Vancouver Crime Analysis

Final Report

SS-9850

1. Introduction

Regarded as the most promising city to live in entire North America, Vancouver is like a world in a nutshell. Be it the food, beaches, parks adventure activities, and landscapes, the city entices tourists from across the globe. With lofty standards of living and a warmer climate across the year, thousands of immigrants emigrate to Vancouver to study or work. On the contrary, poverty, parental neglect, low self-esteem, theft, or assaults, make the city a residence of an ever-increasing number of homeless people and criminals. The nature of the crime is unanticipated and particularly depends on the time of the year, weather, and social and economic factors. The crime rate in Vancouver is growing every year which drives the process of analysis and predictions for the Police department rather intricate.

Police departments and intelligence agencies around the world expend millions of dollars on crime investigations and predictions but fail to foretell most of the illicit incidents. In this project, our main aim is to aid the crime bureau with advanced statistical tools and techniques that can assist them to forecast any unlawful activity across the city. Historical data collected from the Vancouver Crime department is pre-processed and then used to visualize patterns through graphical plots and charts to accurately forewarn upcoming criminalities in Vancouver. In this project, we used a time-series forecasting based model called Prophet that enables law officials to develop a crime mapping model to locate the areas that are more prone to such illegal exercises.

2. Dataset Source and Description

We have taken the dataset from Kaggle with the name “Crime in Vancouver”. Kaggle is a platform that allows users to find and publish datasets. The original dataset comes from Vancouver’s open data catalog, provided by the Vancouver Police Department. The compiled dataset on Kaggle was uploaded 4 years ago in 2017. Therefore, the dataset report carries information from January 1st, 2003 until December 31st, 2017. Since the data is updated every Sunday morning, we have further extended the compiled dataset from 2018 to 2022 by extracting the new data from the original source. Listed below are the attributes available in the dataset(**Table 1**). It provides information on the type of crime committed and the date, time, and location of the offense. The time of the crime occurrence is reported up to minute precision. The Spatial information includes neighborhood, hundred block, and UTM coordinate features. The dataset covers 24 localities including 22 official neighborhoods, and Stanley Park and Musqueam part of Vancouver city. As of April 15, 2022, all CSV files combined have 701,449 rows and 10 columns(or attributes), where each row represents a

single instance of crime. According to the VPD website, specific filters are applied to ensure the data is relevant to public safety and adheres to the BC Freedom of Information & Protection of Privacy Act. Therefore, crimes like “Offence Against a Person” and “Homicide” are considered sensitive and they only have *year, month,* and *day* attributes, omitting values for *hour, minute,* and *neighbourhood.*

Table 1. Attributes of the Dataset

TYPE	YEAR	MONTH	DAY	HOUR	MINUTE	HUNDRED_BLOCK	NEIGHBOURHOOD	X	Y
Break and Enter Commercial	2018	6	16	18	0	10XX ALBERNI ST	West End	491102.2	5459092
Break and Enter Commercial	2018	12	12	0	0	10XX BEACH AVE	West End	490228.8	5458208
Break and Enter Commercial	2018	4	9	6	0	10XX BEACH AVE	Central Business District	490249.2	5458167
Break and Enter Commercial	2018	10	2	18	31	10XX BEACH AVE	Central Business District	490258.4	5458155
Break and Enter Commercial	2018	2	17	15	0	10XX BEACH AVE	Central Business District	490269.9	5458141
Break and Enter Commercial	2018	5	16	17	0	10XX BOUNDARY RD	Hastings-Sunrise	498275.6	5458125

3. Methods

Exploratory Data Analysis

Exploratory Data Analysis is a critical part of initial investigations of data. The motive is to get useful insights out of data, discover patterns, and test hypotheses that are essential for the business and stakeholders. In simple words, EDA involves analyzing datasets to summarise the main characteristics, often with **visual methods** such as uni-variate visualization, bi-variate visualization, multivariate visualization, and **Dimensionality reduction methods**. In this report, the emphasis is primarily on visualization techniques.

Once EDA is complete and insights are drawn, we can define and refine our important features variable selection, which can be used for supervised and unsupervised machine learning modeling.

We took the following steps involved in EDA

1. Frame(or Ask) Questions
2. Data Collection
3. Data Cleaning
4. Data Preprocessing

5. Data Visualisation & Analysis

Frame Questions

Before performing any analysis, the very first step in EDA is to ask meaningful questions that could be answered using data. The quality of any analysis performed on data depends on the ability to frame the right questions. Exploring the data helped us understand the scope of the data, which helped us frame questions.

Questions that we asked to answer from this case study
1. Which neighborhoods are mentioned in this dataset
2. How many crimes were committed in these neighborhoods? What kinds of crimes are more prevalent in Vancouver as a whole and across the neighborhoods?
3. Which year had the most crimes? Has crime decreased or increased over years?
4. What types of crimes were committed?
5. Are there certain times when a crime is more likely to occur?
6. What types of crimes are most frequent at each hour of the day?

Data Collection

The data collection is thoroughly explained in the data description. The data used is contained in multiplied comma-separated value(CSV) files which were merged into one CSV file.

Data Cleaning

Data cleaning refers to ensuring that the data is correct and useable by identifying any errors such as missing values and outliers and correcting them. Our dataset contains missing values owing to the fact that VPD purposely removed attributes "*hour*", "*minute*" and "*neighborhood*" of "*crimes Offence Against a Person*" and "*Homicide*" for safety reasons. These entries were basically useless in the initial analysis and therefore, these rows were deleted from the dataset. The data was neat and contained only one outlier. Thus, not many cleaning steps were required.

Data Preprocessing

This step involves transforming raw data into understandable and suitable for modeling and visualization. In this step, we have merged "*Year*", "*Month*" and "*Day*" attributes into "Date" for making visualization easier. For a better understanding of the data, we also used a clustering technique as a part of the initial data analysis. To cluster the data, we put all the **quantitative features** on the same scale using Standardization before feeding them to the clustering model.

Data Visualization & Analysis

Data visualization is the graphical representation of information and data. This is where we start exploring the data and get answers to the questions we framed in step 1. We used utilized graphical and dimensionality reduction techniques for EDA.

For graphical representations of our data, we used *Scatter plots, Bar charts, Line charts*, *Multi-line charts*, and *heat maps* to extract useful information from the data. We also used the *K-Means clustering algorithm* to partition groups of data points into smaller clusters. Clustering helped us understand the natural grouping of the dataset and identify a hidden pattern that didn't emerge from typical graphical techniques. We primarily used two R packages for Data Visualization

ggplot2 - An open-source data visualization package for the statistical programming language R

dplyr - One of the core packages of the *tidyverse* in the R programming language, dplyr is primarily a set of functions designed to enable data frame manipulation

Main Data Analysis

The city of Vancouver is known for its world-class quality of life. Economic and political stability, universal healthcare, diversity in culture and environment, as well as world-class education infrastructure are just a few of the reasons Vancouver has consistently been listed as one of the best cities to live in the world. Therefore, safety is a key component of this measurement. Based on this, one question kept popping into my head while working on EDA

Is it possible to accurately forecast the crime rate in Vancouver?

