

# **TIME SERIES ANALYSIS ON LA TRAFFIC COLLISION DATA**

*A project report submitted to ICT Academy of Kerala  
in partial fulfillment of the requirements  
for the certification of*

## **CERTIFIED SPECIALIST IN DATA SCIENCE & ANALYTICS**

Submitted by :

**Mehbooba C**

**Sooraj S**

**R Kannan Pillai**

**Naheeda Kallan**



**ICT ACADEMY OF KERALA**  
THIRUVANANTHAPURAM, KERALA, INDIA  
January 2021

# LIST OF FIGURES

SL NO	FIG NO	FIG NAME	Page No
1	4.1	Dataset details from LA website	9
2	4.2	Dataset details from python information function	10
3	4.3	Time occurred details	11
4	4.4	Victim age details	12
5	4.5	Victim sex column unique values	13
6	4.6	Victim descent column unique values	13
7	5.1	Unique values obtained after extraction of day, month and year.	15
8	5.2	Time Taken to report accident column description	15
9	5.3	Date column after combining date and time of collision occurrence	16
10	5.4	victim sex column unique value counts	17
11	5.5	unique values in victim sex after mapping	17
12	5.6	Area details derived from the data set	18
13	5.7	premise details derived from the data set	19
14	5.8	count of MO codes output	20
15	5.9	latitude and longitude derived from location column	20
16	6.1.1	Age column	21

17	6.1.2	Area column	22
18	6.1.3	victim descent column	23
19	6.1.4	year column	23
20	6.1.5	month column	24
21	6.1.6	weekday column	24
22	6.1.7	victim sex column	25
23	6.1.8	hour column	25
24	6.2.1	Age vs weekday	27
25	6.2.2	age vs hourly	27
26	6.2.3	age vs monthly	27
27	6.2.4	age vs yearly	28
28	6.2.5	age vs premises	29
29	6.2.6	age vs area	29
30	6.2.7	age vs victim descent	30
31	6.2.8	age vs sex	31
32	6.2.2.1	Area vs weekday	32
33	6.2.2.2	area vs monthly	33

34	6.2.2.3	area vs hourly	34
35	6.2.2.4	area vs victim descent	35
36	6.2.3.1	descent vs weekly	35
37	6.2.3.2	descent vs monthly	36
38	6.2.3.3	descent vs premises-	37
39	6.2.3.4	descent vs age	38
40	6.2.3.5	descent vs sex	39
41	6.2.4.1	year vs weekly	40
42	6.2.4.2	year vs monthly	40
43	6.2.4.3	year vs age	41
44	6.2.4.4	year vs victim descent	42
45	6.2.4.5	year vs victim sex	43
46	6.2.5	hour vs sex	44
47	7.1	derived target column	45
48	7.2	descriptive statistics of target column	45
49	8.1	distribution plot of target column	46
50	8.2	Distribution plot of target column	47

51	8.3	Time series plot of no.of collisions	47
52	8.4	Decomposition of time series data	48
53	8.5	Autocorrelation plot of target columns	49
54	8.6	Lag plot of target column	50
55	8.6.1	acf plot of target columns	50
56	8.7	pacf plot of target column	51
57	9.1	Dickey Fuller test result on target column	52
58	9.2	Dickey fuller test result on log transformation of target column	53
59	9.3	plot on log transformation of target column	54
60	9.4	Decomposition plot on log transformation of target column	54
61	9.5	plot on first differencing of log transformation of target column	55
62	9.6	Decomposition plot on first differencing of log transformation of target column-	58
63	9.7	Dickey fuller test result on seasonal differenced data	
64	9.8	plot on first differencing of log transformation of target column	
65	9.9	Decomposition plot on first differencing of log transformation of target column	
66	9.10	Acf of stationary data	
67	9.11	pacf of stationary data	

68	10.1	grid search result for arima model	
69	10.2	Prediction and original values for arima model	
70	10.3	Grid search results for seasonal arima model	
71	10.4	Forecast and original no.of collisions for seasonal arima model	
72	10.5	Forecast on no:of collisions for FB prophet model	
73			
74			


## TABLE OF CONTENTS

CHAPTER NO	CHAPTERS	Page No
	ABSTRACT	4
1	PROBLEM DEFINITION	5
2	INTRODUCTION	7
3	PROPOSED METHOD	8
4	DATA UNDERSTANDING	9
5	DATA CLEANING FOR EDA	14
6	EXPLORATORY DATA ANALYSIS	19
7	DATA PREPROCESSING	42
8	UNDERSTAND TIME SERIES DATA	44

9	MAKING TIME SERIES DATA STATIONARY	51
10	MODELLING AND EVALUATION	58
11	WEB DEPLOYMENT	69
12	CONCLUSION	70
	REFERENCES	71

## **ABSTRACT**

Road accidents are one of the most relevant causes of injuries and death worldwide, and therefore, they constitute a significant field of research on the use of advanced algorithms and techniques to analyze and predict traffic accidents and determine the most relevant elements that contribute to road accidents. The dataset here reflects traffic collision incidents in the City of Los Angeles dating back to 2010. This data is obtained from a traffic collision data set maintained by the city of Los Angeles. While it doesn't directly measure traffic, it measures a closely-related proxy. The data set contains 18 columns which includes dates (reported, occurred), time, area details, crime code, MO code, victim details (age, sex, descent) and location details.

The purpose of the project is to do exploratory data analysis in Tableau and predict the number of collisions that will occur per month based on time series analysis. The predictive analysis with the time series models is implemented using python for the traffic collision data set. The process of modeling, as well as results, are interpreted using root mean square error and AIC value. The major contribution of this project is to give a clear view of how the traffic collision data set can be used to generate a more secure mobility environment, and ultimately, save lives.



# **CHAPTER 1: PROBLEM DEFINITION**

## **1.1 Project Overview**

Traffic is an issue that's familiar to pretty much everyone. While traffic collision doesn't directly measure traffic, it measures a closely-related proxy. It's not a stretch to hypothesize that more traffic correlates with more collisions which directly cause more traffic. A number of factors contribute to the risk of collisions, including vehicle design, speed of operation, road design, weather, road environment, driving skills, impairment due to alcohol or drugs, and behavior, notably aggressive driving, distracted driving, speeding and street racing. There may be some hidden pattern in the time of occurrence of collision apart from these factors. This hidden pattern can be extracted by applying a time series analysis.

## **1.2 Problem Statement**

This is a univariate time series analysis problem, which is used to predict the no. of monthly traffic collisions in Los Angeles. Here we need to develop the target column, which is the no. of collisions in Los Angeles from the available data set. The ideal outcome is to provide information on how much collisions is expected to occur.

### **1.3 Objectives**

The objective of this project is to implement a time series model to predict the no. of monthly collisions in Los Angeles. Further more, this involves exploratory data analysis to understand the hidden patterns in time, victim age, victim descent, victim sex, area and premise. This also involves analysing the traffic collision patterns vary by time of day, day of week, and time of year. As an application of this project, the results can be used to implement countermeasures to reduce the no. of collisions.

### **1.4 Domain Knowledge**

Los Angeles , city, seat of Los Angeles county, southern California, U.S. It is the second most populous city and metropolitan area (after New York City) in the United States. The lifestyle of Los Angeles residents (who are called Angelenos) relies on the automobile, idealizes the single-family dwelling, and favours informality. Los Angeles is a place of extraordinary ethnic and racial diversity, owing largely to immigration, and, like other world cities, it reflects a growing gap between rich and poor. Los Angeles is ranked number one city in the US with major problems in traffic collisions.

In recent years, frequent traffic accidents have become an important factor threatening people's travel safety. The frequent occurrence of traffic accidents has always been an important problem troubling traffic safety management, so exploring the law and characteristics of case occurrence in a space area has profound significance for the prevention of traffic accidents. Its high data brings unprecedented challenges to the public security organs, especially in the situation of constantly changing the traffic space environment, and the case space of different types of accidents usually shows different rules and characteristics. Based on real traffic accident data and machine learning technology, analyzing traffic accident cases from the perspective of time and space can reveal the distribution law and the causes behind traffic accidents in a scientific and profound way, so as to formulate different prevention strategies according to different types of traffic accidents and make relevant departments respond to traffic accidents in a more directional and targeted way.

Time domain is the most commonly used research in the study of traffic accidents. Time domain analysis has the advantages of strong time localization ability and strong interpretability, but it is unable to obtain more information about the change of time series, so some researches attempt to explore in frequency domain analysis. Although frequency domain has the function of accurate frequency positioning, it is only suitable for stationary time series analysis. However, with the change of time, traffic accidents are often subject to the comprehensive influence of a variety of factors, most of which are non-stationary sequences. Generally, they not only have the

characteristics of trend and periodicity, but also have the problems of mutability, randomness, and “multi-time scale” structure, showing a multi-level evolution rule. For the study of such non-stationary time series, the corresponding time information of a certain frequency band or the frequency domain information of a certain period are usually required. Obviously, simple time domain analysis and frequency domain analysis are obviously weak.

## **CHAPTER 2: INTRODUCTION**

It is known that US Governments turn to Advanced Traffic Management Systems in order to solve traffic congestion and adopt new transport management plans and utilize transport resources. Unfortunately, major cities are still waiting for traffic to be resolved, where Los Angeles is ranked number one city in the US with major problems in it. City Departments are interested in improving traffic situations and therefore adopt information from popular navigation platforms in order to understand and analyze current situations.

In this project we have conducted analysis of traffic collisions in the Los Angeles County area. This data begins in January 2010 and is updated weekly. In this particular project, we use data from January 2010 - January 2021, which ends up being ~551K rows. Each row corresponds to a collision. Although traffic analysis attracts enough attention from researchers and predicting traffic patterns is a goal for major businesses. We used different interactive visuals in order to show traffic collision events clearly as time dependent patterns and different sliced information. After data cleaning and preparation for further analysis, files were extracted into Tableau for better visualization. Here we are analyzing how collision patterns vary by time of day, day of the week, and time of year.

A time series is a collection of observations of well-defined data items obtained through repeated measurements over time. For example, measuring the value of retail sales each month of the year would comprise a time series. This is because sales revenue is well defined, and consistently measured at equally spaced intervals. Data collected irregularly or only once are not time series. An observed time series can be decomposed into three components: the trend (long term direction), the seasonal (systematic, calendar related movements) and the irregular (unsystematic, short term fluctuations). Time series analysis accounts for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for.

## CHAPTER 3: PROPOSED METHOD

The proposed system has data which has collision details in Los Angeles. This proposed system can use this data to understand the collision patterns and predict no. of monthly collisions using time series analysis. The following steps were adopted for the exploratory data analysis and prediction modelling purpose.

- 1. Data understanding :** From the data we are having, it should be converted into the desired format for creating time series models to predict no. of collisions. For this, first we need to understand the data fully. That is we need to address the questions: what each column represents, what is data type, is there any problem in data type etc. During this step, we got a clear idea of the data.
- 2. Data cleaning for EDA :** For doing EDA, the data must be cleaned. The data is transcribed from original paper traffic reports, so it's very likely that there are errors. Before doing the EDA in Tableau, cleaning was necessary.

- 3. Exploratory Data Analysis :** It is the process of summarizing the main characteristics of a data set to understand what the data can tell us beyond the formal modelling task. Here we conducted univariate and bivariate analysis to understand the data. Numerous plots were implemented in Tableau
- 4. Data preprocessing for time series modelling :** Here the data contains many features apart from the time of collision. For univariate time series analysis, only the time column and the count of collisions is required. For this purpose data was processed to create the target column(No.of monthly collisions).
- 5. Understanding time series data :** Before doing any time series analysis, understanding time series components is necessary. Therefore, time series plots were derived from the processed data. This helps us to apply the right algorithm for the time series modelling.
- 6. Making time series data stationary :** The time series data should be stationary while applying in a time series model. Stationarity can be checked using Dickey Fuller test. In this step, data was checked for stationarity and appropriate techniques were applied to make the data stationary
- 7. Modelling and evaluation:** As part of modelling, ARIMA, seasonal ARIMA and FB prophet models were applied on the data set. The models were evaluated using AIC and RMSE.
- 8. Web deployment of time series model :** After selecting the best model, a web app is created using flask app and model is deployed. During this step, we also tried to host the website using pythonanywhere.com.

## CHAPTER 4: DATA UNDERSTANDING

The dataset is taken from the open data portal city of Los Angeles. This data set (and a bunch of others) is actively maintained by the city of Los Angeles and is freely available to the public. The dataset reflects traffic collision incidents in the City of Los Angeles dating back to 2010. This data is transcribed from original paper traffic reports, so it's very likely that there are errors. The data begins in January 2010 and is updated weekly. In this particular project, we use data from January 2010 - January 2021, which ends up being ~551K rows. Each row corresponds to a collision. The data possess 18 columns.

Dataset link:

<https://drive.google.com/file/d/1vv4wBQrzLdEl2MoRGcmnTjKMA6FldVGs/view?usp=sharing>

About this Dataset

Updated

January 12, 2021

Data Last Updated

January 12, 2021

Metadata Last Updated

November 30, 2020

Date Created

June 8, 2017

Views

17.4K

Downloads

5,516

Data Provided by

Los Angeles Police Department

Dataset Owner

LAPD OpenData

Contact Dataset Owner

Data Owner

Department

LAPD

Committed Update Frequency

Refresh rate

Weekly

Location Specified

Does this data have a Location column? (Yes or No)

Yes

Attachments

MO\_CODES\_Numerical\_20180627.pdf

Topics

Category

Public Safety

Tags

lapd, traffic, traffic data, police, safe city, traffic collision

Licensing and Attribution

License




Figure 4.1 Dataset details from LA website

The explanation of these features are as follows:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 551105 entries, 0 to 551104
Data columns (total 18 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   DR Number                            551105 non-null  int64
1   Date Reported                        551105 non-null  datetime64[ns]
2   Date Occurred                       551105 non-null  datetime64[ns]
3   Time Occurred                       551105 non-null  int64
4   Area ID                             551105 non-null  int64
5   Area Name                           551105 non-null  object
6   Reporting District                  551105 non-null  int64
7   Crime Code                          551105 non-null  int64
8   Crime Code Description               551105 non-null  object
9   MO Codes                            462919 non-null  object
10  Victim Age                          469462 non-null  float64
11  Victim Sex                          542439 non-null  object
12  Victim Descent                      541577 non-null  object
13  Premise Code                        550138 non-null  float64
14  Premise Description                 550137 non-null  object
15  Address                             551105 non-null  object
16  Cross Street                       524726 non-null  object
17  Location                            551105 non-null  object
dtypes: datetime64[ns](2), float64(2), int64(5), object(9)
memory usage: 75.7+ MB

```

Figure 4.2 Dataset details from python information function

### 4.1. DR Number

DR number means Division of Records Number, which is an official file number made up of a 2 digit year, area ID, and 5 digits. Its data type is integer as this value is a number. Each number is identical for a collision record. There are 551105 unique values in this column and no missing values.

### 4.2. Date Reported

This column indicates the date in which a collision is reported and it is given in MM/DD/YYYY format. There are no missing values in this column. Here the data type is date & time. There are 4027 reported days.

### 4.3. Date Occurred

This column indicates the date in which a collision is occurred and it is given in MM/DD/YYYY format. There are no missing values in this column. Here the data type is date & time. There are 4027 distinct days.

## 4.4. Time Occurred

This is the time at which the particular collision is occurred and it is given in 24 hr military time. There are no missing values in this column. This column has values from 1 to 2359. This can be considered as 00:01 to 23:59.

```
count    551105.000000
mean      1357.158881
std        596.391528
min         1.000000
25%        930.000000
50%       1430.000000
75%       1820.000000
max       2359.000000
Name: Time Occurred, dtype: float64
```

Figure 4.3 Time occurred details

## 4.5. Area ID

The Los Angeles Police Department (LAPD) has 21 Community Police Stations referred to as Geographic Areas within the department. These Geographic Areas are sequentially numbered from 1-21. These numbers are given in the “Area ID” column. It’s data type is integer and it is a nominal data as it simply represents an area. There are no missing values in this column.

## 4.6. Area Name

The 21 Geographic Areas or Patrol Divisions are also given a name designation that references a landmark or the surrounding community that it is responsible for. It is a nominal data and its data type is object. For example 77th Street Division is located at the intersection of South Broadway and 77th Street, serving neighborhoods in South Los Angeles. Each area name has a corresponding area ID.

## 4.7. Reporting District

A code used in producing reports to group data into geographic sub-areas within an area. Even though it is a nominal data, this is represented as an integer in the reporting district column. There are 1332 reporting districts in this data set.



## 4.8. Crime Code

Crime code indicates the crime committed. For this dataset, all Crime Code is 997. This is the code corresponding to the crime. There are no null values. Even though the data type is integer, it is a nominal data as it simply represents a crime.

## 4.9. Crime Code Description

This defines the Crime Code provided. For this dataset, all values in this column is Traffic Collision, which is corresponding to the crime code 997. This is a nominal data and its data type is object.

## 4.10. MO Codes

MO code is Modus Operandi code (“Mode of operation”). This is the activity associated with the suspect in commission of the crime. Here each numerical value represents a description associated with the crime as given in the PDF. See attached PDF for list of MO Codes in numerical order. This column has ~16% missing values. Here the data type is object. For a single collision, there are multiple MO Codes associated with it. It is nominal data.

PDF link:

[https://drive.google.com/file/d/1mycSL6UGTNKGgP-gSWTAIrEbM1X\\_h-l/view?usp=sharing](https://drive.google.com/file/d/1mycSL6UGTNKGgP-gSWTAIrEbM1X_h-l/view?usp=sharing)

## 4.11. Victim Age

This column gives the age of the victim and it is a two digit numeric. This is ratio data. Its data type is float. There are ~14.8% missing values in this column. Victims are between the age 10 and 99.

```
count    469462.000000
mean      41.208673
std       16.288216
min       10.000000
25%       28.000000
50%       38.000000
75%       51.000000
max       99.000000
Name: Victim Age, dtype: float64
```

Figure 4.4 Victim age details

## 4.12. Victim Sex

This column represents victim's sex. It is nominal data. Here the data type is object and values are represented in such a way that F for female, M for male and X for unknown. But there are 6 unique values in this column. There are ~1.5% missing values in this column. But in the dataset there are 6 unique values for this column. This column needs to be cleaned.

```
Victim Sex column unique values: ['M' 'F' 'X' nan 'H' 'N']
```

Figure 4.5 Victim sex column unique values

## 4.13. Victim Descent

Descent Code represents the code for the origin or background of a person in terms of family or nationality. Here it is a nominal data and data type is object. This column has the values which is indicated as follows: A - Other Asian, B – Black, C – Chinese, D – Cambodian, F – Filipino, G – Guamanian, H - Hispanic/Latin/Mexican, I - American Indian/Alaskan Native, J – Japanese, K – Korean, L – Laotian, O – Other, P - Pacific Islander, S – Samoan, U – Hawaiian, V – Vietnamese, W – White, X – Unknown, Z - Asian Indian. By analyzing the data set using python, 21 distinct values were found in this column. Here X, nan and '-' indicates same meaning.:

```
Victim Descent column unique values: ['H' 'X' 'B' 'F' 'O' 'W' 'A' nan 'K' 'J' 'P' 'Z' 'C' 'I' 'V' 'D' 'U' 'S' 'L' 'G' '-']
```

Figure 4.6 Victim descent column unique values

## 4.14. Premise Code

This represents the code for the type of structure or location where the incident took place. There are 118 premise codes. This column has a data type as float. This is nominal data as it simply indicates a premise.

## 4.15. Premise Description

This column defines the Premise Code provided in the “Premise Code” column. There are 117 premise descriptions. Premise code and premise description column needs to be compared. This is the nominal data and data type of this column is object.

## 4.16. Address

This column indicates the street address of a crime incident rounded to the nearest hundred block to maintain anonymity. This is nominal data. Data type of this column is object.

#### **4.17. Cross Street**

This gives the cross Street of rounded Address. It is an object data type column. It is nominal data.

#### **4.18. Location**

This column indicates the latitude/longitude coordinates of the location where the crime incident occurred. Actual address is omitted for confidentiality. XY coordinates reflect the nearest 100 blocks. Some location fields with missing data are noted as (0°, 0°). There is also a small number (~740) of collisions that do not have valid latitude/longitude coordinates.

## **CHAPTER 5: DATA CLEANING FOR EDA**

This is the process of detecting and correcting corrupt or inaccurate records from a record-set, table, or database and refers to identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data, and then replacing, modifying, or deleting the dirty or coarse data. For doing EDA, the data must be cleaned. The data is transcribed from original paper traffic reports, so it's very likely that there are errors. Before doing the EDA in Tableau, cleaning was necessary. Therefore, in depth analysis was conducted on the data for the cleaning purpose. The following steps were done as part of the data cleaning process

## 5.1. DATE COLUMN CORRECTION

Here as a first step, year, month and weekday were extracted from the date occurred column.

```
unique values in day column : [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31]
unique values in month column : [ 1  2  3  4  5  6  7  8  9 10 11 12]
unique values in year column : [2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021]
```

figure 5.1. Unique values obtained after extraction of day, month and year.

Then the difference between the reporting date and occurred date is found out. During this process the difference was obtained in nano second time format. This column is then processed accordingly to get the difference as days. By taking unique values in this column, it was found that there are 431 unique values and value ranged from 0 to 3662 days. This means that there are cases in which collision is reported after years.

```
count      551105.000000
mean        2.253946
std         33.913850
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         3662.000000
Name: Time taken to report accident in days, dtype: float64
```

figure 5.2. Time taken to report accident column description

The time occurred column was processed to get hour and minute separately. As part of getting time and date of collision occurrence in the same column, these columns combined.

```

103563    2010-01-01 02:30:00
114542    2010-01-01 14:30:00
118319    2010-01-01 02:50:00
103887    2010-01-01 00:20:00
87292     2010-01-01 19:50:00
...
550771    2021-01-09 05:45:00
550780    2021-01-08 15:00:00
550792    2021-01-09 13:20:00
550977    2021-01-09 06:25:00
550618    2020-12-30 14:15:00
Name: Date Occurred, Length: 551105, dtype: datetime64[ns]

```

figure 5.3. Date column after combining date and time of collision occurrence

## 5.2. VICTIM SEX COLUMN CORRECTION

Earlier, the Victim sex column contained 6 unique values. There were H, N and nan values. Here H can be considered F and N can be considered M . This may be a mistake occurred during the conversion process of data. By considering this fact, we can convert H as F and N as M. Also there are only 0.027% of total values as H and 0.002% as N.

```

M      59.078164
F      38.107879
X       2.784829
H       0.027100
N       0.002028
Name: Victim Sex, dtype: float64

```

figure 5.4. Victim sex column unique value counts.

This column is then cleaned by converting nan as unknown(X). The values H and N were converted to F and M respectively. The values in this column are mapped to get proper values for visualization .

