AMOD 5430H-Project

Yashaswini Reddy Terala

About Dataset

find and load a dataset with at least 3000 samples and 10 attributes. You need to include at least one numerical attribute type, and one categorical attribute. The dataset cannot be included with the base distribution of R. (1 mark)

I'm using the R programming language to visualise the video game sales data I downloaded from Kaggle for this assignment. One of the trusted sites to use datasets from is regarded as Kaggle. [1]

Dataset:

Preprocessing and visualising a dataset with temporal and/or spatial data is necessary for this project. For this, I've chosen the "Crime in Los Angeles" dataset.

Description:

```
Crimes committed in the City of Los Angeles since the year 2020 are depicted in this dataset. The data w

Our dataset has 28 columns which are described below.
```

1. DR NO - Division of Records Number: A formal file number composed of a 5-digit region ID, a 2-digit year, and a 2-digit year.
2. Date Rptd - The day the crime was reported
3. DATE OCC - The time the offence was committed
4. TIME OCC - Observed in 24-hour military time.
5. AREA - The LAPD is divided into 21 Geographic Areas, which are Community Police Stations. These geographic regions are numbered 1 through 21 in order.
6. AREA NAME – Each of the 21 Geographic Areas or Patrol Divisions is given a name that refers to a local landmark or the neighbourhood it is in charge of.
7. Rpt Dist No. A four-digit identifier that identifies a geographical area's subdivision. For statistical comparisons, all crime records include the "RD" that the offence took place in.
8. Part 1-2
9. Crm Cd - The crime committed is indicated. (Identical to Crime Code 1)
10. Crm Cd Desc - Provides a definition of the Crime Code.
11. Mocodes - Modus Operandi: The suspect's alleged involvement in the crime's commission.
12. Victim Age - The victim's age
13. Victim Sex - Gender of the Victim

14. Victim Descent - The victim's descent
15. Premis Cd - The kind of building, vehicle, or setting where the crime was committed.
16. Premis Desc - Describes the given Premise Code.
17. Weapon Utilized Cd: The kind of weapon that was used in the crime.
18. Weapon Desc - Provides a description of the submitted Weapon Used Code. Status - The case's status. (IC is the standard)
19. Status Desc - Describes the supplied Status Code.
20. Crm Cd 1 - The crime committed is indicated. The first and most serious crime code is 1. The offences listed in Crime Codes 2, 3, and 4 are each less serious. More serious crimes have lower crime class numbers.
21. Crm Cd 2 – crime code 2. May have a code for a different, less serious offence than Crime Code 1.
22. Crm Cd 3 - crime code 3. May have a code for a different, less serious offence than Crime Code 1.
23. Crm Cd 4 - crime code 4. May have a code for a different, less serious offence than Crime Code 1.
24. LOCATION – Nearest Street address of crime incident.
25. Cross Street - Cross Street of rounded Address.
26. LAT – Approximate Latitude of the location.
27. LON – Approximate Longitude of the location.

    Accuracy and precision, completeness and comprehensiveness, reliability and consistency, timeliness and relevance, granularity and uniqueness, availability, and accessibility are few of the crucial qualities of any data set.

Precision and Accuracy: Accuracy relates to how exact the data is. A dataset typically cannot contain any inaccurate information and must accurately represent the data without giving the wrong impression. This precision and accuracy have a part related to its intended application. Accuracy and precision could be off-target or more expensive than necessary if it is not known how the data will be used. Since our dataset was originally handwritten and then digitalized, it contains some erroneous or missing data. To use our sample, we can use a preprocessing technique for data cleansing.

Reliability and Consistency: In today's contexts, numerous systems consume and/or get data from the same source. No matter where or from what source the data was gathered, it cannot conflict with a value found in a separate source or gathered by a different system. To put it another way, dependability in the context of data quality traits refers to the absence of contradiction between two pieces of information from distinct sources or systems. A constant and reliable system must be used to gather and store the data without error or irrational variation. The LAPD is the primary source of our data. The city of Los Angeles directly released our dataset.

Timeliness and Relevance: Data collection efforts must be justified by a legitimate rationale, which calls for data to be timely and relevant. Data that is gathered too soon or too late may be erroneous and lead to bad conclusions. Every week, data is gathered and updated for our database. Dates in the current dataset range from February 10, 2020, to November 23, 2022.

Completeness and Comprehensiveness: Inaccurate and incomplete facts are both detrimental. The overall picture was only partially displayed due to gaps in the data collection process. Uninformed decisions will be made if there isn't a clear picture of how operations are progressing. To ascertain whether the criteria are being met, it is crucial to comprehend the entire set of specifications that make up a full set of data. We need to consider whether all necessary data is available when evaluating data completeness. In this instance, we do have a thorough sample.

Accessibility and Availability: Because of restrictions imposed by laws and regulations, this quality can occasionally be challenging. Regardless of the difficulty, people still require the proper level of access to the data to do their duties. This assumes that the data is real and that access to it can be obtained. Our sample is easily reachable. Granularity and Uniqueness: The degree of detail at which data is gathered is crucial because failure to do so may lead to misunderstandings and erroneous judgments. The meaning of data that has been combined, summarised, and otherwise altered may differ from that which is inferred by the data at a lower level. It is necessary to define an adequate level of granularity to ensure that there is enough uniqueness and distinguishing characteristics to be seen. This is necessary for activities to run smoothly.

The numerous factors that influence data quality can each have a varied priority depending on the company. Depending on an organization's stage of growth or even its current business cycle, the prioritising may alter. When analysing data, we should specify key terms for our company. Then, specify the requirements for reliable, accurate data using these traits.

```
# Libraries used for Visualizations
library(ggplot2)
library(hrbrthemes)
library(viridis)
library(plotly)
library(ggthemes)
library(gganimate)
library(modelr)
library(naniar)
library(janitor)
library(data.table)
library(GGally)
library(scatterplot3d)
library(PerformanceAnalytics)
library(tidyverse)
library(leaflet)
library(stringr)
library(rgdal)
library(lubridate)
library(forecast)
library(DT)
library(prophet)
library(caret)
library(highcharter)
library(xts)
library(tidyr)
library(systemfonts)
library(gdtools)
library(classInt)
library(maptools)
library(rgdal)
library(tidyr)
library(RColorBrewer)
library(spdep)
library(tmap)

# Packages needed for Data Manipulation
library(dplyr)

# Library needed to creae a Table
library(knitr)
```

```r
# Package needed in case of creating plot grid or subplot
library(cowplot)

rm(list=ls())

fillColor = "#FFA07A"
fillColor2 = "#F1C40F"
```

```r
## Data loading in progress
data<-read.csv("Crime_Data.csv")
# Let's construct a data frame to visualise observations.
data.frame<-data.frame(data)
data.frame_3k<-head(data.frame(data),3100)
```

608592 observations of crimes reported between 2010 and November 23, 2022, make up our dataset. We have 28 properties to define our data, as was previously said. We compiled 3100 samples of crime report data into a dataframe and labelled it as data. frame 3k.

```r
## Checking the dataset
is.null(data.frame)
```

```
## [1] FALSE
```

In order to perform data preprocessing, we are looking for null data in our dataset.

```r
## Cleaning the dataset
str(data.frame)
```

```
## 'data.frame':    608592 obs. of  28 variables:
##  $ DR_NO         : int  10304468 190101086 200110444 191501505 191921269 200100501 200100502 2001005
##  $ Date.Rptd     : chr  "01/08/2020 12:00:00 AM" "01/02/2020 12:00:00 AM" "04/14/2020 12:00:00 AM" "
##  $ DATE.OCC      : chr  "01/08/2020 12:00:00 AM" "01/01/2020 12:00:00 AM" "02/13/2020 12:00:00 AM" "
##  $ TIME.OCC      : int  2230 330 1200 1730 415 30 1315 40 200 1925 ...
##  $ AREA          : int  3 1 1 15 19 1 1 1 1 17 ...
##  $ AREA.NAME     : chr  "Southwest" "Central" "Central" "N Hollywood" ...
##  $ Rpt.Dist.No   : int  377 163 155 1543 1998 163 161 155 101 1708 ...
##  $ Part.1.2      : int  2 2 2 2 2 1 1 2 1 1 ...
##  $ Crm.Cd        : int  624 624 845 745 740 121 442 946 341 341 ...
##  $ Crm.Cd.Desc   : chr  "BATTERY - SIMPLE ASSAULT" "BATTERY - SIMPLE ASSAULT" "SEX OFFENDER REGISTRA
##  $ Mocodes       : chr  "0444 0913" "0416 1822 1414" "1501" "0329 1402" ...
##  $ Vict.Age      : int  36 25 0 76 31 25 23 0 23 0 ...
##  $ Vict.Sex      : chr  "F" "M" "X" "F" ...
##  $ Vict.Descent  : chr  "B" "H" "X" "W" ...
##  $ Premis.Cd     : int  501 102 726 502 409 735 404 726 502 203 ...
##  $ Premis.Desc   : chr  "SINGLE FAMILY DWELLING" "SIDEWALK" "POLICE FACILITY" "MULTI-UNIT DWELLING (
##  $ Weapon.Used.Cd: int  400 500 NA NA NA 500 NA NA NA NA ...
##  $ Weapon.Desc   : chr  "STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)" "UNKNOWN WEAPON/OTHER WEAPO
##  $ Status        : chr  "AO" "IC" "AA" "IC" ...
##  $ Status.Desc   : chr  "Adult Other" "Invest Cont" "Adult Arrest" "Invest Cont" ...
##  $ Crm.Cd.1      : int  624 624 845 745 740 121 442 946 341 341 ...
##  $ Crm.Cd.2      : int  NA NA NA 998 NA 998 998 998 998 NA ...
##  $ Crm.Cd.3      : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ Crm.Cd.4      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ LOCATION      : chr  "1100 W  39TH                          PL" "700 S  HILL
##  $ Cross.Street  : chr  "" "" "" "" ...
##  $ LAT           : num  34 34 34 34.2 34.2 ...
##  $ LON           : num  -118 -118 -118 -118 -118 ...
```

```
summary(data.frame)
```

```
##      DR_NO            Date.Rptd           DATE.OCC            TIME.OCC
##  Min.   :      817   Length:608592      Length:608592      Min.   :   1
##  1st Qu.:201710442   Class :character   Class :character   1st Qu.: 900
##  Median :211210646   Mode  :character   Mode  :character   Median :1415
##  Mean   :211363110                                         Mean   :1337
##  3rd Qu.:220607947                                         3rd Qu.:1900
##  Max.   :229921795                                         Max.   :2359
##
##       AREA        AREA.NAME          Rpt.Dist.No      Part.1.2
##  Min.   : 1.00   Length:608592      Min.   : 101   Min.   :1.000
##  1st Qu.: 6.00   Class :character   1st Qu.: 622   1st Qu.:1.000
##  Median :11.00   Mode  :character   Median :1142   Median :1.000
##  Mean   :10.72                      Mean   :1118   Mean   :1.416
##  3rd Qu.:16.00                      3rd Qu.:1617   3rd Qu.:2.000
##  Max.   :21.00                      Max.   :2199   Max.   :2.000
##
##      Crm.Cd      Crm.Cd.Desc          Mocodes             Vict.Age
##  Min.   :110.0   Length:608592      Length:608592      Min.   : -1.00
##  1st Qu.:330.0   Class :character   Class :character   1st Qu.: 12.00
##  Median :442.0   Mode  :character   Mode  :character   Median : 31.00
##  Mean   :502.4                                         Mean   : 29.97
##  3rd Qu.:626.0                                         3rd Qu.: 45.00
##  Max.   :956.0                                         Max.   :120.00
##
##    Vict.Sex         Vict.Descent         Premis.Cd     Premis.Desc
##  Length:608592      Length:608592      Min.   :101.0   Length:608592
##  Class :character   Class :character   1st Qu.:101.0   Class :character
##  Mode  :character   Mode  :character   Median :203.0   Mode  :character
##                                        Mean   :302.4
##                                        3rd Qu.:501.0
##                                        Max.   :971.0
##                                        NA's   :7
##  Weapon.Used.Cd   Weapon.Desc          Status           Status.Desc
##  Min.   :101.0    Length:608592      Length:608592      Length:608592
##  1st Qu.:308.0    Class :character   Class :character   Class :character
##  Median :400.0    Mode  :character   Mode  :character   Mode  :character
##  Mean   :361.8
##  3rd Qu.:400.0
##  Max.   :516.0
##  NA's   :393224
##     Crm.Cd.1        Crm.Cd.2         Crm.Cd.3         Crm.Cd.4
##  Min.   :110.0   Min.   :210.0    Min.   :434.0    Min.   :821.0
##  1st Qu.:330.0   1st Qu.:998.0    1st Qu.:998.0    1st Qu.:998.0
##  Median :442.0   Median :998.0    Median :998.0    Median :998.0
##  Mean   :502.2   Mean   :955.8    Mean   :982.3    Mean   :989.3
##  3rd Qu.:626.0   3rd Qu.:998.0    3rd Qu.:998.0    3rd Qu.:998.0
```

