Data Analytics II Scenario Week Report

Cycle

Your

Way

Team I:

Evgenii Osinin	Giulia Zhang	Hoijin Kim	Jack Lim	Mihnea Macovei	Paul Ferri	Tania Turdean
19064610	**19112842**	**19081830**	**19076676**	**19030283**	**19110673**	**19004997**

Word Count: 1993

Table of Contents

1. Describe your idea

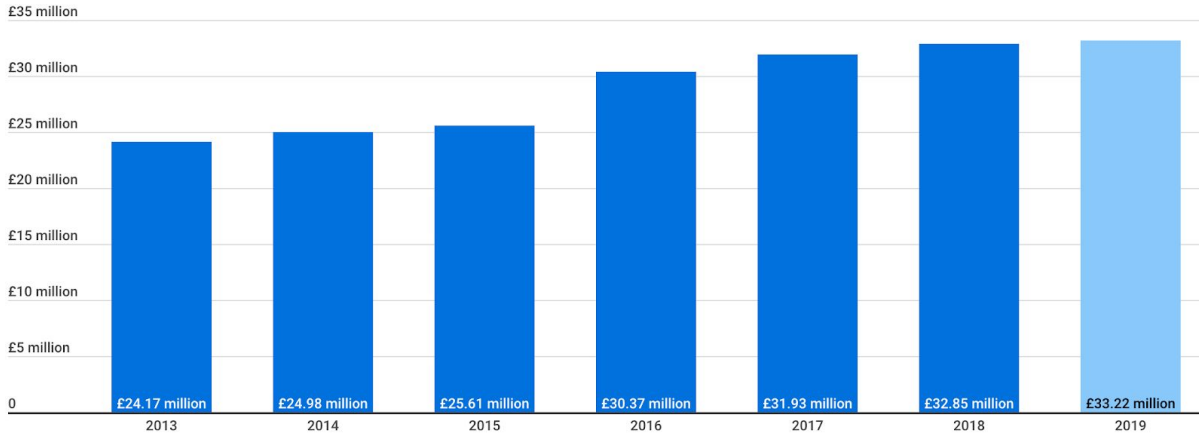
1.1. Start-up Summary

CycleYourWay is a cycling map-app based on crowdsourced information, whose main purpose is to encourage cycling and improve cycling experience. It also allows users to share real-time data to provide accurate information to other users. Its core feature is informing users about the safety conditions of different areas in London.

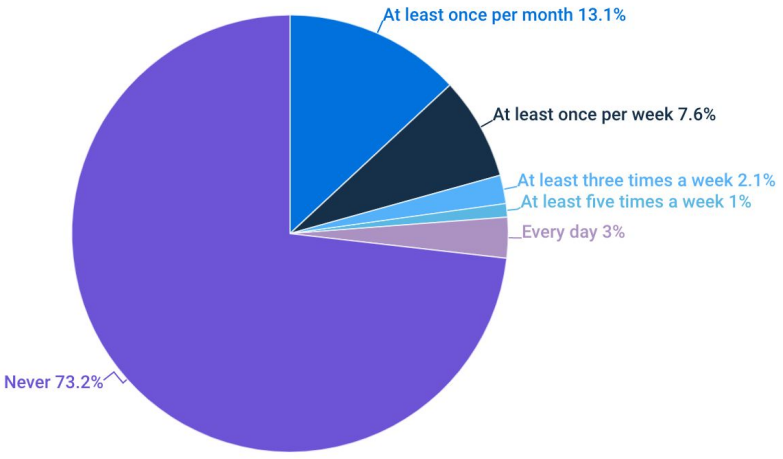
1.1.1. Cycling statistics

Yearly spend

The graph below shows a steady 1-3% increase in bicycles spent. (1)



How often do British people cycle?



The graph above illustrates that only 3% of British cycles every day and shows the potential of our business idea as a significant market (73.2% never cycled) is undiscovered. (1)

1.2. Lean Model Canvas

1.2.1. Objectives

We are committed to becoming London's best cycling app with environmental considerations placed at the core of our business' principles: we are focused on encouraging cycling to contribute to climate change. Socially, starting from London, we aim to improve people's general health and reduce ecological footprint.

We decided to start our journey from London and then expand to the rest of the UK for two main reasons:

1. London being the worst city in Europe for health costs from air pollution. (2)

Existing alternatives

Traffic congestion presents significant environmental, social, and economic costs. Therefore, encouraging people to cycle is an important aspect of addressing this problem. Currently, the market offers various mobile apps to improve the cycling experience (3):

STRAVA for tracking your activity

Cycle streets for route planning

First Aid for cyclists for first aid

2. London having the highest casualty rates by traffic. (see section 3)

Existing alternatives

The Transport for London (Tfl) is committed to make cycling in London safer by coming up with (4):

- A Cycle Safety Action Plan
- The Direct Vision standard

(see Appendix 6.1.)

1.2.2. Lean Canvas

Lean Canvas

Designed for: CycleYourWay

Designed by: Team I

Problem	Solution	Unique Value Proposition	Unfair Advantage	Customer Segments
Concern over safety is the greatest barrier to people cycling in London (61% of people think cycling is dangerous)	Provide users with safety routes to their destination Provide users with real time alerts such as weather forecast and key parts of the road that is dangerous	Create positive environmental change by reducing carbon footprint and improve general health of people in England Give its users the ability to co-create value by allowing then to share inputs and improve data's accuracy	Support from UK government emergency travel fund of £250million to improve bike lanes and cycle-only junctions Partner with weather forecast applications which differentiates this app with others	London: -cyclist for work -cyclist for school -cyclist for exercise -cyclist for shopping -cyclist for sight-seeing
Existing Alternatives Cycling apps: Strava Cycle streets TfL initiatives:	**Key Metrics** • The number of active users • The number of downloads of the app • The average time spent on the app • The frequency of users inputs	**High-Level Concept** Cycling app for accurate safety condition forecasts	**Channels** Touchpoints: • Before purchase: Social media platforms, bike shops, sports retail shops • While purchase: Mobile App store and website • After purchase: Email and text updates	**Early Adopters** • young cyclists • inexperienced cyclists • tourists

Cost Structure (estimated)

fixed costs:

-navigation app £40,000
-discovery stage £5000
-production stage £5000- £10000
-Testing and deployment £5000

Total App cost = £55,000 – £60,000

Variable costs:

Maintaining and optimising of App:
- 50% of total app cost

- The creation of the app will be outsourced to professional teams where the current team I will run the operation.

Revenue Structure

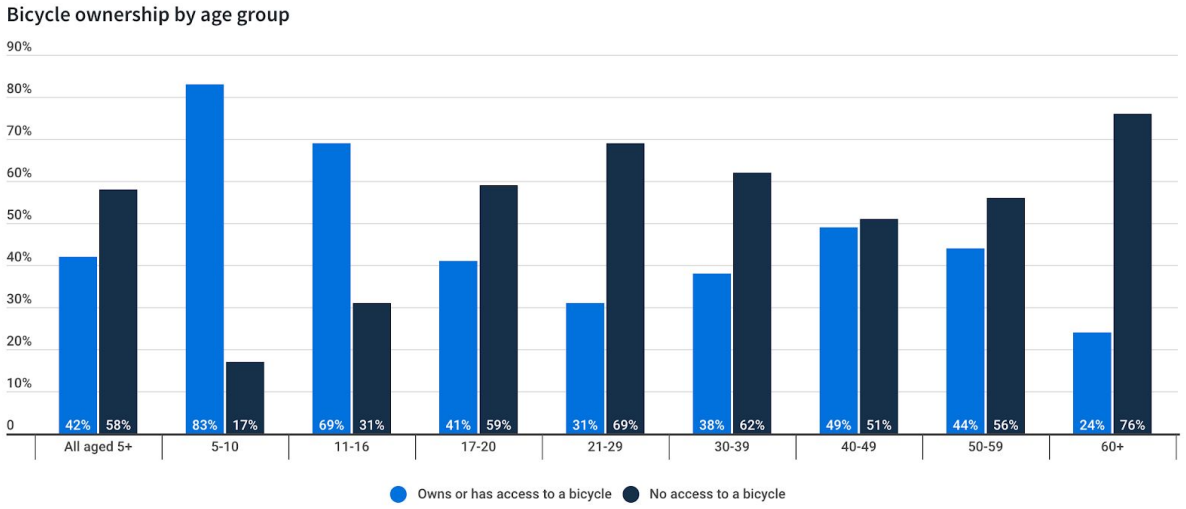
Start-up cost £ I million (coming from investors such as Angel fund and private ventures)

Revenue methods: Mainly from branded pin advertisement on Map and other in-app banner/video advertisements

Revenue in the case of having 80,000 daily active users: £576,000 (£0.02 generated by every click of completion of ads)
Revenue from Food and Beverage branded pin = £100,000

Early Adopters

Tourists, young and inexperienced cyclists are the first in line to use the app as they are most likely to be concerned with the area's safety. As it is possible to see on the graph below, younger members of the population have the highest bicycle ownership.



Unfair advantage

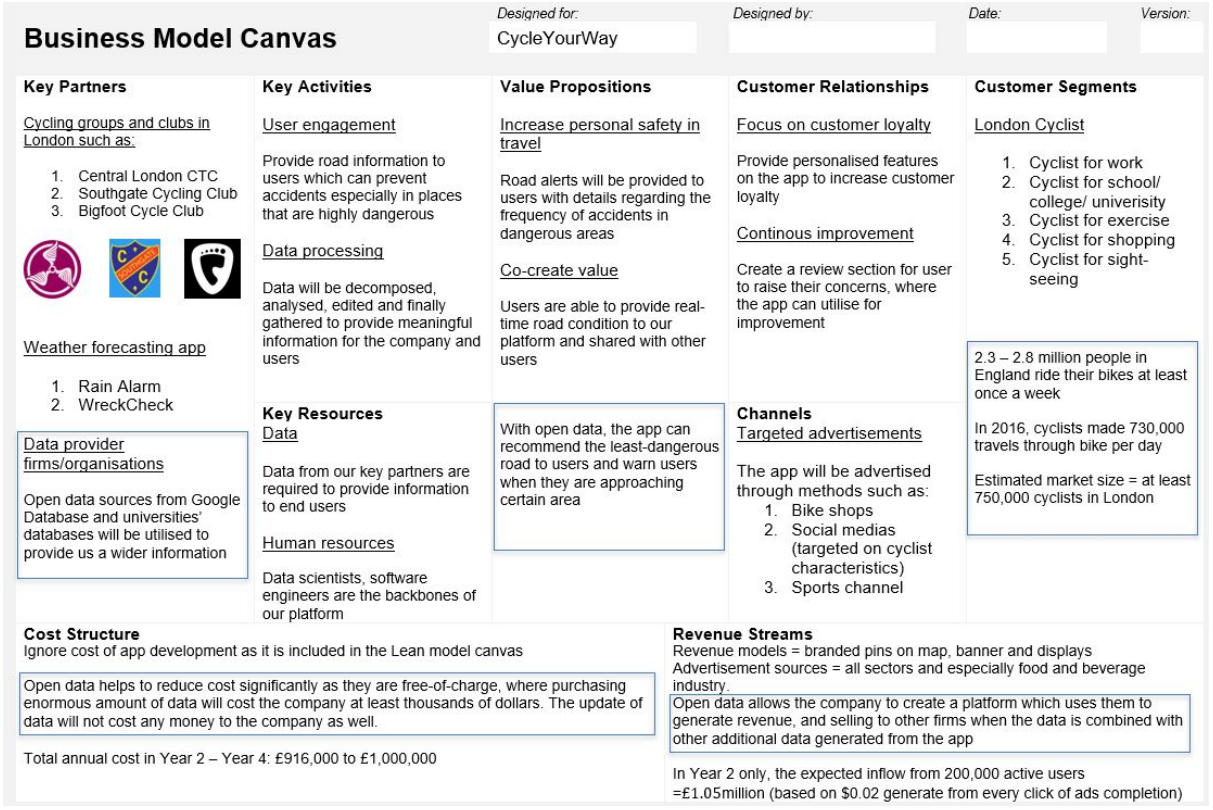
During the Covid-19 pandemic, cycling became more popular in many cities, especially London. Hence, the demand for new and second-hand bikes skyrocketed (5), giving huge confidence to our business idea as cycling is expected to expand in England and turn into a norm for more people. Besides, the UK government announced an emergency travel fund of £250 million to set up pop-up bike lanes, safer junctions, and cycle-only corridors. (6) With this support, improved facilities will contribute to protecting cyclists, continuously convincing people to ride (61% of people in England thought cycling is dangerous in a pre-lockdown survey). (7)

Additionally, our analysis is based on multidimensional data (6D) which ensures the accuracy of our forecasts.

1.3. Business Model Canvas

The lean canvas focuses on understanding how our app works and supports our business ideas. The cost and revenue structures are restricted to the first-year operation as the app development cost is a one-off capital expenditure. Advertisements in the app are estimated to generate nearly £700,000 revenue in the first year with a basis of 80,000 daily active users.

A Business Model Canvas was conducted to provide a wider picture of the operation in the next 2-4 years. The revenue and cost structures are estimated with a larger basis of active users.



