FinalReport

Jay Bharadva

12/6/2021

Introduction

1. I have always wanted to work on a public safety domain from the beginning, as I am also prepared to take this to my final year project by doing the same analysis to provide suggestion based system on the entire country's crime data, So as to see at this dataset, it's the subset of the dataset for the final year project.
2. Buffalo is included in one of the top 100 most dangerous cities in the U.S.A. It is useful to analyze and predict the possibility of crime and take precautions accordingly.
3. Civilians, newcomers, and travelers to Buffalo can use this analysis to know the degree and type of threat in one particular area. In this way, users can keep in mind the safety measures.

Background Of Data

Dataset : Crime Indcident of Buffalo, NY. (2009 - 2021), Last update : October 21, 2021. This dataset is provided and maintained by Buffalo city Police Department and published at data.buffalony.gov. Proper data is available from 2009 to the current date, but its far fetched to the year 1951.

Dataset dimentions : Instances(274903), Variables(34)

Source : https://data.buffalony.gov/Public-Safety/Crime-Incidents/d6g9-xbgu (https://data.buffalony.gov/Public-Safety/Crime-Incidents/d6g9-xbgu)

About the variables

```
data <- read.csv("C:/Users/Checkout/Desktop/201/Project/Project/Crime_Incidents.csv")

#dimentions
sprintf("Dimentions of Main Dataset : %d X %d" , dim(data)[1] , dim(data)[2])
```

```
## [1] "Dimentions of Main Dataset : 274903 X 34"
```

```
#missing values
sprintf("Total %d missing values in %d(instances) X %d(variables) = %d" , sum(is.na(d
ata)), dim(data)[1], dim(data)[2], dim(data)[1]*dim(data)[2])
```

```
## [1] "Total 587219 missing values in 274903(instances) X 34(variables) = 9346702"
```

```
sprintf("Overall percentage of Missing Values in whole dataset : %f" , sum(is.na(dat
a))/prod(dim(data)) * 100 )
```

```
## [1] "Overall percentage of Missing Values in whole dataset : 6.282633"
```

As there's these many missing values in the dataset, I will have to clean it. But I am going to use only few variable(attributes) out of 34, only ones which are useful for EDA at this moment.

```
#extracting useful variables for analysis
data_exp <- subset(data ,select = c("case_number","incident_datetime","zip","hour_of_
day", "day_of_week", "incident_type_primary" , "police.district", "council.district",
"latitude", "longitude", "census.tract"))

# Dimention and Each variable's datatype (class) in above subset.
str(data_exp)
```

```
## 'data.frame':    274903 obs. of  11 variables:
##  $ case_number          : chr  "21-2590758" "21-2540744" "21-2540255" "21-2550922"
...
##  $ incident_datetime    : chr  "09/16/2021 05:20:02 PM" "09/11/2021 04:36:00 PM" "
09/11/2021 06:00:00 AM" "09/12/2021 09:20:00 PM" ...
##  $ zip                  : chr  "" "14203" "14220" "14220" ...
##  $ hour_of_day          : int  17 16 6 21 16 22 5 12 9 9 ...
##  $ day_of_week          : chr  "THURSDAY" "SATURDAY" "SATURDAY" "SUNDAY" ...
##  $ incident_type_primary: chr  "LARCENY/THEFT" "LARCENY/THEFT" "LARCENY/THEFT" "LA
RCENY/THEFT" ...
##  $ police.district      : chr  "" "District B" "District A" "District A" ...
##  $ council.district     : chr  "" "ELLICOTT" "LOVEJOY" "LOVEJOY" ...
##  $ latitude             : num  NA 42.9 42.9 42.9 42.9 ...
##  $ longitude            : num  NA -78.9 -78.8 -78.8 -78.9 ...
##  $ census.tract         : chr  "" "165" "2" "2" ...
```

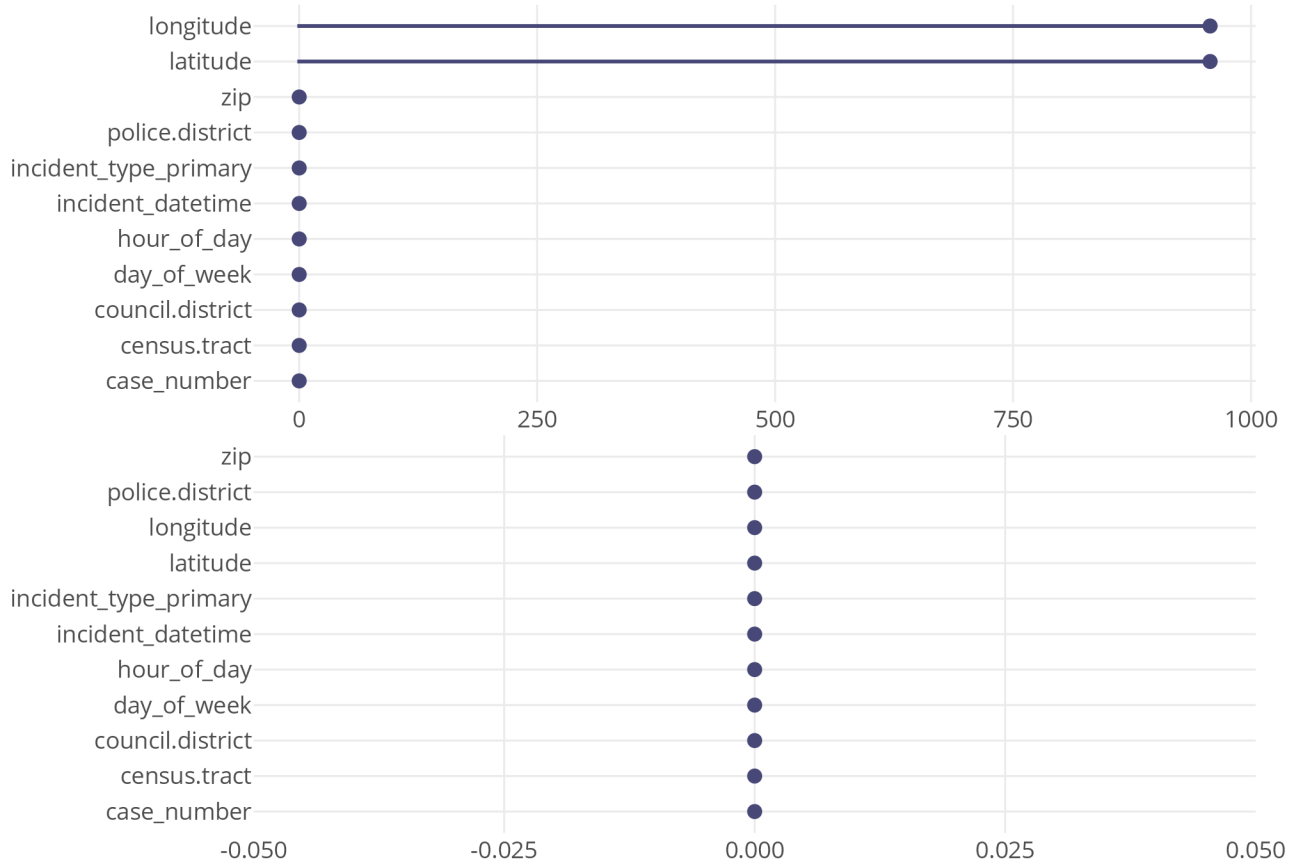
```
sprintf("Total NA values before omit : %d" ,sum(is.na(data_exp)))  #basically, there'
s 1914 "NA" values
```

```
## [1] "Total NA values before omit : 1914"
```

```
beforeOmit <- gg_miss_var(data_exp)

data_exp <- na.omit(data_exp) #removing those "NA" values
sprintf("Total missing values after omit : %d" ,sum(is.na(data_exp)))  #basically, th
ere's 1914 "NA" values
```

```
## [1] "Total missing values after omit : 0"
```

```
afterOmit <- gg_miss_var(data_exp) + labs("Missing Values after Omit")
subplot(beforeOmit , afterOmit , nrows = 2)
```




```
# But, still there might be missing/incomplete values other form
# By looking at csv file closely, there's a empty cell as well as there's few data wi
th "UNKNOWN" as an entry
```

```
#removing unknown and blank data from zip
sprintf("Incomplete values in zip variable BEFORE: Total empty cells = %d, Total UNKN
OWN cells = %d ",sum(data_exp$zip == ""), sum(data_exp$zip == "UNKNOWN"))
```

```
## [1] "Incomplete values in zip variable BEFORE: Total empty cells = 0, Total UNKNOW
N cells = 2830 "
```

```
data_exp = data_exp[!(data_exp$zip == "UNKNOWN"), ]
sprintf("Incomplete values in zip variable AFTER: Total empty cells = %d, Total UNKNO
WN cells = %d ",sum(data_exp$zip == ""), sum(data_exp$zip == "UNKNOWN"))
```

```
## [1] "Incomplete values in zip variable AFTER: Total empty cells = 0, Total UNKNOWN
cells = 0 "
```

```
#Same for the rest variables
#setting up day_of_week variable
data_exp = data_exp[!(data_exp$day_of_week == "null"), ]
data_exp$day_of_week <- toupper(data_exp$day_of_week)

#police.district
sprintf("Incomplete values in police.district variable BEFORE: Total empty cells = %
d, Total UNKNOWN cells = %d ",sum(data_exp$police.district == ""), sum(data_exp$polic
e.district == "UNKNOWN"))
```

```
## [1] "Incomplete values in police.district variable BEFORE: Total empty cells = 0,
Total UNKNOWN cells = 82 "
```

```
data_exp = data_exp[!(data_exp$police.district == "UNKNOWN"), ]
sprintf("Incomplete values in police.district variable AFTER: Total empty cells = %d,
Total UNKNOWN cells = %d ",sum(data_exp$police.district == ""), sum(data_exp$police.d
istrict == "UNKNOWN"))
```

```
## [1] "Incomplete values in police.district variable AFTER: Total empty cells = 0, T
otal UNKNOWN cells = 0 "
```

```
#counsil.district
sprintf("Incomplete values in council.district variable BEFORE: Total empty cells = %
d, Total UNKNOWN cells = %d ",sum(data_exp$council.district == ""), sum(data_exp$coun
cil.district == "UNKNOWN"))
```

```
## [1] "Incomplete values in council.district variable BEFORE: Total empty cells = 0,
Total UNKNOWN cells = 93 "
```

```
data_exp = data_exp[!(data_exp$council.district == "UNKNOWN"), ]
sprintf("Incomplete values in council.district variable AFTER: Total empty cells = %
d, Total UNKNOWN cells = %d ",sum(data_exp$council.district == ""), sum(data_exp$coun
cil.district == "UNKNOWN"))
```

```
## [1] "Incomplete values in council.district variable AFTER: Total empty cells = 0,
Total UNKNOWN cells = 0 "
```

```
#setting census.tract and census.tract.2010 variables
# data_exp = data_exp[!(data_exp$census.tract != data_exp$census.tract.2010), ]
```

```
#removing redundancy in incident_type_primary
data_exp$incident_type_primary <- toupper(data_exp$incident_type_primary)

#formating incident_datetime variable
data_exp$incident_datetime <- parse_date_time(data_exp$incident_datetime, '%m/%d/%y %
I:%M:%S %p')

#setting up hour_of_day to its right form
data_exp$hour_of_day <- format(data_exp$incident_datetime, format = "%H")

#adding new variable "year" & "Date"
data_exp$year <- format(data_exp$incident_datetime, format = "%Y")
data_exp$Date <- format(data_exp$incident_datetime, format = "%m/%d/%Y")
data_exp$month <- format(data_exp$incident_datetime, format = "%m")
```

```
dataFrom2009 <- data_exp %>%
  filter(year >= 2009)
dim(dataFrom2009)
```

