DTSC 3010 Section 020 Final Project

Do Ngan

2024-10-12

My research question: **What are the key patterns in crime occurrences, and How can we predict crime types and identify whether crime reports are timely or delayed based on demographic factors, location, and times?**

Read data

```
final_crime_data <- read.csv("D:/R Project/new_crime_data.csv")
summary(final_crime_data)
```

```
##      DR_NO             Date.Rptd           DATE.OCC            TIME.OCC
##  Min.   :200100856   Length:9999        Length:9999        Min.   : 100
##  1st Qu.:210113301   Class :character   Class :character   1st Qu.:1000
##  Median :221008127   Mode  :character   Mode  :character   Median :1400
##  Mean   :219620601                                         Mean   :1379
##  3rd Qu.:230914520                                         3rd Qu.:1845
##  Max.   :242112445                                         Max.   :2359
##      AREA          AREA.NAME          Rpt.Dist.No      Part.1.2
##  Min.   : 1.00   Length:9999        Min.   : 101   Min.   :1.000
##  1st Qu.: 5.00   Class :character   1st Qu.: 557   1st Qu.:1.000
##  Median :10.00   Mode  :character   Median :1068   Median :2.000
##  Mean   :10.45                      Mean   :1092   Mean   :1.667
##  3rd Qu.:16.00                      3rd Qu.:1613   3rd Qu.:2.000
##  Max.   :21.00                      Max.   :2197   Max.   :2.000
##     Crm.Cd      Crm.Cd.Desc          Vict.Age        Vict.Sex
##  Min.   :330   Length:9999        Min.   : 2.00   Length:9999
##  1st Qu.:330   Class :character   1st Qu.:29.00   Class :character
##  Median :354   Mode  :character   Median :37.00   Mode  :character
##  Mean   :436                      Mean   :39.99
##  3rd Qu.:624                      3rd Qu.:50.00
##  Max.   :624                      Max.   :83.00
##  Vict.Descent        Premis.Cd      Premis.Desc           Status
##  Length:9999        Min.   :101.0   Length:9999        Length:9999
##  Class :character   1st Qu.:104.0   Class :character   Class :character
##  Mode  :character   Median :501.0   Mode  :character   Mode  :character
##                     Mean   :345.7
##                     3rd Qu.:502.0
##                     Max.   :958.0
##  Status.Desc          LOCATION              LAT             LON
##  Length:9999        Length:9999        Min.   : 0.00   Min.   :-118.7
```

```
## Class :character   Class :character    1st Qu.:34.02    1st Qu.:-118.4
## Mode  :character   Mode  :character    Median :34.06    Median :-118.3
##                                        Mean   :34.02    Mean   :-118.2
##                                        3rd Qu.:34.17    3rd Qu.:-118.3
##                                        Max.   :34.33    Max.   :   0.0
## Time.to.reports   Delayed_Report    Time_Slots_Happening Weekdays_of_DateOcc
## Min.   :   0.00   Min.   :0.0000    Length:9999          Length:9999
## 1st Qu.:   0.00   1st Qu.:0.0000    Class :character     Class :character
## Median :   1.00   Median :0.0000    Mode  :character     Mode  :character
## Mean   :  29.91   Mean   :0.3828
## 3rd Qu.:   5.00   3rd Qu.:1.0000
## Max.   :1744.00   Max.   :1.0000
```

```r
ncol(final_crime_data)
```
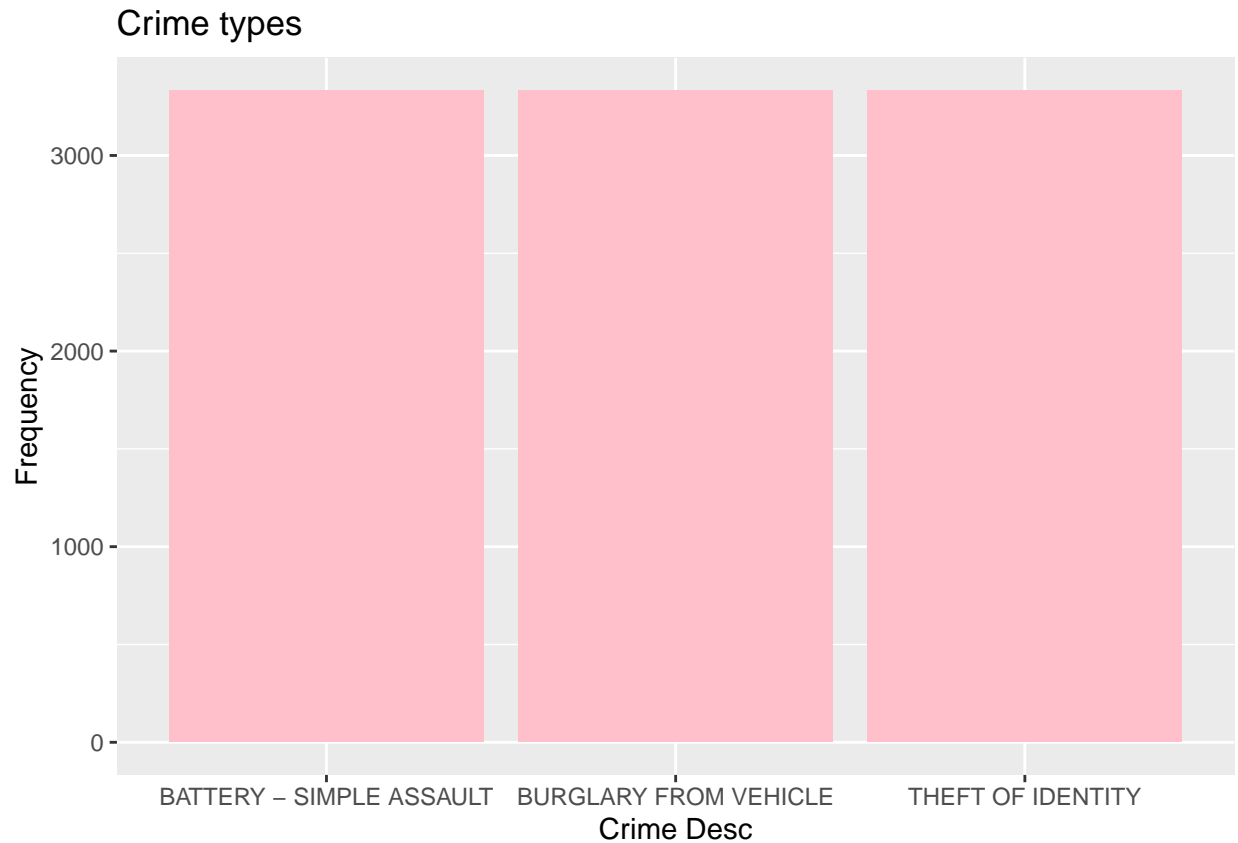
```
## [1] 24
```

