Appendix

Ryan S Dunn

11/28/2021

Data Importing and Engineering

Import the relevent data sets for EDA and model development

```
#import the data sets for EDA
library(readxl)

#import the police shootings since 2015 data
police_post2015 <- read.csv("~/Documents/USD MS-ADS/Applied Data Mining
502/Final Project/PoliceShootings_post2015.txt")

#import supplementary income, poverty, race, and high school graduation data
for data blending/joining
median_income <- read_excel("~/Documents/USD MS-ADS/Applied Data Mining
502/Final Project/MedianHouseholdIncome2015.xlsx")
povery_level <- read_excel("~/Documents/USD MS-ADS/Applied Data Mining
502/Final Project/PercentagePeopleBelowPovertyLevel.xlsm")
race_city <- read_excel("~/Documents/USD MS-ADS/Applied Data Mining 502/Final
Project/ShareRaceByCity.xlsm")
hs_grad <- read_excel("~/Documents/USD MS-ADS/Applied Data Mining 502/Final
Project/PercentOver25CompletedHighSchool.xlsm")
```

Import the necessary libraries

```
library(ggplot2)
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
```

Develop a median income data frame to join onto the police_shootings dataframe

```
#develop a median income object to join onto the police shootings data frame
income_df <- data.frame((median_income))

#change data types as needed
income_df$Median.Income <- as.numeric(income_df$Median.Income)
```

```r
income_df$Geographic.Area <- as.factor(income_df$Geographic.Area)

#aggregate the median income via the median median income of each state
income_table <- aggregate(x = income_df$Median.Income,
          by = list(income_df$Geographic.Area),
          FUN =  median)

#save the income_table as a data frame and convert the names of the columns
income_table <- as.data.frame(income_table)
income_table <- rename(income_table, "State" = "Group.1")
income_table <- rename(income_table, "Median.Income" = "x")

#view the developed object
income_table

##    State Median.Income
## 1     AK       50000.0
## 2     AL       38304.0
## 3     AR       33750.0
## 4     AZ       39000.0
## 5     CA       54667.0
## 6     CO       50220.5
## 7     CT       69200.0
## 8     DC       70848.0
## 9     DE       57448.0
## 10    FL       44679.0
## 11    GA       35833.0
## 12    HI       63453.0
## 13    IA       45714.0
## 14    ID       41250.0
## 15    IL       47969.0
## 16    IN       43359.0
## 17    KS       42500.0
## 18    KY       37632.0
## 19    LA       38569.0
## 20    MA       66370.0
## 21    MD       70511.0
## 22    ME       42227.0
## 23    MI       41228.0
## 24    MN       47188.0
## 25    MO       36852.5
## 26    MS       31800.0
## 27    MT       41875.0
## 28    NC       37000.0
## 29    ND       48702.0
## 30    NE       44167.0
## 31    NH       52636.0
## 32    NJ       75357.5
## 33    NM       37337.0
## 34    NV       50153.0
```

```
## 35     NY       56250.0
## 36     OH       43967.5
## 37     OK       37896.0
## 38     OR       43125.0
## 39     PA       45793.5
## 40     RI       71786.0
## 41     SC       34250.0
## 42     SD       43409.0
## 43     TN       37746.0
## 44     TX       43069.5
## 45     UT       52500.0
## 46     VA       40833.0
## 47     VT       43354.0
## 48     WA       45013.0
## 49     WI       44167.0
## 50     WV       36250.0
## 51     WY       51384.0
```

Develop a povery rate data frame to join onto the police_shootings dataframe

```
#develop a poverty rate object to join onto the police shootings data frame
pr_df <- data.frame(povery_level)

#change the data types as needed
pr_df$poverty_rate  <- as.numeric(pr_df$poverty_rate)
pr_df$Geographic.Area <- as.factor(pr_df$Geographic.Area)

#aggregate the poverty rate via the median poverty rate of each state
pr_table <- aggregate(x = pr_df$poverty_rate,
          by = list(pr_df$Geographic.Area),
          FUN =  median)

#save the pr_table as a data frame and convert the names of the columns
pr_table <- as.data.frame(pr_table)
pr_table  <- rename(pr_table, "State" = "Group.1")
pr_table <- rename(pr_table, "Median.Below.Poverty" = 'x')

#view the object
pr_table

##    State Median.Below.Poverty
## 1     AK                14.95
## 2     AL                19.10
## 3     AR                22.30
## 4     AZ                20.35
## 5     CA                13.40
## 6     CO                11.55
## 7     CT                 7.70
## 8     DC                18.00
## 9     DE                11.10
## 10    FL                15.00
```

```
## 11    GA                 23.50
## 12    HI                 11.10
## 13    IA                 10.70
## 14    ID                 16.10
## 15    IL                 12.20
## 16    IN                 14.80
## 17    KS                 12.80
## 18    KY                 19.50
## 19    LA                 21.00
## 20    MA                  8.20
## 21    MD                  7.45
## 22    ME                 17.50
## 23    MI                 16.10
## 24    MN                 11.60
## 25    MO                 18.50
## 26    MS                 26.45
## 27    MT                 12.80
## 28    NC                 17.95
## 29    ND                  8.85
## 30    NE                 11.60
## 31    NH                 10.50
## 32    NJ                  6.40
## 33    NM                 19.70
## 34    NV                 10.20
## 35    NY                  9.60
## 36    OH                 13.30
## 37    OK                 18.80
## 38    OR                 16.20
## 39    PA                 10.80
## 40    RI                  8.55
## 41    SC                 22.20
## 42    SD                 11.10
## 43    TN                 19.45
## 44    TX                 17.00
## 45    UT                  9.35
## 46    VA                 11.80
## 47    VT                 14.20
## 48    WA                 12.30
## 49    WI                 11.50
## 50    WV                 19.15
## 51    WY                  6.40
```

Develop a percent of population over 25 years old that has graduated from high school data frame to join onto the police_shootings dataframe

```
#develop a hs rate object to join onto the police shootings data frame
hs_df <- data.frame(hs_grad)

#change the data types as needed
hs_df$Geographic.Area <- as.factor(hs_df$Geographic.Area)
hs_df$percent_completed_hs <- as.numeric(hs_df$percent_completed_hs)
```

```r
#aggregate the hs_df as a data frame and covert the names of the columns
hs_table <- aggregate( x = hs_df$percent_completed_hs,
                       by = list(hs_df$Geographic.Area),
                       FUN = median)

hs_table <- as.data.frame(hs_table)
hs_table <- rename(hs_table, "State" = "Group.1")
hs_table <- rename(hs_table, "Over.25.Grad.Rate" = "x")

#view the object
hs_table

##    State Over.25.Grad.Rate
## 1     AK             88.00
## 2     AL             81.15
## 3     AR             81.10
## 4     AZ             84.25
## 5     CA             87.50
## 6     CO             92.35
## 7     CT             93.20
## 8     DC             89.30
## 9     DE             89.50
## 10    FL             88.40
## 11    GA             79.30
## 12    HI             92.50
## 13    IA             91.10
## 14    ID             87.50
## 15    IL             89.80
## 16    IN             86.90
## 17    KS             90.00
## 18    KY             82.45
## 19    LA             80.00
## 20    MA             93.90
## 21    MD             91.10
## 22    ME             91.70
## 23    MI             89.90
## 24    MN             90.90
## 25    MO             85.35
## 26    MS             78.30
## 27    MT             91.80
## 28    NC             83.60
## 29    ND             90.00
## 30    NE             91.00
## 31    NH             91.90
## 32    NJ             92.60
## 33    NM             84.50
## 34    NV             89.90
## 35    NY             92.00
## 36    OH             89.60
```

```
## 37    OK              83.80
## 38    OR              89.75
## 39    PA              90.30
## 40    RI              91.25
## 41    SC              81.75
## 42    SD              90.10
## 43    TN              82.00
## 44    TX              80.40
## 45    UT              93.15
## 46    VA              86.00
## 47    VT              90.30
## 48    WA              91.60
## 49    WI              91.20
## 50    WV              84.00
## 51    WY              93.70
```

Join the developed data frame data onto the police shootings data

```
#develop the final_df object from the police shootings and left joined data
from the developed objects
final_df <- left_join(police_post2015, pr_table, by = c("state" = "State"))
final_df <- left_join(final_df, income_table, by = c("state" = "State"))
final_df <- left_join(final_df, hs_table, by = c("state" = "State"))

#create the regional column data frame
head(final_df)
```

```
##   id               name       date  manager_of_death      armed age gender
race
## 1  3        Tim Elliot 2015-01-02             shot         gun  53      M
A
## 2  4   Lewis Lee Lembke 2015-01-02             shot         gun  47      M
W
## 3  5 John Paul Quintero 2015-01-03 shot and Tasered    unarmed  23      M
H
## 4  8    Matthew Hoffman 2015-01-04             shot toy weapon  32      M
W
## 5  9  Michael Rodriguez 2015-01-04             shot   nail gun  39      M
H
## 6 11  Kenneth Joe Brown 2015-01-04             shot         gun  18      M
W
##             city state signs_of_mental_illness threat_level        flee
## 1       Shelton    WA                    True       attack Not fleeing
## 2         Aloha    OR                   False       attack Not fleeing
## 3       Wichita    KS                   False        other Not fleeing
## 4 San Francisco    CA                    True       attack Not fleeing
## 5         Evans    CO                   False       attack Not fleeing
## 6       Guthrie    OK                   False       attack Not fleeing
##   body_camera longitude latitude is_geocoding_exact Median.Below.Poverty
## 1       False  -123.122   47.247               True                12.30
## 2       False  -122.892   45.487               True                16.20
```

```
## 3        False    -97.281   37.695               True                 12.80
## 4        False   -122.422   37.763               True                 13.40
## 5        False   -104.692   40.384               True                 11.55
## 6        False    -97.423   35.877               True                 18.80
##   Median.Income Over.25.Grad.Rate
## 1       45013.0             91.60
## 2       43125.0             89.75
## 3       42500.0             90.00
## 4       54667.0             87.50
## 5       50220.5             92.35
## 6       37896.0             83.80
```

