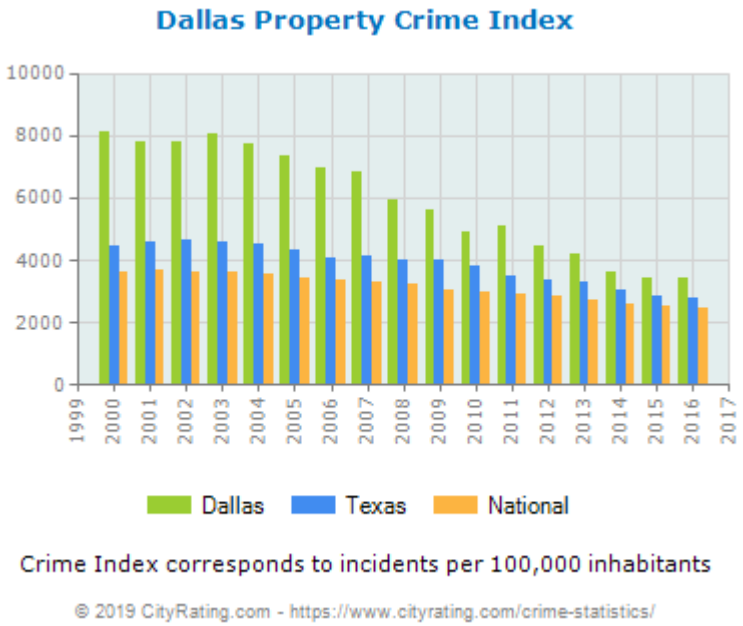
DALLAS CRIME DATA ANALYSIS

INTRODUCTION

Crime has been prevalent over the years and has become a huge concern to us. It should be taken on a serious note and analyzed properly to reduce the crime rate. Hundreds of crimes are reported daily across the country despite having strict rules and regulations. Crime analysis involves identification of patterns and trends in crime and disorder. It plays a major role in finding solutions to crime incidents and formulating crime prevention strategies. This information helps the law enforcement agencies to utitlise the resources in an effective way.

Although the crime statistics in Dallas report an overall downward trend in crime based on data from 18 years, crimes do happen everyday.



We aim at further reducing the crime rate by analyzing the crime incident reports that had happened over the years. This analysis can help alert security officials to be prepared on how to deal with them in the future. For this purpose, we have data that the Dallas Crime department - Records Management System has recorded from June 2014 till August 2019.

PROBLEM DEFINITION

The objective of this project is to analyze the trend in crimes from the year June 2014 to August 2019.

- Find the time of day and day of week when most crimes tend to occur
- Analyze the trend of different categories of crime
- Find the most used weapons in crime scenes
- Identify areas in the city where there are less crimes
- Relationship between crime types and victim’s characteristics
- Year on year increase in overall crimes and also specific types of crimes
- Predict the crime rate (for each category of crime) for the next year

EXTRACTION, EXPLORATION AND PREPROCESSING OF DATA

For this project, we use the crime data reported in the city of Dallas, beginning June 2014 to August 2019. The data is collected from www.dallasopendata.com (https://www.dallasopendata.com/Public-Safety/Police-Incidents/qv6i-rri7). This information reflects the crimes reported to the Dallas Police Department. There are about 100 columns and over 585K records.

We use the read.delim() function to read the crime data and store it in a dataframe. We can either read the data directly from the URL or download and read it from the file. Here, we are using the second approach to read the data in R.

```
library(dplyr)
library(tidyr)
library(lubridate)
```

```
crimedata <- read.delim(input_file, header = TRUE,sep = "\t", check.names = FALSE, stringsAsFact
ors = FALSE)
```

Now that we have read the data, let us understand the data, its fields and its types. str() function displays the structure of the data in a compact way.

```
str(crimedata)
```

```
## 'data.frame':    586280 obs. of  100 variables:
##  $ Incident Number w/year                  : chr  "178571-2017" "207705-2017" "206977-201
7" "197181-2017" ...
##  $ Year of Incident                        : int  2017 2017 2017 2017 2017 2017 2017 2017
2017 2017 ...
##  $ Service Number ID                       : chr  "178571-2017-01" "207705-2017-01" "20697
7-2017-02" "197181-2017-02" ...
##  $ Watch                                   : int  1 1 3 1 1 1 3 3 3 3 ...
##  $ Call (911) Problem                      : chr  "58 - ROUTINE INVESTIGATION" "6XA - MAJO
R DIST  AMBULANCE" "7X - MAJOR ACCIDENT" "11B - BURG OF BUS" ...
##  $ Type of Incident                        : chr  "TAKE WEAPON FROM AN OFFICER (ATT)" "ASS
AULT -BODILY INJURY ONLY" "INJURED PERSON- PUBLIC PROPERTY (OTHER THAN FIREARM) (NO OFFENSE)"
"BURGLARY OF BUILDING - FORCED ENTRY" ...
##  $ Type  Location                          : chr  "Medical Facility" "Other" "Highway, Str
eet, Alley ETC" "Outdoor Area Public/Private" ...
##  $ Type of Property                        : chr  "N/A" "N/A" "N/A" "Outdoor Area Public/P
rivate" ...
##  $ Incident Address                        : chr  "" "" "" "" ...
##  $ Apartment Number                        : chr  "" "" "" "" ...
##  $ Reporting Area                          : int  2043 1137 9202 4360 2037 2037 2043 9202
4315 1132 ...
##  $ Beat                                    : int  116 226 153 744 122 122 116 153 456 228
...
##  $ Division                                : chr  "CENTRAL" "NORTHEAST" "CENTRAL" "SOUTH C
ENTRAL" ...
##  $ Sector                                  : int  110 220 150 740 120 120 110 150 450 220
```

The summary() function provides the detailed summary of data.

```
summary(crimedata)
```

```
##  Incident Number w/year Year of Incident Service Number ID
##  Length:586280          Min.   :2005     Length:586280
##  Class :character       1st Qu.:2015     Class :character
##  Mode  :character       Median :2017     Mode  :character
##                         Mean   :2017
##                         3rd Qu.:2018
##                         Max.   :2109
##
##      Watch       Call (911) Problem Type of Incident   Type  Location
##  Min.   :1.000   Length:586280      Length:586280      Length:586280
##  1st Qu.:1.000   Class :character   Class :character   Class :character
##  Median :2.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1.913
##  3rd Qu.:3.000
##  Max.   :3.000
##
##  Type of Property   Incident Address   Apartment Number   Reporting Area
##  Length:586280      Length:586280      Length:586280      Min.   :1001
##  Class :character   Class :character   Class :character   1st Qu.:1248
##  Mode  :character   Mode  :character   Mode  :character   Median :3058
##                                                           Mean   :3134
##                                                           3rd Qu.:4317
##                                                           Max.   :9611
##                                                           NA's   :715
##       Beat          Division             Sector      Council District
##  Min.   :  3.0   Length:586280      Min.   :  0.0   Length:586280
##  1st Qu.:237.0   Class :character   1st Qu.:230.0   Class :character
```

Looking at the summary of data, we understand that there is one observation for each crime incident in the data frame. We have 586280 observations (rows) of 100 variables (columns) where each row is a crime incident reported to the Dallas Police Department. For the ease of data analysis, we select only the fields that are neccessary.

```
crimedata <- crimedata %>% select(`Incident Number w/year`,`Year of Incident`,`Type of Incident
`,Beat,Division, `Type of Property`,`Date1 of Occurrence`,`Month1 of Occurence`,`Time1 of Occurr
ence`,`Day1 of the Week`,`Day1 of the Year`,`Victim Age`,`Victim Gender`, `Offense Status`, `UCR
Offense Name`, `Victim Condition`, `Weapon Used`)
```

Now with the selected fields, we can start cleaning the data and make it ready for analysis.

- Incident Number should be unique

Each incident has a unique identifier associated with it which is stored in the variable `Incident number w/ year`. However we have some instances where two or more rows have the same identifier. These duplicated instances should be removed. We use the duplicated() function to remove the duplicates.

```
crimedata <- crimedata[!duplicated(crimedata$`Incident Number w/year`), ]
```

- Date of Occurrence should be between June 2014 and August 2019

Date of Occurrence is a field that stores the date when the incident was reported. Hence it should be of type *Date*. The values should be from June 2014 to August 2019.

```
crimedata$`Date1 of Occurrence` <- as.Date(crimedata$`Date1 of Occurrence`, format = "%m/%d/%Y")
crimedata <- crimedata %>% filter(between(`Date1 of Occurrence`, as.Date("2014-06-01"), as.Date(
"2019-08-31")))
```

- Year of Incident, Month of Occurrence, Day of the Week, Day of the Year should be derived from `Date of Occurrence`

It makes much sense to have the Year, Month, Day of week and Day of year values to be derived from the fields `Date of Occurrence` . This can be easily done with the *lubridate* package.

```
crimedata$`Year of Incident` <- with(crimedata, ifelse(`Year of Incident` == year(`Date1 of Occu
rrence`), `Year of Incident` , year(`Date1 of Occurrence`) ))
crimedata$`Month1 of Occurence` <- with(crimedata, ifelse(`Month1 of Occurence` == month(`Date1
of Occurrence`), `Month1 of Occurence`, month(`Date1 of Occurrence`)))
crimedata$`Day1 of the Year` <- with(crimedata, ifelse(`Day1 of the Year` == yday(`Date1 of Occu
rrence`), `Day1 of the Year`, yday(`Date1 of Occurrence`)))
crimedata$`Day1 of the Week` <- with(crimedata, ifelse(`Day1 of the Week` == wday(`Date1 of Occu
rrence`), `Day1 of the Week`, wday(`Date1 of Occurrence`)))
```

- Victim Age - Reset values greater than 125 to 125

```
crimedata$`Victim Age`[crimedata$`Victim Age` > 125 ] = 125
```

- Victim Age cannot be a negative value.

```
crimedata$`Victim Age` <- ifelse(crimedata$`Victim Age`< 0 , abs(crimedata$`Victim Age`) , crime
data$`Victim Age`)
```

The variable `Division` tells us in which part of the Dallas city the crime incident took place. It is categorized into 7 divisions - CENTRAL, NORTHEAST, SOUTH CENTRAL, SOUTHWEST, NORTH CENTRAL, NORTHWEST, SOUTHEAST. Therefore, it has to be of type *Factor*

```
unique(crimedata$Division)
```

```
##  [1] "CENTRAL"       "NORTHEAST"     "SOUTH CENTRAL" "SOUTHWEST"
##  [5] "NORTH CENTRAL" "NORTHWEST"     "SOUTHEAST"     "SouthEast"
##  [9] "SouthWest"     ""              "Central"       "NorthWest"
## [13] "NorthEast"     "South Central" "North Central"
```

It's clear that the values here are not unique, hence converting them all to Upper case help make the analysis easy.

- Division - Change all the values to Upper case

```
tmp <- crimedata %>% mutate(new_division = toupper(Division))
crimedata$Division <- tmp$new_division
crimedata$Division <- as.factor(crimedata$Division)
```

In order to find out how the crime rate trends throughout the year, we create a new varaiable named `Season` based on the `Month of Occurence`. It takes the values - *Spring*, *Summer*, *Fall*, *Winter* and is of type *Factor*.

- Find the Season of the Year i.e., Spring/Summer/Fall/Winter

```
crimedata <- crimedata %>% mutate(Season = ifelse(`Month1 of Occurence` %in% c(3,4,5), "Spring",
                                    ifelse(`Month1 of Occurence`%in% c(6,7,8), "Summer",
                                           ifelse(`Month1 of Occurence` %in% c(9,10,11), "Fall",
                                                  ifelse(`Month1 of Occurence` %in% c(12,1,2),
"Winter", NA)))))
crimedata$Season <- as.factor(crimedata$Season)
```

- Columns - Victim Gender, Victim Condition, Offense Status should of type *Factor*

```
crimedata$`Victim Gender` <- as.factor(crimedata$`Victim Gender`)
crimedata$`Victim Condition` <- as.factor(crimedata$`Victim Condition`)
crimedata$`Offense Status` <- as.factor(crimedata$`Offense Status`)
```

We can also try to find out the time of day most crimes tend to happen. For this, we extract the hour of the day from the `Time of Occurrence` variable.