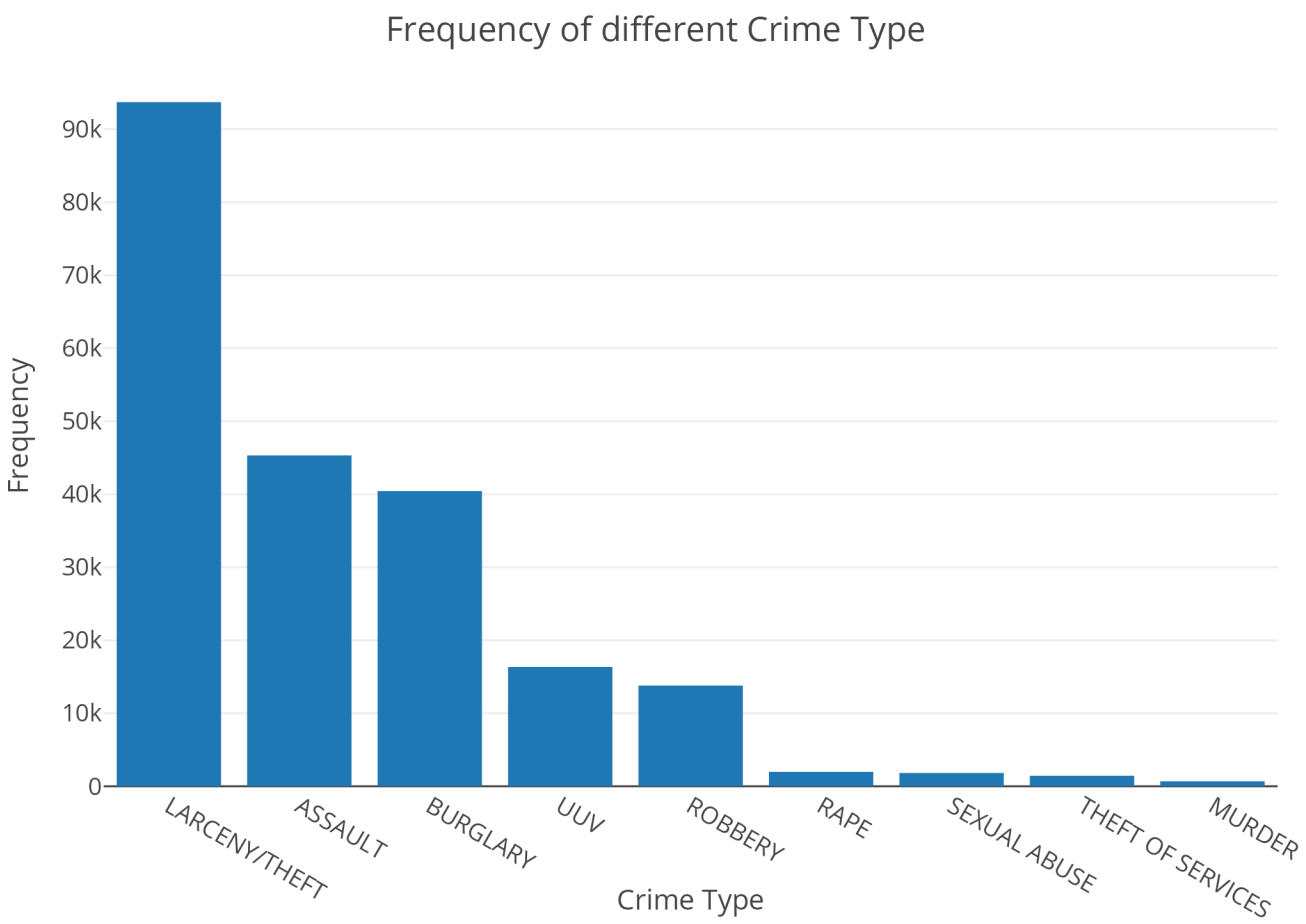
```
## [1] 215777     14
```

Potential Questions for EDA and Further analysis.

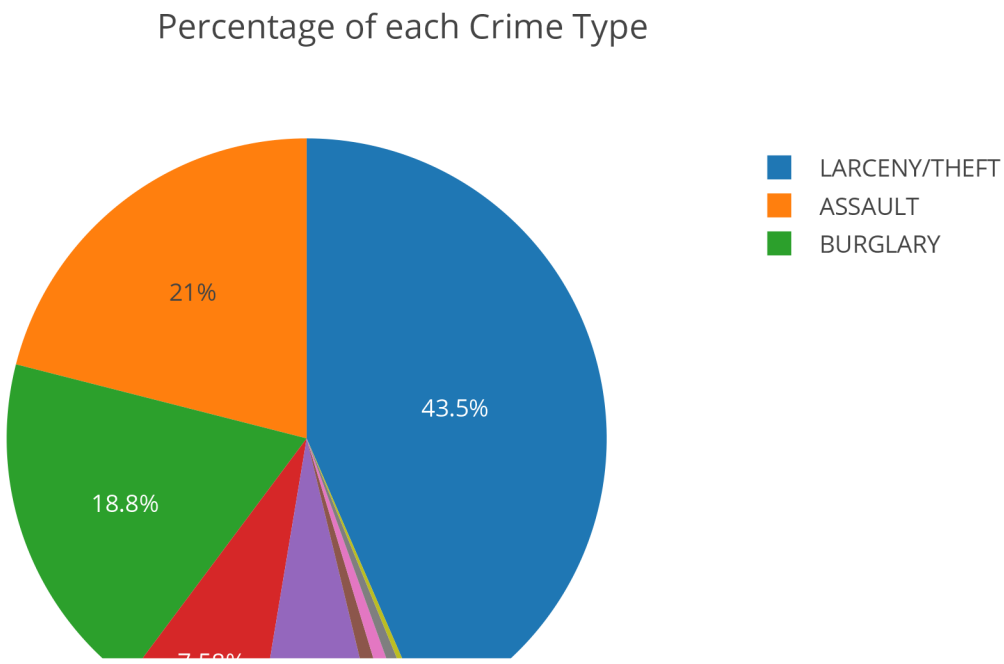
1. Which crime occurs the most frequently (Frequency > 500)?
2. What's the frequencies of top 6 crime types over the different days of week?
3. Is there any noticeable factor of crime happening rate over different time parameters (Years, Months, Days, Hours)?
4. Is there any noticeable factor of crime happening rate over different location parameters (Zip, Police District, Council District)?
5. What's the trend in daily crimes over the years (Is it upward or down draft)?
6. What are the hot-zones of a different crimes on the city map?
7. Is dimentionality reduction useful here? If so, what are the benifits and limitations of doing so?

