Task 1
Data Visualization application and principles and visual analytics for exploratory data analysis

VISUALISATION PRINCIPLES AND VISUAL ANALYTICS

FABIO PALLIPARAMBIL

SUBMITTED ON: 15/01/2021

Table of Contents

1.0 Introduction

In this assignment, we will explore the dataset which is provided by Kaggle(1), The **UK Car Accidents 2005-2015** dataset will be used to experiment data visualization techniques and acquire insights from the data.

2.0 Part1 - Data Exploration

2.1 Data Exploration

The given problem has 3 datasets. They are **Accidents05015, Casualities0515** and **Vehicles0515**. In this scenario, we are focusing mainly on Accident05015. The data set **Accidents05015** has 32 features and 1780653 instances.

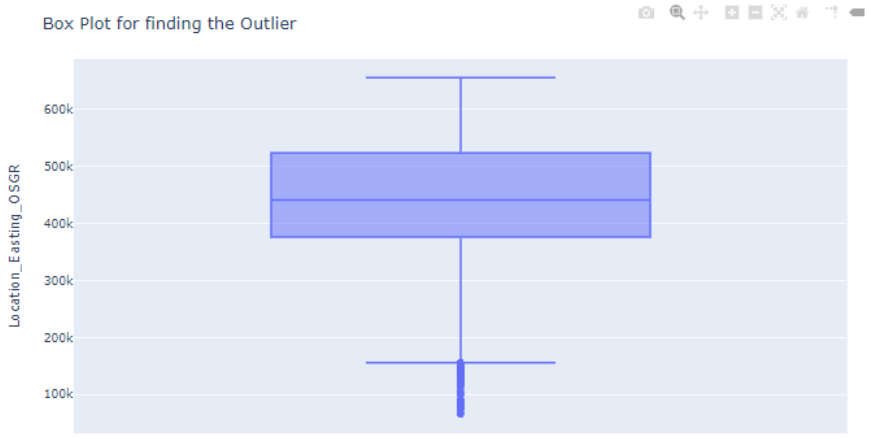
	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude
0	200501BS00001	525680.0	178240.0	-0.191170	51.489096
1	200501BS00002	524170.0	181650.0	-0.211708	51.520075
2	200501BS00003	524520.0	182240.0	-0.206458	51.525301
3	200501BS00004	526900.0	177530.0	-0.173862	51.482442
4	200501BS00005	528060.0	179040.0	-0.156618	51.495752
...	...	...	...	...	...
1780648	2015984139115	312087.0	570791.0	-3.376671	55.023855
1780649	2015984139715	320671.0	569791.0	-3.242159	55.016316
1780650	2015984140215	311731.0	586343.0	-3.387067	55.163502
1780651	2015984140515	328273.0	570137.0	-3.123385	55.020580
1780652	2015984141415	314050.0	579638.0	-3.348646	55.103676

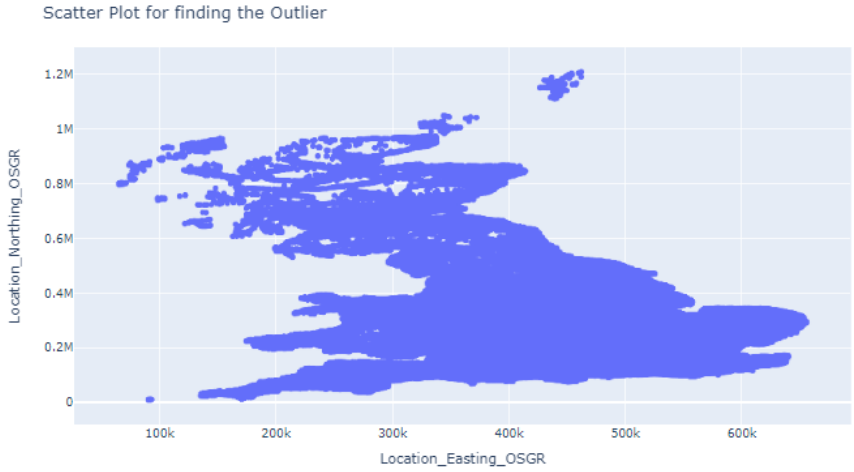
1780653 rows × 32 columns