Add in a region area by state (grouoped state data)

```
final_df <- final_df %>% mutate(Region =
                     case_when(state == 'AL' ~ 'Southeast',
                               state == 'AK' ~ 'West',
                               state == 'AZ' ~ 'Southwest',
                               state == 'AR' ~ 'Southeast',
                               state == 'CA' ~ 'West',
                               state == 'CO' ~ 'West',
                               state == 'CT' ~ 'Northeast',
                               state == 'DE' ~ 'Northeast',
                               state == 'DC' ~ 'Southeast',
                               state == 'FL' ~ 'Southeast',
                               state == 'GA' ~ 'Southeast',
                               state == 'GU' ~ 'West',
                               state == 'HI' ~ 'West',
                               state == 'ID' ~ 'West',
                               state == 'IL' ~ 'Midwest',
                               state == 'IN' ~ 'Midwest',
                               state == 'IA' ~ 'Midwest',
                               state == 'KS' ~ 'Midwest',
                               state == 'KY' ~ 'Southeast',
                               state == 'LA' ~ 'Southeast',
                               state == 'ME' ~ 'Northeast',
                               state == 'MD' ~ 'Northeast',
                               state == 'MA' ~ 'Northeast',
                               state == 'MI' ~ 'Midwest',
                               state == 'MN' ~ 'Midwest',
                               state == 'MS' ~ 'Southeast',
                               state == 'MO' ~ 'Midwest',
                               state == 'MT' ~ 'West',
                               state == 'NE' ~ 'Midwest',
                               state == 'NV' ~ 'West',
                               state == 'NH' ~ 'Northeast',
                               state == 'NJ' ~ 'Northeast',
                               state == 'NM' ~ 'Southwest',
                               state == 'NY' ~ 'Northeast',
                               state == 'NC' ~ 'Southeast',
                               state == 'ND' ~ 'Midwest',
```

```r
                                 state == 'OH' ~ 'Midwest',
                                 state == 'OK' ~ 'Southwest',
                                 state == 'OR' ~ 'West',
                                 state == 'PA' ~ 'Northeast',
                                 state == 'PR' ~ 'Southeast',
                                 state == 'RI' ~ 'Northeast',
                                 state == 'SC' ~ 'Southeast',
                                 state == 'SD' ~ 'Midwest',
                                 state == 'TN' ~ 'Southeast',
                                 state == 'TX' ~ 'Southwest',
                                 state == 'UT' ~ 'West',
                                 state == 'VA' ~ 'Southeast',
                                 state == 'VT' ~ 'Northeast',
                                 state == 'WA' ~ 'West',
                                 state == 'WV' ~ 'Southeast',
                                 state == 'WI' ~ 'Midwest',
                                 state == 'WY' ~ 'West'))
```

Add in an Armed Flag attribute to the final dataframe

```r
final_df <- final_df %>% mutate(Armed.Flag =
                     case_when(armed == 'undertermed' ~ '0',
                               armed == 'unarmed' ~ '0',
                               armed == 'NA' ~ '0'))
#repalce all NA's in the Armed.Flag with a 1 flag
final_df[is.na(final_df)] <- 1
```

Add in an Is.Minority Flag for classification prediction modeling to the final dataframe

```r
#develop an attribute that depicts if a person is a minority or not
final_df <- final_df %>% mutate(Is.Minority =
                                  case_when(race == 'W' ~ '0'))
final_df[is.na(final_df)] <- '1'

#display a contigency table to review that the output of the above mutation
is correct
a <- table(final_df$race, final_df$Is.Minority)
a

##
##         0    1
##         0  882
##    A    0  106
##    B    0 1555
##    H    0 1085
##    N    0   91
##    O    0   47
##    W 2969    0
```

View the output of the final dataframe prior to EDA

```r
#view the output of the final dataframe
head(final_df)
```

```
##   id               name       date  manner_of_death      armed age gender
race
## 1  3        Tim Elliot 2015-01-02             shot        gun  53      M
A
## 2  4   Lewis Lee Lembke 2015-01-02             shot        gun  47      M
W
## 3  5 John Paul Quintero 2015-01-03 shot and Tasered    unarmed  23      M
H
## 4  8    Matthew Hoffman 2015-01-04             shot toy weapon  32      M
W
## 5  9  Michael Rodriguez 2015-01-04             shot   nail gun  39      M
H
## 6 11  Kenneth Joe Brown 2015-01-04             shot        gun  18      M
W
##            city state signs_of_mental_illness threat_level        flee
## 1       Shelton    WA                    True       attack Not fleeing
## 2         Aloha    OR                   False       attack Not fleeing
## 3       Wichita    KS                   False        other Not fleeing
## 4 San Francisco    CA                    True       attack Not fleeing
## 5         Evans    CO                   False       attack Not fleeing
## 6       Guthrie    OK                   False       attack Not fleeing
##   body_camera longitude latitude is_geocoding_exact Median.Below.Poverty
## 1       False  -123.122   47.247               True                12.30
## 2       False  -122.892   45.487               True                16.20
## 3       False   -97.281   37.695               True                12.80
## 4       False  -122.422   37.763               True                13.40
## 5       False  -104.692   40.384               True                11.55
## 6       False   -97.423   35.877               True                18.80
##   Median.Income Over.25.Grad.Rate    Region Armed.Flag Is.Minority
## 1       45013.0             91.60      West          1           1
## 2       43125.0             89.75      West          1           0
## 3       42500.0             90.00   Midwest          0           1
## 4       54667.0             87.50      West          1           0
## 5       50220.5             92.35      West          1           1
## 6       37896.0             83.80 Southwest          1           0
```

Begin Exploratory Data Analysis

```
summary(final_df)

##        id           name               date           manner_of_death   
##  Min.   :   3   Length:6735        Length:6735        Length:6735       
##  1st Qu.:1898   Class :character   Class :character   Class :character  
##  Median :3737   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :3727                                                           
##  3rd Qu.:5554                                                           
##  Max.   :7347                                                           
##     armed                age           gender              race          
##  Length:6735        Min.   : 1.00   Length:6735        Length:6735       
##  Class :character   1st Qu.:26.00   Class :character   Class :character  
##  Mode  :character   Median :34.00   Mode  :character   Mode  :character  
```

```
##                     Mean   :35.36
##                     3rd Qu.:45.00
##                     Max.   :92.00
##      city               state             signs_of_mental_illness
##  Length:6735        Length:6735        Length:6735
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  threat_level           flee            body_camera           longitude
##  Length:6735        Length:6735        Length:6735        Min.    :-160.01
##  Class :character   Class :character   Class :character   1st Qu.:-111.91
##  Mode  :character   Mode  :character   Mode  :character   Median : -92.85
##                                                           Mean   : -92.44
##                                                           3rd Qu.: -82.08
##                                                           Max.   :   1.00
##     latitude     is_geocoding_exact Median.Below.Poverty Median.Income
##  Min.   : 1.00   Length:6735        Min.   : 6.40        Min.   :31800
##  1st Qu.:32.86   Class :character   1st Qu.:12.30        1st Qu.:38304
##  Median :35.77   Mode  :character   Median :15.00        Median :43359
##  Mean   :34.96                      Mean   :15.49        Mean   :45278
##  3rd Qu.:39.89                      3rd Qu.:18.80        3rd Qu.:50220
##  Max.   :71.30                      Max.   :26.45        Max.   :75358
##  Over.25.Grad.Rate    Region           Armed.Flag         Is.Minority
##  Min.   :78.30     Length:6735        Length:6735        Length:6735
##  1st Qu.:82.45     Class :character   Class :character   Class :character
##  Median :87.50     Mode  :character   Mode  :character   Mode  :character
##  Mean   :86.46
##  3rd Qu.:89.90
##  Max.   :93.90
```

Fleeing Contigency Tables by race

```
#table for armed
armed_table <- table(final_df$Armed.Flag, final_df$race)
armed_table
```

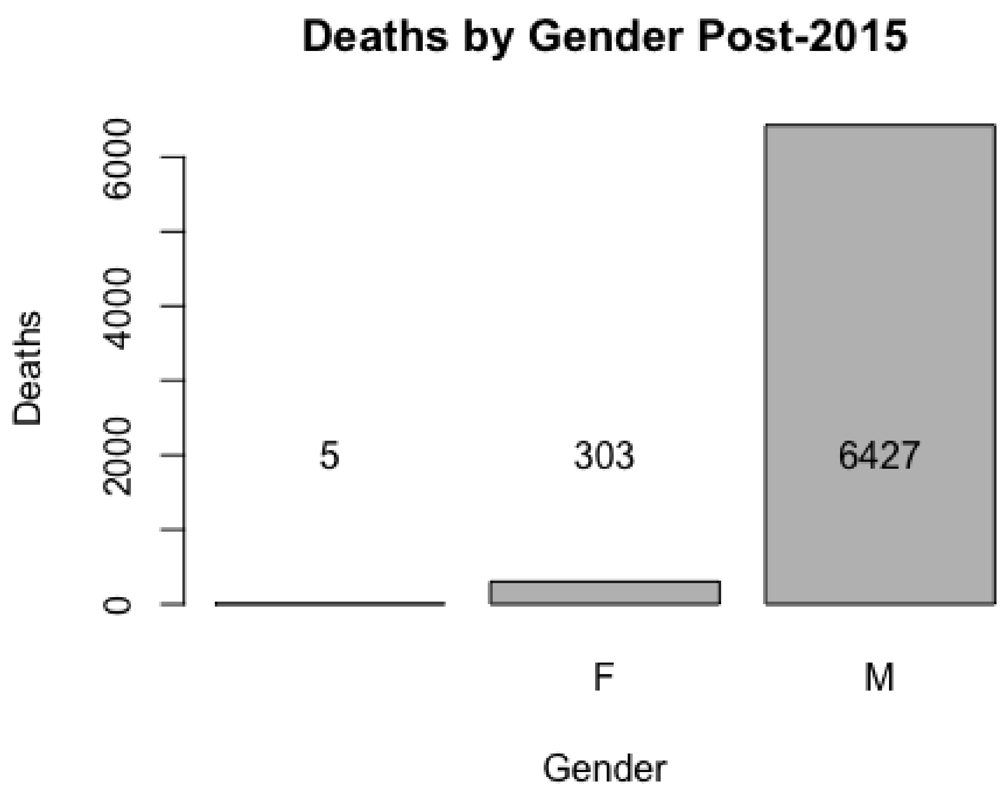
```
##
##             A    B    H    N    O    W
##   0   16    8  137   79    6    5  175
##   1  866   98 1418 1006   85   42 2794
```

```
#view the proportionas of the armed_table
round(prop.table(armed_table, margin = 2)*100,1)
```