```

array(['Female', 'Male', 'Unknown'], dtype=object)

```

figure 5.5. Unique values in victim sex after mapping.

## 5.3 VICTIM DESCENT COLUMN CORRECTION

The victim Descent column contained '-', nan and X for unknown values. All these were converted to X for further processing. Also this column is mapped to the corresponding values of abbreviations for clear visualization purposes in Tableau.

#### 5.4 AREA ID AND AREA NAME

These columns are processed together to check whether the same area ID indicates different areas. During the process no mistakes were found. This is achieved by considering these columns as a new dataframe, removing duplicate values and then sort by area ID. Thus area ID for a particular area name is extracted from the data and it is obtained as follows:

Area ID	Area Name
1	Central
2	Rampart
3	Southwest
4	Hollenbeck
5	Harbor
6	Hollywood
7	Wilshire
8	West LA
9	Van Nuys
10	West Valley
11	Northeast
12	77th Street
13	Newton
14	Pacific
15	N Hollywood
16	Foothill
17	Devonshire
18	Southeast
19	Mission
20	Olympic
21	Topanga

figure 5.6. Area details derived from the data set

#### 5.5. PREMISE CODE AND PREMISE DESCRIPTION

These columns are also processed like the area columns. These are obtained as shown in the figure. Here while checking for the missing values, two values were found missing in the description column and one in the premise ID column. This caused the difference in the no.of missing values in the data understanding step. Also we checked whether the 418.0 indicates any premise description other than NaN value. No such values were found.

Premise Code		Premise Description
80618	115.0	AIRCRAFT
91959	103.0	ALLEY
254473	208.0	AUTO SALES LOT
285013	408.0	AUTO SUPPLY STORE*
125884	605.0	AUTOMATED TELLER MACHINE (ATM)
...	...	...
96962	122.0	VEHICLE, PASSENGER/TRUCK
152046	213.0	WAREHOUSE
134430	121.0	YARD (RESIDENTIAL/BUSINESS)
117277	NaN	NaN
47720	418.0	NaN

118 rows × 2 columns

figure 5.7 Premise details derived from the data set

## 5.6. CRIME CODE AND CRIME DESCRIPTION

These columns contained only one unique value.997 indicating traffic collision as crime. Therefore, these columns are removed from further processing.

## 5.7. MO CODE COLUMN

MO codes column contained more than one MO codes separated by a space. Each code has 4 digits. This column is processed to find the max string length (49)and then the column is divided into 10 (obtained as  $(49+1)/5$ ). The output was obtained as a dataframe. This is then converted to a numpy array for reshaping purposes. Then the value count was taken after converting the reshaped array into a dataframe. Thus the count of each MO code is obtained as a dataframe.

