Does increasing the number of speed cameras reduce the numbers of car accidents resulting in death in the Australian Capital Territory (ACT).

Abstract

The purpose of this investigation is to try and find some relationship between the number of speed cameras used in the Australian Capital Territory (ACT) and the number of car accidents resulting in death. The suburb with the most deaths was filtered and detected as Tuggeranong (*Williams*. E, 2018). Gower's similarity metric was then used successfully to identify the top 6 Suburbs which are key areas to improve safety measures. The density of cameras and their effectiveness was investigated in this report by using ggplot(). This investigation shines a light on the most dangerous Suburbs in terms of road fatalities and identifies a need to improve the way in which this data is collected while showing patterns in the data that is available.

Introduction

In the pursuit of road safety, jurisdictions around the world have implemented various measures to mitigate the occurrence and severity of car accidents. Among these measures, speed cameras have emerged as a widely employed technology aimed at enforcing speed limits and deterring speeding behaviours. The Australian Capital Territory (ACT), with its commitment to ensuring road safety, has extensively adopted speed cameras as a means of reducing car accident fatalities.

The question of whether speed cameras can effectively contribute to the reduction of car accident fatalities in the ACT is a subject of great significance and ongoing investigation. (Kinra. S, Pilkington. P, 2005) *"All but one of the studies showed effectiveness of cameras up to three years or less after their introduction"* The study found a significant reduction in fatal and injury crashes at camera sites compared to control sites without cameras.

A study has shown that speed cameras have been effective in reducing car crash numbers. (Wilson. C, Willis. C, Hendrikz. J.K, Le Brocque. R, Bellamy. N 2010) *"All 28 studies found a lower number of crashes in the speed camera areas after implementation of the program".* Through this exploration, we hope to gain valuable insights into the efficacy of speed cameras in reducing car accident fatalities, contributing to the ongoing discourse on road safety measures in the ACT.

Data

Data location - www.data.act.gov.au

https://www.data.act.gov.au/api/views/426s-vdu4-0(CSV) - Traffic speed camera locations
https://www.data.act.gov.au/Transport/Traffic-speed-camera-locations/426s-vdu4Mobile - Speed Camera Visits and Stays
https://www.data.act.gov.au/api/views/6jn4-m8rx-0(CSV) - ACT Road Crash Data

Table 1: Table showing the datasets used in this assessment

Dataset	Number of Observations	Number of Variables	Original/converted
crash	71796	18	Original
locations	1233	9	Original
speed_camera	62620	9	Original
speed_locations	64273	20	left_join() data
Camera.accidents	133	11	full_join() data

Table 2: Table showing the variable types used in this assessment

Name of Variable	Variable class	Other information
SUBURB LOCATION	Factor / Categorical	Switched to factor using as.factor() 133 levels
cars_checked	Integer	Created using mutate()
number_of_cameras	Integer	Created using mutate()
max.speed	Integer	Created using mutate() (Km/H)
avg.Speed	Numeric	Created using mutate() (Km/H)
Posted.Speed	Numeric	(Km/H)
CRASH_SEVERITY	Factor / Categorical	Switched to factor using as.factor() 3 levels
ROAD_CONDITION	Factor / Categorical	Switched to factor using as.factor() 4 levels
number_of_crashes	Numeric	Created using mutate()
scans_per_cam	Numeric	Created using mutate()
Speed_above_limit	Integer	Created using mutate()

For this assignment, the three datasets that were combined were all obtained from the Australian government website. Each of the data were collected as an observational type study. As this is an observational type study, no interventions would have been introduced into the collection of the data. When reading the metadata for the speed_camera, it says that no speed data will be available after 2018. So I will need to filter the years after 2018 out of my data.