4.1. Understand the target variable

Summary and Visualize the frequency of each crime type

```r
summary(final_crime_data$Crm.Cd.Desc)
```

```
##    Length     Class      Mode
##      9999 character character
```

```r
library("ggplot2")
ggplot(final_crime_data ,aes(x =  Crm.Cd.Desc))+
  geom_bar( fill= "pink")+
  labs(title = "Crime types", x = "Crime Desc", y ="Frequency")
```

Crime Type has 3 different levels: Battery - Simple Assault, Burglary from Vehicle, Theft of identity.

4.2. Understand categorical variables

Victim Sex

```
table(final_crime_data$Crm.Cd.Desc, final_crime_data$Vict.Sex)
```

```
##
##                           F    M
##   BATTERY - SIMPLE ASSAULT 1630 1703
##   BURGLARY FROM VEHICLE    1377 1956
##   THEFT OF IDENTITY        1909 1424
```

```
chisq.test(table(final_crime_data$Crm.Cd.Desc, final_crime_data$Vict.Sex))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(final_crime_data$Crm.Cd.Desc, final_crime_data$Vict.Sex)
## X-squared = 170.01, df = 2, p-value < 2.2e-16
```

The p-value is extremely small, indicating that the result is highly statistically significant => reject null hypothesis that Crm.Cd.Desc and Vict.Sex are independent.

```
library(ggplot2)
ggplot(final_crime_data, aes(x = Crm.Cd.Desc, fill = Vict.Sex)) +
  geom_bar( position = "fill" ) +
  labs(y = 'Proportion', x = 'Crime Type')
```
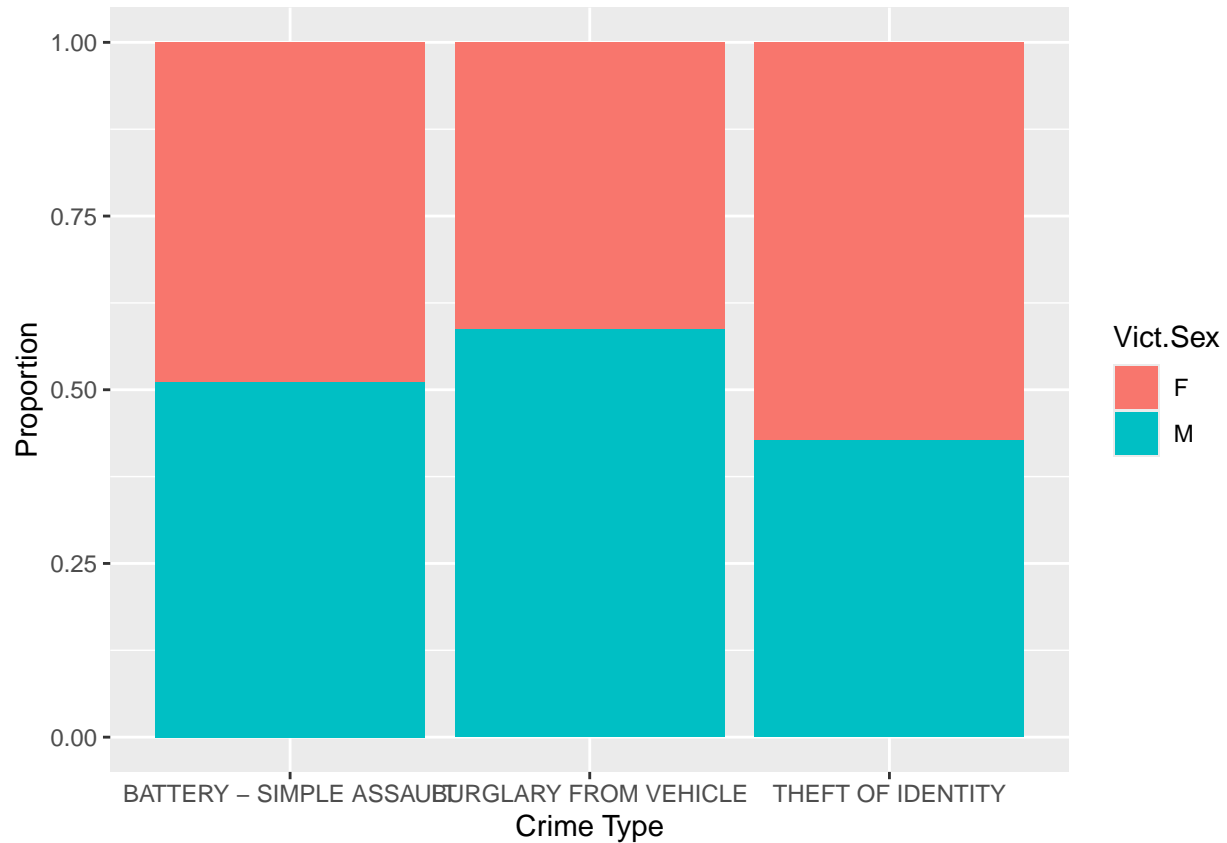

General observation shows that the crime types occurs is quite balanced between Male and Female.