```

0      2485753
3101    401369
3701    333232
3401    333078
3004    238254
...
431      1
1819     1
1208     1
1207     1
1817     1
Name: 0, Length: 329, dtype: int64

```

figure 5.8 Count of MO codes output

Here 0 indicates missing values. Then MO codes are extracted as a separate CSV file. Zero values can be removed during the visualisation process in Tableau.

## 5.8. LOCATION COLUMN

This column contains the latitude and longitude values as (x,y). The location contains some missing values(740) which were given as (0.0, 0.0). These missing values can be neglected during the visualization process. This column is processed by removing the parentheses and splitting the column into latitude and longitude based on the separator comma(,).

**Latitude   Longitude**

34.0001	-118.3138
34.2306	-118.5885
34.3115	-118.4291
34.0006	-118.2871
34.0305	-118.2193

figure 5.9 Latitude and Longitude derived from location column.



# CHAPTER 6: EXPLORATORY DATA ANALYSIS

In statistics, exploratory data analysis is the process of summarizing the main characteristics of data set to understand what the data can tell us beyond the formal modelling task. With EDA, we can do univariate analysis for understanding the distribution of each feature and bivariate analysis is used for understanding the features relation with target variables.

A count plot can be used to understand the density of underlying distribution of a single numerical or categorical data. Also collision patterns can be found using a bar plot or pie plot. Here EDA is done using Tableau.

## 6.1 UNIVARIATE ANALYSIS

### 6.1.1 AGE COLUMN

From the plot, it is clear that age column data is skewed towards the right. There are hardly any victims below age 15.

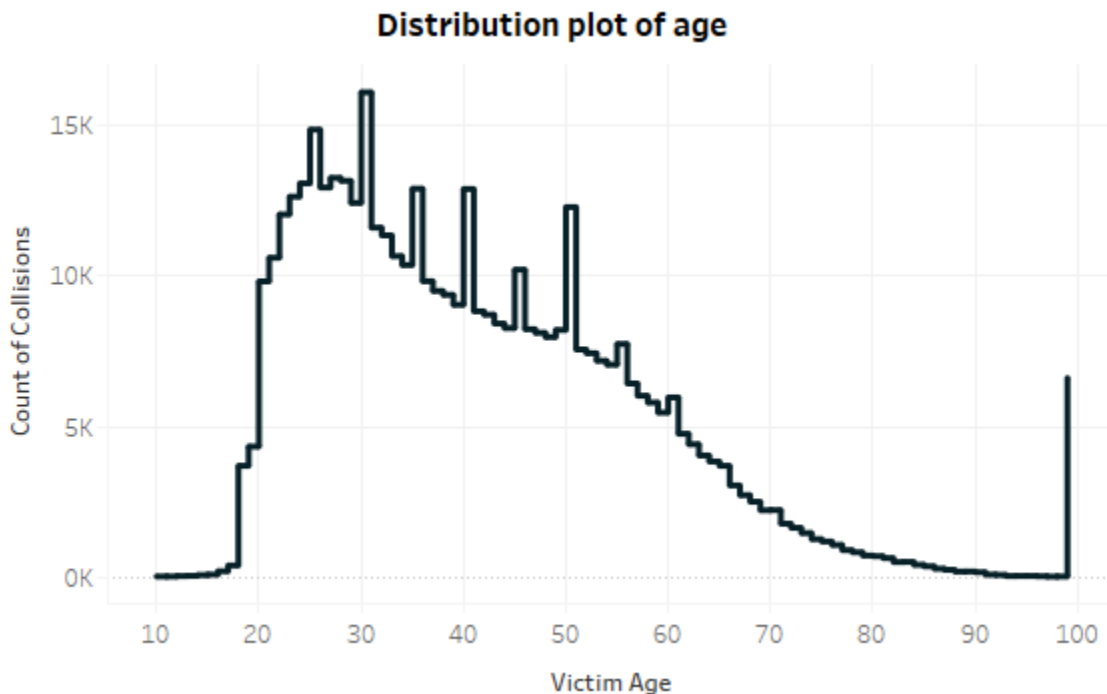


Figure 6.1 Distribution plot of age

Most victims are in their 20s. The number of collision victims per age generally decreases after age 30. There are spikes at most multiples of 5 (25, 30, 35, etc). This suggests that some ages are

estimated and that official identification (such as a driver's license) isn't always used in collision reports. Age 99 seems to be a catch-all bucket. It seems unlikely that there are actually as many age-99 victims as are shown above. This plot also raises questions, how are collisions with multiple victims dealt with and what's going on with the spike at age 99.

## 6.1.2 AREA COLUMN

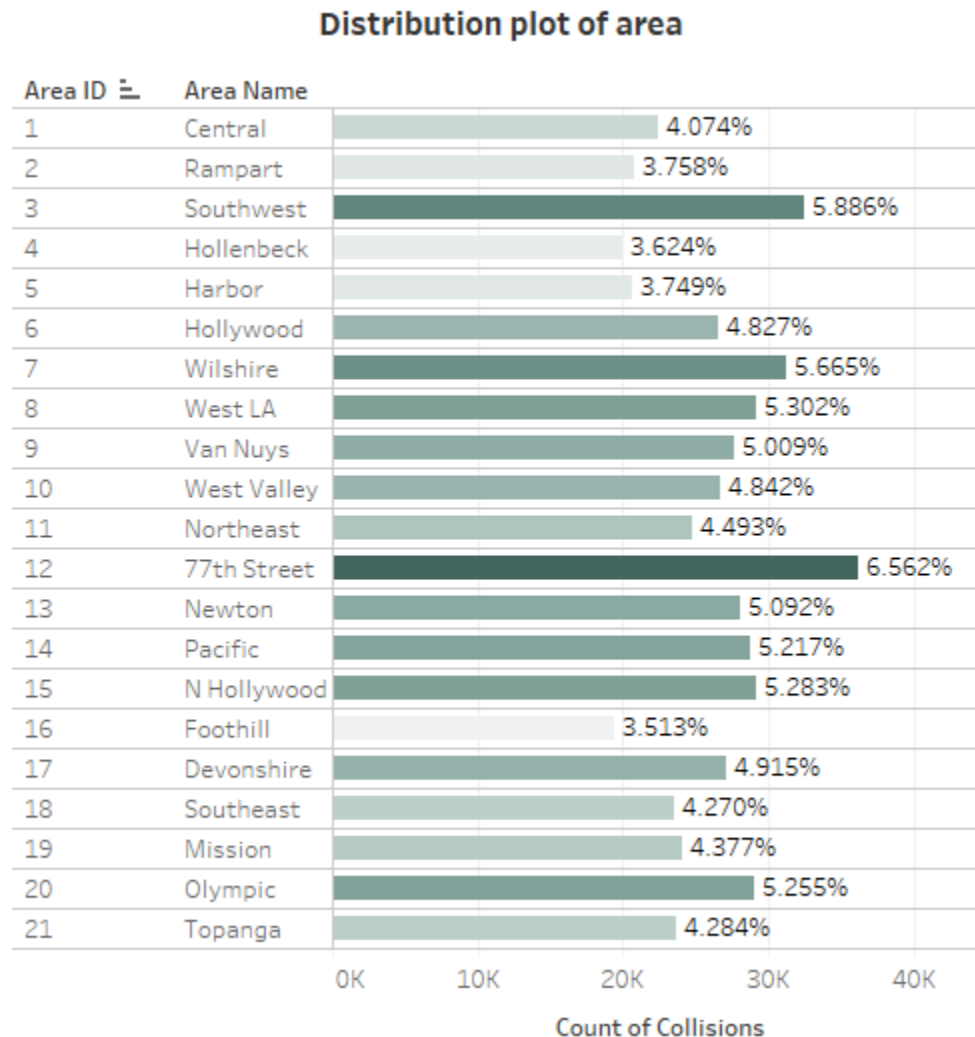


Figure 6.2 Distribution plot of area

From the graph, it is clear that 77th street has most no. of collisions and Foothill area have least no. of collisions. Southwest area also have more no. of collisions. Some areas obviously have more collisions than others. But without additional information such as size or traffic density per area, this graphic isn't too informative.

### 6.1.3 VICTIM DESCENT COLUMN

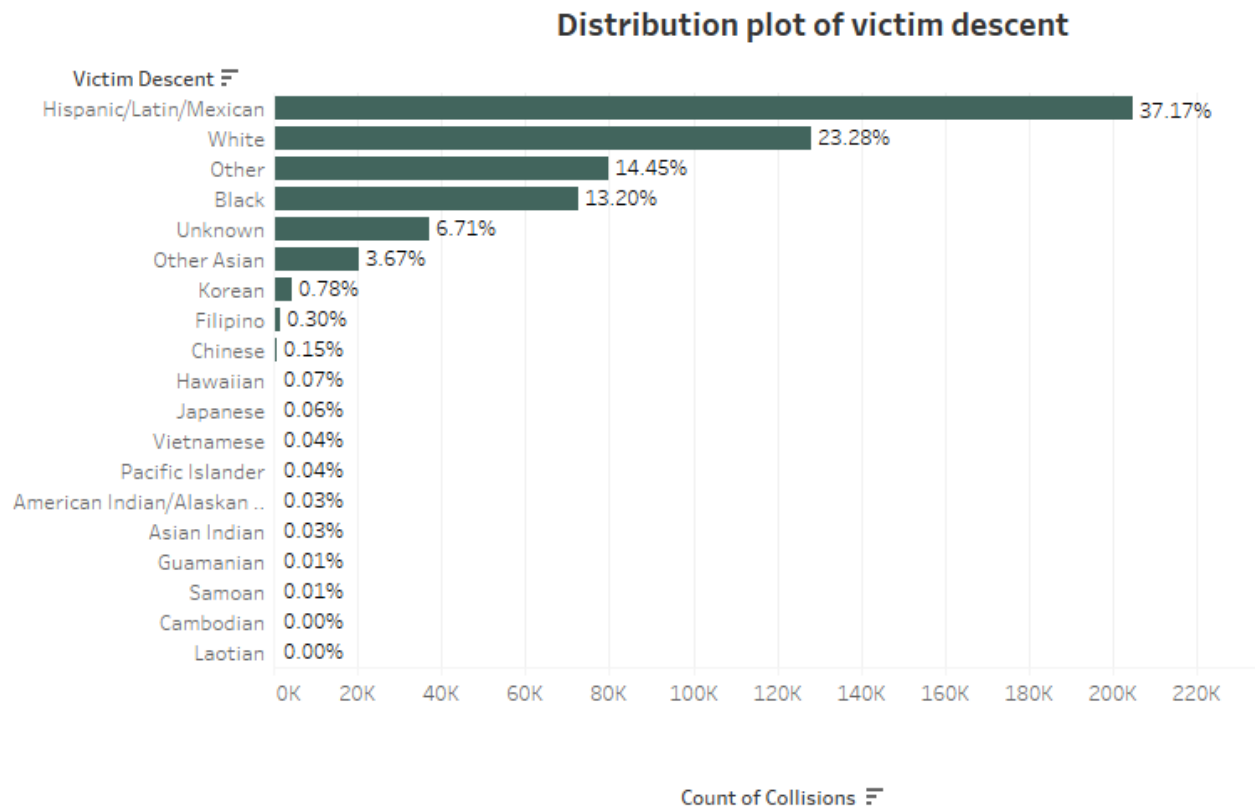


Figure 6.3 Distribution plot of victim descent

Most no. of victims are from H(Hispanic/Latin/Mexican). The white people are also more in the no. of victims. There are only a few victims of Japanese, Chinese, Vietnamese, Asian Indian, American Indian/Alaskan Native, Cambodian, Pacific Islander, Guamanian, Laotian, Samoan and Hawaiian. There are also a significant no. of victims from unknown descent.

### 6.1.4 YEAR COLUMN

From the graph, we can conclude the following:

- The no. of collisions are approximately the same from 2010 to 2014.
- There is a significant increase in the no. of collisions from 2014 to 2019.
- In 2020, there is a significant decrease in the no. of collisions, may be due to the decrease in traffic due to corona pandemic. 2021 has lesser collisions as data has values up to date 9<sup>th</sup> January 2021.

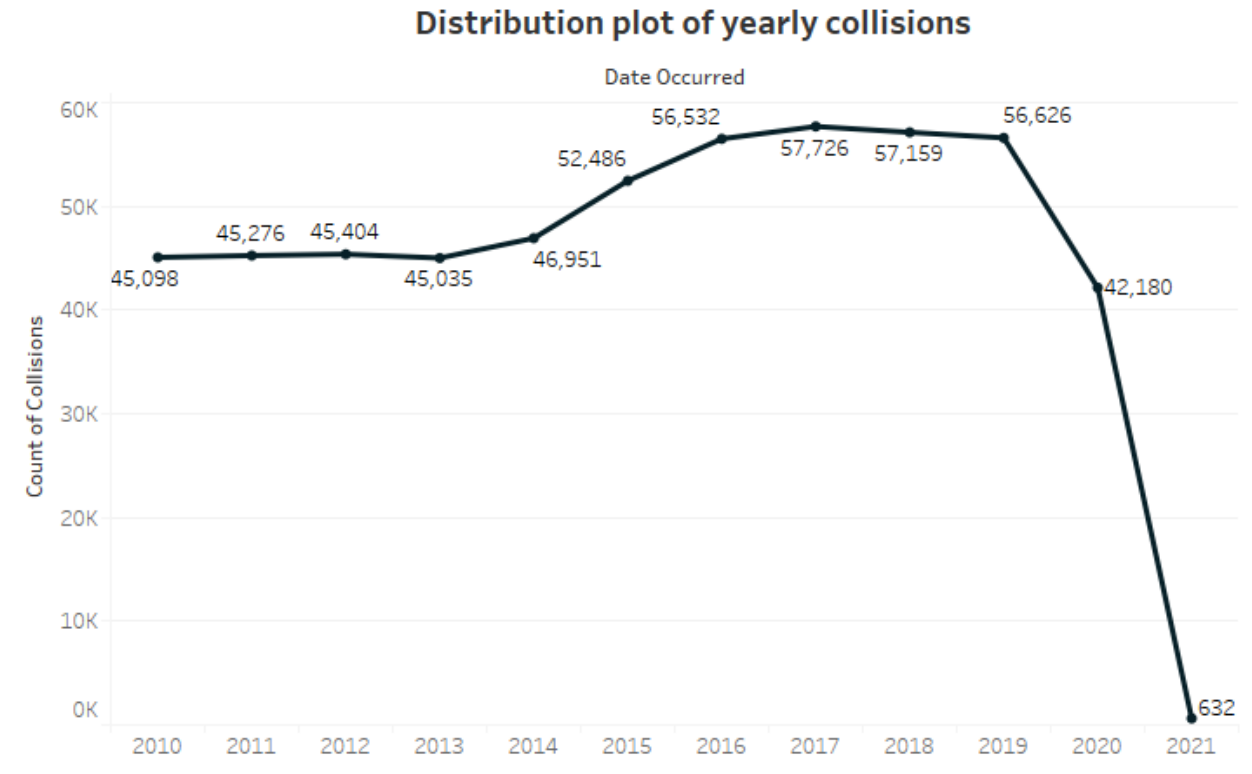


Figure 6.4 Distribution plot of yearly collisions

The sudden increase in collision count can be either due to the proper implementation of traffic collision data collision or due to the sudden increase in traffic during these years.

### 6.1.5 MONTH COLUMN

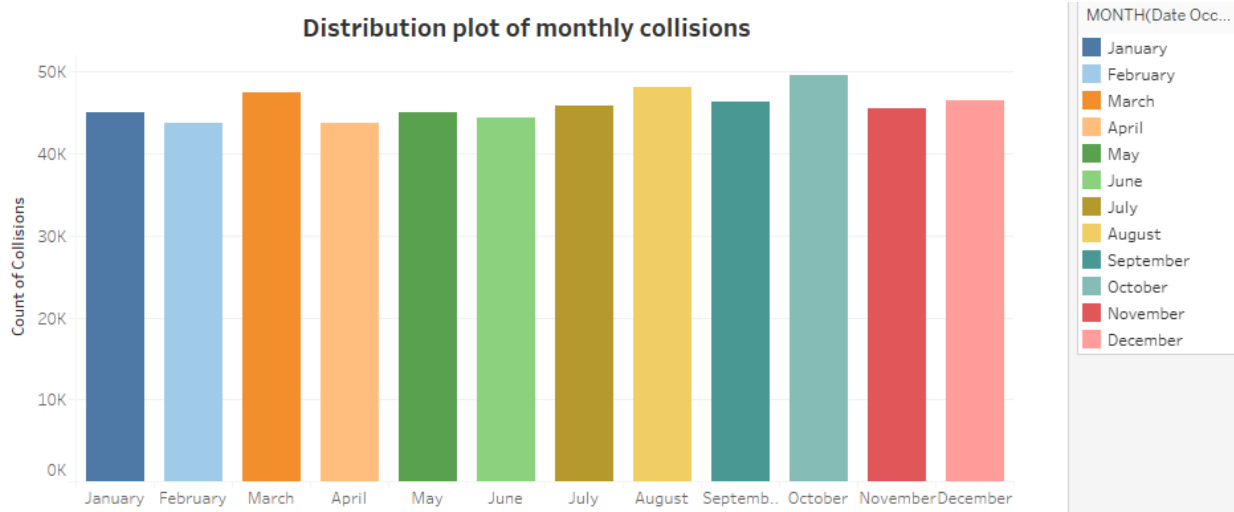


Figure 6.5 Distribution plot of monthly collisions

Most no. of collisions occurred during March and October. Lesser no.of collisions are there in February and April. All the months have approximately collision counts. But this can also indicate the seasonality. This can be further analysed during the time series analysis

### 6.1.6 WEEKDAY COLUMN

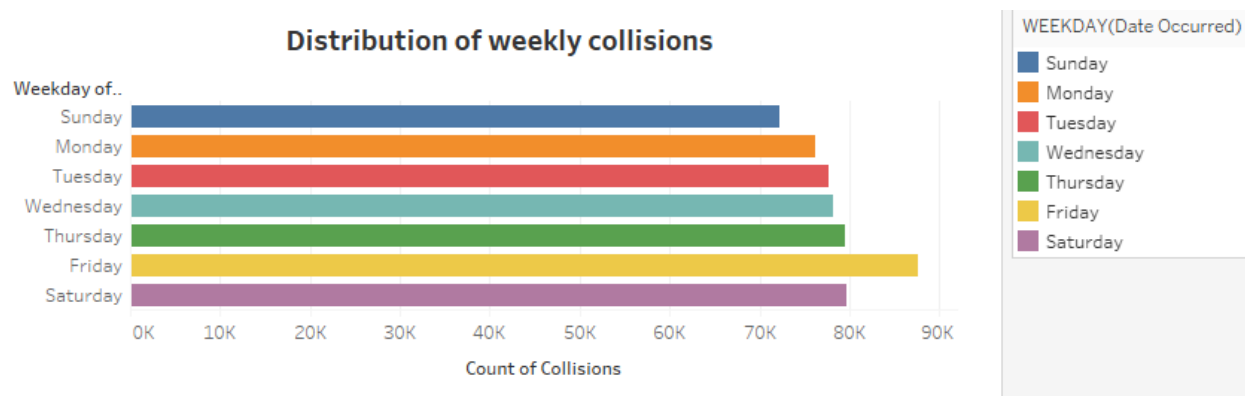


Figure 6.6 Distribution plot of weekly collisions

Collisions are increasing from Sunday to Friday, with a sharp increase from Thursday to Friday and at their weekly minimum on Sunday and maximum on Friday. This indicates a presence of weekly seasonality in the data. By analysing the data for Friday on hourly collisions, we can find the peak hour in Friday. There may be many reasons for the increase in collisions on Friday. This can be identified by analysing other social factors and can be used to implement counter measures to reduce the no.of collisions.

### 6.1.7 VICTIM SEX COLUMN

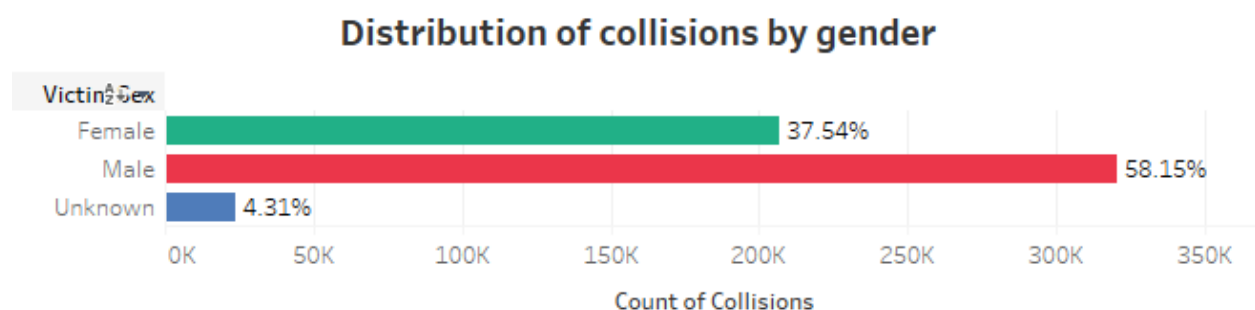


Figure 6.7 Distribution plot of collisions by gender

This plot tells us that given that a collision occurred, the victim is much more likely to be male than female. 58.15% collision victims are males and 37.54% are females. There is also 4.31% unknown sex victims. This work would be more interesting if I had the total number of drivers by gender, allowing for a collisions per capita measure. This will be a recurring shortcoming of this data and would be one of the main extensions of this analysis.

### 6.1.8 HOUR COLUMN

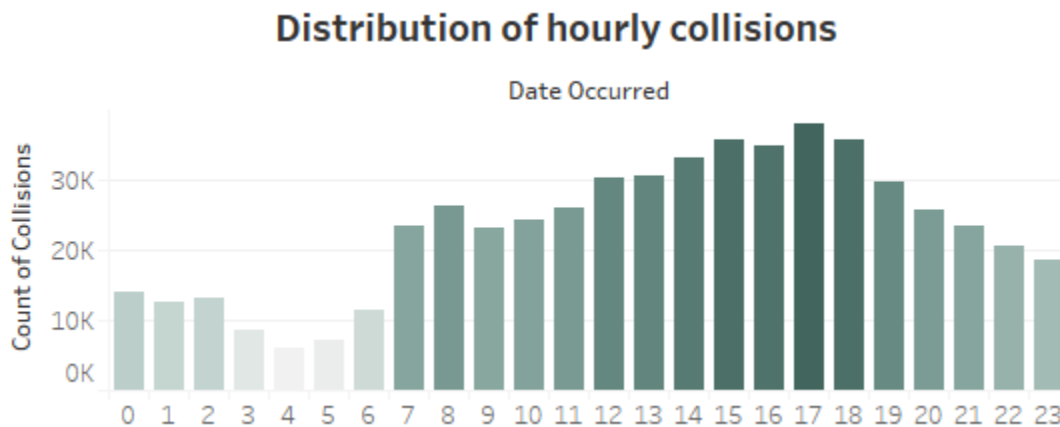


Figure 6.8 Distribution plot of collisions by hour

From the graph, it is clear that collisions are sharply increasing from ~4am to ~8am, decreasing from ~8am to ~9am, generally increasing from ~9am to ~5pm and sharply decreasing from ~5pm to ~4am. No. of collisions have their daily minimum at ~4am and daily maximum at ~5pm.

### 6.1.9 PREMISE COLUMN

By analysing the premise column, it is clear that most of the collisions occurred in the street. Proper measures need to be taken in order to reduce the no. of collision in street. There is no availability in the data regarding pedestrians were involved or not. This needs further attention. Also parking lot has second most no. of reported collisions. Collisions in parking lot can be generally minor as it may occur during parking. Proper counter measures are needed in this case also.

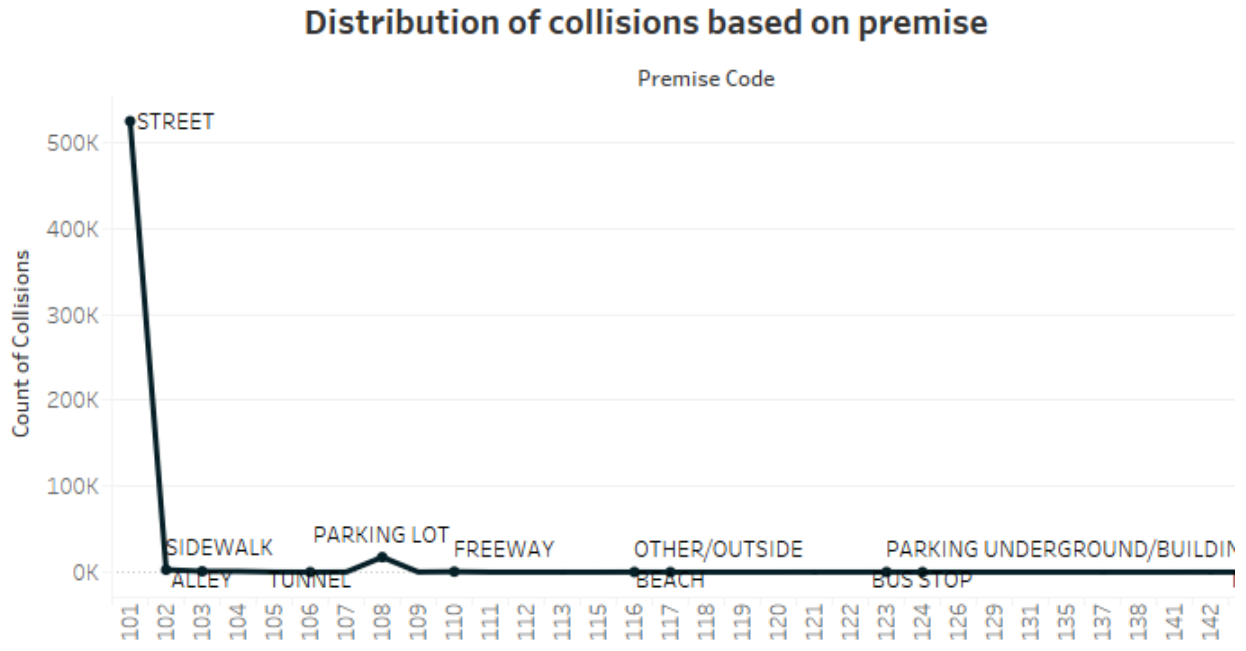


Figure 6.10 Distribution plot of collisions by premise

#### 6.1.10 TIME TAKEN TO REPORT ACCIDENT

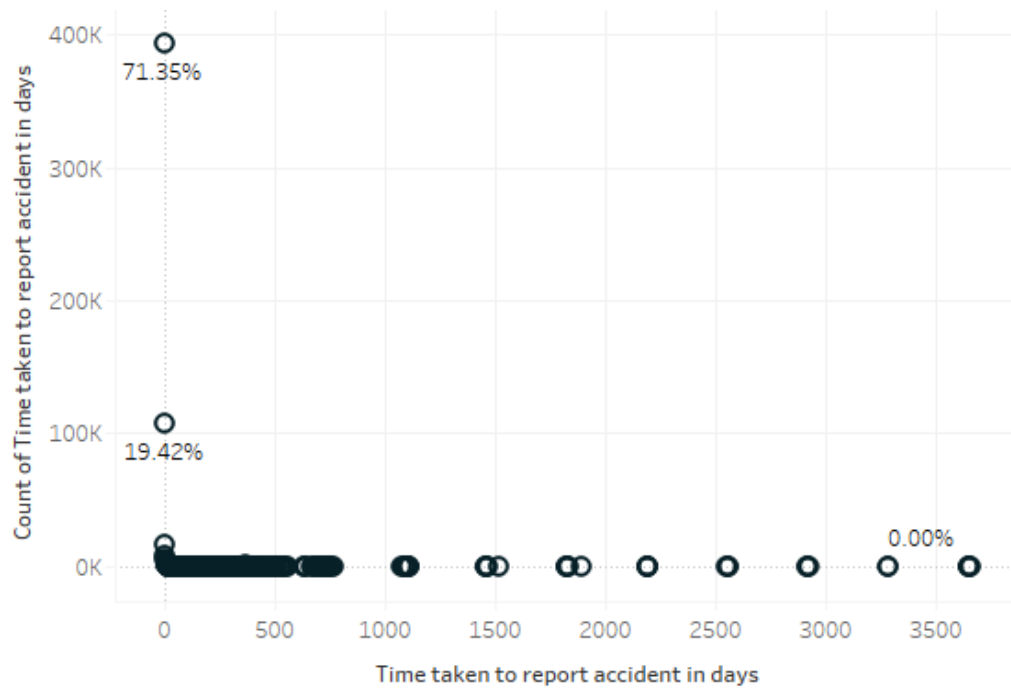


Figure 6.10 Distribution plot of collisions by time taken to report accident

By analysing the time taken to report column, most of the collisions were reported on the day of collision occurrence. There are only few collisions reported after 1 or 2 years.

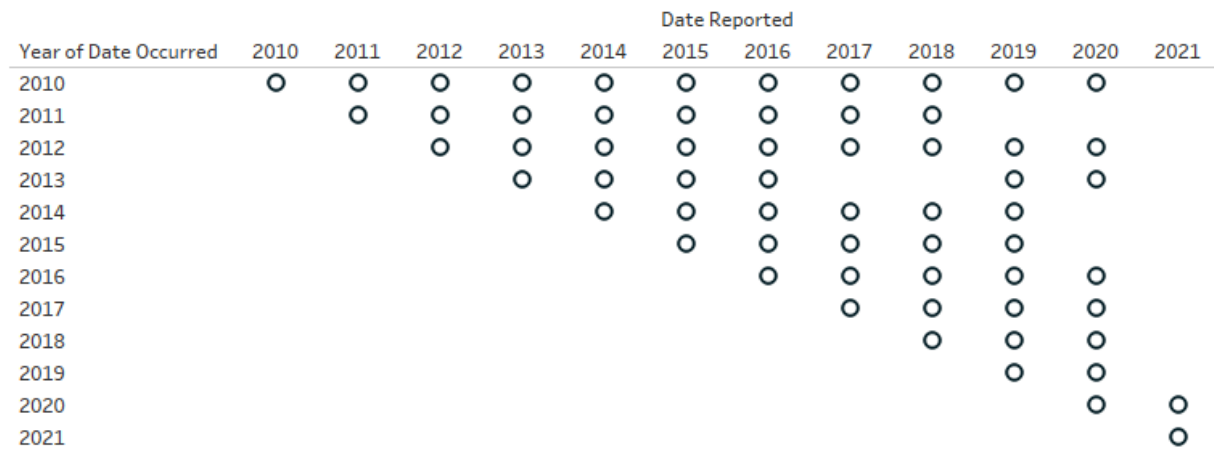


Figure 6.11 Plot between date of occurrence and date reported

The collisions occurred in 2010 reported even in 2020 also. Few collisions are reporting lately. This requires further consideration. Also an analysis on the severity of collision and late reporting can be conducted to analyze this problem further.

## 6.2 BI-VARIATE ANALYSIS

### 6.2.1 AGE vs WEEKDAY

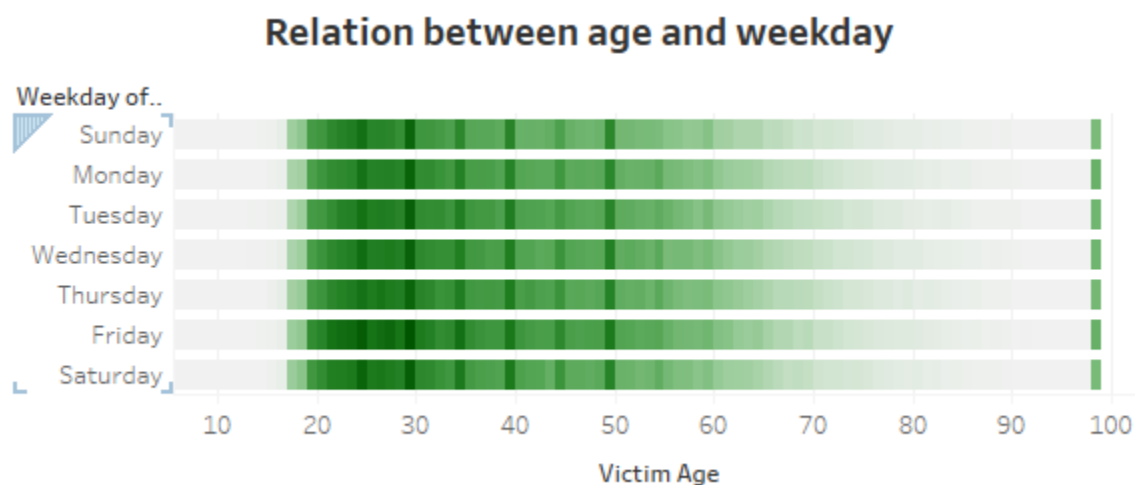


Figure 6.12 Plot between age and weekday



From this graph, it is clear that the no.of collisions are more for ages which are multiples of 5. This may be due to rounding the age during the recording process. Most no.of victims are in the age of 30. There is an unusual spike in age 99.

### 6.2.2 AGE vs HOUR

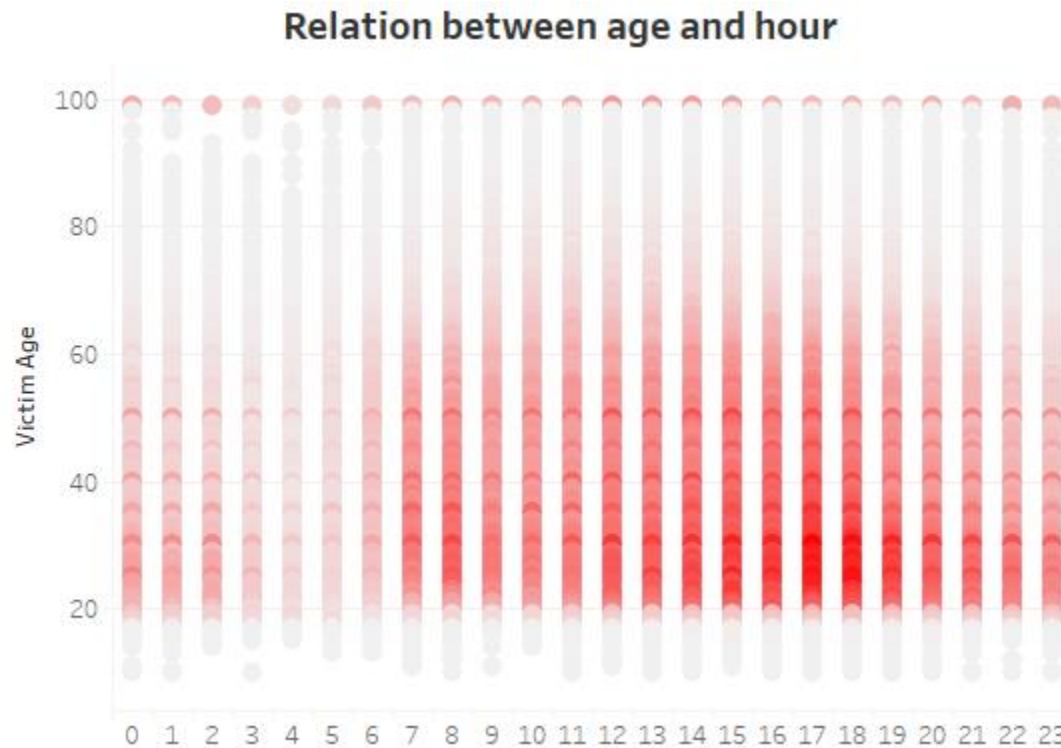


Figure 6.13 Plot between age and hour

Accidents increased from 7.00 AM onwards with peak at 5.00PM. That is, 5.00 PM is the most accident-prone hour of the day. People in the age 25 to 30 cause more no.of collisions at 5.00 PM.

### 6.2.3 AGE vs MONTH

In every month, middle aged people are met with more accidents. Regarding the age 99, approximately the same no.of collisions are occurring within this age. This needs further analysis.

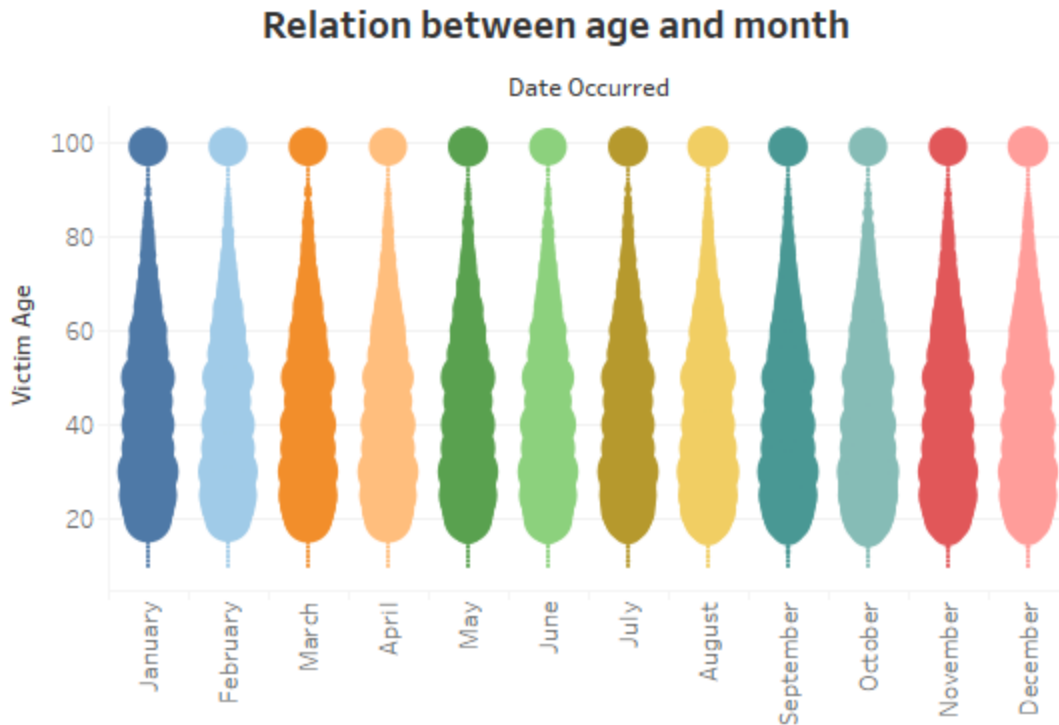


Figure 6.14 Plot between age and month

### 6.2.5 AGE vs YEAR

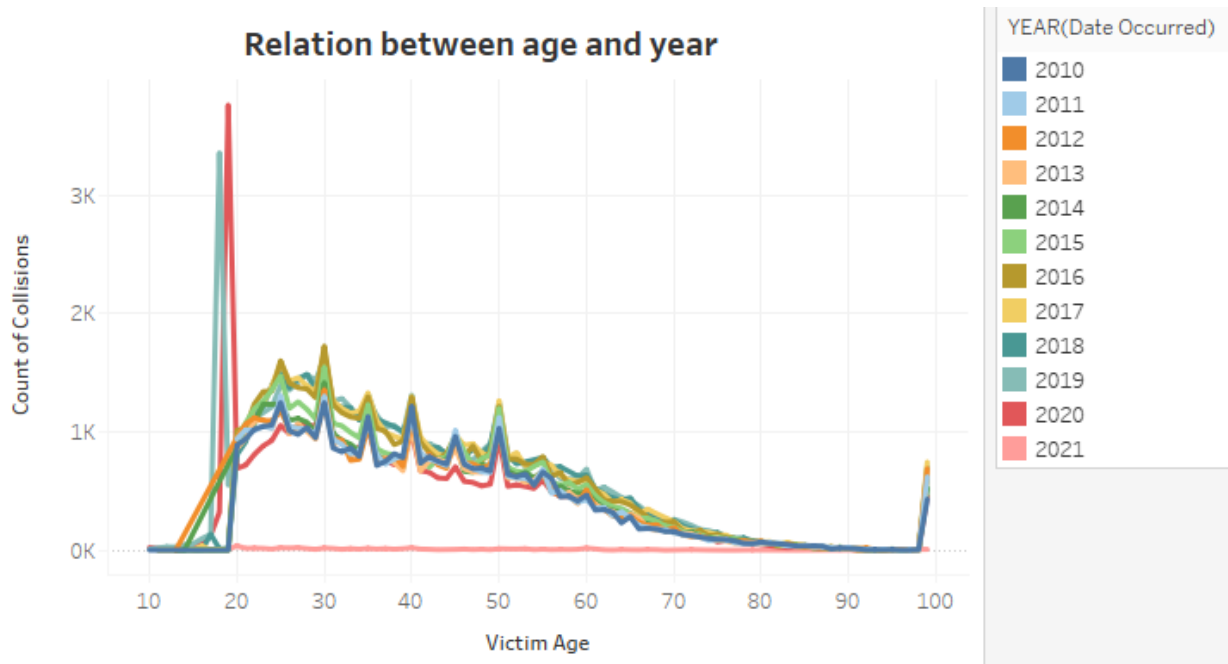


Figure 6.15 Plot between age and year

Apart from the peaks at multiples of 5 ages, there is an interesting pattern obtained from this graph. During the 2018, peoples with 18 years old caused more no.of collisions and in 2019, peoples with 19 years old caused more no.of collisions. Therefore, this age group requires further attention and they must be analysed accordingly. Proper awareness programme or policies must be taken in to action for this age group. During the past years, youth is the highest in the no.of victims.

### 6.2.6 AGE vs PREMISES

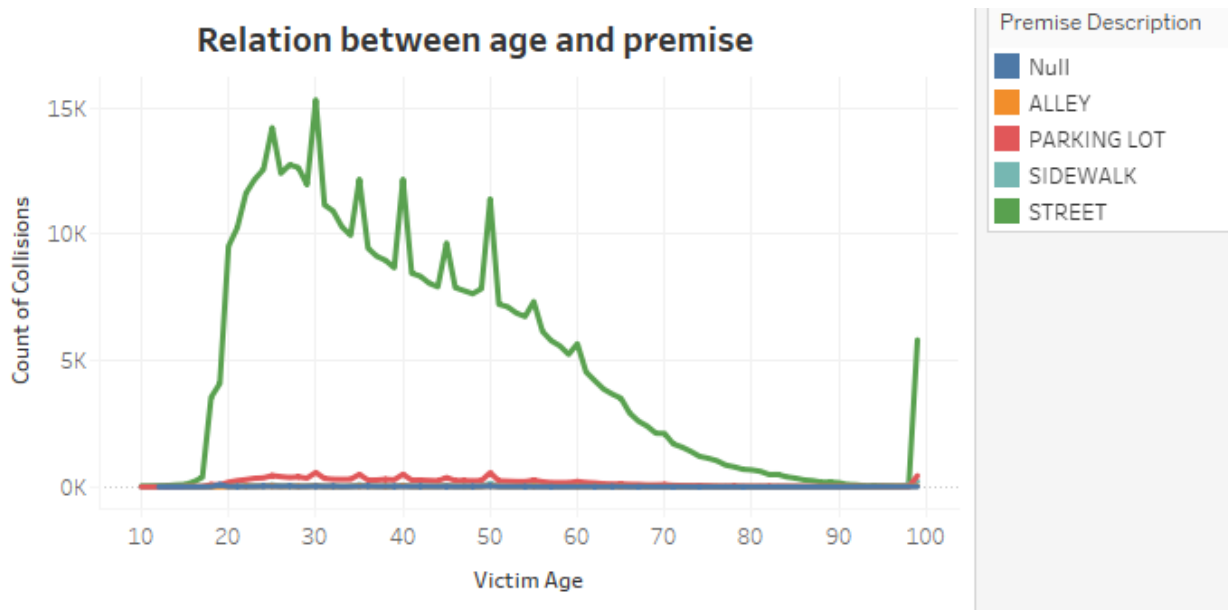


Figure 6.16 Plot between age and premise

Most no.of collisions in street is caused by the age group 30. Within the parking lot, the no.of collisions caused by all age groups is approximately the same. There is also a change in peak for the ages as multiples of 10 in street.

### 6.2.7 AGE vs AREA

77<sup>th</sup> Street is the most accident occurring area. Wilshire, Topanga also have high accident rates. Foothill is one of the safest places in the city. Within all areas the distribution of collisions by various age people is approximately in the same pattern. Proper measures need to be taken in high accident prone areas.

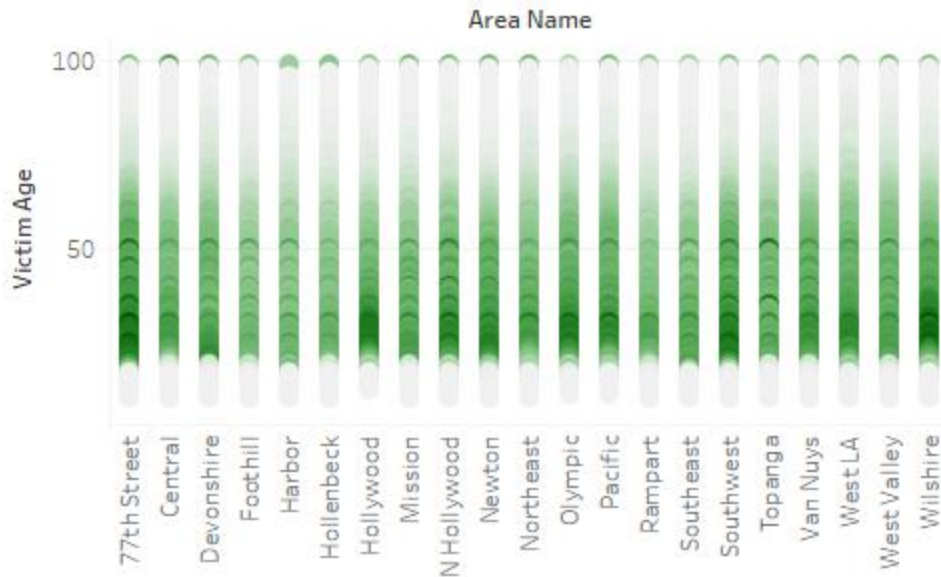


Figure 6.17 Plot between age and area

### 6.2.7 AGE vs VICTIM DESCENT

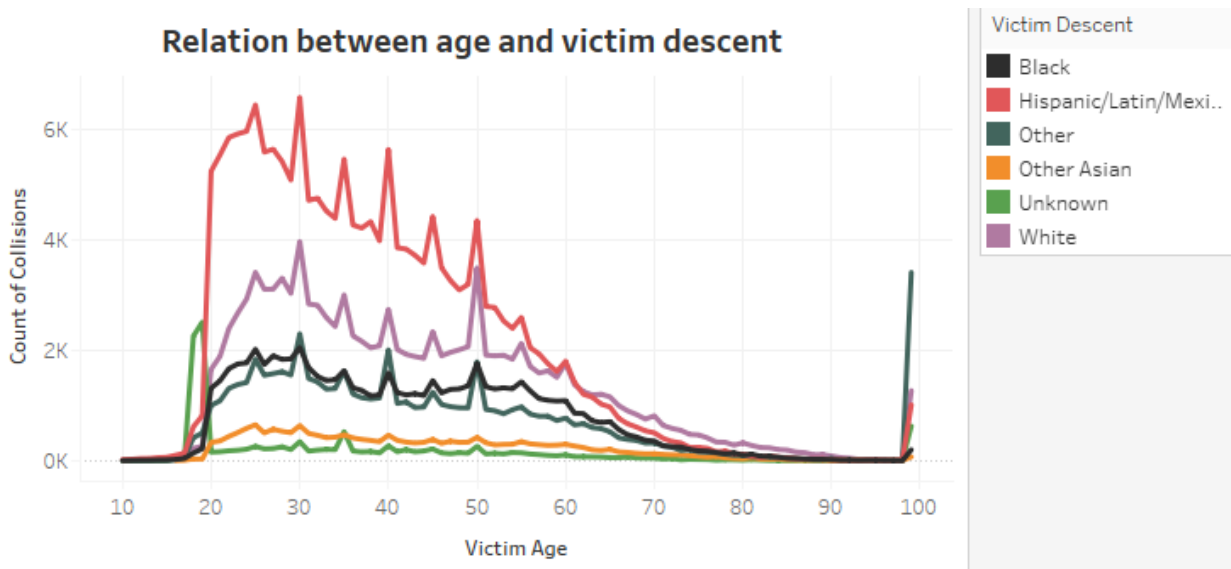


Figure 6.18 Plot between age and victim descent.

Most of the collisions were caused by Hispanic/Latin/Mexican people. White people take second place. For age 50, both Hispanic/Latin/Mexican people and White people caused approximately the same no. of collisions. People with unknown victim descent have more victims in the age group 15 to 20. Also here victim descent 'other' have more victims in the age 99. All other victim descent were neglected during this plotting, since only the top 6 were taken into account.

### 6.2.8 AGE vs VICTIM SEX



Figure 6.19 Plot between age and victim sex

From the plot, it is clear that males dominate the collisions. There is a sharp increase in age 50 of male. Age 99 has a very unusual rise for males mainly. There is also an interesting point revealed in this plot. People with in the age 15 to 20 is mainly denoted as unknown sex.

## 6.2.9 AREA vs VICTIM SEX

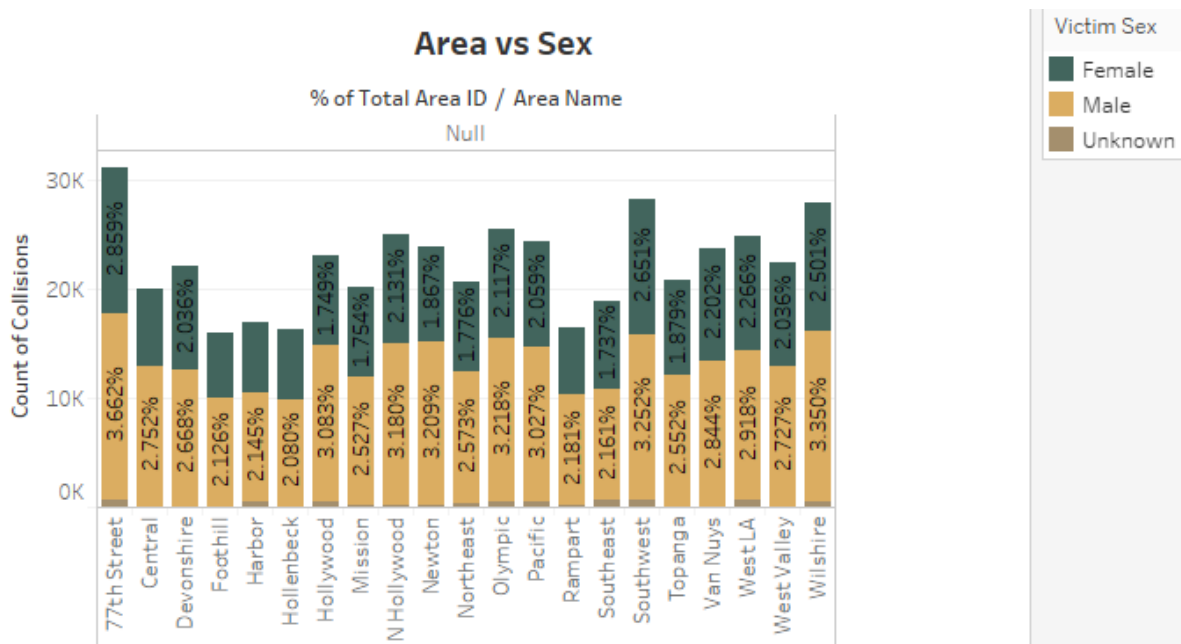
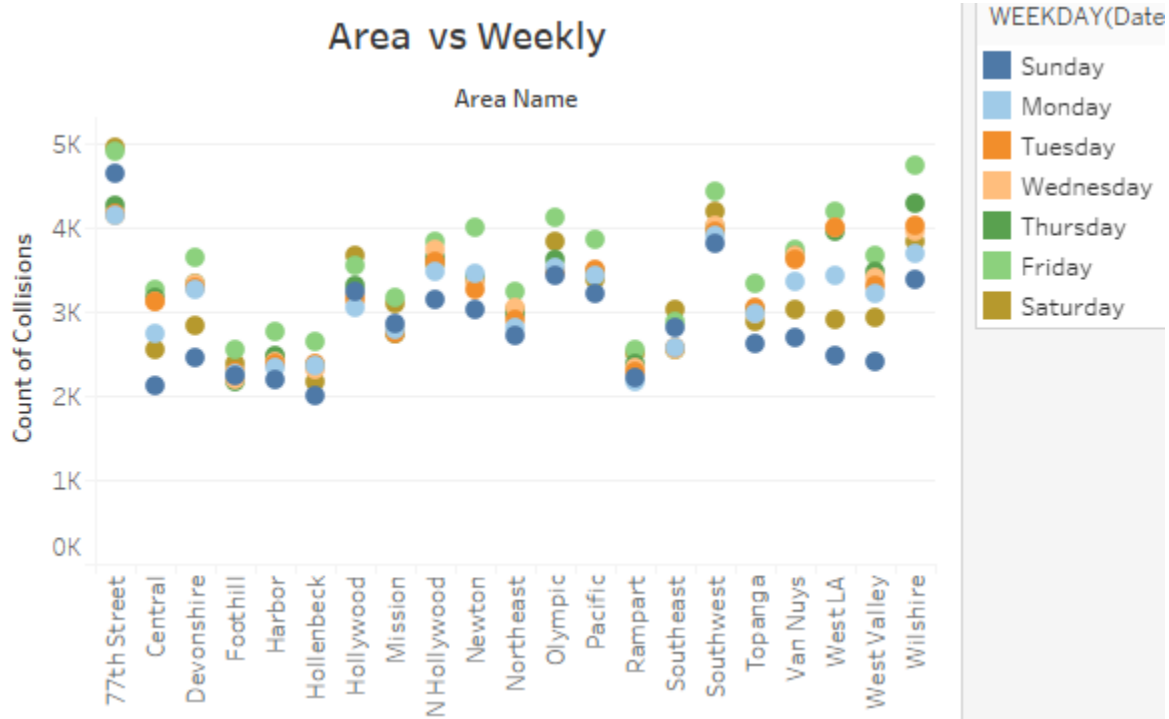


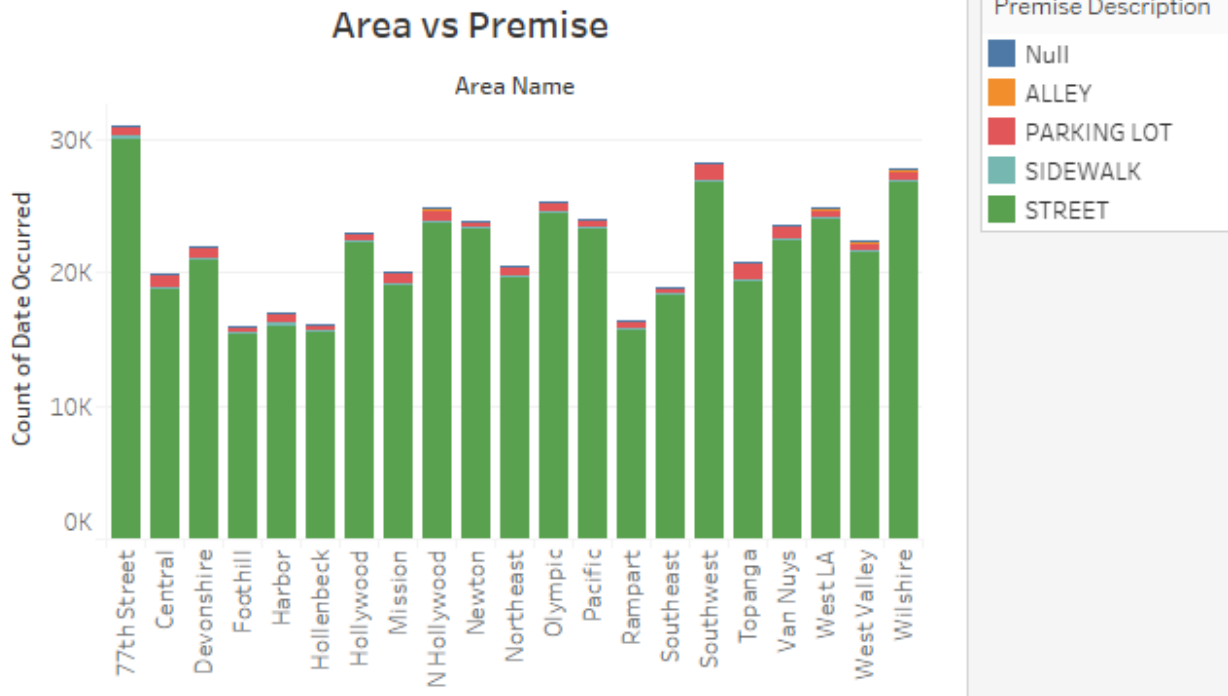
Figure 6.20 Plot between area and victim sex

## 6.2.10 AREA vs WEEKDAY

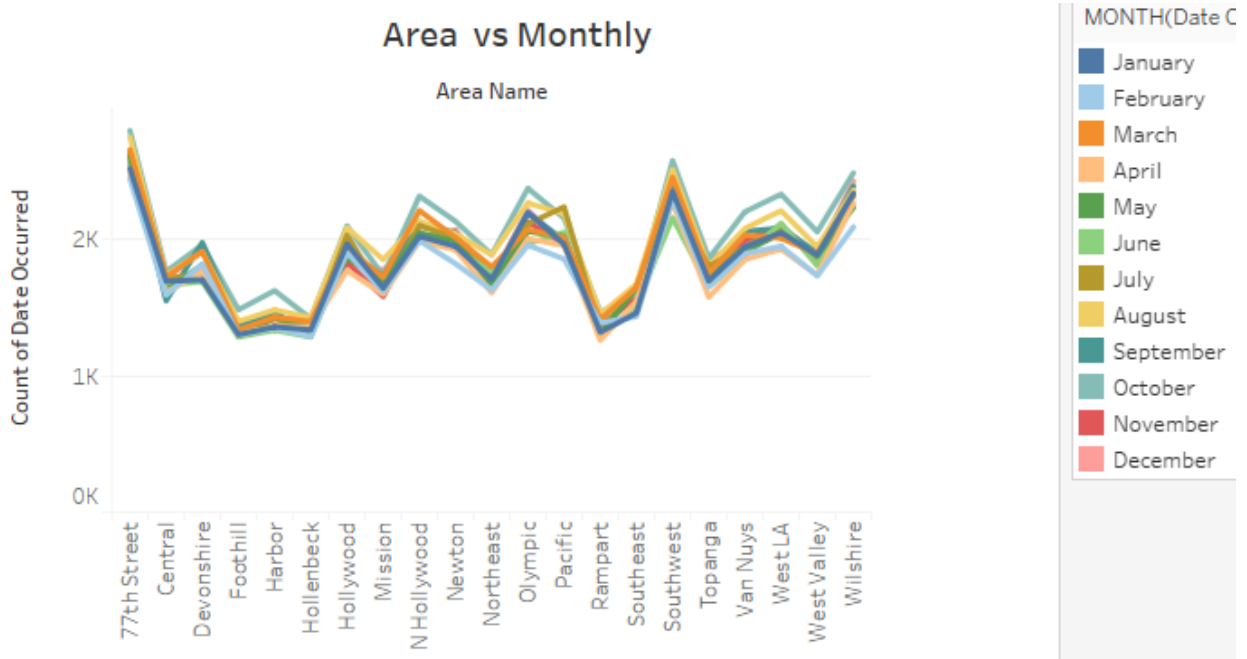


- 77<sup>th</sup> Street has high collisions occurring on Saturday, Friday, Sunday.
- Wilshire and Southwest also have a large number of victims.
- Friday tops in most areas.
- Sunday the least.

## 6.2.11 AREA vs PREMISE

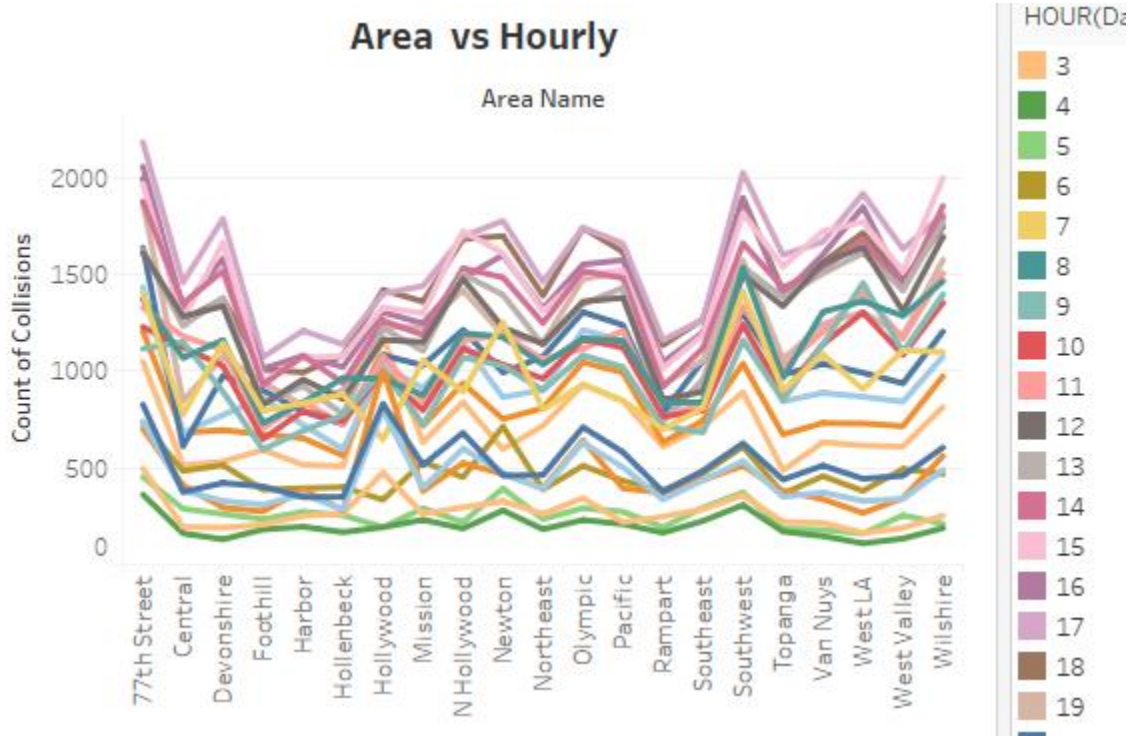


## 6.2.12 AREA vs MONTH



- 77<sup>th</sup> Street had the highest collision in the city and October tops in that.
- Hollenbeck can be the safest place to drive in the city.
- Most collisions occurred in October for most of the cities.

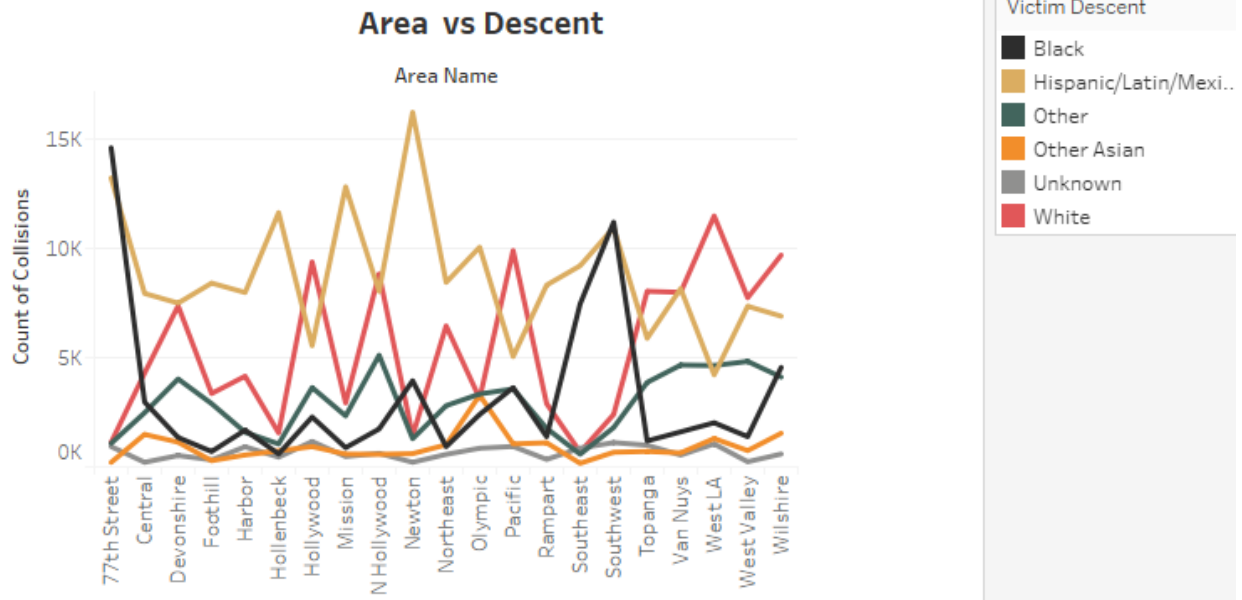
### 6.2.13 AREA vs HOUR



- 77<sup>th</sup> Street, Wilshire, Southwest, West LA tops the list.
- At 5.00 PM most people lost their lives.
- Accidents increase in evening time in most areas.

### 6.2.14 AREA vs VICTIM DESCENT

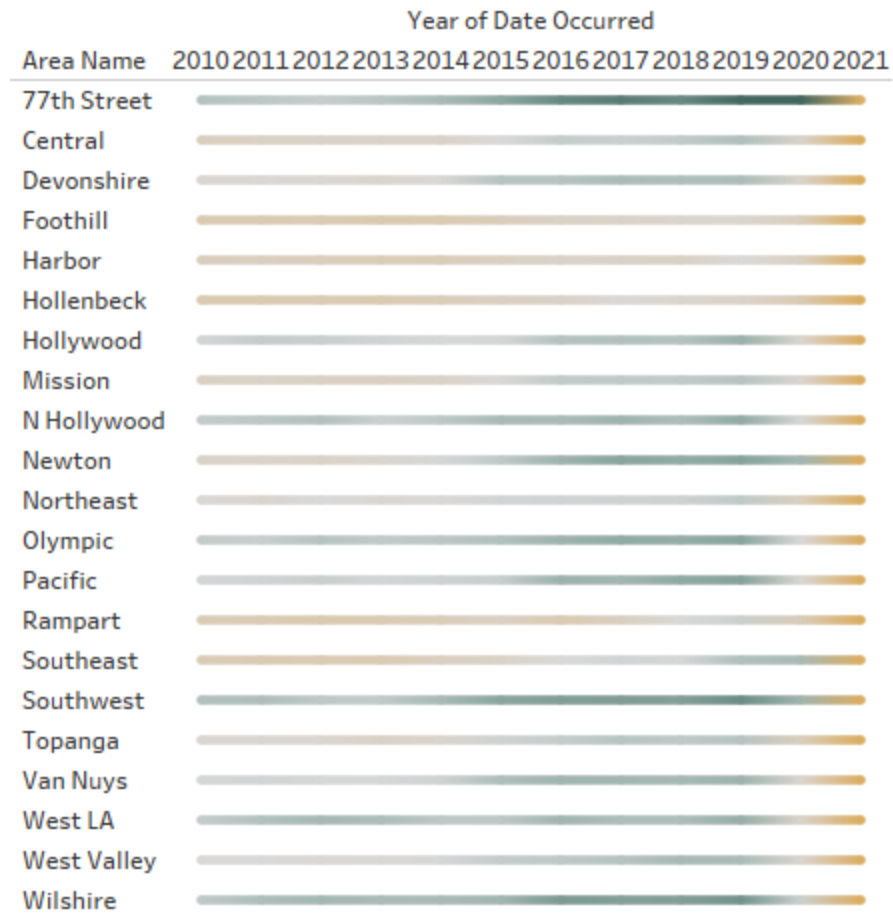




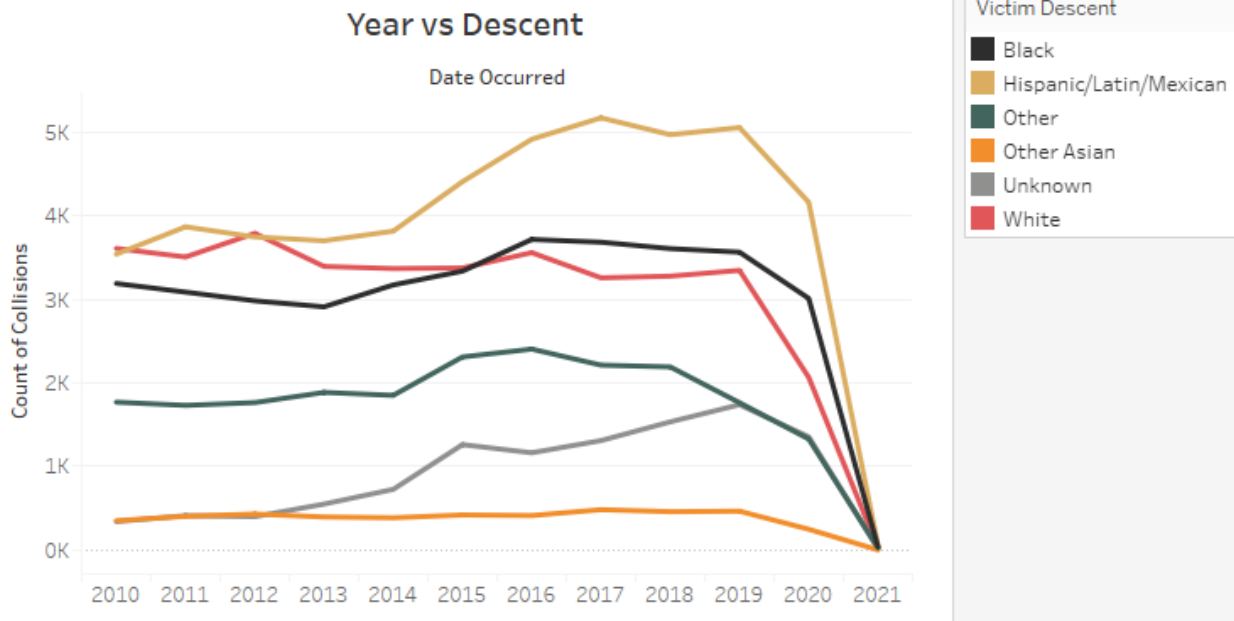
- 77<sup>th</sup> Street gives the highest count for black groups.
- Hispanic/Latin/Mexican is more in Newton, Olympic, Mission, Hollenbeck
- White group is more in West LA and Pacific.

## 6.2.15 AREA vs YEAR

## Year vs Area



### 6.2.16 VICTIM DESCENT vs YEAR



- Hispanic/Latin/Mexican – Friday has the highest point in the graph.
- White took 2nd place.
- Most victims are on weekends.

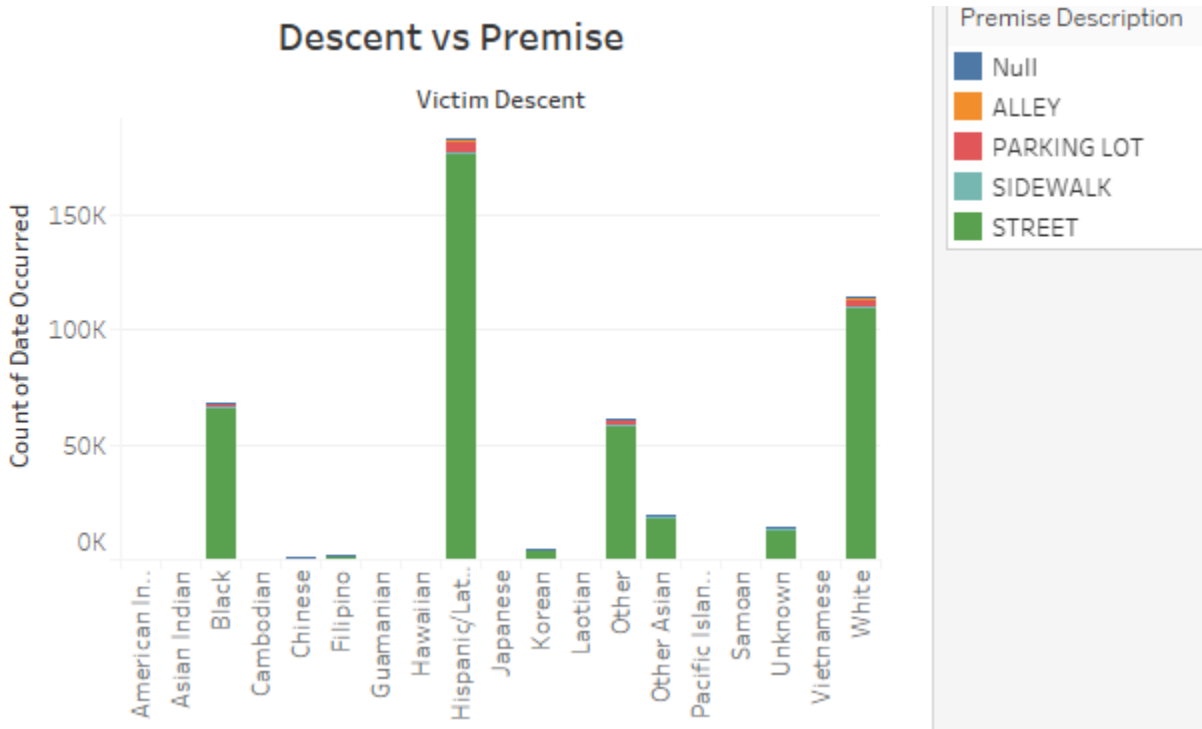
## 6.2.17 VICTIM DESCENT vs MONTH

## Descent vs Monthly

Victim Desc..	Date Occurred											
	January	February	March	April	May	June	July	August	Septemb..	October	November	December
American In..	16	12	18	13	12	6	16	11	10	16	16	20
Asian Indian	15	8	10	10	11	17	16	9	18	19	5	7
Black	5,493	5,344	5,896	5,516	5,838	5,642	5,738	5,959	5,643	5,952	5,634	5,803
Cambodian	2	3	1	1		1	4	1			1	
Chinese	64	73	71	47	76	70	72	71	60	65	58	65
Filipino	128	149	145	116	126	114	136	141	125	150	120	108
Guamanian	3	4	3	2	4	2	6		4	5	1	5
Hawaiian	13	16	26	14	10	17	17	28	20	27	15	27
Hispanic/La..	14,936	14,438	15,461	14,578	14,904	14,582	15,377	16,104	15,545	16,725	15,284	15,781
Japanese	27	26	24	24	15	18	29	27	29	28	28	25
Korean	356	336	330	361	321	333	369	315	352	365	339	368
Laotian		1	1				2	1				2
Other	4,996	4,796	5,233	4,884	5,033	5,053	5,126	5,362	5,074	5,569	5,085	5,211
Other Asian	1,662	1,586	1,622	1,484	1,567	1,557	1,593	1,698	1,612	1,668	1,611	1,673
Pacific Islan..	15	12	11	14	17	13	22	15	20	21	22	12
Samoan	2	4	3	3	3	3		4	2	3	1	
Unknown	1,046	1,029	1,058	981	1,056	1,062	1,172	1,334	1,318	1,463	1,191	1,228
Vietnamese	18	11	23	20	16	11	17	25	11	18	25	19
White	9,589	9,132	9,802	9,098	9,484	9,347	9,486	10,057	9,751	10,211	9,401	9,406

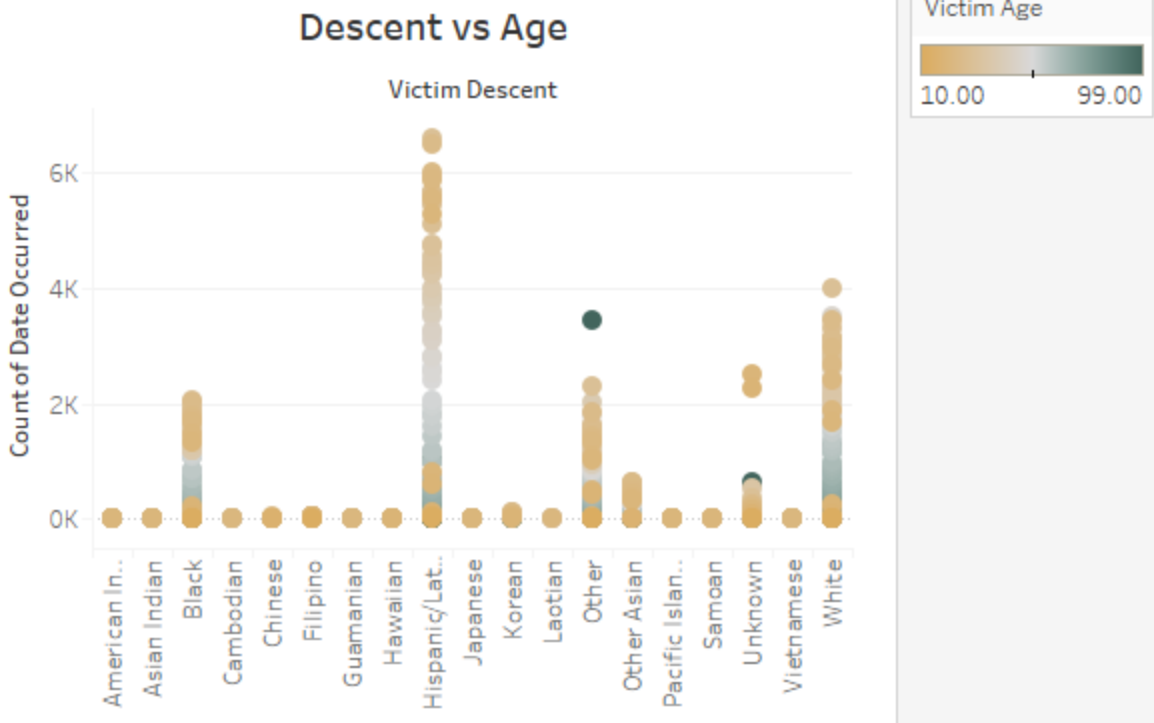
- Hispanic/Latin/Mexican records most victims.