Therefore, the objective of this project is to predict the daily crime rate in Vancouver using a historical dataset. One approach is to use a predictive model to forecast the “future” after the change point and see if the actual data line up with the forecast. In this report, we have trained a time series predictive algorithm developed by Facebook called Prophet.

In 2017, Facebook Core Data Science Team open-sourced Prophet. As stated on its Github page, *“Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.”*

It is a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

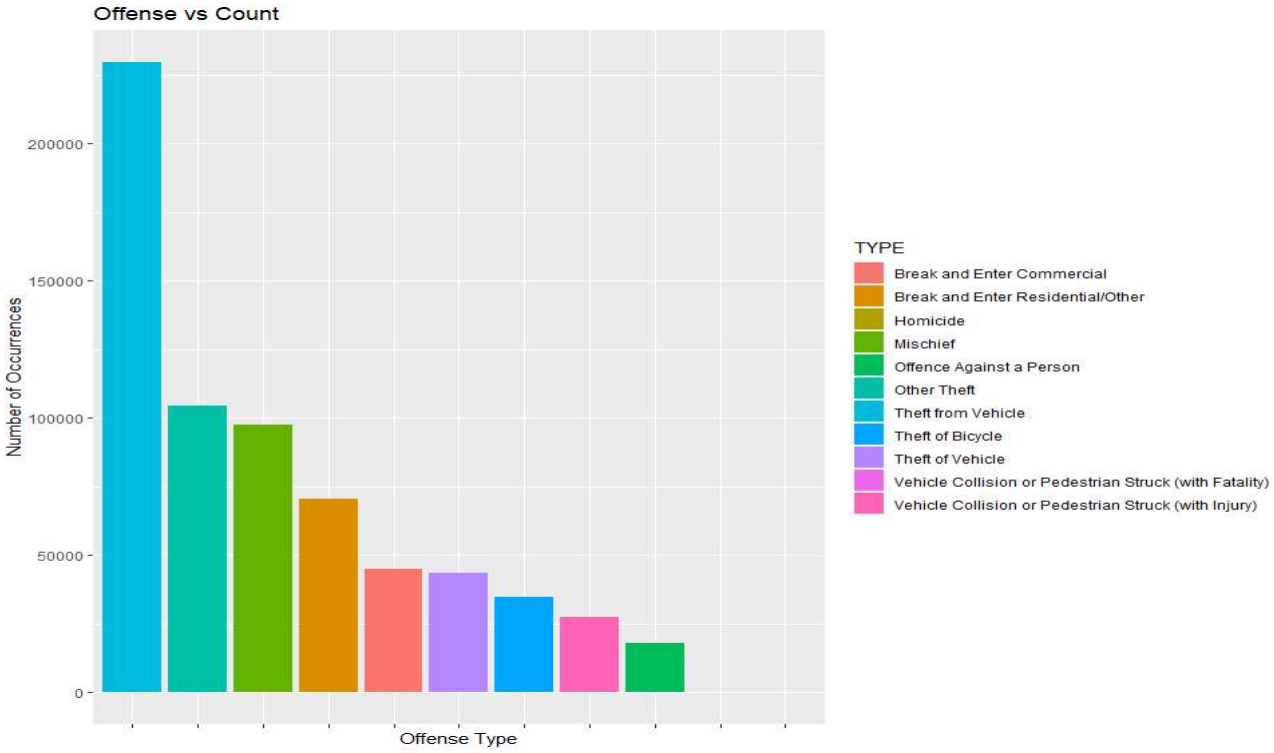
$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- g(t): piecewise linear or logistic growth curve for modeling 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

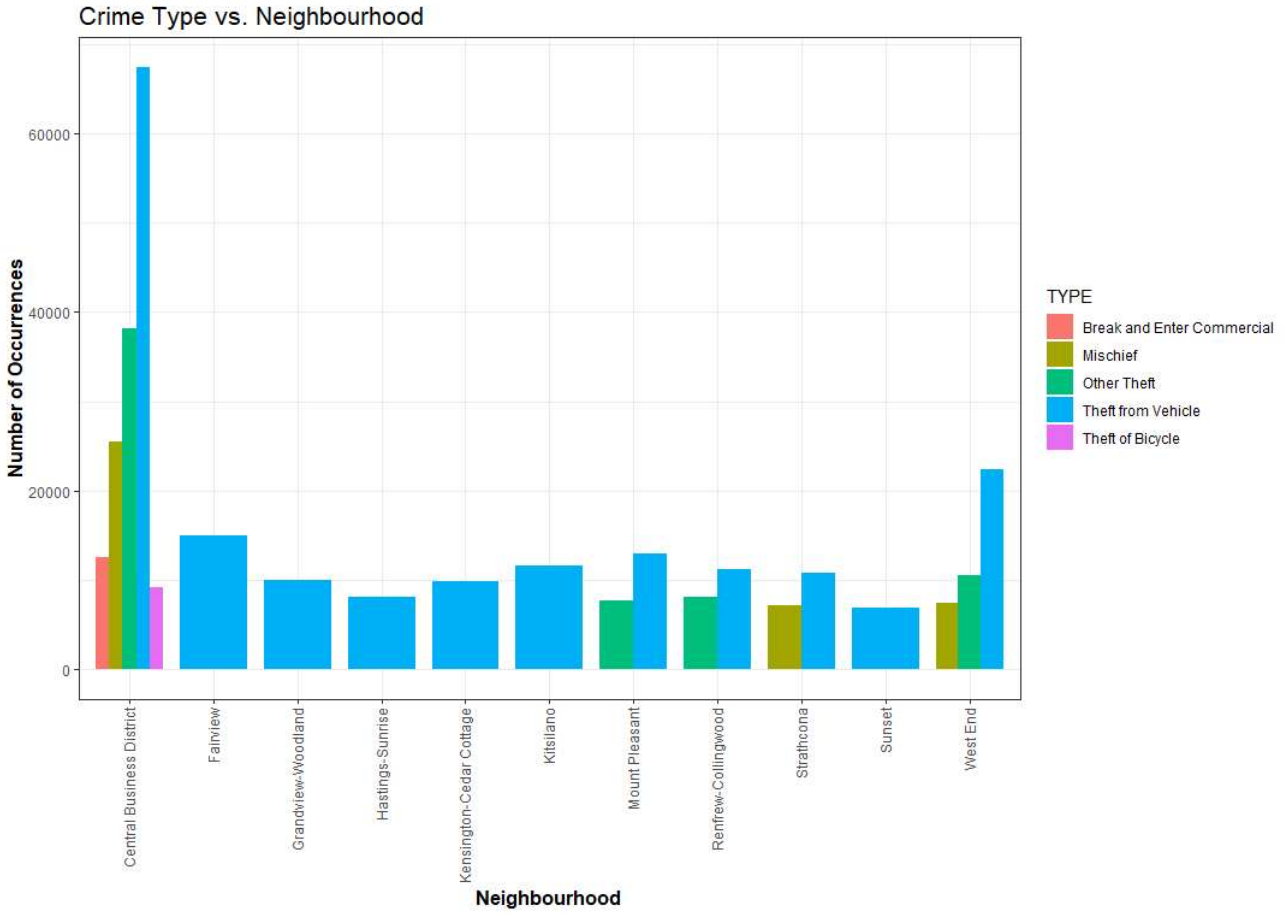
We have used the Prophet package and its Prophet() function to fit the model. We used cross-validation to test and train the model. The initial model was trained for the first 11 years and forecasted and validated for the next 365 days. In total, the model was cross-validated on 15 forecasts. The results and performance are shown in **4.2**