The table below shows CycleYourWay's user target plan. It aims to double its users every year as start-ups like Waze experienced a significant rise in users in the early years. (8)

Year	Optimal target of active users	% of increase in active users	Optimal revenue in million pounds	Worst scenario target of active users (bottom line)
2	200,000	100%	1.05	150,000
3	500,000	150%	2.78	300,000
4	1,000,000	100%	5.50	700,000
5	2,000,000	100%	11	1,200,000

The cost structure below illustrates the cost associated with each component and the annual cost (£916,000). In year 2, the business should generate £100,000 or at least break-even. (9)

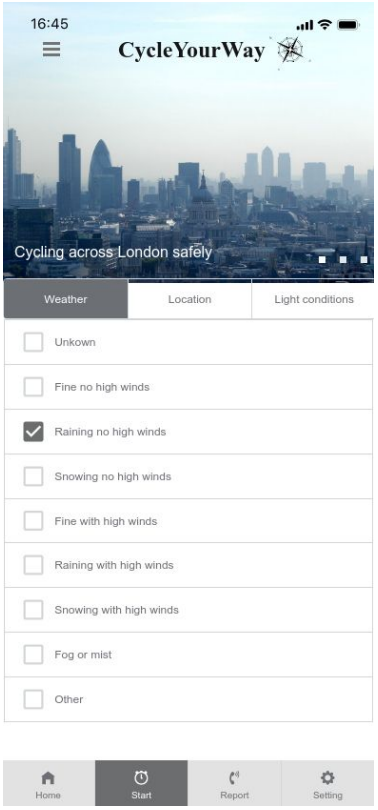
Cost component	Annual Amount (in pounds)
Administrative salaries (7 group mates)	336,000
Marketing budget	500,000
Maintenance and upgrade of app	30,000
General Expenses (rent, water, electricity)	50,000

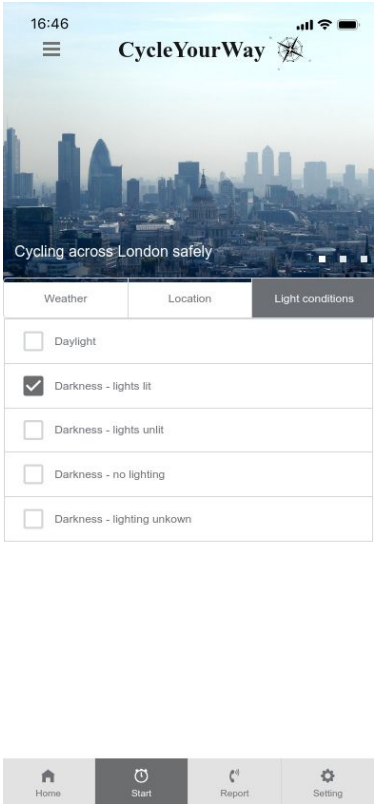
Key Partners

1.4. Mock ups

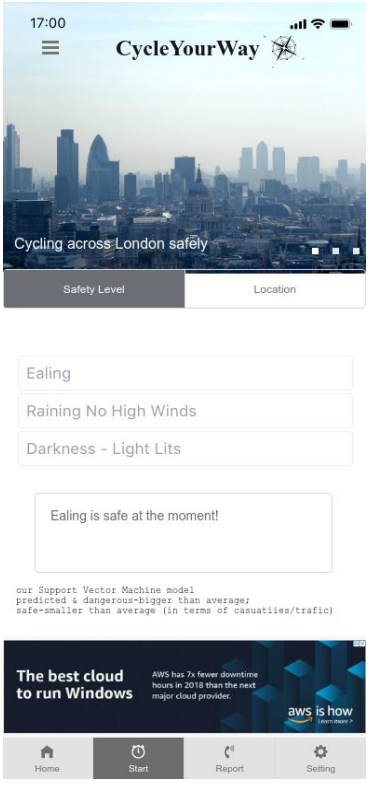
The mock-ups describe how the app works.

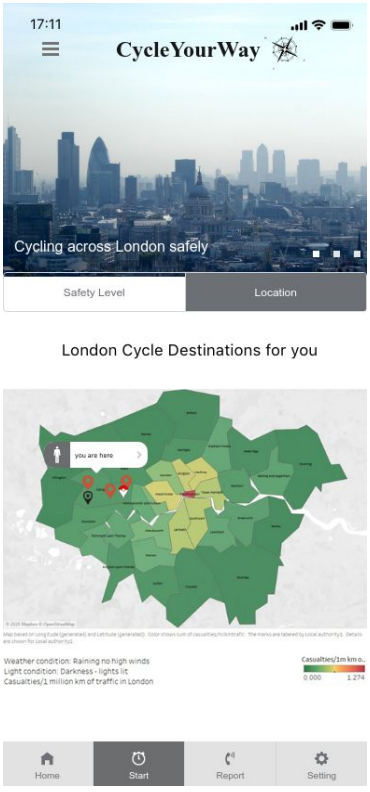
Input pages:

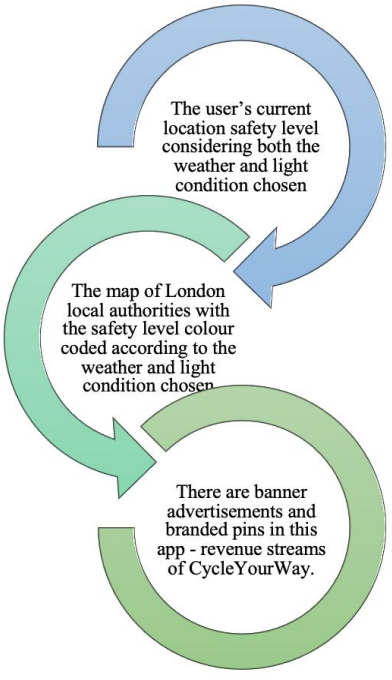




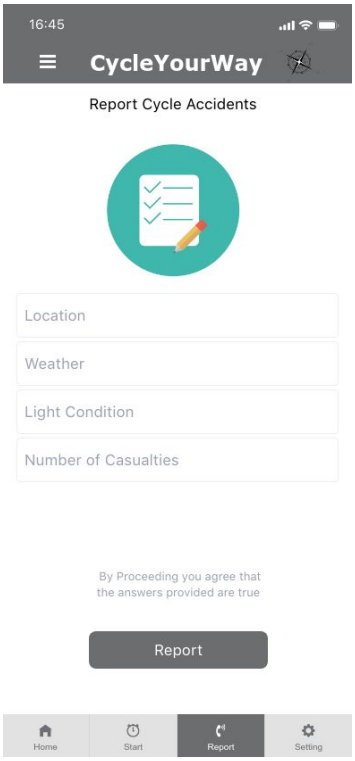
Output pages:

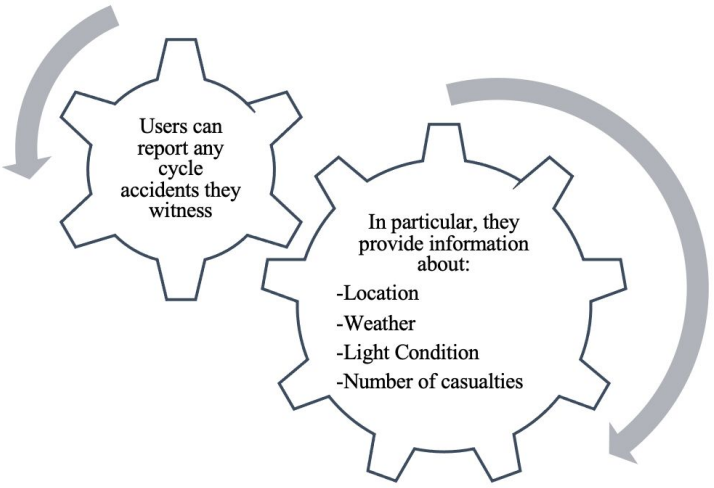




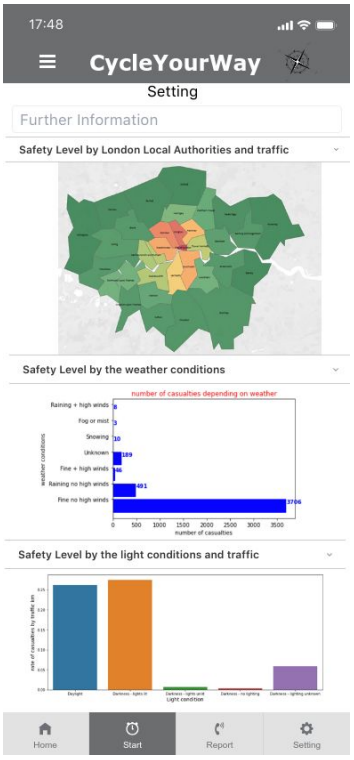


Report Tab:





Further Information:



CycleYourWay provides further information on safety level

Specifically, by three variables:
-London Local Authorities
-Weather Conditions
-Light Conditions

2. Open Data you have Investigated

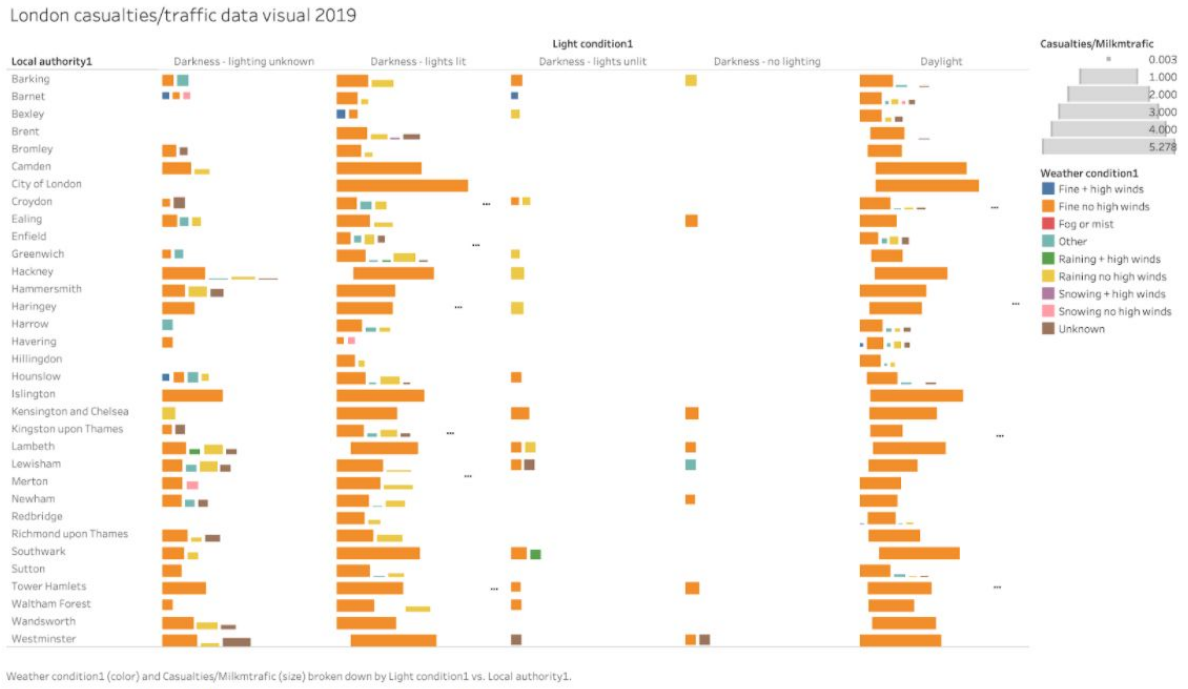
2.1. Data Sources considered

NR	DATA	CONTENT	SOURCE	DATE	SIZE (rows x columns x sheets)	FORMAT
1	TRA0413	Traffic data pedal cycle UK regions & sub-regions -in vehicle km	Department of transport statistics	1993-2019	28 x 28	ods
2	Personalised report	Nr of road accident casualties by UK regions for different road users	Department of transport statistics	2019	89 x 5	ods
3	ras30040	Road casualties/bil km traffic – UK regions and local authorities	Department of transport statistics	2009-2019	222 x 13	ods
4	Personalised report	Casualties by Road user type -in UK	Department of transport statistics	2010-2019	81 x 3	ods
5	Personalised report	Nr of casualties- by local authority (London), road user type, light condition, weather condition	Department of transport statistics	2018-2019	4414 x 6	ods
6	Tra8904	Traffic in km of motor vehicle – local authorities (UK)	Department of transport statistics	1993-2019	233 x 29	ods
7	tra0307	Traffic distribution by time of the day & day of the week UK	Department of transport statistics	20006-2019	25x8 x 14 (years)	ods
8	ras30011	The number of road accident casualties by different road users and age	Department for Transport	2005-2013	66x11x3 (gender)	xls
9	ras30012	The number of road accident casualties by severity and by time of the day (in 2-hour intervals)	Department for Transport	2003-2013	54 x 12	xls
10.	ras10014	The number of road accidents by region, local authority and road class	Department for Transport	2010-2013	8 x 163 x 4	xls
11.	STATS19	Reported road casualties by local authority, year and age. (In the UK)	Road Safety Statistics	2015-2019	5 x 65363	ods

The table describes the content, source, date, size, and format of the data considered. The purpose is explained throughout the report. (the rest of the data considered in Appendix 6.4.)

