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.:210109908   Class :character   Class :character   1st Qu.: 945
##  Median :220916554   Mode  :character   Mode  :character   Median :1400
##  Mean   :219457344                                         Mean   :1365
##  3rd Qu.:230907875                                         3rd Qu.:1840
##  Max.   :242112226                                         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.: 564   1st Qu.:1.000
##  Median :11.00   Mode  :character   Median :1101   Median :2.000
##  Mean   :10.54                      Mean   :1100   Mean   :1.667
##  3rd Qu.:16.00                      3rd Qu.:1638   3rd Qu.:2.000
##  Max.   :21.00                      Max.   :2199   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   :39.82
##  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   :344.4
##                     3rd Qu.:502.0
##                     Max.   :971.0
##  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   :34.03   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   :  27.25   Mean   :0.3784
##  3rd Qu.:   4.00   3rd Qu.:1.0000
##  Max.   :1583.00   Max.   :1.0000
```

```r
ncol(final_crime_data)
```

```
## [1] 24
```

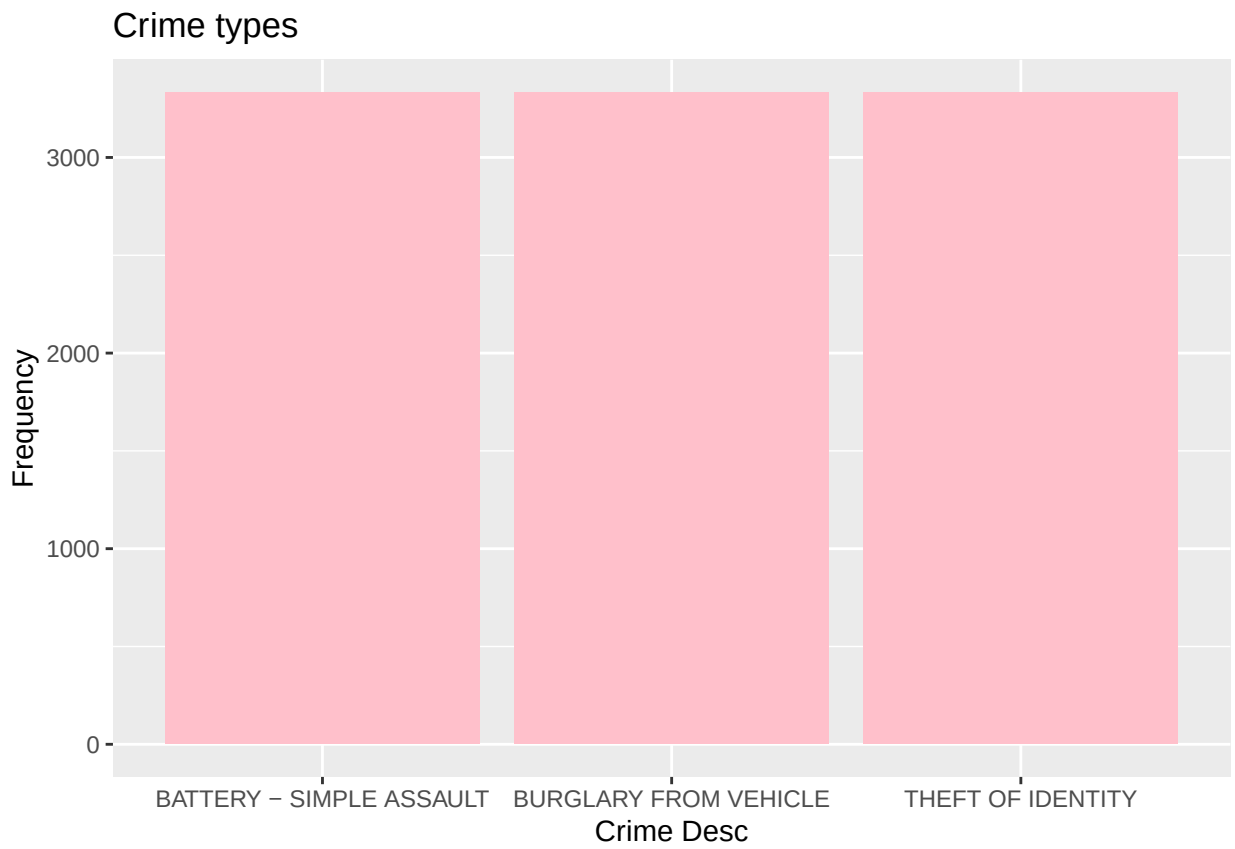
4.1. Understand the target variable

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)
```

