sim state + year + income + miles$$

Multicollinearity

We now check the multicollinearity present in our model.

A traditional VIF metric would not be appropriate as it is determined with respect to a single coefficient. We used a generalised collinearity diagnostic (GVIF) introduced by Fox & Monette (1992) where a fair comparison between variables is made by considering the following rule of thumb:

$$(GVIF^{\frac{1}{2*df}})^2 > 10$$

While we check for multicollinearity in this step we further note that because the goal of our model is prediction, the multicollinearity doesn't matter. This is because while multicollinearity makes the interpretation of the coefficients difficult (as it inflates the variances) the predicted values stay the same.

The first model is given by:

$$fatalities \sim state + year + age + income + miles + seatbelt + speed65 + drinkage$$

```
mod1 = lm(fatalities ~ state + year + age + income + miles + seatbelt + speed65 + drinkage, data = data_nona)
```

We note that the generalised variance inflation factors are given by:

```
library(car)
vif(mod1)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## state     1.271835e+06 34        1.229620
## year      5.451176e+01  1        7.383208
## age       6.105957e+01  1        7.814062
## income    6.792564e+01  1        8.241701
## miles     1.333916e+02  1       11.549527
## seatbelt  6.954472e+00  1        2.637133
## speed65   3.378593e+00  1        1.838095
## drinkage  1.639897e+00  1        1.280584
```

So we see here that by considering the squares of the $GVIF^{\frac{1}{2DF}}$ we see that `age`, `income` and `miles` are covariates which are highly correlated with one or more other covariates

The second model is given by:

$$fatalities \sim state + year + income + miles$$

```
mod2 = lm(fatalities ~ state + year + income + miles, data = data_nona)
```

We note that the generalized variance inflation factors are given by:

```
library(car)
vif(mod2)
```

```
##               GVIF Df GVIF^(1/(2*Df))
## state  3505.51903 34        1.127532
## year     32.50520  1        5.701333
## income   63.18947  1        7.949181
## miles    97.55114  1        9.876798
```

And so we see `income` and `miles` are highly correlated with one another.

Inferences

We now check the significance of our variables with by using a combination of t-tests (with the `summary()` command) and f-test (with the `anova()` command). We do this from the model evaluation perspective, with the goal of seeing whether we can discard predictors (so that both our models are less complex).

Looking at the first model obtained by the AIC criterion

Consider first the summary table for our first model obtained with the AIC criterion:

```
summary(mod1)
```

```
##
## Call:
## lm(formula = fatalities ~ state + year + age + income + miles +
##     seatbelt + speed65 + drinkage, data = data_nona)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0045416 -0.0010340 -0.0000825  0.0010196  0.0069654
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.271e+00  3.443e-01   6.597 1.53e-10 ***
## stateAL     -1.801e-03  3.307e-03  -0.545 0.586252
## stateAR     -7.293e-04  3.395e-03  -0.215 0.830032
## stateAZ     -1.221e-03  2.909e-03  -0.420 0.675031
## stateCA      4.943e-03  6.498e-03   0.761 0.447337
## stateCO     -8.663e-03  2.553e-03  -3.393 0.000770 ***
## stateCT     -2.206e-02  3.996e-03  -5.520 6.59e-08 ***
## stateDC     -1.887e-02  3.904e-03  -4.835 2.00e-06 ***
## stateDE     -1.327e-02  2.914e-03  -4.554 7.26e-06 ***
## stateFL     -2.594e-03  5.857e-03  -0.443 0.658089
## stateGA     -3.146e-03  3.045e-03  -1.033 0.302191
## stateHI     -1.201e-02  2.552e-03  -4.706 3.63e-06 ***
## stateIA     -8.732e-03  3.561e-03  -2.452 0.014691 *
## stateID     -6.472e-04  2.081e-03  -0.311 0.755962
## stateIL     -7.812e-03  3.828e-03  -2.041 0.042037 *
## stateIN     -8.088e-03  3.384e-03  -2.390 0.017373 *
## stateKS     -9.049e-03  3.039e-03  -2.978 0.003104 **
## stateKY     -4.102e-03  3.165e-03  -1.296 0.195765
## stateLA      9.113e-04  2.404e-03   0.379 0.704844
## stateMA     -1.982e-02  3.952e-03  -5.015 8.41e-07 ***
## stateMD     -1.183e-02  3.120e-03  -3.793 0.000175 ***
## stateME     -1.302e-02  3.307e-03  -3.937 9.93e-05 ***
## stateMI     -6.427e-03  3.612e-03  -1.779 0.076030 .
## stateMN     -1.356e-02  3.023e-03  -4.486 9.81e-06 ***
## stateMO     -5.852e-03  3.625e-03  -1.614 0.107356
## stateMS      5.262e-03  2.665e-03   1.974 0.049118 *
## stateMT     -2.850e-03  2.815e-03  -1.012 0.312046
## stateNC     -2.152e-03  3.564e-03  -0.604 0.546330
## stateND     -1.325e-02  2.982e-03  -4.444 1.18e-05 ***
## stateNE     -1.077e-02  3.007e-03  -3.581 0.000391 ***
## stateNH     -1.615e-02  2.725e-03  -5.926 7.36e-09 ***
## stateNJ     -1.741e-02  4.211e-03  -4.135 4.45e-05 ***
## stateNM      2.089e-03  2.136e-03   0.978 0.328694
## stateNV     -3.318e-03  2.655e-03  -1.250 0.212099
## stateNY     -8.517e-03  4.708e-03  -1.809 0.071293 .
```

```
## year        -1.151e-03  1.784e-04  -6.451 3.66e-10 ***
## age          1.191e-03  5.070e-04   2.349 0.019349 *
## income       5.619e-07  1.546e-07   3.633 0.000321 ***
## miles       -6.182e-08  2.128e-08  -2.906 0.003896 **
## seatbelt    -3.875e-03  1.370e-03  -2.828 0.004947 **
## speed65yes  -4.455e-04  3.873e-04  -1.150 0.250830
## drinkageyes  2.025e-04  6.192e-04   0.327 0.743883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001777 on 353 degrees of freedom
## Multiple R-squared:  0.8978, Adjusted R-squared:  0.8859
## F-statistic: 75.61 on 41 and 353 DF,  p-value: < 2.2e-16
```

It seems to suggest that `seatbelt`, `speed65` and `drinkage` are not significant. We use the F-test to verify this:

```
anova(lm(fatalities ~ state + year + age + miles + income + seatbelt + speed65 + drinkage, data = data_nona))
```

```
## Analysis of Variance Table
##
## Response: fatalities
##            Df    Sum Sq    Mean Sq  F value    Pr(>F)
## state      34 0.0070114 0.00020622  65.3159 < 2.2e-16 ***
## year        1 0.0025831 0.00258312 818.1563 < 2.2e-16 ***
## age         1 0.0000405 0.00004055  12.8419 0.0003866 ***
## miles       1 0.0000560 0.00005595  17.7222 3.247e-05 ***
## income      1 0.0000565 0.00005649  17.8914 2.984e-05 ***
## seatbelt    1 0.0000359 0.00003591  11.3740 0.0008274 ***
## speed65     1 0.0000039 0.00000386   1.2215 0.2698254
## drinkage    1 0.0000003 0.00000034   0.1069 0.7438828
## Residuals 353 0.0011145 0.00000316
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Thus we see:

- At the 5% level, given the covariates `speed65`, `seatbelt`, `income`, `miles`, `age`, `year` and `state` are in the model, the model containing `drinkage` is not statistically significant. So we can drop it.
- Then at the 5% level, given the covariates `seatbelt`, `income`, `miles`, `age`, `year` and `state` are in the model, the model containing `speed65` is not statistically significant. So we can drop it.
- Hence we see that the AIC model becomes (by the F-test)

$$fatalities \sim state + year + age + income + miles + seatbelt$$

```
# update mod1 to the most updated version
mod1 = lm(fatalities ~ state + year + age + income + miles + seatbelt, data = data_nona)
```

Looking at the second model obtained by the BIC criterion

Again we consider the summary and anova tables:

```
summary(mod2)
```

```
##
## Call:
## lm(formula = fatalities ~ state + year + income + miles, data = data_nona)
##
## Residuals:
```

```
##        Min         1Q     Median         3Q        Max
## -0.0044278 -0.0011354 -0.0000566  0.0011045  0.0076860
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.237e+00  2.761e-01   8.103 8.64e-15 ***
## stateAL      6.616e-03  1.345e-03   4.918 1.34e-06 ***
## stateAR      7.862e-03  1.305e-03   6.024 4.24e-09 ***
## stateAZ      5.638e-03  1.258e-03   4.481 1.00e-05 ***
## stateCA      1.774e-02  4.761e-03   3.726 0.000226 ***
## stateCO     -2.525e-03  1.045e-03  -2.416 0.016195 *
## stateCT     -1.346e-02  1.508e-03  -8.927  < 2e-16 ***
## stateDC     -1.082e-02  1.464e-03  -7.394 1.03e-12 ***
## stateDE     -6.741e-03  9.344e-04  -7.214 3.28e-12 ***
## stateFL      1.159e-02  2.221e-03   5.220 3.05e-07 ***
## stateGA      4.273e-03  1.669e-03   2.560 0.010889 *
## stateHI     -6.773e-03  8.264e-04  -8.195 4.53e-15 ***
## stateIA     -4.853e-04  1.035e-03  -0.469 0.639358
## stateID      4.364e-03  1.124e-03   3.882 0.000123 ***
## stateIL      1.227e-03  1.770e-03   0.693 0.488595
## stateIN      2.342e-04  1.379e-03   0.170 0.865260
## stateKS     -1.722e-03  9.698e-04  -1.775 0.076696 .
## stateKY      4.210e-03  1.272e-03   3.309 0.001032 **
## stateLA      6.786e-03  1.309e-03   5.185 3.63e-07 ***
## stateMA     -1.000e-02  1.271e-03  -7.868 4.33e-14 ***
## stateMD     -4.926e-03  1.145e-03  -4.303 2.17e-05 ***
## stateME     -4.564e-03  1.088e-03  -4.197 3.42e-05 ***
## stateMI      2.117e-03  1.696e-03   1.248 0.212819
## stateMN     -6.239e-03  1.094e-03  -5.705 2.45e-08 ***
## stateMO      2.786e-03  1.360e-03   2.048 0.041263 *
## stateMS      1.214e-02  1.524e-03   7.964 2.26e-14 ***
## stateMT      3.564e-03  1.120e-03   3.183 0.001585 **
## stateNC      6.043e-03  1.527e-03   3.958 9.13e-05 ***
## stateND     -5.464e-03  1.215e-03  -4.498 9.29e-06 ***
## stateNE     -3.285e-03  9.248e-04  -3.552 0.000434 ***
## stateNH     -9.714e-03  8.557e-04 -11.351  < 2e-16 ***
## stateNJ     -7.404e-03  1.515e-03  -4.887 1.55e-06 ***
## stateNM      6.218e-03  1.323e-03   4.699 3.74e-06 ***
## stateNV      2.791e-03  8.608e-04   3.243 0.001296 **
## stateNY      2.200e-03  2.156e-03   1.021 0.308100
## year        -1.117e-03  1.402e-04  -7.968 2.19e-14 ***
## income       5.867e-07  1.518e-07   3.866 0.000131 ***
## miles       -9.652e-08  1.851e-08  -5.214 3.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001808 on 357 degrees of freedom
## Multiple R-squared:  0.893,  Adjusted R-squared:  0.8819
## F-statistic: 80.51 on 37 and 357 DF,  p-value: < 2.2e-16
```