2.2. Exploring data sources with visualisations

Visual 1



Data prepared in Python, represented in Tableau

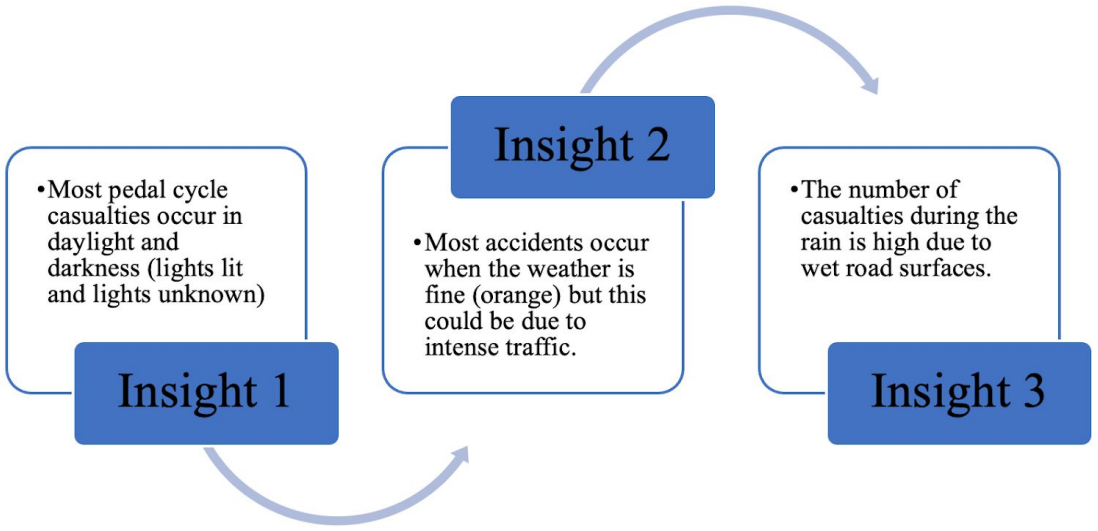
This heatmap represents 6-Dimensional data of the pedal cycle casualties:

Traffic in London local authorities
Traffic by daylight and darkness
Number of casualties
Weather conditions
Light conditions
Local Authority

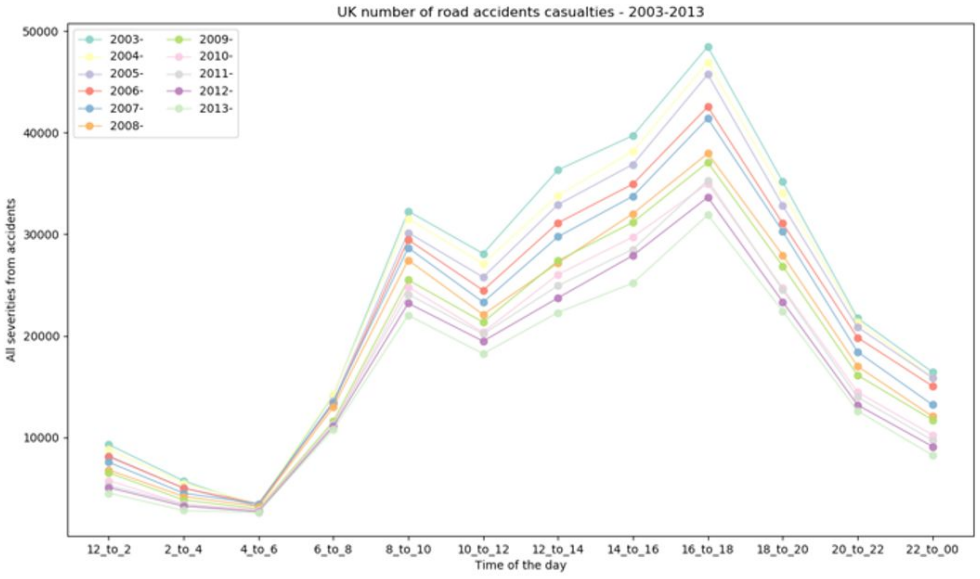
Three datasets (5, 6 and 7) from the table (section 2.1.) have been merged to create this visual.

The size of the points	Cycle casualty by Local authorities, Weather and Light conditions. Casualties were divided by traffic in: - London local authorities - Daylight and Darkness in 2019
Colours	Coloured according to 9 weather conditions
Rows	Each London local authority
Columns	5 different light conditions

This visual shows the casualty rate by traffic compared to 3 different variables. For example, the casualty rate by traffic in Barnet according to each specified weather condition and the light condition is shown. In Barnet, with the light condition darkness-lights unlit, most of the casualties occur when there are high winds.



Visual 2

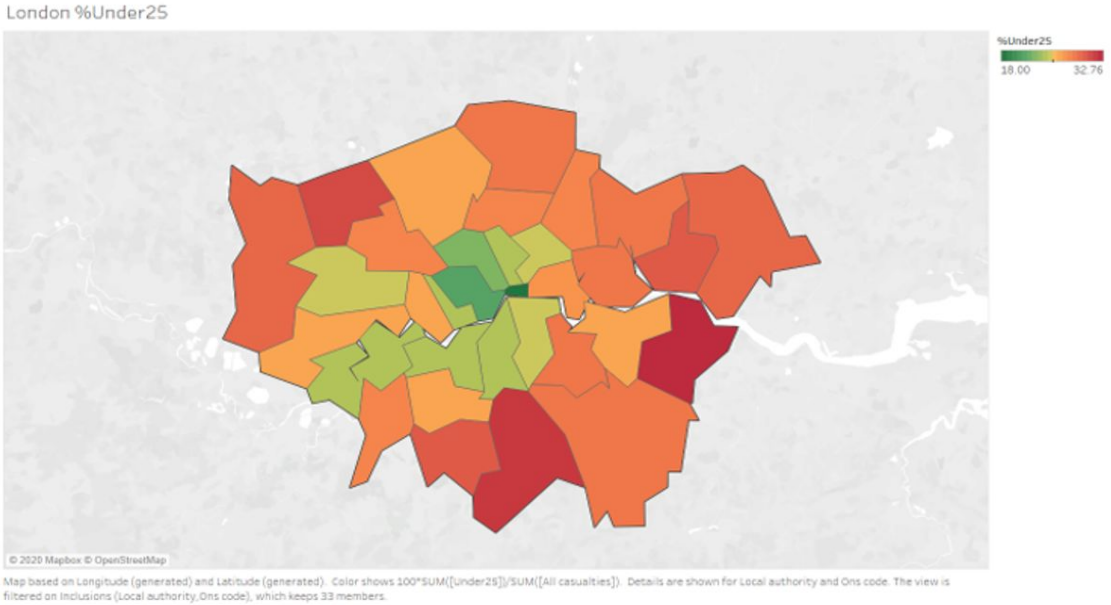


The dataset 9 (section 2.1.) used to plot this graph in Python

X-axis	Time of the day in 2-hour interval
Y-axis	The number of car accident casualties

Most accidents occur between 16-18. This could be due to higher traffic at that time. Each colour represents a year. The number of accidents tends to decrease over the years.

Visual 3



a local map of London made using Tableau and the dataset 11 (section 2.1.)

Every accident from 2015 to 2019 was used. The map illustrates whether young people are more involved in road accidents.

In most local authorities, people under 25 are involved in approximately 30% of the accidents, higher than the average age of driving people.

Therefore, people under 25 are more likely to have accidents in London.

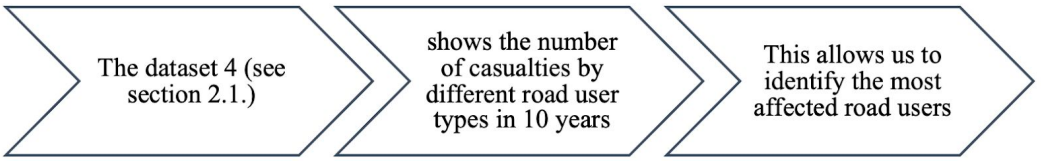
This could be due to less road experiences.

See Appendix 6.2 for more.

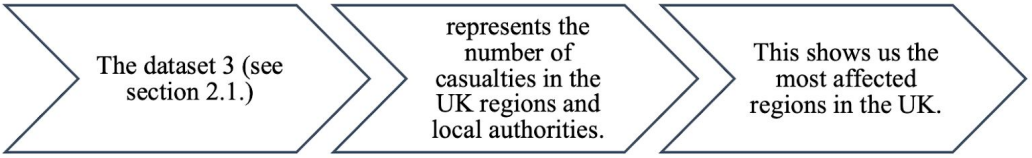
2.3. Data sources selected for CycleYourWay

Two datasets have been selected to decide our target location and users:

- **Data set 4**



- **Date set 3**



The visuals are presented in section 3.

For our app we have used three datasets which are showcased in visual 1 (section 2.2). Among all the datasets, we have selected dataset 5 (section 2.1) as the others only provide one variable. Dataset 5 provides the casualty rate according to 3 different variables:

1. Location
2. Weather conditions
3. Light conditions

3. How you have used Open Data

3.1. Target location

We created a map in Tableau to show the rates of vehicle casualties per billion kilometres of traffic for different UK regions.

Map based on Longitude (generated) and Latitude (generated). Color shows sum of casualties/1bilkmTraffic. The marks are labeled by Region. Details are shown for Region.

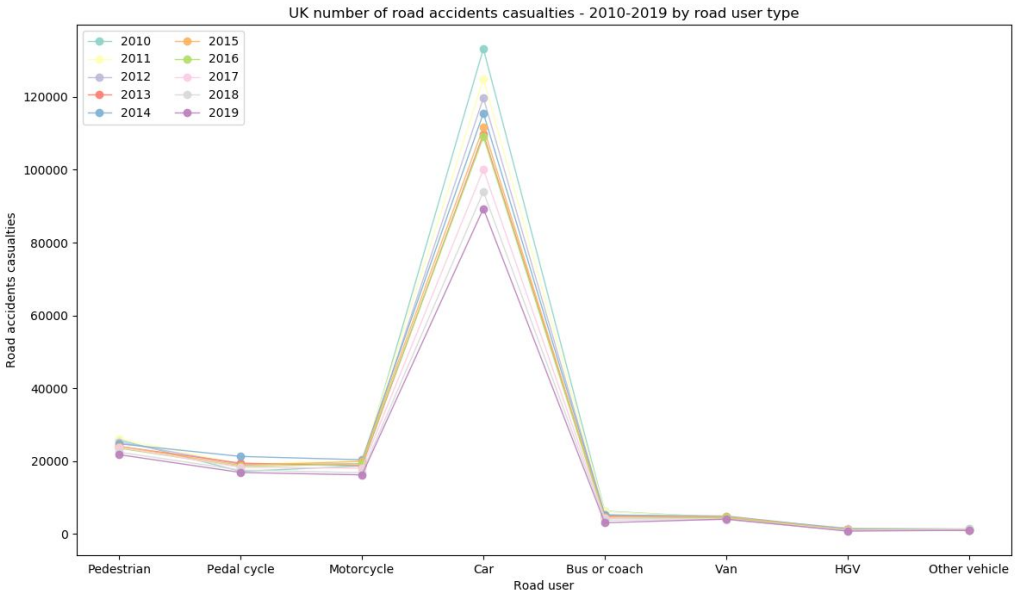
Based on dataset 3 (section 2.1.)

Originally, this map showed casualty rate figures by region and local authority. For our purpose, we deleted the local authority rows.

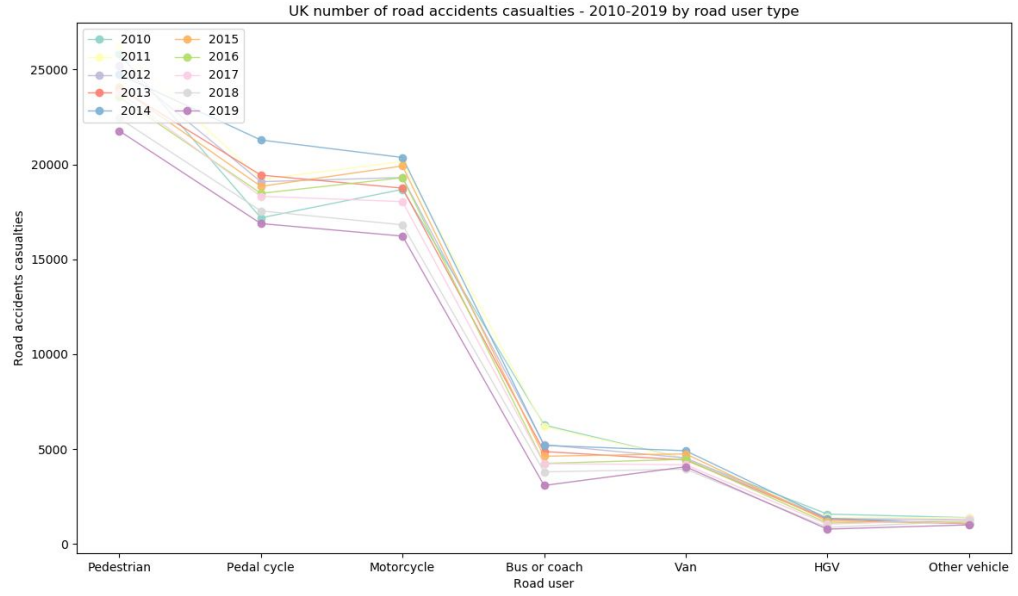
As presented on the map, London stands out with the highest casualty rate. Thus, we decided to base our business in London and then expand it to other regions if the app proves viable.

3.2. Target users

We visualised road accidents casualties by different types of road users based on dataset 4 (section 2.1.) using python. The graph below shows the yearly numbers of casualties in the UK from 2010 to 2019. Due to cars being the largest road user type, it also happens to be the vehicle with the largest number of casualties.



To investigate the difference between other road users, we eliminated 'car' road user data and generated the graph below.



For most years, pedal cyclists were the third highest casualty group. Since many apps targeted at car users exist and as environmental considerations are at the core of our business, we decided to focus on cyclists.

3.3. Open data to add value

Open data was used to decide our inputs and outputs for the app.

The open data 5 (section 2.1.) enabled us to decide which variables to consider as inputs to measure the safety for cyclists. Then it was merged with open data 6 and 7 (section 2.1.) to consider traffic in different locations and light conditions.