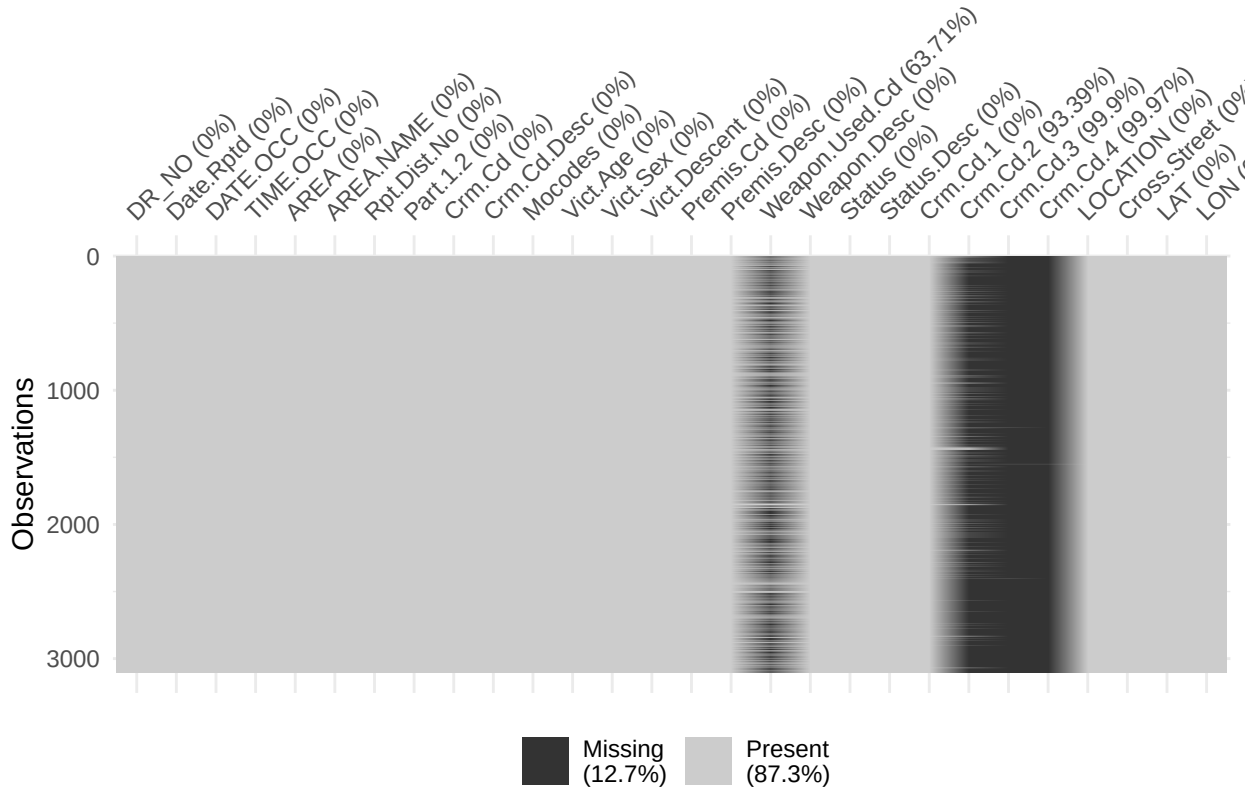
```
##  Max.   :956.0   Max.   :999.0    Max.   :999.0    Max.   :999.0
##  NA's   :6       NA's   :562263   NA's   :607033   NA's   :608548
##    LOCATION         Cross.Street            LAT             LON
##  Length:608592      Length:608592      Min.   : 0.00   Min.   :-118.7
##  Class :character   Class :character   1st Qu.:34.01   1st Qu.:-118.4
##  Mode  :character   Mode  :character   Median :34.06   Median :-118.3
##                                        Mean   :33.95   Mean   :-117.9
##                                        3rd Qu.:34.16   3rd Qu.:-118.3
##                                        Max.   :34.33   Max.   :   0.0
##
```

We are using summary() to summarise the statistical features of our sample in order to better understand our data. I notice many necessary numerical factor variables. So let's explain a few of the facts.

```
sum(duplicated(data.frame_3k))
```

```
## [1] 0
```

```
vis_miss(data.frame_3k)
```



We don't have any duplicate data, as can be shown. We can see that the Weapons used column contains NA values. CRM 2, 3, and 4. We are initially renaming our columns to make them simpler to understand and use.

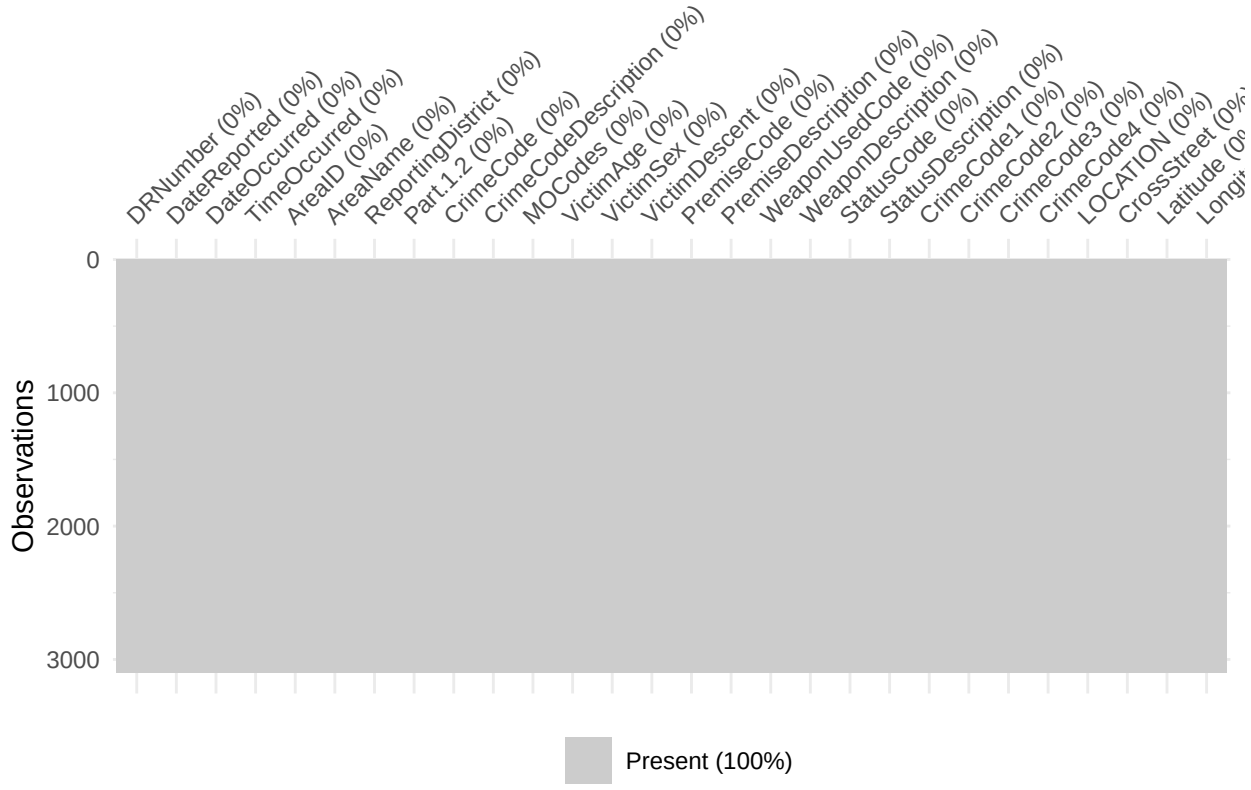
```
data.frame = data.frame %>%
  rename(DRNumber = `DR_NO`) %>%
  rename(DateReported = `Date.Rptd`) %>%
  rename(DateOccurred = `DATE.OCC`) %>%
  rename(TimeOccurred = `TIME.OCC`) %>%
  rename(AreaID = `AREA`) %>%
  rename(AreaName = `AREA.NAME`) %>%
  rename(ReportingDistrict = `Rpt.Dist.No`) %>%
  rename(CrimeCode = `Crm.Cd`) %>%
  rename(CrimeCodeDescription = `Crm.Cd.Desc`) %>%
  rename(MOCodes = `Mocodes`) %>%
  rename(VictimAge = `Vict.Age`) %>%
  rename(VictimSex = `Vict.Sex`) %>%
  rename(VictimDescent = `Vict.Descent`) %>%
  rename(PremiseCode = `Premis.Cd`) %>%
  rename(PremiseDescription = `Premis.Desc`) %>%
  rename(WeaponUsedCode = `Weapon.Used.Cd`) %>%
  rename(WeaponDescription = `Weapon.Desc`) %>%
  rename(StatusCode = `Status`) %>%
  rename(StatusDescription = `Status.Desc`) %>%
  rename(CrimeCode1 = `Crm.Cd.1`) %>%
  rename(CrimeCode2 = `Crm.Cd.2`) %>%
  rename(CrimeCode3 = `Crm.Cd.3`) %>%
  rename(CrimeCode4 = `Crm.Cd.4`) %>%
  rename(CrossStreet = `Cross.Street`) %>%
  rename(Latitude = `LAT`) %>%
  rename(Longitutde = `LON`)
```

Data Preprocessing

We're currently cleaning up our data. The initial stage in the visualisation of our data is data cleaning. We are replacing all values for a numerical property that are not applicable as part of our data cleaning procedure with the minimal value.

```
cleandata <- data.frame %>%
  mutate(
    WeaponUsedCode = replace_na(
      WeaponUsedCode,min(WeaponUsedCode, na.rm = TRUE)
      )
    ) %>%
  mutate(
    PremiseCode = replace_na(
      PremiseCode,min(PremiseCode,na.rm = TRUE)
    )
  ) %>%
  mutate(
    CrimeCode1 = replace_na(
      CrimeCode1,min(CrimeCode1,na.rm = TRUE)
    )
  ) %>%
  mutate(
    CrimeCode2 = replace_na(
```

```
      CrimeCode2,min(CrimeCode2,na.rm = TRUE)
    )
  ) %>%
  mutate(
    CrimeCode3 = replace_na(
      CrimeCode3,min(CrimeCode3,na.rm = TRUE)
    )
  ) %>%
  mutate(
    CrimeCode4 = replace_na(
      CrimeCode4,min(CrimeCode4,na.rm = TRUE)
    )
  )
cleandata_3k <-head(data.frame(cleandata),3100)
vis_miss(cleandata_3k)
```



From the plot, it is clear that all of the values for the weapon used code, CRM codes 2, 3, and 4, which are not applicable or not currently available, are handled. Additionally, each criminal report is a distinct entry in the data frame, as can be seen.

analyze the data using statistical measures such as mean, standard deviation etc

There are 13 catgeorial qualities and a total of 15 numerical attributes.

```
summary(cleandata)
```

```
##     DRNumber          DateReported       DateOccurred        TimeOccurred
##  Min.   :      817   Length:608592      Length:608592      Min.   :   1
##  1st Qu.:201710442   Class :character   Class :character   1st Qu.: 900
##  Median :211210646   Mode  :character   Mode  :character   Median :1415
##  Mean   :211363110                                         Mean   :1337
##  3rd Qu.:220607947                                         3rd Qu.:1900
##  Max.   :229921795                                         Max.   :2359
##      AreaID        AreaName         ReportingDistrict    Part.1.2
##  Min.   : 1.00   Length:608592      Min.   : 101      Min.   :1.000
##  1st Qu.: 6.00   Class :character   1st Qu.: 622      1st Qu.:1.000
##  Median :11.00   Mode  :character   Median :1142      Median :1.000
##  Mean   :10.72                      Mean   :1118      Mean   :1.416
##  3rd Qu.:16.00                      3rd Qu.:1617      3rd Qu.:2.000
##  Max.   :21.00                      Max.   :2199      Max.   :2.000
##    CrimeCode     CrimeCodeDescription   MOCodes            VictimAge
##  Min.   :110.0   Length:608592        Length:608592      Min.   : -1.00
##  1st Qu.:330.0   Class :character     Class :character   1st Qu.: 12.00
##  Median :442.0   Mode  :character     Mode  :character   Median : 31.00
##  Mean   :502.4                                          Mean   : 29.97
##  3rd Qu.:626.0                                          3rd Qu.: 45.00
##  Max.   :956.0                                          Max.   :120.00
##   VictimSex         VictimDescent       PremiseCode    PremiseDescription
##  Length:608592      Length:608592      Min.   :101.0   Length:608592
##  Class :character   Class :character   1st Qu.:101.0   Class :character
##  Mode  :character   Mode  :character   Median :203.0   Mode  :character
##                                        Mean   :302.4
##                                        3rd Qu.:501.0
##                                        Max.   :971.0
##  WeaponUsedCode  WeaponDescription   StatusCode        StatusDescription
##  Min.   :101.0   Length:608592      Length:608592      Length:608592
##  1st Qu.:101.0   Class :character   Class :character   Class :character
##  Median :101.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :193.3
##  3rd Qu.:400.0
##  Max.   :516.0
##    CrimeCode1      CrimeCode2      CrimeCode3      CrimeCode4
##  Min.   :110.0   Min.   :210.0   Min.   :434.0   Min.   :821
##  1st Qu.:330.0   1st Qu.:210.0   1st Qu.:434.0   1st Qu.:821
##  Median :442.0   Median :210.0   Median :434.0   Median :821
##  Mean   :502.2   Mean   :266.8   Mean   :435.4   Mean   :821
##  3rd Qu.:626.0   3rd Qu.:210.0   3rd Qu.:434.0   3rd Qu.:821
##  Max.   :956.0   Max.   :999.0   Max.   :999.0   Max.   :999
##    LOCATION         CrossStreet           Latitude       Longitutde
##  Length:608592      Length:608592      Min.   : 0.00   Min.   :-118.7
##  Class :character   Class :character   1st Qu.:34.01   1st Qu.:-118.4
##  Mode  :character   Mode  :character   Median :34.06   Median :-118.3
##                                        Mean   :33.95   Mean   :-117.9
##                                        3rd Qu.:34.16   3rd Qu.:-118.3
##                                        Max.   :34.33   Max.   :   0.0
```

We can see from the summary feedback that there are no NA values and that we can obtain statistical data

for every numerical attribute.

```
## Extracting month from time stamp of date crime occurred
Occureddate <- as.POSIXct(cleandata$DateOccurred, format = "%m/%d/%Y %H:%M:%S")
cleandata$MonthOfCrime<- format(Occureddate, format = "%m")
cleandata$DayofCrime<- format(Occureddate, format = "%d")
cleandata$YearofCrime<- format(Occureddate, format = "%Y")

ReportedDate <- as.POSIXct(cleandata$DateReported, format = "%m/%d/%Y %H:%M:%S")
cleandata$RMonthOfCrime<- format(ReportedDate, format = "%m")
cleandata$RDayofCrime<- format(ReportedDate, format = "%d")
cleandata$RYearofCrime<- format(ReportedDate, format = "%Y")
```

The statistical measures of the characteristics show that the dataset contains a significant amount of NAs, which are essentially incomplete or missing values. Other methods may have been used to add some actual data to the blank fields. However, because there are a lot of missing values, we used the R built-in function mutate to replace all of the numerical columns' "NA" values with the data frame's minimum value. The timestamp value column in the DateOccurred and DateReported columns has the format "%m/%d/%Y%H:%M:%S," yet it appears in the raw file as a char column. We utilise the R built-in function as.POSIXCT, which is used to access as per format request, to extract the date, month, and year from the columns.