```
anova(mod2)
```

```
## Analysis of Variance Table
##
## Response: fatalities
##            Df    Sum Sq    Mean Sq F value    Pr(>F)
## state      34 0.0070114 0.00020622  63.098 < 2.2e-16 ***
```

```
## year        1 0.0025831 0.00258312 790.369 < 2.2e-16 ***
## income      1 0.0000520 0.00005198  15.904 8.084e-05 ***
## miles       1 0.0000889 0.00008885  27.187 3.134e-07 ***
## Residuals 357 0.0011668 0.00000327
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that all covariates are significant and nothing can be dropped. So we don't change our model obtained by the BIC criteria

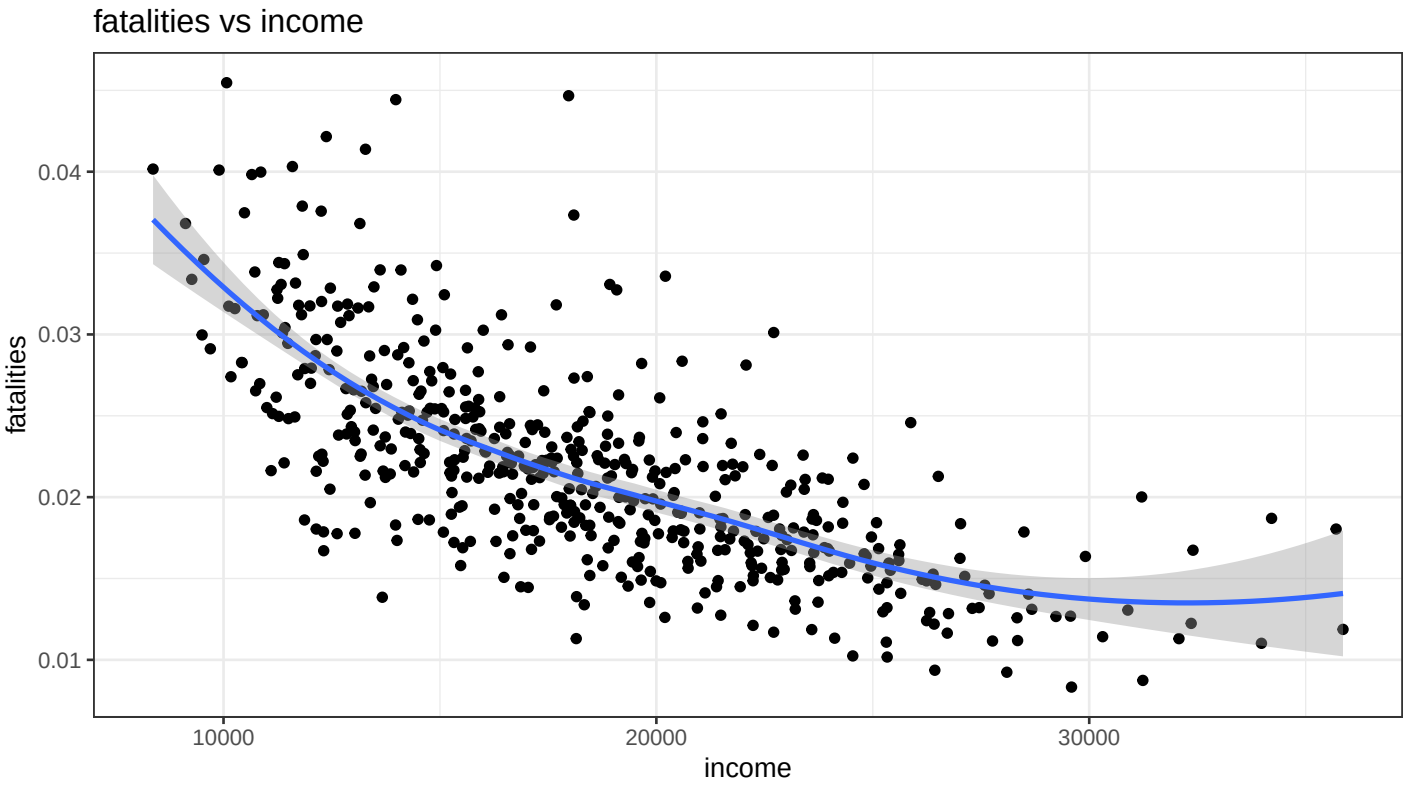
Note: we henceforth will refer to the model obtained by the AIC criteria as the larger model and the model obtained by the BIC criteria as the smaller model

Polynomial and interaction terms

Due to our goal of trying to find a balance between a model which fits well versus a model with a smaller number of covariates (that is to be parsimonious) we consider the polynomial and interaction term only for the larger model we have presented

We now consider the polynomial a polynomial term for the `income` variable because of the relationship we have between `income` and `fatalities`. That is, it looks quadratic!

```
ggplot(data = df, mapping = aes(x = income, y = fatalities)) +
  geom_point() +
  geom_smooth() +
  labs(title = 'fatalities vs income') +
  theme_bw()
```



We now consider the polynomial a polynomial term for the `income` variable because of the relationship we have between `income` and `fatalities`. That is, it looks quadratic!

Observing our larger model, to see if polynomial terms would increase the fit, we fit an arbitrarily high degree for `income` (degree 3) and read the resulting anova table:

```
anova(lm(fatalities ~ state + year + age + miles + seatbelt +
           income +
           I(income^2) +
           I(income^3), data = data_nona))
```

```
## Analysis of Variance Table
##
## Response: fatalities
##              Df    Sum Sq    Mean Sq  F value    Pr(>F)
## state        34 0.0070114 0.00020622  65.4080 < 2.2e-16 ***
## year          1 0.0025831 0.00258312 819.3103 < 2.2e-16 ***
## age           1 0.0000405 0.00004055  12.8600 0.0003830 ***
## miles         1 0.0000560 0.00005595  17.7471 3.207e-05 ***
## seatbelt      1 0.0000458 0.00004583  14.5378 0.0001621 ***
## income        1 0.0000466 0.00004656  14.7688 0.0001441 ***
## I(income^2)   1 0.0000057 0.00000568   1.8006 0.1805041
## I(income^3)   1 0.0000001 0.00000009   0.0276 0.8681934
## Residuals   353 0.0011129 0.00000315
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is no significant change so we don't include the quadratic term.

Next for our larger model, we consider an interaction term. By considering relevant contextual information our intuitive argument is that an increase or decrease in seat belt usage rate, should change the impact of total miles on the fatality rate.

we consider the interaction term between `miles` and `seatbelt`

```
anova(lm(fatalities ~ state + year + age + income + miles*seatbelt, data = data_nona))
```

```
## Analysis of Variance Table
##
## Response: fatalities
##                 Df    Sum Sq    Mean Sq F value    Pr(>F)
## state           34 0.0070114 0.00020622  67.118 < 2.2e-16 ***
## year             1 0.0025831 0.00258312 840.730 < 2.2e-16 ***
## age              1 0.0000405 0.00004055  13.196 0.0003220 ***
## income           1 0.0000686 0.00006863  22.339 3.303e-06 ***
## miles            1 0.0000438 0.00004381  14.258 0.0001869 ***
## seatbelt         1 0.0000359 0.00003591  11.688 0.0007024 ***
## miles:seatbelt   1 0.0000310 0.00003105  10.104 0.0016095 **
## Residuals      354 0.0010877 0.00000307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Thus by the F-test, we see that given all the covariates in the larger model, the interaction term is still statistically significant. Hence we update our larger model to include this:

```
mod1 = lm(fatalities ~ state + year + age + income + miles + seatbelt + miles:seatbelt, data = data_nona)
```

The larger model is thus:

$$fatalities \sim state + year + age + income + miles + seatbelt + miles : seatbelt$$