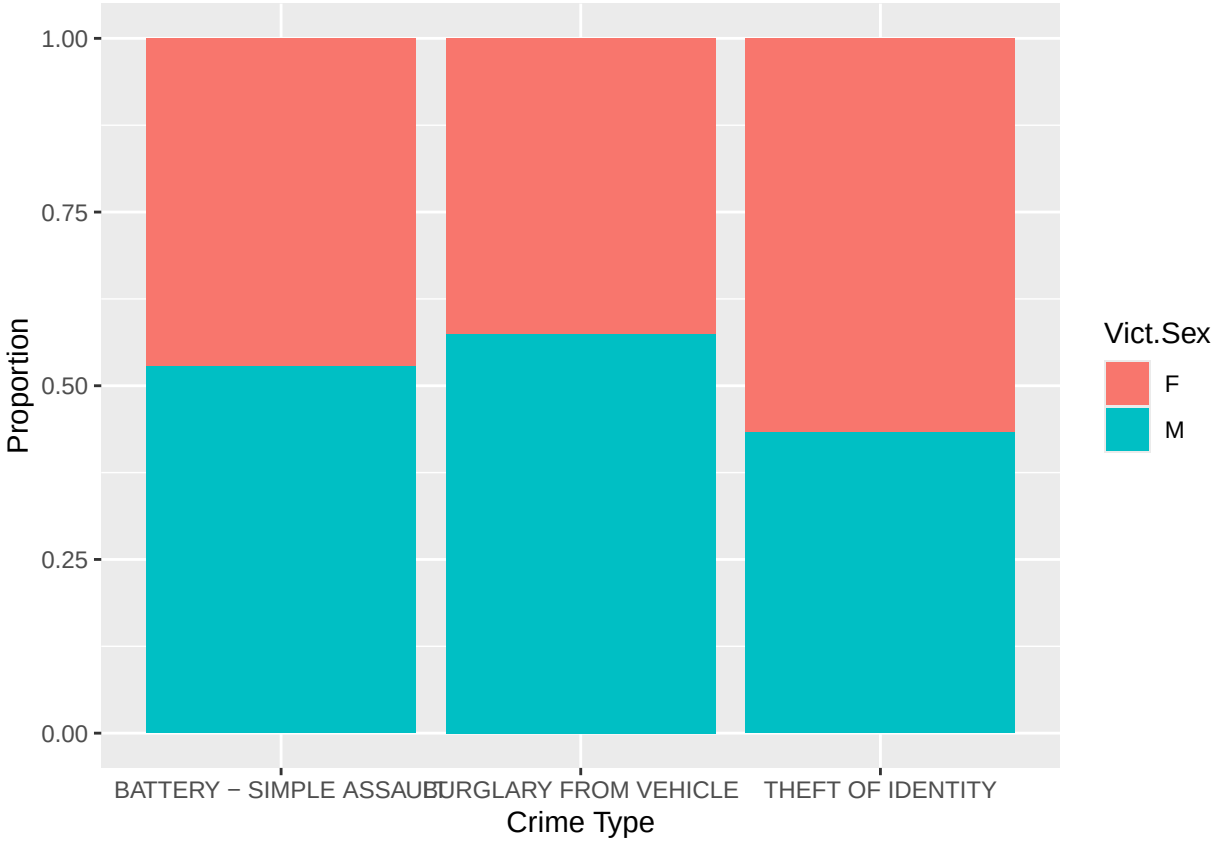
```
##
##                               F    M
##   BATTERY - SIMPLE ASSAULT 1573 1760
##   BURGLARY FROM VEHICLE    1419 1914
##   THEFT OF IDENTITY        1890 1443
```

```
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 = 138.51, 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       943
## 2 Hollywood     595
## 3 77th Street   587
## 4 N Hollywood   525
## 5 Southwest     511
## 6 Southeast     508
```