Q.1) Which crime occurs the most frequently (Frequency > 500)?

```
df_prime_type <- sort(table(dataFrom2009$incident_type_primary),decreasing = TRUE)
df_prime_type <- data.frame(df_prime_type[df_prime_type > 500])
colnames(df_prime_type) <- c("Category", "Frequency1")
df_prime_type$Percentage <- df_prime_type$Frequency1 / sum(df_prime_type$Frequency1)

fig1 <- plot_ly(df_prime_type, x = ~Category , y = ~Frequency1) %>%
  add_bars() %>%
  layout(xaxis = list(title = "Crime Type"), yaxis = list(title = "Frequency") , titl
e = "Frequency of different Crime Type")

fig <- plot_ly(df_prime_type, labels = ~Category, values = ~Frequency1, type = 'pie')

fig <- fig %>% layout(title = 'Percentage of each Crime Type',

         xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),

         yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
         margin = 0.05
         )
fig1
```



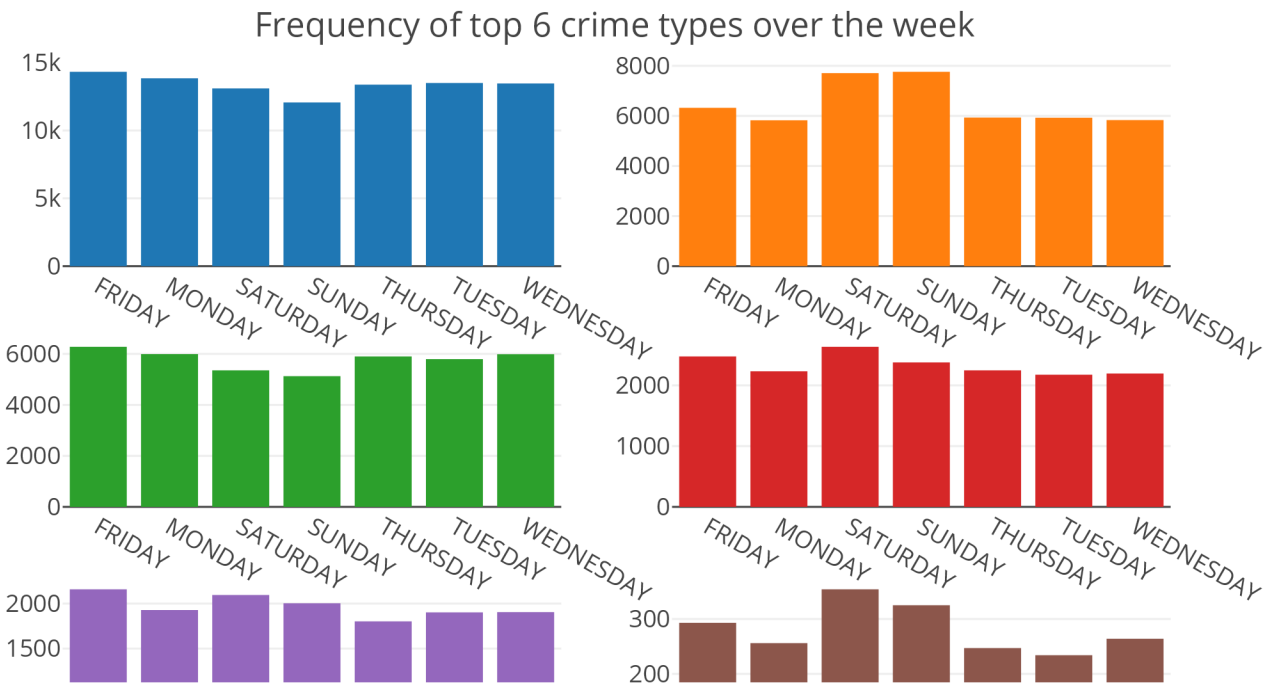

```
fig
```

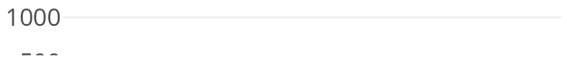


Can be seen from the bar plot that LARCENY/THEFT has the highest frequency. So this information can be used to make a suggestions to the Buffalo police department to arrange the police force accordingly.

Q.2) What's the frequencies of top 6 crime types over the different days of week?

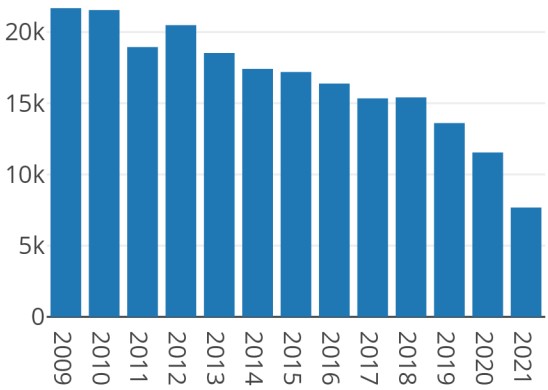
```
temp = dataFrom2009 %>%
  filter(incident_type_primary == "LARCENY/THEFT")
p1 <- plot_ly(temp, x = ~day_of_week , name = "LARCENY/THEFT") %>%
  add_histogram()

temp = dataFrom2009 %>%
  filter(incident_type_primary == "ASSAULT")
p2 <- plot_ly(temp, x = ~day_of_week , name = "ASSAULT") %>%
  add_histogram()

temp = dataFrom2009 %>%
  filter(incident_type_primary == "BURGLARY")
p3 <- plot_ly(temp, x = ~day_of_week , name = "BURGLARY") %>%
  add_histogram()

temp = dataFrom2009 %>%
  filter(incident_type_primary == "UUV")
p4 <- plot_ly(temp, x = ~day_of_week , name = "UUV") %>%
  add_histogram()

temp = dataFrom2009 %>%
  filter(incident_type_primary == "ROBBERY")
p5 <- plot_ly(temp, x = ~day_of_week , name = "ROBBERY") %>%
  add_histogram()

temp = dataFrom2009 %>%
  filter(incident_type_primary == "RAPE")
p6 <- plot_ly(temp, x = ~day_of_week , name = "RAPE") %>%
  add_histogram()

subplot(p1, p2, p3, p4, p5, p6, nrows = 3, margin = 0.05) %>% layout(legend = list(or
ientation = 'h') , title = "Frequency of top 6 crime types over the week")
```

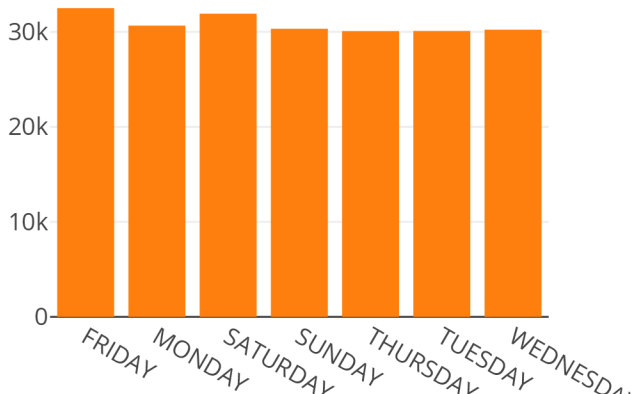


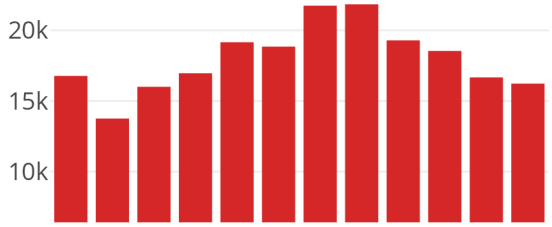


Q.3) Is there any noticeable factor of crime happening rate over different time parameters (Years, Months, Days, Hours)?

```
p1 <- plot_ly(dataFrom2009, x = ~year , name = "Number of Crime ~ 2009-2021") %>%
  add_histogram()

p2 <- plot_ly(dataFrom2009, x = ~day_of_week , name = "Number of Crime ~ Days Of Wee
k") %>%
  add_histogram()

p3 <- plot_ly(dataFrom2009, x = ~hour_of_day , name = "Number of Crime ~ Hour of Da
y") %>%
  add_histogram()

p4 <- plot_ly(dataFrom2009, x = ~month , name = "Number of Crime ~ Month Of Year") %
>%
  add_histogram()

subplot(p1, p2, p3, p4, nrows = 2, margin = 0.05) %>% layout(legend = list(orientatio
n = 'h'))
```







Let's breakdown the above plots..,

Plot 1) Numeber of Crime ~ 2009-2021 : The highest crimes happened during year 2009, and from there the frequency is getting decrease steadily throghout the sequence of year.