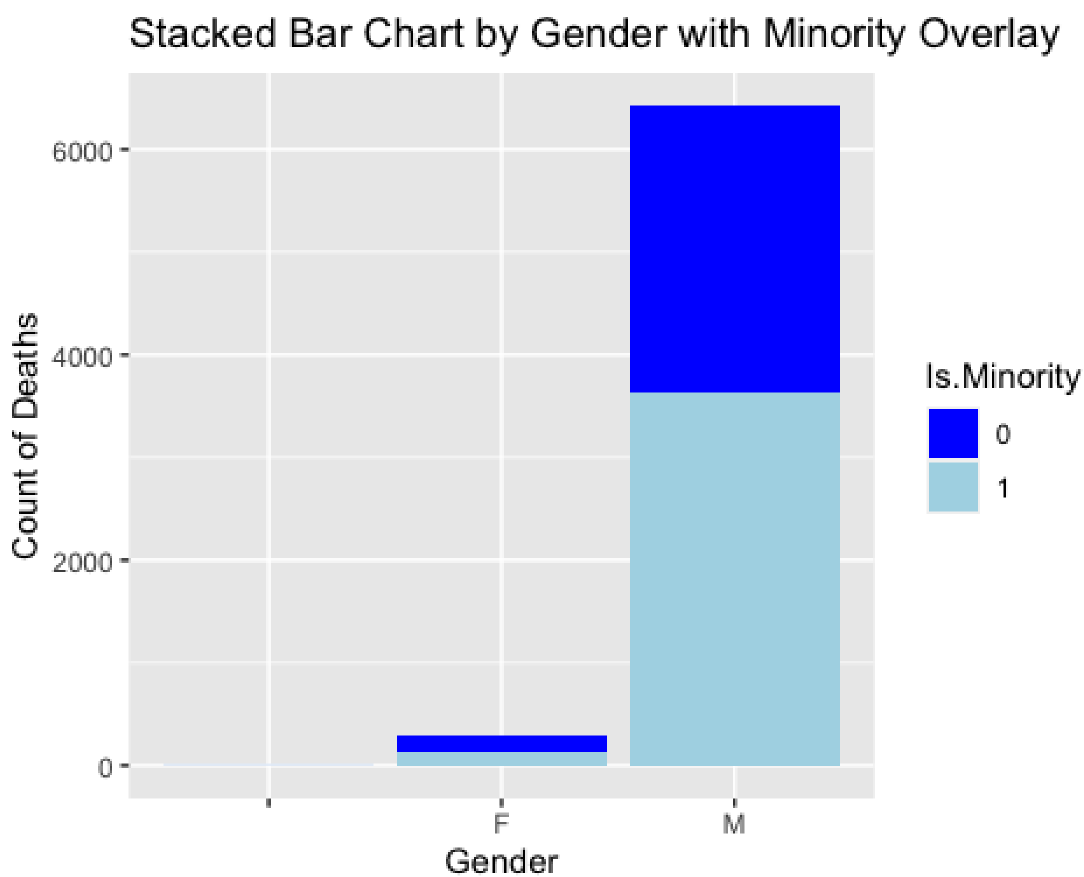
```
##
##             A    B    H    N    O    W
##   0  1.8  7.5  8.8  7.3  6.6 10.6  5.9
##   1 98.2 92.5 91.2 92.7 93.4 89.4 94.1
```

Deaths by Gender bar chart

```
gender_summary_post <- table(police_post2015$gender)

gender_post <- barplot(gender_summary_post[order(gender_summary_post,
                                                  decreasing = FALSE)],
                main = "Deaths by Gender Post-2015",
                xlab = 'Gender',
                ylab = 'Deaths')
text(gender_post,  + 2000 , gender_summary_post, font=1)
```



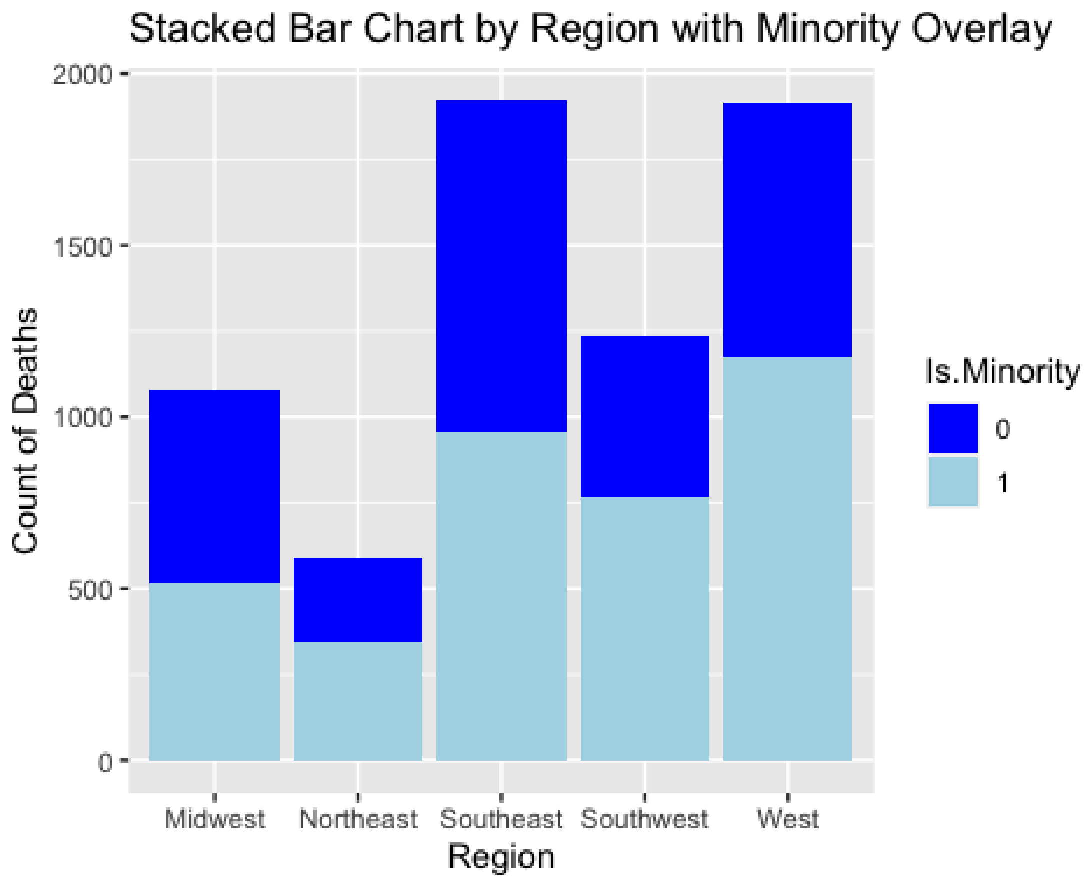

```
#police shootings by gender bar chart using ggplot with minority overlay
ggplot(final_df, aes(gender)) + geom_bar(aes(fill= Is.Minority)) +
  ggtitle("Stacked Bar Chart by Gender with Minority Overlay") +
  scale_fill_manual(values = c("blue","lightblue")) + labs(x="Gender",
                                                           y="Count of
Deaths")
```