Central areas 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 = 8482.2, 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

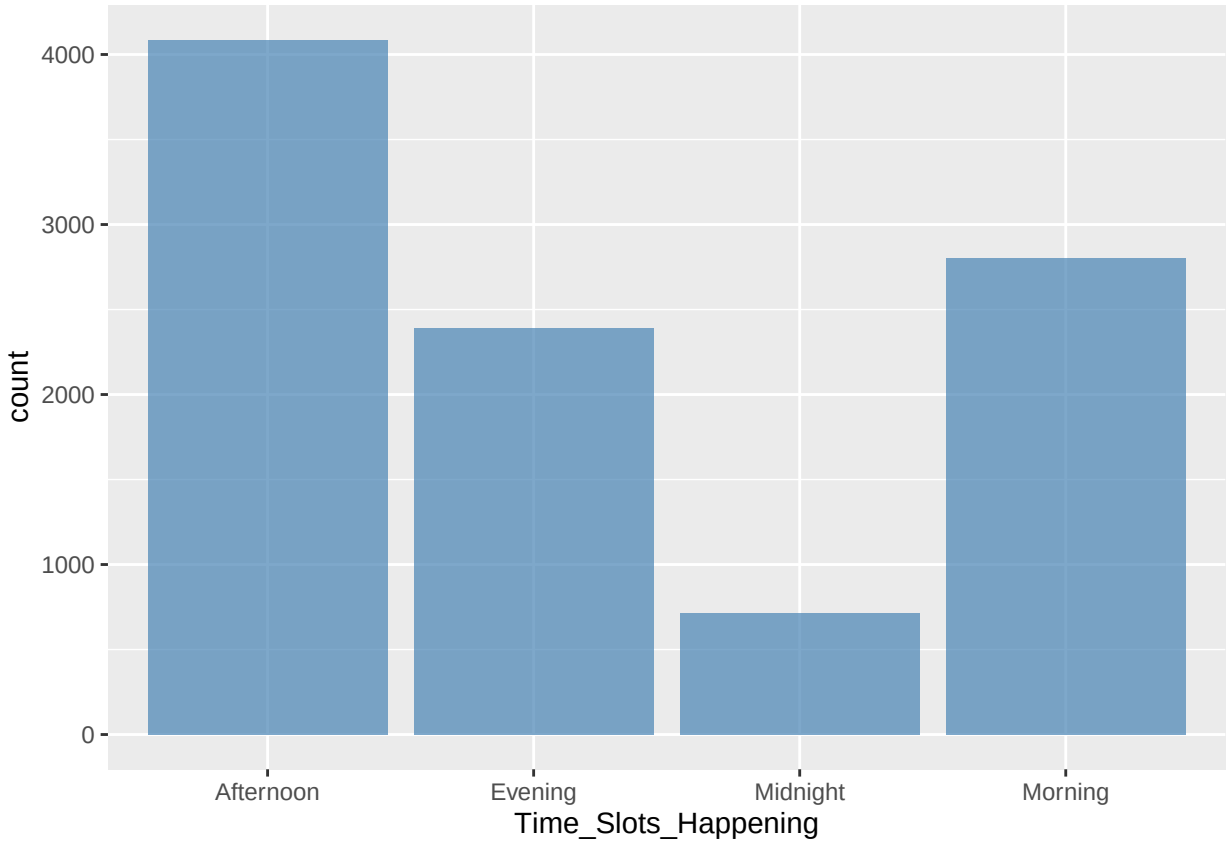
```
#The frequency of victim Descent
vict_descent_counts <- final_crime_data %>% group_by(Vict.Descent) %>%
  summarise(count = n()) %>% arrange(desc(count))
vict_descent_counts
```

```
## # A tibble: 18 x 2
##    Vict.Descent count
##    <chr>        <int>
##  1 H             3881
##  2 W             2642
##  3 B             1925
##  4 O              852
##  5 A              278
##  6 K              118
##  7 C               96
##  8 F               81
##  9 X               43
## 10 J               25
## 11 V               20
## 12 I               12
## 13 Z               12
## 14 P                6
## 15 U                5
## 16 D                1
## 17 G                1
## 18 L                1
```