354 rows × 10 columns

```
In [28]: fmergedata2019['casualties/milkmtrafic']=fmergedata2019['All casualties']/fmergedata2019['Trafifcmillkm']/fmergedata201

         fmergedata2019
```

Out[28]:

	Local authority	Road user	Light condition	Weather condition	All casualties	Trafifcmillkm	traffic	casualties/milkmtrafic
0	City of London	Pedal cycle	Daylight	Fine no high winds	74	169	13398	3.268173
1	City of London	Pedal cycle	Daylight	Raining no high winds	4	169	13398	0.176658
2	City of London	Pedal cycle	Daylight	Fine + high winds	1	169	13398	0.044165
3	Barking	Pedal cycle	Daylight	Unknown	3	809	13398	0.027678
4	Barking	Pedal cycle	Daylight	Fine no high winds	36	809	13398	0.332135
...	...	...	...	...	...	...	...	...
349	Wandsworth	Pedal cycle	Darkness - lighting unknown	Fine no high winds	9	931	3251	0.297355
350	Wandsworth	Pedal cycle	Darkness - lighting unknown	Raining no high winds	4	931	3251	0.132158
351	Westminster	Pedal cycle	Darkness - lighting unknown	Unknown	7	890	3251	0.241931
352	Westminster	Pedal cycle	Darkness - lighting unknown	Fine no high winds	11	890	3251	0.380177
353	Westminster	Pedal cycle	Darkness - lighting unknown	Raining no high winds	3	890	3251	0.103685

354 rows × 8 columns

Looking at the merged dataset, we decided to calculate casualties/traffic to make sure the number of casualties reflects the volume of traffic to improve the accuracy and added the result as an extra column. This helped us to gain ideas about our outputs, informing the users whether it's safe to cycle at the moment by looking at the values.

```
In [40]: mergedata2["casualties/milkmtrafic2"].loc[mergedata2["casualties/milkmtrafic2"] != "Dangerous"]="Safe"
```

```
In [41]: mergedata2
```

Out[41]:

:cident year	Local authority	Road user	Light condition	Weather condition	All casualties	Trafficmillkm	traffic	casualties/milkmtrafic	Local authority1	Light condition1	Weather condition1	casualties/milkmtrafic2
2018	6	Pedal cycle	4	8	1	167	13534	0.044244	City of London	Daylight	Unknown	Safe
2018	6	Pedal cycle	4	1	71	167	13534	3.141346	City of London	Daylight	Fine no high winds	Dangerous
2018	6	Pedal cycle	4	5	5	167	13534	0.221222	City of London	Daylight	Raining no high winds	Safe
2018	0	Pedal cycle	4	1	28	798	13534	0.259256	Barking	Daylight	Fine no high winds	Dangerous
2018	0	Pedal cycle	4	5	2	798	13534	0.018518	Barking	Daylight	Raining no high winds	Safe
...	...	...	...	...	...	...	...	...	...	...	...	...
2018	26	Pedal cycle	3	5	1	878	3264	0.034894	Richmond upon Thames	Darkness - no lighting	Raining no high winds	Safe
2018	26	Pedal cycle	3	4	1	878	3264	0.034894	Richmond upon Thames	Darkness - no lighting	Raining + high winds	Safe
2018	29	Pedal cycle	3	1	1	1003	3264	0.030546	Tower Hamlets	Darkness - no lighting	Fine no high winds	Safe
2018	30	Pedal cycle	3	3	1	979	3264	0.031294	Waltham Forest	Darkness - no lighting	Other	Safe
2018	32	Pedal cycle	3	1	1	906	3264	0.033816	Westminster	Darkness - no lighting	Fine no high winds	Safe

s × 13 columns

It was then categorised into either safe or dangerous depending on whether the value is below or above the average (one of the outputs shown to users).

```
z=pd.DataFrame(z)
z.reset_index(drop=True, inplace=True)
z1=z["Local authority"][0]

test_point2= np.array([[x1,y1,z1]])


#OUTPUTS

# 1. Making a prediction
clf_trained.predict(test_point2)
```

```
Tell me the light conditions at this moment -  Darkness - lights lit
Tell me the weather conditions at this moment -  Raining no high winds

Show me where are you on the map - Ealing
```

```
Tell me the light conditions at this moment -  Darkness - lights lit
Tell me the weather conditions at this moment -  Raining no high winds
Show me where are you on the map - Ealing
```

Out[52]:
```
array(['Safe'], dtype=object)
```

The variables chosen are asked to users and when they insert their answers, the output, either safe or dangerous is shown (we use a *Support Vector Machine* to predict this).

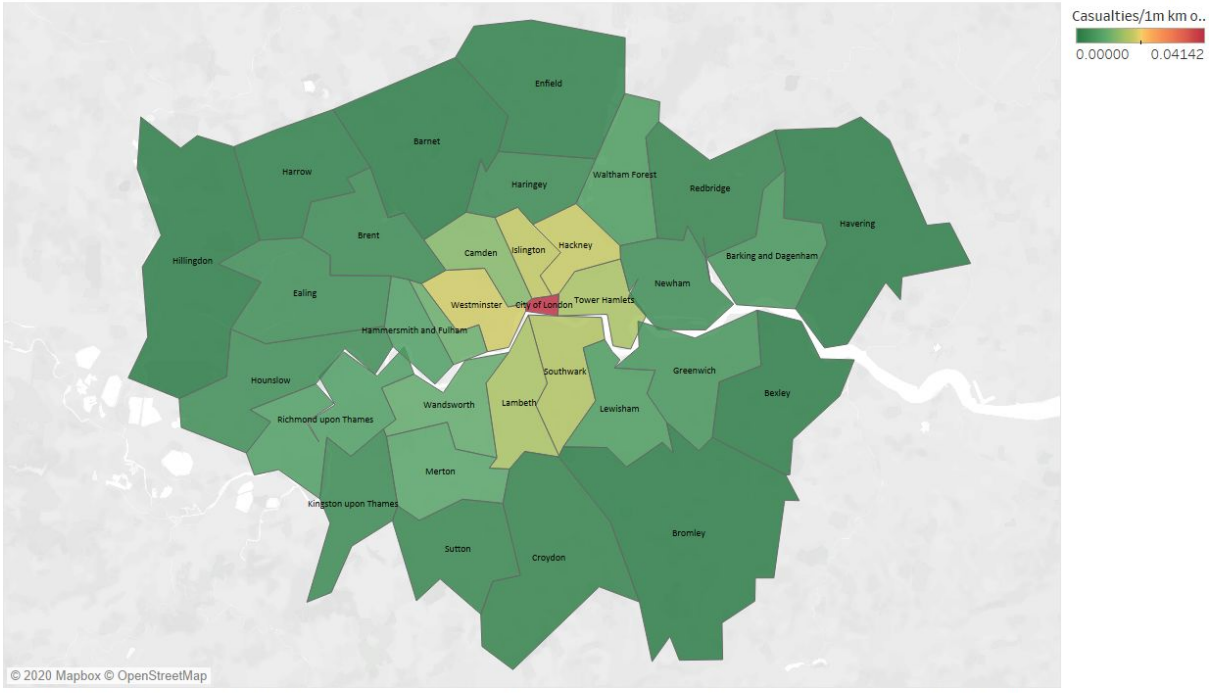
[53]:
```
#2. Describe

b=fmergedata2019.loc[fmergedata2019["Light condition"]==x1]
c=b.loc[b["Weather condition"]==y1]
c
```

t[53]:

	Local authority	Road user	Light condition	Weather condition	All casualties	Trafifcmillkm	traffic	casualties/milkmtrafic	Local authority1	Light condition1	Weather condition1	casualties/milkmtrafic2
151	6	Pedal cycle	1	5	7	169	3251	1.274073	City of London	Darkness - lights lit	Raining no high winds	Dangerous
153	0	Pedal cycle	1	5	4	809	3251	0.152088	Barking	Darkness - lights lit	Raining no high winds	Safe
155	1	Pedal cycle	1	5	1	2089	3251	0.014725	Barnet	Darkness - lights lit	Raining no high winds	Safe
160	3	Pedal cycle	1	5	3	1098	3251	0.084043	Brent	Darkness - lights lit	Raining no high winds	Safe
163	4	Pedal cycle	1	5	1	1555	3251	0.019781	Bromley	Darkness - lights lit	Raining no high winds	Safe
166	5	Pedal cycle	1	5	6	478	3251	0.386106	Camden	Darkness - lights lit	Raining no high winds	Dangerous
169	7	Pedal cycle	1	5	2	1507	3251	0.040823	Croydon	Darkness - lights lit	Raining no high winds	Safe
172	8	Pedal cycle	1	5	5	1425	3251	0.107929	Ealing	Darkness - lights lit	Raining no high winds	Safe
175	9	Pedal cycle	1	5	2	2046	3251	0.030068	Enfield	Darkness - lights lit	Raining no high winds	Safe

This last column allowed us to come up with another output but this time, only considering the weather and the light condition as our inputs.

Weather condition: Raining no high winds
Light condition: Darkness - lights lit
Casualties/1m km of traffic in London



Map based on Longitude (generated) and Latitude (generated). Color shows sum of casualties/milkmtrafic. The marks are labeled by Local authority1. Details are shown for Local authority1.

This map is the result, showing different local authorities with colour codes representing the safety level with the weather and light condition chosen by the user.

See Appendix 6.4. for more.

4. Technical Considerations

4.1. Data preparation and merges for the app

2018 data

	A	B	C	D	E	F	G
1	Accident year	Local authority	Ons code	Road user	Light condition	Weather condition	All casualties
2	2018	City of London	E09000001	Pedestrian	Daylight	Fine no high winds	59
3	2018	City of London	E09000001	Pedestrian	Daylight	Raining no high winds	2
4	2018	City of London	E09000001	Pedestrian	Daylight	Snowing no high winds	1
5	2018	City of London	E09000001	Pedestrian	Darkness - lights lit	Fine no high winds	17
6	2018	City of London	E09000001	Pedestrian	Darkness - lights lit	Raining no high winds	5
7	2018	City of London	E09000001	Pedal cycle	Daylight	Unknown	1
8	2018	City of London	E09000001	Pedal cycle	Daylight	Fine no high winds	71
9	2018	City of London	E09000001	Pedal cycle	Daylight	Raining no high winds	5
10	2018	City of London	E09000001	Pedal cycle	Darkness - lights lit	Fine no high winds	21
11	2018	City of London	E09000001	Pedal cycle	Darkness - lights lit	Raining no high winds	2
12	2018	City of London	E09000001	Motorcycle	Daylight	Fine no high winds	42
13	2018	City of London	E09000001	Motorcycle	Daylight	Raining no high winds	4
14	2018	City of London	E09000001	Motorcycle	Daylight	Raining + high winds	1
15	2018	City of London	E09000001	Motorcycle	Darkness - lights lit	Fine no high winds	13
16	2018	City of London	E09000001	Motorcycle	Darkness - lights lit	Raining no high winds	5
17	2018	City of London	E09000001	Car (Includes taxis and minibus)	Daylight	Fine no high winds	11
18	2018	City of London	E09000001	Car (Includes taxis and minibus)	Daylight	Other	1
19	2018	City of London	E09000001	Car (Includes taxis and minibus)	Darkness - lights lit	Fine no high winds	15
20	2018	City of London	E09000001	Bus or coach	Daylight	Fine no high winds	14
21	2018	City of London	E09000001	Bus or coach	Darkness - lights lit	Fine no high winds	5
22	2018	City of London	E09000001	Other vehicle	Darkness - lights lit	Fine no high winds	1
23	2018	Barking	E09000002	Pedestrian	Daylight	Unknown	3
24	2018	Barking	E09000002	Pedestrian	Daylight	Fine no high winds	59
25	2018	Barking	E09000002	Pedestrian	Daylight	Raining no high winds	11

The table illustrates dataset 5 (section 2.1.)

- We selected only the rows with our target user, pedal cyclists, in Python.
- Then, we merged it with datasets 6 and 7 (section 2.1.).

6	LA Code	Local Authority		1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010R
7	**K03000001**	**Great Britain**		**412.336**	**421.540**	**429.722**	**441.123**	**450.324**	**458.491**	**466.960**	**466.168**	**472.630**	**483.710**	**486.665**	**493.948**	**493.872**	**501.082**	**505.439**	**500.566**	**495.825**	**492.073**
8	**E92000001**	**England**		**355.306**	**363.157**	**370.167**	**379.944**	**387.799**	**395.057**	**402.518**	**402.031**	**407.647**	**416.369**	**418.481**	**424.408**	**424.376**	**429.593**	**433.021**	**428.494**	**424.402**	**421.703**
9	**S92000003**	**Scotland**		**34.938**	**35.765**	**36.496**	**37.542**	**38.347**	**38.941**	**39.532**	**39.319**	**39.829**	**41.285**	**41.789**	**42.474**	**42.520**	**43.859**	**44.426**	**44.197**	**43.932**	**43.197**
10	**W92000004**	**Wales**		**22.093**	**22.618**	**23.060**	**23.638**	**24.178**	**24.493**	**24.909**	**24.818**	**25.153**	**26.056**	**26.395**	**27.066**	**26.976**	**27.629**	**27.992**	**27.875**	**27.491**	**27.172**
11	**E12000001**	**North East**		**16.516**	**16.815**	**17.099**	**17.445**	**17.824**	**18.080**	**18.514**	**18.260**	**18.455**	**18.786**	**18.775**	**18.891**	**18.721**	**18.852**	**19.000**	**18.660**	**18.486**	**18.306**
12	E06000005	Darlington		682	695	708	734	765	780	798	777	797	823	819	809	812	817	806	794	784	777
13	E06000047	Durham	1	3.111	3.192	3.254	3.338	3.472	3.542	3.623	3.606	3.627	3.708	3.734	3.801	3.792	3.865	3.959	3.901	3.864	3.840
14	E06000001	Hartlepool		551	559	567	580	599	603	619	610	623	637	641	636	628	629	621	617	607	593
15	E06000002	Middlesbrough		1.134	1.149	1.163	1.173	1.191	1.203	1.242	1.217	1.233	1.241	1.238	1.242	1.224	1.211	1.204	1.176	1.172	1.148
16	E06000057	Northumberland	1	2.139	2.207	2.258	2.291	2.322	2.353	2.446	2.407	2.461	2.488	2.493	2.557	2.550	2.616	2.652	2.587	2.555	2.568
17	E06000003	Redcar and Cleveland		841	855	868	879	892	899	924	903	889	902	908	896	898	920	924	916	893	890
18	E06000004	Stockton-on-Tees		1.288	1.305	1.326	1.362	1.383	1.410	1.448	1.447	1.457	1.483	1.482	1.473	1.441	1.445	1.431	1.405	1.425	1.405
19	**E11000007**	**Tyne and Wear ITA**		6.769	6.853	6.954	7.088	7.199	7.289	7.414	7.295	7.369	7.505	7.460	7.478	7.377	7.349	7.404	7.265	7.185	7.083
20	E08000037	Gateshead		1.680	1.707	1.731	1.782	1.816	1.840	1.852	1.807	1.814	1.847	1.836	1.818	1.782	1.741	1.779	1.744	1.746	1.704
21	E08000021	Newcastle upon Tyne		1.669	1.689	1.712	1.718	1.744	1.762	1.802	1.764	1.798	1.818	1.800	1.812	1.793	1.763	1.778	1.746	1.717	1.706
22	E08000022	North Tyneside		1.172	1.180	1.198	1.222	1.238	1.253	1.279	1.248	1.274	1.298	1.279	1.297	1.286	1.272	1.289	1.268	1.251	1.244
23	E08000023	South Tyneside		687	696	706	718	727	732	745	743	749	760	763	762	748	758	748	736	731	725
24	E08000024	Sunderland		1.561	1.581	1.607	1.648	1.675	1.703	1.736	1.733	1.734	1.781	1.782	1.789	1.768	1.815	1.809	1.771	1.739	1.705
25	**E12000002**	**North West**		**46.518**	**47.375**	**48.387**	**49.601**	**50.678**	**51.661**	**52.474**	**52.424**	**53.214**	**54.250**	**54.531**	**55.602**	**55.247**	**55.778**	**56.026**	**55.572**	**55.407**	**55.070**
26	E06000008	Blackburn with Darwen		615	624	637	650	664	670	670	677	672	687	689	706	715	706	730	714	700	705
27	E06000009	Blackpool		540	544	552	558	560	565	574	568	574	581	572	575	572	550	557	556	551	550
28	E10000004	Cheshire	1	6.872	7.017	7.183	7.473	7.680	7.824	7.948	7.976	8.121	8.018	8.208	8.380	8.321	8.527	8.528	8.422	..	..