- Most groups have the highest victims in October and august.
- March also shows a high number of victims.

### 6.2.17 VICTIM DESCENT vs PREMISE



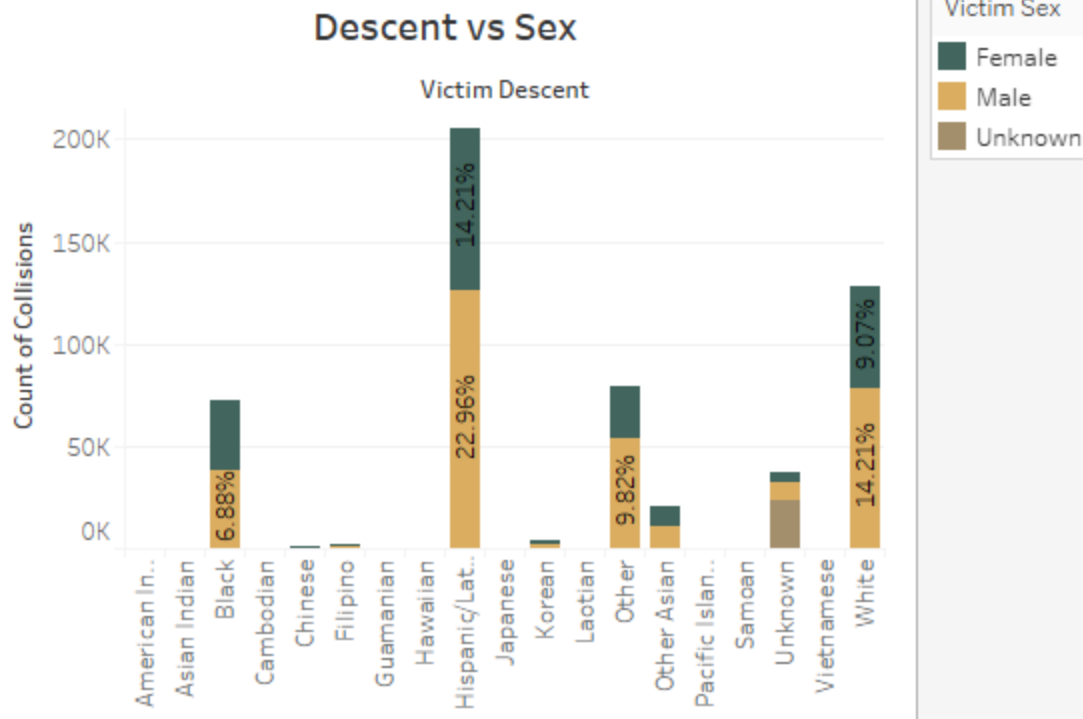
- Street has the highest number of victims.
- Parking lot comes in 2<sup>nd</sup>.

### 6.2.18 VICTIM DESCENT vs AGE



- Most victims are in the age group 20-30.
- Victims having age 99 and others are high, due to data misinformation.
- Unknown people having the age group of 18-19 also have a high count.

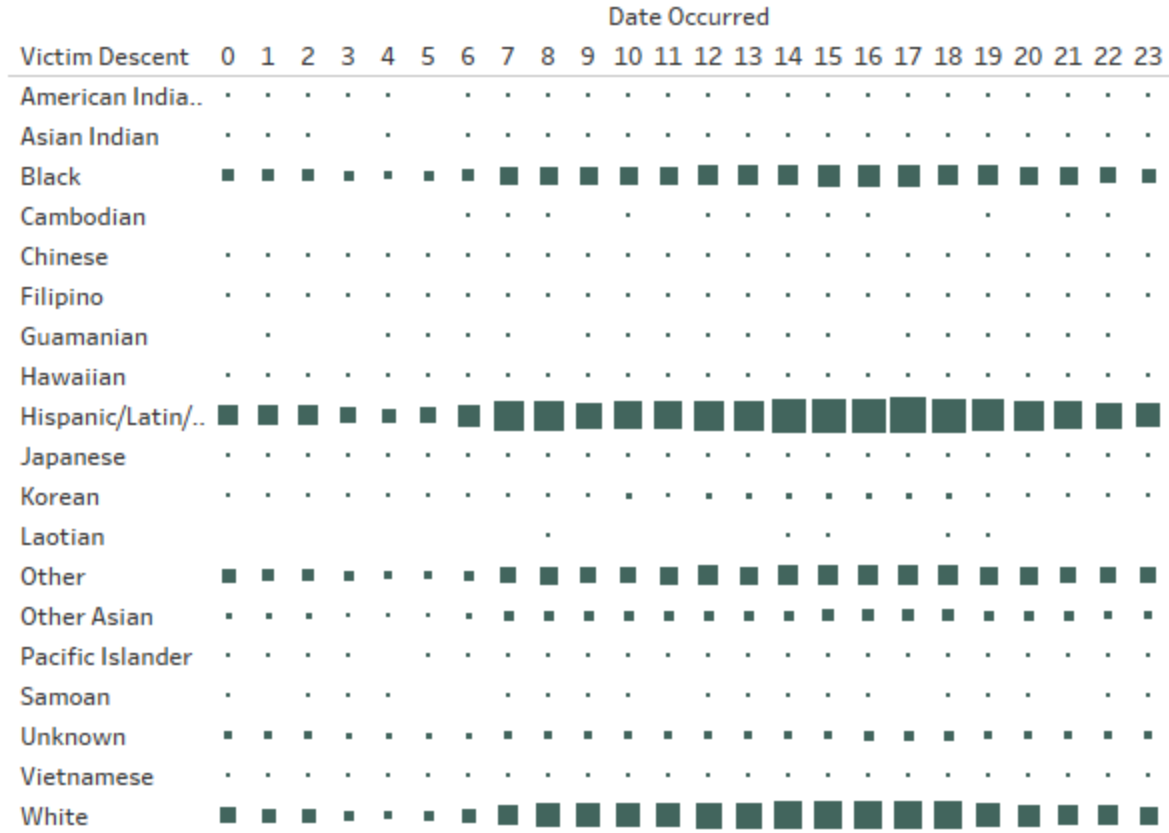
#### 6.2.18 VICTIM DESCENT vs SEX



- Males take majority in all descents.
- In the case of black, both females and male have almost equal victims.
- Majority of unknown descent is missing values.

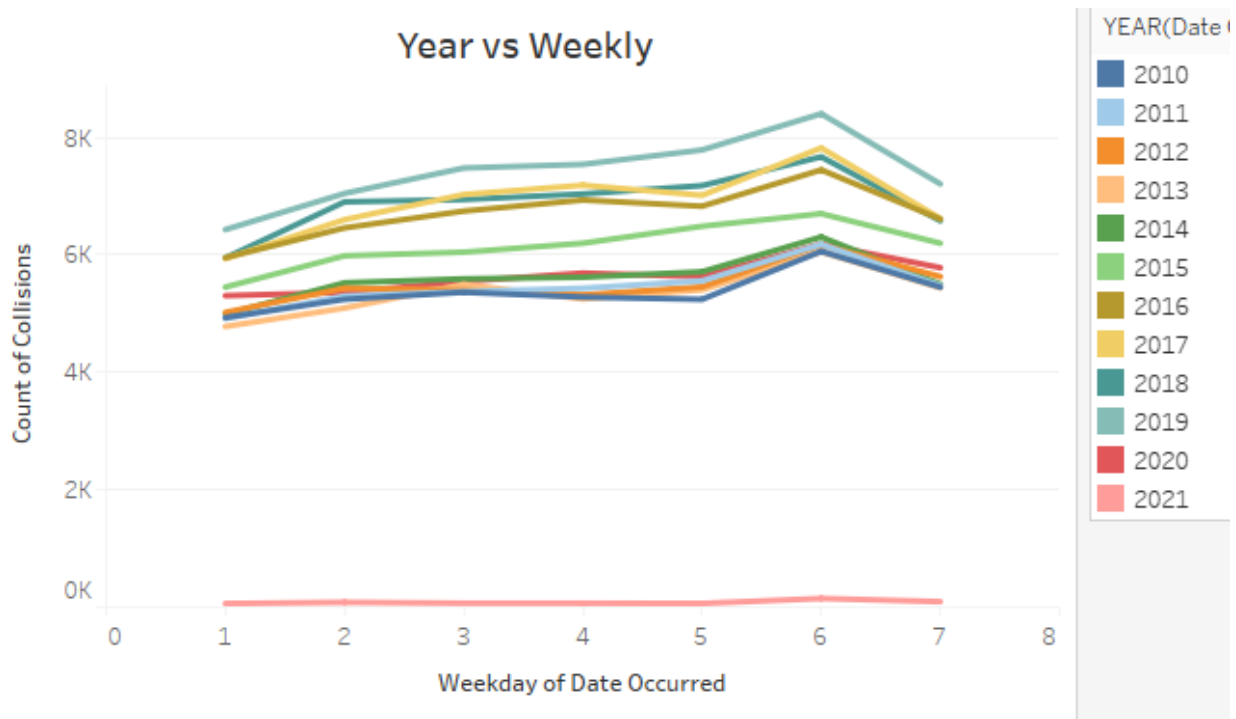
#### 6.2.18 VICTIM DESCENT vs HOUR

## Descent vs Hourly



### 6.2.19 YEAR vs WEEK





- 2019 is at the top of the list in past years.
- However, 2017 overtook 2018 claiming 2<sup>nd</sup> spot.
- 2020 is very low due to Covid and lockdown.

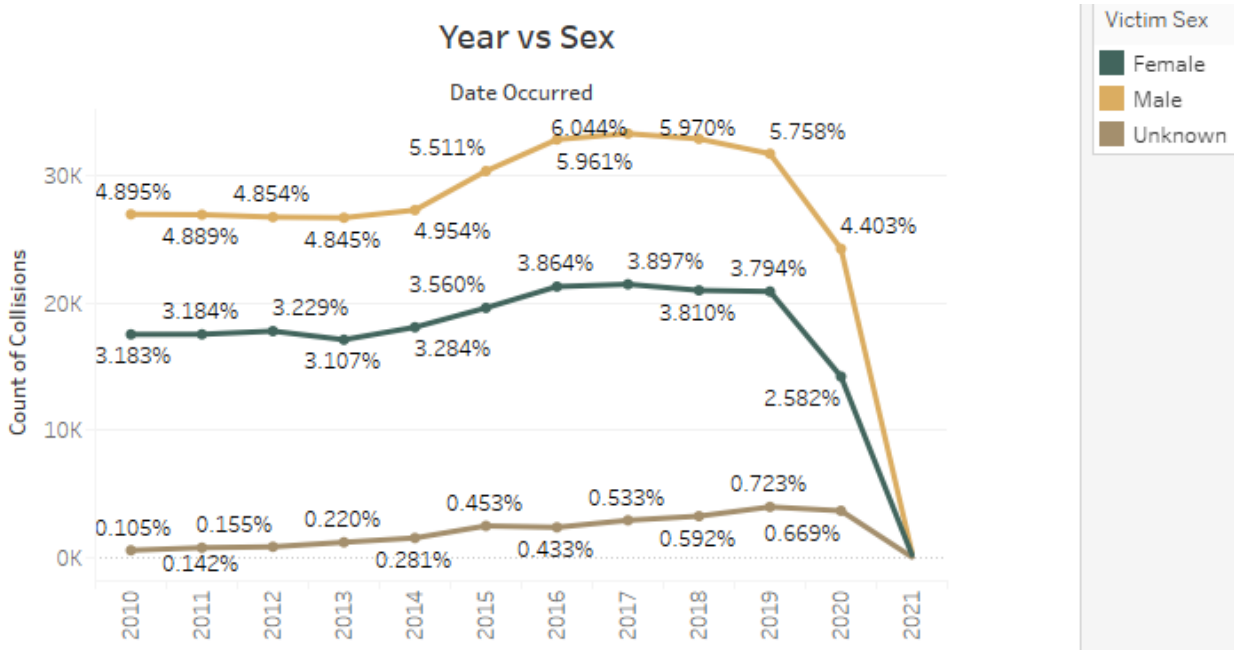
## 6.2.20 YEAR vs MONTH

**Year vs Monthly**

Year of Date Occurred	Date Occurred											
	January	February	March	April	May	June	July	August	September	October	November	December
2010	3,090	2,945	3,204	3,046	3,173	3,021	3,123	3,041	2,999	3,383	3,094	3,452
2011	2,942	3,086	3,346	3,117	3,085	3,068	3,226	3,331	3,246	3,342	3,189	3,256
2012	3,200	3,129	3,539	3,155	3,246	3,114	3,098	3,339	3,178	3,380	2,992	3,054
2013	2,991	2,760	3,165	3,097	3,226	2,974	3,024	3,444	3,114	3,410	3,188	3,121
2014	3,065	2,942	3,269	3,081	3,279	3,241	3,187	3,478	3,409	3,572	3,229	3,489
2015	3,400	2,895	3,278	3,564	3,560	3,500	3,655	3,977	3,690	4,036	3,690	3,820
2016	3,531	3,760	3,937	3,941	3,856	4,048	3,877	4,110	3,983	4,036	3,851	4,044
2017	3,752	3,597	4,257	3,906	4,072	3,943	3,980	4,153	3,902	4,430	4,062	4,147
2018	3,866	3,743	4,126	3,916	3,832	3,921	4,155	4,219	4,027	4,462	4,038	3,951
2019	3,896	3,788	4,232	4,055	4,400	4,203	4,559	4,647	4,608	4,773	4,341	4,390
2020	4,056	4,335	3,385	2,288	2,764	2,815	3,314	3,423	3,438	3,481	3,163	3,036
2021	592											

- 2019 shows a steady increase through the months.
- Notice the sharp decline in April, May, June in 2020 due to COVID.

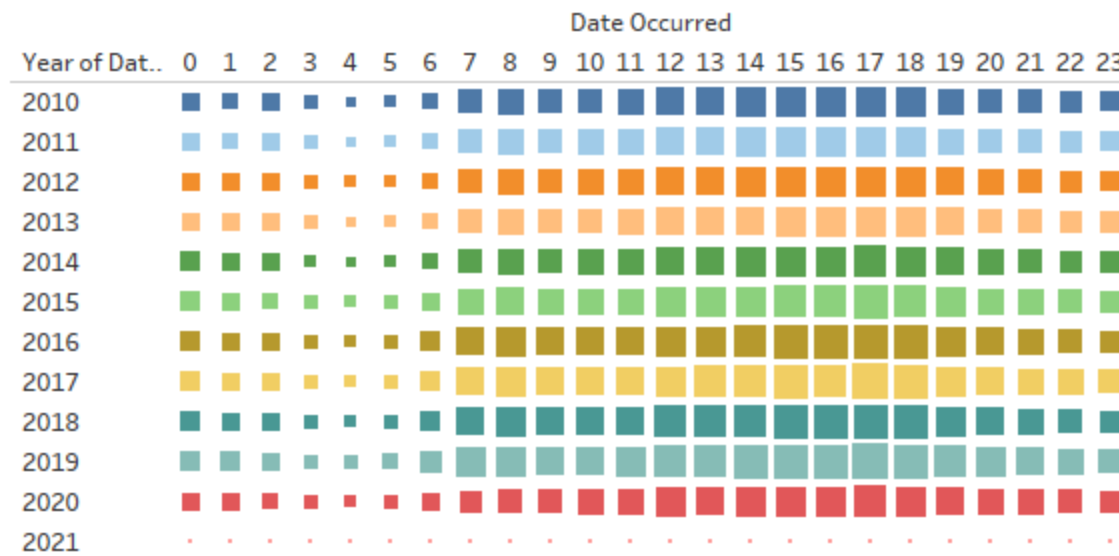
## 6.2.20 YEAR vs VICTIM SEX



- Collisions increase drastically from 2014 in all three sex categories.
- From 2019 onwards it decreases.
- Male in 2017 have the highest % (6.04) of the total collisions.

## 6.2.20 YEAR vs HOUR

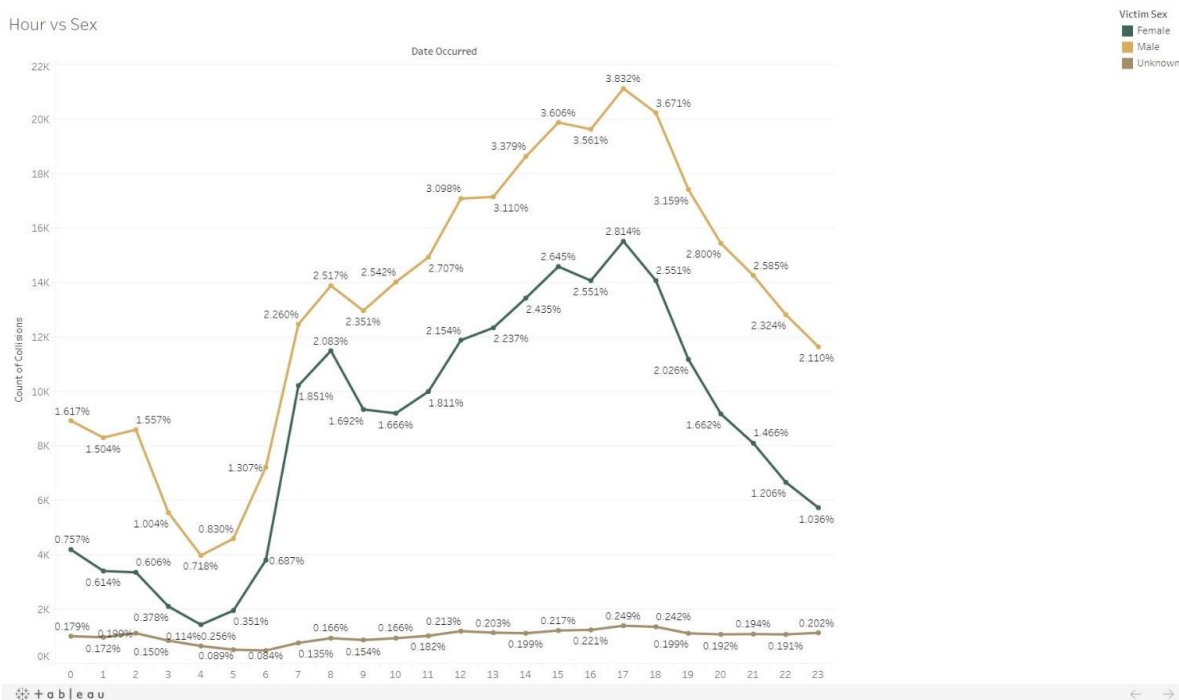
## Year vs Hourly



## 6.2.20 MONTH HOUR vs SEX

Vs

Hour vs Sex



- Males which collide around 5.00 PM have the highest percentage.
- For females also 5.00 PM is the most accident-prone hour.

- We can observe a steady increase from 4.00 AM onwards.

## **CHAPTER 7: DATA PREPROCESSING**

Data preparation involves the method of converting raw data to make it suitable for machine learning models. Time series models contain dates as index and target column. This step involved developing a column of no. of collisions per month using group by method in pandas. Here the no. of collisions in January 2021 is dropped since it contained data up to day 9 only. The no. of collisions were verified by taking the sum of no. of collisions.

No_of_Collisions	
Months	
2010-01-01	3723
2010-02-01	3492
2010-03-01	3900
2010-04-01	3670
2010-05-01	3809
...	...
2020-08-01	3703
2020-09-01	3672
2020-10-01	3740
2020-11-01	3386
2020-12-01	3234

Figure7.1 Derived target column

No_of_Collisions	
count	132.000000
mean	4170.250000
std	547.361686
min	2429.000000
25%	3722.000000
50%	4106.000000
75%	4660.000000
max	5285.000000

Figure7.2 Descriptive statistics of target column

From the descriptive statistics of the target column, it is clear the minimum no. of collisions in a month was 2429 and maximum no. of collisions was 5285. The average no. of collisions in a month is 4170. The data set for building the model contains 132 months data.

## CHAPTER 8: UNDERSTAND TIME SERIES DATA

A time series is simply a series of data points ordered in time. In a time series, time is often the independent variable and the goal is usually to make a forecast for the future. However, there are other aspects that come into play when dealing with time series.

- Is it stationary?
- Is there a seasonality?
- Is the target variable auto-correlated?

Ideally, we want to have a stationary time series for modelling. Of course, not all of them are stationary, but we can make different transformations to make them stationary.

Before understanding these properties, a distribution plot of the target column is plotted. The distribution plot for target column obtained as follows:

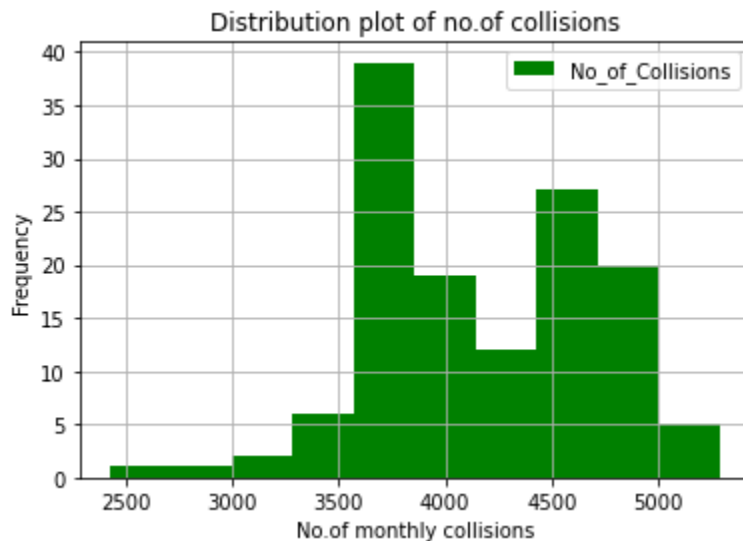


Figure8.1 Distribution plot of target column.

From the distribution plot, it is clear that the frequent no.of collisions is between 3500 and 4000. This plot can also be plotted as a density plot as shown below.

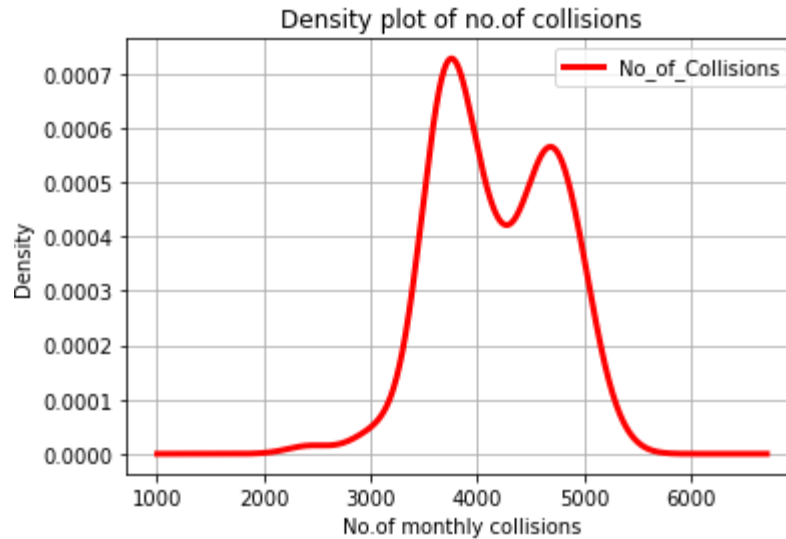


Figure8.2 Distribution plot of target column.

From the density plot, it is clear that the target column is approximately normally distributed. The above plots only indicate the frequency of no.of collisions. A further analysis on the data can be obtained using the time series plot. Here the dataset has time as an index column. Therefore, the plot is obtained as shown below:.

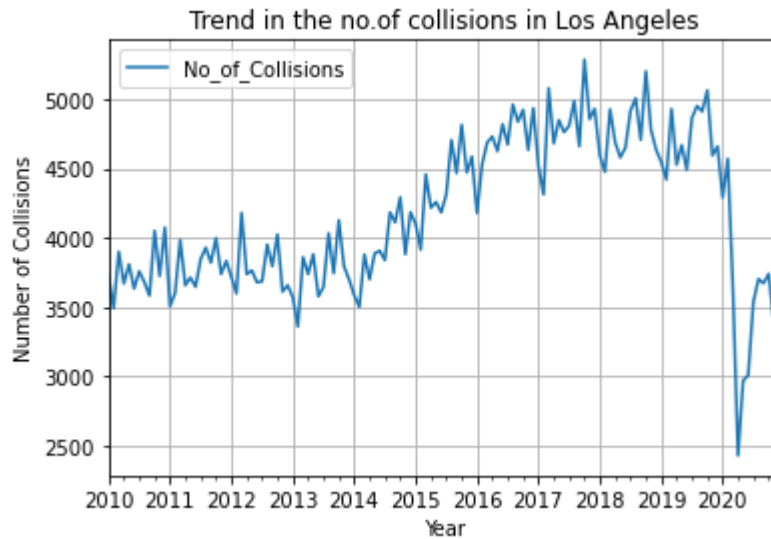


Figure8.3 Time series plot of no.of collisions

Now the graph is more clear. The no.of collisions between 2010 and 2014 is almost linear. There is a significant increase in the no. of collisions from 2014 to 2019. In 2020, there is a fall in the no. of collisions, which may be due to the corona pandemic. It indicates that traffic is increasing from 2010 to 2019 and the pandemic caused a sudden decrease in the traffic, and thus the collisions. This also shows a seasonality in data.

## 8.1 COMPONENTS OF TIME SERIES

A time series is composed of mainly trend, seasonality, and noise. The component parts of a time series were obtained using automated decomposition methods. Trend is how the series data increases or decreases over time. Is it moving higher or lower over the time frame? The series is either uptrend or downtrend, both of which are non-stationary. Seasonality refers to a repeating periodic or cyclical pattern with regular intervals within a series. The pattern is within a fixed time period and it repeats itself at regular intervals. There can be upward or downward swings but it continues to repeat over a fixed period of time as in a cycle. Cyclicality could repeat but it has no fixed period. In general, noise captures the irregularities or random variation in the series. It can have erratic events or simply random variation. It has a short duration. It is hard to predict due to its erratic occurrence. Level basically depicts baseline value for the time series.

Here we can use automated decomposition methods such as seasonal decompose with additive or multiplicative models. An additive model is linear [ $y(t) = \text{Level} + \text{Trend} + \text{Seasonality} + \text{Noise}$ ]. It is useful when the variations around the trend does not vary with the level of the time series. Components are added together. A multiplicative model is non-linear [ $y(t) = \text{Level} * \text{Trend} * \text{Seasonality} * \text{Noise}$ ]. A non-linear seasonality has an increasing or decreasing frequency and/or amplitude over time. It is useful when the trend is proportional to the level of the time series. Components are multiplied together. From the time series plot in figure 8.3, it is clear that the trend is varying with level of time series. Therefore, here we can use a multiplicative model for decomposing.

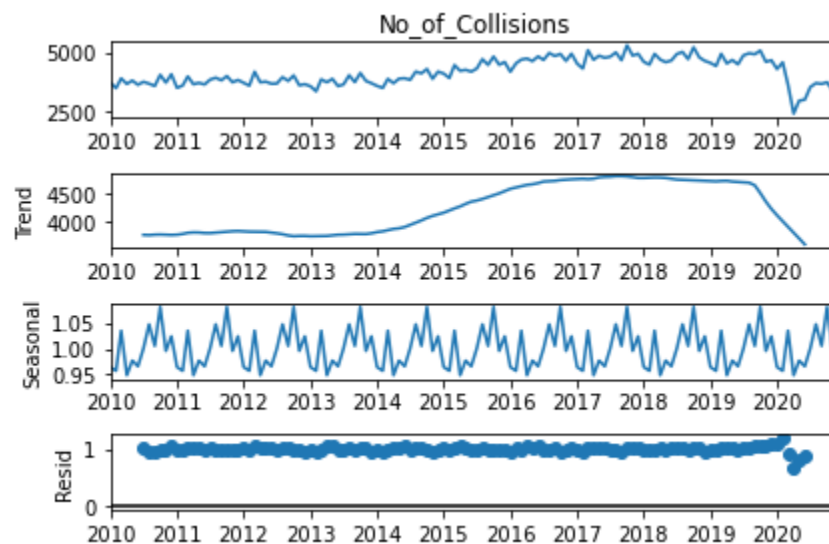


Figure8.4 Decomposition of time series data

After doing the decomposition we can understand time series components separately. From the plot it is clear that the data has seasonality. The trend is increasing till 2019. After 2019, the



trend is decreasing. Resid indicates the remaining part after extracting the trend and seasonality components. This suggests that the time series is not stationary.

## 8.2 AUTOCORRELATION

Autocorrelation is the similarity between observations as a function of the time lag between them. Much like correlation, autocorrelation gives a measure of the strength and direction of a relationship between two time series. Autocorrelation is done with a variable and its lag. It's a correlation with itself, hence autocorrelation. Basically, we are using the same time series and creating a second series, which is shifted by a time step. It is common to use the autocorrelation (ACF) plot to visualize the autocorrelation of a time-series.

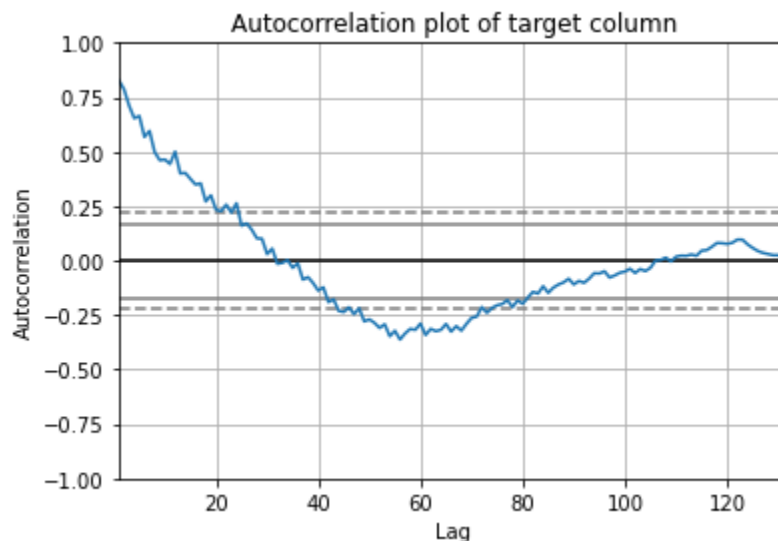


Figure8.5 Autocorrelation plot of target column

From the graph, we can see that there is a positive correlation with the first 10-to-18 lags that is perhaps significant for the first 10 lags. Here the plot displays autocorrelation for values up to 130 lags. For finding the correlation between the target column, against its previous step, we can use lag plot. The value for the same is obtained as 0.84 which indicates the correlation between the data and its lag variable by a time step shows generally low to medium correlation at 0.80. This indicates that there is a significant, positive relationship. Here the lag plot is shown in figure 8.6. The plot above shows the vacation data on the  $y(t)$  against its previous time step, previous day, the  $y(t+1)$ . we can think of the data shifted by a month and then plotted, removing the first data point. Clearly, we see a positive relationship, though it has a broader range of scatter. The correlation between the vacation data and its lag variable by a time step shows generally medium to strong correlation at 0.84. This indicates that there is a strong, positive relationship.

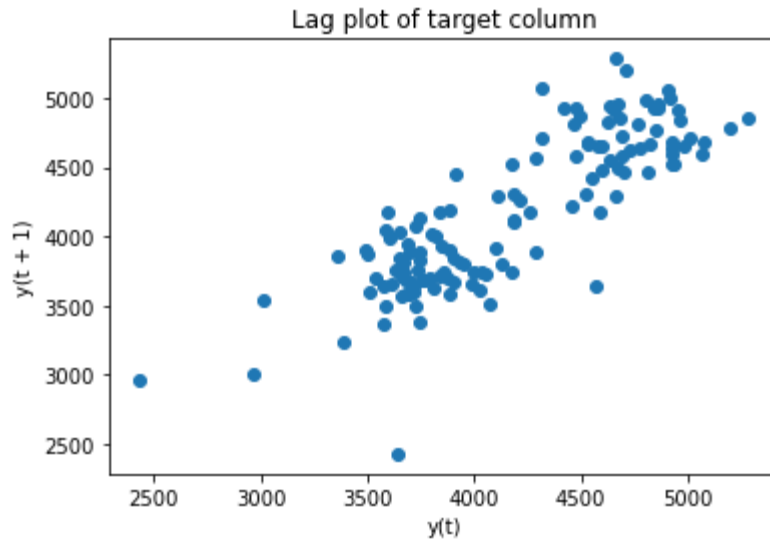


Figure8.6 Lag plot of target column

Autocorrelation can also be plotted using ACF function. An acf (Autocorrelation function) is plotted to check for autocorrelation between lags. The x-axis shows the number of lags where the y-axis shows the correlation value. Note that correlation measure runs from 0 to 1.

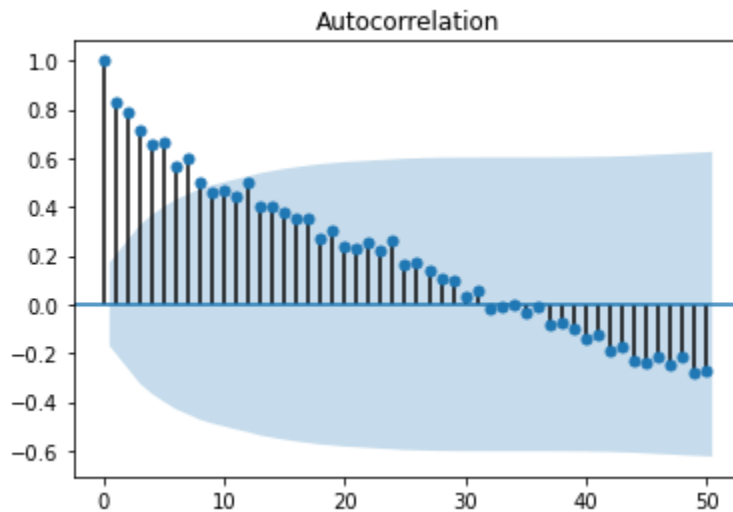


Figure8.6.1 ACF plot of target column

The results show positive and negative correlation. The scores all extend beyond the blue shaded region, which denotes statistical significance. For each time period, the measure is of its current time value's with its prior time value. It shows strong positive, autocorrelation up to 8 lags.

The partial autocorrelation function (PACF) gives the partial correlation of a time series with its own lagged values. It controls for other lags. The idea for the PACF is that we perform correlation

between a variable and itself lagged and then we subtract that effect from the variable and then find out what residual correlation is left over from that variable and further lags. For example, a PACF of order 3 returns the correlation between our time series ( $t_1, t_2, t_3, \dots$ ) and its own values lagged by 3 time points ( $t_4, t_5, t_6, \dots$ ), but only after removing all effects attributable to lags 1 and 2. If partial autocorrelation values are close to 0, then values between observations and lagged observations are not correlated with one another. Inversely, partial autocorrelations with values close to 1 or -1 indicate that there exist strong positive or negative correlations between the lagged observations of the time series. The `plot_pacf()` function also returns confidence intervals, which are represented as blue shaded regions. If partial autocorrelation values are beyond this confidence interval region, then you can assume that the observed partial autocorrelation values are statistically significant.

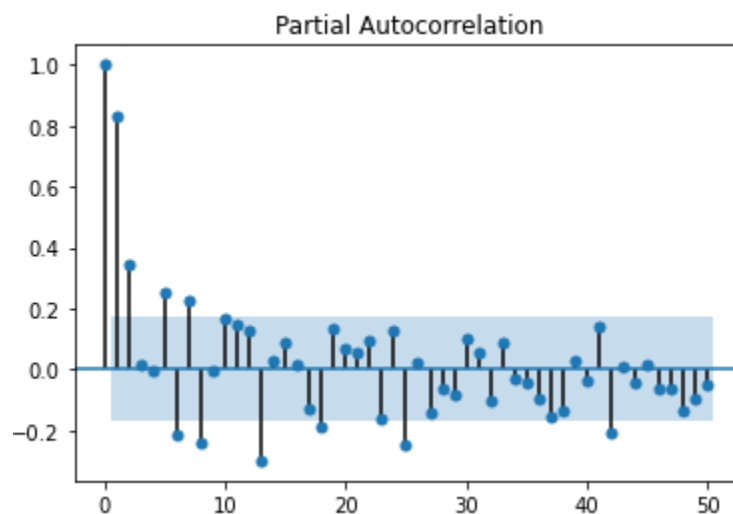


Figure8.7 PACF plot of target column

Strong partial autocorrelation at the first two lags. The candlesticks extend beyond the margin of uncertainty for lags 5 and 7 going in the positive direction. In terms of negative correlation, lag 6, 8, 13, 18 and 42 show statistical significance in terms of negative correlation.

# CHAPTER 9: MAKING TIME SERIES DATA STATIONARY

In certain situations, we need to make the data stationary, though, there are no hard and fast rules. We can remove stationarity by removing trend and seasonality such that data have constant mean and variance. In particular data domains and situations, time series should be stationary before applying any analysis. Weakly stationary data can also be acceptable. There are three characteristics of a stationary series:

1. It has a constant mean over time, i.e., the mean does not change with the passage of time.
2. It has a constant variance over time, i.e., the variance/volatility in the data does not change over time.
3. It's autocorrelation remains the same over time. Autocorrelation is a situation in which time series data is influenced by its own historical values. We will learn about this shortly.

Dickey-Fuller tests the statistical test that we run to determine if a time series is stationary or not. Without going into the technicalities of the Dickey-Fuller test, it tests the null hypothesis that a unit root is present. If it is, then  $p > 0$ , and the process is not stationary. Otherwise,  $p = 0$ , the null hypothesis is rejected, and the process is considered to be stationary.