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)
```

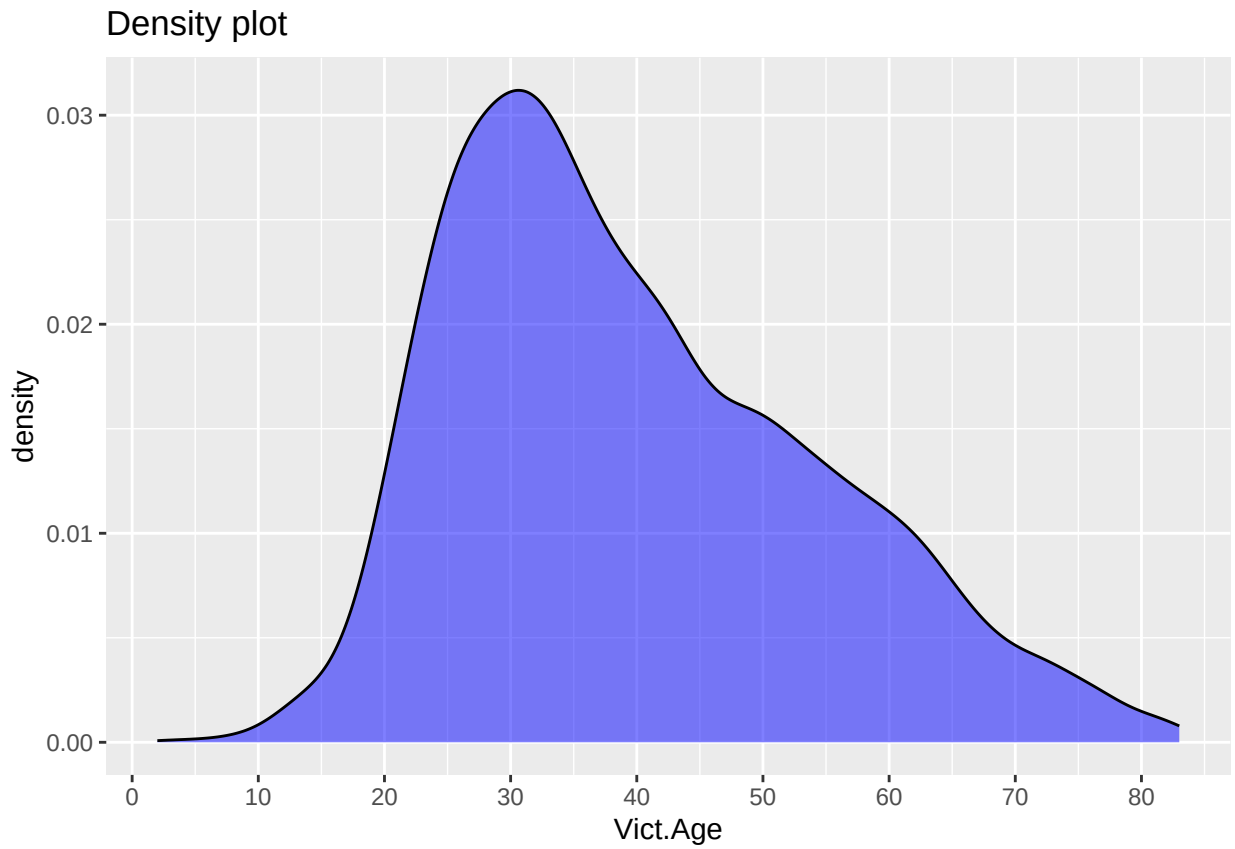


Crimes often occurs in the afternoon (12PM - 18:59PM) compared to others periods of the day. This indicates that preventive solutions during the afternoon can be effective in addressing crimes.

4.3. Understand continuous variables

Victim Age

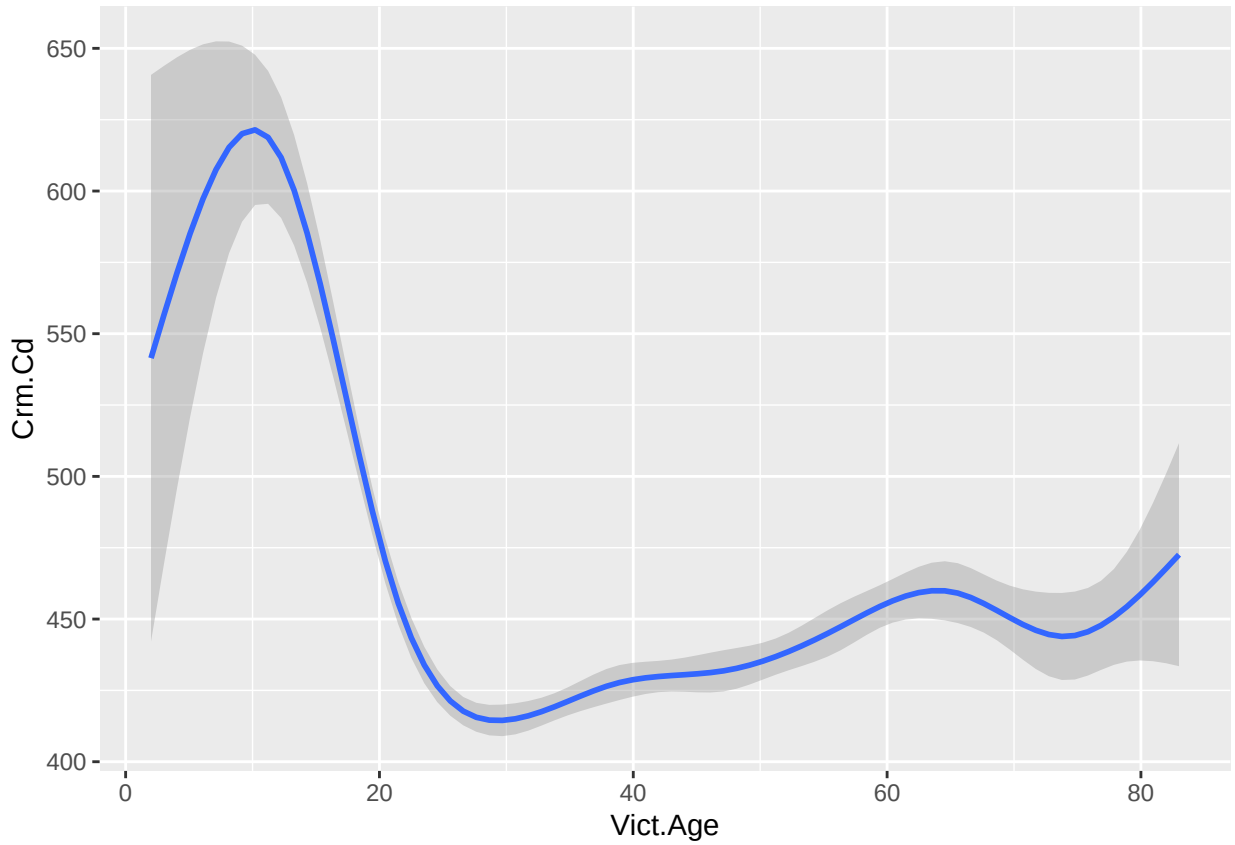
```
ggplot(final_crime_data, aes(x = Vict.Age)) +
  geom_density(fill = 'blue', alpha = 0.5) +
  labs(title = 'Density plot', x = 'Vict.Age', y = 'density')+
  scale_x_continuous(breaks = seq(0, max(final_crime_data$Vict.Age), by = 10))
```



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 = -2.8798, df = 5240.6, p-value = 0.003995
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.5493682 -0.2943072
```

```
## sample estimates:
## mean of x mean of y
##  39.15574  40.07758
```