Univariate analysis

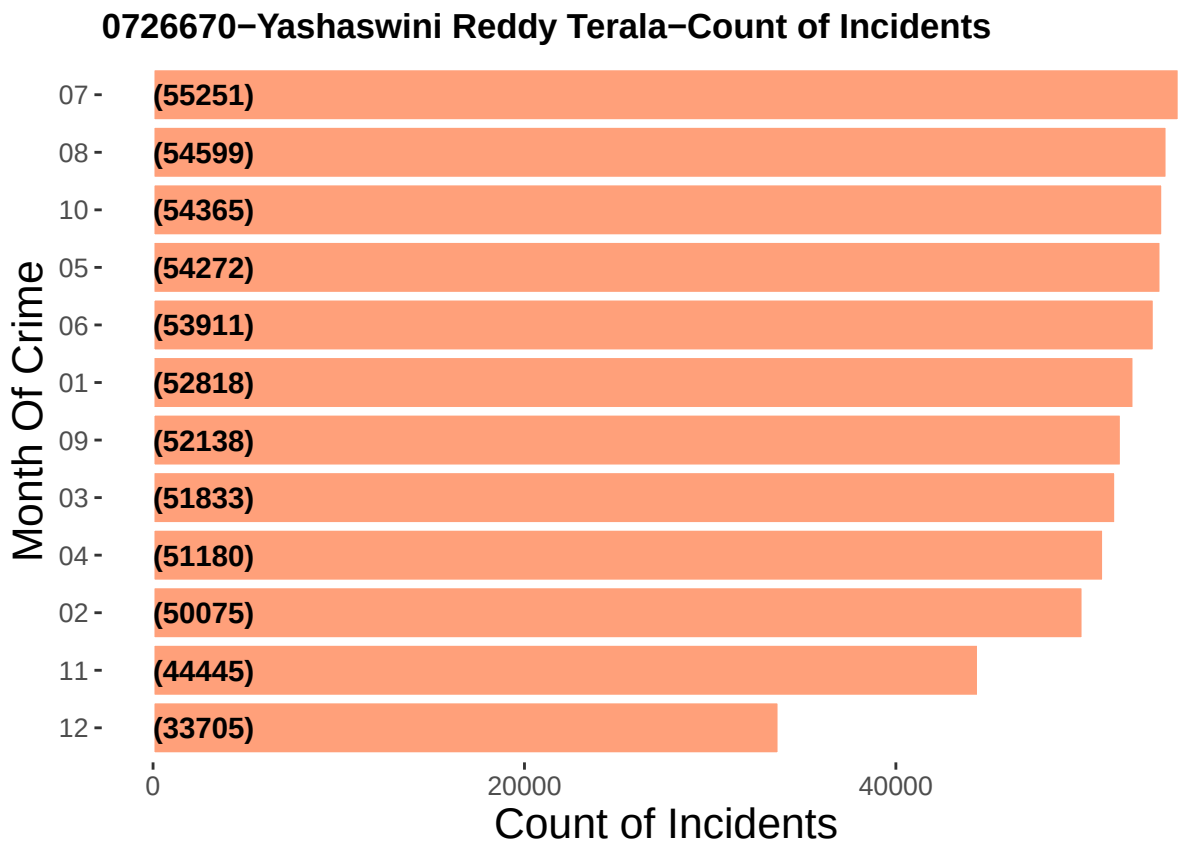
We begin each analysis with a univariate analysis, which is a quantitative data exploration. These studies help us by describing the single variables we are interested in, which helps us decide on the precise bivariate and multivariate analyses we should perform.

The simplest type of data analysis is called a univariate analysis. Uni stands for one, hence there is just one variable in the data. Each variable must be analysed separately for univariate data. A question, or more specifically, a research question, is the reason why data is collected. Frequency calculations, measures of central tendency, and measures of dispersion are the three primary categories of univariate studies.

Graphs can be used to describe univariate data. When comparing distinct groupings of data or different types of data, a bar graph is particularly helpful. Monitoring alterations over time is useful. A pie chart provides a summary of the data set that has been divided into more manageable chunks, which is reflected in each slice of the pie. The slices of the pie show the relative size of that group or category, while the entire pie symbolises 100%. The frequency distribution, as its name implies, depicts how frequently an event occurs in the data. Histograms show the same categorical variables against the category of data, much like bar charts do. The number of components in each category is indicated by the height of the bars. The number of data points in a range is indicated by the bin.A frequency polygon is used to compare data sets or represents the cumulative frequency distribution, much like the histogram.

```
cleandata %>%
  group_by(MonthOfCrime) %>%
  summarise(CountIncidents = n()) %>%
  mutate(MonthOfCrime = reorder(MonthOfCrime,CountIncidents)) %>%

ggplot(aes(x = MonthOfCrime,y = CountIncidents)) +
  geom_bar(stat='identity',colour="white", fill =fillColor) +
  geom_text(aes(x = MonthOfCrime, y = 1, label = paste0("(",CountIncidents,")",sep="")),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = 'Month Of Crime', y = 'Count of Incidents',
```

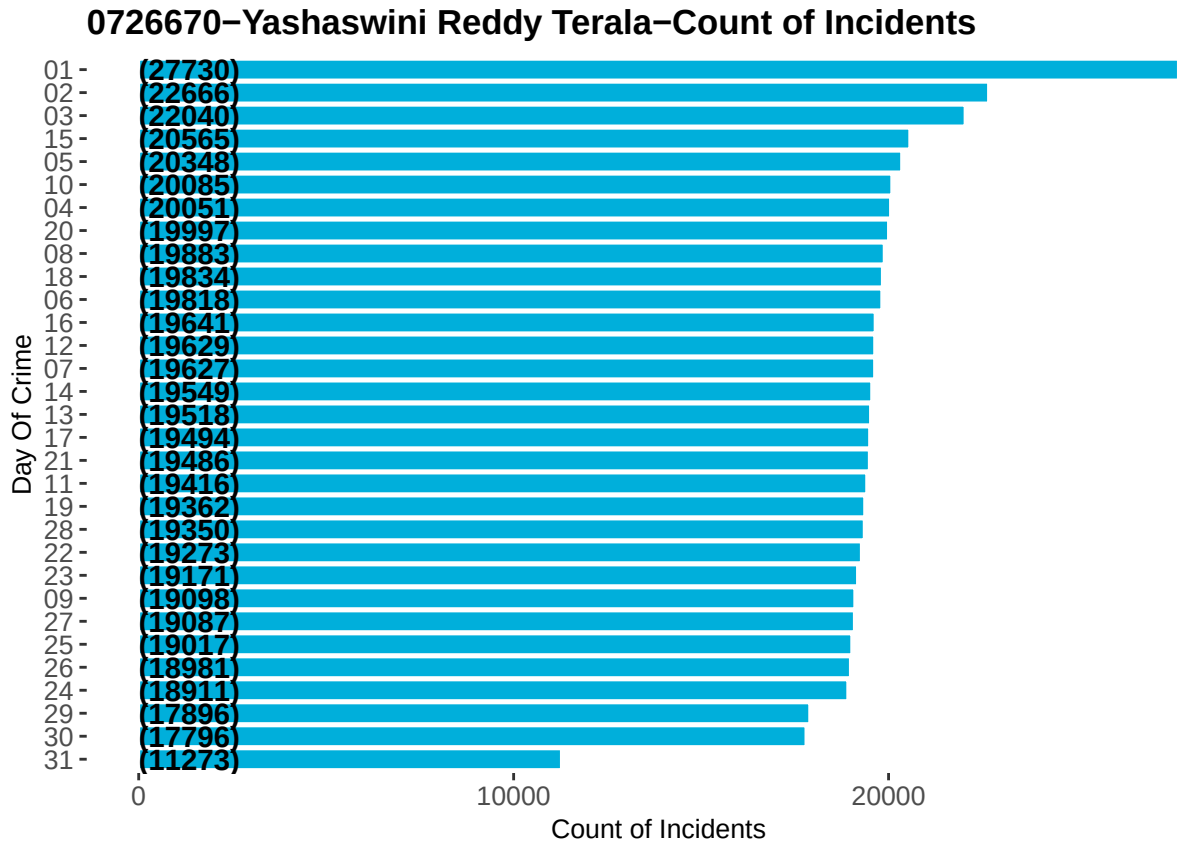
```
      title = '0726670-Yashaswini Reddy Terala-Count of Incidents') +
  coord_flip() +
  theme(
    plot.title = element_text( face = "bold"),
    axis.title.x = element_text(size = 16),
    axis.title.y = element_text(size = 16),
    axis.text.x = element_text(size = 10, angle = 0),
    axis.text.y = element_text(size = 10),
    legend.position = "none",
    panel.background = element_rect(fill='transparent'),
    plot.background = element_rect(fill='transparent', color=NA),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    legend.background = element_rect(fill='transparent'),
    legend.box.background = element_rect(fill='transparent')
    )
```



We observe that the month of July witnessed the highest number of crimes.

```
cleandata %>%
  group_by(DayofCrime) %>%
  summarise(CountIncidents = n()) %>%
  mutate(DayofCrime = reorder(DayofCrime,CountIncidents)) %>%

  ggplot(aes(x = DayofCrime,y = CountIncidents)) +
  geom_bar(stat='identity',colour="white", fill ="#00AFDB") +
  geom_text(aes(x = DayofCrime, y = 1, label = paste0("(",CountIncidents,")",sep="")),
```

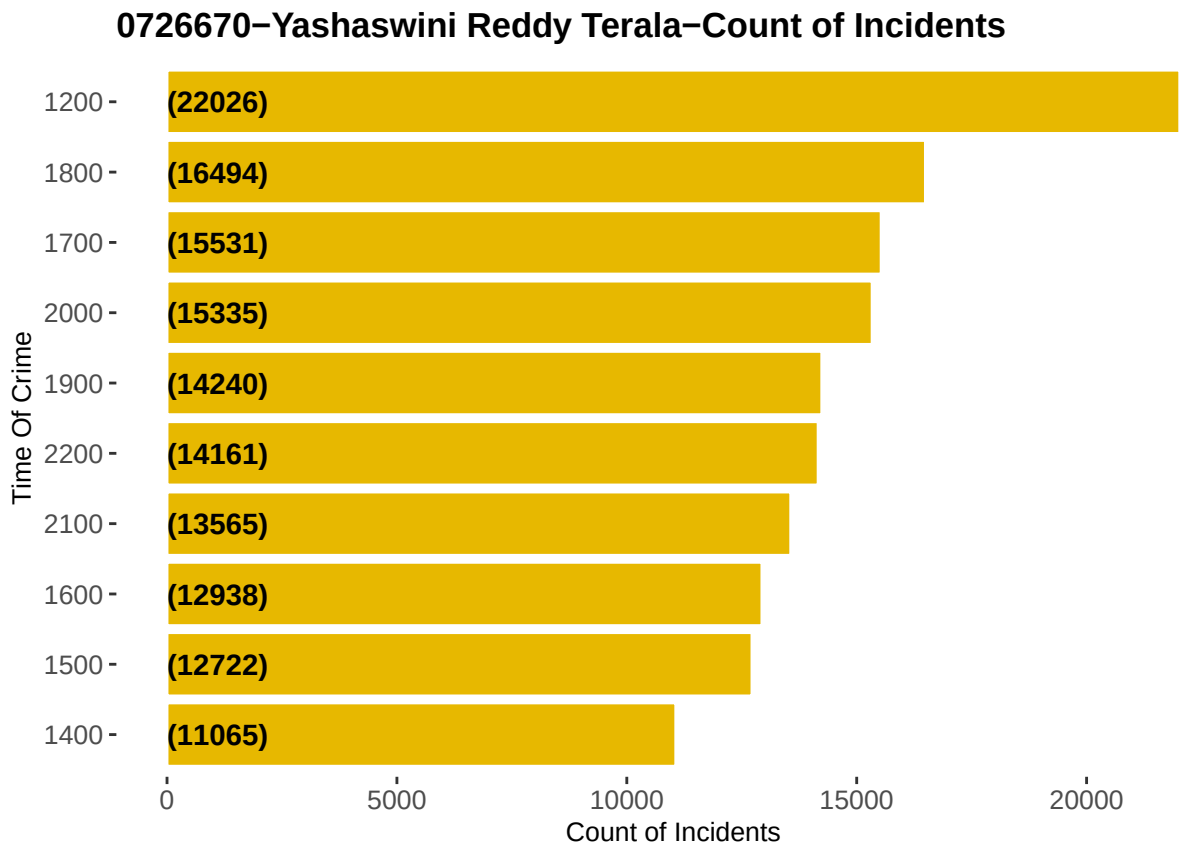
```
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = 'Day Of Crime', y = 'Count of Incidents',
       title = '0726670-Yashaswini Reddy Terala-Count of Incidents') +
  coord_flip() +
  theme(
    plot.title = element_text( face = "bold"),
    axis.title.x = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    axis.text.x = element_text(size = 10, angle = 0),
    axis.text.y = element_text(size = 10),
    legend.position = "none",
    panel.background = element_rect(fill='transparent'),
    plot.background = element_rect(fill='transparent', color=NA),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    legend.background = element_rect(fill='transparent'),
    legend.box.background = element_rect(fill='transparent')
    )
```