The smaller model is thus:

$$fatalities \sim state + year + income + miles$$

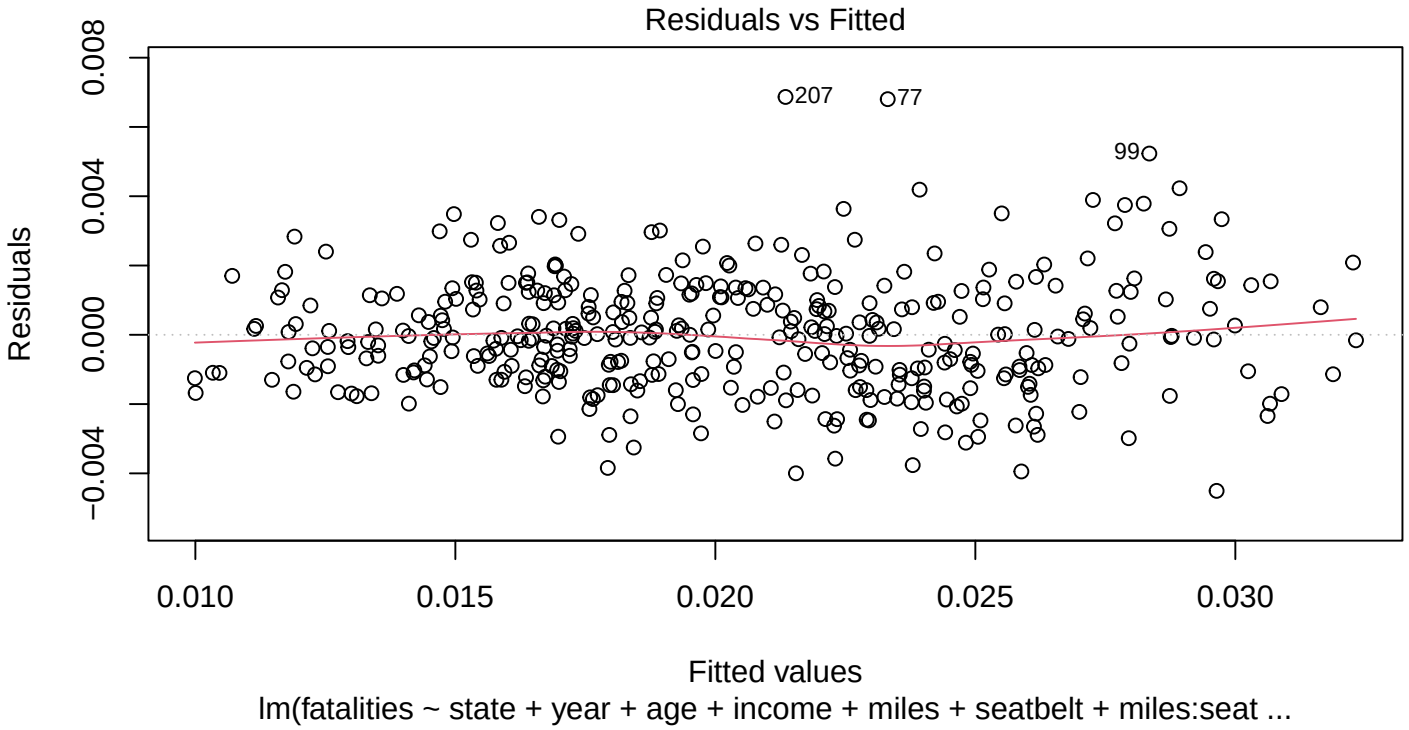
Checking Assumptions

For our larger model, which is now:

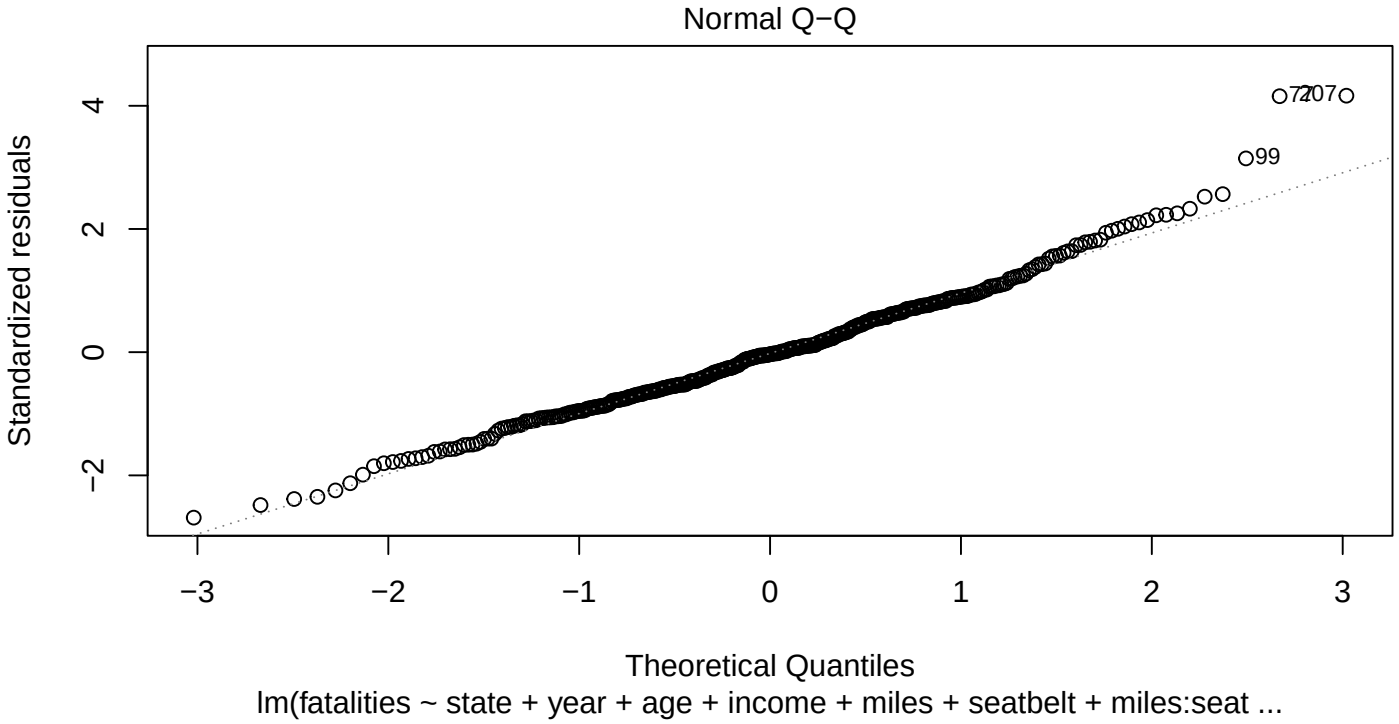
$$fatalities \sim state + year + age + income + miles + seatbelt + miles : seatbelt$$

We check for heteroscedasticity and for linearity

```
plot(mod1, which = 1)
```




```
plot(mod1, which = 2)
```



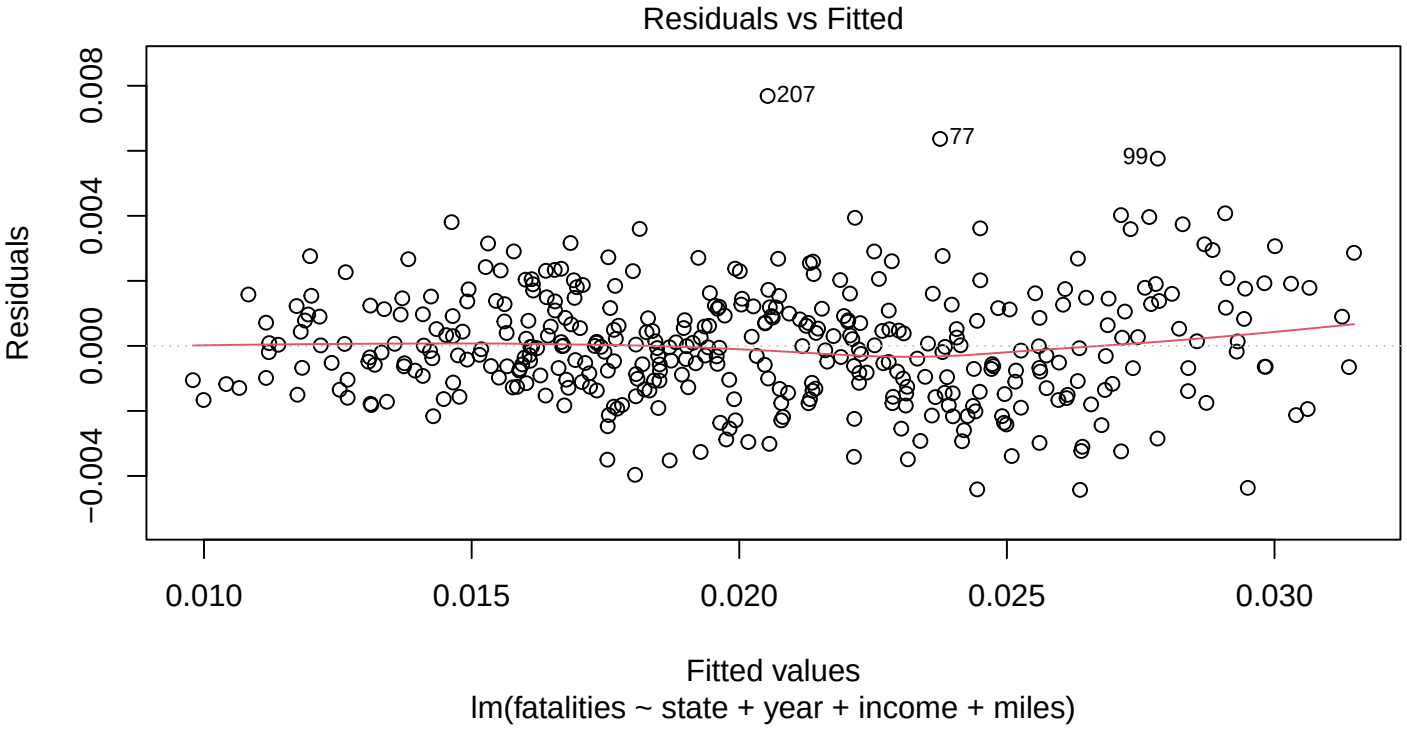
- As the spread of the residuals seems to be fairly constant and the red line is horizontal, the assumption of constant variance and linearity is satisfied
- As the QQ plot shows the stanardised resiudals lie along the line quite tightly, the assumption of normality is satisfied.