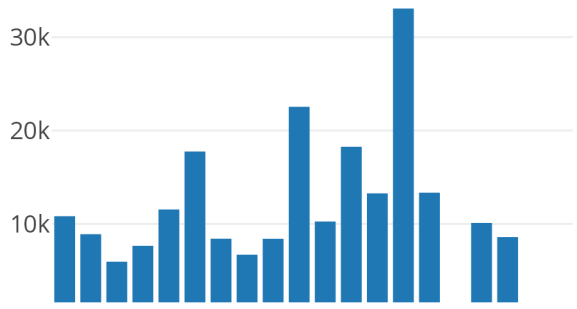
Plot 2) Numeber of Crime ~ Days Of Week : There's a minor trend during the weekend, for other days its pretty same.

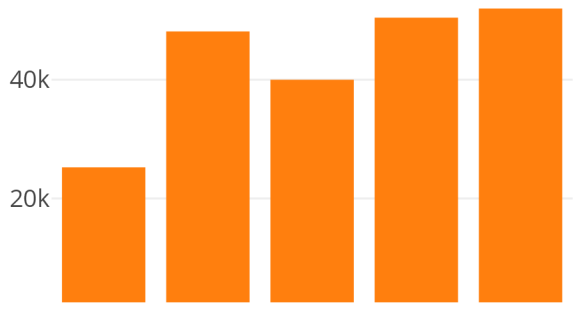
Plot 3) Numeber of Crime ~ Hour Of Day : Ahh…,During the mid night around 12AM, the crime count is almost around 20k, which is the highest. Then during the day there's a noticeable counts at 12PM.

Plot 4) Numeber of Crime ~ Month Of Year : July & August months are the ones with highest crime frequency from year 2009-2021.

Q.4) Is there any noticeable factor of crime happening rate over different location parameters (Zip, Police District, Council District)?

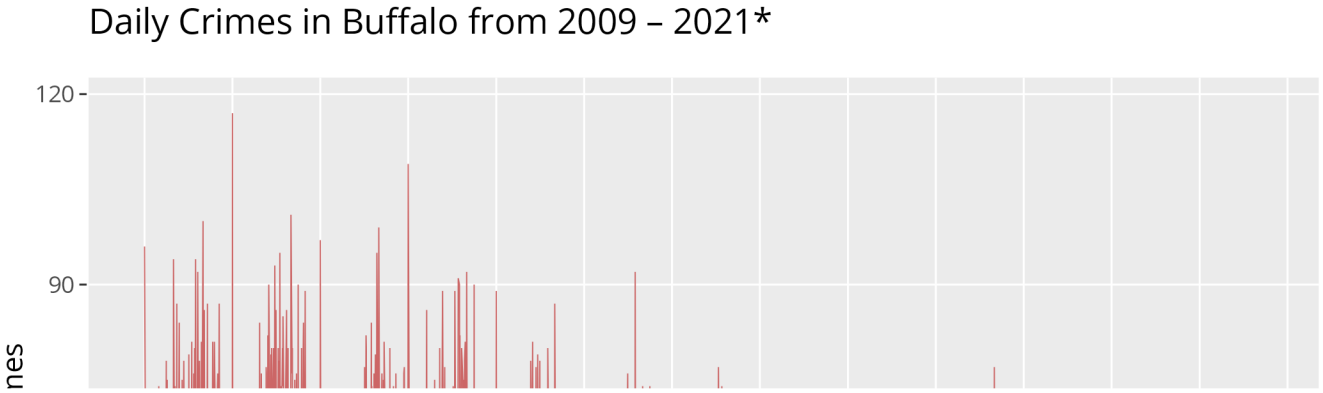
```
p1 <- plot_ly(dataFrom2009, x = ~zip , name = "Number of Crime ~ Zip") %>%
  add_histogram()

p2 <- plot_ly(dataFrom2009, x = ~police.district , name = "Number of Crime ~ Police D
istricts") %>%
  add_histogram()

p3 <- plot_ly(dataFrom2009, x = ~council.district , name = "Number of Crime ~ Council
Districts") %>%
  add_histogram()

subplot(p1, p2, p3, nrows = 2, margin = 0.05) %>% layout(legend = list(orientation =
'h'))
```

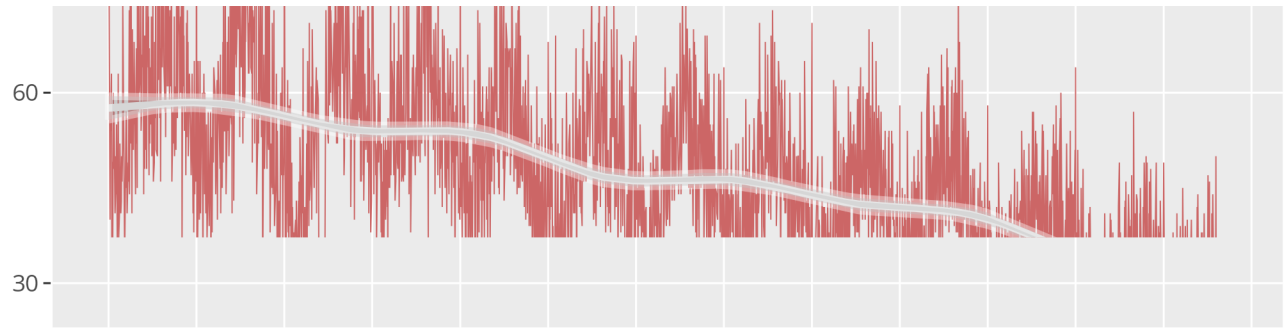




Q.5) What's the trend in daily crimes over the years (Is it upward or down draft)?

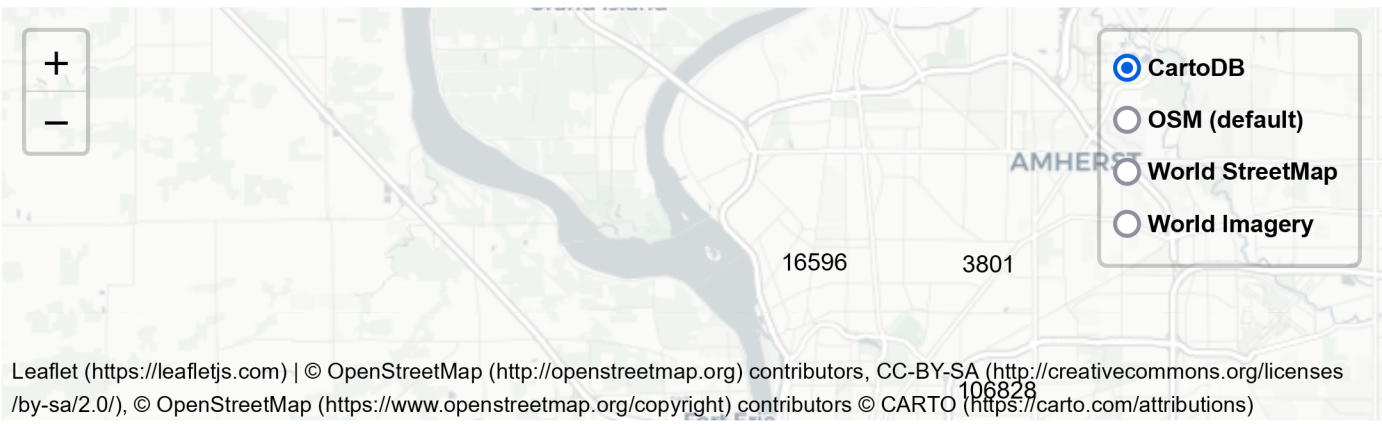
```
detach("package:ggbiplot", unload=TRUE)
detach("package:ggmap", unload=TRUE)
detach("package:plyr", unload=TRUE)

df_crime_daily <- dataFrom2009 %>%
  mutate(Date = as.Date(Date, "%m/%d/%Y")) %>%
  group_by(Date) %>%
  summarize(count = n()) %>%
  arrange(Date)

plot <- ggplot(df_crime_daily, aes(x = Date, y = count)) +
  geom_line(color = "#CC6666", size = 0.05) +
  geom_smooth(color = "#ffffff") +
  # fte_theme() +
  scale_x_date(breaks = date_breaks("1 year"), labels = date_format("%Y")) +
  labs(x = "Year of Crime", y = "Number of Crimes", title = "Daily Crimes in Buffalo
from 2009 - 2021*")
ggplotly(plot)
```