The above graph depicts that 1st of month witnesses more crimes.

```
cleandata %>%
  group_by(TimeOccurred) %>%
  summarise(CountIncidents = n()) %>%
  arrange(desc(CountIncidents)) %>%
  mutate(TimeOccurred = as.integer(TimeOccurred)) %>%
```

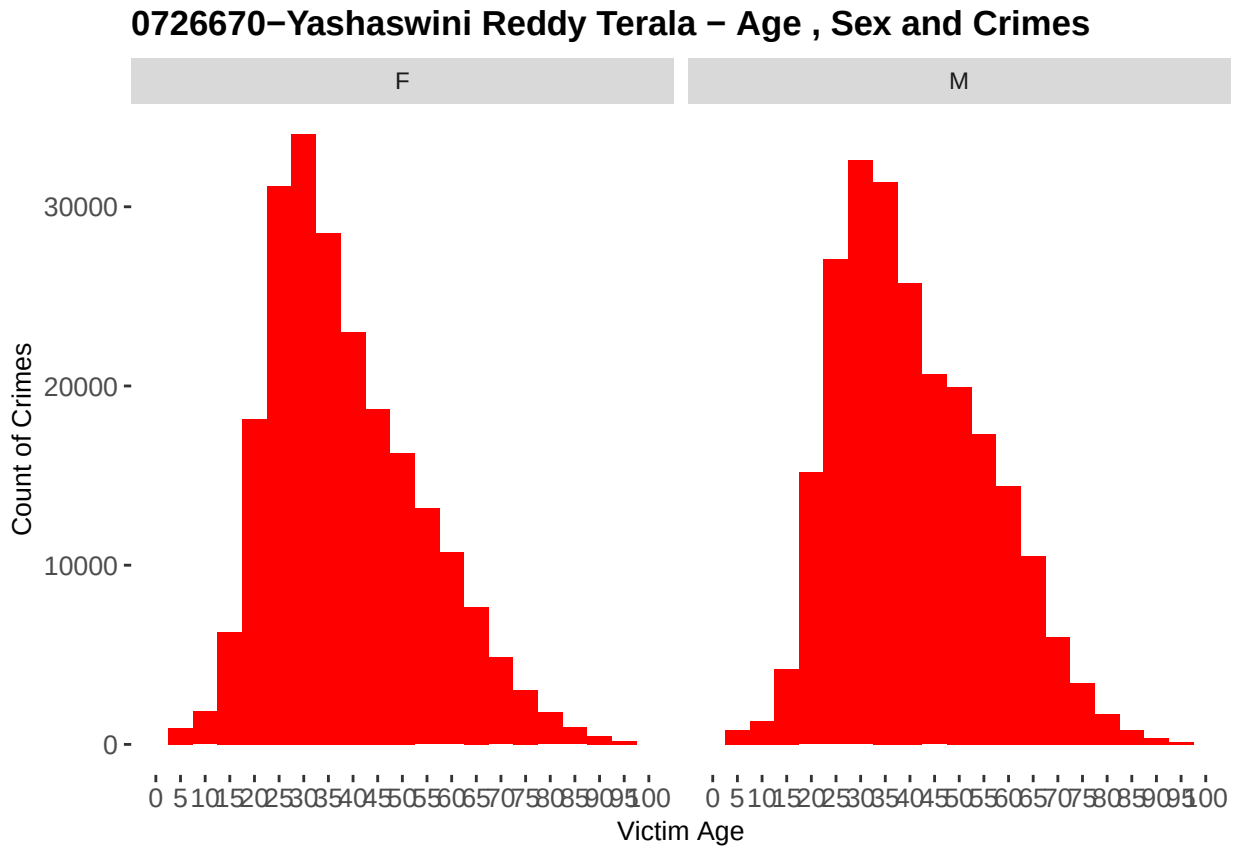
```
mutate(TimeOccurred = reorder(TimeOccurred,CountIncidents)) %>%
head(10) %>%

ggplot(aes(x = TimeOccurred,y = CountIncidents)) +
geom_bar(stat='identity',colour="white", fill ="#E7B800") +
geom_text(aes(x = TimeOccurred, y = 1, label = paste0("(",CountIncidents,")",sep="")),
          hjust=0, vjust=.5, size = 4, colour = 'black',
          fontface = 'bold') +
labs(x = 'Time Of Crime', y = 'Count of Incidents',
     title = '0726670-Yashaswini Reddy Terala-Count of Incidents') +
coord_flip() +
theme(
  plot.title = element_text( face = "bold"),
  axis.title.x = element_text(size = 10),
  axis.title.y = element_text(size = 10),
  axis.text.x = element_text(size = 10, angle = 0),
  axis.text.y = element_text(size = 10),
  legend.position = "none",
  panel.background = element_rect(fill='transparent'),
  plot.background = element_rect(fill='transparent', color=NA),
  panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
  legend.background = element_rect(fill='transparent'),
  legend.box.background = element_rect(fill='transparent')
  )
```



Most crimes happen at 1200 hours. Why would someone conduct crimes more frequently during the day

than at night is really strange.

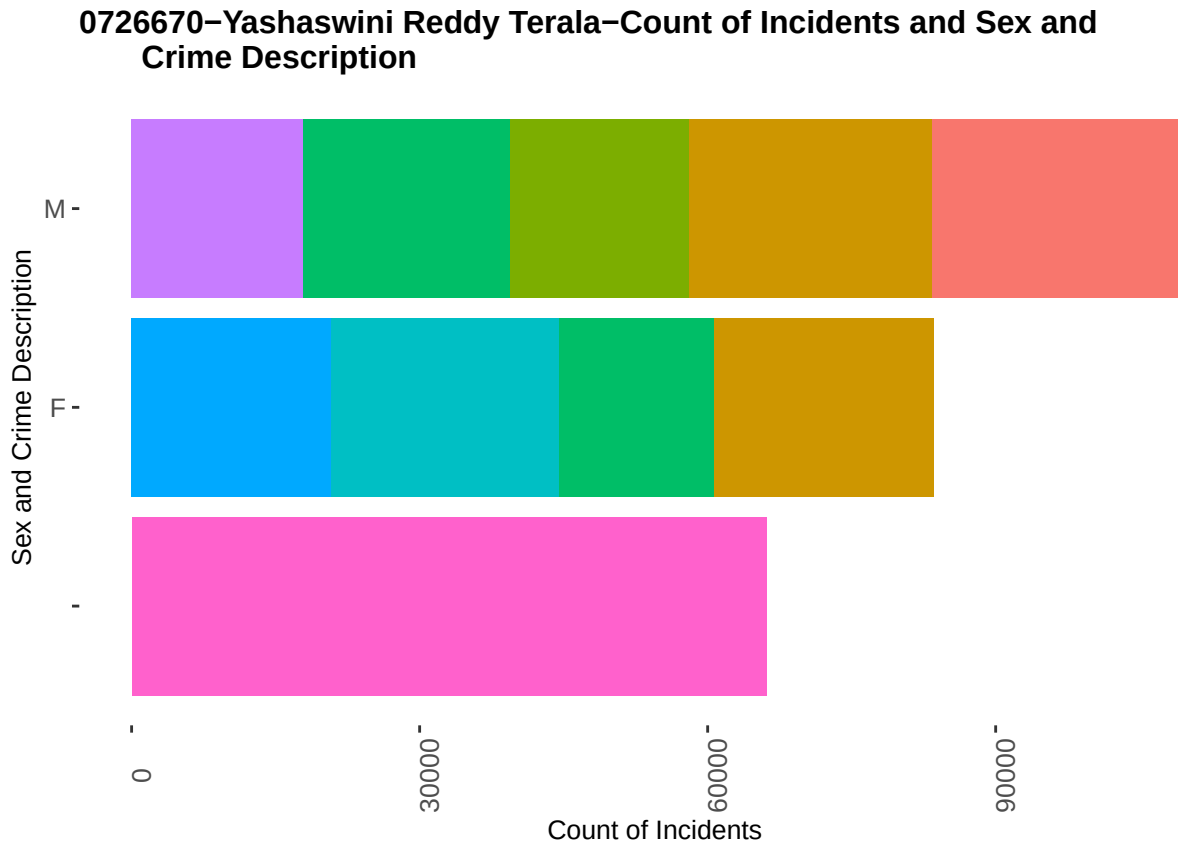
```
breaks = seq(0,100,5)

ListSex = c("M","F")

cleandata %>%
  filter( VictimSex %in% ListSex ) %>%
  ggplot(aes(VictimAge)) +
  geom_histogram(binwidth = 5,fill = c("red")) +
  facet_wrap(~ VictimSex) +
  scale_x_continuous(limits = c(0, 100),breaks=breaks ) +
  labs(x = 'Victim Age', y = 'Count of Crimes',
       title = '0726670-Yashaswini Reddy Terala - Age , Sex and Crimes') +
  theme(
    plot.title = element_text( face = "bold"),
    axis.title.x = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    axis.text.x = element_text(size = 10, angle = 0),
    axis.text.y = element_text(size = 10),
    legend.position = "none",
    panel.background = element_rect(fill='transparent'),
    plot.background = element_rect(fill='transparent', color=NA),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    legend.background = element_rect(fill='transparent'),
    legend.box.background = element_rect(fill='transparent')
    )
```



We look at the victims of crime's ages and genders. Most victims are between 25 and 30 years old. In the age range of 25 to 30, it is obvious that women are victims more often than men.

```
cleandata %>%
  filter(!is.na(VictimSex)) %>%
  group_by(VictimSex,CrimeCodeDescription) %>%
  tally() %>%
  ungroup() %>%
  mutate(VictimSex = reorder(VictimSex,n)) %>%
  arrange(desc(n)) %>%
  head(10) %>%

  ggplot(aes(x = VictimSex,y = n, fill =CrimeCodeDescription)) +
  geom_bar(stat='identity') +
  labs(x = 'Sex and Crime Description', y = 'Count of Incidents',
       title = '0726670-Yashaswini Reddy Terala-Count of Incidents and Sex and
       Crime Description') +
  coord_flip() +
  theme(
    plot.title = element_text( face = "bold", size = 12),
    axis.title.x = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    axis.text.x = element_text(size = 10, angle = 90),
    axis.text.y = element_text(size = 10),
    legend.position = "toptop",
    panel.background = element_rect(fill='transparent'),
```

```
    plot.background = element_rect(fill='transparent', color=NA),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    legend.background = element_rect(fill='transparent'),
    legend.box.background = element_rect(fill='transparent')
    )
```



We note that while we do not have the category of Intimate Partner - Simple Assault for males, we have for females.

Bivariate Analysis

In a bivariate analysis, there are two variables, and the analysis focuses on the relationship between the two variables as well as the cause of the two variables. For instance, the Super Bowl champions' point totals from 1960 through 2010.

When measuring and quantifying results, multivariate analytical approaches take into account the significant variables that may have an impact on this relationship. To explore the relationship between variables, one can employ a variety of multivariate analytical approaches. The most common method is multiple regression analysis, which explains how the usual value of the dependent variable varies when any one of the independent factors is altered while the other independent variables are maintained constant. Factor analysis, route analysis, and MANOVA are further methods.

These bivariate analysis types are described. The degree to which one variable has an impact on another is displayed using scatterplots. Regression analysis is used to examine the relationships between the data. The relationship between the variables is examined through correlation coefficients. "0" denotes that there is no relationship between the variables, while "1" indicates a relationship, either positive or negative.

Multivariate

Three or more variables are present in multivariate data. For instance, a web developer can use multivariate variables to measure the correlation between the variables when he or she wishes to look at the click and conversion rates of four different web sites among men and women. Multivariate analysis is used to examine a patient's varied responses to a medicine in a pharmaceutical experiment.

Any business that wants to be successful in the current global economy must concentrate on acquiring and interpreting market research surveys using the statistical techniques of bivariate and multivariate analysis in order to distinguish itself from competitors.

The primary advantage of multivariate analysis is that it produces results that are more accurate since more independent factors that influence the dependent variables are taken into account. The conclusions make greater sense and can be used in real-world situations.

Only one variable is analysed at a time in univariate statistics. Two variables are compared in bivariate statistics. Multiple variables are compared using multivariate statistics.