####

Deaths by Region

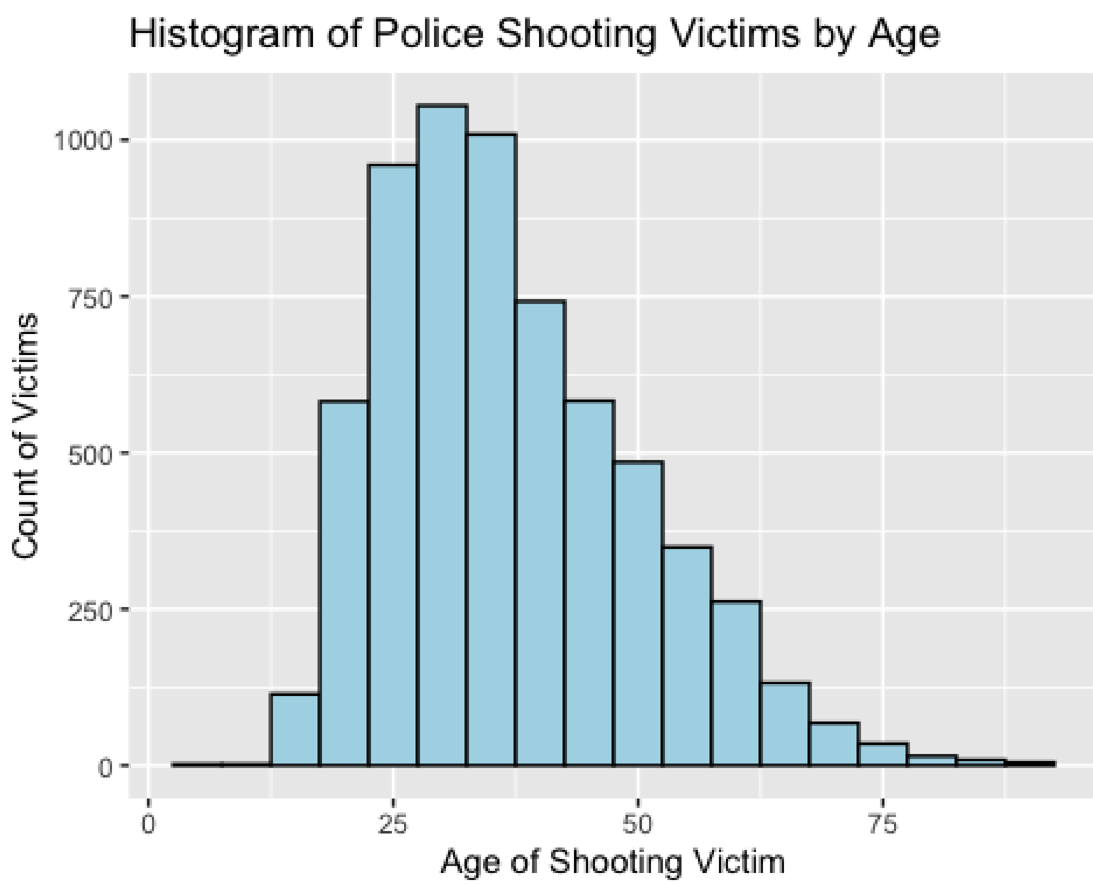
```
#contingency table to view the deaths by region
cont_table_region <- table(final_df$Region)
cont_table_region

##
##   Midwest Northeast Southeast Southwest      West
##      1078       588      1922      1234      1913

#bar chart with deaths by region with minority overlay using ggplot
ggplot(final_df, aes(Region)) + geom_bar(aes(fill= Is.Minority)) +
  ggtitle("Stacked Bar Chart by Region with Minority Overlay") +
  scale_fill_manual(values = c("blue","lightblue")) + labs(x="Region",
                                                           y="Count of
Deaths")
```




```
region_summary_table <- table(final_df$Region)
```

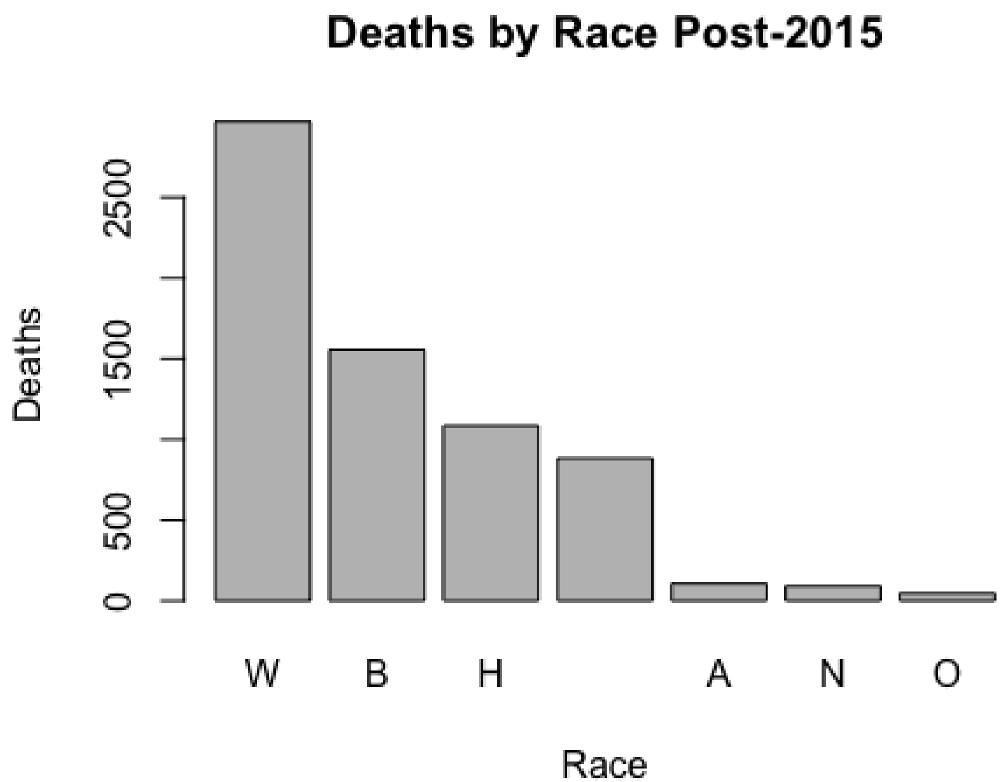
Deaths by Age histogram

```
#histogram of age post-2015
ggplot(data = police_post2015, aes(age)) +
  geom_histogram(binwidth = 5, color='black',fill ='lightblue') +
  ggtitle("Histogram of Police Shooting Victims by Age") +
  labs(x ='Age of Shooting Victim', y = 'Count of Victims')

## Warning: Removed 326 rows containing non-finite values (stat_bin).
```



Deaths by Race bar chart

```
#create a table of the deaths by race
race_summary_post <- table(police_post2015$race)

#develop the bar chart in decreasing order
race_post <- barplot(race_summary_post[order(race_summary_post,
                                             decreasing = TRUE)],
               main = "Deaths by Race Post-2015",
               xlab = 'Race',
               ylab = 'Deaths')
```




```
# W = White, B = Black, H = Hispanic, A = Asian,
# Empty = Unknown, N = Native American, O = Other

#return the vector of only the deaths by race category
race_summary_post

##
##          A    B    H    N    O    W
##   882  106 1555 1085   91   47 2969
```

Develop contingency tables of police shootings by race, and associated proportions/percentages of whole

```
#contingency tables and percentages of shooting by race
cont_table_race <- table(police_post2015$race)
prop_table_race <- prop.table(cont_table_race)
perc_table_race <- prop.table(cont_table_race) * 100

race_table <- rbind(cont_table_race, prop_table_race, perc_table_race)
rownames(race_table) <- c("Count","Proportion","Percentage")
race_table

##                                   A            B            H            N
## Count      882.0000000 106.00000000 1555.0000000 1085.0000000 91.00000000
## Proportion   0.1309577   0.01573868    0.2308834    0.1610987  0.01351151
```

```
## Percentage  13.0957684   1.57386785   23.0883445   16.1098738  1.35115071
##                       O            W
## Count      47.000000000 2969.0000000
## Proportion  0.006978471    0.4408315
## Percentage  0.697847068   44.0831477

#race_table["Count", "B"]
```

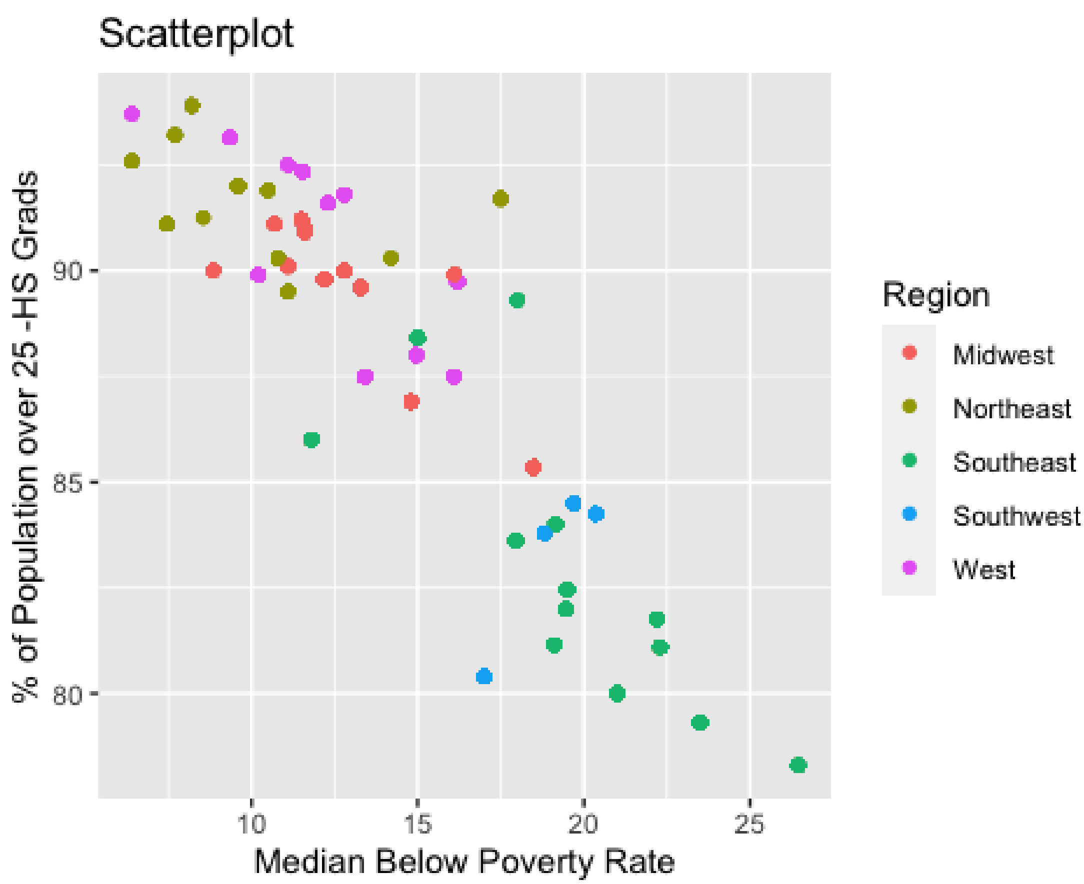
From the total U.S. Population statistics in 2019, develop the race proporitions of the U.S. and determine the associated distributions of police shootings by race relative to race proportion in the U.S.

```
#from: https://www.visualcapitalist.com/visualizing-u-s-population-by-race/ -
- retrieve U.S. Population and Demographic data

#estimated U.S. Populations as of 2019
total_pop <- 328239523

#estimated U.S. race demographic proportions
white_pop <- .601 * total_pop
black_pop <- .122 *total_pop
hisp_pop <- .185 *total_pop
asian_pop <- .056 * total_pop
other_pop <- 100 - white_pop -black_pop - hisp_pop - asian_pop

#develop an object by race of the count of deaths by the population
proportion
white_prop <- (race_table["Count","W"] / white_pop) * 100
black_prop <- (race_table["Count","B"] / black_pop) * 100
hisp_prop <-  (race_table["Count","H"] / hisp_pop)  * 100
asian_prop <- (race_table["Count","A"] / asian_pop) * 100

#print the developed race proportions of deaths by police shooting
print(black_prop)

## [1] 0.00388311

print(hisp_prop)

## [1] 0.001786764

print(white_prop)

## [1] 0.001505029

print(asian_prop)

## [1] 0.0005766695
```