Figure 1 A snip shot of the Features and number of instances in Accidents05015 dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1780653 entries, 0 to 1780652
Data columns (total 32 columns):
 #   Column                                       Dtype
---  ------                                       -----
 0   Accident_Index                               object
 1   Location_Easting_OSGR                        float64
 2   Location_Northing_OSGR                       float64
 3   Longitude                                    float64
 4   Latitude                                     float64
 5   Police_Force                                 int64
 6   Accident_Severity                            int64
 7   Number_of_Vehicles                           int64
 8   Number_of_Casualties                         int64
 9   Date                                         object
 10  Day_of_Week                                  int64
 11  Time                                         object
 12  Local_Authority_(District)                   int64
 13  Local_Authority_(Highway)                    object
 14  1st_Road_Class                               int64
 15  1st_Road_Number                              int64
 16  Road_Type                                    int64
 17  Speed_limit                                  int64
 18  Junction_Detail                              int64
 19  Junction_Control                             int64
 20  2nd_Road_Class                               int64
 21  2nd_Road_Number                              int64
 22  Pedestrian_Crossing-Human_Control            int64
 23  Pedestrian_Crossing-Physical_Facilities      int64
 24  Light_Conditions                             int64
 25  Weather_Conditions                           int64
 26  Road_Surface_Conditions                      int64
 27  Special_Conditions_at_Site                   int64
 28  Carriageway_Hazards                          int64
 29  Urban_or_Rural_Area                          int64
 30  Did_Police_Officer_Attend_Scene_of_Accident  int64
 31  LSOA_of_Accident_Location                    object
dtypes: float64(4), int64(23), object(5)
memory usage: 434.7+ MB
```

Figure 2 The Datatypes in the data set.





2.2 Data-Pre-processing

While exploring data I came across a lot of missing values and I decide to delete all the missing values(Figure: 3) because some of the missing value is just 7 percentage of the whole dataset.

Display the NaN in the DataSet

Accident_Index	0
Location_Easting_OSGR	138
Location_Northing_OSGR	138
Longitude	138
Latitude	138
Police_Force	0
Accident_Severity	0
Number_of_Vehicles	0
Number_of_Casualties	0
Date	0
Day_of_Week	0
Time	151
Local_Authority_(District)	0
Local_Authority_(Highway)	0
1st_Road_Class	0
1st_Road_Number	0
Road_Type	0
Speed_limit	0
Junction_Detail	0
Junction_Control	0
2nd_Road_Class	0
2nd_Road_Number	0
Pedestrian_Crossing-Human_Control	0
Pedestrian_Crossing-Physical_Facilities	0
Light_Conditions	0
Weather_Conditions	0
Road_Surface_Conditions	0
Special_Conditions_at_Site	0
Carriageway_Hazards	0
Urban_or_Rural_Area	0
Did_Police_Officer_Attend_Scene_of_Accident	0
LSOA_of_Accident_Location	129471

dtype: int64

After Dropping the NaN in the DataSet

Accident_Index	0