- Extract the hour of the day when the crime took place.

```
crimedata <- crimedata %>% mutate(`Hour of the Day` = sub(":.*", "",`Time1 of Occurrence`))
```

Variable `Type of Property` stores the target item of the incident. Example : Motor vehicle, Apartment. It cannot take numeric values.

- `Type of Property` cannot have numeric values

```
crimedata$`Type of Property` <- as.factor(crimedata$`Type of Property`)
crimedata$`Type of Property` <- droplevels(crimedata$`Type of Property`, exclude = c(910,920,932
,510))
```

`UCR Offense Name` stores the type of crime incident that took place. For all the crimes that happened in the year 2019, there is no value for `UCR Offense Name`. This can be extracted from the column `Type of Incident`.

To do this, we create a dataframe which maps the unique `Type of Incident` to its corresponding `UCR Offense Name` called *offenseNames*. Based on this mapping, we find all the missing values for the column `UCR Offense Names`.

```
temp <- crimedata[!duplicated(crimedata$`Type of Incident`), ]
offenseNames <- temp %>% dplyr::select(`Type of Incident`, `UCR Offense Name`)

offenseNames$`UCR Offense Name`[which(startsWith(offenseNames$`Type of Incident`, "ARSON"))] <
- "ARSON"
offenseNames$`UCR Offense Name`[which(grepl(paste(c("^GRAFFITI", "^CRUELTY TO", "^CRIM MISCHIE
F"), collapse="|"),offenseNames$`Type of Incident` ))] <- "VANDALISM & CRIM MISCHIEF"
offenseNames$`UCR Offense Name`[which(grepl(paste(c("^ASSAULT", "^DEADLY CONDUCT"), collapse=
"|"),offenseNames$`Type of Incident` ))] <- "ASSAULT"
offenseNames$`UCR Offense Name`[which(startsWith(offenseNames$`Type of Incident`, "BMV"))] <-
"THEFT/BMV"
offenseNames$`UCR Offense Name`[which(grepl(paste(c("^CREDIT CARD", "^COMPUTER SECURITY", "^DE
CEPTIVE", "^FRAUD", "^THEFT OF SERVICE", "^FALSE STATEMENT", "^TAMPER W" , "^SECURE EXE", "^FA
IL TO", "FALSE ALARM"), collapse="|"),offenseNames$`Type of Incident` ))] <- "FRAUD"
offenseNames$`UCR Offense Name`[which(startsWith(offenseNames$`Type of Incident`, "CRIMINAL TR
ESPASS"))] <- "CRIMINAL TRESPASS"
offenseNames$`UCR Offense Name`[which(grepl(paste(c("^DELIVERY", "^MAN DEL", "^POSS CONT", "^P
OSS MARIJUANA"), collapse="|"),offenseNames$`Type of Incident` ))] <- "NARCOTICS & DRUG"
offenseNames$`UCR Offense Name`[which(grepl(paste(c("^DISORDERLY", "^DISRUPT", "^ILLUMINA", "^
ONLINE IMPRESS", "^STALKING", "^SEX OFFENDERS", "^HARASSMENT"), collapse="|"),offenseNames$`Ty
pe of Incident` ))] <- "DISORDERLY CONDUCT"
offenseNames$`UCR Offense Name`[which(startsWith(offenseNames$`Type of Incident`, "DWI"))] <-
"DWI"
offenseNames$`UCR Offense Name`[which(startsWith(offenseNames$`Type of Incident`, "ESCAPE"))]
 <- "ESCAPE"
offenseNames$`UCR Offense Name`[which(startsWith(offenseNames$`Type of Incident`, "EVADING"))]
<- "EVADING"
```

```
length(unique(crimedata$`UCR Offense Name`))
```

```
## [1] 49
```

There are 50 different types of crime. We can group similar categories of crime into one and make this number smaller.

- Group similar offense types

```
crimedata$`Crime Type` <- crimedata$`UCR Offense Name`

crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("ASSAULT", "AGG ASSAULT - NFV"))] =
"ASSAULT"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("BURGLARY-BUSINESS", "BURGLARY-RESIDE
NCE"))] = "BURGLARY"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("THEFT/BMV", "THEFT/SHOPLIFT", "OTHER
THEFTS", "THEFT ORG RETAIL", "EMBEZZLEMENT"))] = "THEFT"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("ROBBERY-BUSINESS", "ROBBERY-INDIVIDU
AL"))] = "ROBBERY"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("ACCIDENT MV", "MOTOR VEHICLE ACCIDEN
T"))] = "ACCIDENT"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("NARCOTICS & DRUGS", "NARCOTICS & DRU
G" ,"DRUNK & DISORDERLY", "DWI", "LIQUOR OFFENSE", "INTOXICATION MANSLAUGHTER"))] = "DRUGS"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("TRAFFIC VIOLATION", "TRAFFIC FATALIT
Y"))] = "TRAFFIC"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("MURDER", "SUDDEN DEATH&FOUND BODIES"
, "VANDALISM & CRIM MISCHIEF", "WEAPONS", "ARSON", "TERRORISTIC THREAT", "KIDNAPPING", "HUMAN TR
AFFICKING", "OFFENSE AGAINST CHILD", "ORANIZED CRIME"))] = "VIOLENCE"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("DISORDERLY CONDUCT" ,"CRIMINAL TRESP
ASS", "EVADING", "RESIST ARREST", "FAIL TO ID", "GAMBLING", "ESCAPE", "FRAUD", "UUMV", "FORGE &
 COUNTERFEIT"))] = "NONVIOLENCE"
crimedata$`Crime Type`[which(crimedata$`Crime Type` %in% c("NOT CODED", "LOST", "ANIMAL BITE",
"OTHERS", "FOUND", "INJURED FIREARM", "INJURED HOME", "INJURED OCCUPA", "INJURED PUBLIC"))] = "O
THERS"
crimedata$`Crime Type` <- as.factor(crimedata$`Crime Type`)
```

Similarly, let us group all the similar weapon categories and store it as *Factor*

```
crimedata$`Weapon Type` <- ""
crimedata$`Weapon Type`[which(grepl("gun",crimedata$`Weapon Used`, ignore.case = TRUE))] <- "Gu
n"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("Rifle", "Missile/Rock"))] = "Gun"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("Hands-Feet"))] = "Hands/Feet"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("Vehicle", "MOTOR VEHICLE"))] = "Ve
hicle"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("None"))] = "No Weapons"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("Threats"))] = "Threat"
crimedata$`Weapon Type`[which(grepl("knife",crimedata$`Weapon Used`, ignore.case = TRUE))] = "Kn
ife"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("Other Cutting Stabbing Inst.", "SW
ITCHBLADE", "AXE", "ICE PICK"))] = "Knife"
crimedata$`Weapon Type`[which(grepl("fire",crimedata$`Weapon Used`, ignore.case = TRUE))] = "Fir
e"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("Explosives", "Gas/Carbon Monoxide"
, "Burn/Scald"))] = "Knife"
crimedata$`Weapon Type`[which(grepl("drugs",crimedata$`Weapon Used`, ignore.case = TRUE))] = "Dr
ugs"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("ANY WEAPON OF FORCE DEADLY DISEAS
E, ETC", "Omission/Neglect"))] = "Drugs"
crimedata$`Weapon Type`[which(crimedata$`Weapon Used` %in% c("Other", "Blunt", "Stangulation",
"Assault", "Crowbar", "Asphixation", "BlackJack/Club", "Omission"))] <- "Others"
crimedata$`Weapon Type`[which(crimedata$`Weapon Type` == "")] <- "Others"
crimedata$`Weapon Type` <- as.factor(crimedata$`Weapon Type`)
```

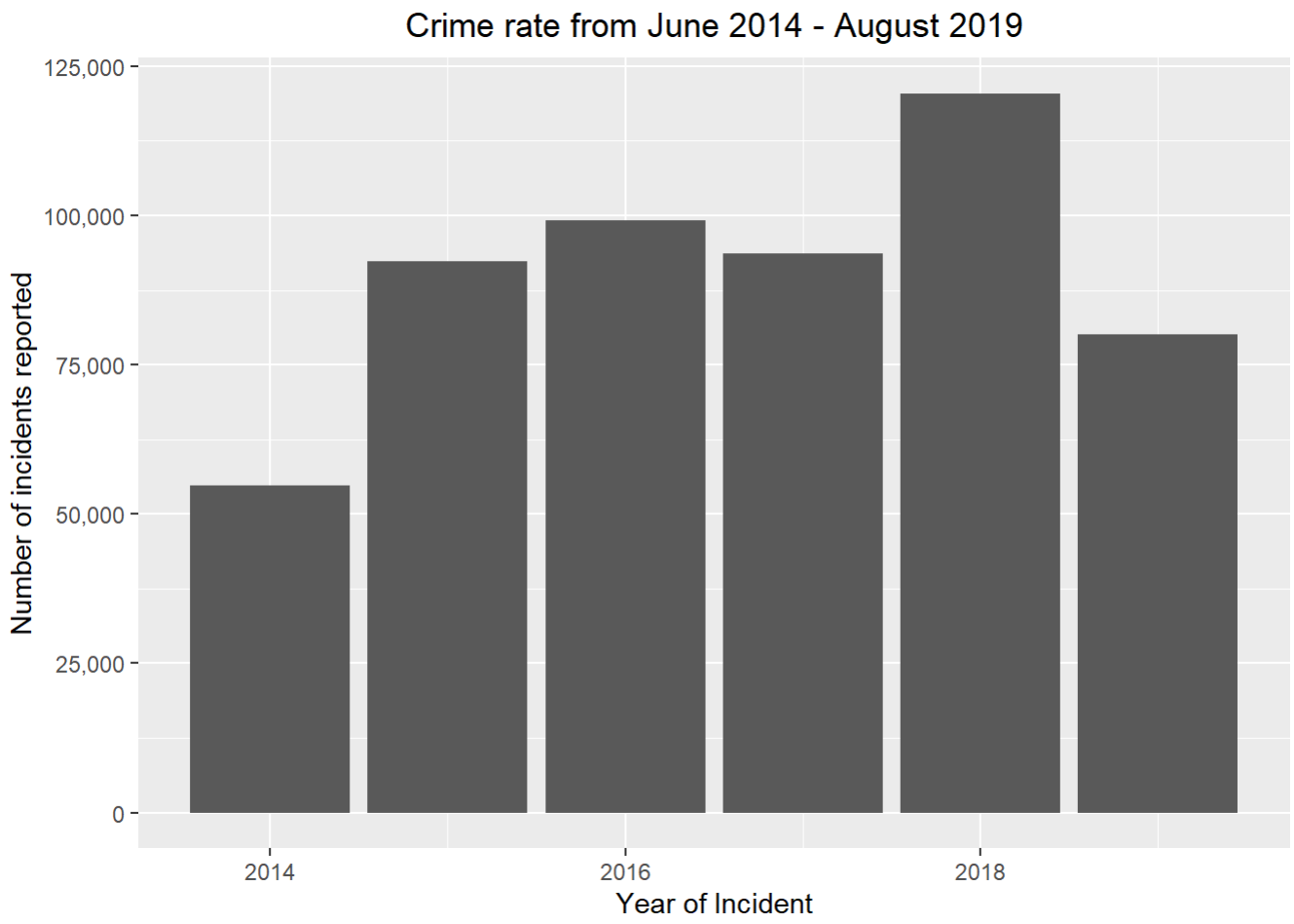
DATA VISUALIZATION

Data Visualization is a powerful way to understand the data and its underlying patterns. *ggplot2* is a package to draw graphics, which implements grammar of graphics. We will perform analysis on Dallas crime data using parameters like time, year, crime type, victim condition, weapons used etc. For this analysis we will use the powerful aggregation functions like *summarise*, *tally* from *dplyr* package.

```
library(ggplot2)
library(RColorBrewer)
library(scales)
```

Let us have a look at the crime rates in Dallas from June 2014 to August 2019.