4. Results

4. 1 Exploratory Data Analysis

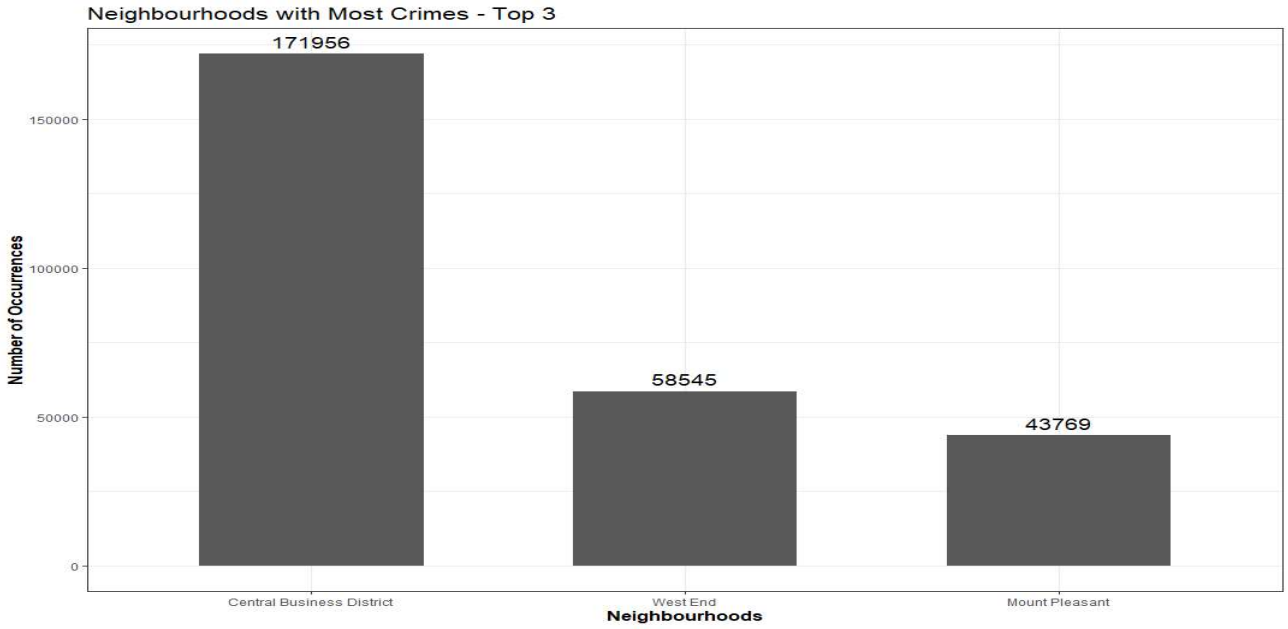


The above chart shows that *"Theft from Vehicle"* (34%), *"Other Theft"* (15%), and *"Mischief"* (14%) are the most common crimes in the city.