AREA.NAME

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
area_name_counts <- final_crime_data %>% group_by(AREA.NAME) %>%
  summarise(count = n()) %>% arrange(desc(count))
head(area_name_counts)
```

```
## # A tibble: 6 x 2
##   AREA.NAME    count
##   <chr>        <int>
## 1 Central        931
## 2 Hollywood      550
## 3 77th Street    548
## 4 Pacific        527
## 5 Southwest      520
## 6 Wilshire       490
```

Central are has the highest rate in Crime type in the dataset.

Premis Code

```
chisq.test(table(final_crime_data$Crm.Cd.Desc, final_crime_data$Premis.Cd),
           simulate.p.value = TRUE, B = 10000)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 10000
##  replicates)
##
## data:  table(final_crime_data$Crm.Cd.Desc, final_crime_data$Premis.Cd)
## X-squared = 8227.9, df = NA, p-value = 9.999e-05
```

The p-value is extremely small, indicating that the result is highly statistically significant => reject null hypothesis that Crm.Cd.Desc and Premis.Cd are independent.

Victim Descent

```
#The frequency of victim Descent
vict_descent_counts <- final_crime_data %>% group_by(Vict.Descent) %>%
  summarise(count = n()) %>% arrange(desc(count))
vict_descent_counts
```

```
## # A tibble: 17 x 2
##    Vict.Descent count
##    <chr>        <int>
##  1 H             3754
##  2 W             2608
##  3 B             1985
##  4 O              963
##  5 A              252
##  6 K              124
##  7 F               98
##  8 C               84
##  9 J               33
## 10 X               33
## 11 V               24
## 12 I               13
## 13 Z               12
## 14 P                9
## 15 U                3
## 16 L                2
## 17 S                2
```

H (Hispanic/Latin/Mexican) occupies the highest in victims' rates

Time_Slots_Happening

```
time_slot_trend <- final_crime_data %>% group_by(Time_Slots_Happening) %>%
  summarise(count = n())
ggplot(time_slot_trend, aes(x = Time_Slots_Happening, y = count))+
  geom_bar(stat = "identity", fill = "steelblue", alpha = 0.7)
```
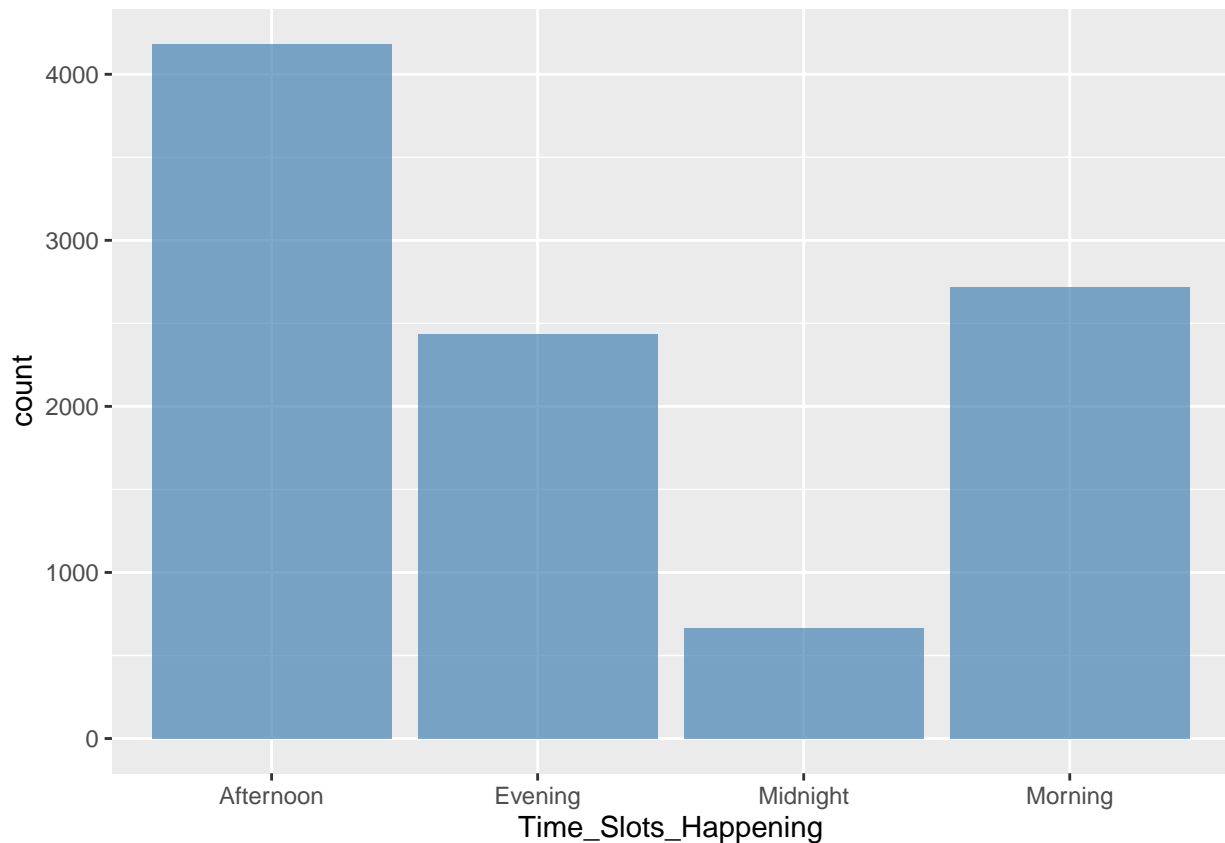

Crimes often occurs in the afternoon (12PM - 18:59PM) compared to others periods of the day. This indicates that preventive solutions during the afternoon can be effective in addressing crimes.

4.3. Understand continuous variables

Victim Age