Methods

First the packages are loaded into r using library()[1] (dplyr, stingr, lubridate, cluster, ggplot2, VIM) and the datasets are to be imported using read.csv()[2]. The variable 'Camera.location' from the 'speed_camera' dataset is to be renamed using dplyr rename() to 'LOCATION.CODE. This is to be used as a primary key to match the two datasets 'speed_camera' and 'locations' by left_join()[3]. The 'date' requires to be reformatted, for this I will use the lubridate package to split the date up into three columns, 'year', 'month' and 'day'[4] and the years after 2018 are filtered out of the data.

For the next process, the suburb will be extracted from the address line by using sub(). This will then be merged into the dataframe using cbind()[5]. The str_detect() is required as some of the data is not from ACT and from NSW, this will be subsetted using base r square brackets[6]. The column is to be renamed to 'SUBURB_LOCATION' as this will be used to join the datasets later on. When observing the data using unique() for the 'crash' dataset variable 'SUBURB_LOCATION', some clean up processing is required as there are some formatting issues. These were fixed by using gsub()[7]. When the two datasets were scanned using table() it was noticed that Canberra was split up in the 'crash' dataset as 'Canberra airport', 'city' and 'Canberra central'. These will be converted into one category 'Canberra' to match the 'speed_locations' dataset also using gsub()[8]. The objects in this column require to be reformatted to match the objects in the column for the other dataset to be merged. This will be achieved using toupper() and tolower()[9]. The Crash data date will also be converted into year and dates after 2018 will be filtered out[10].

For the final merge, I will first subset the two sets of data by using select(). The new data set 'number of speed_cameras' will be filtered as only the results where the camera detected speeding should be selected, this will be achieved by using filter() with 'or' statement. The two sets of data are then grouped_by() 'SUBURB_LOCATION' and then mutate() to show the count of each data[11]. The mutate() will also be needed to show the sum, max and average as the group_by() requires this for it to be represented in the data. This data will then be joined using full_join() using 'SUBURB_LOCATION' as the primary key[12]. Row 96 will be removed from the data as this is the data that is not attributed to a suburb. Finally, from this data, two new columns will be made which will be 'speed_above_limit' which will deduct the 'Posted.Speed' from the 'max.speed' and 'scans_per_cam' which will divide the 'cars_checked' by the 'number_of_cameras' used for each suburb. These two columns are formed using mutate()[13].

As the data is full_joined, there will be some NAs coerced in the 'number_of_crashes' and speed columns. Using sum(is.na() there are four missing observations for the 'number_of_crashes'. As these suburbs had a number of cameras positioned, it will be assumed that the number of crashes may not have been recorded due to data collection mismatch. I will first look to see if there is any correlation between my datasets using the cor() and type "pearson" for a pearson correlation coefficient[14]. There does appear to be a positive correlation between the number of crashes and the number of cameras at (0.63). There also seems to be weak correlation for the 'max.speed' and 'number_of_crashes' (0.42)[15]

Because of these reasons, I will impute the missing data using a linear regression model. To impute the rest of the data, I will use the kNN() from the VIM package(16).