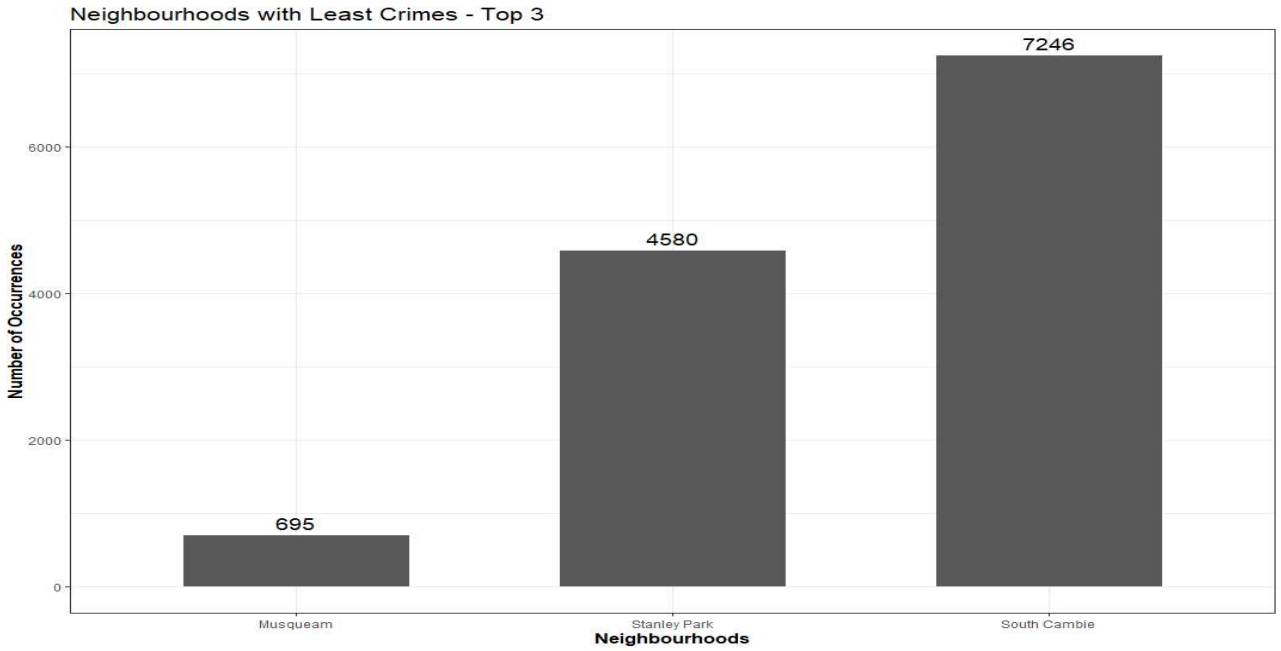


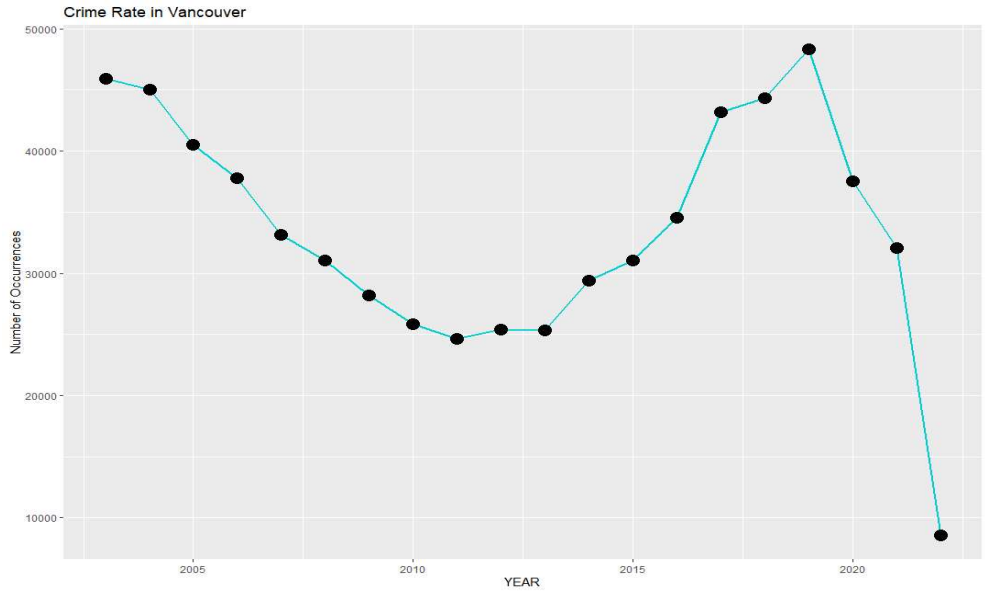
The above chart might not look very intuitive at first glimpse. However, it tells us the top 20 crime offenses across neighborhoods. Clearly, *Theft from Vehicle* makes up the majority of the list. Besides *Theft from vehicles*, *Mischief* is very common in Strathcona and West End. Also, Central Business District is a hub of top offenses having 5 of the top 20 crimes across neighborhoods.

This chart below goes along with the previous chart. The neighborhoods with the most crime happenings are *Central Business District, West End, and Mount Pleasant*. **Does this mean they are the most unsafe neighborhoods in the city?** Crime Density (per 1000 residents) could be a more clear indicator.

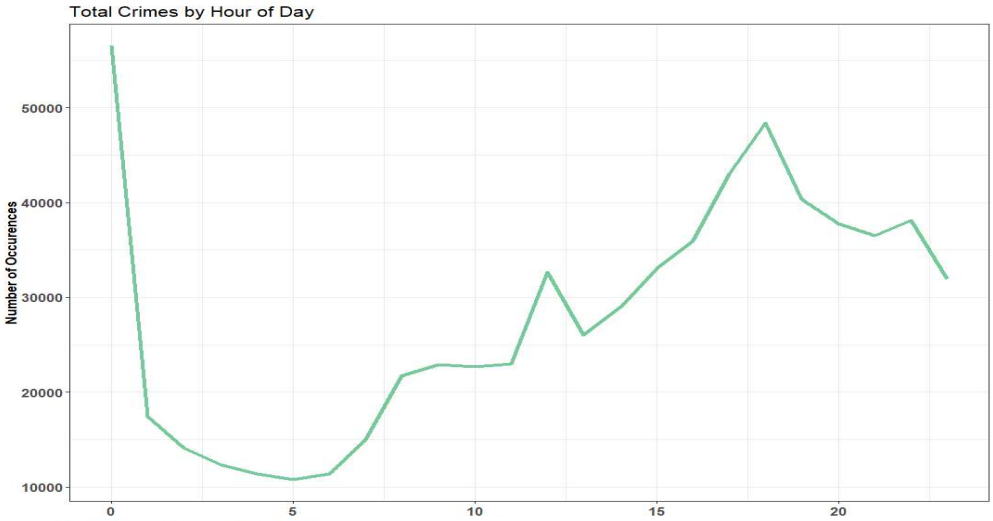


The chart below shows the top-3 Neighborhoods in the city with the least total crimes since 2003. Again, Does this mean they are the safest too? Perhaps not