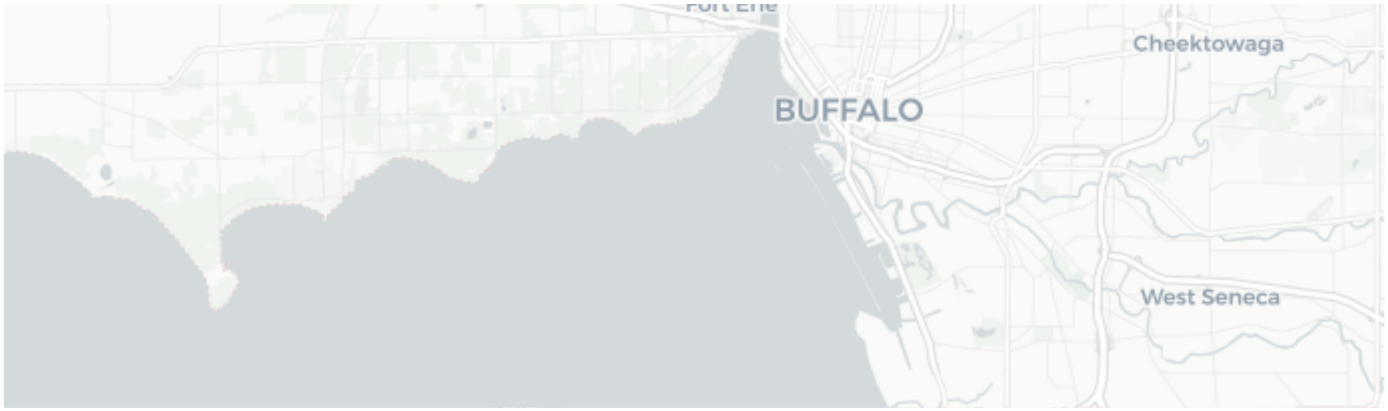




The daily crime is decreasing every year till the date. Which reflects that Buffalo Police department might be taking the precautions using this dataset. But this dataset can be used further more precisely for suggestion based application. ## Q.6) What are the hot-zones of a different crimes on the city map?

```
dataMap <- dataFrom2009
dataMap$popup <- paste(
                    "<b>Incident #: </b>", dataMap$case_number,"<br>" ,
                    "<b>Category: </b>", dataMap$incident_type_primary,
                    "<br>", "<b>Day of week: </b>", dataMap$day_of_week,
                    "<br>", "<b>Date & Time: </b>", dataMap$incident_datetime,
                    "<br>", "<b>PD district: </b>", dataMap$police.district,
                    "<br>", "<b>Council district: </b>", dataMap$council.district)

leaflet(dataMap, width = "100%") %>% addTiles() %>%
  addTiles(group = "OSM (default)") %>%
  addProviderTiles(providers$CartoDB.Positron, group = "CartoDB") %>%
  addProviderTiles(provider = "Esri.WorldStreetMap",group = "World StreetMap") %>%
  addProviderTiles(provider = "Esri.WorldImagery",group = "World Imagery") %>%
  addMarkers(lng = ~longitude, lat = ~latitude, popup = dataMap$popup, clusterOptions
= markerClusterOptions()) %>%
  addLayersControl(
    baseGroups = c("CartoDB","OSM (default)","World StreetMap", "World Imagery"),
    options = layersControlOptions(collapsed = FALSE))
```






```
# save html to png
# saveWidget(t, "temp.html", selfcontained = FALSE)
```

Zoom in for more detail & click on pin point for the crime basic information.

Preparing data for Correlation & PCA

To calculate the correlation between the variables, the first requirement is that all the variables should be of numeric type.

Variables which needs to be converted from character datatype to numeric datatype

zip, hour_of_day, census.tract, year, month

For Categorical variables :

Conversion : char -> factor -> unclass these factors -> numeric

Variables : day_of_week, incident_type_primary, police.district, council.district