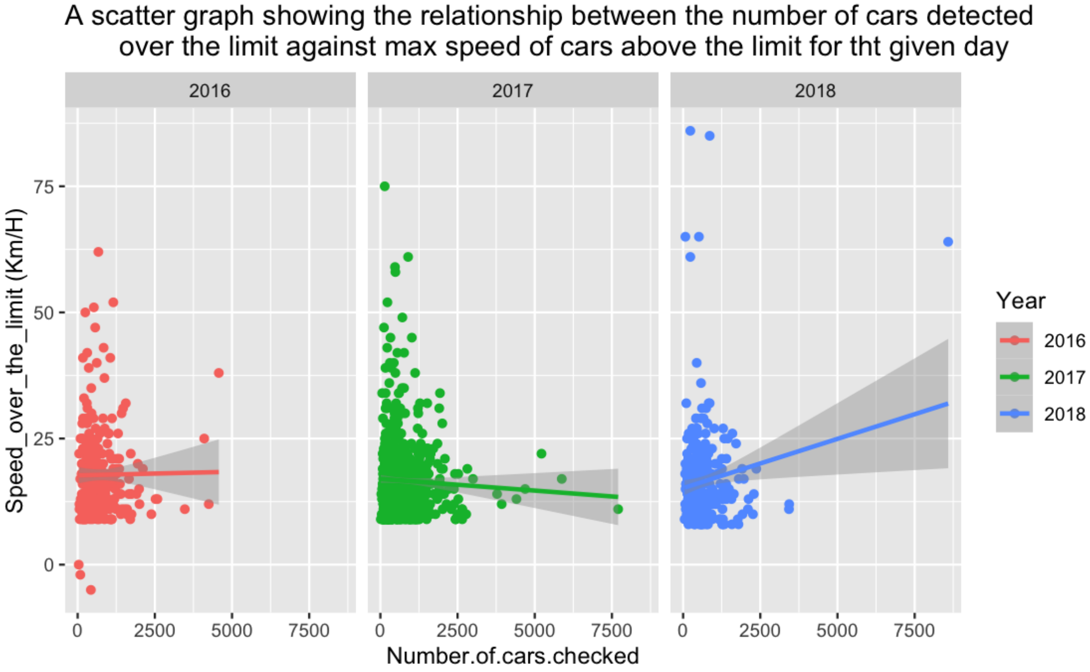
Results

From the 'crash' dataset, I will use the filter() to show only results for "Fatal". This was summarised as a count and arranged() in descending order(17). The top result for this was the suburb "Tuggeranong" (*Williams. E*, 2018) which had more than double the amount of fatalities than the second highest. I have identified this as a key area for traffic control measures to be increased to improve safety.

I would like to perform a similarity measure on the 'Camera.accidents' merged dataset to locate the closely related suburbs to Tuggeranong. I used the gower similarity metric using daisy() for this. The character variables were required to be converted using the as.factor(). The similarity was calculated by deducting the result from 1. This was then converted into a matrix by using the as.matrix(). The matrix was subsetted by the row for "Tuggeranong" which is row 116(18). A function was created called 'get_top_5' which is to be used to select the top five highest similarity measures to Tuggeranong(19). The results were printed using the print() and subsetting using square brackets. The top five suburbs were 'Wright', 'Tennent', 'Tharwa', 'Russell', 'Jerrabomberra'