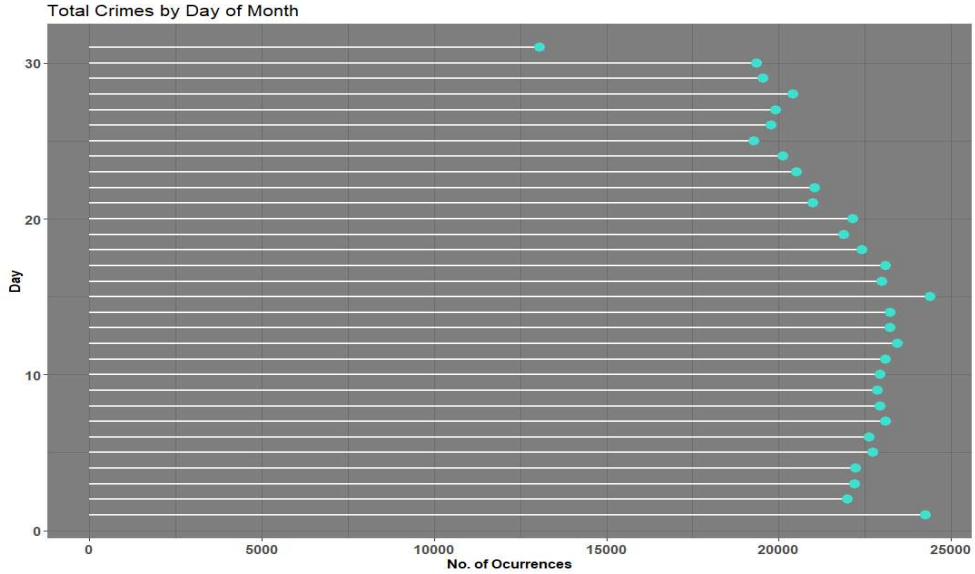




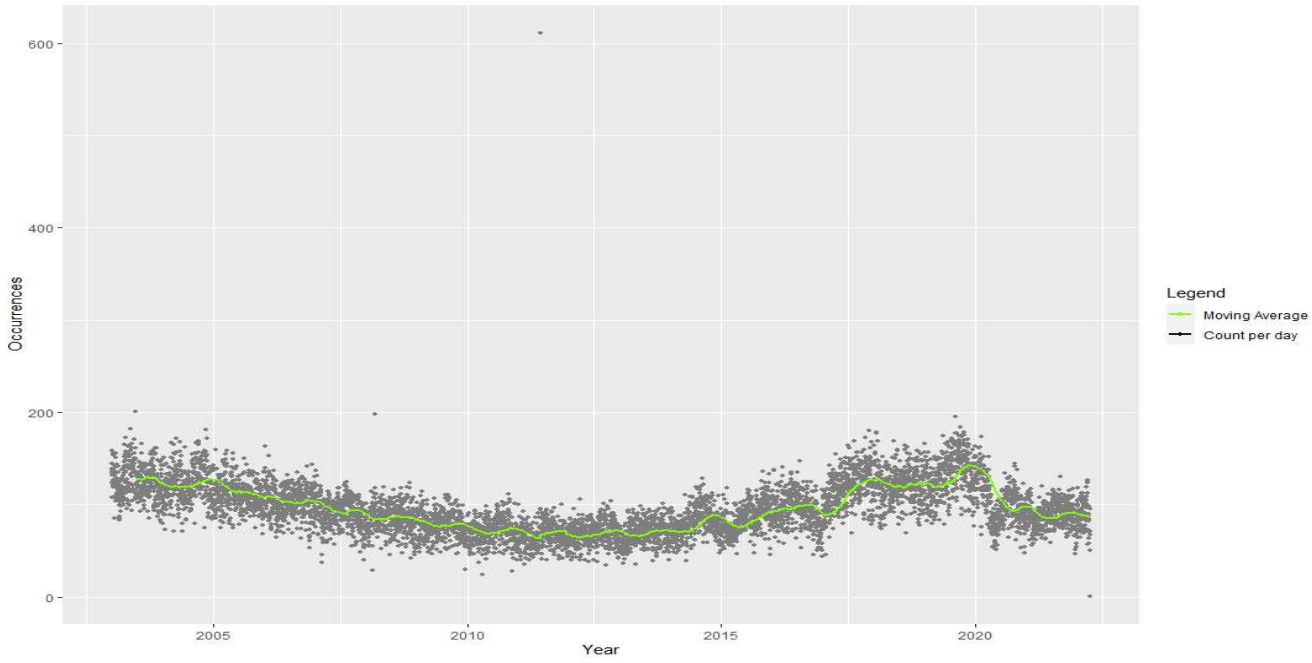
From this chart, we can see that the crime rate in Vancouver showed a *decreasing* trend from 2003 until 2013, and then it started *increasing sharply* till 2019. The year 2022 crime count is around 9000 as of April.



It is clearly visible that crime is more likely in the evening and at midnight and very less likely to occur between 12 A.M and 6 A.M



From the chart, It is unmistakably visible that daily average crime is higher at the beginning of the month, with a peak in the middle of the month and gradually decreases as it approaches the month's end.



The average number of daily crimes decreased from nearly 130 to 80 in the period 2003-2011. That's Remarkable! Vancouver hosted Winter Olympics in 2010. Are these two things correlated?