Location_Easting_OSGR	0
Location_Northing_OSGR	0
Longitude	0
Latitude	0
Police_Force	0
Accident_Severity	0
Number_of_Vehicles	0
Number_of_Casualties	0
Date	0
Day_of_Week	0
Time	0
Local_Authority_(District)	0
Local_Authority_(Highway)	0
1st_Road_Class	0
1st_Road_Number	0
Road_Type	0
Speed_limit	0
Junction_Detail	0
Junction_Control	0
2nd_Road_Class	0
2nd_Road_Number	0
Pedestrian_Crossing-Human_Control	0
Pedestrian_Crossing-Physical_Facilities	0
Light_Conditions	0
Weather_Conditions	0
Road_Surface_Conditions	0
Special_Conditions_at_Site	0
Carriageway_Hazards	0
Urban_or_Rural_Area	0
Did_Police_Officer_Attend_Scene_of_Accident	0
LSOA_of_Accident_Location	0

dtype: int64

Figure 3 Handling the missing values

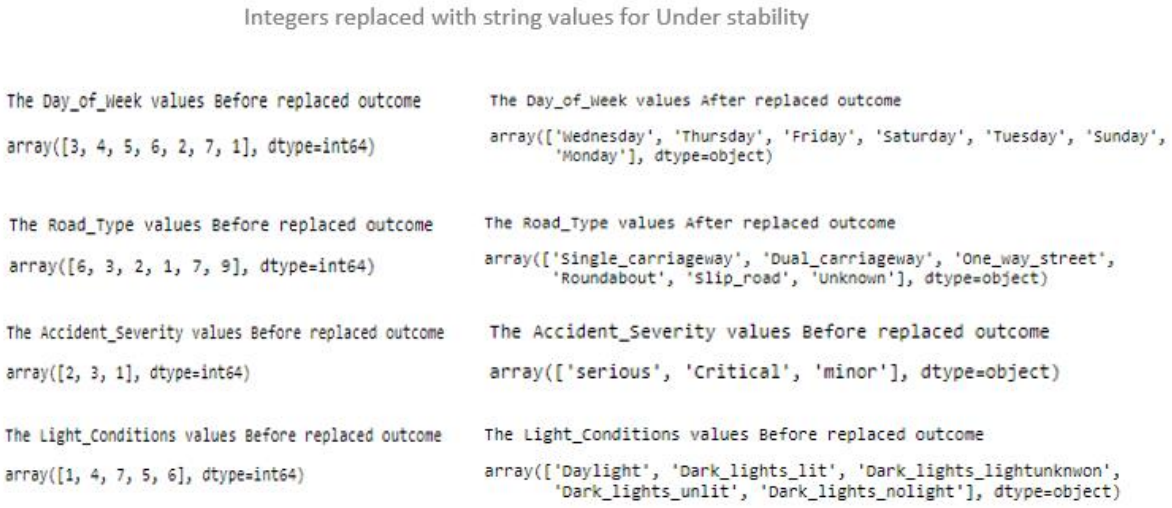
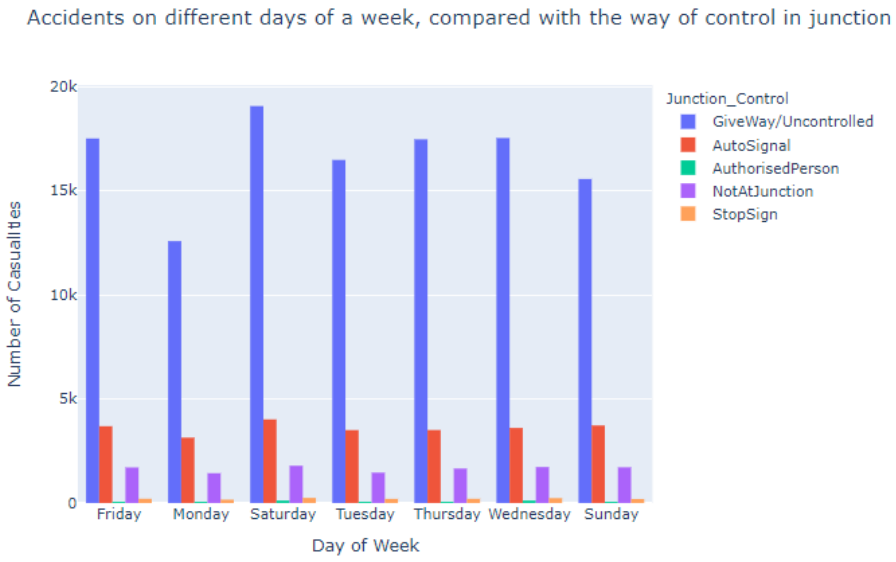


Figure 4 Integers values to String Values for understandability

2.3 Data Visualisation

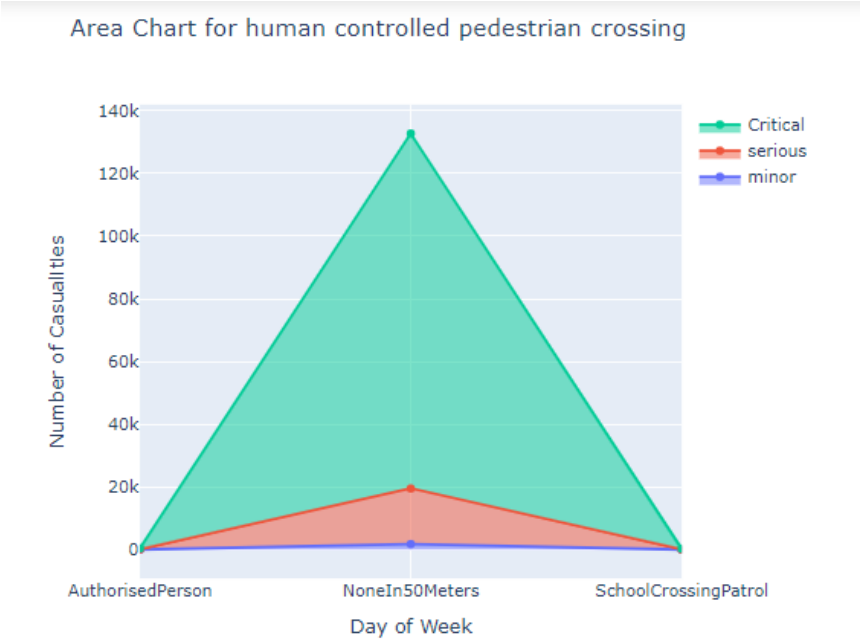
For data visualisation, I have used all the features but only **10 %** of instances from the whole dataset(**178000 instances**). This is done due to high complexity for the computational process.

1. **Distribution (Histogram)**
   In this visualisation, I wanted to find whether there was any relationship between the **number of causalities** and the **day of the week**.



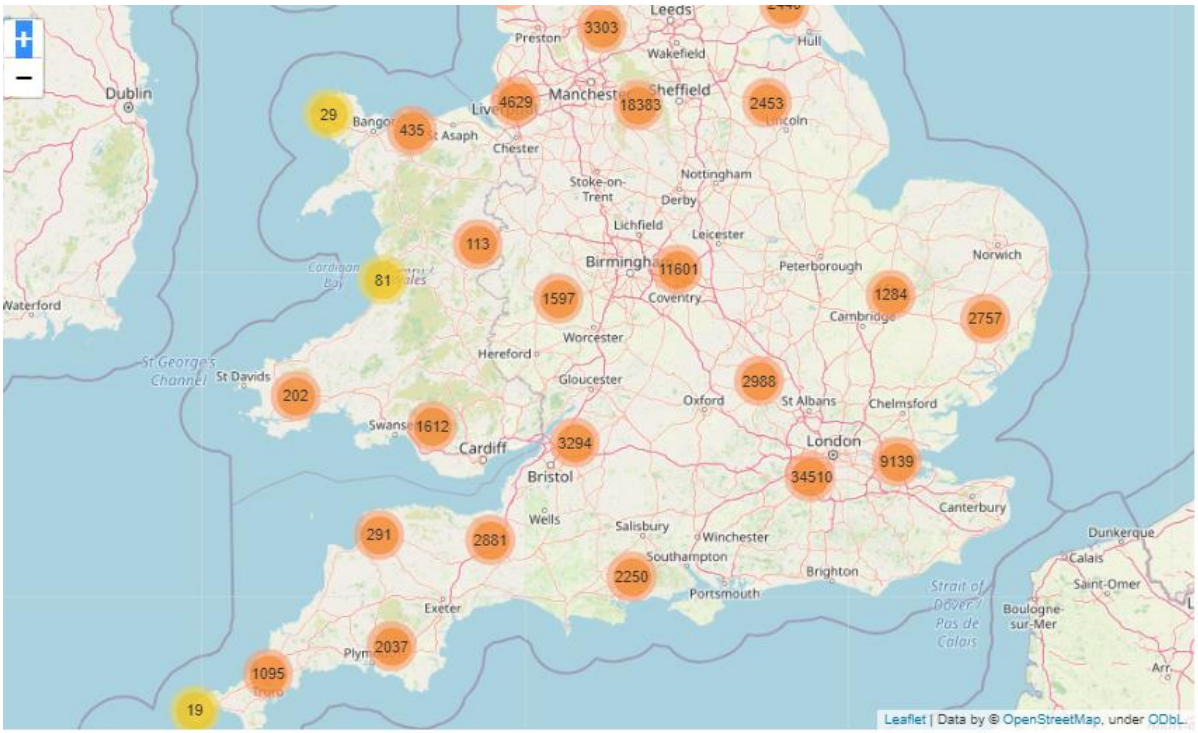
2. **Trend (Area Chart)**

In this visualisation I am trying to find the number of casualties when the pedestrian crossing is controlled human, And what feature has the Highest Number of Casualties compared to accident severity.