```
ggplot(final_crime_data, aes(x = Vict.Age)) +
  geom_density(fill = 'blue', alpha = 0.5) +
  labs(title = 'Density plot', x = 'Vict.Age', y = 'density')+
  scale_x_continuous(breaks = seq(0, max(final_crime_data$Vict.Age), by = 10))
```

Density plot
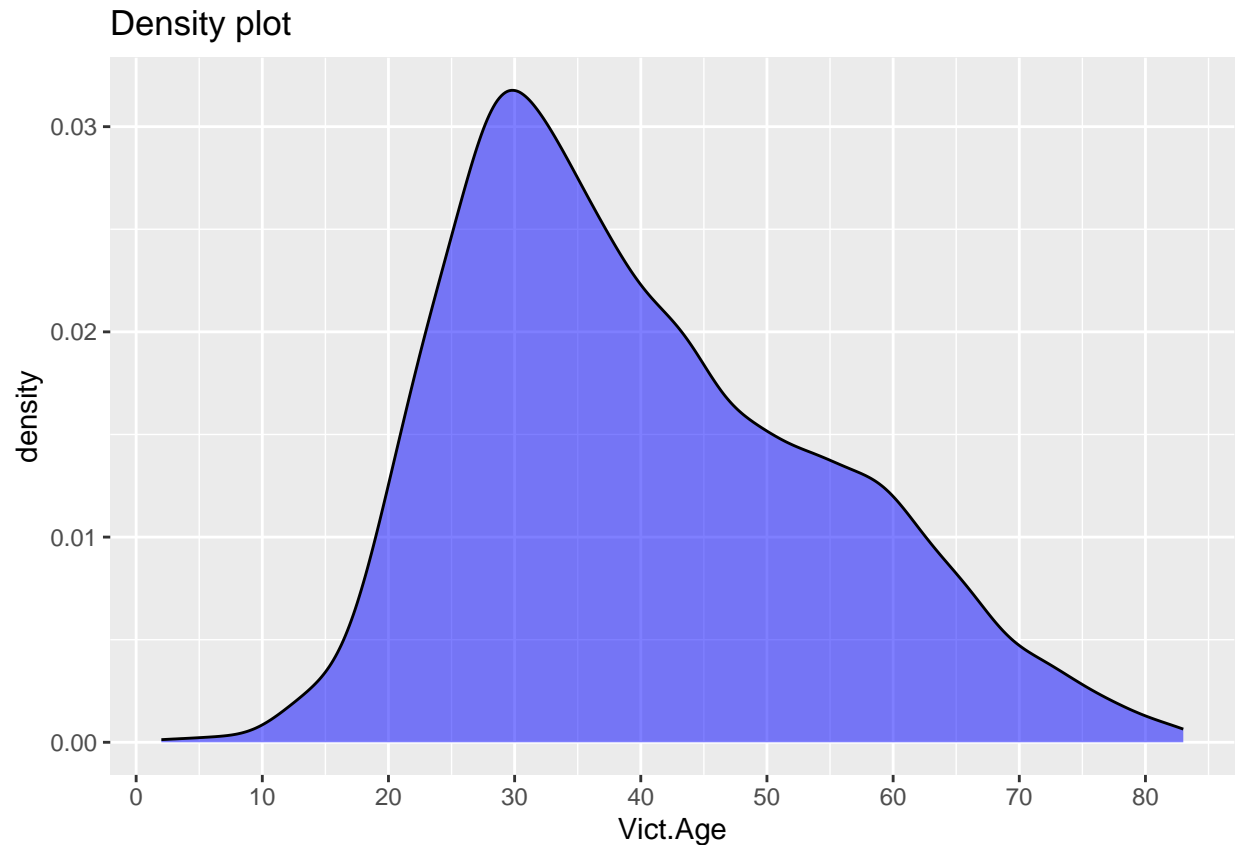
Age group between 25 and 35 has the most victims. After that, the density decrease, showing fewer victims in older age groups => The 25 - 35 age group could be related to crime occurrences due to higher activity levels outside such as work, commuting,... The older group may spend more time in private space at home, reducing exposuring crime.