k- Means Clustering

```
K-means clustering with 2 clusters of sizes 11, 8

Cluster means:
  Break and Enter Commercial Break and Enter Residential/Other   Homicide   Mischief
1                       0.4347961                              0.6167288  0.4025508  0.4356430
2                      -0.5978446                             -0.8480021 -0.5535073 -0.5990091
  Offence Against a Person Other Theft Theft from Vehicle Theft of Bicycle Theft of Vehicle
1                0.3724804   0.3739818          0.3743268         0.4436054          0.6986316
2               -0.5121605  -0.5142250         -0.5146993        -0.6099575         -0.9606184
  Vehicle Collision or Pedestrian Struck (with Fatality)
1                                              0.4456867
2                                             -0.6128193
  Vehicle Collision or Pedestrian Struck (with Injury)
1                                           0.5782096
2                                          -0.7950382

Clustering vector:
 [1] 1 1 1 1 1 2 2 1 2 1 1 2 2 2 1 1 2 1 2

Within cluster sum of squares by cluster:
[1] 122.20992   8.96773
 (between_SS / total_SS =  33.7 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

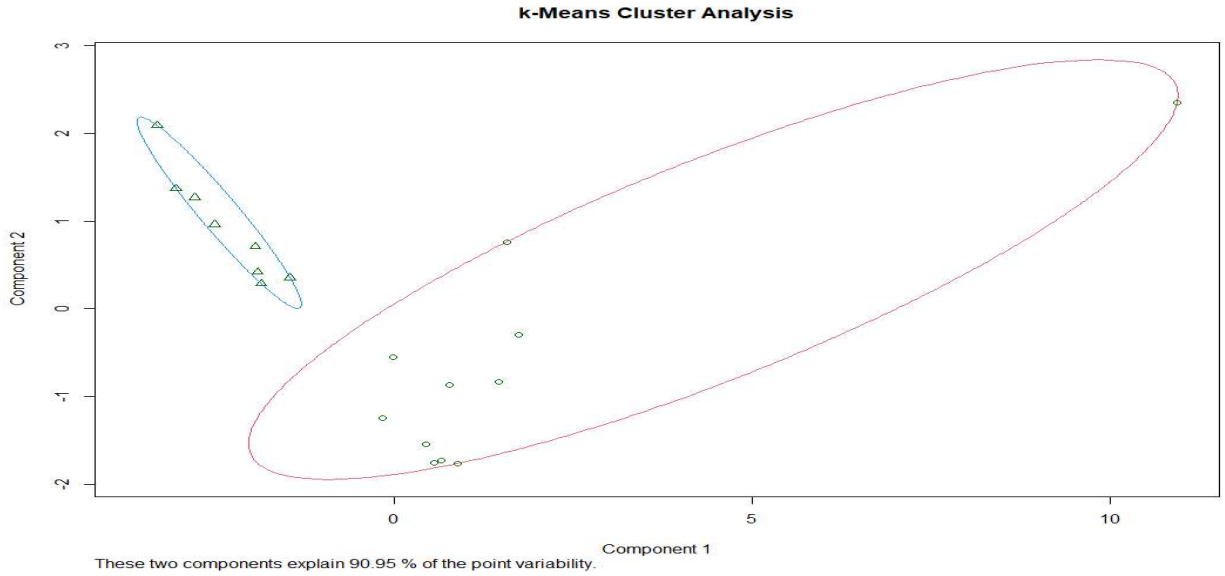
We first partition data points (neighborhoods) into *k* clusters in which each neighborhood belongs to the cluster with the nearest mean (serving as a prototype of the cluster)

We determined the number of clusters to fit the model using the *"within-group sum of squares".* We found that 2 clusters are optimal for this dataset. The above output shows that the first cluster has 11 Neighborhoods and the second cluster has 8 Neighborhoods.

In **cluster means,** attributes with negative values signify "lower than most" and positive values signify "higher than most". Thus, cluster 1 has neighborhoods with low *"Break and Enter Commercial Break", "Homicide", "Mischief"* and so on. On the other hand, cluster 2 has neighborhoods with high *"Break and Enter Commercial Break", "Homicide", "Mischief"* and so on.

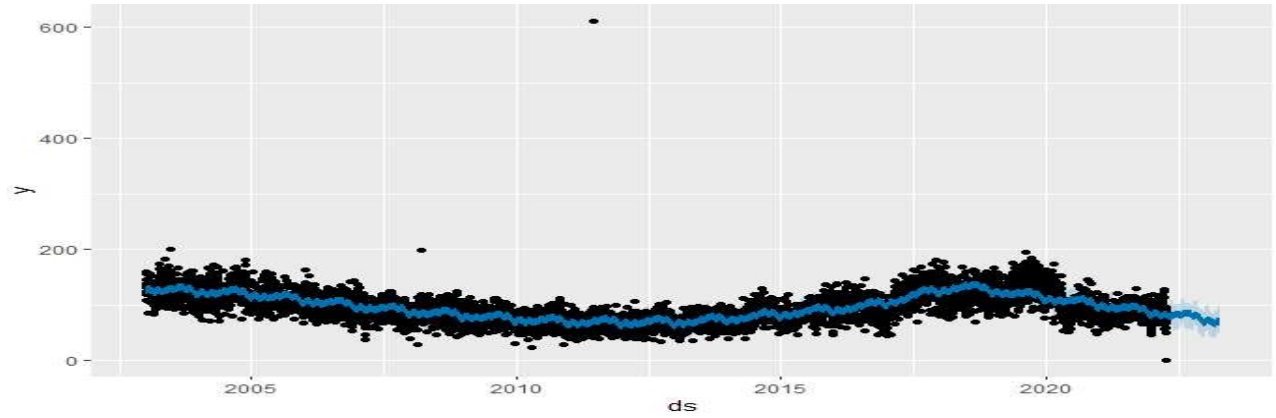
The clustering vector shows which neighborhood belongs to which cluster.

A measurement that is more relative would be the within and between. withinss tells us the sum of the square of the distance from each data point to the cluster center. Lower is better. Between tells us the sum of the squared distance between cluster centers. Ideally, we want cluster centers far apart from each other.



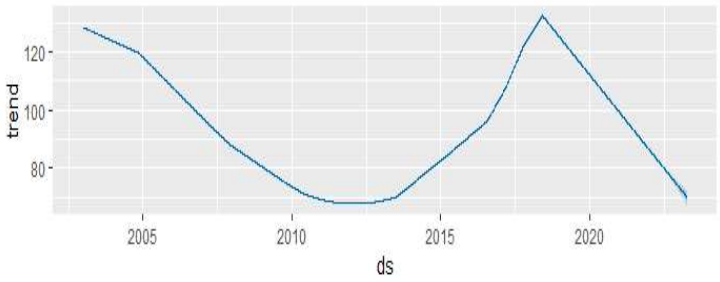
The above plot shows the neighborhoods and the clusters they belong to. We used the first two components of PCA to explain the data. The data must have high correlated variables, thus high explainability(~91%) by the two components.

4.2 Main Data Analysis

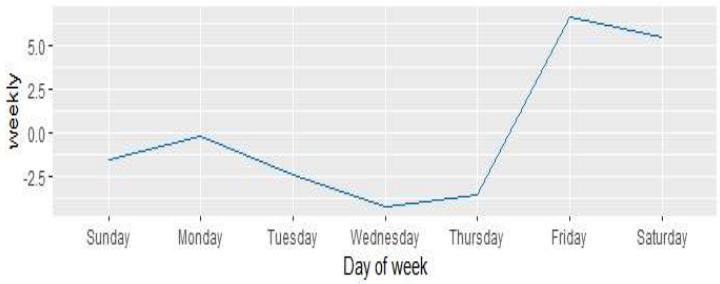


After fitting the data to the model, we forecasted daily crimes on future dates. The above plot shows the **real** data, **forecasted** data, and confidence interval.

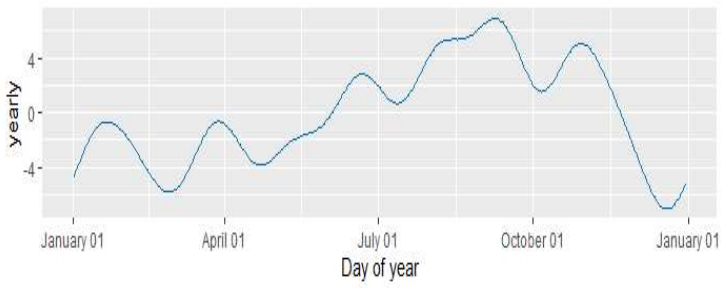
Since the Prophet model is of additive type. The above chart or time series can be decomposed into simple and more insightful components



There was a downward trend from 2003 leading to a dip in 2012. The trend again took a steep upward curve until 2019 and is dropping sharply since then.



Fridays and ***Saturdays*** are the most popular days for crimes and the opposite from Tuesday to Thursday.



This plot shows the seasonality of the data. Considering the yearly seasonality, the second half of the year shows a higher number of crimes than the first half.

Performance Evaluation

Predicted Values

y	ds	yhat	yhat_lower	yhat_upper	cutoff
79	2014-05-16	79.45079	57.94785	100.90543	2014-05-15
92	2014-05-17	79.32246	58.32440	100.10908	2014-05-15
87	2014-05-18	70.66769	49.73817	90.88581	2014-05-15
77	2014-05-19	72.21190	51.26275	93.32064	2014-05-15
99	2014-05-20	69.93735	48.87568	91.32103	2014-05-15
69	2014-05-21	68.36629	48.21619	89.48453	2014-05-15

We used time-series cross-validation to measure forecast error using historical data as mentioned previously. The above table shows the predicted(yhat) and actual values(y) of daily crime.

Performance Metrics

horizon	mse	rmse	mae	mape	mdape	smape	coverage
37 days	574.5957	23.97073	18.26791	0.1949883	0.1362580	0.1734365	0.6783668
38 days	590.3251	24.29661	18.49980	0.1988833	0.1374739	0.1759644	0.6711761
39 days	608.3564	24.66488	18.80196	0.2029625	0.1394729	0.1790884	0.6655698
40 days	617.1466	24.84244	18.94665	0.2050673	0.1421605	0.1807126	0.6630104

We then compared predicted values with actual values and computed some useful statistics of the prediction performance. The “Measuring the mean absolute percent error (**MAPE**)” is the measurement of errors in absolute terms. The above table shows the performance metrics and results. The result was 19.4% error for forecasts of 36 days and 21% for forecasts of 91 days.

5. Conclusion/Discussion

We can conclude that the forecast has a low error and that the occurrence of crimes in Vancouver can be somewhat accurately forecasted, which was a surprise given the random nature of the data. Not many more questions can be answered by looking at the data’s crimes Indicators. But that’s okay. There are certainly other interesting things to do with this data such as classifications. For example, predicting type of crime at certain hour of day in a certain location.

Predicting crime is no easy task due to its dependency on multiple factors. We do believe that having a larger dataset could lead to further improvements in our network. Furthermore, certain factors were not available and using them alongside the other variables we introduced could lead to even better results. We also believe neural network based approach suach as LSTM is the way forward and potentially give more good results.

6. References

“Diagnostics | Prophet.” *Meta Open Source*, https://facebook.github.io/prophet/docs/diagnostics.html. Accessed 21 April 2022.

“Crime in Vancouver.” *Kaggle*, https://www.kaggle.com/datasets/wosaku/crime-in-vancouver. Accessed 21 April 2022.

“Quick Start | Prophet.” *Meta Open Source*, https://facebook.github.io/prophet/docs/quick_start.html#r-api. Accessed 21 April 2022.

“VPD OPEN DATA.” *VPD GeoDash*, https://geodash.vpd.ca/opendata/. Accessed 21 April 2022.

“facebook/prophet: Tool for producing high quality forecasts for time series data that has multiple seasonality with linear or non-linear growth.” *GitHub*, https://github.com/facebook/prophet. Accessed 21 April 2022.

Li, Susan. “Exploring, Clustering and Mapping Toronto's Crimes | by Susan Li.” *Towards Data Science*, 30 October 2017, https://towardsdatascience.com/exploring-clustering-and-mapping-torontos-crimes-96336efe490f?gi=f4e1b4741582. Accessed 21 April 2022.

7. Appendix for R code

Contents

Section 1. Import Data and Libraries