For our smaller model, which is now:

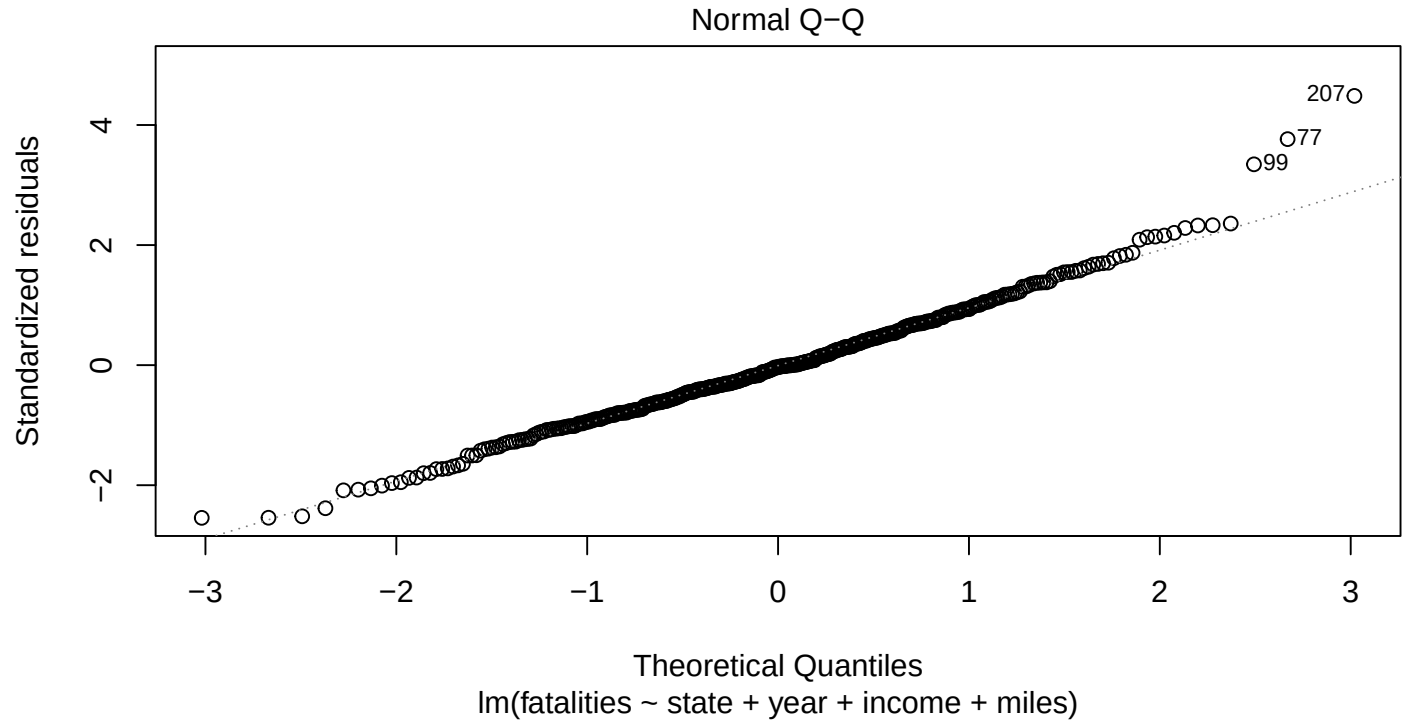
$$fatalities \sim state + year + income + miles$$

We check for heteroscedasticity and for linearity

```
plot(mod2, which = 1)
```




```
plot(mod2, which = 2)
```



Similar to the larger model, the smaller model has the linearity assumption, constant variance assumption and normality assumption satisfied.

Unusual Observations

For our larger model and smaller model we consider:

- Outliers
- High leverage observations

- Influential observations

Larger Model

Our larger model is:

$$fatalities \sim state + year + age + income + miles + seatbelt + miles : seatbelt$$

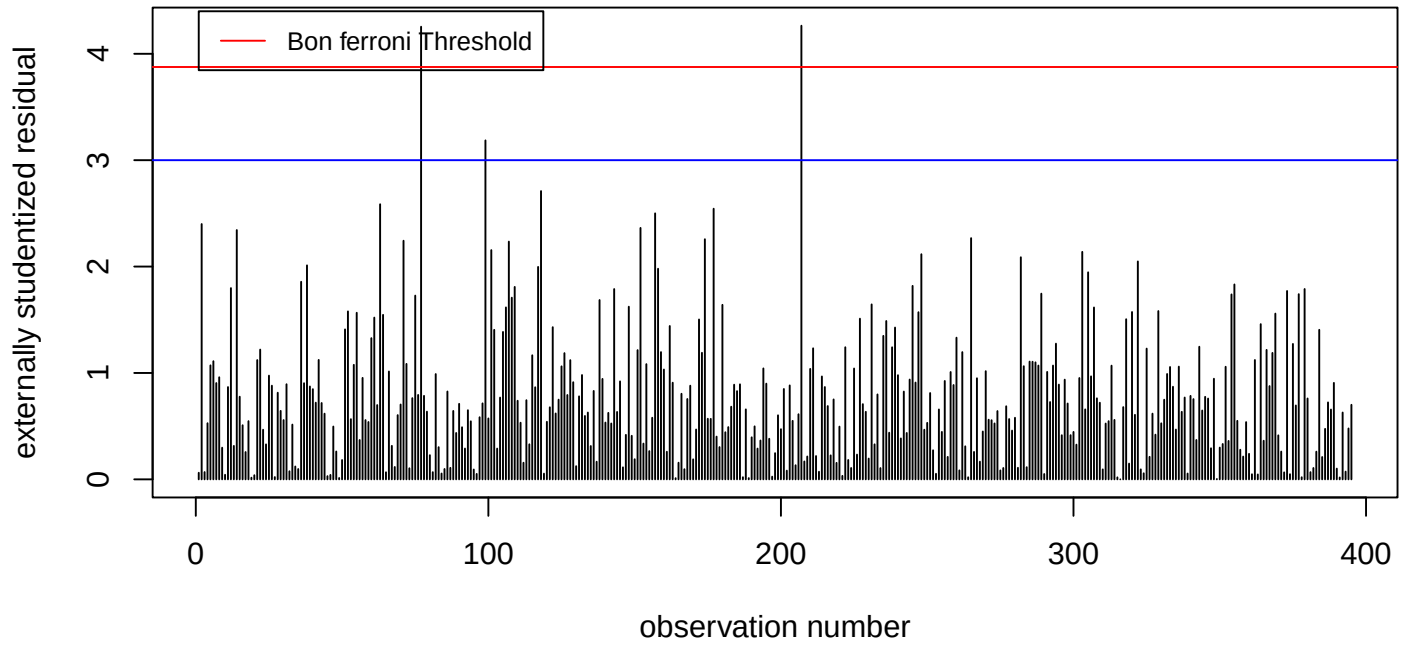
OUTLIERS

Firstly we look at potential outliers by looking at the externally studentized residuals. We assign observation numbers so that we can keep track of the observation we are referring to.

```
# number of observations
n = nrow(data_nona)
# create a new column for data frame
data_nona = data_nona %>% mutate(obs_num = 1:n)
# number of coefficients for our model
p = length(mod1$coefficients)
rst_df1 = data.frame(obs_num = 1:n, rst = rstudent(mod1))
```

1. The studentized residuals follow a t distribution with $n - 1 - p = 354$ degrees of freedom
2. We compare the magnitude of the externally studentized residuals with the magnitude of the $1 - \alpha/(2n)$ quantile (bon-ferroni correction taken into account) and see whether any residuals are greater than such a quantile
3. We take $\alpha = 0.05$ as our significance level
4. We also look at the threshold of a studentized residual of greater than 3 as that could also indicate a potential outlier (The Pennsylvania State University, STAT462 Applied Regression Analysis)

```
# our threshold
threshold = qt(1 - 0.05/(2*n), df = n-p-1, lower.tail = TRUE)
# plot the threshold alongside the residual value of 3
plot(abs(rst_df1$rst), type = 'h', xlab = 'observation number', ylab = 'externally studentized residual')
abline(h = threshold, col = 'red')
abline(h = 3, col = 'blue')
legend(1, 4.4, legend = c('Bon ferroni Threshold'), col = c('red'), lty = 1, cex = 0.8)
```



```
# see if the magnitude of any of our studentized residuals are greater than the threshold
rst_df1 %>% filter(abs(rst) > threshold) %>% arrange(desc(abs(rst)))
```

```
##     obs_num      rst
## 207     207 4.264002
## 77       77 4.254802
```

```
rst_df1 %>% filter(abs(rst) > 3) %>% arrange(desc(abs(rst)))
```

```
##     obs_num      rst
## 207     207 4.264002
## 77       77 4.254802
## 99       99 3.186248
```

- We flag two observations: 77, and 207 for having exceptionally high externally studentized residuals.
- Furthermore we flag another observation: 99 having studentized residuals above 3

HIGH LEVERAGE

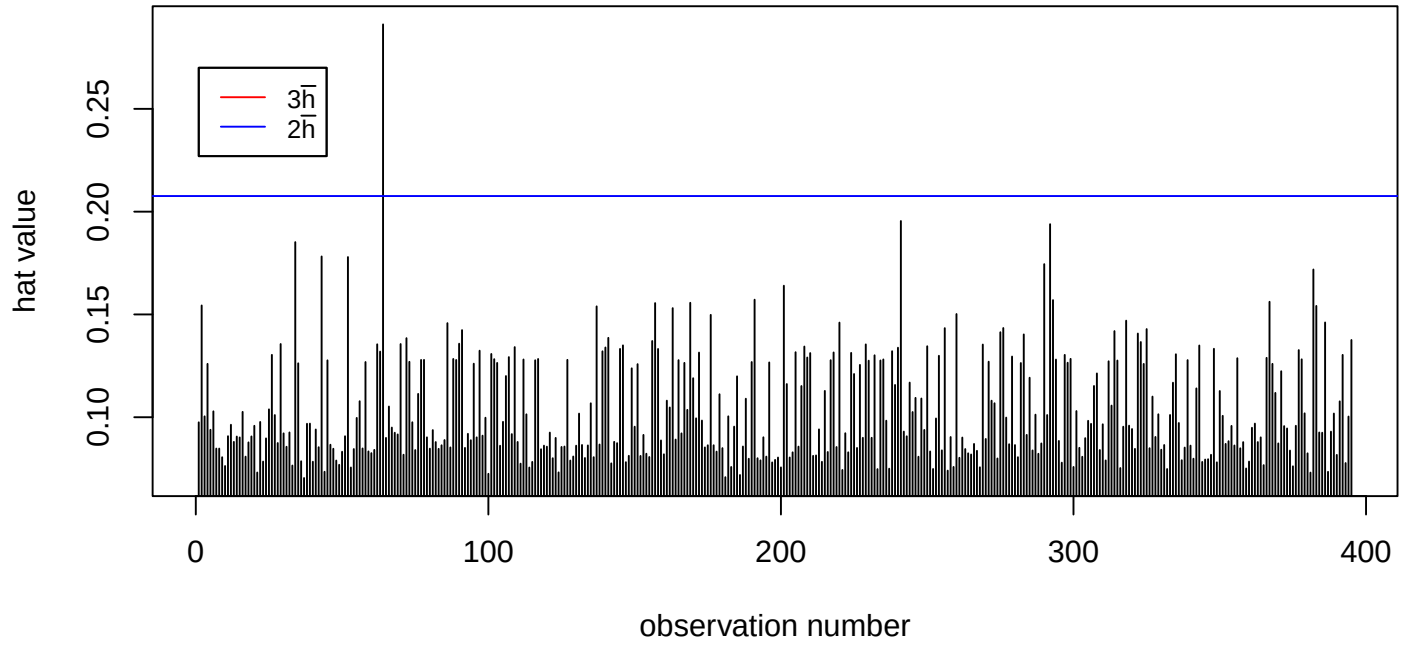
We use the `influence.measures()` command on our model to obtain values to do with leverage and influence:

```
influence_measures_1 = influence.measures(mod1)
influence_df_1 = data.frame(obs_num = 1:n, influence_measures_1$infmat, row.names = NULL)
```