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    544697   27235    1.87 0.0106 *
## Residuals   9978 145346947   14567
## ---
## 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 Devonshire          42.7
## 2 Southwest           36.8
## 3 77th Street         36.5
## 4 Topanga             36.2
## 5 West Valley         31.8
## 6 N Hollywood         31.2
```

The ANOVA test result and data of average time of report show the difference between areas in reporting crime occurrences.Devonshire 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. N Hollywood area has the shorter 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()
## * Learn how to get started at https://www.tidymodels.org/start/
```

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

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

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

```
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.   : 100.0
##  1st Qu.:210111230   Class :character   Class :character   1st Qu.: 955.5
##  Median :220915189   Mode  :character   Mode  :character   Median :1400.0
##  Mean   :219487029                                         Mean   :1363.3
##  3rd Qu.:230913504                                         3rd Qu.:1830.0
##  Max.   :242112226                                         Max.   :2359.0
##       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.: 563   1st Qu.:1.000
##  Median :11.00   Mode  :character   Median :1101   Median :2.000
##  Mean   :10.54                      Mean   :1101   Mean   :1.669
##  3rd Qu.:16.00                      3rd Qu.:1644   3rd Qu.:2.000
##  Max.   :21.00                      Max.   :2199   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   :39.83
```

```
##  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.:103.0   Class :character
##  Mode  :character   Mode  :character   Median :501.0   Mode  :character
##                                        Mean   :343.3
##                                        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   :34.03
##                                                           3rd Qu.:34.17
##                                                           Max.   :34.33
##       LON         Time.to.reports   Delayed_Report   Time_Slots_Happening
##  Min.   :-118.7   Min.   :   0.00   Min.   :0.0000   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   :  26.88   Mean   :0.3799
##  3rd Qu.:-118.3   3rd Qu.:   4.00   3rd Qu.:1.0000
##  Max.   :   0.0   Max.   :1583.00   Max.   :1.0000
##  Weekdays_of_DateOcc
##  Length:7499
##  Class :character
##  Mode  :character
##
##
##
```

```
summary(test)
```

```
##      DR_NO            Date.Rptd           DATE.OCC            TIME.OCC
##  Min.   :200104286   Length:2500        Length:2500        Min.   : 100
##  1st Qu.:210107238   Class :character   Class :character   1st Qu.: 930
##  Median :221005632   Mode  :character   Mode  :character   Median :1400
##  Mean   :219368301                                         Mean   :1369
##  3rd Qu.:230814910                                         3rd Qu.:1900
##  Max.   :242111297                                         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.: 588   1st Qu.:1.000
##  Median :10.00   Mode  :character   Median :1099   Median :2.000
##  Mean   :10.52                      Mean   :1099   Mean   :1.661
##  3rd Qu.:16.00                      3rd Qu.:1621   3rd Qu.:2.000
##  Max.   :21.00                      Max.   :2196   Max.   :2.000
##      Crm.Cd                      Crm.Cd.Desc     Vict.Age
##  Min.   :330.0   BATTERY - SIMPLE ASSAULT:831   Min.   : 4.00
##  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   :39.78
##  3rd Qu.:624.0                                  3rd Qu.:49.00
##  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.:104.0   Class :character
##  Mode  :character   Mode  :character   Median :501.0   Mode  :character
##                                        Mean   :347.6
##                                        3rd Qu.:502.0
##                                        Max.   :971.0
##     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   :34.02
##                                                           3rd Qu.:34.17
##                                                           Max.   :34.32
##       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   :  28.38   Mean   :0.374
##  3rd Qu.:-118.3   3rd Qu.:   4.00   3rd Qu.:1.000
##  Max.   :   0.0   Max.   :1464.00   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