```
library(dplyr)
library(readr)
library(ggplot2)
library(viridis)
library(tidyverse)
library(lubridate)
library(zoo)
library(cluster)


#deleted rows with year-2017 manually from spreadhsheet crime.csv
#merge all csv files(main.csv and 2017-2022)(Total 7 seven files combined)
df <- list.files(path = "./R project/Data/", full.names = TRUE) %>%
  lapply(read.csv) %>%
  bind_rows
```

Section 2. Data Pre-processing

```
#deleted rows with year-2017 manually from spreadhsheet crime.csv
#merge all csv files(main and from 2017-2022)
df <- list.files(path = "./R project/Data/", full.names = TRUE) %>%
  lapply(read.csv) %>%
  bind_rows

#remove latitute and longitude columns
drops <- c("Latitude","Longitude")
df <- df[, !(names(df) %in% drops)]

#count missing values in each column
colSums(is.na(df))
```

```
#remove rows with empty values in neighborhood
df <- df[!df$NEIGHBOURHOOD=="",]

#count missing values again
#rows with empty neighborhood values had all NA values in the dataset,
#so no missing values in the dataset now
colSums(is.na(df))
```

Section 3. Create Visualizations

```
##Create Visualizations

#plot crime type barplot
df2 <- df %>% group_by(TYPE) %>%   mutate(count_name_occurr = n())
g1<-ggplot(data=df2, aes(x=reorder(TYPE,-count_name_occurr), fill=TYPE)) +
  geom_bar(stat="count")
g1 + labs(title = "Offense vs Count", x="Offense Type", y="Number of Occurrences")
+ theme(axis.text.x = element_blank())


#plot crime rate from 2003 to 2022
g2 <- ggplot(df,aes(x=YEAR)) + geom_line(stat = "count", colour =
"darkturquoise",size=1) + geom_point(stat = "count",colour="black",size=5)
g2 + labs(title = "Crime Rate in Vancouver", y="Number of Occurrences")

#crime rate in vancouver by type over the years
g3 <- ggplot(data = df,aes(x=YEAR, group=NEIGHBOURHOOD, colour=NEIGHBOURHOOD)) +
geom_line(stat = "count") + facet_wrap(~NEIGHBOURHOOD, scales="free")
g3 +  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
            strip.text = element_text(size = 14),
            axis.text = element_text( size = 14 ),
            axis.title = element_text( size = 16, face = "bold"),
            legend.key.size = unit(1, 'cm')) + labs(y="Number of Occurrences")


#top 3 dangerous neighborhoods

#group neighborhoods
neighborhd <- group_by(df, NEIGHBOURHOOD)
#count rows in each group
crime_location <- summarise(neighborhd, n=n())
#order from highest to lowest
crime_location_order <- crime_location[order(crime_location$n,decreasing = TRUE),]
#get top 3
top3_location <- head(crime_location_order,3)

#plot the barplot
ggplot(aes(x=reorder(NEIGHBOURHOOD,-n),y=n), data=top3_location) +
geom_bar(stat='identity', width = 0.6) +
```

```
  geom_text(aes(label = n), stat = 'identity', data = top3_location, hjust = 0.5,
vjust = -0.5, size = 5) +
  xlab('Neighbourhoods') +
  ylab('Number of Occurrences') +
  ggtitle('Neighbourhoods with Most Crimes - Top 3') +
  theme_bw() +
  theme(plot.title = element_text(size = 16),
        axis.title = element_text(size = 12, face = "bold"))


#safest neighborhoods
#order from lowest to highest
crime_location_order_i <- crime_location[order(crime_location$n,decreasing =
FALSE),]
least3_location <- head(crime_location_order_i,3)

#plot barplot
ggplot(aes(x=reorder(NEIGHBOURHOOD,n),y=n), data=least3_location) +
geom_bar(stat='identity', width = 0.6) +
  geom_text(aes(label = n), stat = 'identity', data = least3_location, hjust =
0.5, vjust = -0.5, size = 5) +
  xlab('Neighbourhoods') +
  ylab('Number of Occurrences') +
  ggtitle('Neighbourhoods with Least Crimes - Top 3') +
  theme_bw() +
  theme(plot.title = element_text(size = 16),
        axis.title = element_text(size = 12, face = "bold"))


#top crime type across neighbourhoods
type_location_group <- group_by(df,NEIGHBOURHOOD,TYPE)
type_location_group_count <- summarise(type_location_group,n=n())
type_location_group_order <-
type_location_group_count[order(type_location_group_count$n,decreasing = TRUE),]
type_by_location_top20 <- head(type_location_group_order, 20)

ggplot(aes(x = NEIGHBOURHOOD, y=n, fill = TYPE), data=type_by_location_top20) +
  geom_bar(stat = 'identity', position = position_dodge(), width = 0.8) +
  xlab('Neighbourhood') +
  ylab('Number of Occurrences') +
  ggtitle('Crime Type vs. Neighbourhood') + theme_bw() +
  theme(plot.title = element_text(size = 16),
        axis.title = element_text(size = 12, face = "bold"),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = .4))

#monthly average crime
crime_month_group <- group_by(df,MONTH)
crime_month_count <- summarise(crime_month_group,n=as.integer(n()/20))
crime_month_count$MONTH <- ordered(crime_month_count$MONTH)
levels(crime_month_count$MONTH) <- c('Jan', 'Feb', 'March', 'Apr', 'May', 'June',
```

```
'July', 'Aug', 'Sept', 'Oct', 'Nov', 'Dec')

#barplot
ggplot(aes(x=MONTH,y=n),data=crime_month_count) +
  geom_bar(stat="identity", width=0.8, fill=viridis(12)) +
  geom_text(aes(label=n), stat="identity", data = crime_month_count, hjust = -0.1,
vjust = 0, size = 5)+
  coord_flip() + ylab("Number of Occurrences") + xlab("Month") + ggtitle("Average
Crime by Month") +
  theme_bw() + theme(plot.title = element_text(size = 16),
                     axis.title = element_text(size = 12, face = "bold"))