Scatterplot of HS Grad Rate, Median Below Povery by Region

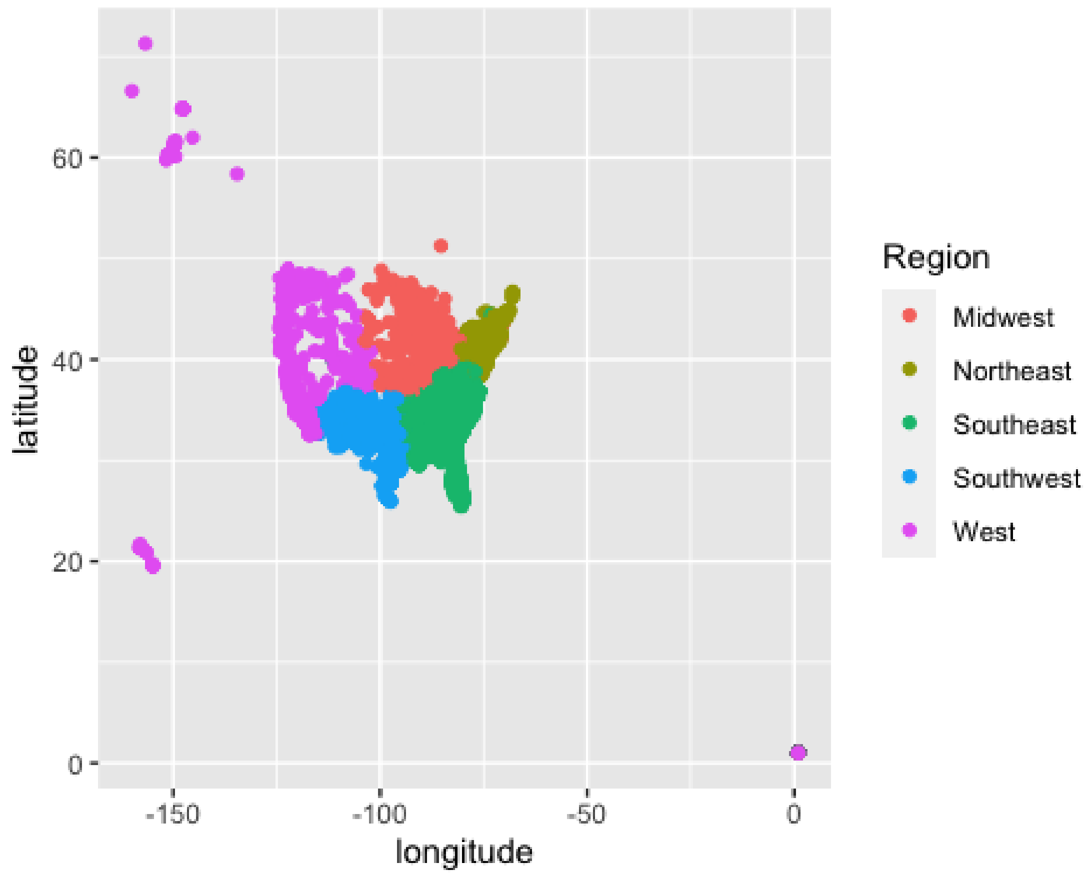
```
ggplot(data=final_df) +
  geom_point(mapping = aes( x = Median.Below.Poverty,
```

```
                               y = Over.25.Grad.Rate, color = Region)) +
  ggtitle("Scatterplot") + xlab("Median Below Poverty Rate") +
  ylab("% of Population over 25 -HS Grads")
```



Scatterplot of Lat $ Long by Region

```
ggplot(data=final_df) +
  geom_point(mapping = aes( x = longitude, y = latitude, color = Region))
```



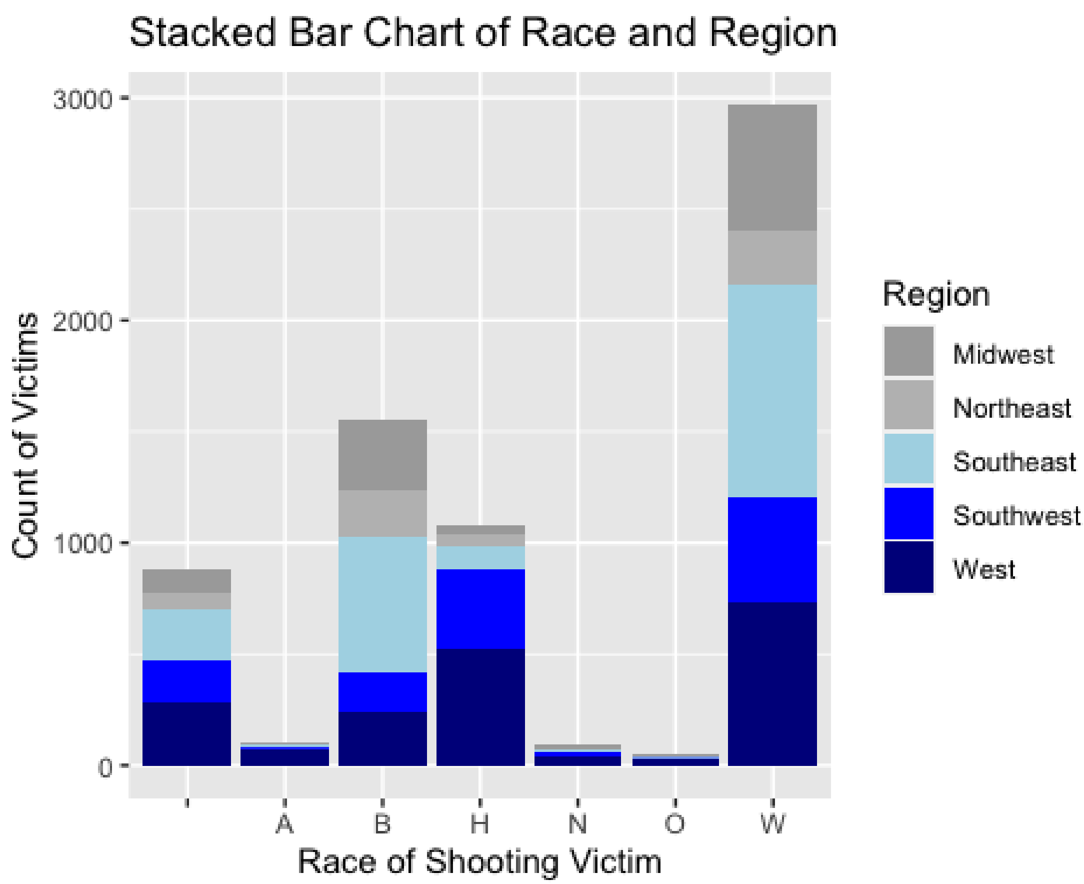
Race and Region Contigency Tables

```
#count of race deaths by regions
race_region_cont <- table(final_df$race,final_df$Region)
race_region_cont

##
##      Midwest Northeast Southeast Southwest West
##          103        80       232       188  279
##    A      10         4        15         9   68
##    B     319       212       602       181  241
##    H      52        50       101       358  524
##    N      23         1         3        27   37
##    O       7         2         7         3   28
##    W     564       239       962       468  736

#proportion of race deaths by regions
round(prop.table(race_region_cont, margin = 2)*100,1)

##
##      Midwest Northeast Southeast Southwest West
##          9.6      13.6      12.1      15.2 14.6
##    A     0.9       0.7       0.8       0.7  3.6
##    B    29.6      36.1      31.3      14.7 12.6
```