```
ggplot(data = final_crime_data) +
  geom_smooth(mapping = aes(x = Vict.Age, y = Crm.Cd))
```

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```
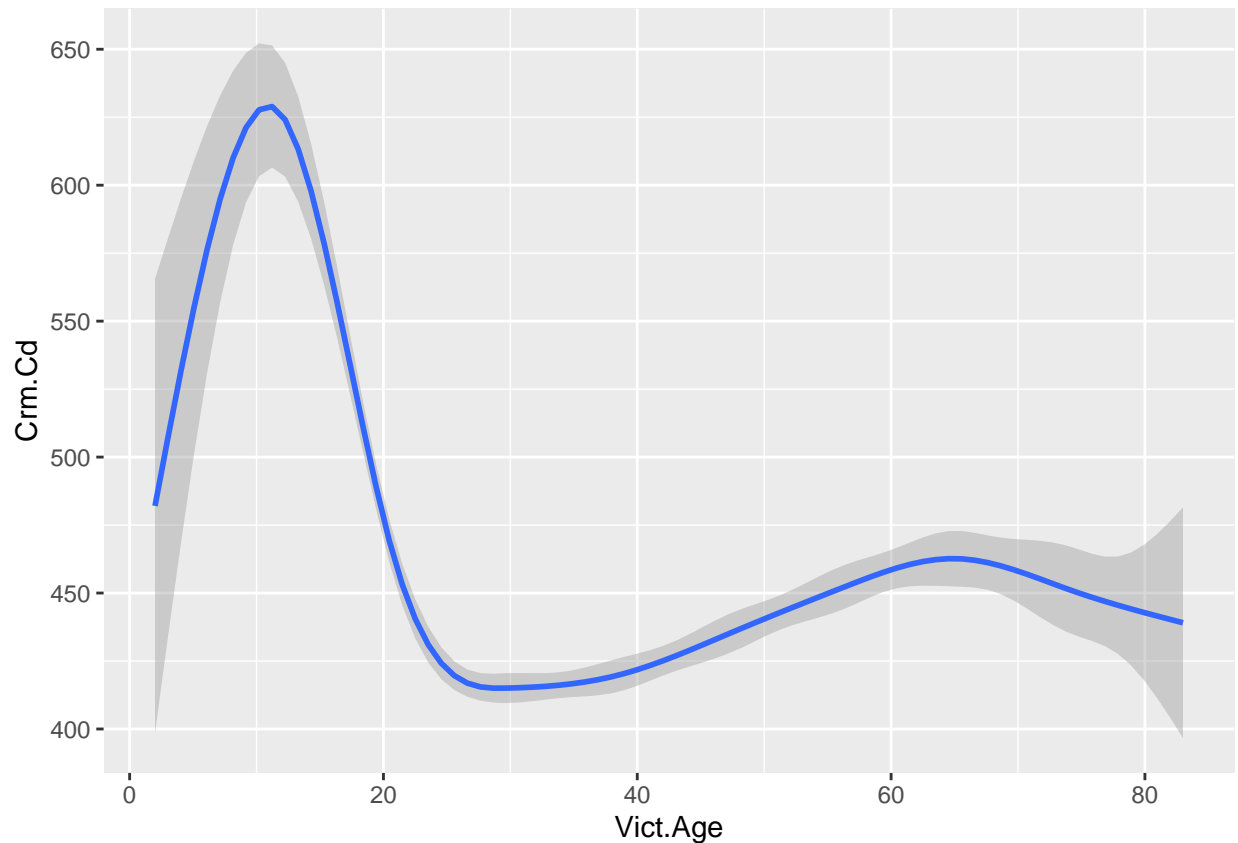
A noticeable focus of crime occurs around age 10, and Crm.Cd 624 (Battery-Simple Assault) => This crime type is more common among younger victims, it can be related to conflicts in academic environment or peer conflicts.

4.4 t-test vs ANOVA-test

t-test: Check the average age of Victims differs significantly between weekends and weekdays?

```
Age_in_Weekends <- final_crime_data %>% filter(Weekdays_of_DateOcc == 'Saturday'
                                               | Weekdays_of_DateOcc == 'Sunday' ) %>%
  select(Vict.Age)
Age_in_Weekdays <- final_crime_data %>% filter(Weekdays_of_DateOcc != 'Saturday'
                                               & Weekdays_of_DateOcc != 'Sunday' ) %>%
  select(Vict.Age)
t.test(Age_in_Weekends, Age_in_Weekdays)
```

```
##
##  Welch Two Sample t-test
##
## data:  Age_in_Weekends and Age_in_Weekdays
## t = -2.1465, df = 5625.8, p-value = 0.03187
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.2946057 -0.0586796
```

```
## sample estimates:
## mean of x mean of y
##  39.51654  40.19318
```

The p-value is smaller than 0.05, indicating that victim age differs between weekends and weekdays. On weekends, victims tend to be younger slightly, reflecting differences in activities, social hours. On weekdays, older victims may be involved in crime occurrences, possibly because of working.

ANOVA-test: comparing time_to_report across different area

```
TimeReport_aov <- aov(Time.to.reports ~ AREA.NAME, data = final_crime_data )
summary(TimeReport_aov)
```

```
##                Df     Sum Sq Mean Sq F value   Pr(>F)
## AREA.NAME      20     898257   44913   2.558 0.000155 ***
## Residuals    9978 175182929   17557
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Average_TimeReport_Areas <- final_crime_data %>%
  group_by(AREA.NAME) %>%
  summarise(Average_Time = mean(Time.to.reports, na.rm = TRUE))%>%
  arrange(desc(Average_Time))
head(Average_TimeReport_Areas)
```

```
## # A tibble: 6 x 2
##   AREA.NAME   Average_Time
##   <chr>              <dbl>
## 1 77th Street         55.6
## 2 Southeast           38.7
## 3 West Valley         38.4
## 4 Southwest           37.7
## 5 Devonshire          35.0
## 6 West LA             34.9
```

The ANOVA test result and data of average time of report show the difference between areas in reporting crime occurrences. 77th Street has the highest average reporting time, indicating the challenges in this area might be slower police response or residents taking longer time to report crimes. Central area has the shortest average time in reporting, indicating this area might have better infrastructure to facilitate quickly reports.

5.1 Splitting data into train (75%) and test (25%) data

```
library("tidymodels")
```

```
## -- Attaching packages ------------------------------------ tidymodels 1.2.0 --
```

```
## v broom        1.0.6     v rsample      1.2.1
## v dials        1.3.0     v tibble       3.2.1
```

```
## v infer        1.0.7     v tidyr        1.3.1
## v modeldata    1.4.0     v tune         1.2.1
## v parsnip      1.2.1     v workflows    1.1.4
## v purrr        1.0.2     v workflowsets 1.1.0
## v recipes      1.1.0     v yardstick    1.3.1


## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```r
library("rpart")
```

```
##
## Attaching package: 'rpart'


## The following object is masked from 'package:dials':
##
##     prune
```

```r
library("rpart.plot")

final_crime_data$Crm.Cd.Desc <- as.factor(final_crime_data$Crm.Cd.Desc)

set.seed(42)
bound <- floor(nrow(final_crime_data) / 4 * 3)
shuffled_crime_data <- final_crime_data[sample(nrow(final_crime_data)),]
train <- shuffled_crime_data[1:bound,]
test <- shuffled_crime_data[(bound+1):nrow(shuffled_crime_data),]
summary(train)
```

```
##      DR_NO            Date.Rptd           DATE.OCC            TIME.OCC
##  Min.   :200100856   Length:7499        Length:7499        Min.   : 100
##  1st Qu.:210115136   Class :character   Class :character   1st Qu.:1000
##  Median :221012273   Mode  :character   Mode  :character   Median :1400
##  Mean   :219738258                                         Mean   :1381
##  3rd Qu.:231004720                                         3rd Qu.:1845
##  Max.   :242112445                                         Max.   :2359
##       AREA          AREA.NAME          Rpt.Dist.No     Part.1.2
##  Min.   : 1.00   Length:7499        Min.   : 101   Min.   :1.000
##  1st Qu.: 5.00   Class :character   1st Qu.: 557   1st Qu.:1.000
##  Median :10.00   Mode  :character   Median :1068   Median :2.000
##  Mean   :10.44                      Mean   :1091   Mean   :1.669
##  3rd Qu.:16.00                      3rd Qu.:1612   3rd Qu.:2.000
##  Max.   :21.00                      Max.   :2197   Max.   :2.000
##      Crm.Cd                     Crm.Cd.Desc      Vict.Age
##  Min.   :330.0   BATTERY - SIMPLE ASSAULT:2502   Min.   : 2.00
##  1st Qu.:330.0   BURGLARY FROM VEHICLE   :2485   1st Qu.:29.00
##  Median :354.0   THEFT OF IDENTITY       :2512   Median :37.00
##  Mean   :436.1                                   Mean   :40.08
```

```
## 3rd Qu.:624.0                              3rd Qu.:50.00
## Max.   :624.0                              Max.   :83.00
##    Vict.Sex          Vict.Descent         Premis.Cd      Premis.Desc
## Length:7499        Length:7499         Min.   :101.0   Length:7499
## Class :character   Class :character    1st Qu.:104.0   Class :character
## Mode  :character   Mode  :character    Median :501.0   Mode  :character
##                                        Mean   :344.5
##                                        3rd Qu.:502.0
##                                        Max.   :958.0
##     Status            Status.Desc          LOCATION             LAT
## Length:7499        Length:7499         Length:7499        Min.   : 0.00
## Class :character   Class :character    Class :character   1st Qu.:34.02
## Mode  :character   Mode  :character    Mode  :character   Median :34.06
##                                                           Mean   :34.02
##                                                           3rd Qu.:34.17
##                                                           Max.   :34.33
##      LON           Time.to.reports    Delayed_Report   Time_Slots_Happening
## Min.   :-118.7   Min.   :  0.00    Min.   :0.0000   Length:7499
## 1st Qu.:-118.4   1st Qu.:  0.00    1st Qu.:0.0000   Class :character
## Median :-118.3   Median :  1.00    Median :0.0000   Mode  :character
## Mean   :-118.2   Mean   : 30.11    Mean   :0.3818
## 3rd Qu.:-118.3   3rd Qu.:  5.00    3rd Qu.:1.0000
## Max.   :  0.0   Max.   :1674.00    Max.   :1.0000
## Weekdays_of_DateOcc
## Length:7499
## Class :character
## Mode  :character
##
##
##
```

summary(test)

```
##     DR_NO             Date.Rptd           DATE.OCC            TIME.OCC
## Min.   :200104077   Length:2500        Length:2500        Min.   : 100
## 1st Qu.:210105809   Class :character   Class :character   1st Qu.: 950
## Median :220910558   Mode  :character   Mode  :character   Median :1400
## Mean   :219267677                                         Mean   :1373
## 3rd Qu.:230815836                                         3rd Qu.:1830
## Max.   :242112436                                         Max.   :2359
##      AREA          AREA.NAME          Rpt.Dist.No      Part.1.2
## Min.   : 1.00   Length:2500        Min.   : 101    Min.   :1.000
## 1st Qu.: 5.00   Class :character   1st Qu.: 563    1st Qu.:1.000
## Median :10.00   Mode  :character   Median :1071    Median :2.000
## Mean   :10.49                      Mean   :1095    Mean   :1.661
## 3rd Qu.:16.00                      3rd Qu.:1623    3rd Qu.:2.000
## Max.   :21.00                      Max.   :2197    Max.   :2.000
##     Crm.Cd                     Crm.Cd.Desc      Vict.Age
## Min.   :330.0    BATTERY - SIMPLE ASSAULT:831   Min.   : 2.00
## 1st Qu.:330.0    BURGLARY FROM VEHICLE   :848   1st Qu.:29.00
## Median :354.0    THEFT OF IDENTITY       :821   Median :37.00
## Mean   :435.6                                   Mean   :39.72
## 3rd Qu.:624.0                                   3rd Qu.:50.00
## Max.   :624.0                                   Max.   :83.00
```