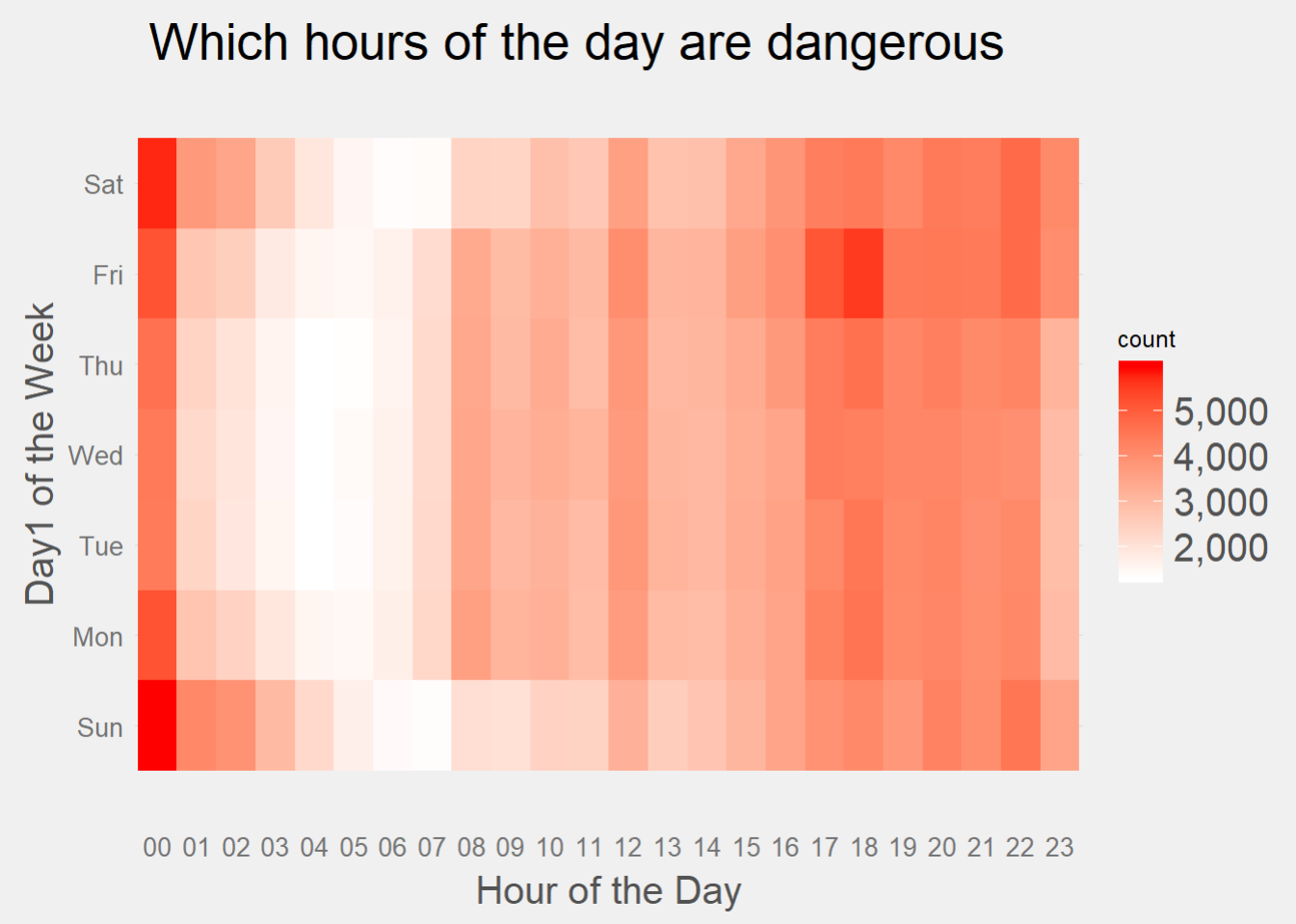
```
ggplot(crimedata, aes(`Year of Incident`)) + geom_bar() + ggtitle("Crime rate from June 2014 - A
ugust 2019") + labs( y = "Number of incidents reported", x = "Year of Incident") + theme(plot.ti
tle = element_text(hjust = 0.5)) + scale_y_continuous(labels = comma)
```



The above plot shows the occurence of crime from June 2014 to August 2019. Note that the crime rates has drastically increased in 2018 compared to the previous years. Let us also look at how crimes varied with respect to time of day, day of week and season.

```
crime_per_day <- crimedata %>%
                  group_by(`Hour of the Day`, `Day1 of the Week`) %>%
                  dplyr::summarise(count = n())
```

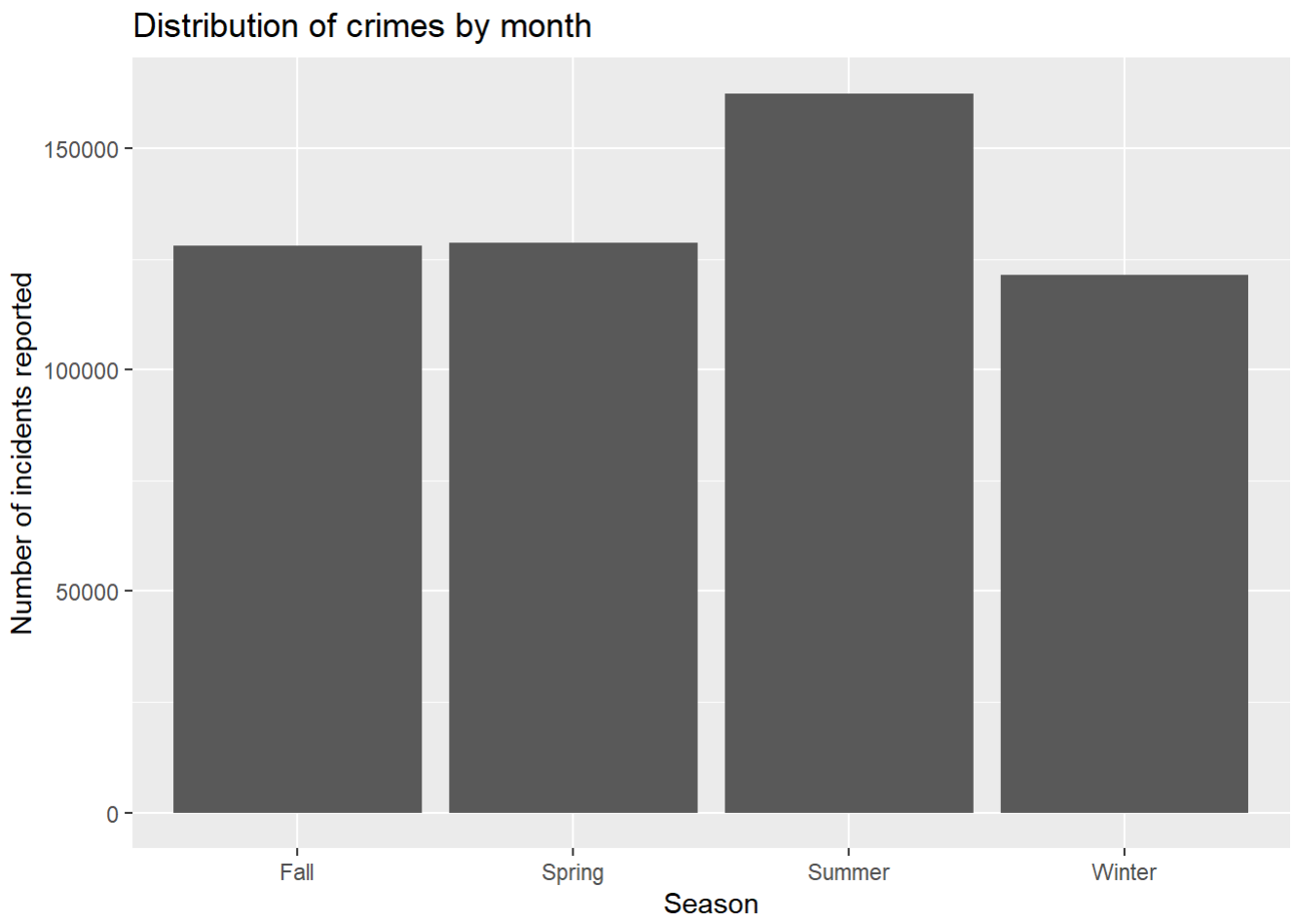
```
days <- c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
week.name <- factor(days, levels = days)
ggplot(crime_per_day, aes(x =`Hour of the Day`, y =`Day1 of the Week`, fill = count)) +
  geom_tile() +
  fte_theme() +
  scale_fill_gradient(low = "White", high = "Red", labels = comma) +
  scale_y_discrete(limits = week.name) +
  ggtitle(" Which hours of the day are dangerous")
```



A closer look at the heat map shows that most theft happens around midnight. Especially on the later part of the day on Fridays and weekends, there are significantly more number of crimes happening in the evening compared to the other days. The bar chart below witnesses more crimes during Summer months compared to others.

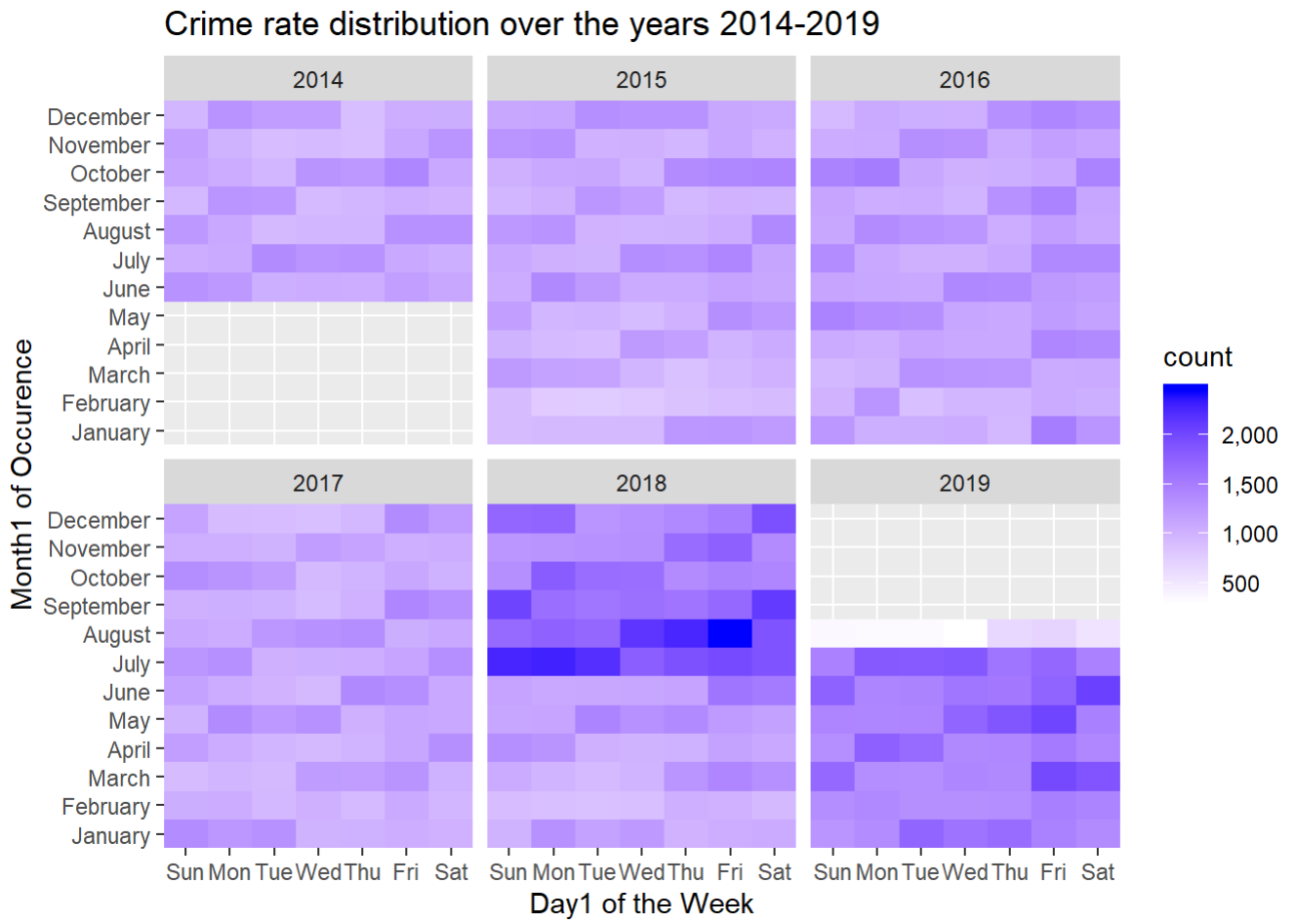
```
crime_month <- crimedata %>% group_by(Season) %>% summarise(n = n())
```

```
ggplot(crime_month, aes(x = Season , y = n)) + geom_bar(stat = "identity") + ggtitle("Distributi
on of crimes by month") +
 labs( y = "Number of incidents reported")
```




```
crime_per_week <- crimedata %>%
                  group_by(`Day1 of the Week`, `Month1 of Occurence`, `Year of Incident`) %>%
                  dplyr::summarise(count = n())
```

```
days <- c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
week.name <- factor(days, levels = days)
ggplot(crime_per_week,aes(x=`Day1 of the Week`, y = `Month1 of Occurence`, fill =count,  na.rm =
TRUE)) +
  geom_tile() +
  scale_fill_gradient(low = "White", high = "Blue", labels = comma) +
  facet_wrap( ~ `Year of Incident`) +
  scale_x_discrete(limits = week.name, expand = c(0, 0)) +
  scale_y_discrete(limits = month.name, expand = c(0, 0)) +
  ggtitle("Crime rate distribution over the years 2014-2019")
```