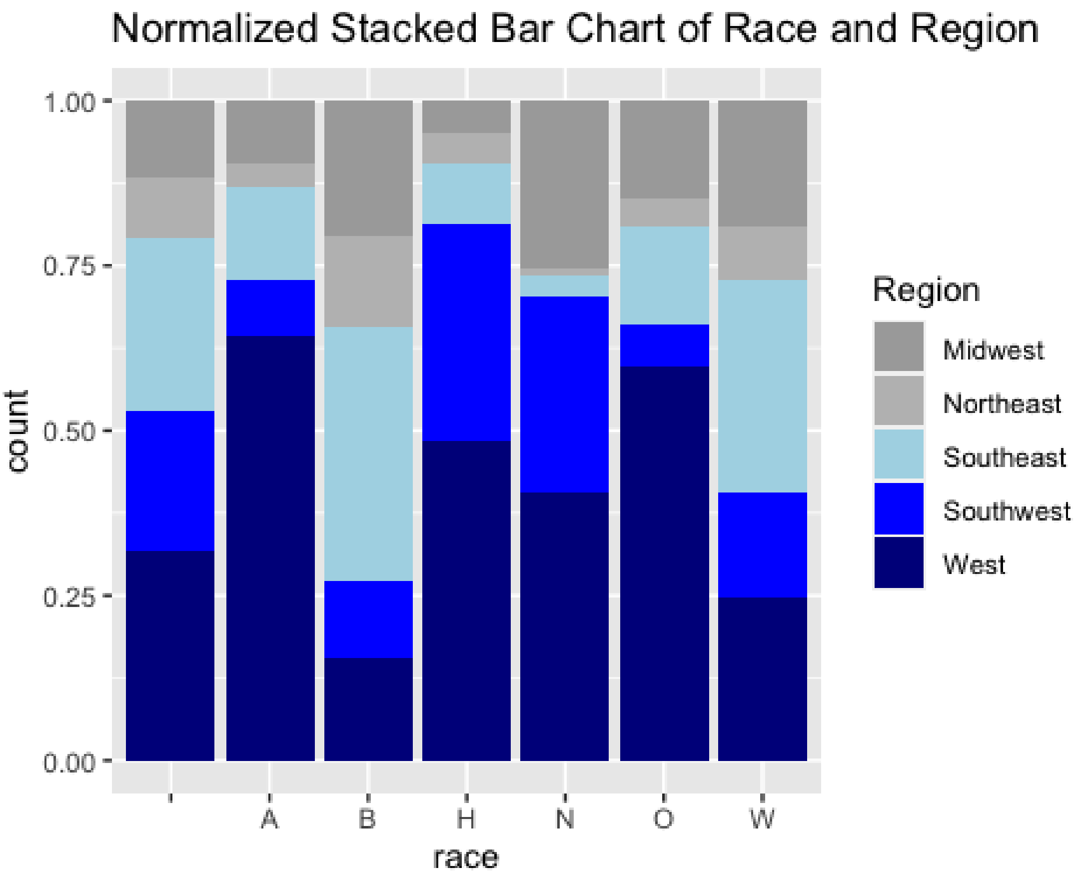
```
##    H      4.8        8.5        5.3        29.0 27.4
##    N      2.1        0.2        0.2         2.2  1.9
##    O      0.6        0.3        0.4         0.2  1.5
##    W     52.3       40.6       50.1        37.9 38.5
```

Bar Chart of Total Police Shootings with Race Overlay

```
ggplot(final_df, aes(race)) + geom_bar(aes(fill=Region)) +
  ggtitle("Stacked Bar Chart of Race and Region") +
scale_fill_manual(values=c("darkgrey","grey","lightblue","blue","darkblue"))
+
  labs( x = 'Race of Shooting Victim' , y='Count of Victims')
```



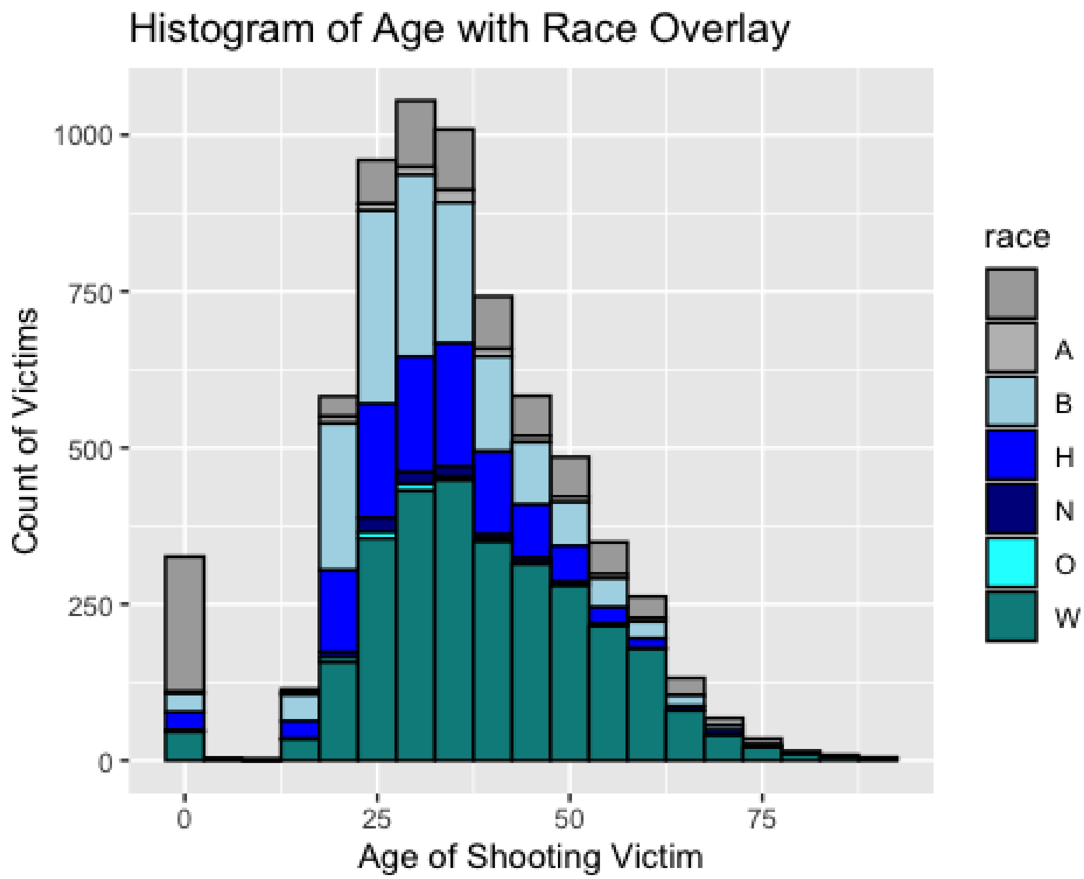

```
#normalized bar chart
ggplot(final_df, aes(race)) + geom_bar(aes(fill=Region) , position = "fill")
+
  ggtitle("Normalized Stacked Bar Chart of Race and Region") +
scale_fill_manual(values=c("darkgrey","grey","lightblue","blue","darkblue"))
```



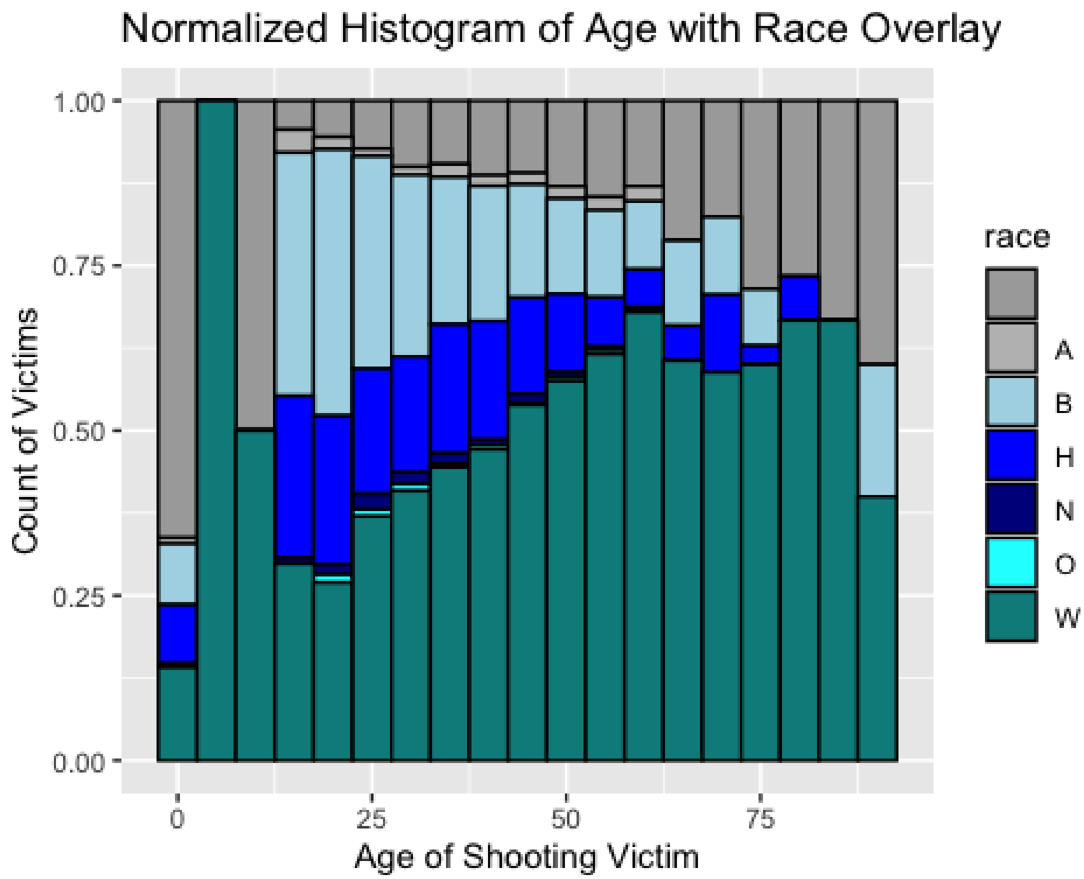
####

Histograms of Age with Race Overlay

```
#histogram of age with race underlay
ggplot(final_df, aes(age)) + geom_histogram(aes(fill=race), color="black",
binwidth = 5) +
  ggtitle("Histogram of Age with Race Overlay") +labs( x = 'Age of Shooting
Victim' , y='Count of Victims') +
scale_fill_manual(values=c("darkgrey","grey","lightblue","blue","darkblue","c
yan","cyan4"))
```