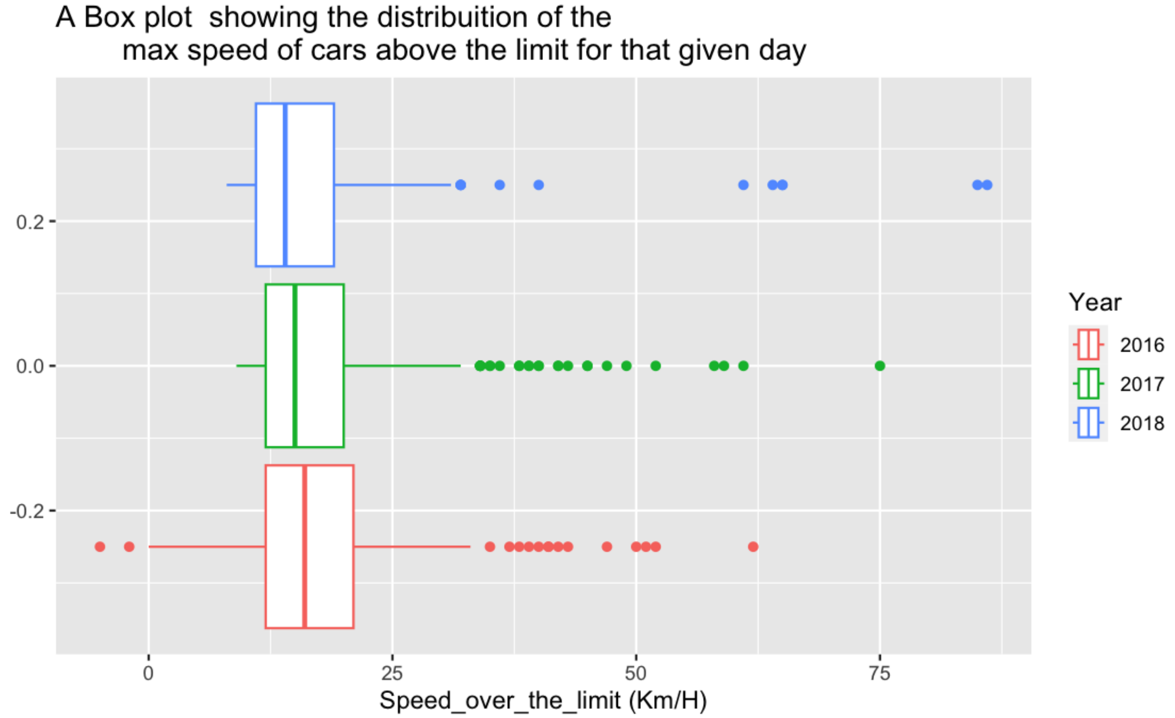
From this information, I wanted to observe the effectiveness of the cameras in tracking speeding and see how effective speed cameras have been by subsetting the data by year. For this I will use the ggplot() including the use of facet_wrap(), geom_point(), geom_smooth as well as labelling my graph correctly using labs()(20).

Graph 1(20):



From Graph 1 it can be seen that the data is fairly consistent in the spread for each of the years recorded. There may be slightly higher speeds recorded for 2017 and 2018. There seems to be a larger size of cameras which checked more cars in 2017. It may be beneficial if some of the cameras were relocated to the above-mentioned suburbs that are high risk. To examine the spread of the data I will look at using a boxplot

Graph 2[21]:



From the boxplot in graph 2, we can see evidence that there are more extreme outliers for the final year of 2018, however, the interquartile range and median does seem to be lower which could be evidence that speed cameras do tend to reduce the amount of speeds driven in the ACT. It could be suggested that speed cameras may reduce the amount of speed driven and therefore the number of crashes as lower speeds would equate to increased control over a vehicle.

Conclusion

It would have been better if the data after 2018 would have been available to use in order to see if the trend of lower speeds would have continued to develop. In this investigation, no hypotheses were tested by using p value with statistical testing such as ANOVA and t-tests. Going into further detail, this would be a good direction to truly explore the relationship of the values in this dataset. It would be beneficial to do more data visualisation on the crash data as well. If possible it would have been useful to find a dataset that is able to use a primary key which merges the crash and speed cameras data rather than subsetting by suburb. The investigation does show that there does seem to be some correlation that speed cameras do make roads safer to drive on and will reduce the number of traffic fatalities.

References:

Kinra. S, Pilkington. P (2005). *Effectiveness of speed cameras in preventing road traffic collisions and related casualties: systematic review*. Doi - 10.1136/bmj.38324.646574.AE.

Swalin. A (2018) *How to handle Missing data.* https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4

Williams. E (2018) *Tuggeranong Parkway is Canberra's speeding danger zone*. https://www.canberratimes.com.au/story/6003116/tuggeranong-parkway-is-canberras-speeding-danger-zone/

Wilson. C, Willis. C, Hendrikz. J.K, Le Brocque. R, Bellamy. N (2010). *Speed cameras and the prevention of road traffic injuries and deaths.* The Cochrane Collaboration.

