1 INFO1903 Project

1.1 Section I: Data analysis

1.1.1 Situation {#situation}

For this project, I wanted to analyse state rainfall data alongside road fatalities, with the primary aim of finding a correlation between the two.

1.1.2 Data Sources {#sources}

Online Sources

- http://data.gov.au/dataset/australian-road-deaths-database/resource/ca07c8e3-672f-4826-a6e5-83fd7127ae0bThe Australian Road Deaths Database) which contains information about the crashes and the fatalities.

- http://www.bom.gov.au/jsp/ncc/cdio/weatherData/av?p$_n ccObsCode = 136\&p_d isplay_t ype = dailyDataFile\&p_s tartYear = \&p_c = \&p_s tn_n um = 086039TheBureauofMeteorology'sDailyRainfallDat$

Download Links: {#downloads}

- https://bitre.gov.au/statistics/safety/files/Fatal$_C rashes_F eb2017.csvCrashdatabase(csv)https : //bi$

- http://www.bom.gov.au/jsp/ncc/cdio/weatherData/av?p$_d isplay_t ype = dailyZippedDataFile\&p_s tn_n um =$ 086039$\&p_c = -1480557288\&p_n ccObsCode = 136\&p_s tartYear = 2017Raindatabase(csv)$

1.1.3 Graphs {#graphs}

After setting up the data, I first graphed the rainfall over time and the crashes over time to see if I could spot any trends among the separate graphs:

Car Crashes per Month assets/crashes$_o ver_t ime.pngassets/crashes_o ver_t ime.png$

Monthly Rainfall assets/rainfall$_o ver_t ime.pngassets/rainfall_o ver_t ime.png$

Crashes and Rainfall over Time After having graphed the data separately, I graphed them on top of each other to get a better idea at a correlation:
assets/rainfall$_v s_d eaths.pngassets/rainfall_v s_d eaths.png$

This final graph is the same data but in a different style: the data points are the
fatalities per month, but the rainfall is instead represented as the colour of the points.

I thought that this would help visualise the correlation, but it doesn't really work.

assets/fatalities$_v s_d ate.pngassets/fatalities_v s_d ate.png$

1.1.4 Discussion {#discussion}

Trends {#trends} The crashes/time graph shows that the average number of fatal car crashes per month has
decreased since 1989, something which I expected to see.

assets/crashes$_o ver_t ime_z oom.pngassets/crashes_o ver_t ime_z oom.png$
$The spike at the start is because of the Kempsey Bus Crash, cited as the most deadly road accident in Australia's history.$
$(https://en.wikipedia.org/wiki/Kempsey_b us_c rash Wikipedia Article)$

In the rainfall/time graph, there is an outlier around 2005 - a heavy rain event.
(https://bom.gov.au/climate/annual$_s um/2005/page 13 - 15.pdf BOM report$)

Results {#results}

Further Research {#further-research}

1.2 Section II: Data Generation {#generation}

1.2.1 Getting the data {#obtaining}

The website had two datasets available: one for each crash, and one for each fatality.
I chose to use the one per crash, as I was not interested in statistics such as gender
or age. However, the process required to obtain and store this data is available with the
methods for the other files.

Due to the nature of the weather, I decided that getting the "total national rainfall"
was not precise enough. Because the crash data sorted by state (and did not give a

precise location), I decided to pick a state and use only crash data from that state.

The BOM data provides rainfall data for every weather station back to the 19th century.
As my crash data location had state-level precision, I reasoned that if I chose a weather
station in the middle of a state, it would give me the best approximate for "average
statewide weather".

I chose Victoria, as it is a small state with a weather station (Flemington station)
somewhat near both the center of the state and the capital city, where I reasoned the
most crashes would occur.

1.2.2 Storing in PostgreSQL {#postgres}

Entering into database

Database Schema

`crashes` **Table**`crashes Table` []@lll@ Column Type Description`crashid character(13)` Internal crash ID`state character varying(3)` State that the crash occured in`day integer` Day of the crash`month character varying(10)` Month of the crash (long name format)`year integer` Year of the crash`hour integer` Hour of the crash`minute integer` Minute of the crash`crashtype character` varying(16) Internal type of the crash`fatalities integer` Number of fatalities`bus boolean` Was a bus involved?`heavytruck boolean` Was a heavy truck involved?`articulatedtruck boolean` Was an articulated truck involved?`speedlimit integer` The speed limit of the crash

`rainfall` **Table**`rainfall Table` []@lll@ Column Type Description`year integer` Year of the measurement`month integer` Month of the measurement (in integer form)`day integer` Day of the measurement`rainfall double precision` Amount of rainfall`period integer` Period measured`quality character(1)` Quality of data

Issues

Date Formatting Looking at the above schema, you might notice: the `month` field of the rainfall table is
an `integer` type, but `crashes.month` is a `varchar(10)`. `crashes.month` is a long month
name, e.g. `January`, `February`.

This was a problem. I had two different formats of data that were needed to do an
SQL JOIN. Luckily, PostgreSQL has a very good set of date formatting commands.

I decided to leave the tables as they were, and convert the data on the fly when doing the
SQL JOIN. The following SQL functions will convert a date in long format to an integer:

```
extract(MONTH from to_date(concat(crashes.month, ' 2000'), 'Month YYYY'))
```

1.2.3 Querying {#querying}

Issues

1.2.4 Graphing {#graphing}

Issues

1.2.5 Notebook {#notebook}

https://nbviewer.jupyter.org/github/lyneca/info1903/blob/gh-pages/INFO1903.ipynbHere is the Jupyter Notebook that contains code for querying and visualising the data.