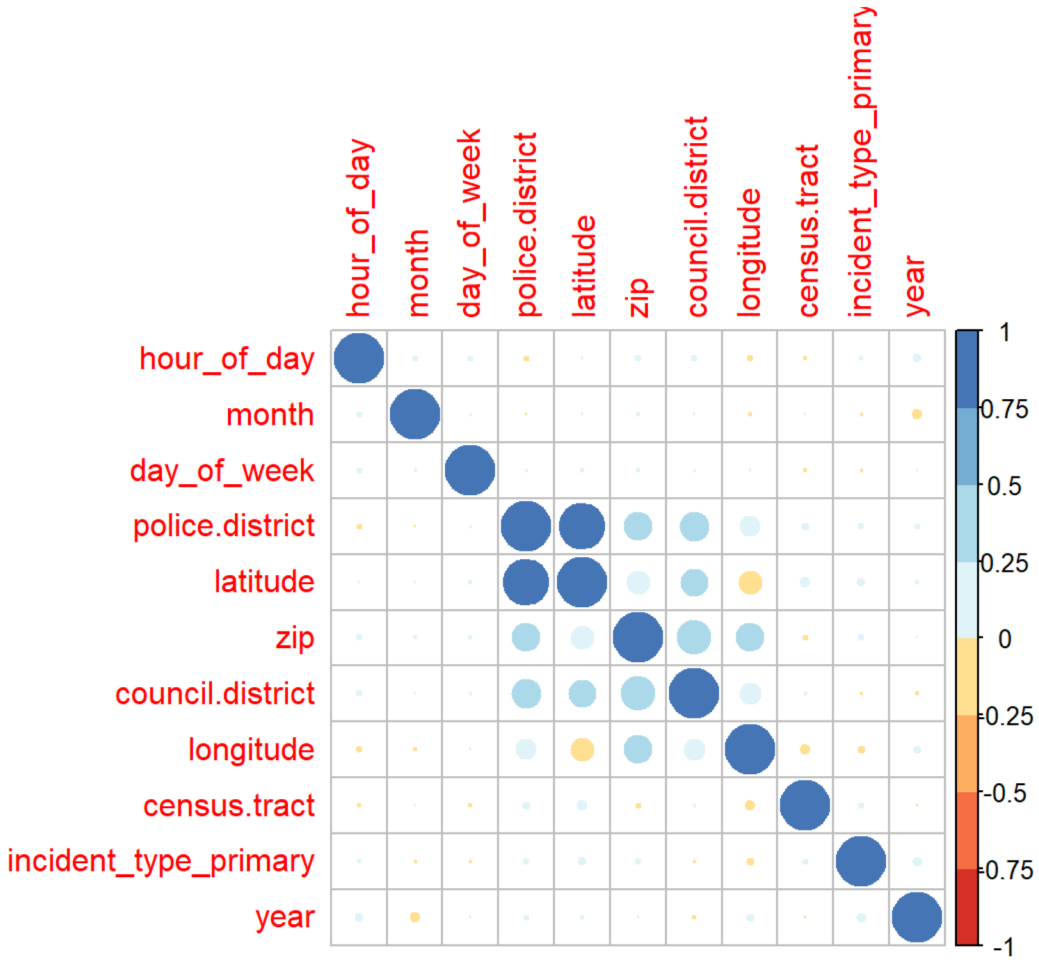
```
#To numeric
dfForCorr <- dataFrom2009
str(dfForCorr)
```

```
## 'data.frame':    215777 obs. of  14 variables:
##  $ case_number          : chr  "21-2540744" "21-2540255" "21-2550922" "21-2560651"
...
##  $ incident_datetime    : POSIXct, format: "2021-09-11 16:36:00" "2021-09-11 06:0
0:00" ...
##  $ zip                  : chr  "14203" "14220" "14220" "14204" ...
##  $ hour_of_day          : chr  "16" "06" "21" "16" ...
##  $ day_of_week          : chr  "SATURDAY" "SATURDAY" "SUNDAY" "MONDAY" ...
##  $ incident_type_primary: chr  "LARCENY/THEFT" "LARCENY/THEFT" "LARCENY/THEFT" "AS
SAULT" ...
##  $ police.district      : chr  "District B" "District A" "District A" "District C"
...
##  $ council.district     : chr  "ELLICOTT" "LOVEJOY" "LOVEJOY" "ELLICOTT" ...
##  $ latitude             : num  42.9 42.9 42.9 42.9 42.9 ...
##  $ longitude            : num  -78.9 -78.8 -78.8 -78.9 -78.8 ...
##  $ census.tract         : chr  "165" "2" "2" "15" ...
##  $ year                 : chr  "2021" "2021" "2021" "2021" ...
##  $ Date                 : chr  "09/11/2021" "09/11/2021" "09/12/2021" "09/13/2021"
...
##  $ month                : chr  "09" "09" "09" "09" ...
##  - attr(*, "na.action")= 'omit' Named int [1:957] 1 18 21 31 39 41 47 64 69 78 ...
##   ..- attr(*, "names")= chr [1:957] "1" "18" "21" "31" ...
```

```
dfForCorr <- transform(dfForCorr , year = as.numeric(gsub("," , "" , year)))
dfForCorr <- transform(dfForCorr , zip = as.numeric(gsub("," , "" , zip)))
dfForCorr <- transform(dfForCorr , hour_of_day = as.numeric(gsub("," , "" , hour_of_d
ay)))
dfForCorr <- transform(dfForCorr , police.district = as.factor(gsub("," , "" , polic
e.district)))
dfForCorr <- transform(dfForCorr , day_of_week = as.factor(gsub("," , "" , day_of_wee
k)))
dfForCorr <- transform(dfForCorr , council.district = as.factor(gsub("," , "" , counc
il.district)))
dfForCorr <- transform(dfForCorr , incident_type_primary = as.factor(gsub("," , "" ,
incident_type_primary)))
dfForCorr <- transform(dfForCorr , census.tract = as.double(gsub("," , "" , census.tr
act)))

#Cponverting categorical data to numeric using "unclass" method
dfForCorr$police.district <- unclass(dfForCorr$police.district)
dfForCorr$council.district <- unclass(dfForCorr$council.district)
dfForCorr$day_of_week <- unclass(dfForCorr$day_of_week)
dfForCorr$incident_type_primary <- unclass(dfForCorr$incident_type_primary)
# dataFrom2009$month <- format(dataFrom2009$incident_datetime, format = "%m")
dfForCorr <- transform(dfForCorr , month= as.numeric(gsub("," , "" , month)))
dfForCorr <- transform(dfForCorr , day_of_week = as.numeric(gsub("," , "" , day_of_we
ek)))
dfForCorr <- transform(dfForCorr , incident_type_primary = as.numeric(gsub("," , "" ,
incident_type_primary)))
dfForCorr <- transform(dfForCorr , police.district = as.numeric(gsub("," , "" , polic
e.district)))
dfForCorr <- transform(dfForCorr , council.district = as.numeric(gsub("," , "" , coun
cil.district)))
drops <- c("case_number","incident_datetime" ,"Date" , "census.tract.2010")
dfForCorr <- dfForCorr[ , !(names(dfForCorr) %in% drops)]
str(dfForCorr)
```

```
## 'data.frame':    215777 obs. of  11 variables:
##  $ zip                  : num  14203 14220 14220 14204 14211 ...
##  $ hour_of_day          : num  16 6 21 16 22 5 0 8 9 0 ...
##  $ day_of_week          : num  3 3 4 2 6 3 4 2 3 4 ...
##  $ incident_type_primary: num  8 8 8 3 3 3 8 5 19 5 ...
##  $ police.district      : num  2 1 1 3 3 5 5 4 2 2 ...
##  $ council.district     : num  2 4 4 2 4 5 9 7 2 6 ...
##  $ latitude             : num  42.9 42.9 42.9 42.9 42.9 ...
##  $ longitude            : num  -78.9 -78.8 -78.8 -78.9 -78.8 ...
##  $ census.tract         : num  165 2 2 15 36 ...
##  $ year                 : num  2021 2021 2021 2021 2021 ...
##  $ month                : num  9 9 9 9 9 9 9 9 9 9 ...
```

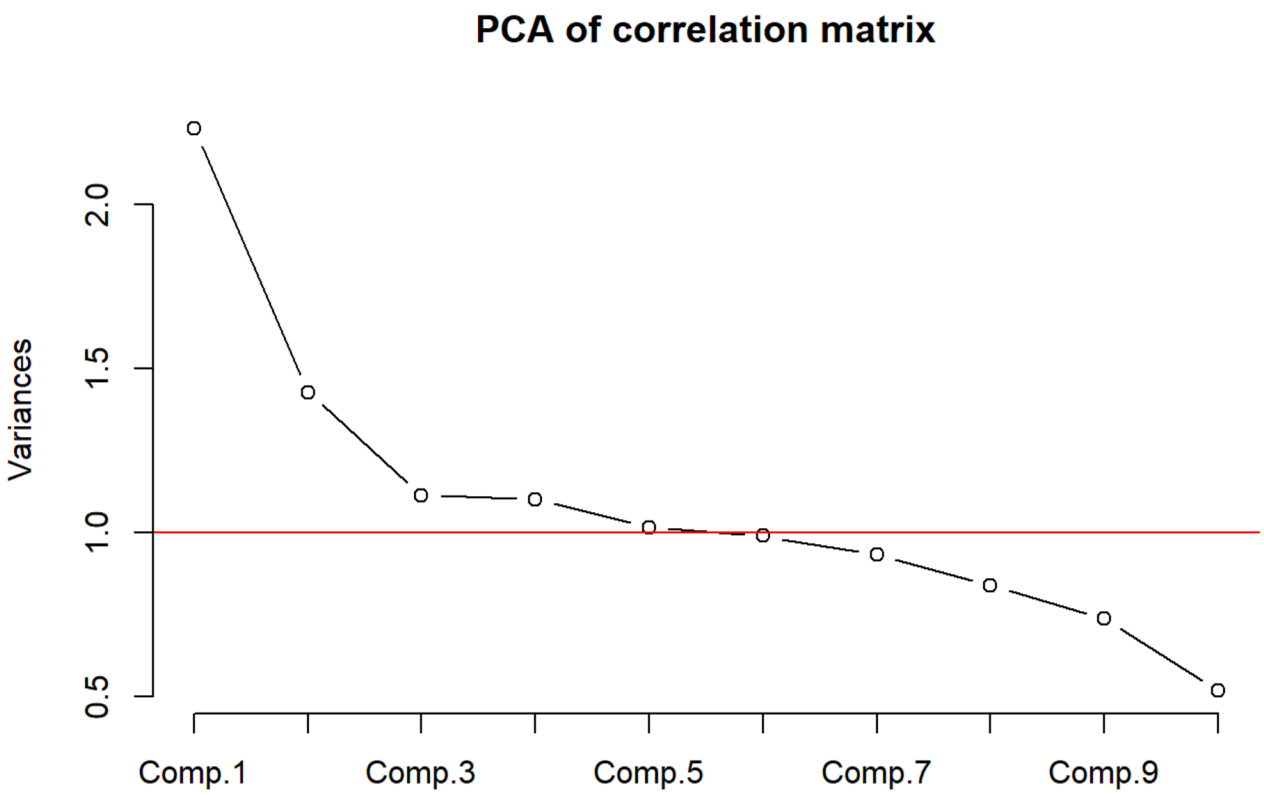
```
M <- cor(dfForCorr)
corrplot(M, order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```



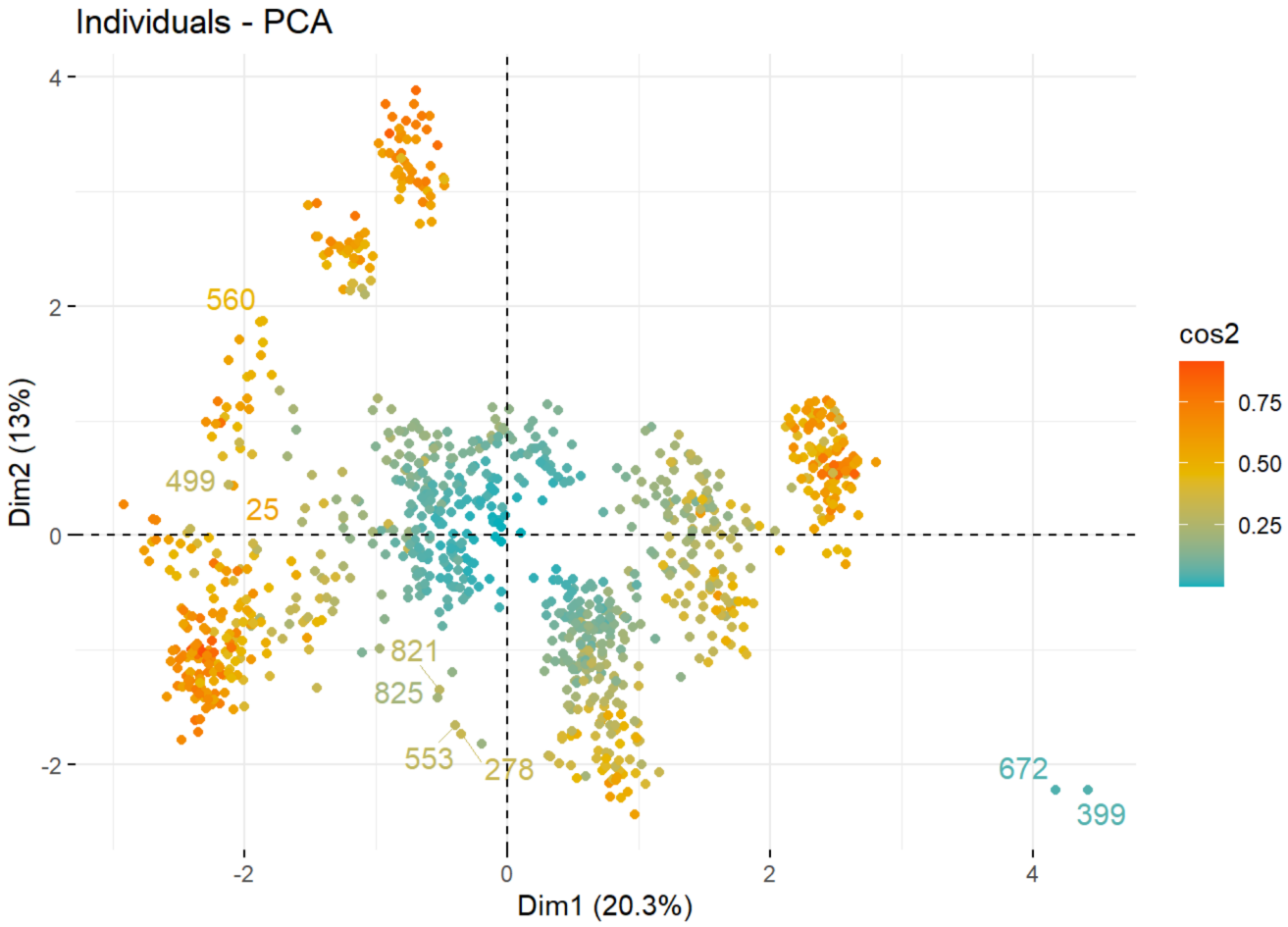