1# Packages

```
library(dplyr)
library(tidyverse)
library(stringr)
library(lubridate)
library(ggplot2)
library(cluster)
library(VIM)
```

2# Load the dataset

```
speed_camera <-
read.csv("Mobile_Speed_Camera_Visits_and_Stays.csv")

locations <-
read.csv("Traffic_speed_camera_locations.csv")

crash <-
read.csv("ACT_Road_Crash_data.csv")
```

3# left_join the datasets

```
speed_camera <- speed_camera %>%

  rename("LOCATION_CODE" =
"Camera.Location")

speed_locations <-  left_join(speed_camera,
locations, by = "LOCATION_CODE")
```

4# reformat the date

```
speed_locations$Day <-
day(dmy(speed_locations$Date))
speed_locations$Month <-
month(dmy(speed_locations$Date))
speed_locations$Year <-
year(dmy(speed_locations$Date))

View(speed_locations)

speed_locations <- speed_locations %>%
  filter(Year == c(2016, 2017, 2018))
```

5# Extract the suburb from the address

```
extract <- sub('.*,\\s(.*),.*', '\\1',
speed_locations$LOCATION.DESCRIPTION)
extract2 <- sub("ACT \\d{4}", "", extract)
extract3 <- trimws(extract2)

suburb_Cam <- cbind(speed_locations,
extract3)
```

6# Filter out non ACT data and rename column

```
suburb_Cam2 <-
suburb_Cam[str_detect(extract3,
"\\s+\\S+")==FALSE,]

colnames(suburb_Cam2)[21] <-
"SUBURB_LOCATION"
```

7# Tidy up data

```
unique(crash$SUBURB_LOCATION)

sub_names <-
crash$SUBURB_LOCATION
sub_names1 <- gsub("RURAL - ", "",
sub_names)

sub_names2 <- gsub("PADDYS  RIVER",
"PADDYS RIVER", sub_names1)

sub_names3 <- gsub("O\"CONNOR",
"CONNOR", sub_names2)

sub_names4 <- gsub("O\"MALLEY",
"MALLEY", sub_names3)
```

8# Further tidy steps to combine ‘CANBERRA’ objects.

```
table(sub_names5)

sub_names5 <- gsub("CANBERRA
AIRPORT", "Canberra", sub_names4)

sub_names6 <- gsub("CANBERRA
CENTRAL", "Canberra", sub_names5)

sub_names7 <- gsub("CAPITAL HILL",
"Canberra", sub_names6)

sub_names8 <- gsub("CITY", "Canberra",
sub_names7)
```

9# Format the suburb to match the data to be merged (first letter uppercase, the rest lowercase)

```
crash$SUBURB_LOCATION <-
paste(toupper(substr(sub_names8, 1, 1)),
tolower(substr(sub_names8, 2,
nchar(sub_names8))), sep = "")
```

10# The format of the crash data date is converted to year and filters any data after 2018 out of the dataset.

```
crash$Day <-
day(dmy(crash$CRASH_DATE))
crash$Month <-
month(dmy(crash$CRASH_DATE))
crash$Year <-
year(dmy(crash$CRASH_DATE))

crash <- crash %>%
  filter(Year == c(2016, 2017, 2018))
```

11# Extracting the variables of interest to be merged.

```
number_of_speed_cameras <- suburb_Cam2
%>%
  select(SUBURB_LOCATION,
Number.Checked, Highest.Speed,
Average.Speed, Posted.Speed) %>%
  filter(Highest.Speed > 0 | Average.Speed >
0) %>%
  group_by(SUBURB_LOCATION) %>%
  mutate(number_of_cameras = n()) %>%
  mutate(cars_checked =
sum(Number.Checked)) %>%
  mutate(avg.speed = mean(Average.Speed))
%>%
  mutate(max.speed = max(Highest.Speed))
%>%
  select(SUBURB_LOCATION,
cars_checked, number_of_cameras,
max.speed, avg.speed, Posted.Speed) %>%
  distinct(SUBURB_LOCATION, .keep_all
= T)

number_of_crashes <- crash %>%
  select(SUBURB_LOCATION,
CRASH_SEVERITY,
ROAD_CONDITION) %>%
  group_by(SUBURB_LOCATION) %>%
  mutate(number_of_crashes = n()) %>%
  distinct(SUBURB_LOCATION, .keep_all
= T)
```