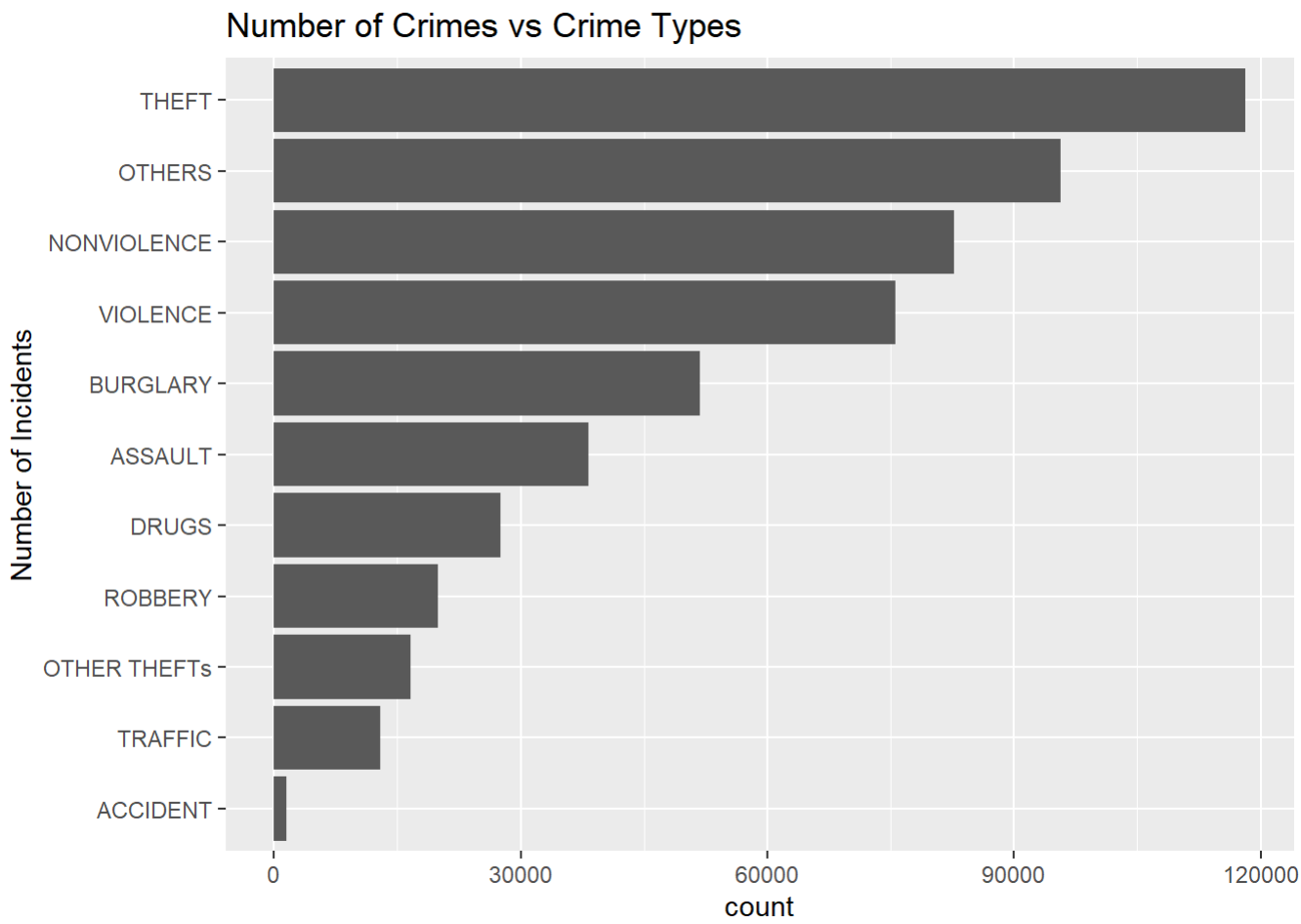


The above heat map briefly shows us the number of crimes that occurred every day of the week for all months from June 2014 till August 2019. This plot proves that crime rates are low during winter (Nov, Dec and Jan) compared to other seasons. It is more evident there is a sharp increase in crime incidents in late 2018 continuing through 2019, in comparison with other years. This can be taken as an improvement to further analyse what type of crime and where in the city there is an increase in crime events.

Now let us try to identify what type of crime is more prevalent in the city.

```
crimedata_top <- within(crimedata,
                        `Crime Type` <- factor(`Crime Type`,
                                               levels=names(sort(table(`Crime Type`),
                                                                 decreasing=FALSE))))
```

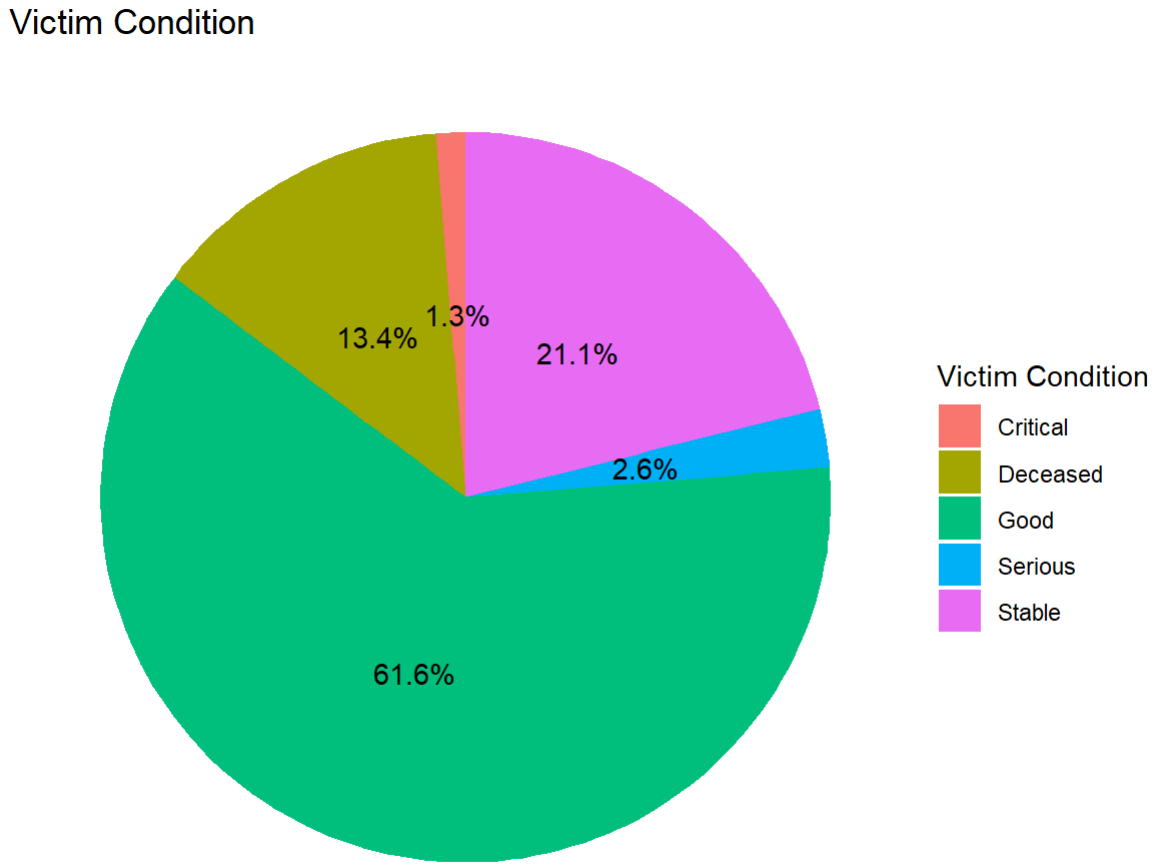
```
ggplot(crimedata_top, aes(x=`Crime Type`, na.rm = TRUE)) + geom_histogram(stat = "count") + coor
d_flip() + ggtitle("Number of Crimes vs Crime Types") + labs(x = "Number of Incidents")
```



This graph tells that there are more number of thefts happening in the city rather than any other type of crime. We shall also see how bad the victims were affected by these crimes and what type of weapons were used in the crime scene.

```
crime_by_victim <- crimedata %>%
                    filter(`Victim Condition` != "") %>%
                    group_by(`Victim Condition`) %>%         dplyr::summarise(value = n()) %>%
                    ungroup() %>%
                    mutate(per=value/sum(value)) %>%
                    arrange(desc(`Victim Condition`))
crime_by_victim$label <- scales::percent(crime_by_victim$per)
```

```
ggplot(data=crime_by_victim)+
  geom_bar(aes(x="", y=per, fill=`Victim Condition`), stat="identity", width = 1)+
  coord_polar("y", start=0)+
  theme_void() +
  geom_text(aes(x=1, y = cumsum(per) - per/2, label=label)) +
  ggtitle("Victim Condition")
```



The pie chart on the victims` condition shows that about 61.6% of the victims are in good condition and 21.1% victims are *Stable*. Although a major percentage of the victims are in good health, there are 1.3% and 13.4% victims in *Critical* and *Deceased* states respectively. We should be aiming at reducing this value. To have more idea on how the victims are affected by the crime incidents, let us do some analysis on the weapons used, what weapon is most likely used in a particular crime scene and how they affected the victims.

Recalling the new factor variable created `Weapon Type` which groups similar `Weapons Used` . Levels of `Weapon Type` are :

```
levels(crimedata$`Weapon Type`)
```

```
## [1] "Drugs"      "Fire"       "Gun"        "Hands/Feet" "Knife"
## [6] "No Weapons" "Others"     "Threat"     "Vehicle"
```