```
dfForFurtherAnalysis = sample_n(dfForCorr , 1000)
pca_cor <- princomp(dfForFurtherAnalysis, cor = TRUE, scores = TRUE)
summary(pca_cor)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
## Standard deviation     1.4934302 1.1948916 1.0552499 1.0493134 1.00722934
## Proportion of Variance 0.2027576 0.1297969 0.1012320 0.1000962 0.09222827
## Cumulative Proportion  0.2027576 0.3325545 0.4337866 0.5338828 0.62611107
##                           Comp.6     Comp.7     Comp.8    Comp.9    Comp.10
## Standard deviation     0.99448701 0.96496319 0.91602606 0.8585710 0.71893894
## Proportion of Variance 0.08990949 0.08465036 0.07628216 0.0670131 0.04698847
## Cumulative Proportion  0.71602056 0.80067092 0.87695308 0.9439662 0.99095465
##                           Comp.11
## Standard deviation     0.315434344
## Proportion of Variance 0.009045348
## Cumulative Proportion  1.000000000
```

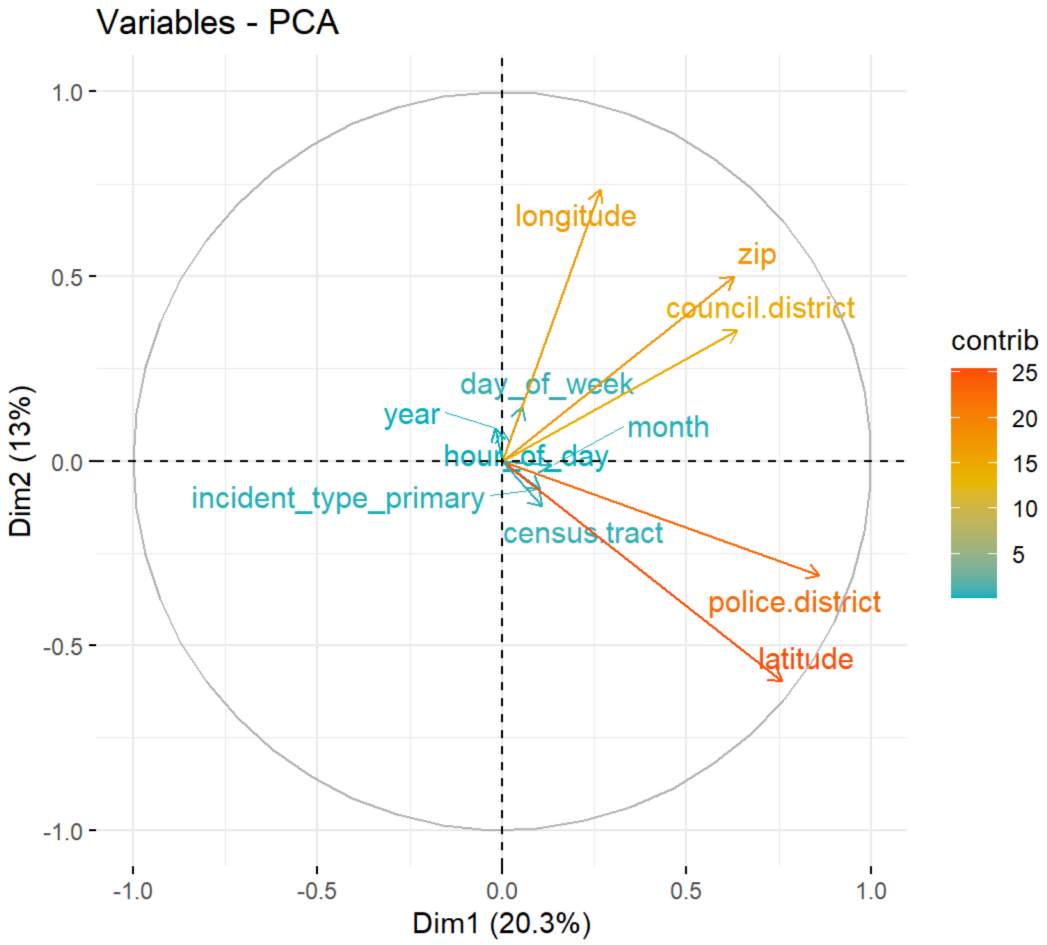
```
# Visualize eigenvalues (scree plot). Show the percentage of variances explained by e
ach principal component.
plot(pca_cor, type = "line", main = "PCA of correlation matrix")
abline(1,0, col = "red")
```



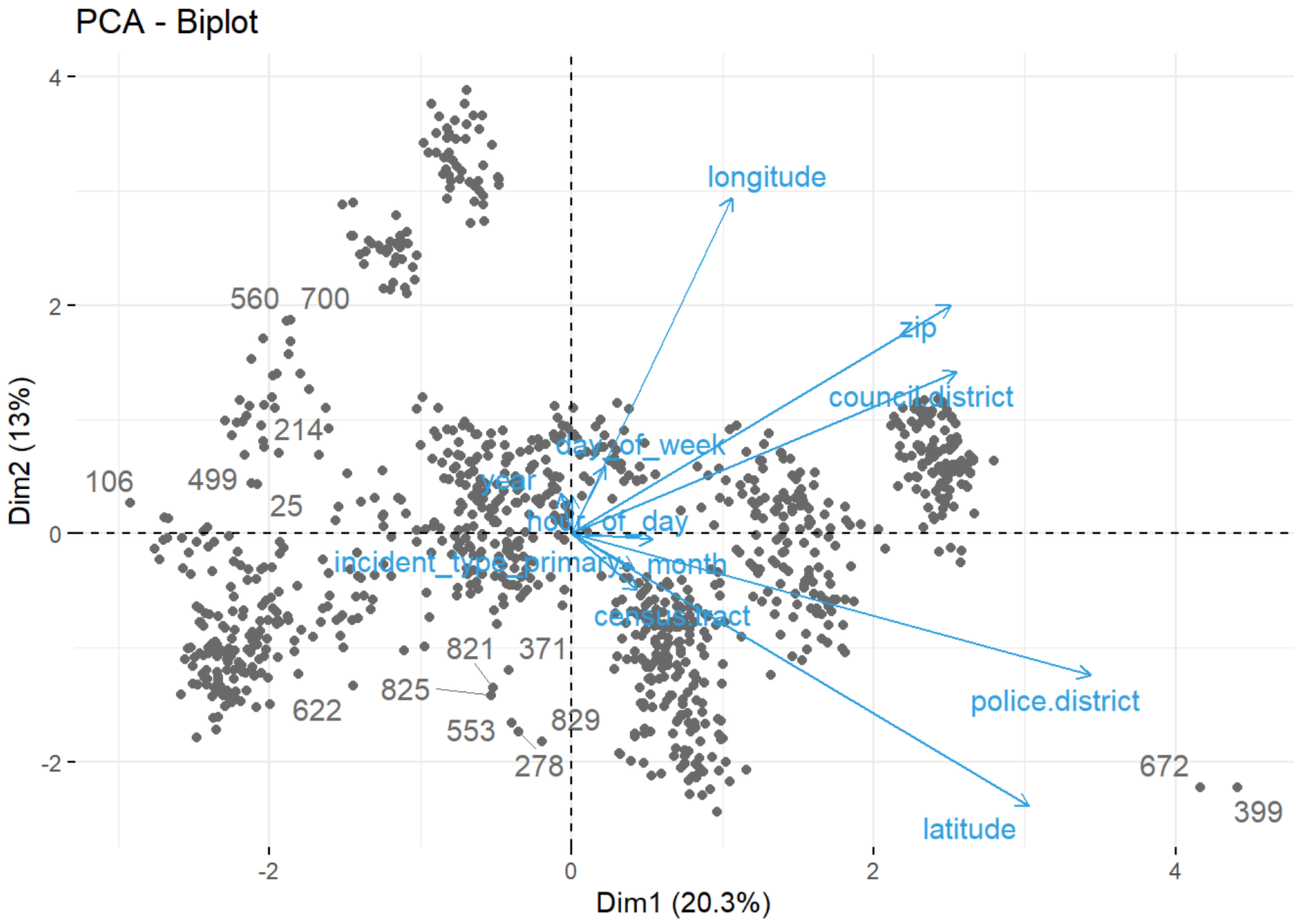