#heatmap month and crime types
crime_count <- df %>% group_by(MONTH,TYPE) %>% summarise(n=n())
crime_count$MONTH <- ordered(crime_count$MONTH)
levels(crime_count$MONTH) <- c('Jan', 'Feb', 'March', 'Apr', 'May', 'June',
'July', 'Aug', 'Sept', 'Oct', 'Nov', 'Dec')

#heatmap

ggplot(crime_count, aes(MONTH,TYPE,fill=n)) +
  geom_tile(size=1,color="white") +
  scale_fill_viridis() +geom_text(aes(label=n),color="white") +
  ggtitle("Crime Type by Month") +
  xlab("Month") +
  ylab("Crime Type") + labs(fill="Total")
  theme(plot.title = element_text(size = 16),
        axis.title = element_text(size = 12, face = "bold"))



#crimes by day of month
crime_count_day <- df %>% group_by(DAY) %>% summarise(n=n())
#plot
ggplot(crime_count_day,aes(x=DAY,y=n)) +
geom_segment(aes(x=DAY,xend=DAY,y=0,yend=n),color="snow1", size=1) +
  geom_point(color="turquoise",size=4) +
  theme_light() +
  coord_flip() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.border = element_blank(),
    axis.ticks.x = element_blank()
  ) +
  xlab("Day") +
  ylab("No. of Ocurrences") + ggtitle("Total Crimes by Day of Month") +
theme_dark() +
  theme(plot.title = element_text(size = 16),
        axis.title = element_text(size = 12, face = "bold"),
```

```
        axis.text = element_text(size=12, face = "bold"))



#heatmap of days and crimetype
crime_month_year <- df %>% group_by(YEAR,MONTH) %>% summarise(n=n())
crime_month_year$MONTH <- ordered(crime_month_year$MONTH)
levels(crime_month_year$MONTH) <- c('Jan', 'Feb', 'March', 'Apr', 'May', 'June',
'July', 'Aug', 'Sept', 'Oct', 'Nov', 'Dec')


#plot
ggplot(crime_month_year, aes(YEAR,MONTH,fill=n)) +
  geom_tile(size=1,color="white") +
  scale_fill_viridis() +geom_text(aes(label=n),color="white") +
  ggtitle("Total Crimes  by Month and Year") +
  ylab("Month") +
  xlab("Year") + labs(fill="Total") +
  theme(plot.title = element_text(size = 16),
      axis.title = element_text(size = 12, face = "bold"),
      axis.text = element_text(size=12, face = "bold"))




#Total Crimes by Hour of Day
ggplot(data=df,aes(x=HOUR)) + geom_line(stat="count", size=1.5, alpha=0.7, color
="mediumseagreen")+ggtitle('Total Crimes by Hour of Day') +
  ylab("Number of Occurences") + xlab("Hour(24-hour clock)") +
  theme_bw() +
  theme(plot.title = element_text(size = 16),
        axis.title = element_text(size = 12, face = "bold"),
        axis.text = element_text(size=12, face = "bold"))


#Crime type by hour of day
hour_crime_type <- df %>% group_by(HOUR,TYPE) %>% summarise(n=n())
#plot
ggplot(data = hour_crime_type,aes(x=HOUR,y=n, group=TYPE)) +
geom_line(aes(color=TYPE),size =1) +
  geom_point(aes(color=TYPE),size=2)  + ggtitle("Total Crimes by Type by Hour of
Day") +
  xlab("Hour") + ylab("No. of Occurrences") + theme_light()

#scatterplot day and crime count
#create date column from three columns
date_crime_count= df %>%
  mutate(date = make_date(YEAR, MONTH, DAY)) %>% group_by(date) %>%
summarise(n=n()) %>%
```

```
  mutate(tma = rollmean(n, k = 180, fill = NA, align = "right"))


clrs = c("Moving Average"="chartreuse1", "Count per day"="black")
#scatterplot and moving average
ggplot(date_crime_count,aes(x=date))  +
  geom_point(aes(y=n,color="Day Count"),size=1) +
  geom_line(aes(y=tma,color="Moving Average"),size=1) +
  scale_color_manual(values=clrs) +
  labs(x = "Year",
       y = "Occurrences",
       color = "Legend")
```

Section 4. Data Clustering

```
#K-MEANS clustering

#group data by type and then by neighborhood
by_groups <- group_by(df, TYPE, NEIGHBOURHOOD)
#count number of rows in each subgroup
groups <- summarise(by_groups, n=n())
#rearrange columns
groups <- groups[c("NEIGHBOURHOOD","TYPE","n")]
#split crime types into columns
type_variable <- spread(groups, key = TYPE, value = n)


#remove categorical variables
z <- type_variable[,-c(1,1)]

#remove missing(NA) values
z <- z[complete.cases(z),]


#scaling data
m <- apply(z, 2, mean)
s <- apply(z, 2, sd)
z <- scale(z, m, s)

#find ideal number of clusters using withing groups sum of squares
gss <- (nrow(z)-1) * sum(apply(z, 2, var))
for (i in 2:10) gss[i] <- sum(kmeans(z, centers=i)$withiness)
#plot
plot(1:10, gss, type='b', xlab='Number of Clusters', ylab='Within groups sum of
squares')

#fit a kmeans model
kc <- kmeans(z,2)
formattable(kc)
```

```
#plot cluster against components results
z1 <- data.frame(z, kc$cluster)
clusplot(z1, kc$cluster, color=TRUE, shade=F, labels=0, lines=0, main='k-Means
Cluster Analysis')
```

Section 5. Time-Series Forecasting with Prophet

Section 5.1 Create date and response dataframe

```
#forecasting with Prophet
library(prophet)

#create date "ds" and crime count column "y"
dataframe <- df %>%
  mutate(ds = make_date(YEAR, MONTH, DAY)) %>% group_by(ds) %>% summarise(y=n())

#tibble to dataframe
dataframe <- as.data.frame(dataframe)
```

Section 5.2 Fit Model and Predict

```
#fit prophet model on entire dataset
m <- prophet(dataframe)

#make future dates
future <- make_future_dataframe(m, periods = 365)
tail(future)

#predict on future
forecast <- predict(m, future)
tail(forecast[c('ds','yhat','yhat_lower','yhat_upper')])

#plot forecast
plot(m, forecast)

#plot forecast components
prophet_plot_components(m, forecast)
```

Section 5.3 Cross-Validation

```
#cross validation
df.cv <- cross_validation(m, initial = 4015, period = 180, horizon = 365, units =
"days")
formattable(head(df.cv))

#evaluate
df.p <- performance_metrics(df.cv)

head(df.p)


#plot mape
#plot_cross_validation_metric(df.cv, metric = 'mdape', rolling_window = 0.1)
```