12# Full join the datasets

```
Camera.accidents <-
full_join(number_of_speed_cameras,
number_of_crashes, by =
"SUBURB_LOCATION")
```

13# Remove blank row and do further mutate() functions

```
Camera.accidents <- Camera.accidents[-96,]

Camera.accidents <- Camera.accidents
%>%
  mutate(scans_per_cam =
(cars_checked/number_of_cameras)) %>%
  mutate(speed_above_limit = (max.speed -
Posted.Speed))
```

14# Correlation between the variables measured using cor()

```
cor(Camera.accidents$number_of_crashes,
Camera.accidents$number_of_cameras,
method = "pearson", use = "complete.obs")

cor(Camera.accidents$number_of_crashes,
Camera.accidents$max.speed, method =
"pearson", use = "complete.obs")
```

15# Using linear regression to impute missing datasets

```
model <- lm(number_of_crashes ~
number_of_cameras + max.speed, data =
Camera.accidents)

I <-
is.na(Camera.accidents$number_of_crashes)
impute_data <- Camera.accidents[I,]

impute_data <-
impute_data[!duplicated(impute_data), ]

imputed_values <- predict(model, newdata =
impute_data)

Camera.accidents$number_of_crashes[I] <-
imputed_values
```

16# Using nearest neighbour to impute missing data

```
Camera.accidents <-
kNN(Camera.accidents) %>%
  select(1:11)
```

17# Selecting all the fatalities

```
crash %>%
  filter(CRASH_SEVERITY == "Fatal")
%>%
  group_by(SUBURB_LOCATION) %>%
  summarise(fatalities = n()) %>%
  arrange(desc(fatalities))
```

18# Gowers similarity metric is performed here

```
Camera.accidents$CRASH_SEVERITY <- as.factor(Camera.accidents$CRASH_SEVERITY)

Camera.accidents$ROAD_CONDITION <- as.factor(Camera.accidents$ROAD_CONDITION)
Camera.accidents$SUBURB_LOCATION <- as.factor(Camera.accidents$SUBURB_LOCATION)

gow_cam <-  1 - daisy(Camera.accidents, metric = "gower")
matrix_gow <- as.matrix(gow_cam)

tuggeranong <- matrix_gow[116,]
```

19# A function was created to retrieve the top 5 matches for the suburb Tuggeranong

```
get_top_5 <- function(vector) {

  sorted_vector <- sort(vector, decreasing = TRUE)

  top_5 <- head(sorted_vector, 5)

  return(top_5)
}

get_top_5(tuggeranong)
```

20# A graph using the ggplot() showing the number of speed cameras

```
filt2 <- speed_locations %>%
 filter(Highest.Speed > 0 | Average.Speed > 10) %>%
  mutate(speed_over_limit = Highest.Speed - Posted.Speed) %>%
 ggplot(aes(x = Number.Checked, y = speed_over_limit, colour = Year))+
 facet_wrap(~Year)+
 geom_point() +
 geom_smooth(method = "lm") +
 labs (y= "Speed_over_the_limit (Km/H)", x = "Number.of.cars.checked", title = "A scatter graph showing the relationship between the number of cars detected
       over the limit against max speed of cars above the limit for that given day")
```

21# A boxplot using the ggplot() showing the distribution of max speed over the limit

```
 speed_locations %>%
 filter(Highest.Speed > 0 | Average.Speed > 10) %>%
 mutate(speed_over_limit = Highest.Speed - Posted.Speed) %>%
 ggplot(aes(x = speed_over_limit, colour = Year))+
   geom_boxplot()+
   labs (x = "Speed_over_the_limit (Km/H)", title = "A Box plot  showing the distribution of the
        max speed of cars above the limit for that given day")
```