3D-Scatterplot

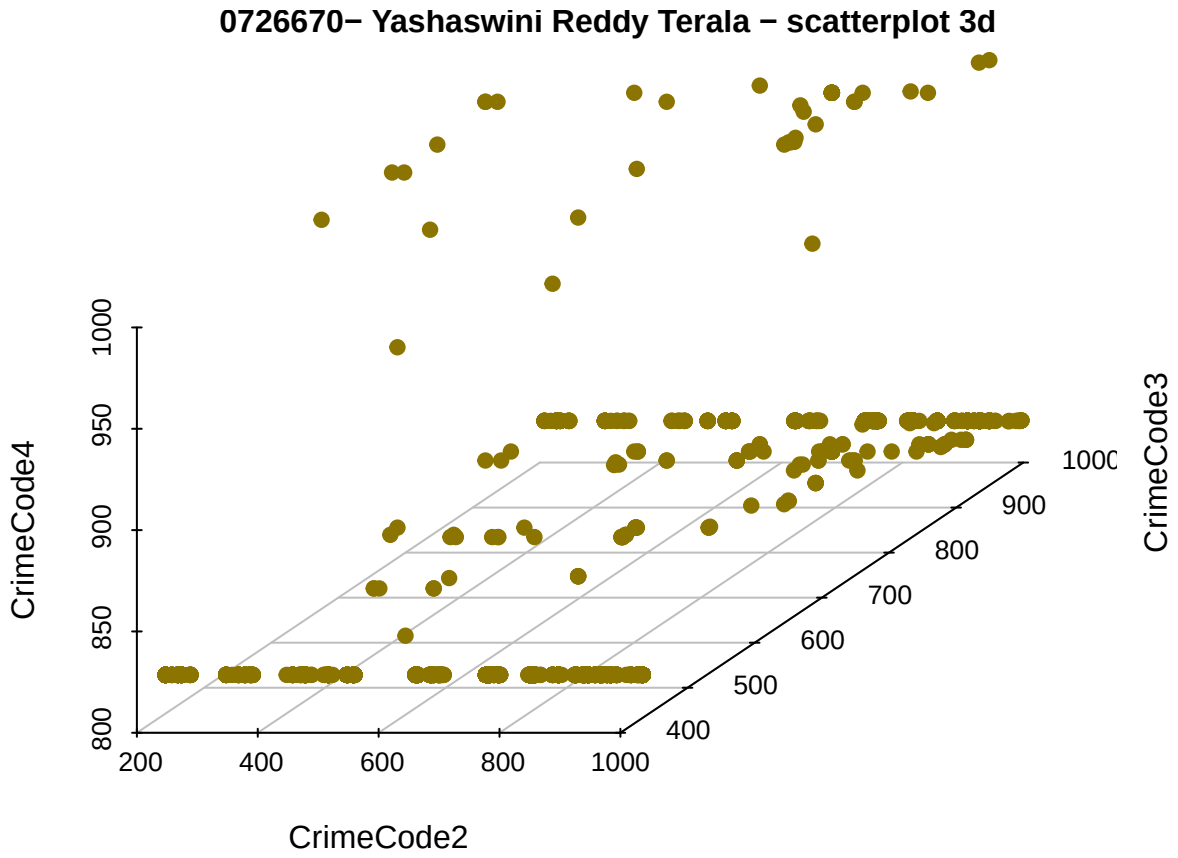
In an effort to visualise the relationship between three variables, 3D scatter plots are used to plot data points on three axes. The values of the columns selected on the X, Y, and Z axes determine the position of the marker that represents each row in the data table.

The size or colour of the markers can be assigned to a fourth variable, giving the map still more dimension.

It is known as a correlation when two variables are related to one another.The markers in a 3D scatter plot are near to forming a straight line in any direction, indicating a high correlation between the associated variables. The connection is negligible or nil if the markers are evenly spaced throughout the 3D scatter plot. Although a correlation can appear to exist, it could not necessarily be the case. The variations between the variables might be explained by their relationship to a fourth variable, or an apparent correlation could just be the result of coincidence.

Using the navigation tools in the visualization's top right corner, you may alter the 3D scatter plot's appearance by zooming in and out and rotating it.

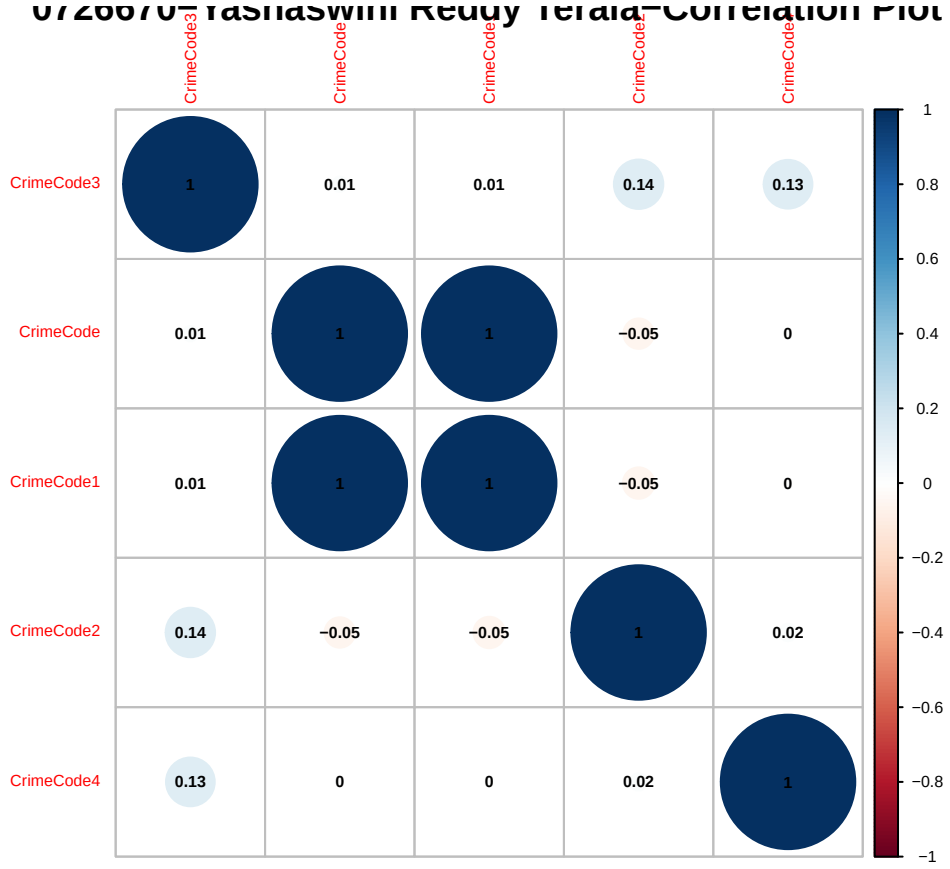
```
dt1<- cleandata[,c(22:24)]
scatterplot3d(dt1, pch = 19, color = "gold4",
   grid = TRUE, box = FALSE,
   mar = c(3, 3, 0.5, 3),
   main = "0726670- Yashaswini Reddy Terala - scatterplot 3d",
   cex.main = 1,
   )
```



Crime codes 2, 3, and 4 are plotted against one another for a variety of reported offences in the 3D scatter plot shown above (colored by product). In an effort to demonstrate the connections between the different criminal codes and identify trends, I chose this layout. Positive correlation is indicated by data points that have increased, and negative correlation is shown by data points that have increased while other data points have decreased.

Correlation Plot:

```
df2 <- cleandata[,c(9,21:24)]
# load library
library(corrplot)
corrplot(cor(df2),method = "circle", addCoef.col= 1, order = "AOE",
         title="0726670-Yashaswini Reddy Terala-Correlation Plot",
         number.cex=0.5,  tl.cex = 0.5,cl.cex = 0.5
         )
```



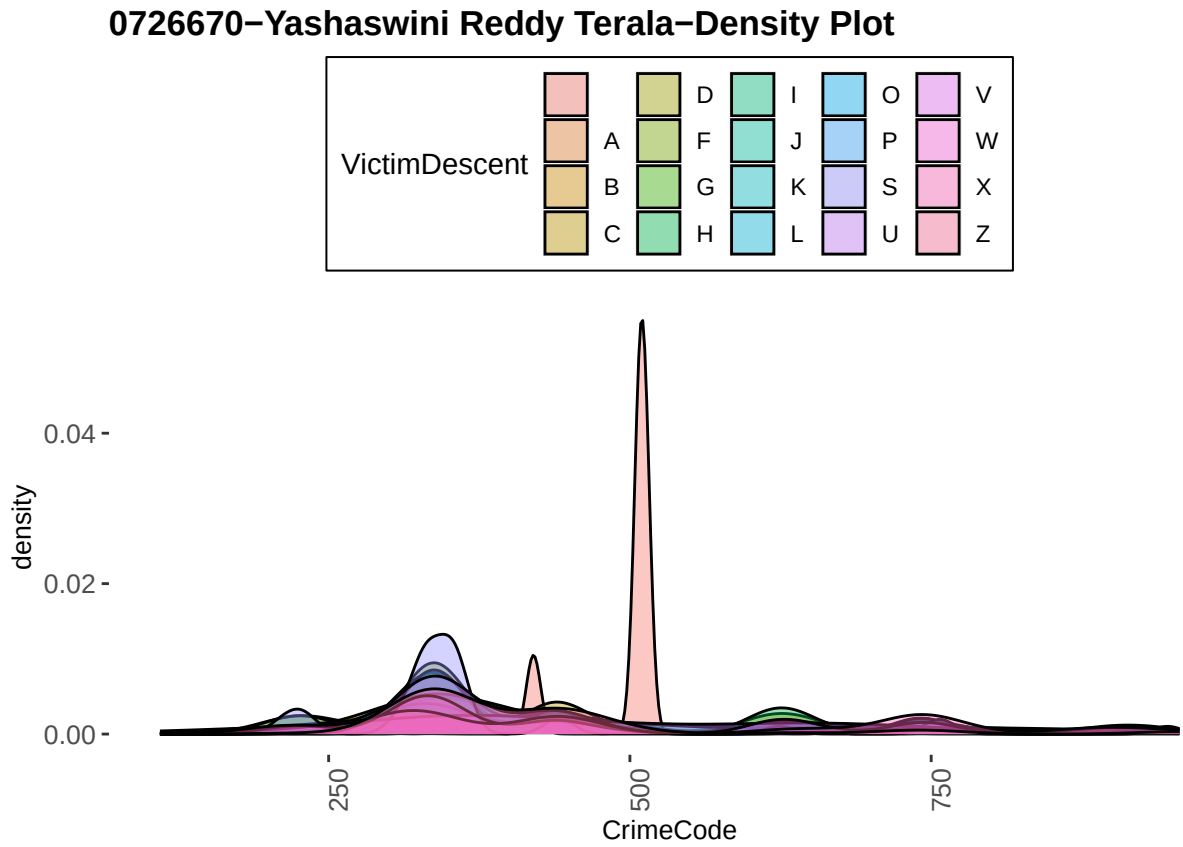
Blue stands for a positive correlation and red for a negative correlation in a circle. The link is stronger when the dot is larger. The diagonal of the matrix, which displays the correlation between each characteristic and itself, is perfectly positive correlated, allowing us to observe that the matrix is symmetrical. Additionally, we can observe that there is no negative correlation and that all of the traits are favourably associated.

Density plot

A density chart with multiple groups shown is referred to as a multi density chart. Comparing their distribution is possible. The problem with this type of chart is that it may rapidly get congested; when groupings overlap, the figure becomes difficult to read. Transparency can be used as a simple workaround. Although it won't totally fix the problem, it is frequently preferable to take into account the examples that are provided further on in this document.

```
# With transparency
p1 <- ggplot(data=cleandata, aes(x=CrimeCode, group=VictimDescent,
                                 fill=VictimDescent)) +
    geom_density(adjust=1.5, alpha=.4) +
  ggtitle("0726670-Yashaswini Reddy Terala-Density Plot")+
  theme(
    plot.title = element_text( face = "bold"),
    axis.title.x = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    axis.text.x = element_text(size = 10, angle = 90),
    axis.text.y = element_text(size = 10),
    legend.position = "top",
    panel.background = element_rect(fill='transparent'),
```

```
    plot.background = element_rect(fill='transparent', color=NA),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    legend.background = element_rect(fill='transparent'),
    legend.box.background = element_rect(fill='transparent')
    )

p1
```



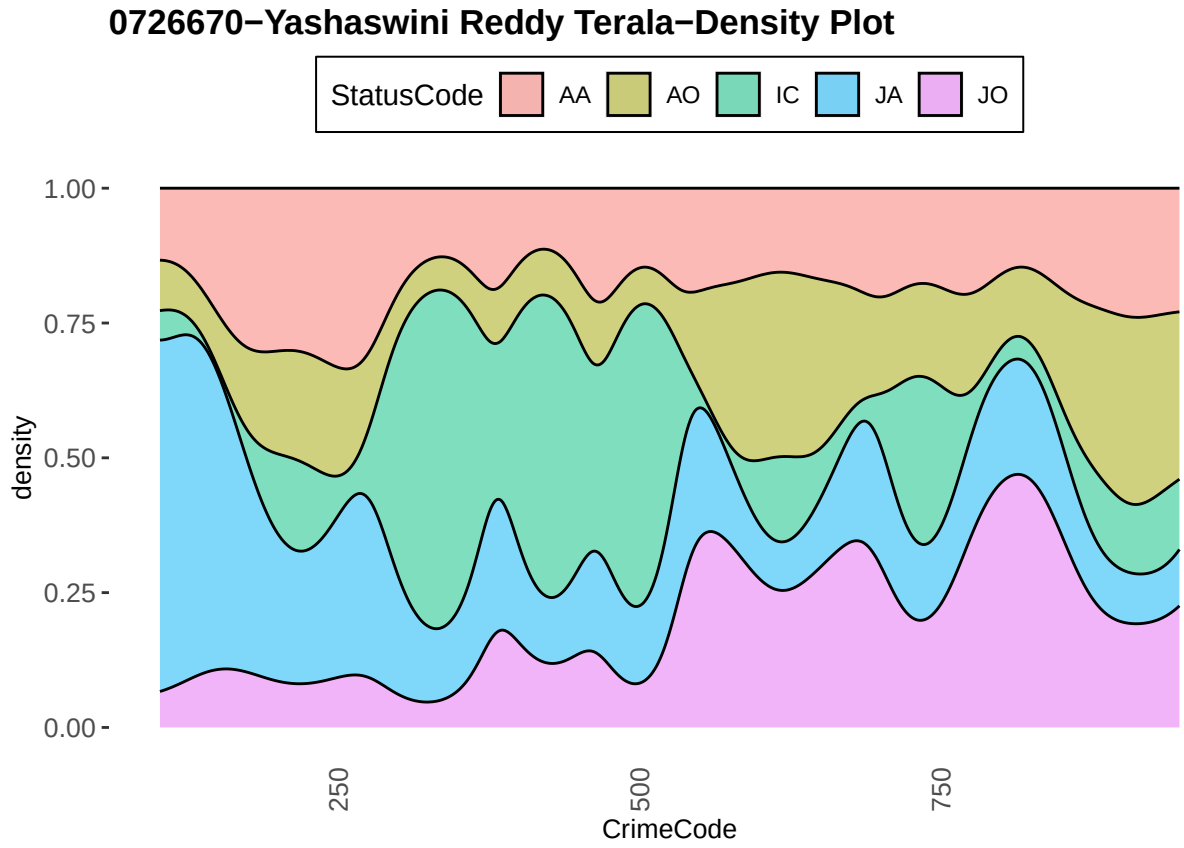
Group distributions are very distinct, making it simple to identify them even when they are on the same chart.