```
# 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:  5847
##
##  ( 2492 1799 1556 )
##
##
## Number of Classes:  3
##
## Levels:
##  BATTERY - SIMPLE ASSAULT BURGLARY FROM VEHICLE THEFT OF IDENTITY
```

```
#Create predicted column in the test data
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                      191                    76
##   BURGLARY FROM VEHICLE                         332                   681
##   THEFT OF IDENTITY                             308                    91
##                           Truth
## Prediction                 THEFT OF IDENTITY
##   BATTERY - SIMPLE ASSAULT                90
##   BURGLARY FROM VEHICLE                   42
##   THEFT OF IDENTITY                      689
```

```
#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     0.624
## 2 sens     macro          0.624
```

```
## 3 spec     macro          0.812
## 4 f_meas   macro          0.586
## 5 kap      multiclass     0.436
```

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 ~59% indicates the balance of precision and recall, indicating the model is doing fairly well but still has room for improvement. kap of ~44% means a moderate level of agreement between the actual values and the model prediction

Decision tree and its performance

```
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))
#Create predicted column in the test data
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                       272                    57
##   BURGLARY FROM VEHICLE                          327                   750
##   THEFT OF IDENTITY                              232                    41
##                            Truth
## Prediction                  THEFT OF IDENTITY
##   BATTERY - SIMPLE ASSAULT               120
##   BURGLARY FROM VEHICLE                   30
##   THEFT OF IDENTITY                      671
```

```
#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     0.677
## 2 sens     macro          0.676
## 3 spec     macro          0.838
## 4 f_meas   macro          0.651
## 5 kap      multiclass     0.515
```

Accuracy of 67.7% indicates the model predicts correctly the crime types about 67.7% of the time. Sensitivity of 67.6% indicates the model captures around 67.6% of the actual crime types across all categories. Specificity

of 83.8% 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 51.5% means a moderate level of agreement between the actual values and the model prediction

Random Forest and its performance

```
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
```

```
#Creating new column test_pred_rf containing predicted values about Crm.Cd.Desc on test data
predResults <- data.frame(predict(random_wf_52,test))
#Create predicted column in the test data
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                      397                    95
##   BURGLARY FROM VEHICLE                         201                   697
##   THEFT OF IDENTITY                             233                    56
##                           Truth
## Prediction                 THEFT OF IDENTITY
##   BATTERY - SIMPLE ASSAULT                94
##   BURGLARY FROM VEHICLE                   35
##   THEFT OF IDENTITY                      692
```

```
#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     0.714
```

```
## 2 sens      macro          0.714
## 3 spec      macro          0.857
## 4 f_meas    macro          0.704
## 5 kap       multiclass     0.572
```

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 (>1 day) or Timely (<= 1 day)

```
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
```

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

```
##
##    0    1
## 6215 3784
```

```
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
## 0.68462659 0.06124955 0.11710389 0.64990722 0.30934299 0.33375884
```

```r
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 1304  261
##          1  337  598
##
##                Accuracy : 0.7608
##                  95% CI : (0.7436, 0.7774)
##     No Information Rate : 0.6564
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4807
##
##  Mcnemar's Test P-Value : 0.002162
##
##             Sensitivity : 0.7946
##             Specificity : 0.6962
##          Pos Pred Value : 0.8332
##          Neg Pred Value : 0.6396
##              Prevalence : 0.6564
##          Detection Rate : 0.5216
##    Detection Prevalence : 0.6260
##       Balanced Accuracy : 0.7454
##
##        'Positive' Class : 0
##
```

The accuracy of nearly 76.1%, 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.