Let us also find how many values of each levels are present in our dataset.

```
crimedata %>% count(crimedata$`Weapon Type`, sort = TRUE)
```

crimedata$`Weapon Type` <fctr>	**n** <int>
Others	414316
No Weapons	51020

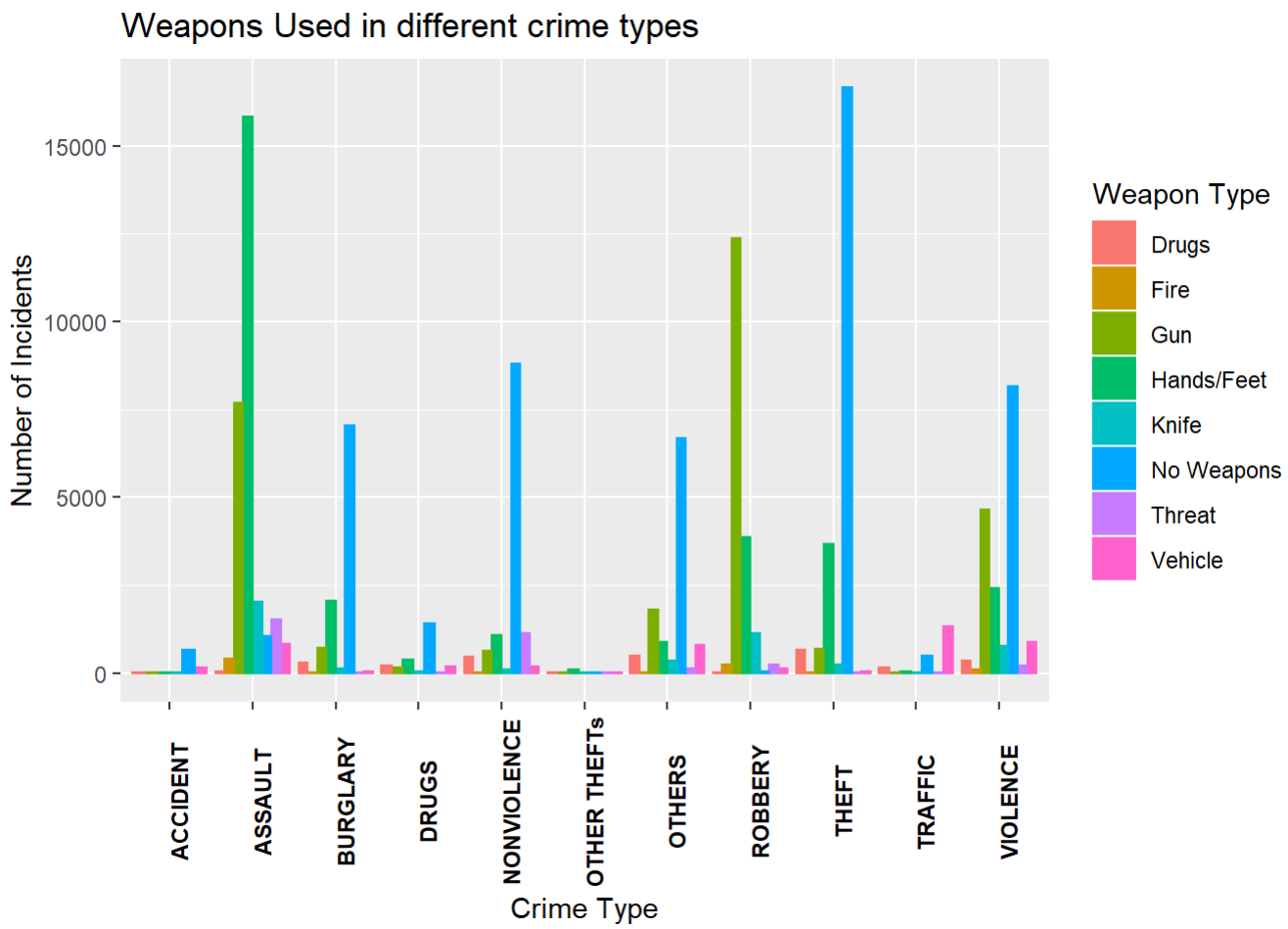
crimedata$`Weapon Type` <fctr>	n <int>
Hands/Feet	30302
Gun	28758
Knife	4773
Vehicle	4551
Threat	3271
Drugs	2754
Fire	759

9 rows

There are 9 levels in `Weapon Type` variable however more than 76% of the incidents fall under *Others* category which includes weapons like Blunt, Assault, Crowbar, Asphixation, BlackJack/Club, etc. For the purposes of this study to accurately analyze the weapon usage in various crime categories and not skew the results, we will be ignoring *Others* category from the dataset.

```
weapons_used <- crimedata %>% filter(`Weapon Type` != "Others") %>% group_by(`Crime Type`, `Weap
on Type`) %>% summarise(count = n())
```

```
ggplot(weapons_used, aes(x = `Crime Type`, y = count, color = `Weapon Type`, fill = `Weapon Type
`)) + geom_bar(stat = "identity", position = position_dodge()) + theme(axis.text.x = element_tex
t(angle=90, face="bold", colour="black")) +
  ggtitle("Weapons Used in different crime types") + labs(y = "Number of Incidents")
```

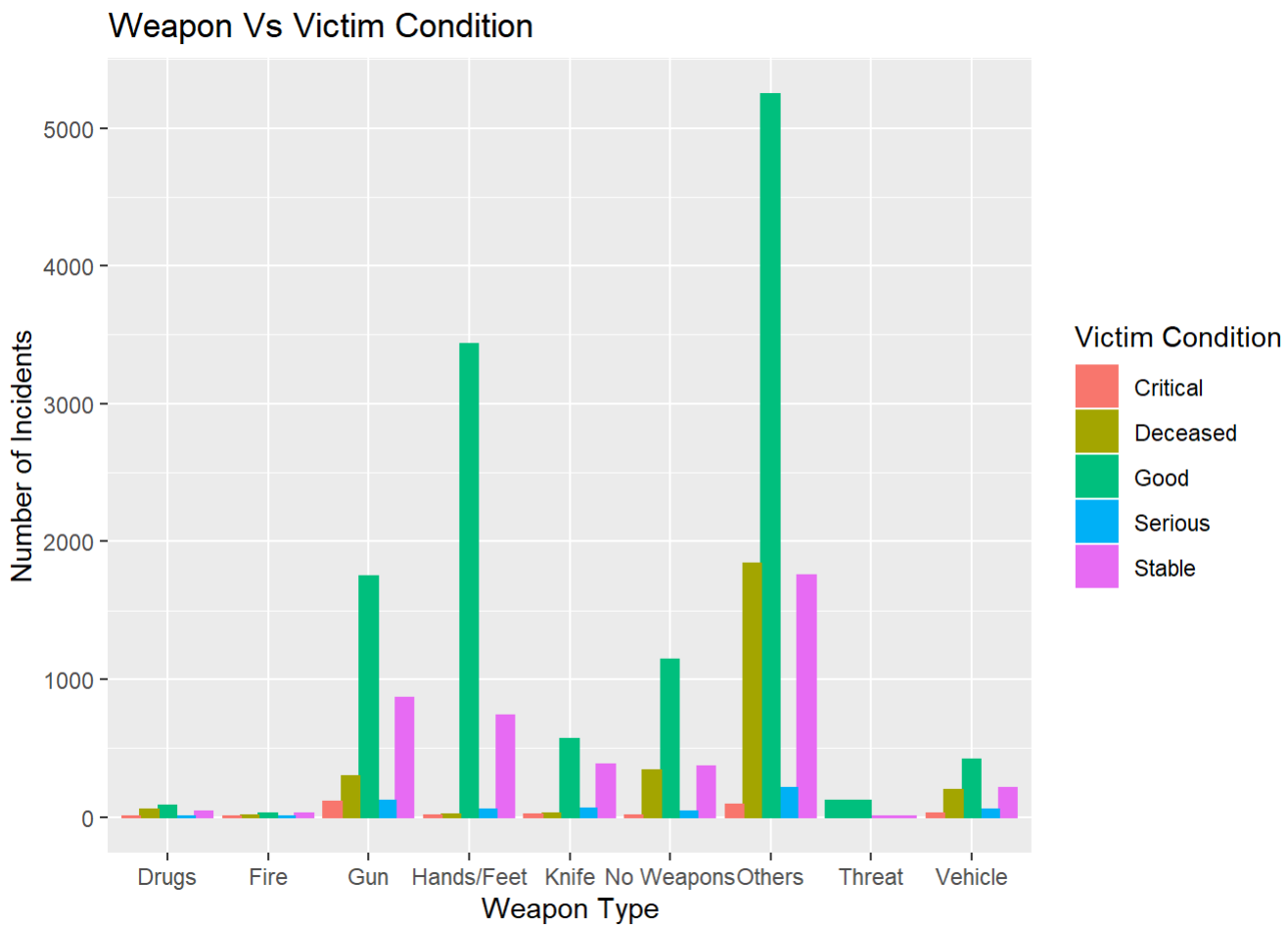


- As we have seen earlier, *Theft* is the most happening crime in the city compared to other crime types
- Most of the *Theft* cases reported doesn't involve any weapon
- Hands/Feet were the mode of operation for most of the reported *Assault* cases
- Non-violent crimes that include Disorderly conduct, Gambling, Fraud, Failing to ID etc. and Burglary cases show no weapons used on most incidents
- Guns were primarily used in large numbers in crimes like Assault, Robbery and Violence. Gun violence results in thousands on deaths and injuries annually in the U.S

Let us also try to understand how the victims were affected by different types of weapons used in various crime scenes.

```
weapons_victim <- crimedata %>% filter(`Victim Condition` != "") %>% group_by(`Weapon Type`, `Vi
ctim Condition`) %>% tally(name = "cnt")
```

```
ggplot(weapons_victim, aes(x = `Weapon Type`, y = cnt, color = `Victim Condition`, fill = `Victi
m Condition`)) + geom_bar(stat = "identity", position =position_dodge()) +
   ggtitle("Weapon Vs Victim Condition") + labs(y = "Number of Incidents")
```

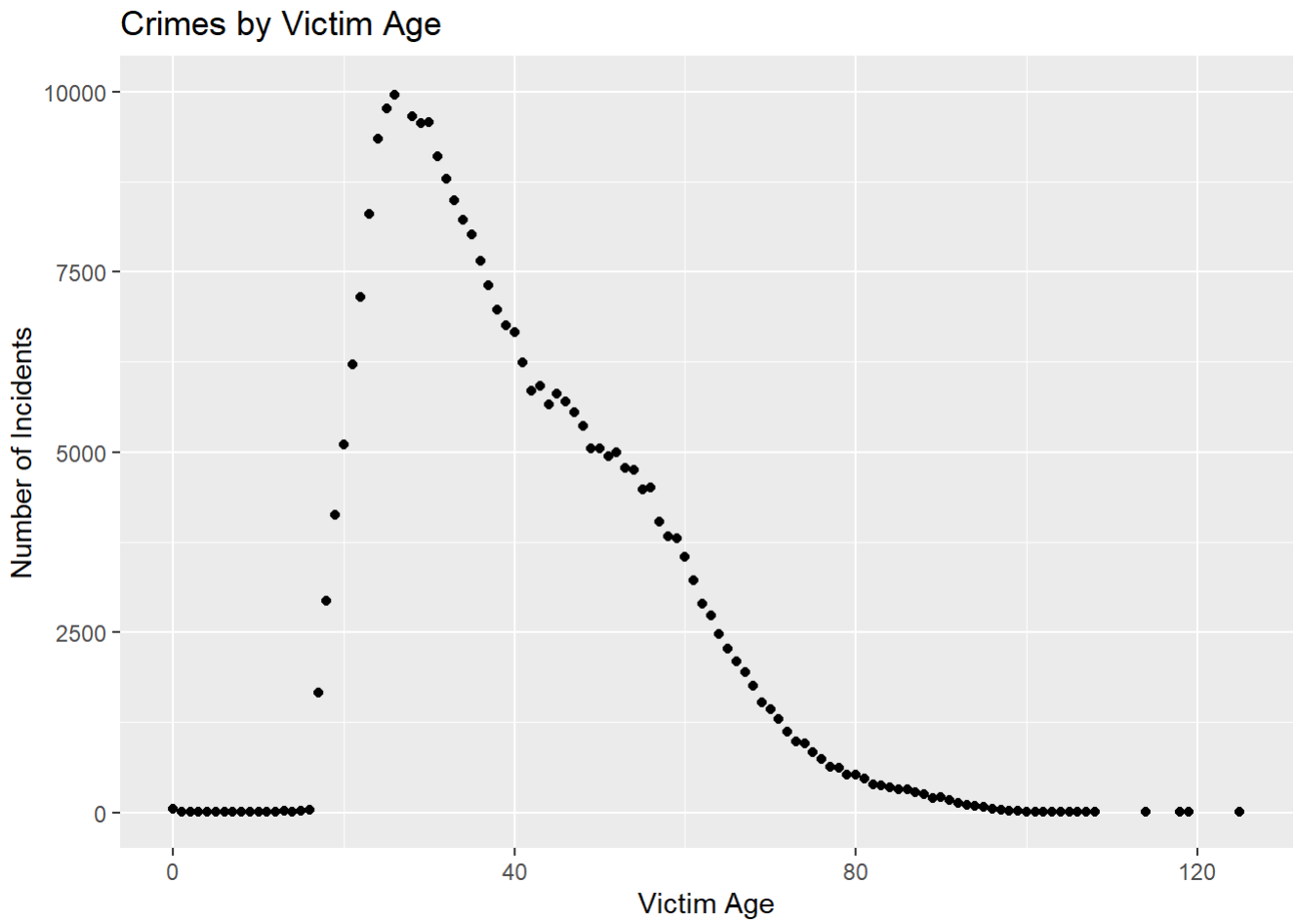


The Weapon type *Others* that includes Blunt, Assault, Crowbar, BlackJack/Club etc., affects more victims in comparison with other types. A majority of them are said to be in Good condition, but not to forget there are also some deaths recorded in this category. This also backs up the fact that gun violence has dropped dramatically in 3 states - New York, California and Texas because of its stringent gun laws. This drop approaches 74% in Dallas.