```
ADF Test Statistic : -1.2190848806089618
p-value : 0.6653324719239092
#Lags Used : 13
Number of Observations Used : 118
weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary
```

Figure9.1 Dickey fuller test result on target column

We see that the p value is greater than 0.05 so we cannot reject the Null hypothesis. Also the test statistics is greater than the critical values. so the data is non stationary. Here we need to apply appropriate techniques to make the data stationary. We can achieve stationarity by removing trend and seasonality. Stationarity of a times series is when we have constant mean and variance. It also means that autocovariance does not depend on time. Our time series should be stationary before applying any analysis. When detrending and/or deseasonalizing a time series we may use one or a combination of approaches such as differencing, seasonal differencing, or using a transformation such as a log transformation. As we perform various operations, we will gain some intuition about how data is being transformed. There are two compelling reasons to stationarize a series:

1. Most statistical forecasting models are based on the assumption that the underlying time series can be transformed to a stationary series using mathematical transformations. It is relatively easy to make predictions on a stationary series – the idea being that you can

assume that its statistical properties will remain the same in the future as in the past! Once the prediction has been made with the stationary series, we need to untransform the series, that is, we reverse the mathematical transformations we applied to it in the past. This gives us the prediction of the original series. Therefore for making a good prediction, it is important to find the most suitable sequence of transformations needed to stationarize a time series. This provides important clues in our search for the right forecasting model. Making a series stationary is an important part of fitting an ARIMA model.

2. Stationarizing a time series also helps us get important sample statistics such as means, variances, and correlations with other variables. As described above, these statistics are useful to predict the future behavior only if the series is stationary. For example, if the series is consistently increasing over time, the sample mean and variance will grow with the size of the sample, and they will always underestimate the mean and variance in future periods. And if the mean and variance of a series are not well-defined, then neither are its correlations with other variables.

## 9.1 APPLYING LOG TRANSFORMATION

Log transformation can be used to stabilize the variance of a series with non-constant variance. This is done using the `log()` function. One limitation of log transformation is that it can be applied only to positively valued time series. Taking a log shrinks the values towards 0. For values that are close to 1, the shrinking is less and for the values that are higher, the shrinking is more, thus reducing the variance.

Log transformation is applied to make the data stationary. Then ADF test static values are printed. But, the data is non stationary even after applying the log transformation