We firstly look at leverage. To see the observations which have high leverage we:

1. Define the average leverage: $\bar{h} = \frac{p}{n}$
2. See which observations have hat value greater than $2\bar{h}$ or $3\bar{h}$

```
h_bar = p/n
plot(influence_df_1$hat, type = 'h', xlab = 'observation number', ylab = 'hat value')
abline(h = 3*h_bar, col = 'red')
abline(h = 2*h_bar, col = 'blue')
# for the legend
three_h_bar = TeX(r'($3\bar{h}$)')
two_h_bar = TeX(r'($2\bar{h}$)')
legend(1, 0.27, legend = c(three_h_bar, two_h_bar), col = c('red', 'blue'), lty = c(1, 1), cex = c(0.8, 0.8))
```



```
# for observations which have leverage greater than 2(average leverage)
influence_df_1 %>% dplyr::select(obs_num, hat) %>% filter(abs(hat) > 2*h_bar) %>% arrange(desc(hat))
```

```
##   obs_num       hat
## 1      64 0.2910959
```

```
# for observations which have leverage greater than 3(average leverage)
influence_df_1 %>% dplyr::select(obs_num, hat) %>% filter(abs(hat) > 3*h_bar) %>% arrange(desc(hat))
```

```
## [1] obs_num hat
## <0 rows> (or 0-length row.names)
```

- We see that the only observation which is higher than $2\bar{h}$ is observation: 64

```
data_nona %>% filter(obs_num %in% c(64))
```

```
##   state year  miles fatalities seatbelt speed65 speed70 drinkage alcohol income
## 1    CA 1985 207600  0.0238921    0.258      no      no      yes      no  16523
##        age enforce obs_num
## 1 33.80183      no      64
```

```
summary(data_nona)
```

```
##      state          year          miles          fatalities
##  NE     : 15   Min.   :1983   Min.   :  3316   Min.   :0.008327
##  AL     : 14   1st Qu.:1989   1st Qu.: 12724   1st Qu.:0.016493
##  KY     : 14   Median :1992   Median : 34003   Median :0.019895
##  MI     : 14   Mean   :1992   Mean   : 45754   Mean   :0.020270
##  MT     : 14   3rd Qu.:1995   3rd Qu.: 59378   3rd Qu.:0.023536
##  NJ     : 14   Max.   :1997   Max.   :285612   Max.   :0.034349
##  (Other):310
##     seatbelt     speed65   speed70   drinkage  alcohol       income
##  Min.   :0.060   no : 93   no :359   no : 14   no :350   Min.   : 9696
##  1st Qu.:0.412   yes:302   yes: 36   yes:381   yes: 45   1st Qu.:16460
##  Median :0.550                                         Median :19290
##  Mean   :0.523                                         Mean   :19754
##  3rd Qu.:0.641                                         3rd Qu.:22862
##  Max.   :0.870                                         Max.   :35863
##
##       age             enforce       obs_num
##  Min.   :29.59   no        : 89   Min.   :  1.0
##  1st Qu.:34.66   primary   : 63   1st Qu.: 99.5
##  Median :35.56   secondary:243   Median :198.0
##  Mean   :35.47                   Mean   :198.0
##  3rd Qu.:36.26                   3rd Qu.:296.5
##  Max.   :39.17                   Max.   :395.0
##
```

- We see that observation 64 has high leverage because it has relatively low value for `seatbelt` and high value for `miles`

INFLUENTIAL POINTS

We now observe the presence of influential points by looking at the dffits value and the cooks distance

```
influence_df_1 %>% arrange(desc(abs(dffit))) %>% dplyr::select(obs_num, dffit) %>% head(10)
```

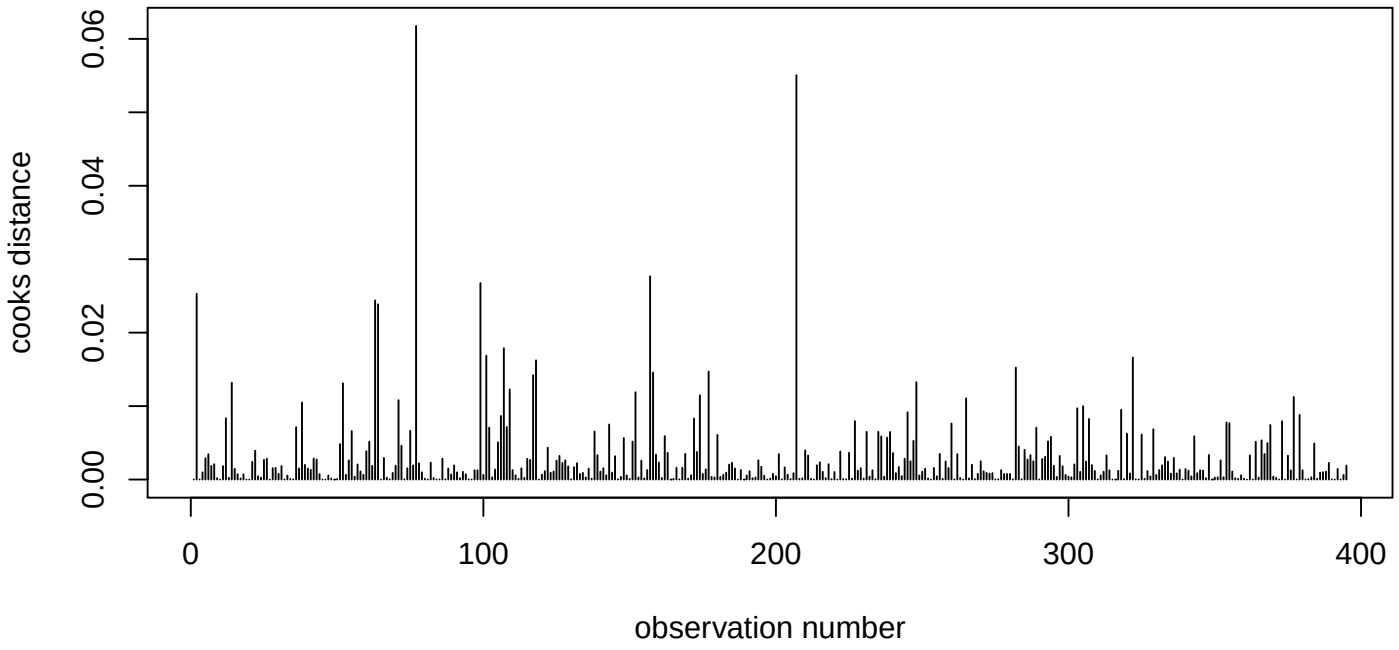
```
##    obs_num      dffit
## 1       77  1.6293238
```

```
## 2      207  1.5384469
## 3      157 -1.0730365
## 4       99  1.0606997
## 5        2 -1.0254195
## 6       63  1.0081511
## 7       64 -0.9909785
## 8      107  0.8610230
## 9      101  0.8359410
## 10     322  0.8289555
```

```
influence_df_1 %>% arrange(desc(abs(cook.d))) %>% dplyr::select(obs_num, cook.d) %>% head(10)
```

```
##    obs_num     cook.d
## 1       77 0.06176456
## 2      207 0.05505514
## 3      157 0.02767257
## 4       99 0.02674950
## 5        2 0.02530572
## 6       63 0.02439752
## 7       64 0.02385837
## 8      107 0.01788025
## 9      101 0.01687017
## 10     322 0.01661020
```

```
plot(abs(influence_df_1$cook.d), type = 'h', xlab = 'observation number', ylab = 'cooks distance')
```



We note that the observations with observation numbers corresponding to 77 and 207 stand out in particular.

```
data_nona %>% filter(obs_num %in% c(77, 207))
```

```
##   state year miles fatalities seatbelt speed65 speed70 drinkage alcohol income
## 1    AK 1993  3918 0.03011741   0.6900     yes      no      yes      no  22711
## 2    MT 1997  9392 0.02821550   0.7255     yes      no      yes      no  19660
##        age   enforce obs_num
## 1 30.46439 secondary      77
## 2 36.82977 secondary     207
```

- Observation 77 is influential becuase it is an outlier
- Observation 207 is also influential because it is an outlier.