The table illustrates dataset 6 (section 2.1.)

We saved the ods file as an Excel file, then copied only the values for London local authorities and 2018 to a different file. This allowed us to merge this dataset with the previous one in Python - we performed an inner merge by local authority - so both casualties and traffic in each local authority are shown.

Time of day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
00:00-01:00	12,0	11,3	12,2	12,9	14,5	21,5	25,1
01:00-02:00	7,4	7,3	7,7	8,0	9,0	13,4	15,4
02:00-03:00	6,0	6,2	6,4	6,7	7,4	9,6	10,3
03:00-04:00	7,4	7,4	7,4	7,8	8,3	9,0	8,6
04:00-05:00	14,4	13,0	12,9	13,2	13,2	10,6	8,6
05:00-06:00	40,8	38,3	37,6	37,3	35,6	19,4	12,7
06:00-07:00	97,8	98,3	96,9	95,8	88,1	34,8	21,5
07:00-08:00	180,1	184,7	183,2	180,6	167,0	60,1	34,5
08:00-09:00	198,2	205,2	204,5	203,0	190,6	97,4	53,5
09:00-10:00	148,8	153,6	154,0	154,8	151,6	133,3	93,4
10:00-11:00	141,1	138,6	139,3	142,5	151,9	162,0	133,7
11:00-12:00	146,4	140,8	143,1	146,3	162,2	176,9	156,8
12:00-13:00	149,2	145,1	147,7	151,1	172,2	179,0	167,2
13:00-14:00	148,7	146,1	149,1	152,3	175,0	170,0	160,0
14:00-15:00	157,7	157,7	161,0	164,0	185,7	159,5	153,3
15:00-16:00	177,7	181,8	186,1	187,8	206,5	150,7	151,3
16:00-17:00	201,1	207,8	209,7	210,4	212,7	148,9	148,2
17:00-18:00	206,1	214,4	215,8	215,8	205,9	143,2	132,4
18:00-19:00	152,7	163,5	166,4	169,5	166,1	122,4	114,1
19:00-20:00	101,0	108,5	112,4	118,1	122,2	94,4	93,2
20:00-21:00	69,2	73,6	77,2	82,8	84,6	67,8	72,1

The table illustrates dataset 7 (section 2.1.)

	A	B
1	Light condition	traffic
2	Daylight	13.534
3	Darkness - lights lit	3264
4	Darkness - lights unlit	3264
5	Darkness - no lighting	3264
6	Darkness - lighting unknown	3264

We wanted to calculate traffic by daylight and darkness for the whole year.
We assumed that daylight falls between 7am - 7pm. We calculated the total traffic for both daylight and darkness in Excel and performed a Python inner merge by light condition with the dataset 5.

The final merge is illustrated in section 3. We used this final merge to calculate our new field - casualties / traffic (by local authority) / traffic (by daylight or darkness) in Python (Appendix 6.3.).

We did exactly the same for the 2019 data.

4.2.Noteworthy technical challenges

4.2.1. Python

We used Python to calculate the average of our new variable - casualty rate by traffic - then categorised it in Safe/Dangerous depending on the average. (see appendix 6.4 for the code)

As our app uses a Support Vector Machine Model, we used Python's LabelEncoder() to label encode our 3 predictor variables - location, weather, light. (see appendix 6.4)

Then, we trained our model on 2018 data and tested it on 2019 data - the model's accuracy rate is 81%. (see appendix 6.4)

We created a Python user interface for reporting accidents - users report the number of cycle casualties, light, weather and location - this is useful as it gives more data (our open-data is updated yearly so we plan to create our own data from users and update our data monthly).

```
In [2]: # Creating an empty Dataframe with column names only
        dfnew = pd.DataFrame(columns=['Local authority','Light condition','Weather condition', 'NrofCasualt
        dfnew
```

Out[2]:

Local authority	Light condition	Weather condition	NrofCasualties_pedalcycle

```
In [*]: light= input("Tell me the light conditions at this moment -  ")
        w=input("Tell me the weather conditions at this moment -  ")
        a=input("Tell me the number of pedal cycle casualties - ")
        loc=input("Show me where are you on the map - ")


        dfnew = dfnew.append({'Local authority':loc,'Light condition':light,'Weather condition': w , 'NrofC
        dfnew
```

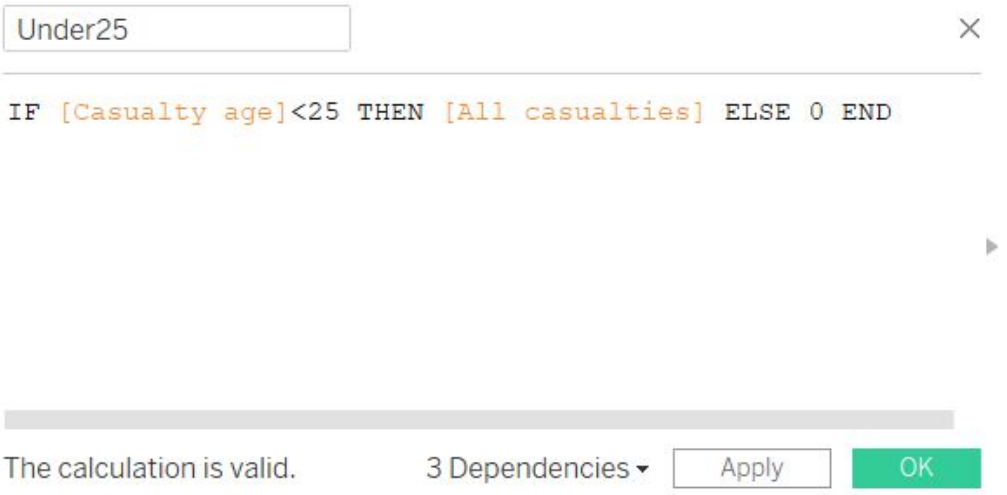
```
Tell me the light conditions at this moment -  darkness
Tell me the weather conditions at this moment -  Fine high winds
Tell me the number of pedal cycle casualties - 2
```

Out[3]:

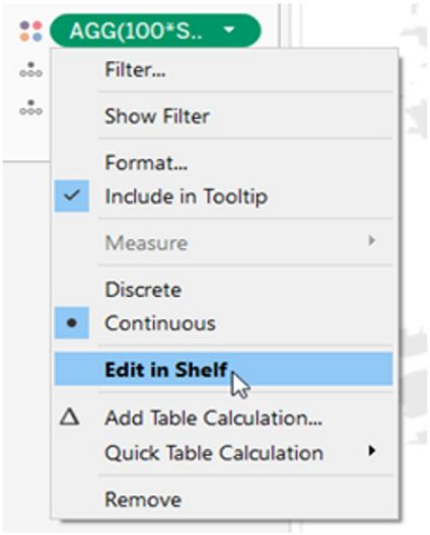
	Local authority	Light condition	Weather condition	NrofCasualties_pedalcycle
0	Camden	darkness	Fine high winds	2

4.2.2. Tableau

One technical challenge was to find the percentage of people under 25 from the dataset 11 (section 2.1.).
Applying an IF/ELSE conditional statement, we created a new Calculated Field in Tableau, representing the accidents involving people under 25.



This Field was used to express the casualties involving people under 25 as a percentage of the total number of casualties in each local authority. The 'Under25' measure was dragged onto the 'Color' mark and the Edit in Shelf function was used to compute the percentage.



Shelf Input:

100*SUM([Under25])/SUM([All casualties])

This allowed us to make a map to support our previous business idea (Visual 3 in Section 2).

5. Conclusions

5.1. Self-critique

CycleYourWay mainly uses open data for analyzing information about accidents and safety conditions in London. Although a lot of open data was used, either individually or merged, CycleYourWay could be doing better:

Value	• Though CycleYourWay cross-analyses many open datasets, we could have used more to optimize the results. For example, one open dataset provided the severity of bike accidents, which would have given us more information about incidents, allowing us to present a more in-depth description of the accident. • The data regarding the number of casualties by local authority in London is only from the last two years (2018, 2019). Although this is valuable, it doesn't provide the whole picture. Thus, the overall safety of an area may be inaccurate as a larger time frame is not considered.
Quality	• The open data that CycleYourWay uses is updated yearly, however, we added the accident report feature to minimise this problem. This feature allows us to update the app monthly (if the number of user reports is enough).
Nature	• CycleYourWay uses quantitative open data to determine the safety of areas in London. Additionally, through the accident report feature, the app uses qualitative data from users to provide information on accidents.

5.2. Other Data CycleYourWay Could Have Used

CycleYourWay would benefit from a wide range of additional open data, as it is at the core of its functions. These are:

Data on Bikes Accessories involved in Accidents

Accessories installed on a bike, such as lights or wheel strips provide other traffic participants a better chance to see the bike riders, preventing cycle accidents. Therefore, this data would help us to determine whether the accidents are due to unequipped bikes, thus providing more in-depth analysis. This can be applied to data on equipment (reflective vests and helmets) worn by cyclists.

Surveys About Bikers Cycling in London

Some surveys provide data freely for everyone. CycleYourWay would benefit substantially from surveys about bikers cycling in London, expanding our data storage. (9)

Data on the Type of Bicycle Involved in Accidents

It would tell us if the bike type involved was suitable for the surface of the road and conditions at the time of the accident. This data would provide a more in-depth analysis of the accident.

6. Appendices

6.1. CycleYourWay

Existing alternatives

The Transport for London (Tfl) has been committed to make cycling in London safer. In particular, It has come up with:

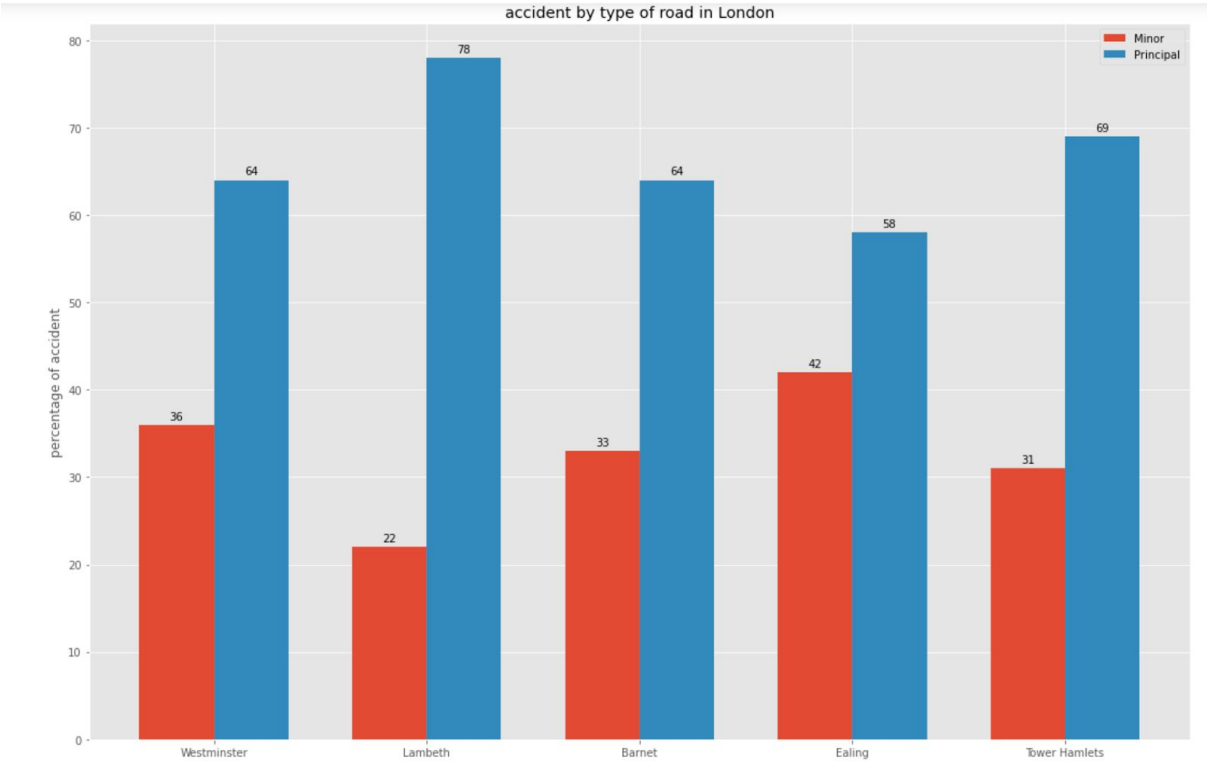
- *A Cycle Safety Action Plan*: it sets out how Transport for London (TfL) will strive to reduce the number of cyclists killed or seriously injured on London's roads.
  The action plan contains 52 separate actions to improve the safety of cyclists, focusing on the following key areas:

    - Safer infrastructure
    - Training and information
    - Communication and enforcement
    - Regulation
    - Vehicle technology
    - Commercial driving and working practices
    - Research
    - Monitoring and continued partnership working

- The Direct Vision standard: Using a star system, the DVS rates HGVs from 0 (lowest) to 5 (highest) based on how much a HGV driver can see directly through their cab windows, as opposed to indirectly through cameras or mirrors.Under the Mayor's proposal, 0-star rated HGVs will be banned or restricted from entering London from 2020, and by 2024 only those rated 3 star, or above, will be allowed into the city. These proposals are subject to consultation.