3. **Overlays**

In the overlays, I have chosen to visualise the locations of accidents happened. (Figure:). (2)



3.0 Part 2: Exploratory interactive visualisation

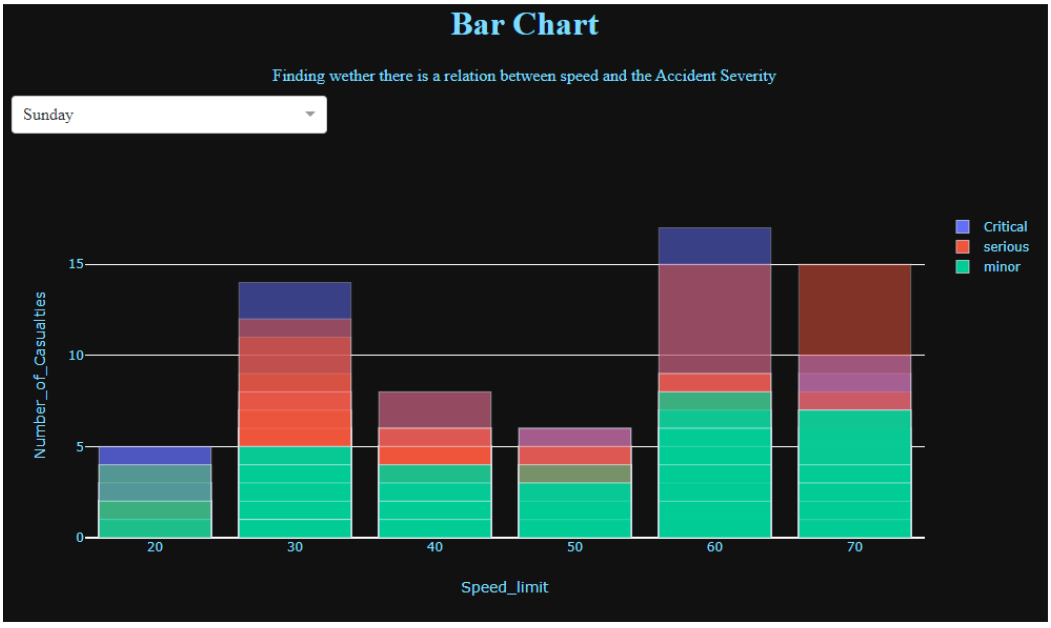
Interactive Visualisation with Dash

1. **Guideline for visualisation**
   I have used **drop downs** and **radio buttons** to filter and interact with the plot more easily. The colours used follows the **HSL colour model**, **Hue** has been mainly used for the plotting and **saturation** for the background because it does look more visually interesting.