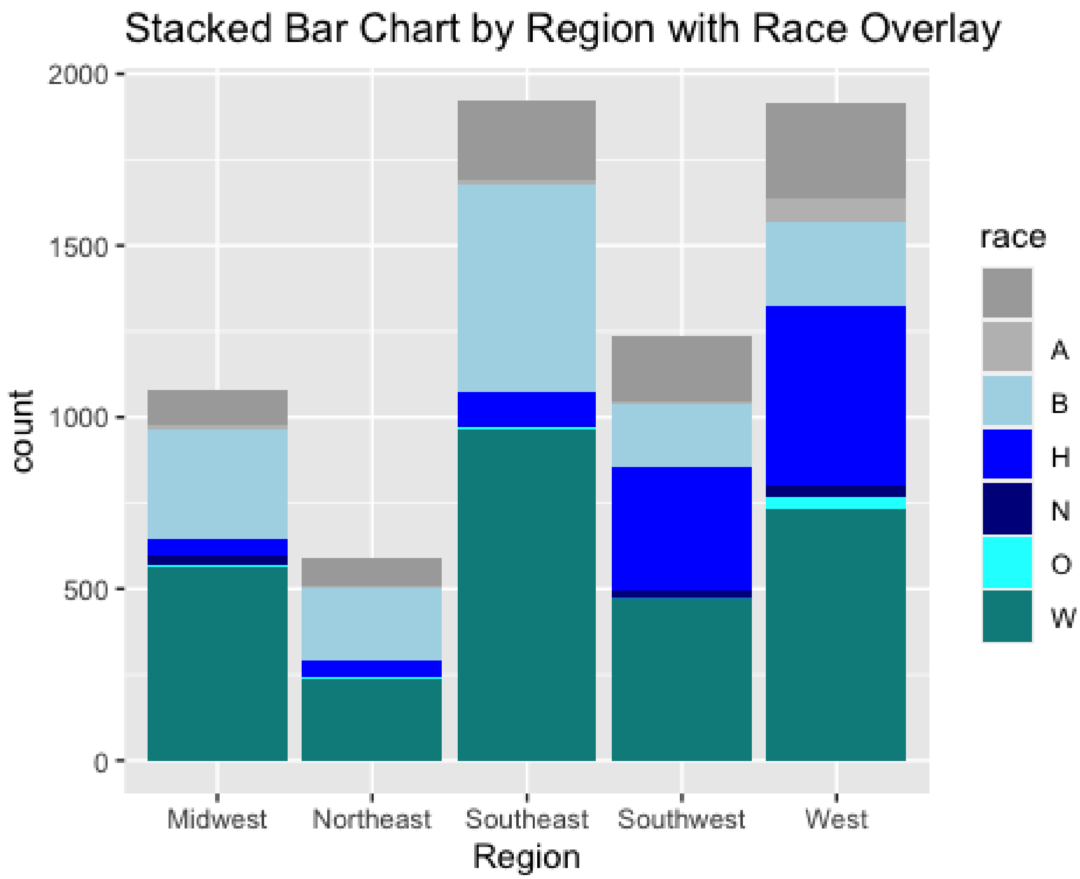

```
#histogram of age with race underlay
ggplot(final_df, aes(age)) + geom_histogram(aes(fill=race), color="black",
                                             binwidth = 5, position = "fill")
+
  ggtitle("Normalized Histogram of Age with Race Overlay") +labs( x = 'Age of
Shooting Victim' , y='Count of Victims') +
scale_fill_manual(values=c("darkgrey","grey","lightblue","blue","darkblue","c
yan","cyan4"))
```



####

Stacked Bar Chart of Deahts by Region with Race Overlay

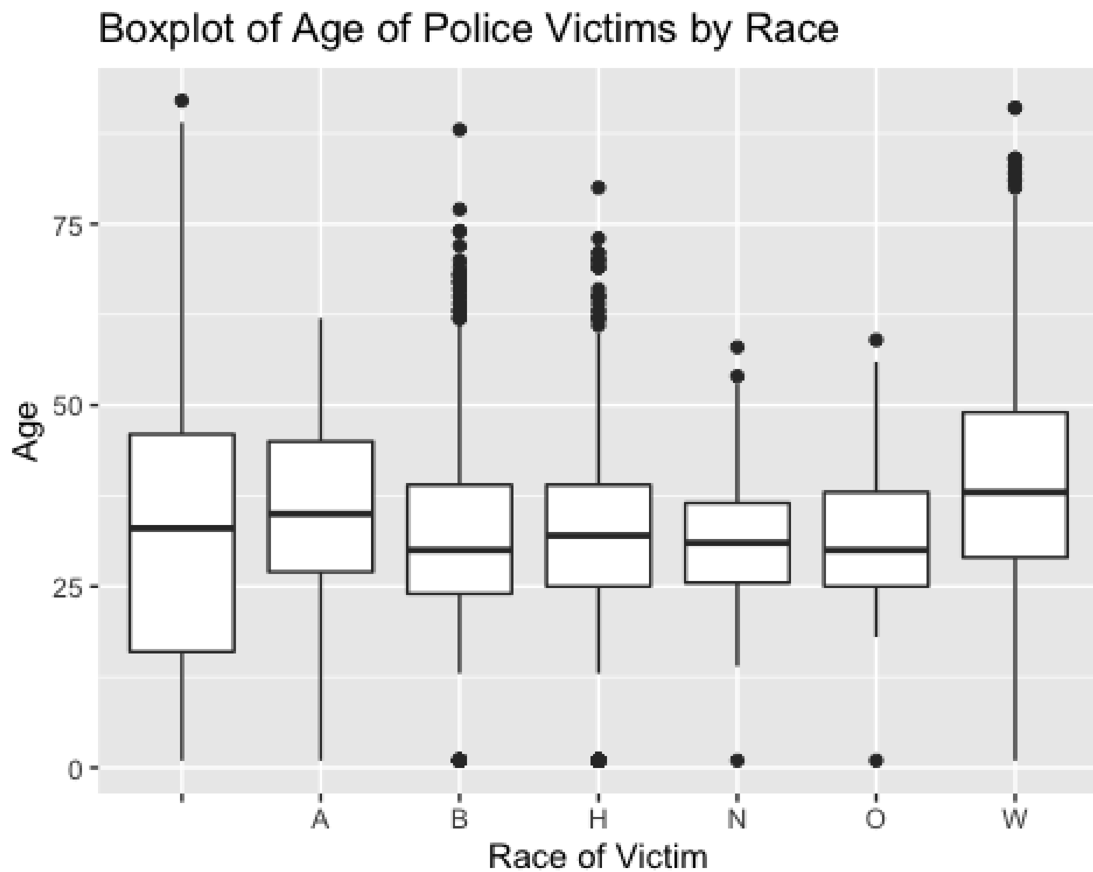
```
ggplot(final_df, aes(Region)) + geom_bar(aes(fill=race)) + ggtitle("Stacked
Bar Chart by Region with Race Overlay") +
scale_fill_manual(values=c("darkgrey","grey","lightblue","blue","darkblue","c
yan","cyan4"))
```



####

Boxplot of Age & Race

```
ggplot(data =final_df, mapping = aes(x=race, y = age)) + geom_boxplot() +
  ggtitle("Boxplot of Age of Police Victims by Race") + labs(x="Race of
Victim", y="Age")
```



#####

End EDA

Begin Machine Learning Models

Partition the data

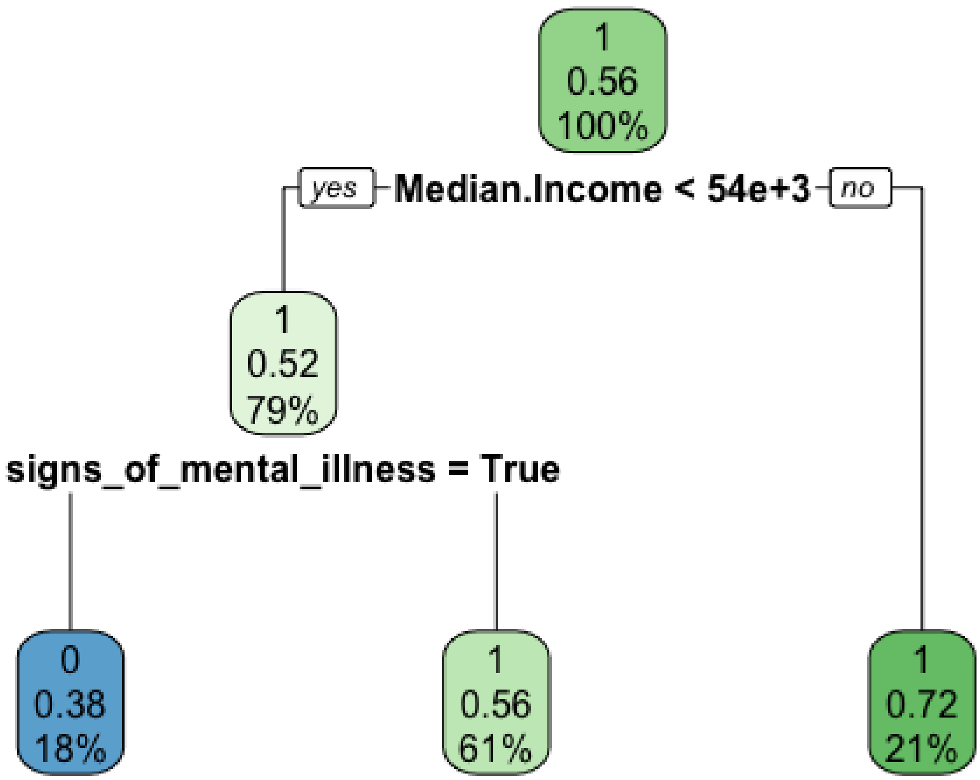
```
#partition the data - set seed for the random number generator
set.seed(7)

#return how many records are in the data set
n <- dim(final_df)[1]
n

## [1] 6735

training_index <- runif(n) < 0.75

shootings_train <- final_df[training_index,]
shootings_test <- final_df[!training_index,]

#validate the data has been partitioned into two data sets - a training of
0.75 and test of 0.25
dim(shootings_train)

## [1] 5058   23
```

```
dim(shootings_test)

## [1] 1677   23
```