```
ADF Test Statistic : -1.1525275508903143
p-value : 0.6936768108315072
#Lags Used : 12
Number of Observations Used : 119
weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary
```

Figure9.2 Dickey fuller test result on log transformation of target column

Then the time series plot is obtained on the transformed column to understand change in the time series data. Here the values in the y axis is changed from maximum value 5000 to a maximum value of 8.6. Also from the graph it is clear that the data is transformed in such a way that it has constant variance.

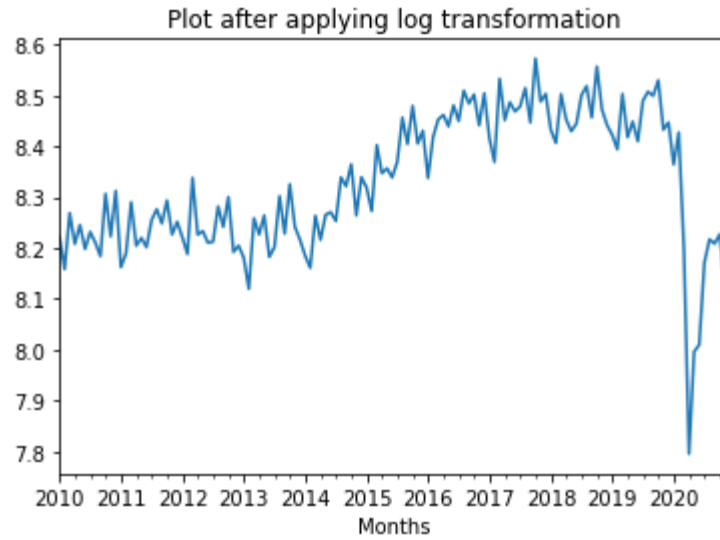


Figure9.3 Plot on log transformation of target column

The decomposed plot is obtained as shown below:

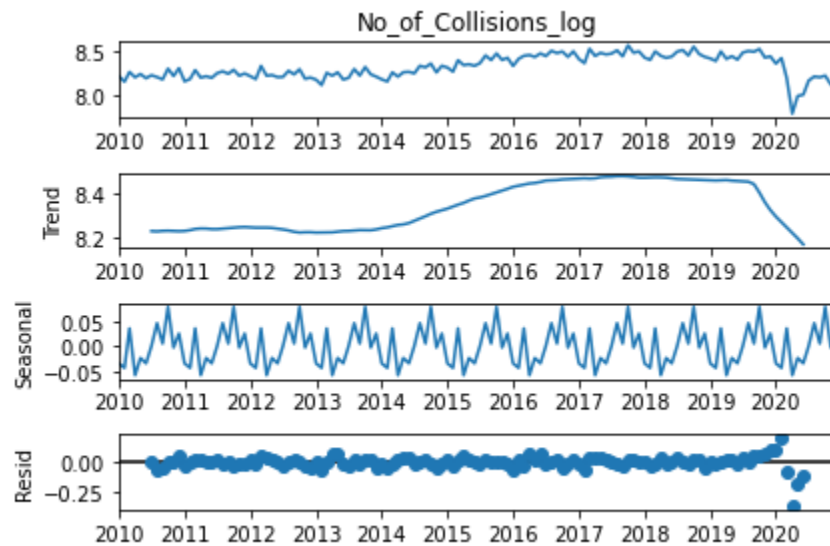


Figure9.4 Decomposition plot on log transformation of target column

From the decomposition plot, it is clear that the values in y axis is decreased after applying the transformation.

## 9.2 APPLYING FIRST DIFFERENCING

Under this technique, we difference the data. That is, given the series  $Z(t)$ , we create the new series:

$$Y(t) = Z(t) - Z(t-1).$$

The differenced data will contain one less point than the original data. Differencing a time series can remove a linear trend from it.

---