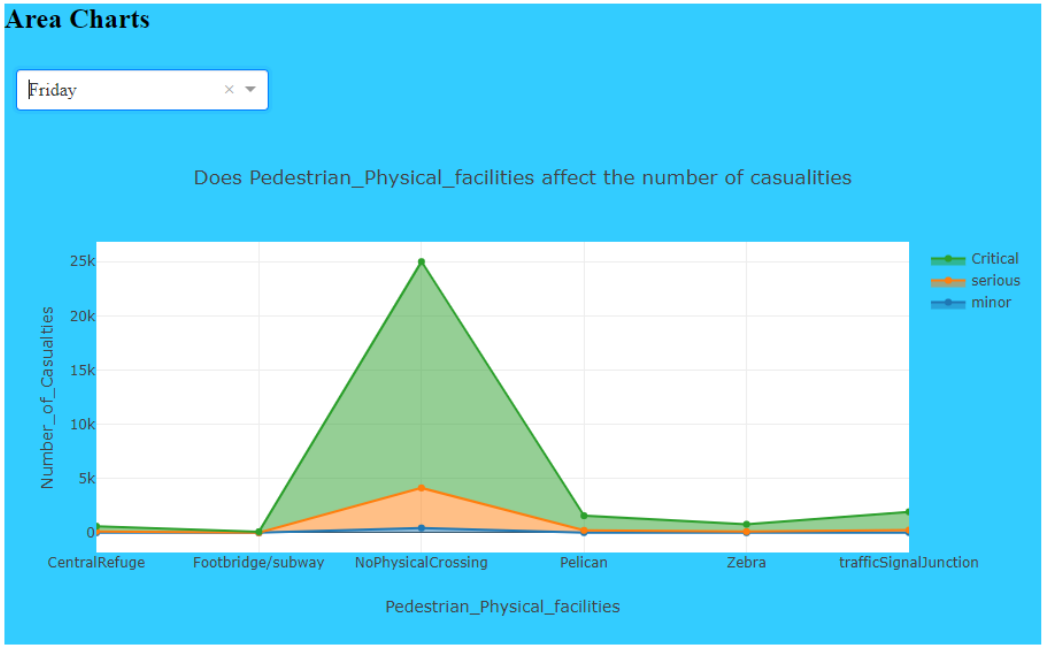
2. **Questions**
   a. which speed limit has an impact on the number of casualties, with related to different days in a week?
   b. Which pedestrian facilities have the highest number of causality and is these accidents severe in different days in a week.
   c. Which weather conditions have the highest and lowest number of causalities. Is the Severity of accident more in urban or rural regions?
   d. What are the different features that cause the number of causalities and no of vehicle?

3. **Interactive Visualisation**

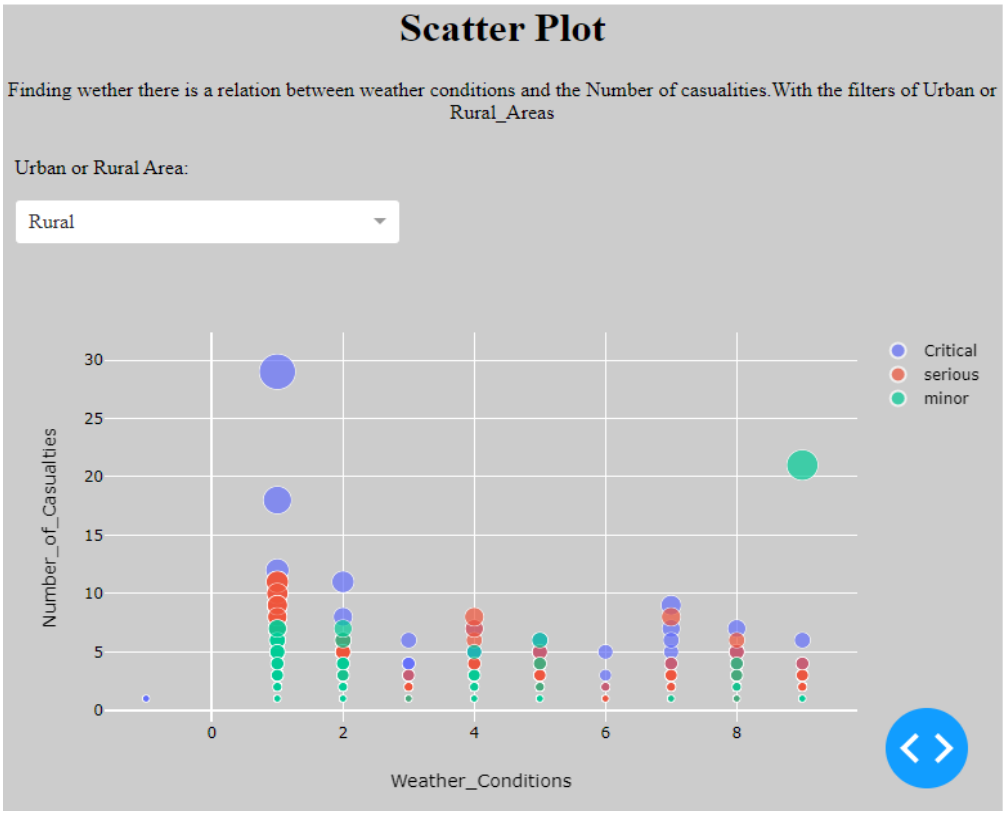
   To answer the question (**a)** I have created a **Bar Chart.** The bar chart is used to compare the number of causalities related to different speed limits, and to find out the accident severity in different speed limits.