```
# Graph of individuals. Individuals with a similar profile are grouped together.
fviz_pca_ind(pca_cor,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```




```
# Graph of variables. Positive correlated variables point to the same side of the plo
t. Negative correlated variables point to opposite sides of the graph.
fviz_pca_var(pca_cor,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```




```
# Biplot of individuals and variables
fviz_pca_biplot(pca_cor, repel = TRUE, silent = TRUE,
                col.var = "#2E9FDF", # Variables color
                col.ind = "#696969"  # Individuals color
                )
```



Summary of observation of PCA analysis

dots -> samples

arrows -> original variables

1. Analysis by angle between two features:

• Acute angle : Strong Positive Correlation

• 90 degree angle : Not likely to be correlated

• 180 degree angle : Strong negative correlation

2. The length of the arrow represents it's weight on the PCs : bigger the length, weights more on PC

Here, in PCA result the "Time" variables shows the less contribution comparing to other variables. Which in the case of our dataset is not useful because "Time" parameters are as useful as "Location" parameters.

Let's first create the basic glm model using "incident_type_primary" as dependant variable and rest as the predictors.

Regression analysis using glm (Generalized Linear Model)

```
#First, let's create a basic model
model <- glm(incident_type_primary ~ . , data =dfForCorr , family = poisson())
summary(model)
```

```
##
## Call:
## glm(formula = incident_type_primary ~ ., family = poisson(),
##     data = dfForCorr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1130  -1.0836   0.0439   0.1820   3.7283
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -9.869e+01  3.699e+00 -26.677  < 2e-16 ***
## zip               2.631e-03  1.704e-04  15.435  < 2e-16 ***
## hour_of_day       3.892e-04  1.052e-04   3.701 0.000215 ***
## day_of_week      -1.377e-03  3.868e-04  -3.560 0.000371 ***
## police.district  -3.831e-03  1.254e-03  -3.054 0.002257 **
## council.district -3.850e-03  3.580e-04 -10.752  < 2e-16 ***
## latitude          6.007e-01  6.059e-02   9.914  < 2e-16 ***
## longitude        -3.456e-01  3.252e-02 -10.626  < 2e-16 ***
## census.tract      1.125e-05  2.341e-06   4.804 1.55e-06 ***
## year              5.139e-03  2.173e-04  23.646  < 2e-16 ***
## month            -3.608e-04  2.348e-04  -1.536 0.124477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 472678  on 215776  degrees of freedom
## Residual deviance: 471248  on 215766  degrees of freedom
## AIC: 1283023
##
## Number of Fisher Scoring iterations: 4
```

Understaning the P-Values:

Basically, the P-Value tells us how much the predictors are useful to us to predict any value latter on. The lower the value the more significant the independent variable is. Here, “month” predictor has the higher P value close to 1 comparing to other P-Values, it means that it is less significant predictor.On the other hand, the rest predictors has less that 5% P-value,which shows the significant impact of the coefficient.

For better fit, only including the variables with the lowest P-Values

Let’s create model with predictors which has high impact on dependent variable.

```
drop <- c("month")
testX = dfForCorr[,!(names(dfForCorr) %in% drop)]

betterModel <- glm(incident_type_primary ~ . -hour_of_day -day_of_week -police.distri
ct -census.tract, data = testX , family = poisson())
summary(betterModel)
```

```
##
## Call:
## glm(formula = incident_type_primary ~ . - hour_of_day - day_of_week -
##     police.district - census.tract, family = poisson(), data = testX)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1060  -1.0851   0.0449   0.1809   3.7221
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -9.637e+01  3.571e+00  -26.98   <2e-16 ***
## zip               2.607e-03  1.703e-04   15.31   <2e-16 ***
## council.district -3.874e-03  3.578e-04  -10.83   <2e-16 ***
## latitude          4.439e-01  3.034e-02   14.63   <2e-16 ***
## longitude        -4.048e-01  2.732e-02  -14.82   <2e-16 ***
## year              5.174e-03  2.171e-04   23.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 472678  on 215776  degrees of freedom
## Residual deviance: 471308  on 215771  degrees of freedom
## AIC: 1283073
##
## Number of Fisher Scoring iterations: 4
```

The glm model equation would be as follows.

incident_type_primary = (zip X 0.003) + (latitude X 0.44) + (year X 0.005) - (longitude X 0.4) - (council.district X 0.004) - 96

Brief of the findings

To put it all in a nutshell, the “LARCENY/THEFT” occurs most frequently in the Buffalo,NY city from year 2009-2021 with 43.5% of all types of violence. Assault being the second most frequent type of violence with 21%.Now if we talk about the different crimes on a different days of week,“LARCENY/THEFT” occurs the same almost all days with >10k frequency. For “ASSAULT” crime type, there’s a noticeable trend during the weekends. Most of the crime happened during the mid night in the months of July & August and of those crimes, majority of crimes took place at 14215 zip code in ELLICOTT council district of the buffalo city with more than 30k counts.But the good thing about this data is that we can clearly see that with each year, the

number of crimes per year is getting decrease(down draft graph).New comers can use the map (shown in question 6) to carefully choose the area being the least violent one of Buffalo city to live in.

Limitations of this study & future work

In the regression analysis, it is easily noticeable that AIC & deviance values of the models are pretty much high i.e.,1283073 & 472678 respectively. While the ideal values for these two statistics should be as lower as it could be. I did try to build the model using the transformed data to get good accuracy but it was giving the same result. The main drawback of the data is that most of the features are multimodal type. I haven't learned till now to handle such kind of distribution of the data.

For the better fir of the model, one could try to extract the detailed features of "Time" variables. For an instance, we can extract "holidays", "weekend days", "week of the year" like information from the data then each one of it as a new variable in the data. Can be added Business Quarters from the "year" variable. From the "hour_of_day" feature, we can add hour zones like "mid night", "morning" etc as well as "Business Hour" like variables. To do the analysis based on the different seasons, we can add one new variable from the "DateTime" variable of the data.