It is also very important to find out what age group people are most affected by the crime incidents happening in the city.

```
victim_affected <- crimedata %>% group_by(`Victim Age`) %>% summarize(n = n())
```

```
ggplot(victim_affected, aes(x=`Victim Age`, y = n)) + geom_point() + ylim(0,10000) + labs(y = "N
umber of Incidents") + ggtitle("Crimes by Victim Age")
```

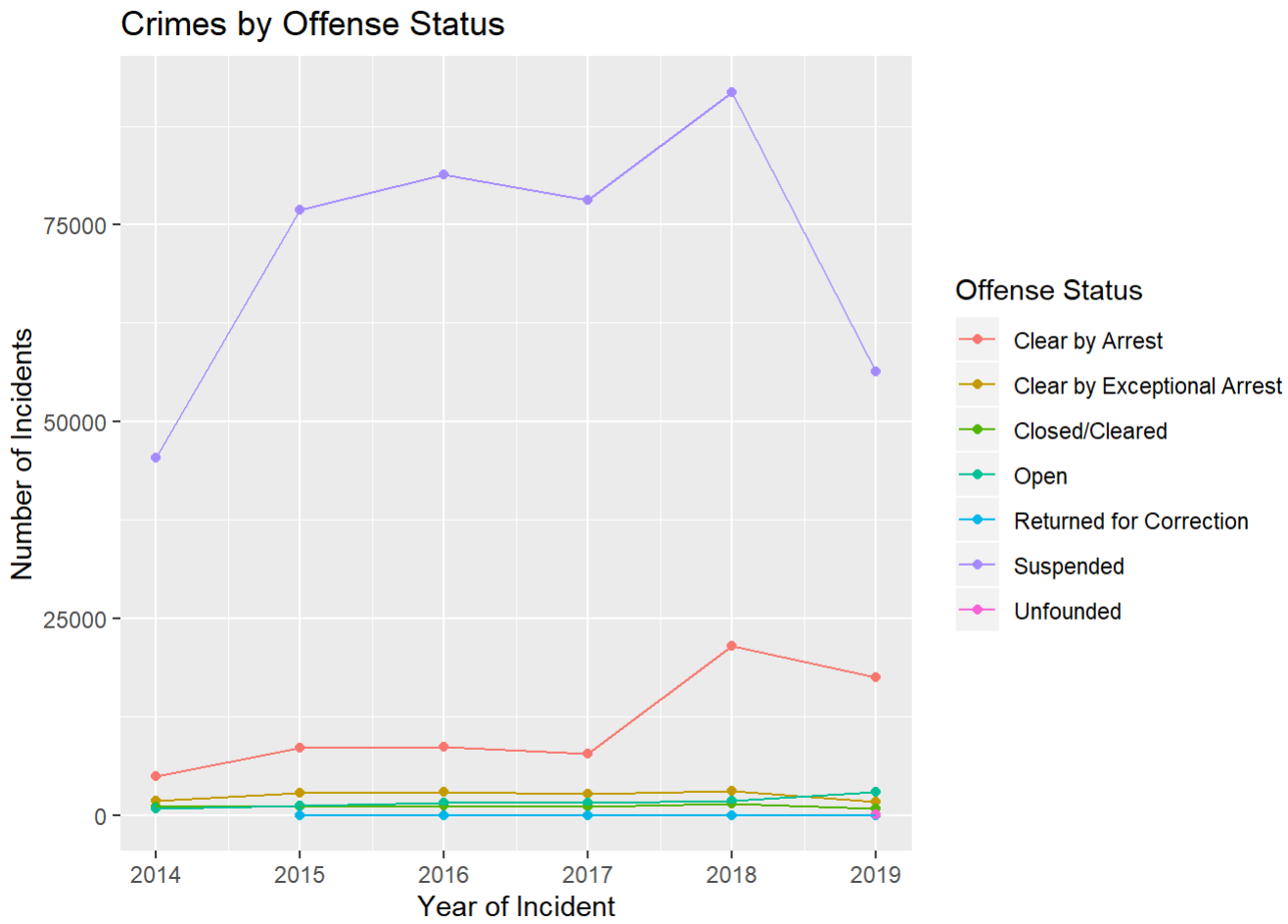


This plots shows us that people in age 20-40 are extremely affected by the crimes. We can also see some casualities in the age group 80-120.

On an average 285 crime events are reported to the Dallas Police Department each day. It will be interesting to find out the status of these crime events.

```
offense_trends <- crimedata %>% filter(`Offense Status` != "") %>% group_by(`Offense Status`,`Ye
ar of Incident`) %>% summarize(count = n())
```

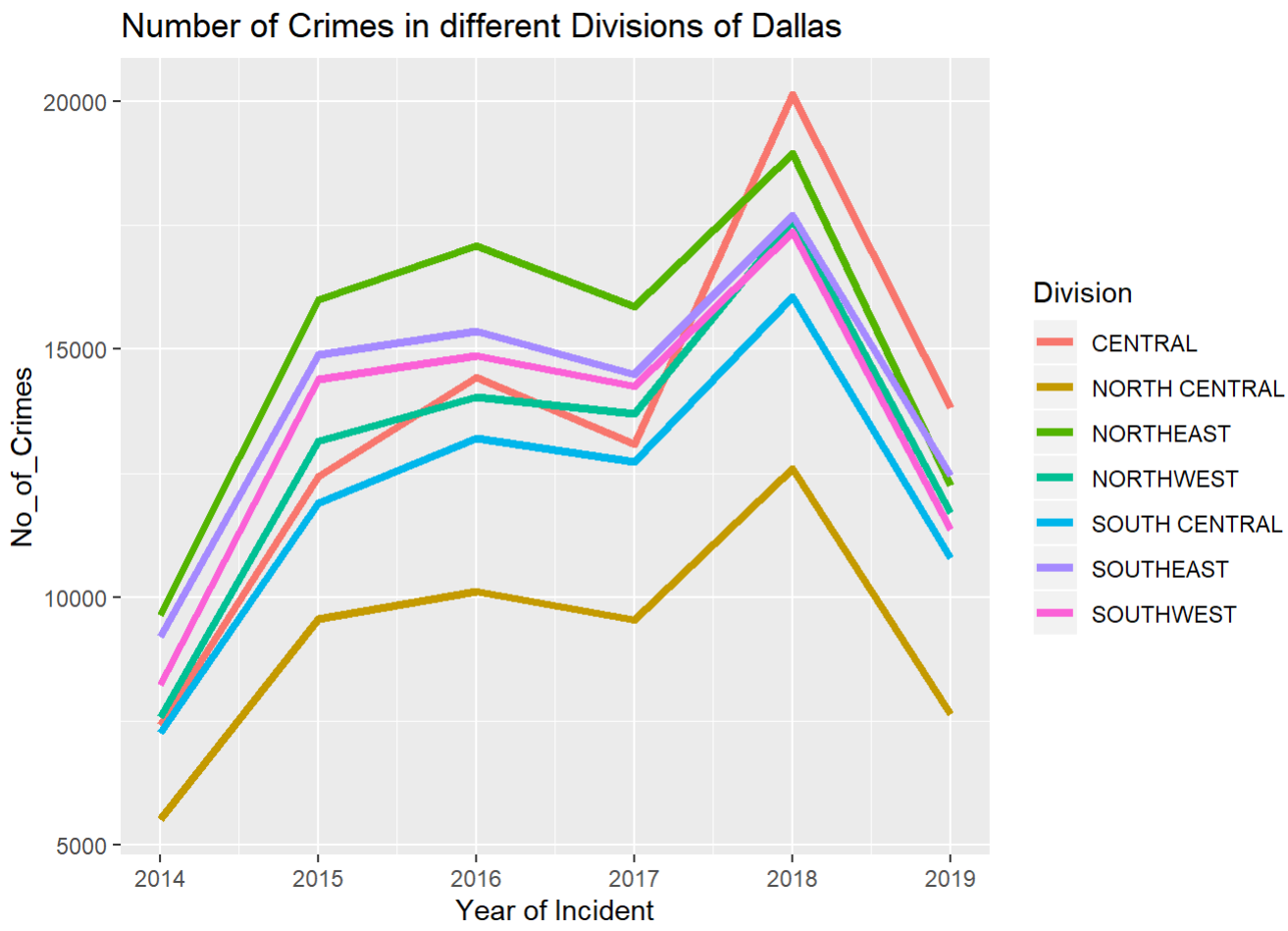
```
ggplot(offense_trends, aes(x=`Year of Incident`, y = count, color = `Offense Status`)) + geom_po
int() + geom_line() + labs(y = "Number of Incidents") + ggtitle("Crimes by Offense Status")
```



A large number of cases are in *Suspended* state - meaning they are not actively being worked, but not *Closed*. When new evidences are received, the cases may become active. After a period of time, they go cold and are then archived. It is also good to note that certain amount of crimes are *Cleared by arrests* and few are even *Closed*.

```
crime_trends <- crimedata %>% filter(Division != "") %>% group_by(Division, `Year of Incident`)
 %>% tally(name = "No_of_Crimes")
```

```
ggplot(crime_trends, aes(x=`Year of Incident`, y = No_of_Crimes , color = Division, size = 0)) +
geom_line(size = 1.5) + ggtitle("Number of Crimes in different Divisions of Dallas")
```

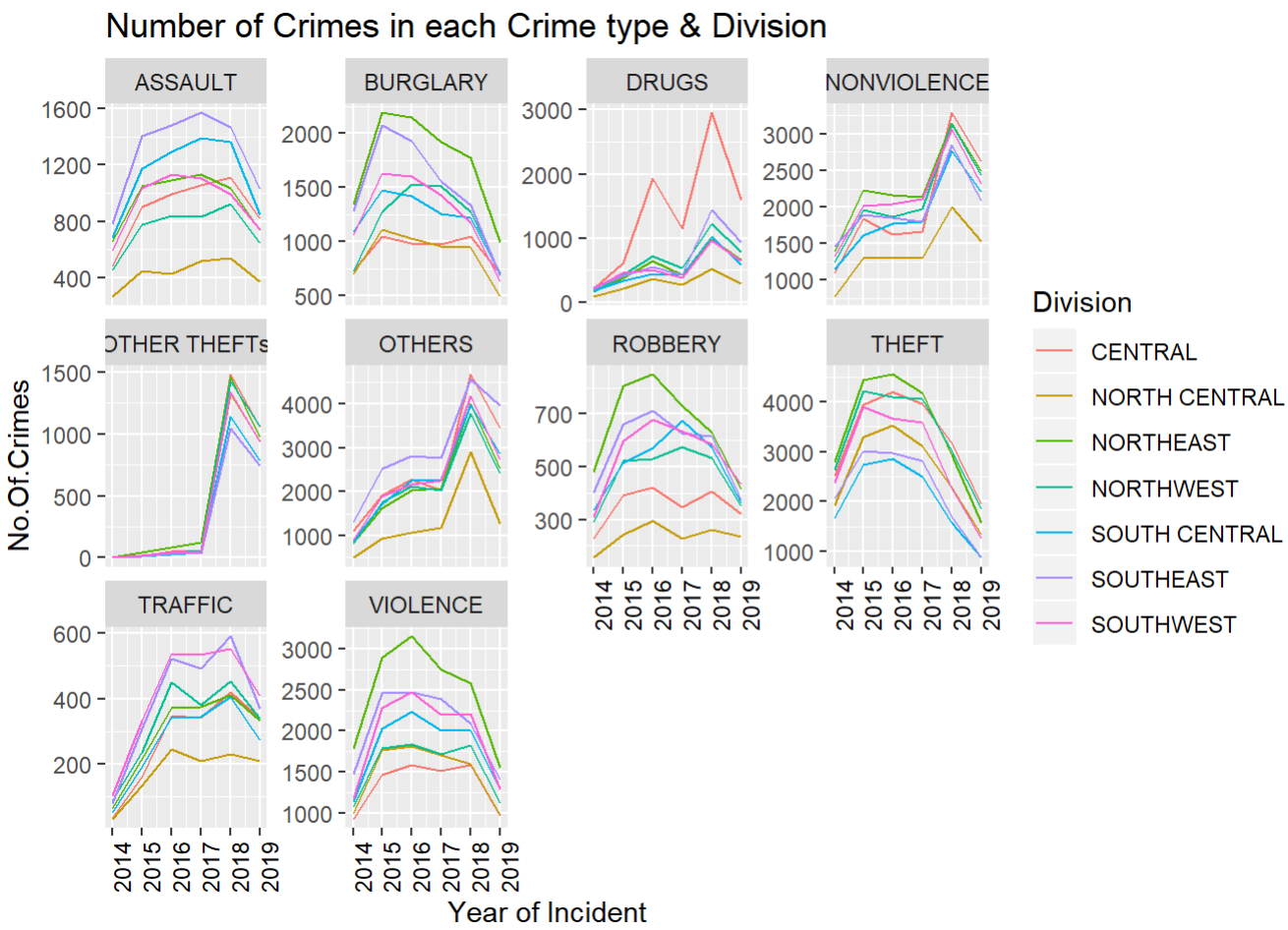


This plot clearly explains how the crime rate has increased in various parts of the city over the years. *NORTH CENTRAL* Dallas has recorded lesser crime rates. *CENTRAL* and *NORTHEAST* Dallas are holding more number of crime incidents.