Stacked density chart

Stacking the groups is an alternative solution. This makes it possible to see which group is most prevalent for a particular value, but it makes it challenging to comprehend the distribution of a group that is not at the chart's bottom. Stacking is the process of dividing a chart over multiple categorical variables that together make up the entire. The categoric variable's items are each represented by a darkened region. These spaces are piled one on top of the other. Area charts, barplots, and streamcharts are the three main, closely related types of graphics where it can be found.

```
# Stacked density plot:
sdensity <- ggplot(data=cleandata,
            aes(x=CrimeCode, group=StatusCode, fill=StatusCode)) +
  geom_density(adjust=1.5, position="fill", alpha= 0.5) +
  ggtitle("0726670-Yashaswini Reddy Terala-Density Plot")+
```

```
  theme(
      legend.position="top",
      panel.spacing = unit(0.1, "lines"),
      axis.ticks.x=element_blank(),
      plot.title = element_text( face = "bold"),
      axis.title.x = element_text(size = 10),
      axis.title.y = element_text(size = 10),
      axis.text.x = element_text(size = 10, angle = 90),
      axis.text.y = element_text(size = 10),
      panel.background = element_rect(fill='transparent'),
      plot.background = element_rect(fill='transparent', color=NA),
      panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
      legend.background = element_rect(fill='transparent'),
      legend.box.background = element_rect(fill='transparent')
    )
sdensity
```

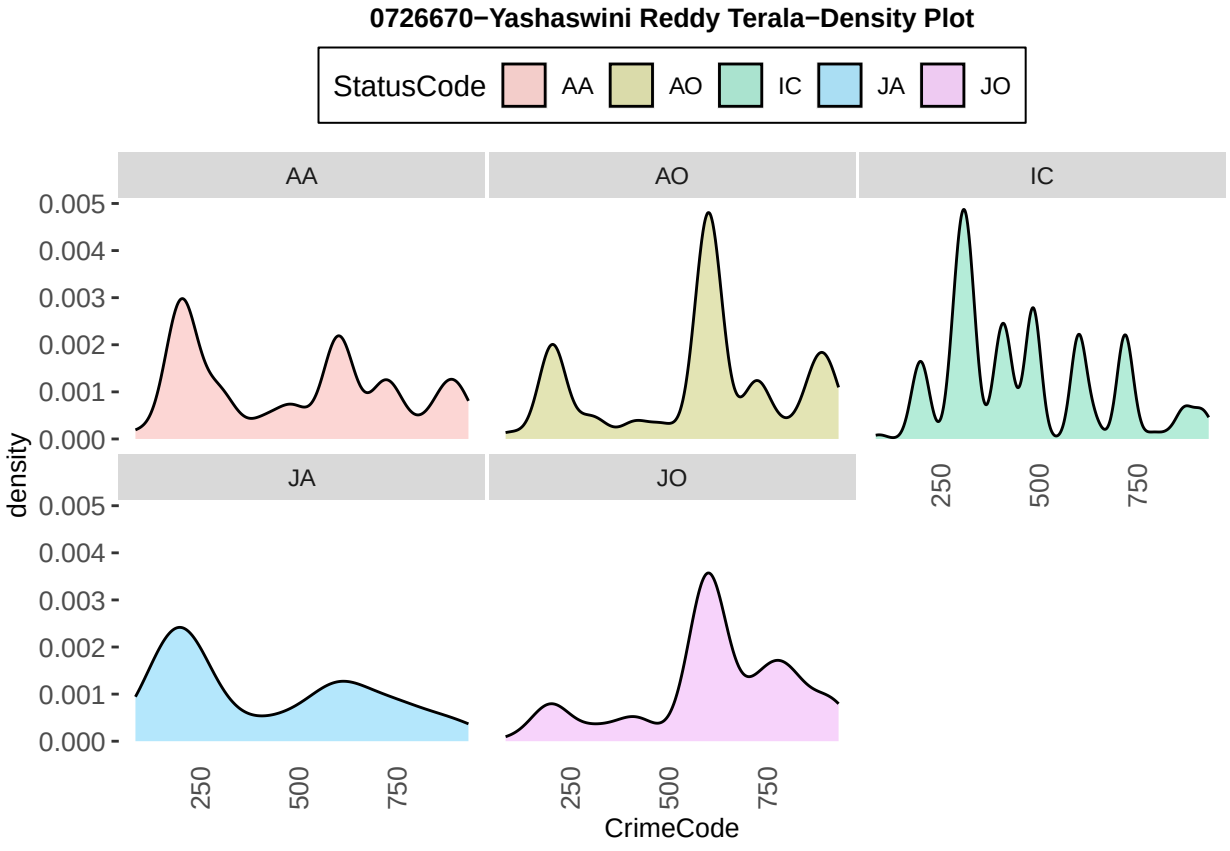


Our diagram is a stacked density chart for the steam area. I used steamgraph because they perform well when the data exhibits a distinct pattern. The figure won't be particularly illuminating if the proportion of each group stays roughly constant across the time period because it will be challenging to read slight differences.

Density plot with facet wrap

In my experience, using small multiples is frequently the best choice. Each group's distribution becomes simple to read, and groupings can still be compared if their X axis bounds are the same.

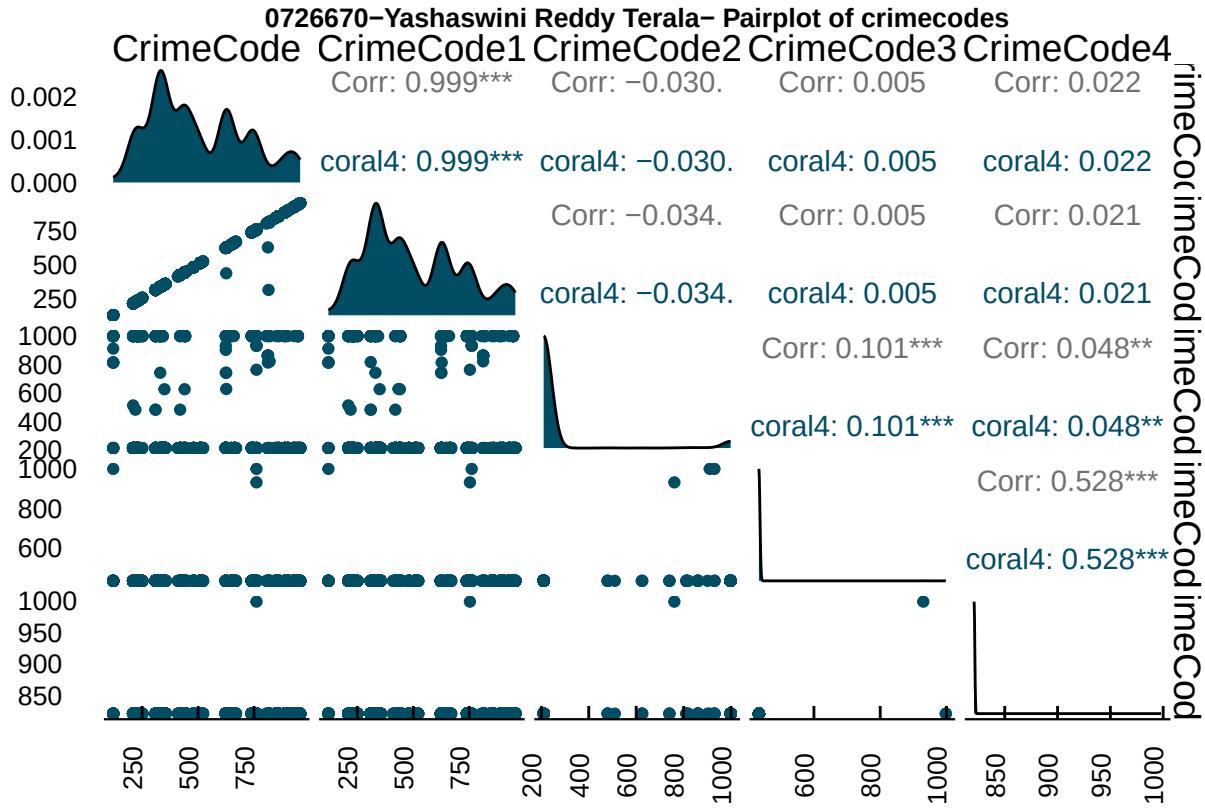
```
# Using Small multiple
ggplot(data=cleandata, aes(x=CrimeCode, group=StatusCode, fill=StatusCode)) +
  geom_density(adjust=1.5, alpha=0.3) +
  ggtitle("0726670-Yashaswini Reddy Terala-Density Plot")+
  facet_wrap(~StatusCode) +
  theme(
      legend.position="top",
      panel.spacing = unit(0.1, "lines"),
      axis.ticks.x=element_blank(),
      plot.title = element_text( hjust = 0.5,size = 10,face = "bold"),
      axis.title.x = element_text(size = 10),
      axis.title.y = element_text(size = 10),
      axis.text.x = element_text(size = 10, angle = 90),
      axis.text.y = element_text(size = 10),
      panel.background = element_rect(fill='transparent'),
      plot.background = element_rect(fill='transparent', color=NA),
      panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
      legend.background = element_rect(fill='transparent'),
      legend.box.background = element_rect(fill='transparent')
    )
```



Pairplot

Using a pair plot, we visualise the relationship between crime codes. A pair plot is produced by the R programming language's ggpairs function. The pairs function from basic R is equivalent to the ggplot2 equivalent in the GGally, which is called ggpairs. Continuous and categorical variables can both be transferred using a data frame. By default, for relationships between their categorical and continuous counterparts, the relationship between the continuous variables is shown in the upper panel, their scatter plots are shown in the lower panel, their density plots are shown on the diagonal, and their histograms and box plots are shown on the sides.

```
#pair plot

# draw the graph
p <- cleandata_3k %>%
  ggpairs(columns = c(9,21:24),
          lower = list(continuous = "points"),
          diag = list(continuous = "densityDiag"),
          aes(color = "coral4"))+
  theme_economist()+
  ggtitle("0726670-Yashaswini Reddy Terala- Pairplot of crimecodes")+
  theme(plot.title = element_text(hjust = 0.5,size = 10,face = "bold"),
    axis.title.x = element_text(size = 8),
    axis.title.y = element_text(size = 8),
    axis.text.x = element_text(size = 10, angle = 90),
    axis.text.y = element_text(size = 10),
    legend.position = "top",
    panel.background = element_rect(fill='transparent'),
    plot.background = element_rect(fill='transparent', color=NA),
    panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    legend.background = element_rect(fill='transparent'),
    legend.box.background = element_rect(fill='transparent')
    )

#change the color
for(i in 1:p$nrow) {
for(j in 1:p$ncol){
  p[i,j] <- p[i,j] +
    scale_fill_economist() +
    scale_color_economist()
  }
}

# show the graph
p
```



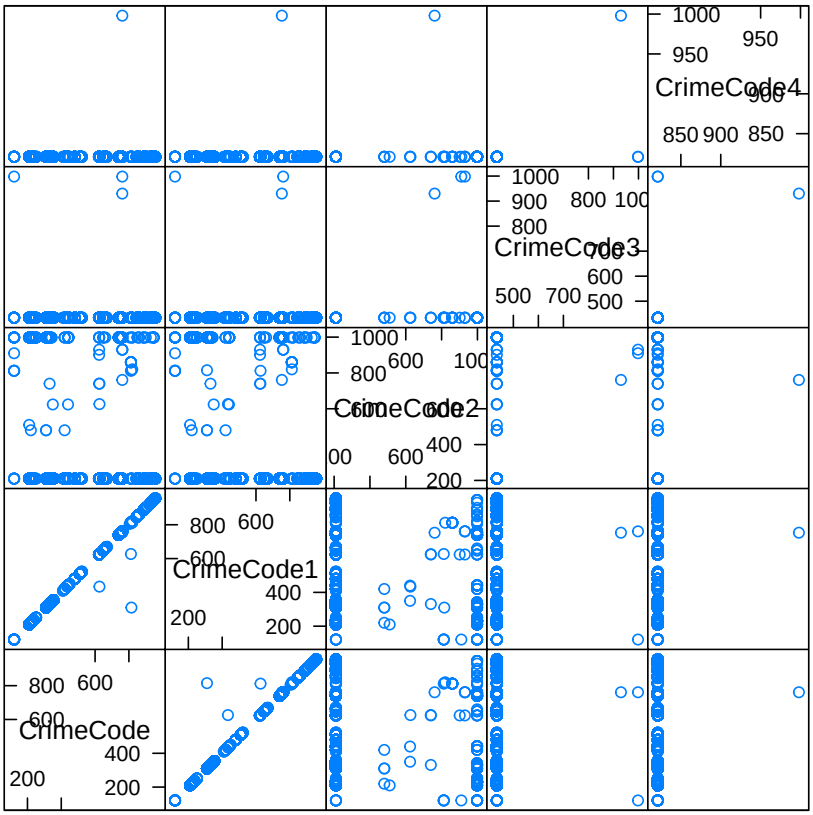
On the graph shown here, we can see correlation coefficients between two of these five traits. The crime code and crime code 1 have the highest association coefficient, indicating that crime code 1 is most frequently reported as the final criminal code. Crime code 2 and crime code show the least association.