```
##     Vict.Sex        Vict.Descent          Premis.Cd       Premis.Desc
##  Length:2500       Length:2500        Min.   :101.0     Length:2500
##  Class :character  Class :character   1st Qu.:108.0     Class :character
##  Mode  :character  Mode  :character   Median :501.0     Mode  :character
##                                       Mean   :349.2
##                                       3rd Qu.:502.0
##                                       Max.   :957.0
##     Status           Status.Desc         LOCATION              LAT
##  Length:2500       Length:2500        Length:2500        Min.   : 0.00
##  Class :character  Class :character   Class :character   1st Qu.:34.02
##  Mode  :character  Mode  :character   Mode  :character   Median :34.06
##                                                          Mean   :34.03
##                                                          3rd Qu.:34.17
##                                                          Max.   :34.33
##      LON           Time.to.reports   Delayed_Report  Time_Slots_Happening
##  Min.   :-118.7   Min.   :   0.00   Min.   :0.000   Length:2500
##  1st Qu.:-118.4   1st Qu.:   0.00   1st Qu.:0.000   Class :character
##  Median :-118.3   Median :   1.00   Median :0.000   Mode  :character
##  Mean   :-118.2   Mean   :  29.31   Mean   :0.386
##  3rd Qu.:-118.3   3rd Qu.:   4.00   3rd Qu.:1.000
##  Max.   :   0.0   Max.   :1744.00   Max.   :1.000
##  Weekdays_of_DateOcc
##  Length:2500
##  Class :character
##  Mode  :character
##
##
##
```

5.2. Classify crime types based on Victim demographics and crime locations (Comparing performance of models: Decision tree, random forest, SVM)

SVM and its performance

```r
# Creating model and training model
library("e1071")
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:tune':
##
##     tune
```

```
## The following object is masked from 'package:rsample':
##
##     permutations
```

```
## The following object is masked from 'package:parsnip':
##
##     tune
```

```
classifier <- svm(Crm.Cd.Desc ~ Vict.Descent + Vict.Age + Vict.Sex +
                  AREA.NAME +Time_Slots_Happening + Weekdays_of_DateOcc + Premis.Cd,
                data = train, type = "C-classification", kernel = "radial")
summary(classifier)
```

```
##
## Call:
## svm(formula = Crm.Cd.Desc ~ Vict.Descent + Vict.Age + Vict.Sex +
##     AREA.NAME + Time_Slots_Happening + Weekdays_of_DateOcc + Premis.Cd,
##     data = train, type = "C-classification", kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  5862
##
##  ( 2462 1815 1585 )
##
##
## Number of Classes:  3
##
## Levels:
##  BATTERY - SIMPLE ASSAULT BURGLARY FROM VEHICLE THEFT OF IDENTITY
```

```
test$test_pred_svm <- predict(classifier, test)
#Confusion matrix
conf_mat(test, truth=Crm.Cd.Desc, estimate = test_pred_svm)
```

```
##                            Truth
## Prediction                  BATTERY - SIMPLE ASSAULT BURGLARY FROM VEHICLE
##    BATTERY - SIMPLE ASSAULT                      190                     92
##    BURGLARY FROM VEHICLE                         326                    649
##    THEFT OF IDENTITY                             315                    107
##                            Truth
## Prediction                  THEFT OF IDENTITY
##    BATTERY - SIMPLE ASSAULT                78
##    BURGLARY FROM VEHICLE                   33
##    THEFT OF IDENTITY                      710
```

```
#get summary metrics
dt_metrics <- metric_set(accuracy, sens, spec, f_meas, kap)
dt_metrics(test, truth = Crm.Cd.Desc, estimate = test_pred_svm)
```

```
## # A tibble: 5 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.620
## 2 sens     macro          0.620
## 3 spec     macro          0.810
```

```
## 4 f_meas    macro         0.582
## 5 kap       multiclass    0.429
```

Accuracy of 62% indicates the model predicts correctly the crime types about 62% of the time. Sensitivity of 62% indicates the model captures around 62% of the actual crime types across all categories. Specificity of 81% indicates the model does a good job in identifying correctly non crime types. It s higher than the sensitivity, showing that the model does better in identifying when there is not a crime happening rather than when it is. f_meas of 58.2% indicates the balance of precision and recall, indicating the model is doing fairly well but still has room for improvement. kap of 42.9% means a moderate level of agreement between the actual values and the model prediction

Decision tree and its performance

```r
tree <- decision_tree() %>% set_engine("rpart") %>% set_mode("classification")
#create recipe
df_recipe <- recipe(Crm.Cd.Desc ~ Vict.Descent + Vict.Age + Vict.Sex +AREA.NAME+
                    Time_Slots_Happening + Weekdays_of_DateOcc + Premis.Cd,
                 data = train) %>% step_normalize((all_numeric()))
#create decision tree workflow
tree_wf <- workflow() %>% add_recipe(df_recipe) %>% add_model(tree) %>% fit(train)
predResults <- data.frame(predict(tree_wf, test))
colnames(predResults) <- c("test_pred_tree")
test <- cbind(test, predResults)
conf_mat(test, truth=Crm.Cd.Desc, estimate = test_pred_tree)
```

```
##                            Truth
## Prediction                 BATTERY - SIMPLE ASSAULT BURGLARY FROM VEHICLE
##    BATTERY - SIMPLE ASSAULT                     282                    70
##    BURGLARY FROM VEHICLE                        327                   737
##    THEFT OF IDENTITY                            222                    41
##                            Truth
## Prediction                 THEFT OF IDENTITY
##    BATTERY - SIMPLE ASSAULT             133
##    BURGLARY FROM VEHICLE                 24
##    THEFT OF IDENTITY                    664
```

```r
#get summary metrics
library(yardstick)
dt_metricsS <- metric_set(accuracy, sens, spec, f_meas, kap)
dt_metricsS(test, truth = Crm.Cd.Desc, estimate = test_pred_tree)
```