6.2. Visualisations

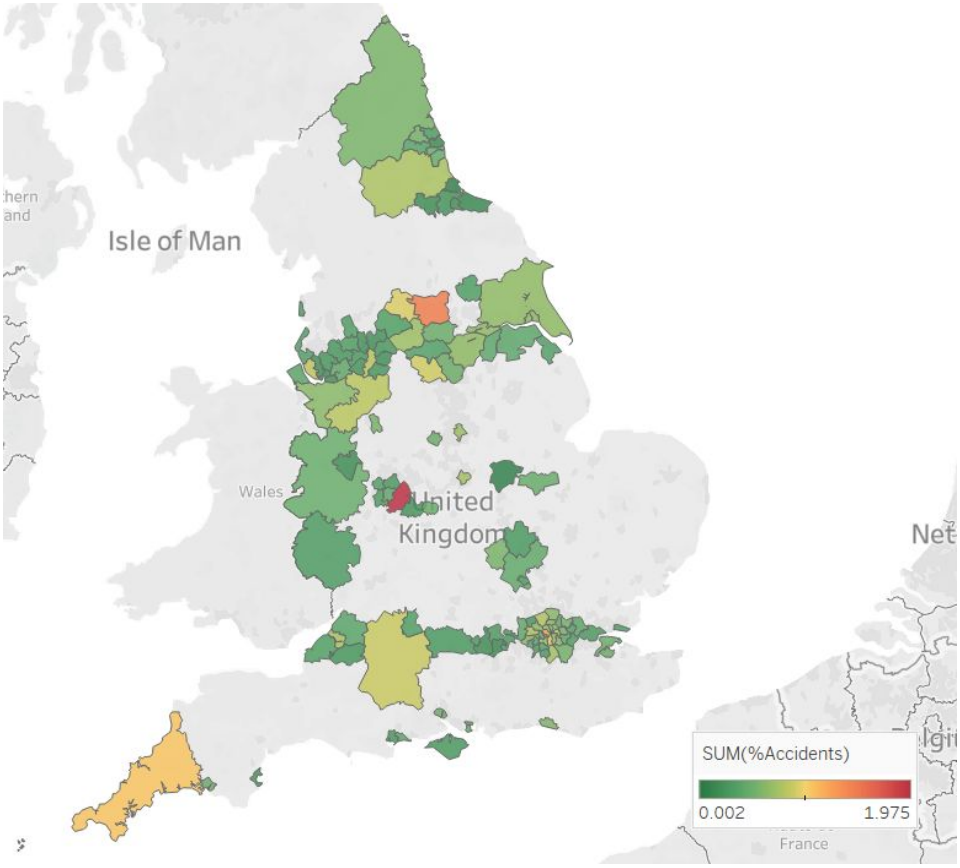
Additional Visual 1



Dataset 10 (section 2.1.) was used to plot this graph in Python. It shows 5 local authorities with the most accidents in London. The percentage of accidents on minor roads and principal roads are represented by red bars and blue bars respectively. Note that the total of bars for Barnet is not 100 as highways are not considered.
The graph illustrates that more accidents happen on principal roads. This could be explained by higher speed and volume of traffic on principal roads.

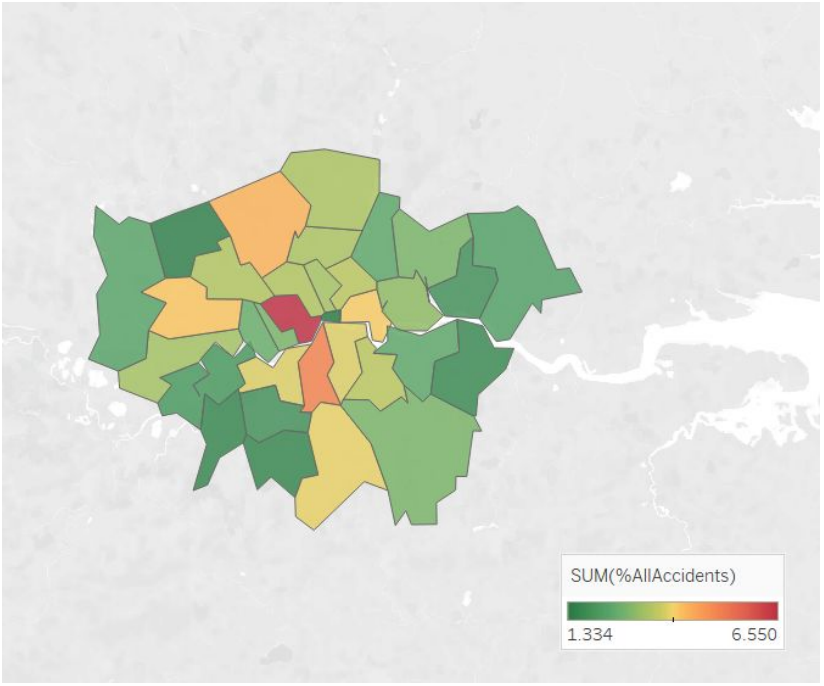
Additional Visual 2

Dataset 10 (from 2.1.)



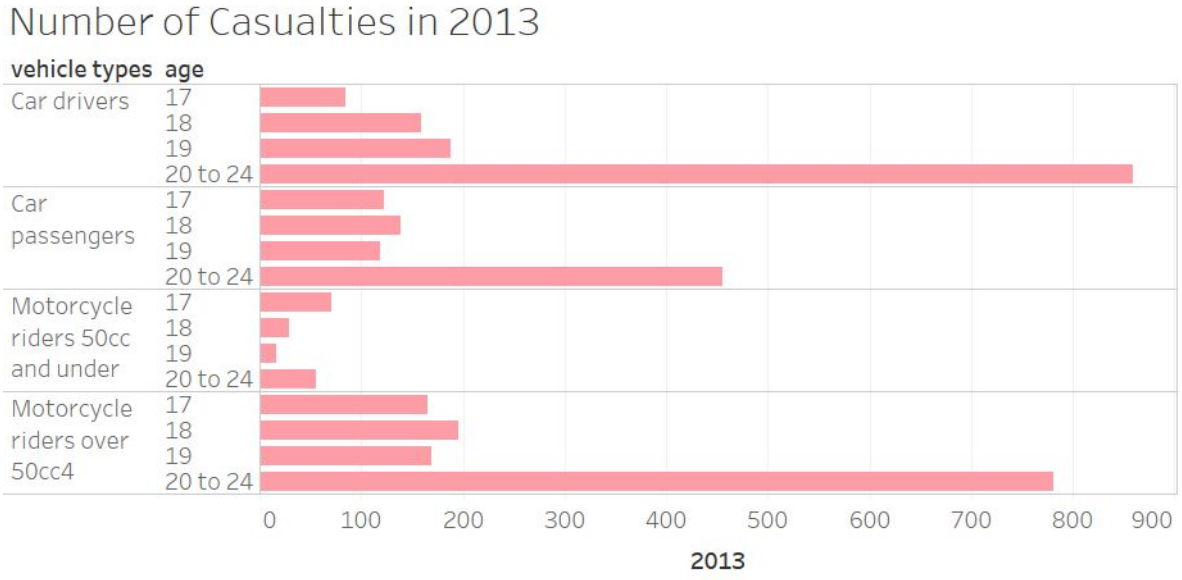
Additional visual 3

Dataset 10 (from 2.1.)



Additional visual 4

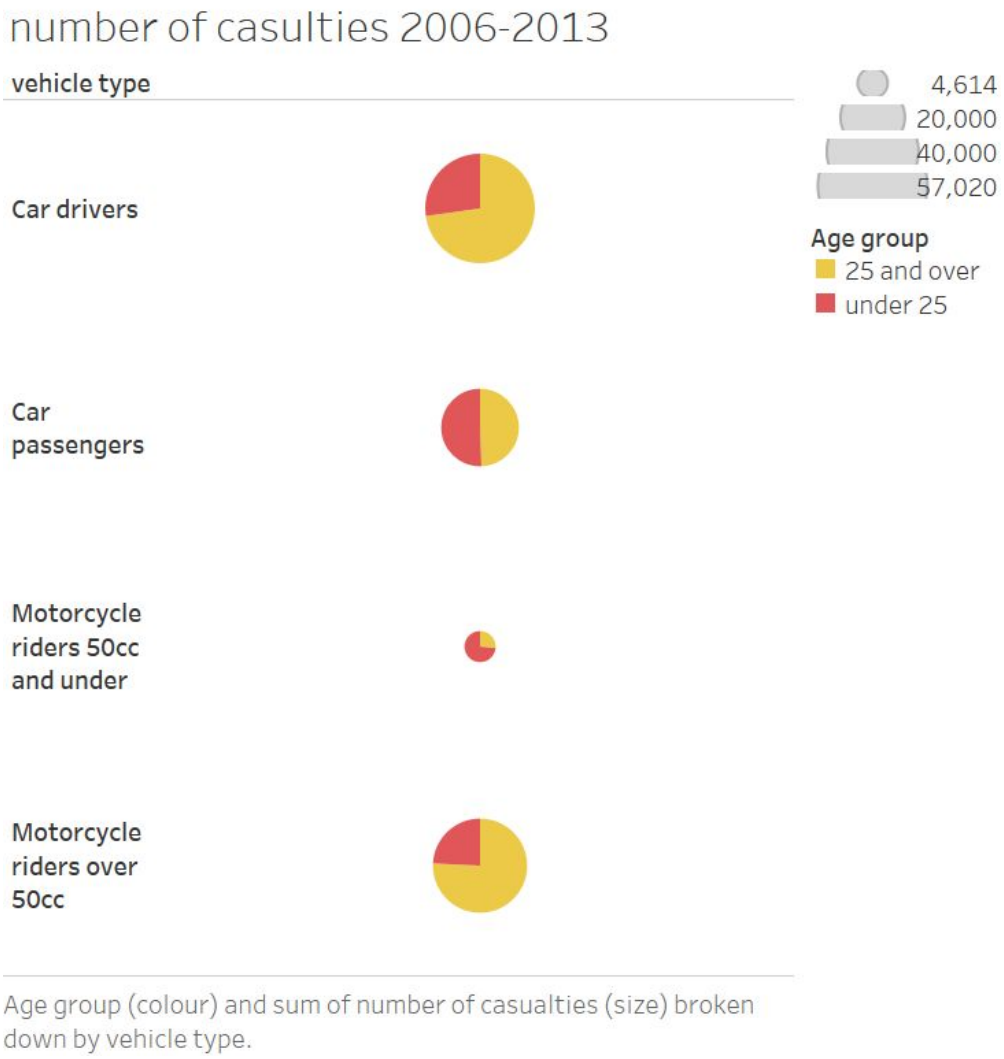
Dataset 8 (from 2.1.)



Sum of 2013 for each age broken down by vehicle types.

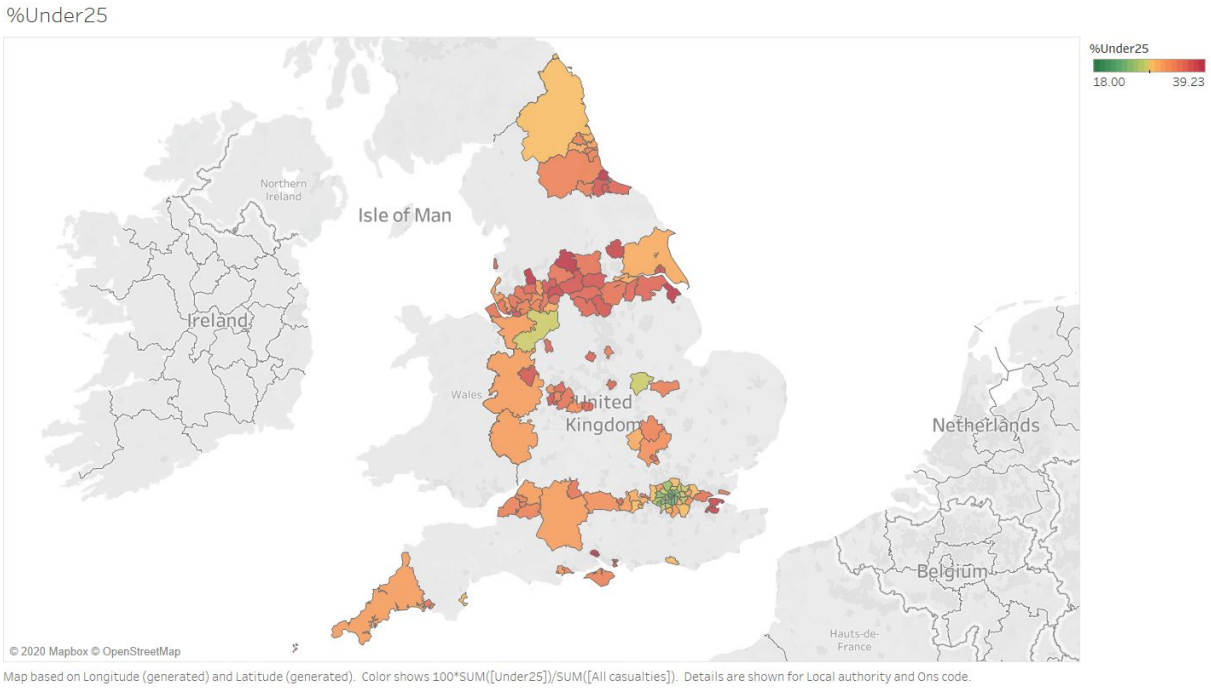
Additional visual 5

Dataset 8 (from section 2.1.)



Additional visual 6

Dataset 11 (from section 2.1.)



Map based on Longitude (generated) and Latitude (generated). Color shows 100*SUM([Under25])/SUM([All casualties]). Details are shown for Local authority and Ons code.