```
ADF Test Statistic : -1.623741145326671
p-value : 0.4707685095042713
#Lags Used : 11
Number of Observations Used : 119
weak evidence against null hypothesis, time series has a unit root, indicating it is non-stationary
```

---

Figure9.8 Dickey fuller test result on first differencing of log transformation of target column

From the test, it is clear that data is not stationary even after applying this transformation.

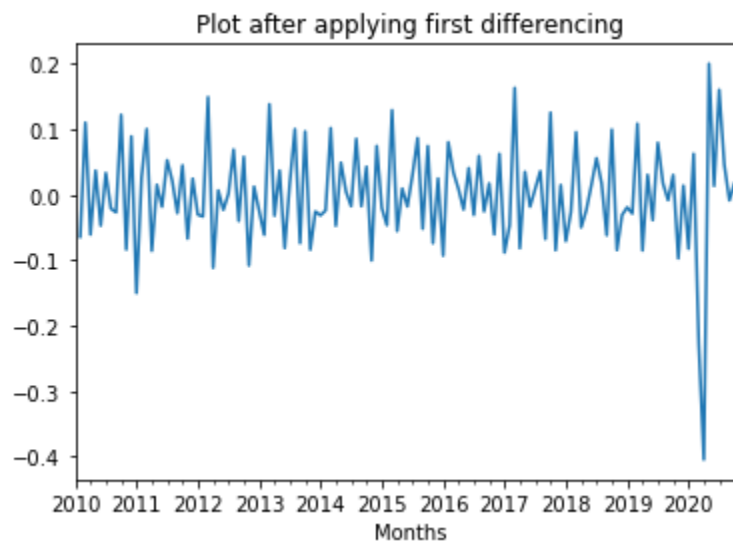


Figure9.5 Plot on first differencing of log transformation of target column

From the time plot, it is clear that the data has a linear trend after applying the first differencing. In this case we can use additive modelling technique for the decomposition of the time series. The output of decomposition is given in figure 9.10.

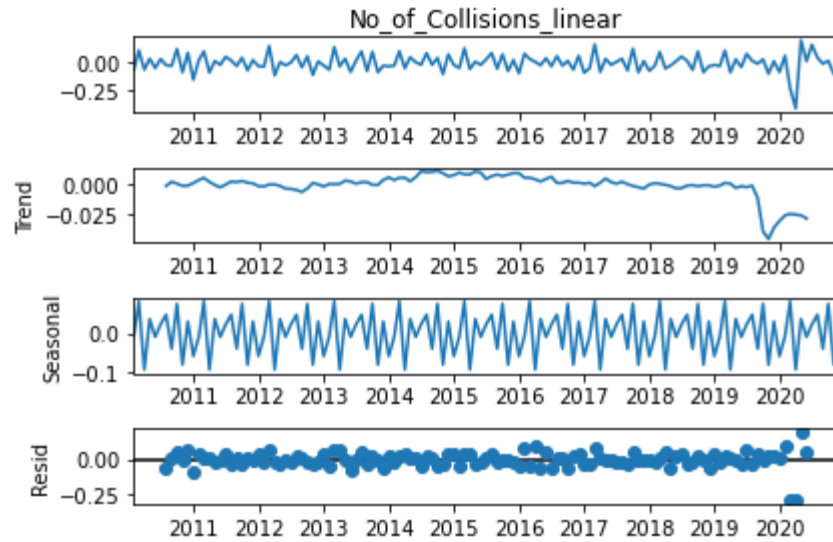


Figure9.6 Decomposition plot on first differencing of log transformation of target column

Here a change in the seasonality and y axis values can be noted

### 9.3 APPLYING SEASONAL DIFFERENCING

If a series has seasonality present in it, then we can use seasonal differencing to remove these periodic patterns. For monthly data, in which there are 12 periods in a season, the seasonal difference of  $Y$  at period  $t$  is  $Y(t) - Y(t-12)$ . for quarterly data, the difference will be based on a lag of 4 data points.

```
ADF Test Statistic : -9.500546281716707
p-value : 3.4545989856931147e-16
#Lags Used : 1
Number of Observations Used : 117
strong evidence for the null hypothesis, reject the null hypothesis. Data has no unit root and is stationary
```

Figure9.7 Dickey fuller test result on seasonal differenced data

After applying the seasonal differencing, the data is made stationary. Now, we applied a combination of log transformation, first differencing and seasonal differencing to make the data stationary. The plots obtained on the stationary data is shown in figure 9.8 and 9.9.



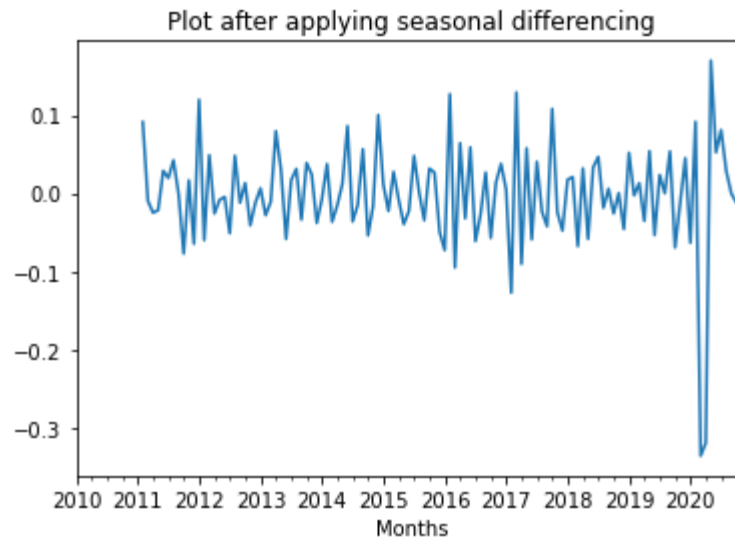


Figure9.8 Plot on first differencing of log transformation of target column

From the above plot, it is clear that the stationary data has a linear trend. The seasonality in the above plot is also not visible.

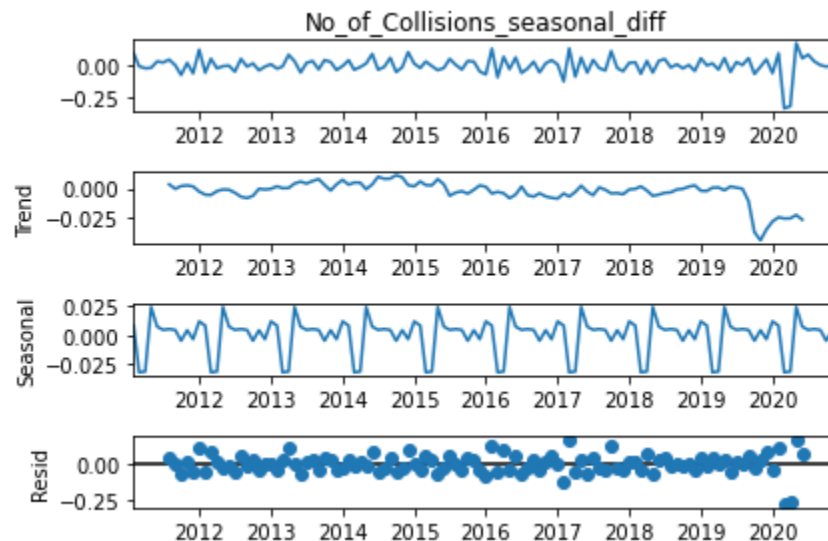


Figure9.9 Decomposition plot on first differencing of log transformation of target column

After decomposition using additive modelling, we can still see a seasonal component. Then an ACF and PACF is plotted for the stationary data.

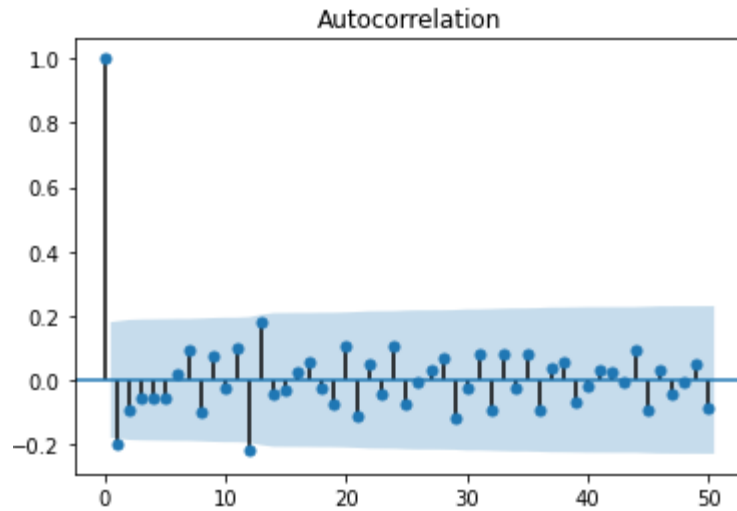


Figure9.10 ACF of stationary data

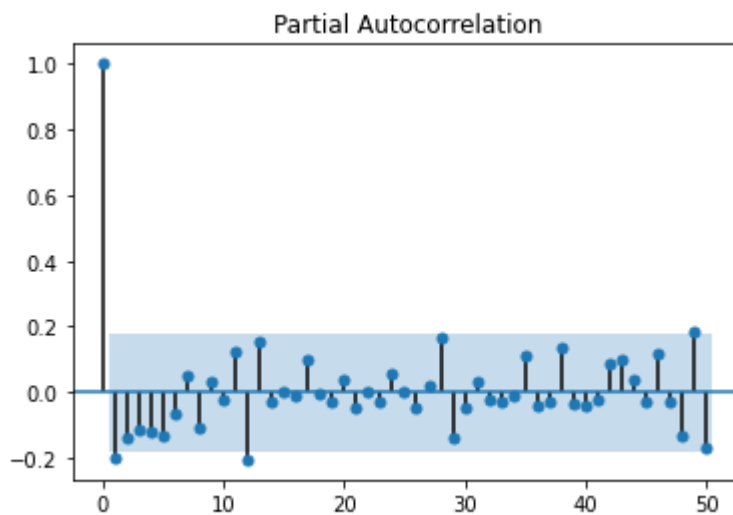


Figure9.11 PACF of stationary data

From the PACF and ACF plots the values of  $p$  and  $q$  can be 1. Here after applying the data transformation techniques, it is clear that the  $d$  value is 1, which is the no.of log differencing made to make the data stationary.

# CHAPTER 10: MODELLING AND EVALUATION

From the time series data visualisation, it is clear that the data is seasonal. Here we obtained values of (p,d,q) after applying the transformations to make the data stationary. We can also use grid search to find the best values. Here we implemented ARIMA, seasonal ARIMA and FB prophet models. ARIMA is not applicable in this case. The model is applied to find the difference between the result in ARIMA and seasonal ARIMA.

## 10.1 ARIMA MODEL

ARIMA is one of the most popular time series forecasting models and as its name indicates is made up of three terms:

1. **AR:** Stands for **autoregression**, which is nothing more than applying a linear regression algorithm using one observation and its own lagged observations as training data.

The AR model uses the following formula:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t$$

Where  $\phi_i$  are the weights of the models learned from the previous observations and  $\epsilon_t$  is the residual error for observation  $t$ . We also call  $p$  the order of the autoregression model, which is defined as the number of lag observations included in the preceding formula. For example:

**AR(1)** is defined as:  $Y_t = \phi_1 Y_{t-1} + \epsilon_t$

**AR(2)** is defined as:  $Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \epsilon_t$

2. **I:** Stands for **integrated**. For the ARIMA model to work, it is assumed that the time series is stationary or can be made stationary. A series is said to be stationary if its mean and variance doesn't change over time. We can make a time series stationary through a transformation that uses differencing of the log between an observation and the one before that, as shown in the following equation:

$$Z_t = \log Y_t - \log Y_{t-1}$$

It is possible that multiple log differencing transformations are needed before the time series is actually made stationary. We call **d**, the number of times we transform the series using log differencing. For example:

**I(0)** is defined as no log differencing needed (the model is then called ARMA).

**I(1)** is defined as 1 log differencing needed.

**I(2)** is defined as 2 log differencing needed.

3. **MA**: Stands for **moving average**. The MA model uses the residual error from the mean of the current observation and the weighted residual errors of the lagged observations. We can define the model using the following formula:

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}$$

Where  $\mu$  is the mean of the time series,  $\epsilon_t$  are the residual errors in the series and  $\theta_q$  are the weights for the lagged residual errors. We call **q** the size of the moving average window. For example:

**MA(0)** is defined as no moving average needed (the model is then called AR).

**MA(1)** is defined as using a moving average window of 1. The formula becomes:

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1}$$

As per the preceding definition, we use the notation **ARIMA(p,d,q)** to define an ARIMA model. The parameters of the ARIMA model are defined as follows:

- **p**: The number of lag observations included in the model, also called the lag order.
- **d**: The number of times that the raw observations are differenced, also called the degree of differencing.
- **q**: The size of the moving average window, also called the order of moving average.

A value of 0 can be used for a parameter, which indicates to not use that element of the model. This way, the ARIMA model can be configured to perform the function of an ARMA model, and even a simple AR, I, or MA model.

Implementing all the code to build an ARIMA model can be very time-consuming. Fortunately, the statsmodels library implements an ARIMA class in the statsmodels.tsa.arima\_model package that provides all the computation needed to train a model with the fit() method and predict values

with the `predict()` method. It also takes care of the log differencing to make the time series stationary. The trick is to find the parameters **p**, **d**, and **q** for building the optimal ARIMA model. For this, we use the ACF and PACF chart as follows:

- The **p** value corresponds to the number of lags (on the **x** abscissa) where the PACF chart crosses the statistical significance threshold for the first time.
- Similarly, the **q** value corresponds to the number of lags (on the **x** abscissa) where the ACF chart crosses the statistical significance threshold for the first time.

Identification of an AR model is often best done with the PACF. For an AR model, the theoretical PACF “shuts off” past the order of the model. The phrase “shuts off” means that in theory the partial autocorrelations are equal to 0 beyond that point. Put another way, the number of non-zero partial autocorrelations gives the order of the AR model. By the “order of the model” we mean the most extreme lag of **x** that is used as a predictor.

Identification of an MA model is often best done with the ACF rather than the PACF. For an MA model, the theoretical PACF does not shut off, but instead tapers toward 0 in some manner. A clearer pattern for an MA model is in the ACF. The ACF will have non-zero autocorrelations only at lags involved in the model.

Here first a grid search is conducted to get AIC value for different values of **p**, **d** and **q**. The output is obtained as follows