Scatterplot Matrix

Another approach to create a scatterplot matrix is by using splom() from the lattice library.

```
library(lattice)
splomdata <- cleandata_3k[,c(9,21:24)]
splom(splomdata,
      main =
        "0726670- Yashaswini Reddy Terala-Scatterplot Matrices using splom",
      xlab("Attributes"), ylab("Attributes"))
```

0726670− Yashaswini Reddy Terala−Scatterplot Matrices using splom



Scatter Plot Matrix

A scatterplot matrix is an attempt to add more dimensions to the conventional 2D scatterplot. It functions by plotting all feasible scatterplots derived from pairs of variables in the data set in a matrix to represent pairs of variables in conventional scatterplots.

It's a great tool for quickly comparing similar data sets (as they are each arranged next to each other in vertical and horizontal directions). There are disadvantages to this method as well, including the fact that it is difficult or impossible to label the individual axes of the smaller scatterplots (due to space restrictions and legibility requirements) and that there is no "global" view of the data. This method is easier to use for analysis in some cases than the parallel coordinates method.

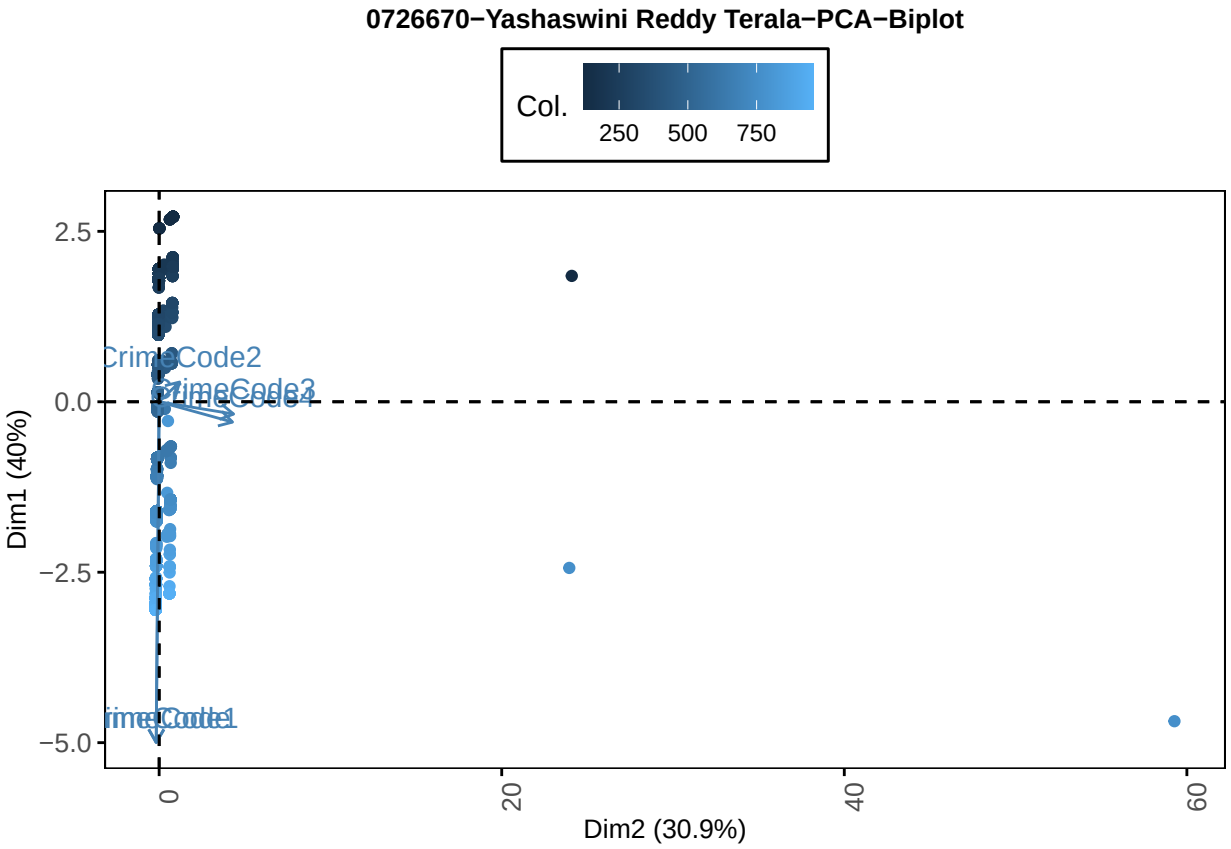
PCA

Principal component analysis (PCA), a multivariate data analysis technique, enables us to draw attention to and show the most important facts in a multivariate data collection.

Through the use of PCA, the data is split into a few additional dimensions (or axes), each of which is a linear combination of the initial variables. In a multivariate dataset, the most important information can be observed by making a scatter plot of the first two dimensions.

```
library("factoextra")
my_data <- cleandata_3k[,c(9,21:24)]# Remove the grouping variable
res.pca <- prcomp(my_data, scale = TRUE)
fviz_pca_biplot(res.pca, col.ind = my_data$CrimeCode,
                palette = "jco", geom = "point",
```

```
              title = "0726670-Yashaswini Reddy Terala-PCA-Biplot")+
coord_flip()+
theme(plot.title = element_text(hjust = 0.5,size = 10,face = "bold"),
  axis.title.x = element_text(size = 10),
  axis.title.y = element_text(size = 10),
  axis.text.x = element_text(size = 10, angle = 90),
  axis.text.y = element_text(size = 10),
  legend.position = "top",
  panel.background = element_rect(fill='transparent'),
  plot.background = element_rect(fill='transparent', color=NA),
  panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
  legend.background = element_rect(fill='transparent'),
  legend.box.background = element_rect(fill='transparent')
  )
```



In the plot described above:

A total of 70.9% (40% + 30.9%) of the data set's total information was stored in dimensions (Dim.) 1 and 2. People with similar profiles are grouped together. Variables that have a positive correlation are shown side by side. The variables that are negatively related are located on the other side of the plots.

Autocorrelation

Serial correlation, another name for autocorrelation, impacts standard errors without impacting unbiased coefficients. It happens when there is a relationship between the time series' error terms.

Incorrect functional forms or omitted variables are the primary causes of autocorrelation. The Durbin-Watson test or a scatter plot with all residuals and a pattern-spotting approach can be used to find autocorrelation.

When performing ARIMA modelling, autocorrelation analysis is crucial since it aids in identifying the AR (Auto Correlation) and MA (Moving Average) components of the ARIMA model.

To verify autocorrelation, we can use the partial autocorrelation (PACF) plot and the autocorrelation function (ACF) plot. ACF plots are simply bar charts that display the correlation coefficients between a time series and its own lags, whereas PACF plots are plots that display the partial correlation coefficients between the series and its own lags.

Beginning with lag 0, which is the time series' correlation with itself, results in a correlation of 1 for ACF and PACF. Additionally, significant-high correlation might be observed in the immediate lags following 0, which may just be the result of the autocorrelation at lag 1 spreading.

The autocorrelation pattern can be explained more readily by adding AR terms than by adding MA terms if the PACF exhibits a quick cutoff while the ACF decays more slowly (i.e., contains substantial spikes at higher lags).

Consider include an MA term in the model if the differenced time series' ACF shows an abrupt cutoff and/or the lag-1 autocorrelation is negative, or if the series appears to be just a little bit "over differenced." The specified number of MA terms is the lag at which the ACF discontinues.

```
w <- table(cleandata$ReportingDistrict)
rep.dis <- as.data.frame(w)
length(unique(rep.dis$Var1))
```

```
## [1] 1191
```

```
head(rep.dis, 5)
```

```
##   Var1 Freq
## 1  101  570
## 2  105  180
## 3  109   16
## 4  111 2366
## 5  112  147
```

As we can see above, there were no crimes committed in these districts since the number of unique values did not match the number of features in the shapefile. Therefore, we maintain these columns when joining by using "all.x = TRUE". This implies that we maintain every row of data in the shapefile. With the second line of code, we override the default behaviour of these, which returns NA results.

```
districts <- readOGR(dsn = "LAPD_Reporting_Districts.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/yashaswini/Desktop/masters/sem3/data visualization/Project/Final/LAPD_Reporting_Distr
## with 1135 features
## It has 7 fields
## Integer64 fields read as strings:  OBJECTID REPDIST PREC
```

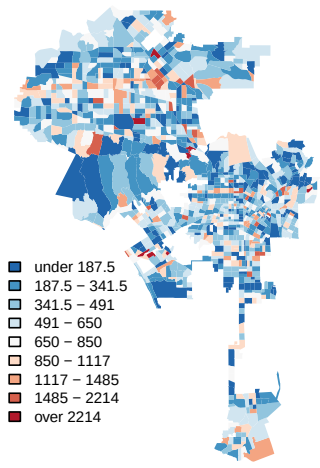
```
head(districts,5)
```

```
## class       : SpatialPolygonsDataFrame
## features    : 5
## extent      : -118.5471, -118.4285, 34.29712, 34.33731  (xmin, xmax, ymin, ymax)
## crs         : +proj=longlat +datum=WGS84 +no_defs
## variables   : 7
## names       : OBJECTID, REPDIST, PREC,      APREC,        BUREAU, BASICCAR,
## min values  :        1,    1705,   17, DEVONSHIRE, VALLEY BUREAU,    17A35, Bureau: VALLEY BUREAU\nDi
## max values  :        5,    1904,   19,    MISSION, VALLEY BUREAU,     19A7,    Bureau: VALLEY BUREAU
```

```
districts@data <- merge(districts@data, rep.dis, by.x = "REPDIST",
                        by.y = "Var1", all.x = TRUE)
districts$Freq[is.na(districts$Freq)] <- 0
length(districts$Freq)
```

```
## [1] 1135
```

```
var <- districts@data[,"Freq"]
breaks <- classIntervals(var, n = 9, style = "fisher")
my_colours <- rev(brewer.pal(9, "RdBu"))
plot(districts,
     main = "0726670-Yashaswini Reddy Terala-Spatial Autocorrelation",
     col = my_colours[findInterval(var, breaks$brks,
                                           all.inside = TRUE)],
       axes = FALSE, border = NA)
legend(x = -118.7, y = 34, legend = leglabs(breaks$brks), fill = my_colours,
       bty = "n", cex = 0.6)
```



A brief summary of the story reveals that the bulk of the districts have fewer numbers of reported offences. There are infrequent instances of large counts, particularly in northern regions; this may indicate a spatial autocorrelation that is closer to zero and so more representative of the data's spatial distribution's randomness.

Crime rates would be a better indicator, however there is no data on population by reported districts.

Spatial Autocorrelation

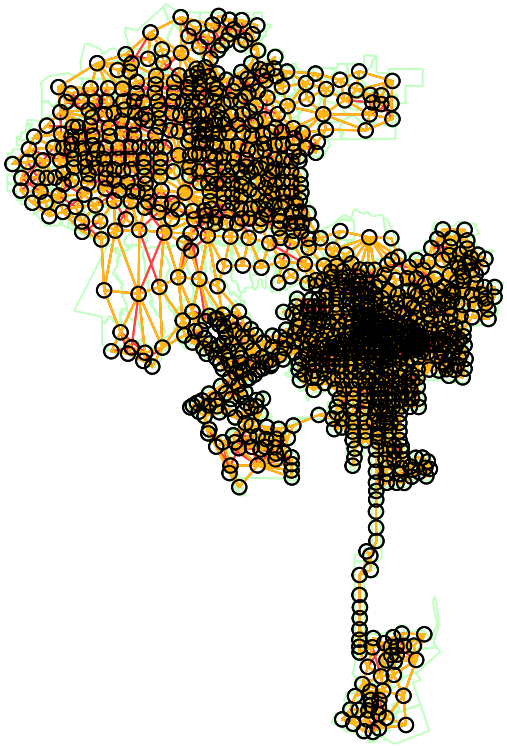
Let's attempt to quantify the association rather than speculating using the plot. A district's crime frequency is the particular variable that is being measured via spatial autocorrelation.

The majority of geographic phenomena should impose some sort of spatial autocorrelation, according to Tobler's first law of geography. People with comparable traits prefer to live in similar neighbourhoods for a variety of reasons, including property pricing, proximity to workplaces, and cultural considerations. This is frequently the case in data containing human components.

There are two ways to depict spatial autocorrelation: globally or locally. While local models allow us to investigate spatial clustering throughout the LA area, global models produce a single measure that represents the entire dataset.

The first step is to assign neighbours to every district. The two ways to do this are Rook and Queen. To see how the neighbourhood districts are distributed over space, we can plot the connections between them.

```
neighbours <- poly2nb(districts)
neighbours2 <- poly2nb(districts, queen = FALSE)
plot(districts, border = 'darkseagreen1',
     main = "0726670-Yashaswini Reddy Terala-Spatial Autocorrelation")
plot(neighbours, coordinates(districts), add=TRUE, col='brown1')
plot(neighbours2, coordinates(districts), add=TRUE, col='darkgoldenrod1')
```



We can visually understand the difference and see that the second method (Rook) returns significantly fewer links.

Global Spatial Autocorrelation

We need to convert our neighbors now that we have them.