Smaller Model

Our smaller model is:

$$fatalities \sim state + year + income + miles$$

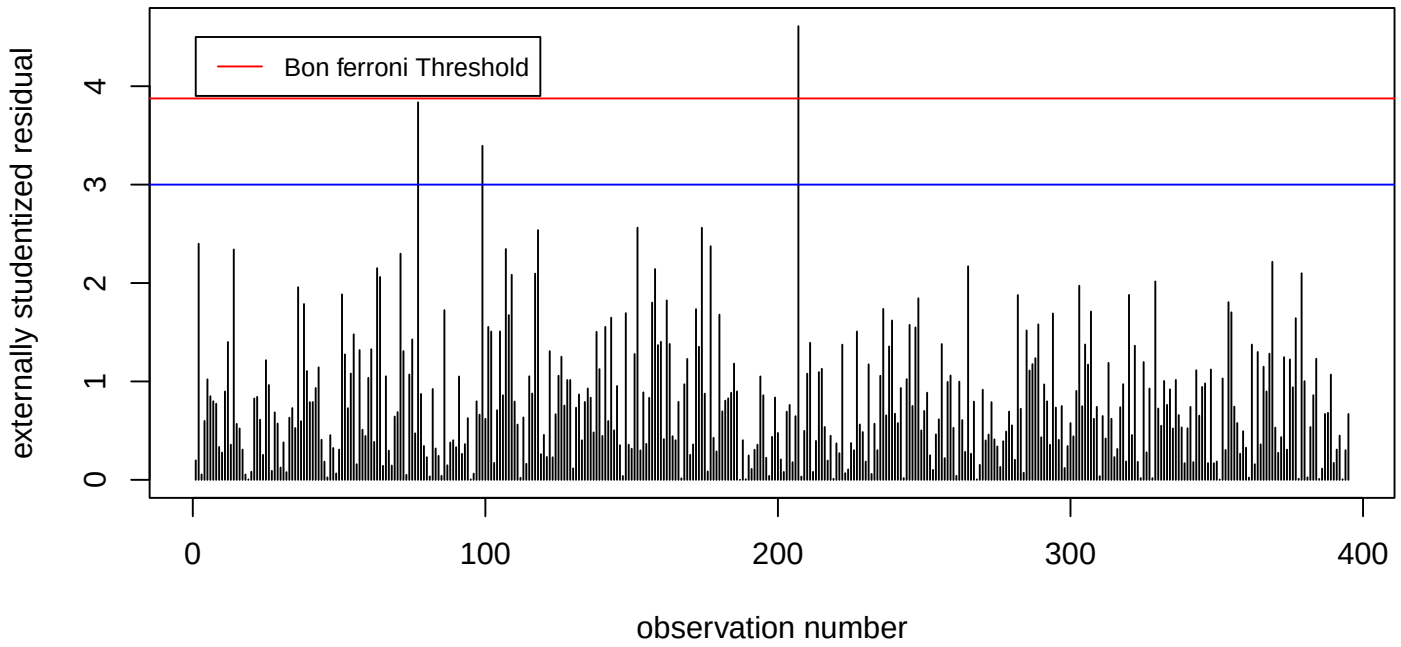
OUTLIERS

Firstly we look at potential outliers by looking at the externally studentized residuals. We assign observation numbers so that we can keep track of the observation we are referring to.

```
# number of coefficients for our model
p = length(mod2$coefficients)
rst_df2 = data.frame(obs_num = 1:n, rst = rstudent(mod2))
```

1. The studentized residuals follow a t distribution with $n - 1 - p = 357$ degrees of freedom
2. We compare the magnitude of the externally studentized residuals with the magnitude of the $1 - \alpha/(2n)$ quantile (bon-ferroni correction taken into account) and see whether any residuals are greater than such a quantile
3. We take $\alpha = 0.05$ as our significance level
4. We also look at the threshold of a studentized residual of greater than 3 as that could also indicate a potential outlier (The Pennsylvania State University, STAT462 Applied Regression Analysis)

```
# our threshold
threshold = qt(1 - 0.05/(2*n), df = n-p-1, lower.tail = TRUE)
# plot the threshold alongside the residual value of 3
plot(abs(rst_df2$rst), type = 'h', xlab = 'observation number', ylab = 'externally studentized residual')
abline(h = threshold, col = 'red')
abline(h = 3, col = 'blue')
legend(1, 4.5, legend = c('Bon ferroni Threshold'), col = c('red'), lty = 1, cex = 0.8)
```




```
# see if the magnitude of any of our studentized residuals are greater than the threshold
rst_df2 %>% filter(abs(rst) > threshold) %>% arrange(desc(abs(rst)))
```

```
##     obs_num      rst
```

```
## 207     207 4.610382
```

```
rst_df2 %>% filter(abs(rst) > 3) %>% arrange(desc(abs(rst)))
```

```
##     obs_num      rst
## 207     207 4.610382
## 77       77 3.836338
## 99       99 3.393801
```

- We flag the observation: 207 for having exceptionally high externally studentized residuals.
- Furthermore we flag another two observation: 77 and 99 for having studentized residual above 3.

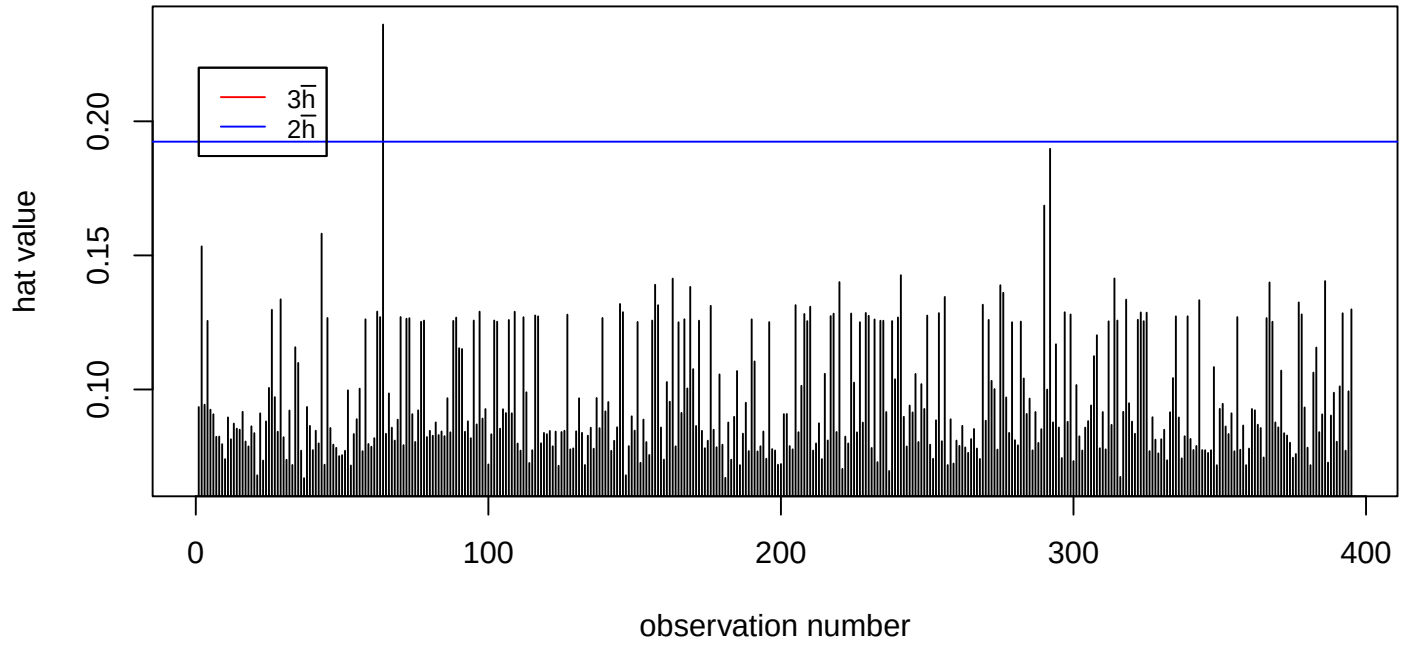
HIGH LEVERAGE

```
influence_measures_2 = influence.measures(mod2)
influence_df_2 = data.frame(obs_num = 1:n, influence_measures_2$infmat, row.names = NULL)
```

We firstly look at leverage. To see the observations which have high leverage we:

1. Define the average leverage: $\bar{h} = \frac{p}{n}$
2. See which observations have hat value greater than $2\bar{h}$ or $3\bar{h}$

```
h_bar = p/n
plot(influence_df_2$hat, type = 'h', xlab = 'observation number', ylab = 'hat value')
abline(h = 3*h_bar, col = 'red')
abline(h = 2*h_bar, col = 'blue')
# for the legend
three_h_bar = TeX(r'($3\bar{h}$)')
two_h_bar = TeX(r'($2\bar{h}$)')
legend(1, 0.22, legend = c(three_h_bar, two_h_bar), col = c('red', 'blue'), lty = c(1, 1), cex = c(0.8, 0.8))
```




```
# for observations which have leverage greater than 2(average leverage)
influence_df_2 %>% dplyr::select(obs_num, hat) %>% filter(abs(hat) > 2*h_bar) %>% arrange(desc(hat))
```

```
##   obs_num       hat
## 1      64 0.2360771
```

```
# for observations which have leverage greater than 3(average leverage)
influence_df_2 %>% dplyr::select(obs_num, hat) %>% filter(abs(hat) > 3*h_bar) %>% arrange(desc(hat))
```

```
## [1] obs_num hat
## <0 rows> (or 0-length row.names)
```

- We see that the observation which is higher than $2\bar{h}$ is observation 64
- We see that there are no observations which are higher than $3\bar{h}$

```
data_nona %>% filter(obs_num %in% c(64))
```

```
##   state year  miles fatalities seatbelt speed65 speed70 drinkage alcohol income
## 1    CA 1985 207600  0.0238921    0.258      no      no      yes      no  16523
##        age enforce obs_num
## 1 33.80183      no      64
```

```
summary(data_nona)
```

```
##      state          year          miles          fatalities
##  NE     : 15   Min.   :1983   Min.   :  3316   Min.   :0.008327
##  AL     : 14   1st Qu.:1989   1st Qu.: 12724   1st Qu.:0.016493
##  KY     : 14   Median :1992   Median : 34003   Median :0.019895
##  MI     : 14   Mean   :1992   Mean   : 45754   Mean   :0.020270
##  MT     : 14   3rd Qu.:1995   3rd Qu.: 59378   3rd Qu.:0.023536
##  NJ     : 14   Max.   :1997   Max.   :285612   Max.   :0.034349
##  (Other):310
##     seatbelt     speed65   speed70   drinkage  alcohol       income
##  Min.   :0.060   no : 93   no :359   no : 14   no :350   Min.   : 9696
##  1st Qu.:0.412   yes:302   yes: 36   yes:381   yes: 45   1st Qu.:16460
##  Median :0.550                                         Median :19290
##  Mean   :0.523                                         Mean   :19754
##  3rd Qu.:0.641                                         3rd Qu.:22862
##  Max.   :0.870                                         Max.   :35863
##
##       age             enforce       obs_num
##  Min.   :29.59   no        : 89   Min.   :  1.0
##  1st Qu.:34.66   primary   : 63   1st Qu.: 99.5
##  Median :35.56   secondary:243   Median :198.0
##  Mean   :35.47                   Mean   :198.0
##  3rd Qu.:36.26                   3rd Qu.:296.5
##  Max.   :39.17                   Max.   :395.0
##
```