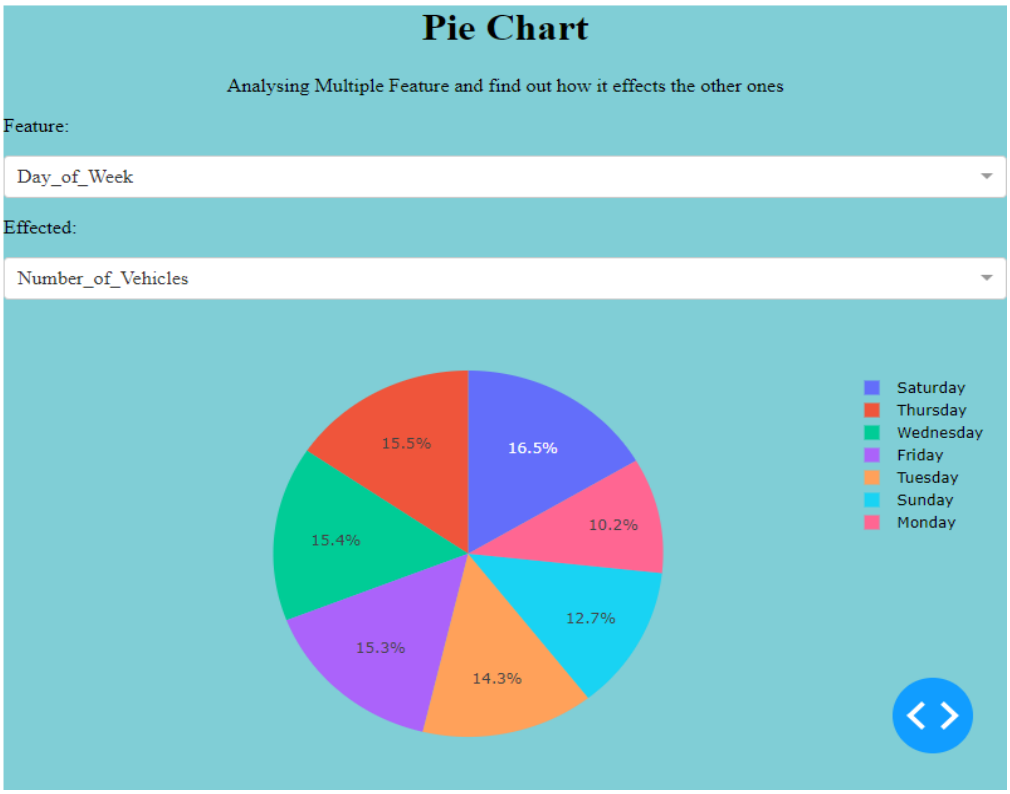
To answer the question (b) I have created **Area Chart.** It is used to analyse the variation in the number of causalities and accident severity related to physical facilities available for pedestrians.



To answer the question (c) I have created a scatter plot. It is used to find out whether weather condition has a role in the number of causalities and accident severity.



To answer the question (d) I have created a Pie plot. It is used to find which values in different features are causing the number of casualties and the number of the vehicle.



4.0 Reference

1. Kaggle.com, UK Car Accidents 2005-2015 (data from UK department for Transport) , 2017 available at[Online]: https://www.kaggle.com/silicon99/dft-accident-data
2. Folium, Folium QuickStart , available at[Online] : https://python-visualization.github.io/folium/quickstart.html