We can then perform a Moran's test. This results in a correlation value between -1 and 1, and the graphic in the introduction shows the regional distribution of these values. Data are randomly distributed when a value of 0 is displayed, while a value of -1 denotes perfect negative spatial autocorrelation.

```
listw <- nb2listw(neighbours)
listw
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 1135
```

```
## Number of nonzero links: 7080
## Percentage nonzero weights: 0.5495934
## Average number of links: 6.237885
##
## Weights style: W
## Weights constants summary:
##      n      nn   S0       S1       S2
## W 1135 1288225 1135 392.1103 4642.774
```

```
moran.test(districts$Freq, listw)
```

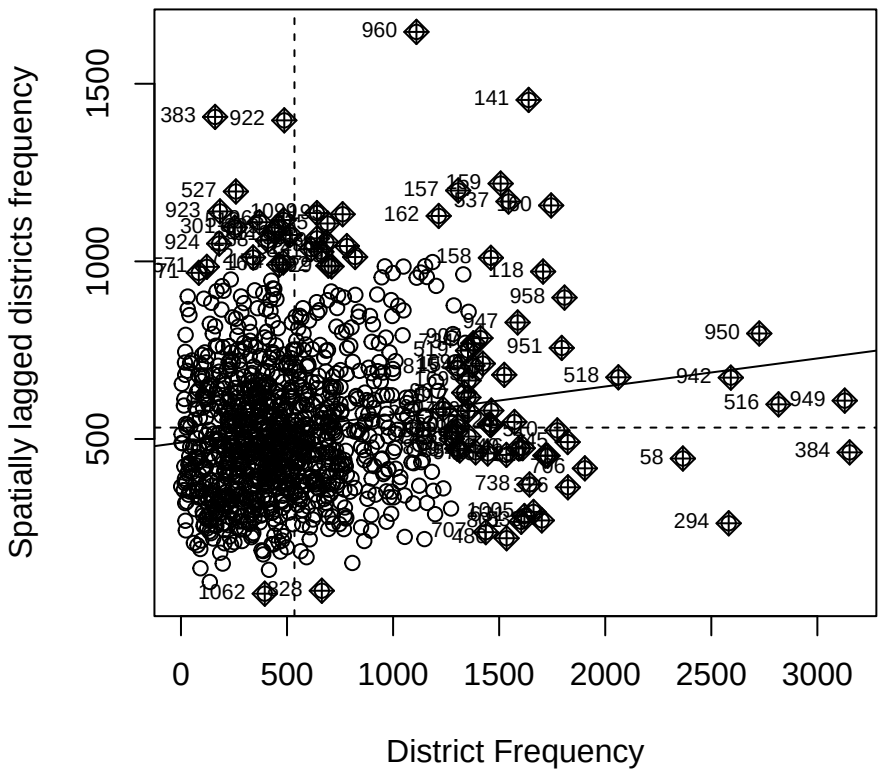
```
##
##  Moran I test under randomisation
##
## data:  districts$Freq
## weights: listw
##
## Moran I statistic standard deviate = 4.5292, p-value = 2.96e-06
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      0.0777030318     -0.0008818342      0.0003010447
```

The Global Moran I statistic of 0.077 indicates that each district has a considerably random distribution of crime frequency. Let's investigate this further, though, to determine if local spatial autocorrelation yields a comparable outcome.

Local Spatial Autocorrelation

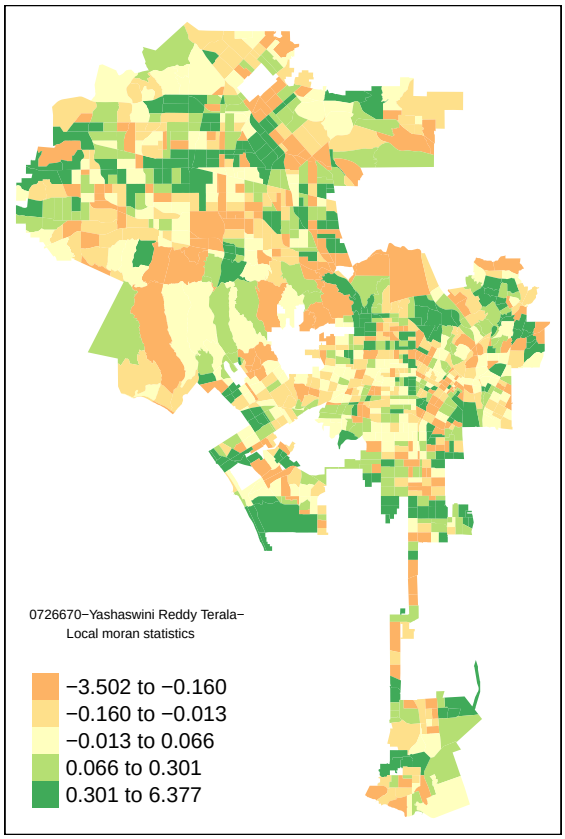
First, a Moran plot is made, which compares each value to its spatially lagged counterpart. It investigates how the data are related to their environs.

```
moran <- moran.plot(districts$Freq,
                    xlab = "District Frequency",
                    ylab = "Spatially lagged districts frequency",
                    title ="0726670-Yashaswini Reddy Terala-Moran plot",
                    listw = nb2listw(neighbours2, style = "W"))
```



It is possible to infer from the scatter plot that each district's crime rate is random. Let's attempt to map it.

```
local <- localmoran(x = districts$Freq, listw = nb2listw(neighbours2,
                                                         style = "W"))

moran.map <- cbind(districts, local)

tm_shape(moran.map ) +
  tm_fill(col = "Ii", style = "quantile",
          title = "0726670-Yashaswini Reddy Terala-
          Local moran statistics",
          legend.show = TRUE)
```

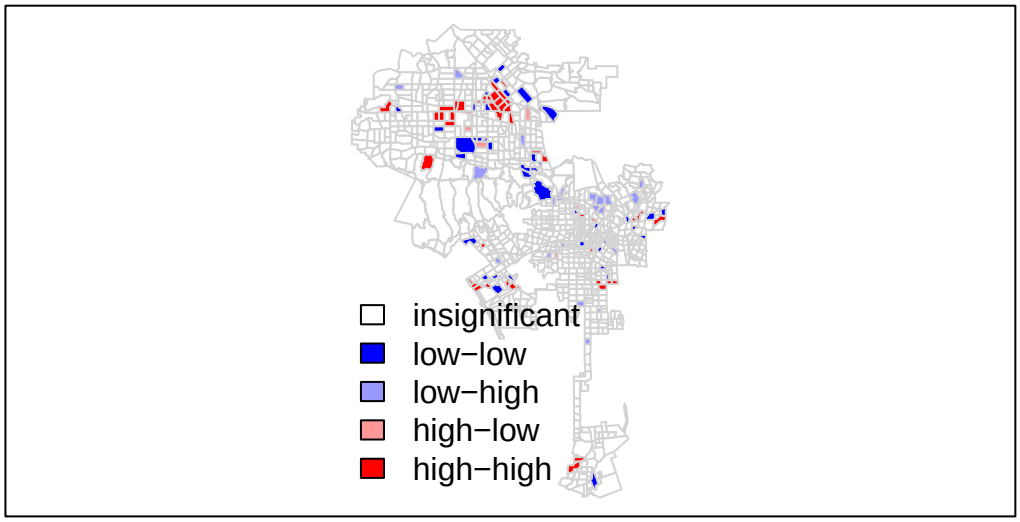


A high score signifies that the district is encircled by other districts with a comparable rating.

The graphic shows that, for the most part, there is little correlation between the number of crimes in one LA reporting district and those in neighbouring districts. However, we may spot a few highly comparable groups in the districts to the north. However, other than the fact that they are similar, it is impossible to determine whether these are clusters with high or low crime rates. Therefore, we run a LISA cluster map.

```
quadrant <- vector(mode="numeric",length=nrow(local))

# centers the variable of interest around its mean
m.crime <- districts$Freq - mean(districts$Freq)

# centers the local Moran's around the mean
m.local <- local[,1] - mean(local[,1])

# significance threshold
signif <- 0.1

# builds a data quadrant
quadrant[m.crime >0 & m.local>0] <- 4
quadrant[m.crime <0 & m.local<0] <- 1
quadrant[m.crime <0 & m.local>0] <- 2
quadrant[m.crime >0 & m.local<0] <- 3
quadrant[local[,5]>signif] <- 0

brks <- c(0,1,2,3,4)
colors <- c("white","blue",rgb(0,0,1,alpha=0.4),rgb(1,0,0,alpha=0.4),"red")
```

```
plot(
  districts,border="lightgray",
     col=colors[findInterval(quadrant,brks,all.inside=FALSE)],
  main = "0726670-Yashaswini Reddy Terala- Local Spatial Autocorrelation"
  )
box()
legend(x = -118.7, y = 34,legend=c("insignificant","low-low","low-high",
                                   "high-low","high-high"),
       fill=colors,bty="n")
```



Districts with high crime reports clustered around other high crime reporting districts were shown on the original Moran map. Surprisingly, two of the cluster variables—districts with low crime rates surrounded by other districts with similar low crime rates (dark blue) and districts with high crime rates surrounding by districts with low crime rates—are not present.

Justifications

Handling Spatial autocorrelation using Global Moran's I tool

We are aware that autocorrelation indicates how similar a given time series is to itself when lagged over a series of time periods. The link between a variable's present value and its previous values is measured by autocorrelation. There are often two approaches to dealing with autocorrelation, with the first approach being the most crucial: Enhance model fit. Make an effort to include structure in the model's data. Include an AR1 model if no additional predictors can be added.

We are interested in autocorrelation because it can be used to identify recurring patterns in a signal. It is aware that the Spatial Autocorrelation (Global Moran's I) tool simultaneously calculates spatial autocorrelation based on feature positions and value. It determines whether the pattern expressed is clustered, scattered, or random in the presence of a collection of features and an associated attribute. To assess the relevance of the Moran's I Index, the tool computes its value along with a z-score, p-value, and other metrics. P-values, which are constrained by the test statistic, are numerical approximations of the area under the curve given a known distribution.

Because the Spatial Autocorrelation (Global Moran's I) tool is an inferential statistic, the analysis's findings must always be understood in light of its null hypothesis. The null hypothesis for the Global Moran's I statistic asserts that the characteristic being studied is distributed randomly among the study area's features; or, to put it another way, the spatial processes producing the observed pattern of values are random chance. Consider being able to pick up the attribute values you are examining and toss them onto your features, letting each value land in any location. This procedure (picking up and dropping the values) is an illustration of a spatial random chance process.

Five statistics are returned by the spatial autocorrelation tool: the z-score, p-value, variance, expected index, and moran's I index. These values are passed as derived output values for potential usage in models or scripts and are written as messages at the bottom of the Geoprocessing window during tool operation.

In other words, because it works with our data, I'm concentrating on contiguity-based weights, where a spatial unit shares a border with another spatial unit. As a result, I determined the neighbours of the queen and rook and carried out both global and local spatial autocorrelation.

It follows that the use of Global Moran's I tool for spatial autocorrelation on our dataset is justified.

Visualizations chosen

I typically use line graphs or segment graphs to display the statistical measures. We must comprehend the distribution of data and produce insights for future decision-making. One stage is to analyse and exhibit the datasets using single-variable univariate approaches, such as line graphs, pie plots, bar graphs, and histograms. Since our primary concern is frequency, histograms, bar graphs, and line segments were used.

The datasets will then be analysed and presented using multivariate approaches, such as pair plot, parallel coordinates, trellis plot, 3D scatter plot, scatter plot matrix, Principal Component Analysis - biplot and Cluster Analysis, star plots, chernoff face, and other geometric projects. To visualise our multivariate data, we are here selecting pairplot, 3d Scatterplot, Scatterplot matrix, Correlation plot, Density Plot, PCA, and Stacked density plot.

A pairplot is utilised because it allows us to observe both the distribution of single variables and relationships between two variables, whereas a 3D scatterplot is used to depict relationships between 3 variables. I choose to use a scatterplot matrix because it makes the connections between various variables easier to see. I utilised a correlation plot because I wanted to know how strongly different variables related to one another. A normal density plot, a facet density plot, and a stacked density plot all have the same objective, which is to examine the distribution of several different variables. In order to identify trends, jumps, clusters, and outliers, I utilised PCA, which is essential and facilitates the representation of multivariate data tables as smaller set of variables (summary indices).

References:

1. https://data.lacity.org/Public-Safety/Crime-Data-from-2020-to-Present/2nrs-mtv8
2. https://spatialanalysis.github.io/workshop-notes/spatial-autocorrelation.html
3. Walter D. Koenig. 1999. Spatial autocorrelation of ecological phenomena. Trends in Ecology & Evolution. 14(1). 22-26. https://doi.org/10.1016/S0169-5347(98)01533-X.
4. https://www.kaggle.com/code/ambarish/eda-lacrimes-maps-timeseriesforecasts-xgboost/report#time-of-crime

5. https://www.kaggle.com/code/ghannay/spatial-autocorrelation-of-la-crime/notebook
6. https://towardsdatascience.com/data-cleaning-in-r-made-simple-1b77303b0b17
7. https://www.marsja.se/how-to-extract-year-from-date-in-r-with-examples/#:~:text=To%20get%20the%20year%20fro
8. https://www.kaggle.com/datasets/cityofLA/los-angeles-crime-arrest-data?select=crime-data-from-2010-to-present.csv
9. https://data.lacity.org/Public-Safety/Crime-Data-from-2020-to-Present/2nrs-mtv8
10. http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
11. https://www.data-to-viz.com/
12. https://www.interaction-design.org/literature/article/information-visualization-an-introduction-to-multivariate-analysis
13. https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57
14. https://rpubs.com/laubert/SACtutorial