- We see that observation 64 has high leverage because it has relatively low value for `seatbelt` and high value for `miles`

INFLUENTIAL POINTS

We now observe the presence of influential points by looking at the dffits value and the cooks distance

```
influence_df_2 %>% arrange(desc(abs(dffit))) %>% dplyr::select(obs_num, dffit) %>% head(10)
```

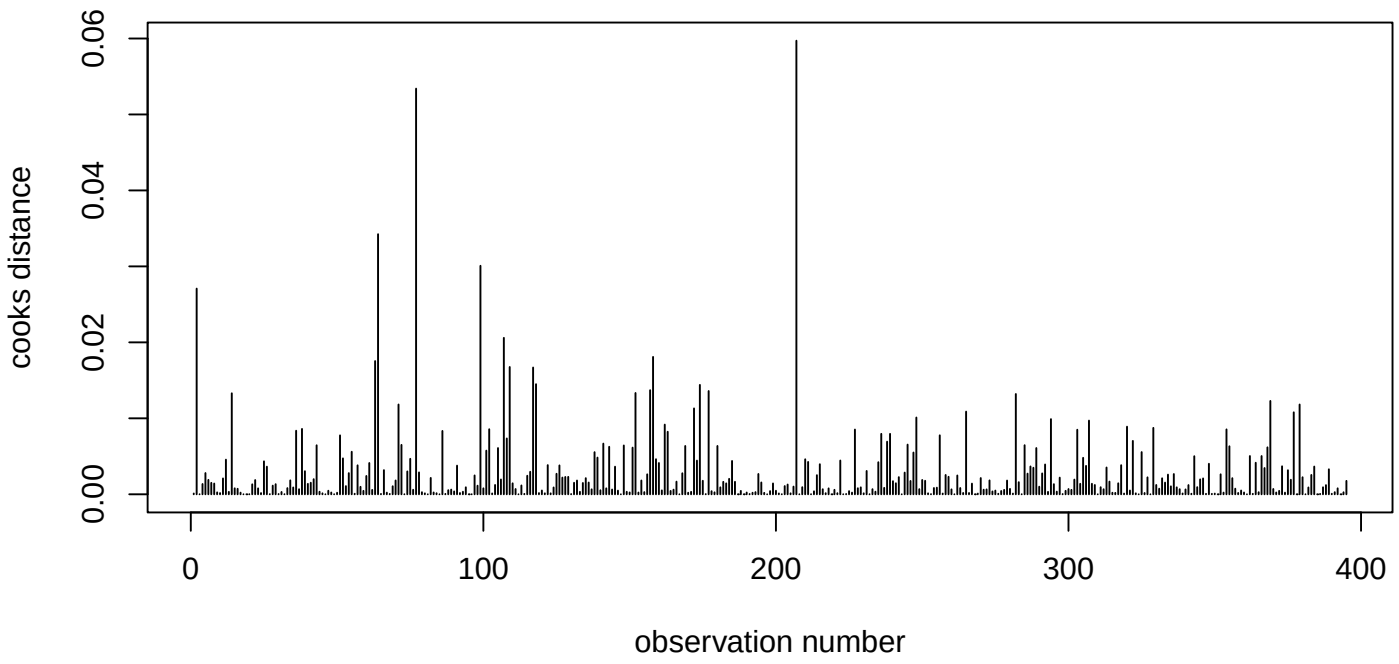
```
##    obs_num      dffit
## 1      207  1.5484739
## 2       77  1.4516542
## 3       64 -1.1459051
## 4       99  1.0847806
## 5        2 -1.0210818
## 6      107  0.8902310
```

```
## 7       158  0.8332262
## 8        63  0.8205617
## 9       109 -0.8017694
## 10      117 -0.8001804
```

```
influence_df_2 %>% arrange(desc(abs(cook.d))) %>% dplyr::select(obs_num, cook.d) %>% head(10)
```

```
##    obs_num     cook.d
## 1      207 0.05971132
## 2       77 0.05340327
## 3       64 0.03424357
## 4       99 0.03008084
## 5        2 0.02707633
## 6      107 0.02059585
## 7      158 0.01808836
## 8       63 0.01754070
## 9      109 0.01675985
## 10     117 0.01669116
```

```
plot(abs(influence_df_2$cook.d), type = 'h', xlab = 'observation number', ylab = 'cooks distance')
```



We note that the observations with observation numbers corresponding to 77 and 207 stand out in particular.

```
data_nona %>% filter(obs_num %in% c(77, 207))
```

```
##   state year miles fatalities seatbelt speed65 speed70 drinkage alcohol income
## 1    AK 1993  3918 0.03011741   0.6900     yes      no      yes      no  22711
## 2    MT 1997  9392 0.02821550   0.7255     yes      no      yes      no  19660
##        age   enforce obs_num
## 1 30.46439 secondary      77
## 2 36.82977 secondary     207
```

- Observation 77 is influential becuase it is an outlier
- Observation 207 is also influential because it is an outlier.

Model Evaluation & Comparison

Our model selection procedure earlier - which considered information criterion - led to a sparse and a large model. To assess the generalisation ability as well as the stability of the model on unseen data, we perform Repeated (N=10) 5-fold Cross-Validation on both models.

$$fatalities \sim state + year + age + income + miles + seatbelt + miles : seatbelt$$

$$fatalities \sim state + year + income + miles$$

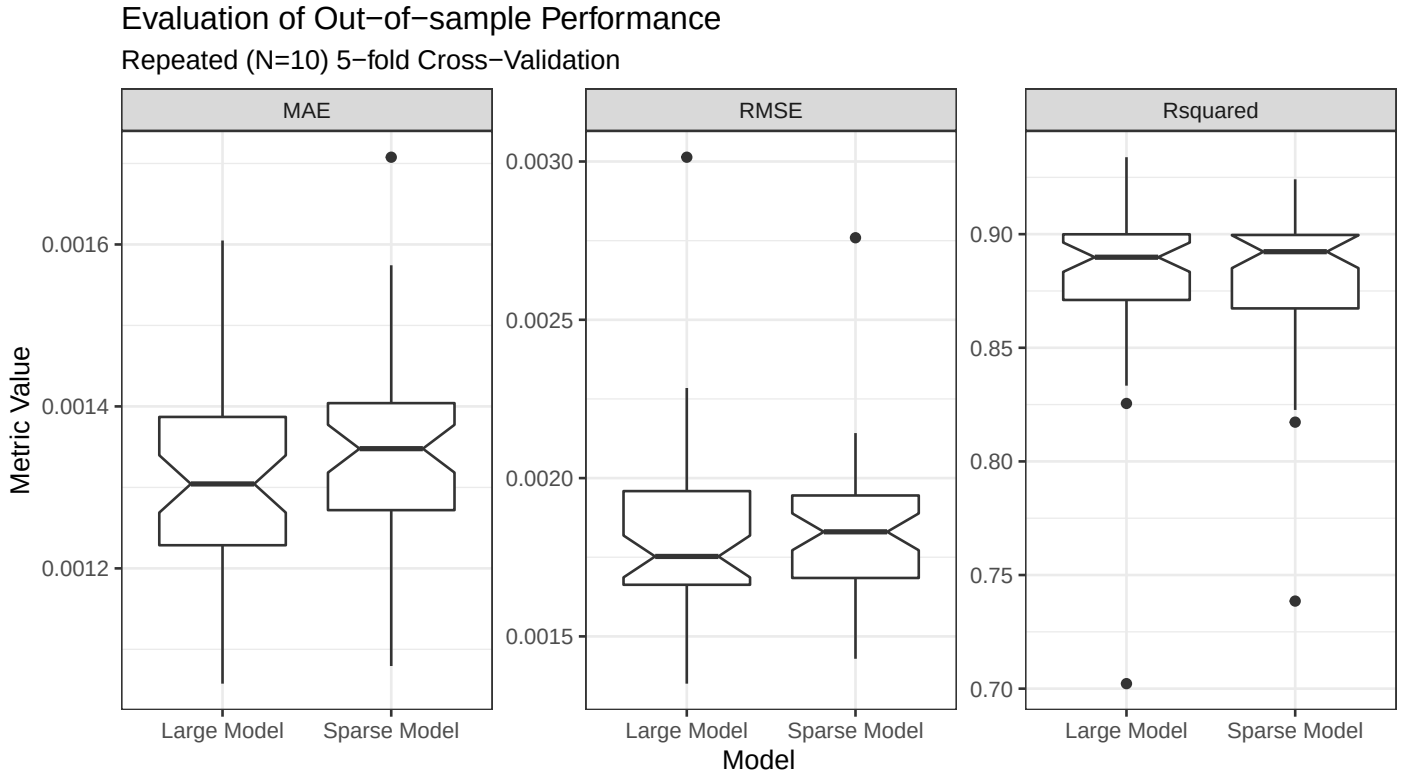
We chose to use three separate metrics, mean absolute error (MAE), root-mean-squared error (RMSE) and R-squared as each metric offers slightly different insight. For instance, MAE weights large and small errors equally whereas RMSE punishes larger errors severely.