It is also very important to identify what type of crime is recorded high in all the divisions of Dallas. For this, let us exclude `Crime type` *Accidents* and look at other crimes.

```
crime_div <- crimedata %>% filter(`Crime Type` != "ACCIDENT" & Division != "") %>% group_by(`Cri
me Type`, Division, `Year of Incident`) %>% tally(name = "No.Of.Crimes")
```

```
ggplot(crime_div, aes(x = `Year of Incident`, y = `No.Of.Crimes`, color = Division)) + geom_line
() +
  facet_wrap( ~ `Crime Type`, scales = "free_y") +theme(axis.text.x = element_text(angle=90, col
our="black")) + ggtitle("Number of Crimes in each Crime type & Division")
```



From our previous analysis we know that *NORTH CENTRAL* Dallas is the safest place in the city. But *Theft* in *NORTH CENTRAL* region is higher than the *SOUTH CENTRAL* region.

MODELING

The main objective of this project is to predict the crime rate for the future year. Our dataset fits right in the supervised learning technique.

Supervised learning is a learning where it takes a sample of input and desired outputs(training data), analyses them and effectively produces correct output data. The correct output produced is entirely based on the training data. Supervised learning can be of two categories:

- Classification : When the output variable is a category
- Regression : When the output variable is a real or continuous value

Clearly, our problem falls under the regression category.

With our visualizations, we learnt that there is a significant increase in crime incidents with respect to time of the day. So, we try to build a model with respect to the time(Hour) of the day the crime incidents happened. We need to prepare our data for modeling. The predictor variables to solve our problems are : Year of Incident and Hour of the Day

```
crime_by_year <- crimedata %>% group_by(`Year of Incident`, `Hour of the Day`) %>% summarise(tot
al = n())
crime_by_year <- as.data.frame(crime_by_year)
crime_by_year$`Hour of the Day` <- as.factor(crime_by_year$`Hour of the Day`)
```

- **ANOVA**

Analysis of Variance (ANOVA) is a statistical method used to test the differences between two or more means.Inferences about means are made by analysing the variance.

```
res.aov <- aov(crime_by_year$total ~  crime_by_year$`Hour of the Day` + crime_by_year$`Year of I
ncident`)
summary(res.aov)
```

```
##                                  Df    Sum Sq  Mean Sq F value   Pr(>F)
## crime_by_year$`Hour of the Day`   23 190802715  8295770    12.0  < 2e-16
## crime_by_year$`Year of Incident`   1  25030502 25030502    36.2 2.02e-08
## Residuals                        119  82292926   691537
##
## crime_by_year$`Hour of the Day`  ***
## crime_by_year$`Year of Incident` ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA proves that the two independent or predictor variables are *statistically significant*

- **LOGISTIC REGRESSION**

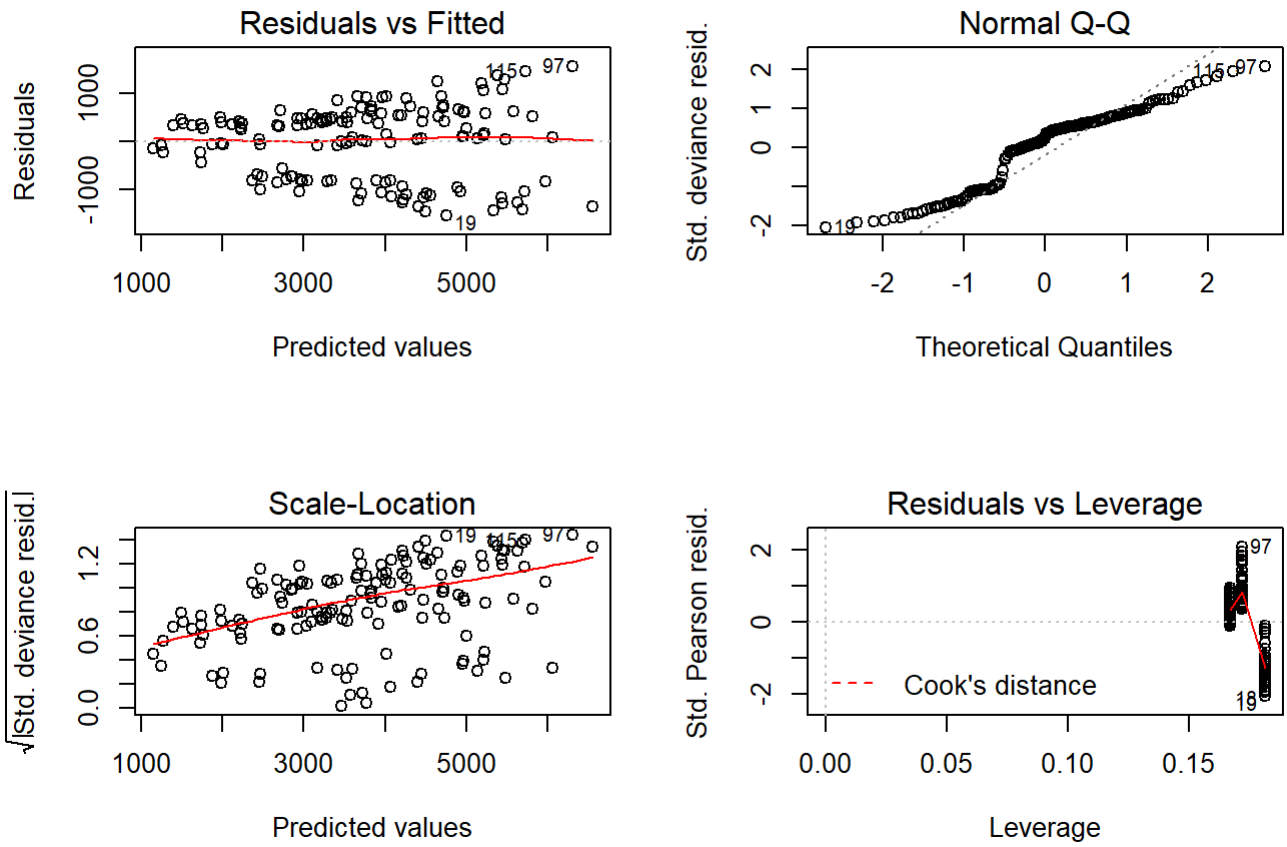
Logistic Regression is a method for fitting a regression curve, y = f(x). The typical use of the model is to predict y given a set of predictors x. The predictors can be continuous, categorical or a mix of both. Here the function to be called is *glm()*.

```
glm_mod <- glm(total ~`Hour of the Day` + `Year of Incident` , data = crime_by_year)
summary(glm_mod)
```

```
##
## Call:
## glm(formula = total ~ `Hour of the Day` + `Year of Incident`,
##     data = crime_by_year)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1549.9   -801.9    210.1    515.5   1569.7
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -486334.50   81824.83  -5.944 2.85e-08 ***
## `Hour of the Day`01   -2540.33     480.12  -5.291 5.62e-07 ***
## `Hour of the Day`02   -2897.17     480.12  -6.034 1.86e-08 ***
## `Hour of the Day`03   -3589.17     480.12  -7.476 1.43e-11 ***
## `Hour of the Day`04   -4056.33     480.12  -8.449 8.47e-14 ***
## `Hour of the Day`05   -4182.17     480.12  -8.711 2.07e-14 ***
## `Hour of the Day`06   -4080.00     480.12  -8.498 6.51e-14 ***
## `Hour of the Day`07   -3604.17     480.12  -7.507 1.22e-11 ***
## `Hour of the Day`08   -2290.33     480.12  -4.770 5.27e-06 ***
## `Hour of the Day`09   -2651.83     480.12  -5.523 1.99e-07 ***
## `Hour of the Day`10   -2340.50     480.12  -4.875 3.40e-06 ***
## `Hour of the Day`11   -2597.50     480.12  -5.410 3.31e-07 ***
## `Hour of the Day`12   -1618.67     480.12  -3.371 0.001009 **
## `Hour of the Day`13   -2485.33     480.12  -5.177 9.30e-07 ***
## `Hour of the Day`14   -2469.33     480.12  -5.143 1.08e-06 ***
## `Hour of the Day`15   -2040.67     480.12  -4.250 4.27e-05 ***
```

On evaluating the Estimate coefficients in the summary, we predict that there will be around 115,884 crime incidents to happen in the year 2020.

```
par(mfrow=c(2, 2))
plot(glm_mod)
```



Looking at the Residuals vs Fitted plot, the prediction made by the model is in x-axis and the accuracy of prediction is on the y-axis. The distance from the line 0 is how bad the prediction was for that value.

Residual = Actual - Predicted

Positive values for the residual means the prediction was too low, negative values means the prediction was too high and 0 means the guess was exactly correct. There is a room for improvement in our model.

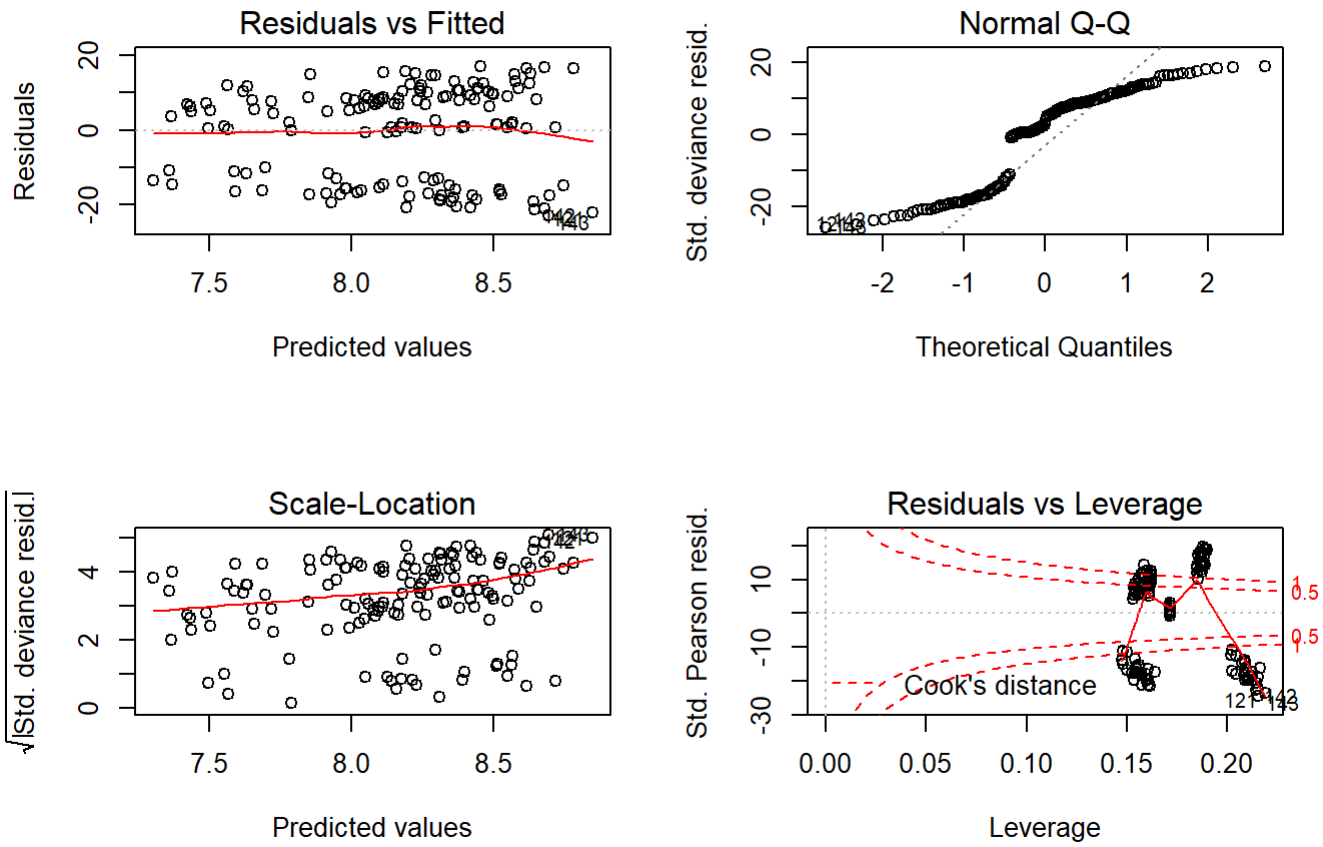
- **POISSON REGRESSION**

Poisson regression is used to model response variables (Y-values) that are counts. It tells which explanatory variables have a statistically significant effect on the response variables. We perform poisson model analysis with glm() where *family = poisson*