6.3. Python code for the app

```
In [2]: import pandas as pd

In [57]: light2018=pd.read_excel("light_traffic_2018.xlsx")
         light2018
```

Out[57]:

	Light condition	traffic
0	Daylight	13534
1	Darkness - lights lit	3264
2	Darkness - lights unlit	3264
3	Darkness - no lighting	3264
4	Darkness - lighting unknown	3264

```
In [58]: light2019=pd.read_excel("lightcondition_trafic2019.xlsx")

In [59]: df = pd.read_excel("2018casualties.xlsx")

In [6]: df
```

Out[6]:

	Accident year	Local authority	Road user	Light condition	Weather condition	All casualties
0	2018	City of London	Pedestrian	Daylight	Fine no high winds	59
1	2018	City of London	Pedestrian	Daylight	Raining no high winds	2
2	2018	City of London	Pedestrian	Daylight	Snowing no high winds	1
3	2018	City of London	Pedestrian	Darkness - lights lit	Fine no high winds	17
4	2018	City of London	Pedestrian	Darkness - lights lit	Raining no high winds	5
...	...	...	...	...	...	...
2176	2018	Westminster	Van / Goods 3.5 tonnes mgw or under	Darkness - lighting unknown	Raining no high winds	2
2177	2018	Westminster	HGV	Daylight	Fine no high winds	1
2178	2018	Westminster	HGV	Darkness - lights lit	Fine no high winds	2

In [60]:
```
df2019=pd.read_excel("casualties2019.xlsx")
```

In [61]:
```
df=df.loc[df["Road user"] == "Pedal cycle"]
```

In [62]:
```
df
```

Out[62]:

	Accident year	Local authority	Road user	Light condition	Weather condition	All casualties
5	2018	City of London	Pedal cycle	Daylight	Unknown	1
6	2018	City of London	Pedal cycle	Daylight	Fine no high winds	71
7	2018	City of London	Pedal cycle	Daylight	Raining no high winds	5
8	2018	City of London	Pedal cycle	Darkness - lights lit	Fine no high winds	21
9	2018	City of London	Pedal cycle	Darkness - lights lit	Raining no high winds	2
...	...	...	...	...	...	...
2121	2018	Westminster	Pedal cycle	Darkness - lights unlit	Fine no high winds	4
2122	2018	Westminster	Pedal cycle	Darkness - no lighting	Fine no high winds	1
2123	2018	Westminster	Pedal cycle	Darkness - lighting unknown	Unknown	1
2124	2018	Westminster	Pedal cycle	Darkness - lighting unknown	Fine no high winds	7
2125	2018	Westminster	Pedal cycle	Darkness - lighting unknown	Raining no high winds	1

359 rows × 6 columns

In [63]:
```
traffic= pd.read_excel("traficloc2018.xlsx")
traffic
```

Out[63]:

	Local authority	Trafficmillkm
0	Barking	798
1	Barnet	2034
2	Bexley	1230
3	Brent	1056

In [64]:
```
mergedata = pd.merge(df,traffic, how='inner', on='Local authority')
```

In [66]:
```
mergedata2=pd.merge(mergedata,light2018,how='inner',on='Light condition')
```

In [67]:
```
mergedata2['casualties/milkmtrafic']=mergedata2['All casualties']/mergedata2['Trafficmillkm']/mergedata2["traffic"]*10(
```

In [69]:
```
mergedata2['Local authority1']=mergedata2['Local authority']
mergedata2['Light condition1']=mergedata2['Light condition']
mergedata2['Weather condition1']=mergedata2['Weather condition']
mergedata2
```

Out[69]:

	Accident year	Local authority	Road user	Light condition	Weather condition	All casualties	Trafficmillkm	traffic	casualties/milkmtrafic	Local authority1	Light condition1	Weather condition1
0	2018	City of London	Pedal cycle	Daylight	Unknown	1	167	13534	0.044244	City of London	Daylight	Unknown
1	2018	City of London	Pedal cycle	Daylight	Fine no high winds	71	167	13534	3.141346	City of London	Daylight	Fine no high winds
2	2018	City of London	Pedal cycle	Daylight	Raining no high winds	5	167	13534	0.221222	City of London	Daylight	Raining no high winds
3	2018	Barking	Pedal cycle	Daylight	Fine no high winds	28	798	13534	0.259256	Barking	Daylight	Fine no high winds
4	2018	Barking	Pedal cycle	Daylight	Raining no high winds	2	798	13534	0.018518	Barking	Daylight	Raining no high winds
...	...	...	...	...	...	...	...	...	...	...	...	...
350	2018	Richmond upon Thames	Pedal cycle	Darkness - no lighting	Raining no high winds	1	878	3264	0.034894	Richmond upon Thames	Darkness - no lighting	Raining no high winds
351	2018	Richmond upon Thames	Pedal cycle	Darkness - no lighting	Raining + high winds	1	878	3264	0.034894	Richmond upon Thames	Darkness - no lighting	Raining + high winds
352	2018	Tower Hamlets	Pedal cycle	Darkness - no lighting	Fine no high winds	1	1003	3264	0.030546	Tower Hamlets	Darkness - no lighting	Fine no high winds
353	2018	Waltham Forest	Pedal cycle	Darkness - no lighting	Other	1	979	3264	0.031294	Waltham Forest	Darkness - no lighting	Other
354	2018	Westminster	Pedal cycle	Darkness -	Fine no high	1	996	3264	0.033816	Westminster	Darkness -	Fine no

In [71]:
```
df2019=df2019.loc[df2019["Road user"] == "Pedal cycle"]
```

In [72]:
```
mergedata2019 = pd.merge(df2019,traffic2019, how='inner', on='Local authority')
```

In [73]:
```
fmergedata2019= pd.merge(mergedata2019,light2019, how='inner', on='Light condition')
```

In [75]:
```
fmergedata2019['Local authority1']=fmergedata2019['Local authority']
fmergedata2019['Light condition1']=fmergedata2019['Light condition']
fmergedata2019['Weather condition1']=fmergedata2019['Weather condition']
```

In [76]:
```
ualties/milkmtrafic']=fmergedata2019['All casualties']/fmergedata2019['Trafifcmillkm']/fmergedata2019["traffic"]*100000
```

In [84]:
```
out=fmergedata2019
out
out.to_excel("visualtableaufinal.xlsx")
```

In [77]:
```
encode=['Local authority','Light condition','Weather condition']

from sklearn.preprocessing import LabelEncoder


# Apply label encoder
label_encoder = LabelEncoder()
for col in set(encode):
    mergedata2[col] = label_encoder.fit_transform(mergedata2[col])
    fmergedata2019[col] = label_encoder.transform(fmergedata2019[col])
```

In [78]:
```
mergedata2
```

Out[78]:

	Accident year	Local authority	Road user	Light condition	Weather condition	All casualties	Trafficmillkm	traffic	casualties/milkmtrafic	Local authority1	Light condition1	Weather condition1
0	2018	6	Pedal cycle	4	8	1	167	13534	0.044244	City of London	Daylight	Unknown
1	2018	6	Pedal cycle	4	1	71	167	13534	3.141346	City of London	Daylight	Fine no high winds

In [79]:
```
#svm

x_train=mergedata2[["Local authority","Light condition","Weather condition"]]
y_train=mergedata2["casualties/milkmtrafic"]


x_test=fmergedata2019[["Local authority","Light condition","Weather condition"]]
y_test=fmergedata2019["casualties/milkmtrafic"]
```

In [80]:
```
y_train
```

Out[80]:
```
0      0.044244
1      3.141346
2      0.221222
3      0.259256
4      0.018518
         ...
350    0.034894
351    0.034894
352    0.030546
353    0.031294
354    0.033816
Name: casualties/milkmtrafic, Length: 355, dtype: float64
```

In [81]:
```
y_train.mean()
```

Out[81]:
```
0.2343371911546619
```

In [82]:
```
mergedata2['casualties/milkmtrafic'] = mergedata2['casualties/milkmtrafic'].astype(float)
mergedata2['casualties/milkmtrafic2'] = mergedata2['casualties/milkmtrafic'].astype(float)
```

In [83]:
```
mergedata2["casualties/milkmtrafic2"].loc[mergedata2["casualties/milkmtrafic"] > 0.2343371911546619]="Dangerous"
```

```
/opt/anaconda3/lib/python3.7/site-packages/pandas/core/indexing.py:205: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_with_indexer(indexer, value)
```

In [84]:
```
mergedata2["casualties/milkmtrafic2"].loc[mergedata2["casualties/milkmtrafic2"] != "Dangerous"]="Safe"
```

In [85]:
```
mergedata2
```

Out[85]:

	Accident year	Local authority	Road user	Light condition	Weather condition	All casualties	Trafficmillkm	traffic	casualties/milkmtrafic	Local authority1	Light condition1	Weather condition1	casualties/milk
0	2018	6	Pedal cycle	4	8	1	167	13534	0.044244	City of London	Daylight	Unknown	
1	2018	6	Pedal cycle	4	1	71	167	13534	3.141346	City of London	Daylight	Fine no high winds	D
2	2018	6	Pedal cycle	4	5	5	167	13534	0.221222	City of London	Daylight	Raining no high winds	
3	2018	0	Pedal cycle	4	1	28	798	13534	0.259256	Barking	Daylight	Fine no high winds	D
4	2018	0	Pedal cycle	4	5	2	798	13534	0.018518	Barking	Daylight	Raining no high winds	
...	...	...	...	...	...	...	...	...	...	...	...	...	
350	2018	26	Pedal cycle	3	5	1	878	3264	0.034894	Richmond upon Thames	Darkness - no lighting	Raining no high winds	
351	2018	26	Pedal cycle	3	4	1	878	3264	0.034894	Richmond upon Thames	Darkness - no lighting	Raining + high winds	
352	2018	29	Pedal cycle	3	1	1	1003	3264	0.030546	Tower Hamlets	Darkness - no lighting	Fine no high winds	
353	2018	30	Pedal cycle	3	3	1	979	3264	0.031294	Waltham Forest	Darkness - no lighting	Other	
354	2018	32	Pedal cycle	3	1	1	906	3264	0.033816	Westminster	Darkness - no lighting	Fine no high winds	

355 rows × 13 columns

In [86]:
```
y_train=mergedata2["casualties/milkmtrafic2"]
```

In [87]:
```
fmergedata2019['casualties/milkmtrafic'] = fmergedata2019['casualties/milkmtrafic'].astype(float)
```

fmergedata2019['casualties/milkmtrafic2'] = fmergedata2019['casualties/milkmtrafic'].astype(float)

In [88]:
```
y_test.mean()
```

Out[88]:
```
0.231860205267536
```

In [89]:
```
fmergedata2019["casualties/milkmtrafic2"].loc[fmergedata2019["casualties/milkmtrafic"] > 0.231860205267536]="Dangerous"
fmergedata2019["casualties/milkmtrafic2"].loc[fmergedata2019["casualties/milkmtrafic2"] != "Dangerous"]="Safe"
```

In [90]:
```
y_test=fmergedata2019["casualties/milkmtrafic2"]
y_train=mergedata2["casualties/milkmtrafic2"]
```

In [91]:
```
#svm
from sklearn import svm
```

In [92]:
```
# Chooses the support vector machine algorithm for our classifier
clf = svm.SVC(gamma='auto')

# Training the classifier
clf_trained = clf.fit(x_train,y_train)

# Scoring the classifier
clf_trained.score(x_test,y_test)
```

Out[92]:
```
0.8192090395480226
```

In [52]:
```
#user input

#light="Daylight"
#weather="Fine no high winds"
#local="Westminster"

light= input("Tell me the light conditions at this moment -  ")
weather=input("Tell me the weather conditions at this moment -  ")
local=input("Show me where are you on the map - ")
```

```
x=fmergedata2019["Light condition"].loc[fmergedata2019["Light condition1"] ==light]

x=pd.DataFrame(x)
x.reset_index(drop=True, inplace=True)
x1=x["Light condition"][0]

y=fmergedata2019["Weather condition"].loc[fmergedata2019["Weather condition1"] ==weather]
y=pd.DataFrame(y)
y.reset_index(drop=True, inplace=True)
y1=y["Weather condition"][0]

z=fmergedata2019["Local authority"].loc[fmergedata2019["Local authority1"] ==local]
z=pd.DataFrame(z)
z.reset_index(drop=True, inplace=True)
z1=z["Local authority"][0]

test_point2= np.array([[x1,y1,z1]])


#OUTPUTS

# 1. Making a prediction
clf_trained.predict(test_point2)
```

```
Tell me the light conditions at this moment -  Darkness - lights lit
Tell me the weather conditions at this moment -  Raining no high winds
Show me where are you on the map - Ealing
```

Out[52]:
```
array(['Safe'], dtype=object)
```

In [53]:
```
#2. Describe

b=fmergedata2019.loc[fmergedata2019["Light condition"]==x1]
c=b.loc[b["Weather condition"]==y1]
c
```

Out[53]:

	Local authority	Road user	Light condition	Weather condition	All casualties	Trafifcmillkm	traffic	casualties/milkmtrafic	Local authority1	Light condition1	Weather condition1	casualties/milkmtrafic2
151	6	Pedal	1	5	7	169	3251	1.274073	City of	Darkness -	Raining no	Dangerous

In [114]:
```
#light dataset

# Creating an empty Dataframe with column names only
dflight = pd.DataFrame(columns=['Light condition', 'Casualties'])

for value in fmergedata2019["Light condition1"].unique():
    a=fmergedata2019["All casualties"].loc[fmergedata2019["Light condition1"]==value].sum()
    dflight = dflight.append({'Light condition': value , 'Casualties': a }, ignore_index=True)

dflight
```

Out[114]:

	Light condition	Casualties
0	Daylight	3511
1	Darkness - lights lit	892
2	Darkness - lights unlit	22
3	Darkness - no lighting	11
4	Darkness - lighting unknown	193

In [177]:
```
# Converts data frame into a csv file without an index column
dflight.to_excel("lightcondition_pedalcycle.xlsx",index=False)
```

In [94]:
```
#weather dataset

# Creating an empty Dataframe with column names only
dfweather = pd.DataFrame(columns=['Weather condition', 'Casualties'])

for value in df2019["Weather condition"].unique():
    a=df2019["All casualties"].loc[df2019["Weather condition"]==value].sum()
    dfweather = dfweather.append({'Weather condition': value , 'Casualties': a }, ignore_index=True)

dfweather
```

Out[94]:

	Weather condition	Casualties
0	Fine no high winds	3706

In [96]:
```
#local authority dataset

# Creating an empty Dataframe with column names only
dflocal = pd.DataFrame(columns=['Local authority', 'Casualties/1millkmtraffic'])

for value in fmergedata2019["Local authority1"].unique():
    a=fmergedata2019["casualties/milkmtrafic"].loc[fmergedata2019["Local authority1"]==value].sum()
    dflocal = dflocal.append({'Local authority': value , 'Casualties/1millkmtraffic': a }, ignore_index=True)

dflocal
```

Out[96]:

	Local authority	Casualties/1millkmtraffic
0	City of London	10.041372
1	Barking	1.014294
2	Barnet	0.381217
3	Bexley	0.247995
4	Brent	0.904495
5	Bromley	0.689726
6	Camden	6.211776
7	Croydon	0.657814
8	Ealing	1.081038
9	Enfield	0.266193
10	Greenwich	0.907859
11	Hackney	5.803785
12	Hammersmith	3.210155
13	Haringey	2.551053
14	Harrow	0.488713
15	Havering	0.180483
16	Hillingdon	0.255743
17	Hounslow	0.916539

6.4. Other ways we used Open Data to add value

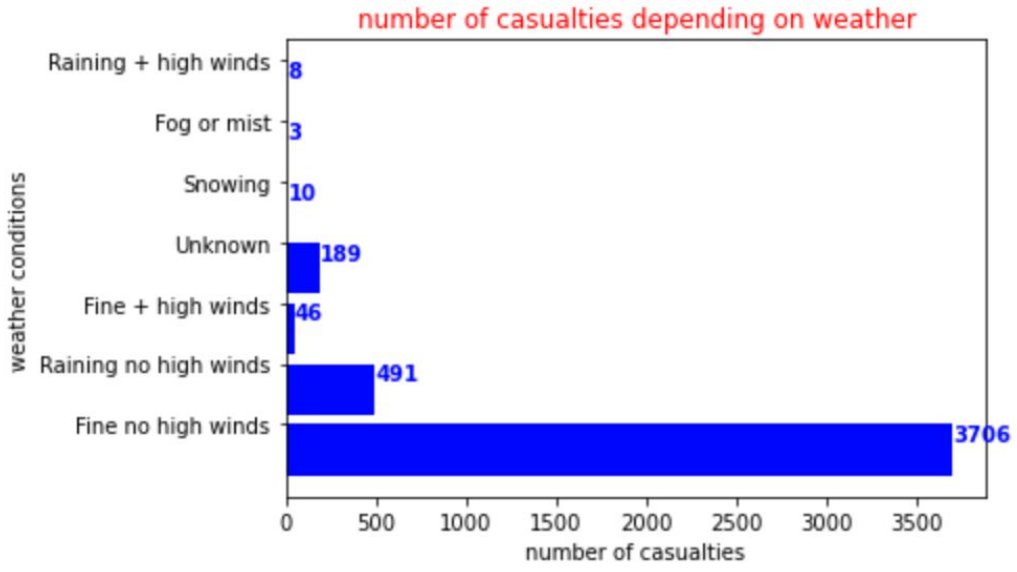
With the merged open data, we came up with another service we could offer to users, giving further information on the safety level by location, weather and light conditions separately.

In [94]:
```
#weather dataset

# Creating an empty Dataframe with column names only
dfweather = pd.DataFrame(columns=['Weather condition', 'Casualties'])

for value in df2019["Weather condition"].unique():
    a=df2019["All casualties"].loc[df2019["Weather condition"]==value].sum()
    dfweather = dfweather.append({'Weather condition': value , 'Casualties': a }, ignore_index=True)

dfweather
```

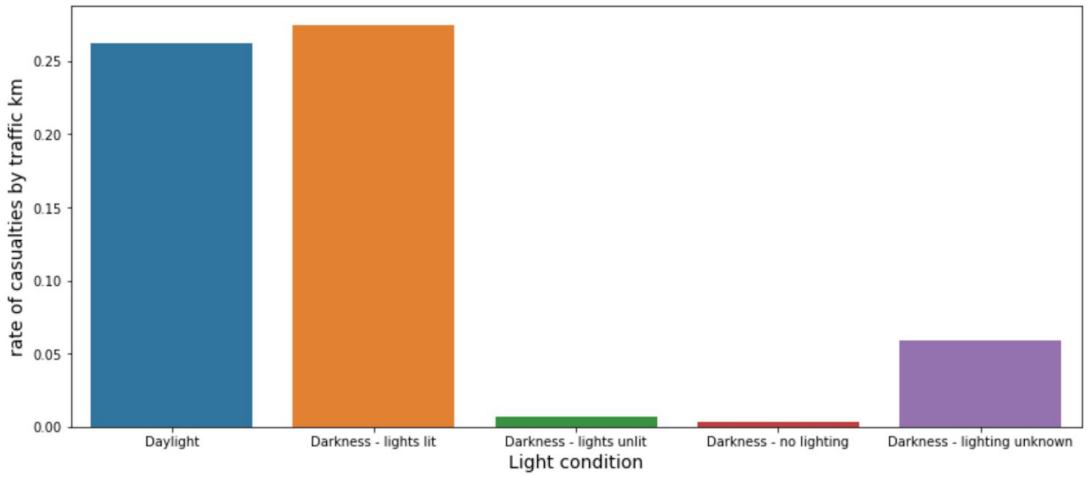
Out[94]:

	Weather condition	Casualties
0	Fine no high winds	3706
1	Raining no high winds	491
2	Fine + high winds	46
3	Unknown	189
4	Other	177
5	Snowing no high winds	8
6	Snowing + high winds	2
7	Fog or mist	3
8	Raining + high winds	8

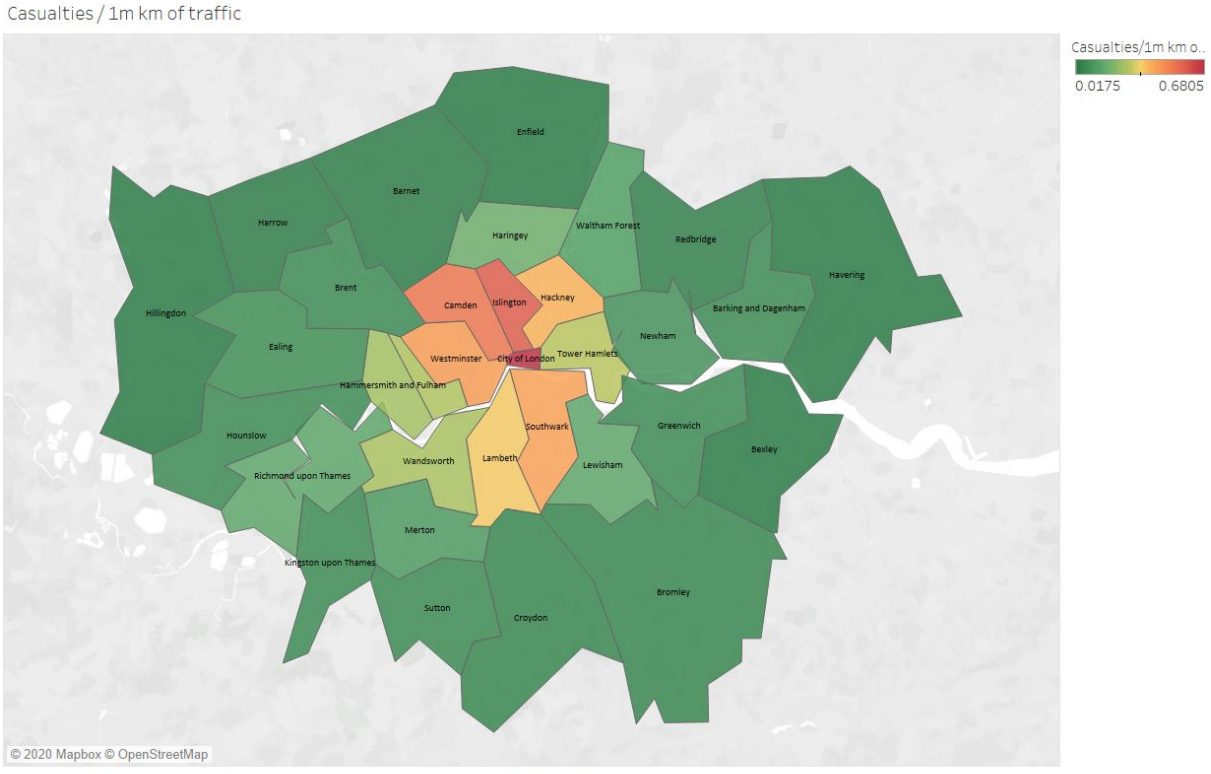
Therefore, we created tables for each variable with either the number of casualties or the rate of casualties using the code above. With these tables, we plotted graphs and included them in our app to allow users to be aware of the conditions they should avoid going out for cycling. The table shown above is illustrating the weather condition with the number of casualties.



This visual illustrates the weather conditions against the number of casualties.



This visual shows the light conditions against the rate of casualties by traffic.



This map shows London local authorities by the rate of casualties by traffic with appropriate colour codes.

6.5. References

[1] Glean A. Cycling Statistics - How popular is cycling in the UK? | 2020 - Finder UK [Internet]. Finder UK. 2020 [Accessed 30 November 2020]. Available from: https://www.finder.com/uk/cycling-statistics

[2]Valentino S. London the worst city in Europe for health costs from air pollution [Internet]. the Guardian. 2020 [Accessed 1 December 2020]. Available from: https://www.theguardian.com/environment/2020/oct/21/london-the-worst-city-in-europe-for-health-costs-from-air-pollution

[3]n.a Best cycling apps: iPhone and Android tools for cyclists - Cycling Weekly [Internet]. Cycling Weekly. 2020 [Accessed 30 November 2020]. Available from: https://www.cyclingweekly.com/group-tests/best-cycling-apps-143222

[4] Making cycling in London safer [Internet]. London Assembly. 2020 [Accessed 1 December 2020]. Available from: https://www.london.gov.uk/what-we-do/transport/cycling-and-walking/making-cycling-london-safer

[5]Coronavirus: Boom time for bikes as virus changes lifestyles [Internet]. BBC News. 2020 [Accessed 1 December 2020]. Available from: https://www.bbc.com/news/business-52564351

[6] £2 billion package to create a new era for cycling and walking [Internet]. GOV.UK. 2020 [Accessed 1 December 2020]. Available from: https://www.gov.uk/government/news/2-billion-package-to-create-new-era-for-cycling-and-walking

[7] Coronavirus: Push for cycling despite safety fears [Internet]. BBC News. 2020 [Accessed 1 December 2020]. Available from: https://www.bbc.com/news/uk-52708687

[8] Priday R. London's cycling boom is slowing. And it's still mostly white men [Internet]. WIRED UK. 2020 [Accessed 1 December 2020]. Available from: https://www.wired.co.uk/article/cycling-london-city-centre-bikes-cycle-superhighway#:~:text=And%20it's%20still%20mostly%20white%20men,-Analysis%20suggests%20that&text=In%202016%2C%20Londoners%20made%20730%2C000,And%20that%20number%20keeps%20rising.

[9]Here are the 10 fastest-growing smartphone apps [Internet]. Business Insider. 2020 [Accessed 1 December 2020]. Available from: https://www.businessinsider.com/fastest-growing-smartphone-apps-comscore-2017-8#1-waze-10 https://en.wikipedia.org/wiki/Moovit

[10]London Cyclists [Internet]. Surveymonkey.co.uk. 2020 [Accessed 1 December 2020]. Available from: https://www.surveymonkey.co.uk/r/TQ625LM

References for Data Sets Used

2,4,5,11 personalised reports:

Road traffic statistics [Internet]. 2020 [cited 1 December 2020]. Available from: https://roadtraffic.dft.gov.uk/custom-downloads/road-accidents/reports/01c891e9-197b-4716-bebf-ef8702bbd8a4

10 ras1:

Reported road accidents (RAS10) [Internet]. GOV.UK. 2020 [Accessed 1 December 2020]. Available from: https://www.gov.uk/government/statistical-data-sets/ras10-reported-road-accident

3,8,9 ras3:

Casualties involved in reported road accidents (RAS30) [Internet]. GOV.UK. 2020 [Accessed 1 December 2020]. Available from: https://www.gov.uk/government/statistical-data-sets/ras30-reported-casualties-in-road-accidents

1,6,7 TRA:

Road traffic statistics (TRA) [Internet]. GOV.UK. 2020 [Accessed 1 December 2020]. Available from: https://www.gov.uk/government/statistical-data-sets/road-traffic-statistics-tra

Other:

Reported Road Casualties Great Britain: Annual Report 2013. [internet] GOV.UK. 2013. [Accessed 19 November 2020]. Available from: https://www.gov.uk/government/statistics/reported-road-casualties-great-britain-annual-report-2013

GB Road Traffic Counts. [internet] Data.Gov.Uk. 2020. [Accessed 18 November 2020]. Available from: https://data.gov.uk/dataset/208c0e7b-353f-4e2d-8b7a-1a7118467acc/gb-road-traffic-counts

Road Safety Data [internet] Data.gov.uk. 2020. [Accessed 18 November 2020]. Available from: https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data

Road traffic statistics. 2020. Reported Road Casualties, Great Britain. [internet] Department for Transport. 2020. [Accessed 19 November 2020]. Available from: https://roadtraffic.dft.gov.uk/custom-downloads/road-accidents/reports/01c891e9-197b-4716-bebf-ef8702bbd8a4

Mock ups images links:

Home Design. Architectural Planning Permission Applications – Authorities Covered. [internet] 2019. [Accessed 1 December 2020] . Available from: https://planningapplications.com/local-authorities-areas-covered/

Wikimedia Commons. A blurry London Skyline. [internet] 2011. [Accessed 1 December 2020]. Available from: https://commons.wikimedia.org/wiki/File:BlurryLondonSkyline.jpg

Conor Bond. 24 Clickable Banner Ads Examples (And Why They Work!). [internet] 2020. [Accessed 1 December 2020]. Available from: https://www.wordstream.com/blog/ws/2020/03/02/banner-ads-example