```
ARIMA(0, 0, 0) - AIC:2042.1449530507878
ARIMA(0, 0, 1) - AIC:1968.1714816937558
ARIMA(0, 1, 0) - AIC:1877.7082425573801
ARIMA(0, 1, 1) - AIC:1856.101006663579
ARIMA(1, 0, 0) - AIC:1884.8416958473683
ARIMA(1, 0, 1) - AIC:1870.3880467114125
ARIMA(1, 1, 0) - AIC:1855.1780685167919
ARIMA(1, 1, 1) - AIC:1857.0931024849115
```

Figure10.1 Grid search result for ARIMA model

From the output, the values of **p**, **d**, **q** for minimum AIC value are obtained as **p=1**, **d=1** and **q=0**. A model is created using these values. The original collision and prediction is obtained as shown in the graph. From the graph, it is clear that we can apply the ARIMA model for seasonal data. The RMSE value is obtained as 4348.797

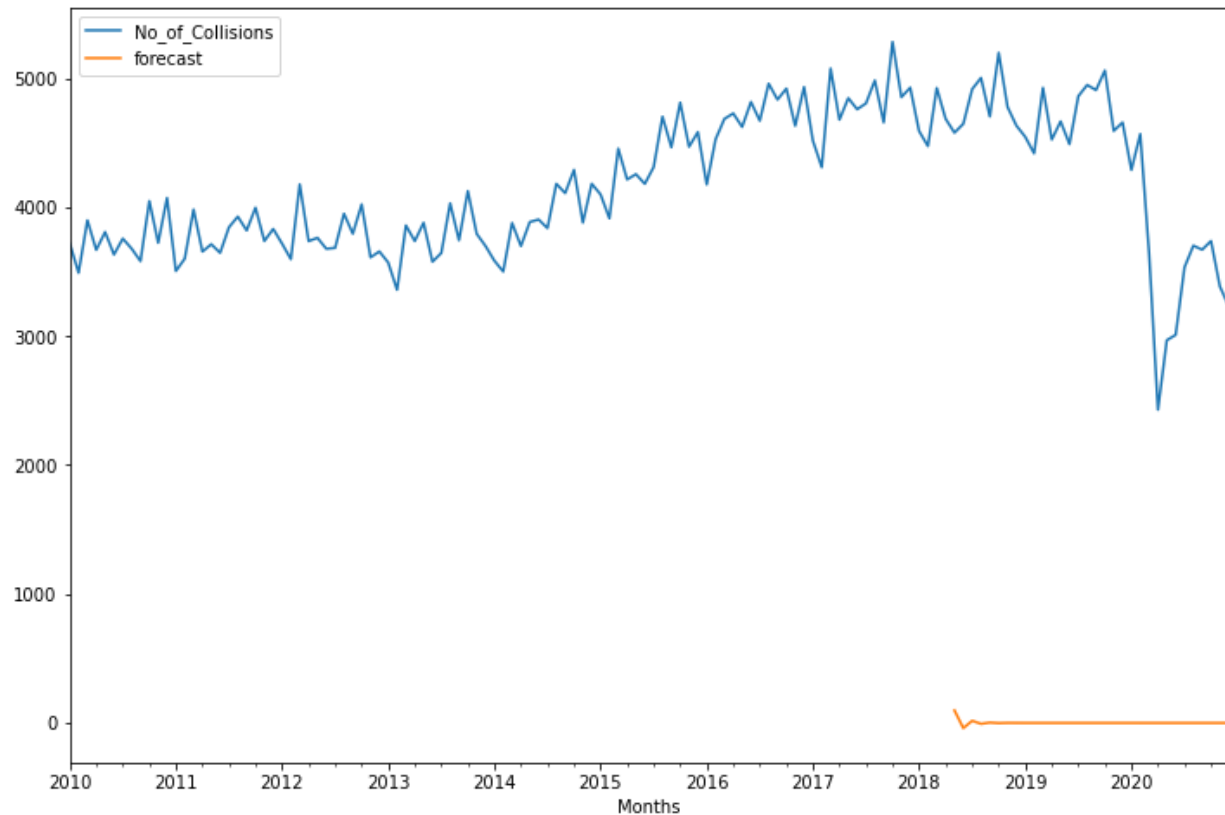


Figure10.2 Prediction and original values for ARIMA model

## 10.2 SEASONAL ARIMA MODEL

An extension to ARIMA that supports the direct modeling of the seasonal component of the series is called SARIMA. Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

**Trend Elements:** There are three trend elements that require configuration. They are the same as the ARIMA model; specifically:

- **p:** Trend autoregression order.
- **d:** Trend difference order.
- **q:** Trend moving average order.

**Seasonal Elements:** There are four seasonal elements that are not part of ARIMA that must be configured, they are:

- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period.

Together, the notation for an SARIMA model is specified as: SARIMA(p,d,q)(P,D,Q)m

Here first a grid search is conducted to get AIC values for different values of (p,d,q) and (P,D,Q). Here the value of m=12 since the data is yearly seasonal. The output is obtained as follows:

ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:2560.1606647905855  
 ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:2185.76925294709  
 ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:1811.4782300489076  
 ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:1638.1319606411864  
 ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:1826.8430168029813  
 ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:1813.35426699815  
 ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:1650.6835504612447  
 ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:1638.0107687911393  
 ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:2449.8499741945634  
 ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:2034.732741110552  
 ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:1728.0284188150295  
 ARIMA(0, 0, 1)x(0, 1, 1, 12)12 - AIC:1562.1332073020228  
 ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:1758.2040012525604  
 ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:1730.924303607983  
 ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:1587.929718598531  
 ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:1561.5303323807475  
 ARIMA(0, 1, 0)x(0, 0, 0, 12)12 - AIC:1861.8637891240114  
 ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:1667.3039483631128  
 ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:1649.5140694407094  
 ARIMA(0, 1, 0)x(0, 1, 1, 12)12 - AIC:1470.52964530674  
 ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:1651.1147445427796  
 ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:1631.0945375696347  
 ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:1485.1522375251195  
 ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:1472.4471672379354  
 ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:1825.845598498955  
 ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:1640.0190733898723  
 ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:1624.8275398810624  
 ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:1454.9588695653088  
 ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:1642.7109054064404



```

ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:1614.336298481281
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:1480.8716225014368
ARIMA(0, 1, 1)x(1, 1, 1, 12)12 - AIC:1456.4303933726542
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:1877.442151612992
ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:1682.2465131730205
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:1660.6434637033144
ARIMA(1, 0, 0)x(0, 1, 1, 12)12 - AIC:1483.2339259247808
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:1651.9301000085227
ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:1644.6323705445352
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:1484.389995453349
ARIMA(1, 0, 0)x(1, 1, 1, 12)12 - AIC:1485.2325871269354
ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:1842.7564074015527
ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:1656.4941733486546
ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:1639.3585194979064
ARIMA(1, 0, 1)x(0, 1, 1, 12)12 - AIC:1469.6685600006656
ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:1644.326998305919
ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:1627.5852766945904
ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:1482.9999257125564
ARIMA(1, 0, 1)x(1, 1, 1, 12)12 - AIC:1471.6685439925427
ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:1839.3425602259344
ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:1655.02161456787
ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:1640.4039151406432
ARIMA(1, 1, 0)x(0, 1, 1, 12)12 - AIC:1469.4660008691399
ARIMA(1, 1, 0)x(1, 0, 0, 12)12 - AIC:1630.930231033749
ARIMA(1, 1, 0)x(1, 0, 1, 12)12 - AIC:1628.7834599562025
ARIMA(1, 1, 0)x(1, 1, 0, 12)12 - AIC:1469.7954207664181
ARIMA(1, 1, 0)x(1, 1, 1, 12)12 - AIC:1471.327504701582
ARIMA(1, 1, 1)x(0, 0, 0, 12)12 - AIC:1826.9160816966569
ARIMA(1, 1, 1)x(0, 0, 1, 12)12 - AIC:1641.826682032726

ARIMA(1, 1, 1)x(0, 1, 0, 12)12 - AIC:1625.8615539993582
ARIMA(1, 1, 1)x(0, 1, 1, 12)12 - AIC:1455.1676997593745
ARIMA(1, 1, 1)x(1, 0, 0, 12)12 - AIC:1631.2323442593613
ARIMA(1, 1, 1)x(1, 0, 1, 12)12 - AIC:1614.6376417391723
ARIMA(1, 1, 1)x(1, 1, 0, 12)12 - AIC:1468.086350153081
ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - AIC:1456.6865078861736

```

Figure10.3 Grid search result for seasonal ARIMA model

Here the best model is obtained as trend order=(0, 1, 1) and seasonal order=(0, 1, 1, 12) with a AIC value of 1454.95. The model is the created using these values and the predicted and original outputs are obtained as follows:

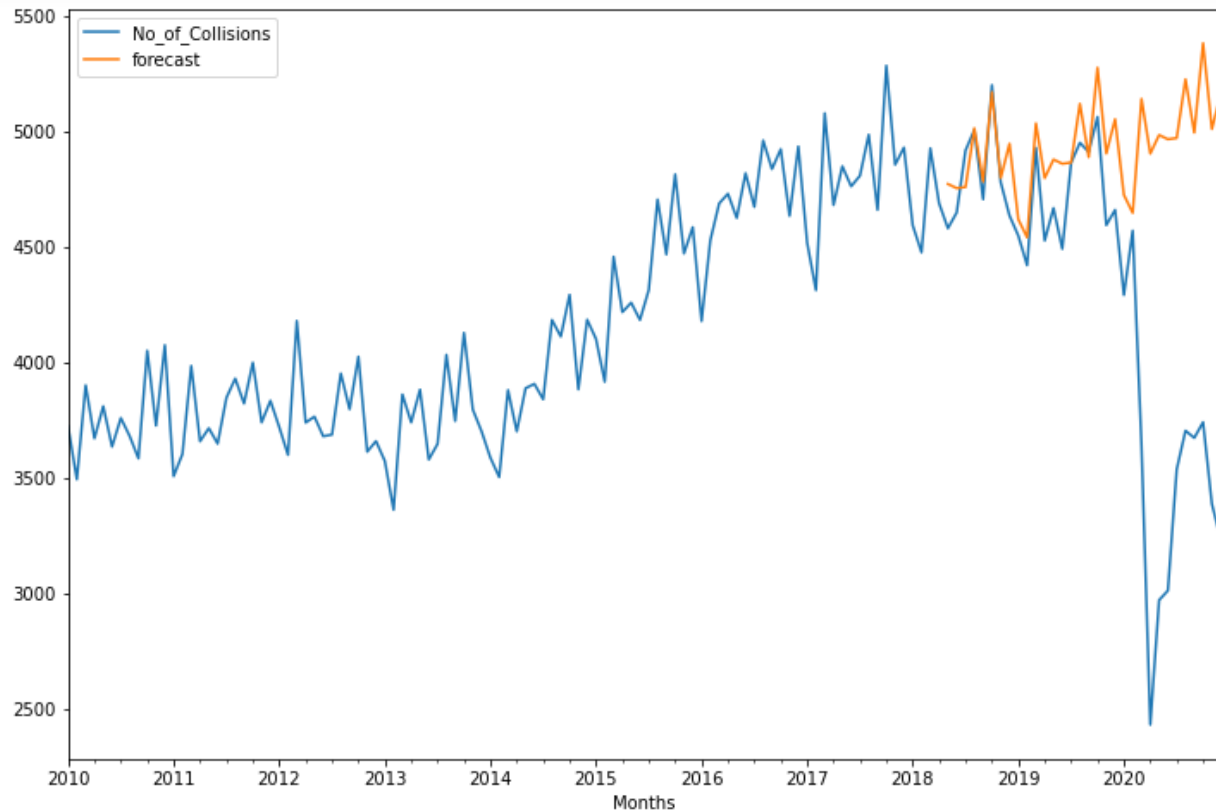


Figure10.4 Forecast and original no.of collisions for seasonal ARIMA model

Here the prediction looks like a case neglecting the effect of corona pandemic. Here the obtained value of RMSE is 1006.572

### 10.3 FB PROPHET MODEL

Prophet is an open source library published by Facebook that is based on **decomposable (trend+seasonality+holidays) models**. It provides us with the ability to make time series predictions with good accuracy using simple intuitive parameters and has support for including impact of custom seasonality and holidays. We use a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- **g(t)**: piecewise linear or logistic growth curve for modelling non-periodic changes in time series

- **s(t)**: periodic changes (e.g. weekly/yearly seasonality)
- **h(t)**: effects of holidays (user provided) with irregular schedules
- **εt**: error term accounts for any unusual changes not accommodated by the model

Using time as a regressor, Prophet is trying to fit several linear and nonlinear functions of time as components.

**Trend:** Trend is modelled by fitting a piecewise linear curve over the trend or the non-periodic part of the time series. The linear fitting exercise ensures that it is least affected by spikes/missing data.

**Seasonality:** To fit and forecast the effects of seasonality, prophet relies on fourier series to provide a flexible model. Seasonal effects  $s(t)$  are approximated by the following function:

$$s(t) = \sum_{n=1}^N \left( a_n \cos \left( \frac{2\pi nt}{P} \right) + b_n \sin \left( \frac{2\pi nt}{P} \right) \right)$$

P is the period (365.25 for yearly data and 7 for weekly data). Parameters  $[a_1, b_1, \dots, a_N, b_N]$  need to be estimated for a given N to model seasonality. The fourier order N that defines whether high frequency changes are allowed to be modelled is an important parameter to set here. For a time series, if the user believes the high frequency components are just noise and should not be considered for modelling, he/she could set the values of N from to a lower value. If not, N can be tuned to a higher value and set using the forecast accuracy.

**Holidays and events:** Holidays and events incur predictable shocks to a time series. For instance, Diwali in India occurs on a different day each year and a large portion of the population buy a lot of new items during this period.

Prophet allows the analyst to provide a custom list of past and future events. A window around such days are considered separately and additional parameters are fitted to model the effect of holidays and events. Prophet() function is used to define a Prophet forecasting model in Python. The most important parameters are:

### Trend parameters

- growth : linear' or 'logistic' to specify a linear or logistic trend
- changepoints : List of dates at which to include potential changepoints (automatic if not specified)
- n\_changepoints : If changepoints in not supplied, you may provide the number of

change points to be automatically included

- `change_point_prior_scale` : Parameter for changing flexibility of automatic change point selection

### **Seasonality & Holiday Parameters**

- `yearly_seasonality` : Fit yearly seasonality
- `weekly_seasonality` : Fit weekly seasonality
- `daily_seasonality` : Fit daily seasonality
- `holidays` : Feed dataframe containing holiday name and date
- `seasonality_prior_scale` : Parameter for changing strength of seasonality model
- `holiday_prior_scale` : Parameter for changing strength of holiday model

`yearly_seasonality`, `weekly_seasonality` & `daily_seasonality` can take values as True, False and no of fourier terms which was discussed in the last section. If the value is True, the default number of fourier terms (10) are taken. Prior scales are defined to tell the model how strongly it needs to consider the seasonal/holiday components while fitting and forecasting.

Before implementing the prophet model, the date is changed from index into a column named 'ds' and target column is renamed as 'y'. The model is then created using the edited dataset. Then prediction is made by making a dataframe of future dates along with historical dates. The obtained prediction contained many values. Here the desired prediction is the yhat value. The output is obtained as shown in figure

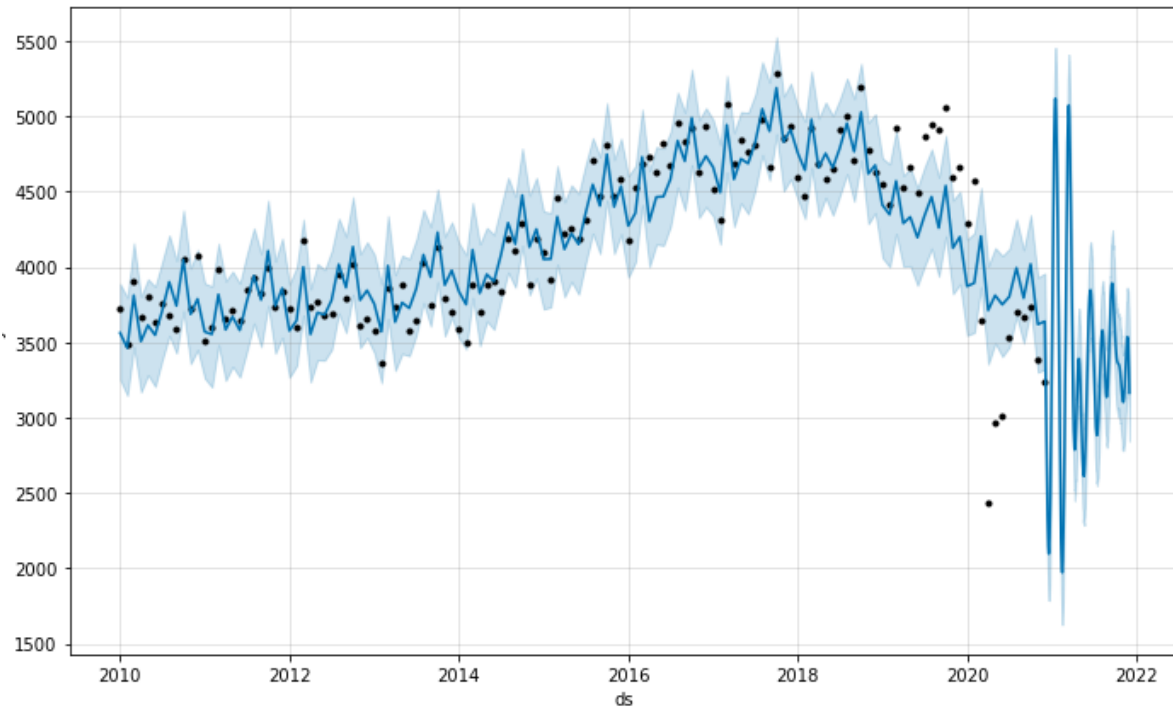


Figure10.5 Forecast on no.of collisions for FB prophet model

Here the black color dots indicate residues and the blue color line indicates the prediction. A cross validation method is applied to the model to get the RMSE values and the result is obtained as shown in the graph.

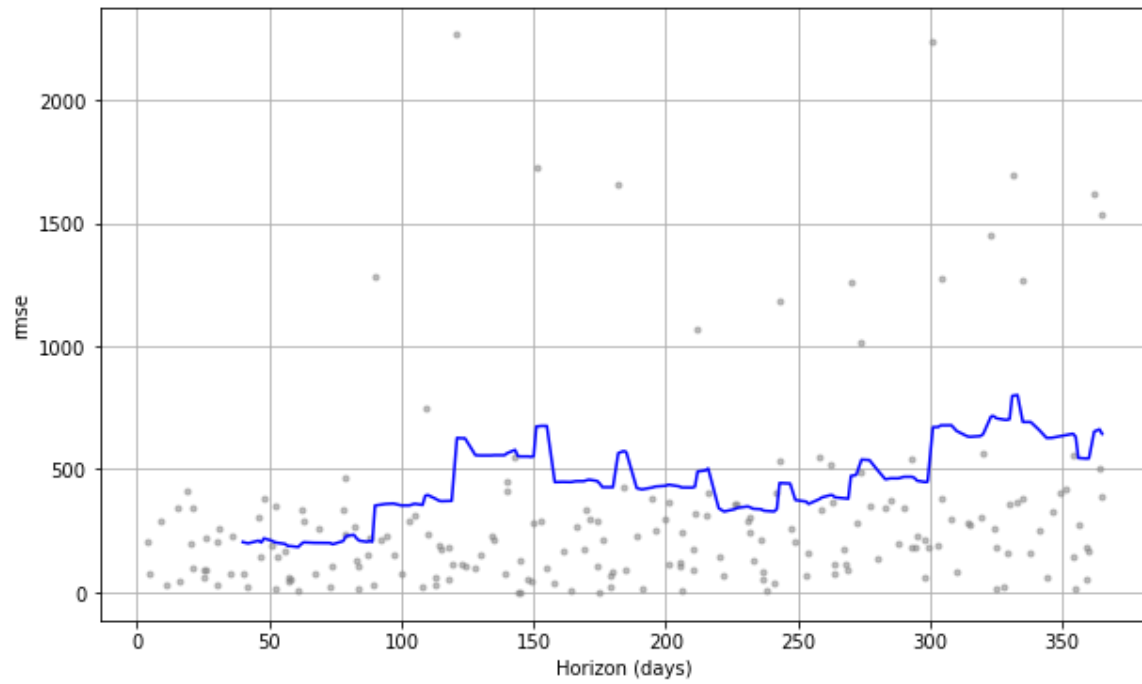


Figure10.6 RMSE for FB prophet model

Here the RMSE value is less than 1000 for all cases. That is this model has a lower value of RMSE compared to seasonal ARIMA. So we can use this model for building the prediction app.

# CHAPTER 11: WEB DEPLOYMENT

A flask app was developed using the fb prophet model. Using HTML and CSS other pages were created. Here the website is launched using pythonanywhere.com

Website link : <http://latrafficcollisionanalysis.pythonanywhere.com/>

## CHAPTER 12: CONCLUSION

Through this project, an attempt has been made to explore time series models for road accident data understanding and analysis. This method enabled to predict the no.of monthly collisions based on the time series models. The main goal was to empirically explore data quality issues, data analysis based on available features, trend analysis and to identify the role of time series hidden patterns, which is said to be the major factor to analyse the traffic collision incident based on time. Detection of accidents risks due to road users related factors could assist in designing appropriate countermeasures in the effort of reducing the socio-economic impact of road accidents which ultimately improve road safety. Another advantage of this approach to road traffic accident data understanding and analysis through machine learning is that hypotheses can be easily formulated for future trends.

In addition to revealing patterns related with road users factors for accident severity, a major contribution of this work includes comparison of time series predictive models for the future collision prediction. We strongly believe that the result of this project will be a major ingredient of the information architecture to be proposed for accident data analysis in Los Angeles. The result of this project will help road safety organizations to revisit their focus of attention in crafting and implementing measures to reduce road safety danger. More specifically the project indicated that in addition to time, victim age, area, victim descent, premise and other factors should be considered while taking enforcement measures. This should address other road users like pedestrians also since collisions are occurring more in streets.

The result can also be used as a hypothesis and/or replicated to other developing countries with similar context in the area of road accident data collection and analysis. Finally the result of this study can also be used to support future research related to machine learning approach, especially in the context of road safety.



## REFERENCES

1. Time Series Forecast Study with Python: Monthly Sales of French Champagne (machinelearningmastery.com)
2. Knowledge Discovery from Road Traffic Accident Data in Ethiopia: Data Quality, Ensembling and Trend Analysis for Improving Road Safety by Tibebe Beshah , Dejene Ejigu , Ajith Abraham , Václav Snášel , Pavel Krömer
3. An End-to-End Project on Time Series Analysis and Forecasting with Python | by Susan Li | Towards Data Science
4. Time Series Analysis in Python - A Comprehensive Guide with Examples - ML+ (machinelearningplus.com)