```
poisson.glm <- glm(total ~`Hour of the Day` +`Year of Incident`, data = crime_by_year, family =
"poisson")
summary(poisson.glm)
```

```
##
## Call:
## glm(formula = total ~ `Hour of the Day` + `Year of Incident`,
##     family = "poisson", data = crime_by_year)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -22.902  -14.645    3.199    9.143   17.018
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.228e+02  1.613e+00  -76.16   <2e-16 ***
## `Hour of the Day`01 -5.579e-01  8.778e-03  -63.55   <2e-16 ***
## `Hour of the Day`02 -6.687e-01  9.100e-03  -73.49   <2e-16 ***
## `Hour of the Day`03 -9.266e-01  9.946e-03  -93.17   <2e-16 ***
## `Hour of the Day`04 -1.148e+00  1.079e-02 -106.38   <2e-16 ***
## `Hour of the Day`05 -1.217e+00  1.108e-02 -109.83   <2e-16 ***
## `Hour of the Day`06 -1.161e+00  1.084e-02 -107.03   <2e-16 ***
## `Hour of the Day`07 -9.330e-01  9.968e-03  -93.60   <2e-16 ***
## `Hour of the Day`08 -4.870e-01  8.585e-03  -56.72   <2e-16 ***
## `Hour of the Day`09 -5.912e-01  8.873e-03  -66.64   <2e-16 ***
## `Hour of the Day`10 -5.008e-01  8.622e-03  -58.08   <2e-16 ***
## `Hour of the Day`11 -5.748e-01  8.826e-03  -65.13   <2e-16 ***
## `Hour of the Day`12 -3.181e-01  8.162e-03  -38.97   <2e-16 ***
## `Hour of the Day`13 -5.418e-01  8.734e-03  -62.04   <2e-16 ***
## `Hour of the Day`14 -5.372e-01  8.721e-03  -61.60   <2e-16 ***
## `Hour of the Day`15 -4.208e-01  8.413e-03  -50.02   <2e-16 ***
```

```
par(mfrow=c(2,2))
plot(poisson.glm)
```



To perform the goodness of fit for this model, we need to look at overdispersion. Overdispersion is a situation where the residual deviance of the glm is large relative to the residual degrees of freedom. If the ratio of the residual deviance to the residual degrees of freedom exceeds 1.5, then the model is said to be overdispersed.

```
poisson.glm$deviance/poisson.glm$df.residual
```

```
## [1] 179.2275
```

One potential solution to avoid overdispersion is to use a quasi model.

- **QUASSIPOSSION REGRESSION**

Over-dispersion is a problem if the conditional variance (residual variance) is larger than the conditional mean. One way to check for and deal with over-dispersion is to run a quasi-poisson model.

```
qpoisson.glm <- glm(total ~`Hour of the Day` +`Year of Incident`, data = crime_by_year, family =
"quasipoisson")
summary(qpoisson.glm)
```

```
##
## Call:
## glm(formula = total ~ `Hour of the Day` + `Year of Incident`,
##     family = "quasipoisson", data = crime_by_year)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -22.902  -14.645    3.199    9.143   17.018
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -122.81119   21.19480  -5.794 5.73e-08 ***
## `Hour of the Day`01   -0.55789    0.11538  -4.835 4.02e-06 ***
## `Hour of the Day`02   -0.66873    0.11961  -5.591 1.46e-07 ***
## `Hour of the Day`03   -0.92664    0.13072  -7.089 1.04e-10 ***
## `Hour of the Day`04   -1.14807    0.14186  -8.093 5.63e-13 ***
## `Hour of the Day`05   -1.21716    0.14566  -8.356 1.39e-13 ***
## `Hour of the Day`06   -1.16070    0.14254  -8.143 4.32e-13 ***
## `Hour of the Day`07   -0.93304    0.13102  -7.121 8.81e-11 ***
## `Hour of the Day`08   -0.48695    0.11284  -4.315 3.31e-05 ***
## `Hour of the Day`09   -0.59122    0.11662  -5.070 1.48e-06 ***
## `Hour of the Day`10   -0.50079    0.11333  -4.419 2.20e-05 ***
## `Hour of the Day`11   -0.57484    0.11601  -4.955 2.42e-06 ***
## `Hour of the Day`12   -0.31807    0.10727  -2.965 0.003658 **
## `Hour of the Day`13   -0.54184    0.11480  -4.720 6.49e-06 ***
## `Hour of the Day`14   -0.53722    0.11463  -4.687 7.45e-06 ***
## `Hour of the Day`15   -0.42080    0.11058  -3.805 0.000225 ***
```

The summary shows that the residual deviance has not changed. The dispersion parameter, which was forced to be 1 in our last model, is allowed to be estimated here. This model again has overdispersion.

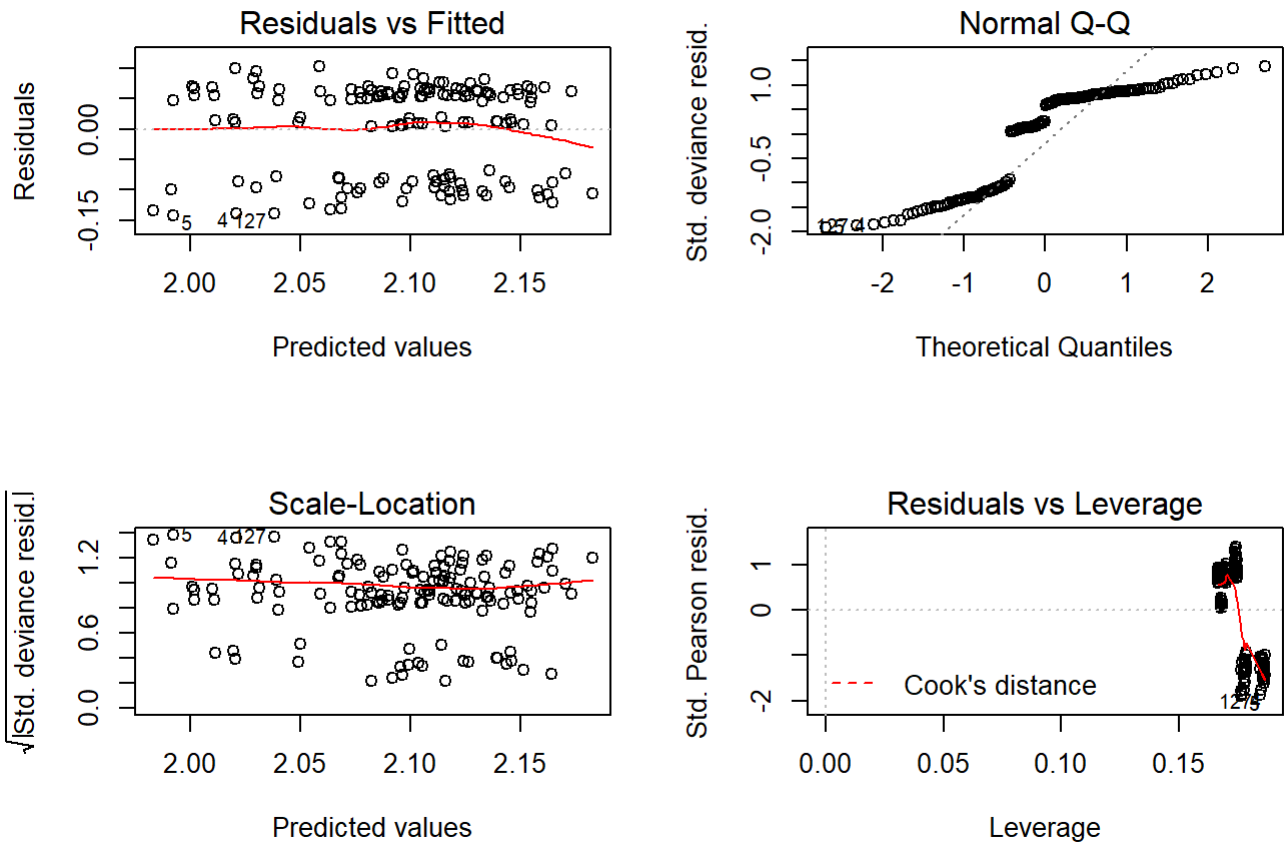
- **LOG TRANSFORMATION OF GLM**

We can transform our data, by performing a mathematical operation and use these transformed values in our statistical tests. We can choose from an infinite number of transformations, but the most common ones are *log* and *sqrt* transformations.

```
log_qpoisson.glm <- glm(log(total) ~ `Hour of the Day` + `Year of Incident`, data = crime_by_yea
r, family = "quasipoisson")
summary(log_qpoisson.glm)
```

```
##
## Call:
## glm(formula = log(total) ~ `Hour of the Day` + `Year of Incident`,
##     family = "quasipoisson", data = crime_by_year)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.14215  -0.08649   0.03165   0.06108   0.10262
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -16.745248   2.820741  -5.936 2.95e-08 ***
## `Hour of the Day`01  -0.067557   0.016324  -4.138 6.56e-05 ***
## `Hour of the Day`02  -0.081773   0.016384  -4.991 2.08e-06 ***
## `Hour of the Day`03  -0.115020   0.016528  -6.959 2.00e-10 ***
## `Hour of the Day`04  -0.143829   0.016655  -8.636 3.11e-14 ***
## `Hour of the Day`05  -0.153142   0.016697  -9.172 1.71e-15 ***
## `Hour of the Day`06  -0.144866   0.016660  -8.696 2.25e-14 ***
## `Hour of the Day`07  -0.114255   0.016525  -6.914 2.51e-10 ***
## `Hour of the Day`08  -0.058602   0.016286  -3.598 0.000468 ***
## `Hour of the Day`09  -0.072305   0.016344  -4.424 2.16e-05 ***
## `Hour of the Day`10  -0.060263   0.016293  -3.699 0.000329 ***
## `Hour of the Day`11  -0.068787   0.016329  -4.212 4.94e-05 ***
## `Hour of the Day`12  -0.037796   0.016200  -2.333 0.021321 *
## `Hour of the Day`13  -0.064829   0.016313  -3.974 0.000122 ***
## `Hour of the Day`14  -0.064641   0.016312  -3.963 0.000127 ***
## `Hour of the Day`15  -0.050087   0.016251  -3.082 0.002554 **
```

```
par(mfrow=c(2, 2))
plot(log_qpoisson.glm)
```



There are several other predicting techniques that can be built to fit a better model and predict crime rates for future years. Such techniques can be used to further investigate and identify crime prone localities, help deploy security systems and make use of them effectively. In this way we can try to reduce future crime rates and bring them under control.