In all three metrics, the notches of the boxplots overlap suggesting that there isn't sufficient evidence to support a significant difference in the out-of-sample performance between the two models. Therefore, in the interest of parsimony, we pick the model with the lesser number of covariates i.e. the sparse model.

```
# This function drops NA rows by default
# This function performs repeated k-fold cross validation
cross_validation = function(cvK, n_sim, df, formula){

  # Setup CV method
  cv_method = trainControl(method = "repeatedcv",
                           number = cvK,
                           repeats = n_sim,
                           returnData = TRUE,
                           returnResamp = "final")


  # Train the model
  model = train(formula,
                data = df %>% drop_na(), # Drop NA rows
                method = "lm",
                trControl = cv_method )

  return(model)

}

# This function takes two models and names for models and returns CV plot
generate_cv_plot = function(model1, model2, name1, name2) {

  # Get the corre
  cv1 = model1$resample %>%
    mutate(Model = name1)
  cv2 = model2$resample %>%
    mutate(Model = name2)
  rbind(cv1, cv2) %>%
    dplyr::select(RMSE, Rsquared, MAE, Model) %>%
    group_by(Model) %>%
    pivot_longer(., cols = 1:3, names_to = "Metric", values_to = "Value") %>%
    ungroup() %>%
    ggplot() +
    aes(x = Model, y = Value) %>%
    geom_boxplot(notch = TRUE) +
```

```
    facet_wrap(facets = ~Metric, scales = "free") +
    theme_bw() +
    labs(
      x = "Model",
      y = "Metric Value",
      title = "Evaluation of Out-of-sample Performance",
      subtitle = "Repeated (N=10) 5-fold Cross-Validation"
    )
}
```

```
# Here I calling cross-validation on two separate models
model1 = cross_validation(cvK = 5,
                          n_sim = 10,
                          df = df,
                          formula = fatalities ~ state + year + age + income + miles + seatbelt + miles:seatb

model2 = cross_validation(cvK = 5,
                          n_sim = 10,
                          df = df,
                          formula = fatalities ~ state + year + income + miles)

# Here I generate a cross-validation plot comparing the two models
generate_cv_plot(model1,
                 model2,
                 "Large Model",
                 "Sparse Model")
```



Model Improvement

For model improvement we chose to use LASSO (L1) regression.

There are two primary motivations for this.

Firstly, a minor consideration is that there are still terms i.e. `miles` and `income` with high multicollinearity. L1 regularisation is particularly useful in tackling multicollinearity as it will penalise variables with minimal contribution to the outcome by shrinking their respective coefficients (Setiyorini et al., 2017).

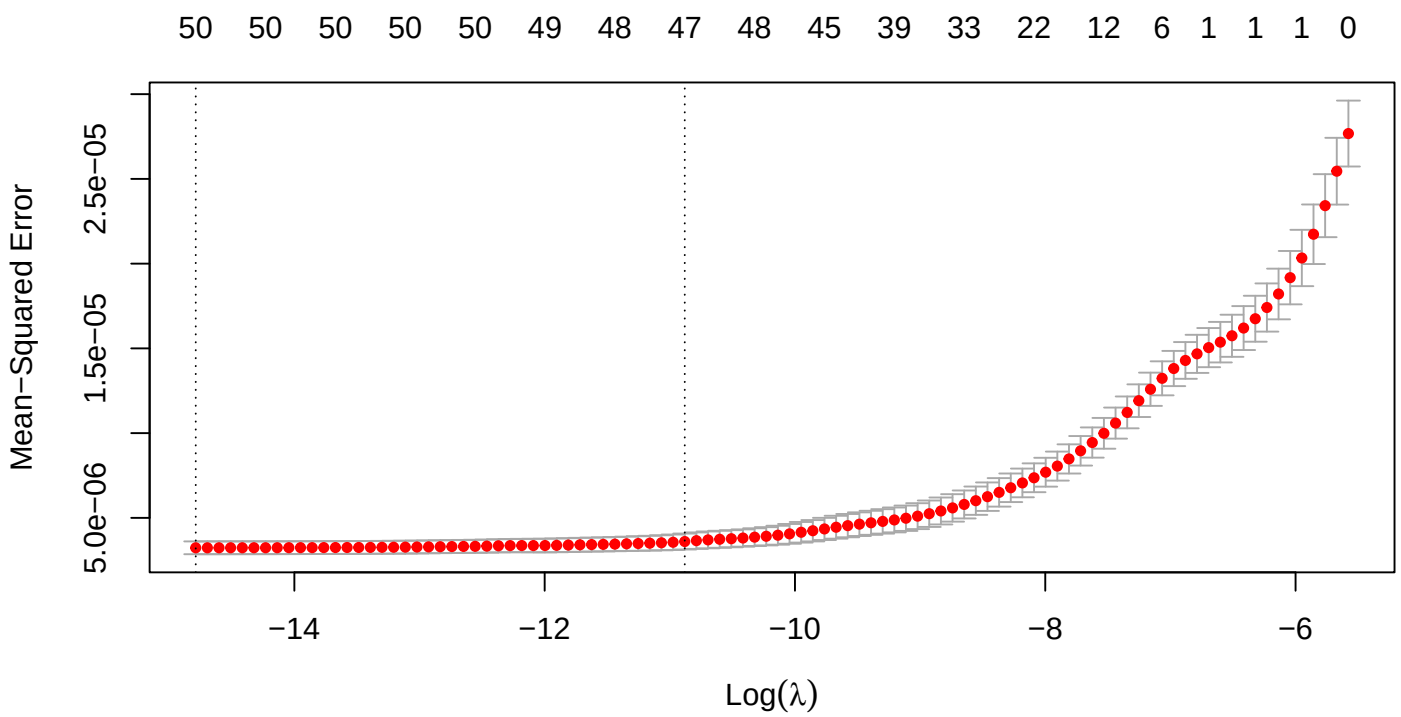
```
car::vif(lm(fatalities ~ state + year + income + miles, df))
```

```
##             GVIF Df GVIF^(1/(2*Df))
## state  683.04376 34        1.100736
## year    19.46923 14        1.111855
## income  29.63178  1        5.443508
## miles   55.59910  1        7.456481
```

Secondly, our data contains 395 data points after removing rows with `NA`. Current guidance, suggests a rule of thumb of about 10 times the amount of data points as there are number of model parameters that are being estimated by the data (Harrell et al., 1996). Given our current final model has a large ratio of estimates to data points, i.e. 52 parametric estimates, our model could be over-fitting i.e. the variance is large. This is another key motivation for selecting L1 over L2 as it allows coefficients to be reduced to 0 if necessary. This should theoretically improve the stability of model as we are essentially increasing the bias slightly in exchange for a lower variance.

Here, lambda is the 'severity of punishment' when estimating coefficients and we use 10-fold cross-validation to select the lambda that produces the lowest out-of-sample MSE. The plot shows that the process still picked a model with 51 parametric estimates albeit with slightly different coefficents.

```
# Generate a model matrix for covariates
X = df %>% drop_na %>% dplyr::select(state, year, income, miles)
model_matrix = model.matrix(~ state + year + income + miles, X)
# Outcome fatalities
y = df %>% drop_na %>% dplyr::select(fatalities) %>% pull()

# Example of a single model w/ LASSO
model = glmnet(model_matrix,y)

# Performs 10-fold cross validation
cv_model = cv.glmnet(x = model_matrix,
                     y = y,
                     # alpha specifies LASSO
                     alpha = 1,
                     type.measure='mse'
                     )

plot(cv_model)
```




```
#coef(cv_model, s = "lambda.min")

lambda_opt=cv_model$lambda.min
```

Here, it's reasonable that we end up not seeing any marked improvement in RMSE performance. This is likely because the rule of thumb is just a general guidance and our model may actually not be over-fitting at all.

```
# Following gives the MSE of the minimum lambda
lasso_MSE = min(cv_model$cvm)
lasso_RMSE = sqrt(lasso_MSE)


# Following calculates CV RMSE for original model

# Setup CV method
cv_method = trainControl(method = "cv",
                         number = 10,
                         returnData = TRUE,
                         returnResamp = "final")

# Train the model
model1 = train(fatalities ~ state + year + miles + income,
               data = data_nona, # Drop NA rows
               method = "lm",
               trControl = cv_method )

original_model = model1$resample %>%
  dplyr::select(RMSE) %>%
  pull() %>%
  mean()

cbind(LASSO = round(lasso_RMSE, 5), Original = round(original_model, 5))
```

```
##       LASSO Original
```

```
## [1,] 0.0018  0.00191
```

Conclusion

Discussion of Findings

Our original goal was to build a model that struck a balance between good fit but a small number of predictors. Our primary finding was that a sparse model with few predictors performed just as well out-of-sample as a larger model despite some of the variables in the larger model appearing statistically significant in-sample.

The covariates of the final model included `state`, `year`, `income` and `miles`. State and year are both reasonable as they were the primary blocking variables in the original study Cohen & Einav (2003). Its possible that the lower socioeconomic states could be associated with poorer overall regulation and dangerous road conditions which would explain its negative coefficient.

Surprisingly, it did not include variables one might intuitively associate more closely with fatalities such as implementation of enforcement, speed limits and seatbelt usage.

Limitations

Our interesting results could be explained by some of the modelling and data limitations. One key limitation of our model is that it is trained on observational data which restricts our observations only to incidents where the outcome has resulted in a death. This creates an unfair bias towards safety belts not being influential on fatalities because an incident where safety belts prevented fatalities would not be included in this data (Levitt & Porter, 2001).

Another important limitation during our modelling process is that the sparsity of the `seatbelt` variable forced us to drop rows containing `NA`. Subsequently, a lot of useful information from other variables is lost, reducing the power of our model.

Another interesting limitation rests in the definition of fatalities. This dataset assumes no difference between occupant i.e. in the vehicle, and non-occupant i.e. outside the vehicle, fatalities. However, some variables e.g. `seatbelt` would only realistically influence the survivability of occupants. It would be ridiculous to suggest the survivability of a pedestrian hit by a car could be influenced by the driver's seatbelt usage. Consequently, we are limited in granularity of interpretations we can make about the covariates present and absent from the model.

Contribution of Persons in the report

1. For the data description and visualisation: Matt and Eva did this
2. For the model building: Mason and Eva did this
3. For the model improvement: Zhuolin and Matt did this
4. For the conclusion: Everyone contributed