```
## # A tibble: 5 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.673
## 2 sens     macro          0.672
## 3 spec     macro          0.836
## 4 f_meas   macro          0.650
## 5 kap      multiclass     0.509
```

Accuracy of 67.3% indicates the model predicts correctly the crime types about 67.3% of the time. Sensitivity of 67.2% indicates the model captures around 67.2% of the actual crime types across all categories. Specificity of 83.6% indicates the model does a good job in identifying correctly non crime types. It s higher than the

sensitivity, showing that the model does better in identifying when there is not a crime happening rather than when it is. f_meas of 65% indicates the balance of precision and recall, indicating the model is doing fairly well but still has room for improvement. kap of 50.9% means a moderate level of agreement between the actual values and the model prediction

Random Forest and its performance

```r
rf <- rand_forest() %>% set_engine("ranger", importance = "impurity") %>%
set_mode("classification")

df_recipe <- recipe(Crm.Cd.Desc ~ Vict.Descent + Vict.Age + Vict.Sex +AREA.NAME+
                    Time_Slots_Happening + Weekdays_of_DateOcc + Premis.Cd,
                  data = train) %>% step_normalize((all_numeric())))

random_wf_52 <- workflow() %>% add_recipe(df_recipe) %>% add_model(rf) %>% fit(train)

summary(random_wf_52)
```

```
##          Length Class       Mode
## pre      3      stage_pre   list
## fit      2      stage_fit   list
## post     1      stage_post  list
## trained  1      -none-      logical
```

```r
#Creating new column test_pred_rf containing predicted values about Crm.Cd.Desc on test data
predResults <- data.frame(predict(random_wf_52,test))
colnames(predResults) <- c("test_pred_rf")
test <- cbind(test, predResults)
#Confusion matrix
conf_mat(test, truth=Crm.Cd.Desc, estimate = test_pred_rf)
```

```
##                             Truth
## Prediction                   BATTERY - SIMPLE ASSAULT BURGLARY FROM VEHICLE
##    BATTERY - SIMPLE ASSAULT                       418                    106
##    BURGLARY FROM VEHICLE                          181                    679
##    THEFT OF IDENTITY                              232                     63
##                             Truth
## Prediction                   THEFT OF IDENTITY
##    BATTERY - SIMPLE ASSAULT                 93
##    BURGLARY FROM VEHICLE                    29
##    THEFT OF IDENTITY                       699
```

```r
#get summary metrics
library(yardstick)
dt_metricsS <- metric_set(accuracy, sens, spec, f_meas, kap)
dt_metricsS(test, truth = Crm.Cd.Desc, estimate = test_pred_rf)
```

```
## # A tibble: 5 x 3
##    .metric  .estimator .estimate
##    <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.718
## 2 sens     macro          0.718
## 3 spec     macro          0.859
```

```
## 4 f_meas    macro         0.710
## 5 kap       multiclass    0.578
```

Three classification models was used to predict crime types. I created the confusion matrix and evaluation metrics to gain details about how well the model performed. Random Forest Model has the best performance than SVM, Decision tree models in almost key metrics: higher accuracy(nearly 72%), better sensitivity, Superior Specificity, Improved F-measure, and stronger Kappa Score. Especially, the higher in accurancy make Random Forest a stronger option to classify Crime Type based on Victim Demographic and Crime Locations, Time. However, with Random Forest, nearly 28% of predictions predicted incorrectly, so there is still room for improvement to further enhance the performance of this model.

5.3. Logistic Regression: Classify Time.to.reports as Delayed (>30 days) or Timely (<= 30days)

```r
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked from 'package:yardstick':
##
##     precision, recall, sensitivity, specificity
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
table(final_crime_data$Delayed_Report)
```

```
##
##    0    1
## 6171 3828
```

```r
glm.fit <- glm(Delayed_Report ~ Time_Slots_Happening + Weekdays_of_DateOcc +
                 Vict.Age + Vict.Sex + Vict.Descent + AREA.NAME + Crm.Cd.Desc ,
               data = train, family = binomial)

predictedprob <- predict(glm.fit, newdata = test, type = "response")
head(predictedprob)
```

```
##       6687        867        503       7704       5866       4489
## 0.72966187 0.09328995 0.08140300 0.67360483 0.27775667 0.31857134
```

```
newdata <- data.frame(test$Time_Slots_Happening, test$Weekdays_of_DateOcc,
                      test$Vict.Age, test$Vict.Sex, test$Vict.Descent,
                      test$AREA.NAME, test$Crm.Cd.Desc ,predictedprob)

glm.pred = factor(ifelse(predictedprob > 0.5, 1 , 0))
test$Delayed_Report <- factor(test$Delayed_Report, levels = c(0, 1))
confusionMatrix(test$Delayed_Report, glm.pred)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1276  259
##          1  353  612
##
##                Accuracy : 0.7552
##                  95% CI : (0.7379, 0.7719)
##     No Information Rate : 0.6516
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.474
##
##  Mcnemar's Test P-Value : 0.0001704
##
##             Sensitivity : 0.7833
##             Specificity : 0.7026
##          Pos Pred Value : 0.8313
##          Neg Pred Value : 0.6342
##              Prevalence : 0.6516
##          Detection Rate : 0.5104
##    Detection Prevalence : 0.6140
##       Balanced Accuracy : 0.7430
##
##        'Positive' Class : 0
##
```

The accuracy of nearly 75.5%, showing this model performs reasonably well in distinguishing whether a crime was reported on time or not. And this model performs better at predicting delayed reports than timely reports, which was shown by the higher PPV than NPV. The p-value (Acc > NIR) is statistically significant, showing that the performance of this model is meaningful.