CART Decision Tree Algorithm

```
#develop two data frames for the CART Decision Tree Algorithm from the
original dataframes
cart_training <- shootings_train
cart_test <- shootings_test

#set categorical variables to factors (training)
cart_training$signs_of_mental_illness <-
factor(cart_training$signs_of_mental_illness)
cart_training$Region <- factor(cart_training$Region)
cart_training$Armed.Flag <- factor(cart_training$Armed.Flag)
#cart_training$Median.Below.Poverty <-
factor(cart_training$Median.Below.Poverty)
#cart_training$Median.Income <- factor(cart_training$Median.Income)
#cart_training$Over.25.Grad.Rate <- factor(cart_training$Over.25.Grad.Rate)

#set categorical variables to factors (test)
cart_test$signs_of_mental_illness <-
factor(cart_test$signs_of_mental_illness)
cart_test$Region <- factor(cart_test$Region)
cart_test$Armed.Flag <- factor(cart_test$Armed.Flag)
#cart_test$Median.Below.Poverty <- factor(cart_test$Median.Below.Poverty)
#cart_test$Median.Income <- factor(cart_test$Median.Income)
#cart_test$Over.25.Grad.Rate <- factor(cart_test$Over.25.Grad.Rate)

#import the C5.0 algorithm library
library(rpart)
library(rpart.plot)

#develop the CART algorithm
cart01 <- rpart(formula = Is.Minority ~ signs_of_mental_illness + Region +
Armed.Flag +
                  Median.Below.Poverty + Median.Income + Over.25.Grad.Rate,
data = cart_training,
                method = "class")

#apply the cart01 model to the test dataset
predict_race = predict(object = cart01, newdata = cart_test, type ="class")

#develop a contingency table of the predicted and actual races of the CART
algorithm
cart_contingency <- table(cart_test$Is.Minority , predict_race)
colnames(cart_contingency) <- c("Predicted No", "Predicted Yes")
row.names(cart_contingency) <- c("Actual No", "Actual Yes")
cart_contingency
```

```
##              predict_race
##               Predicted No Predicted Yes
##   Actual No            169           583
##   Actual Yes           111           814

#plot the CART Algorithm
rpart.plot(cart01)
```

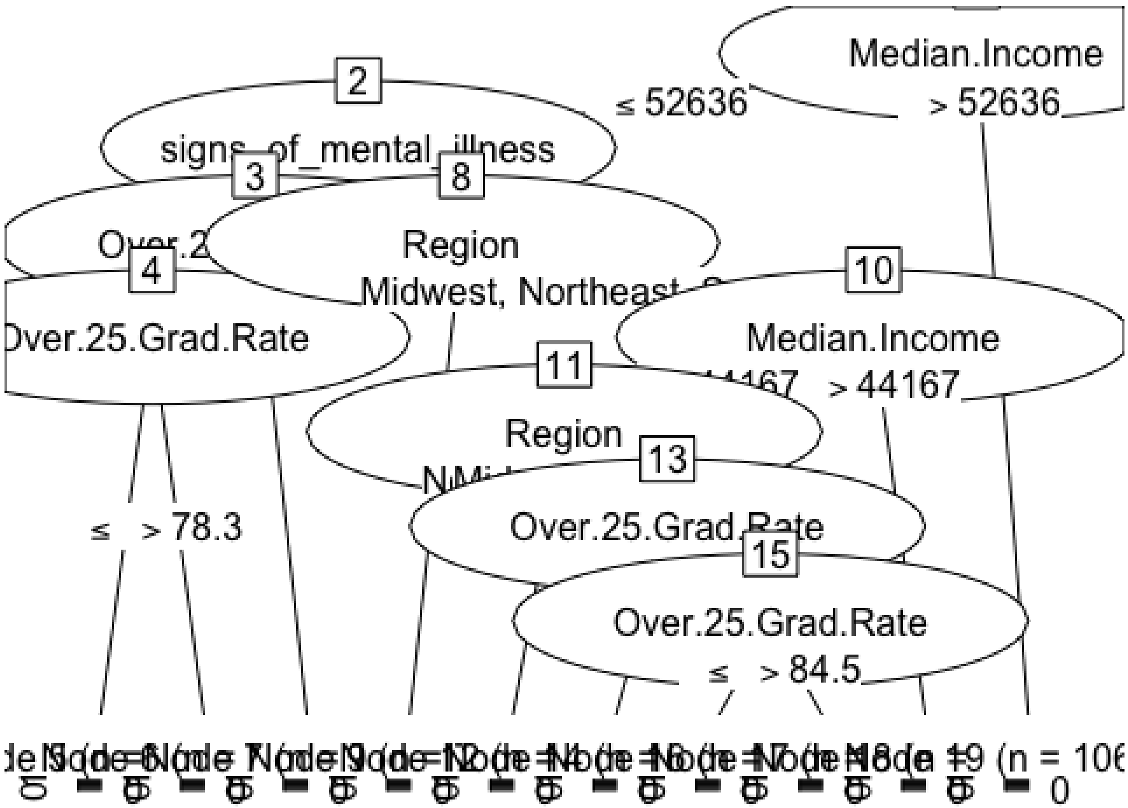


###

C5.0 Algorithm

```
#assign the data sets
c50_train <- cart_training
c50_test <- cart_test

#turn the outcome variable into a factor
c50_train$Is.Minority <- factor(c50_train$Is.Minority)
c50_test$Is.Minority <- factor(c50_test$Is.Minority)

#import the C5.0 algorithm library
library(C50)

#develop the C5.0 algorithm
C5 <- C5.0(formula = Is.Minority ~ signs_of_mental_illness + Region +
Armed.Flag +
             Median.Below.Poverty + Median.Income + Over.25.Grad.Rate, data =
```

```
c50_train,
           minCases=100)

#plot the C5 algorithm
plot(C5)
```

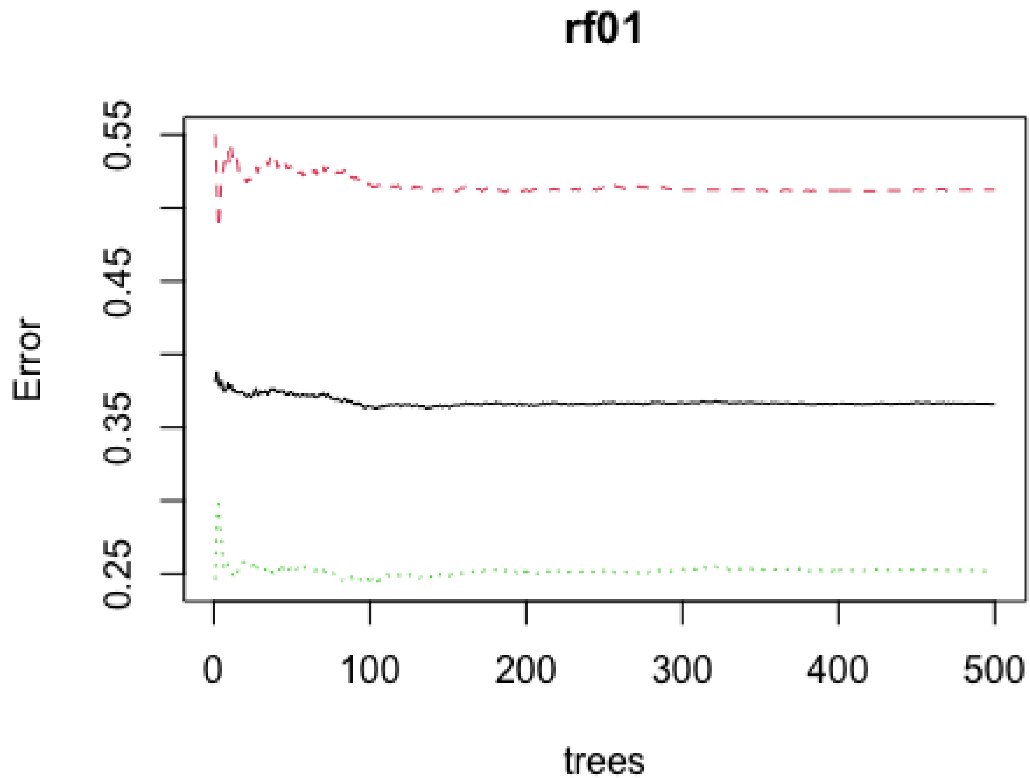



```
C5_predictions <- predict(object = C5, newdata = c50_test)

#develop a contingency table for the actual and predicted values
c5_contingency <- table(c50_test$Is.Minority, C5_predictions)
colnames(c5_contingency) <- c("Predicted No", "Predicted Yes")
row.names(c5_contingency) <- c("Actual No", "Actual Yes")
c5_contingency

##             C5_predictions
##              Predicted No Predicted Yes
##   Actual No           286           466
##   Actual Yes          153           772
```

Random Forests Algorithm

```
#import the randomForest library
library(randomForest)

## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

#prep the random forest data as necessary
rf_train <- c50_train
rf_test <- c50_test

#develop the random forests algorithm
rf01 <- randomForest(formula = Is.Minority ~ signs_of_mental_illness + Region
+ Armed.Flag +
                       Median.Below.Poverty + Median.Income +
Over.25.Grad.Rate, data = c50_train,
                     ntree = 500, type = "classification")

#predict the random forests
rf_predictions <- predict(object = rf01, newdata = rf_test)

#develop a contingency table for the actual and predicted values
rf_contingency <- table(rf_test$Is.Minority, rf_predictions)
colnames(rf_contingency) <- c("Predicted No", "Predicted Yes")
row.names(rf_contingency) <- c("Actual No", "Actual Yes")
rf_contingency

##             rf_predictions
##              Predicted No Predicted Yes
##   Actual No           354           398
##   Actual Yes          198           727

plot(rf01)
```



###

Navie Bayes Classifcation

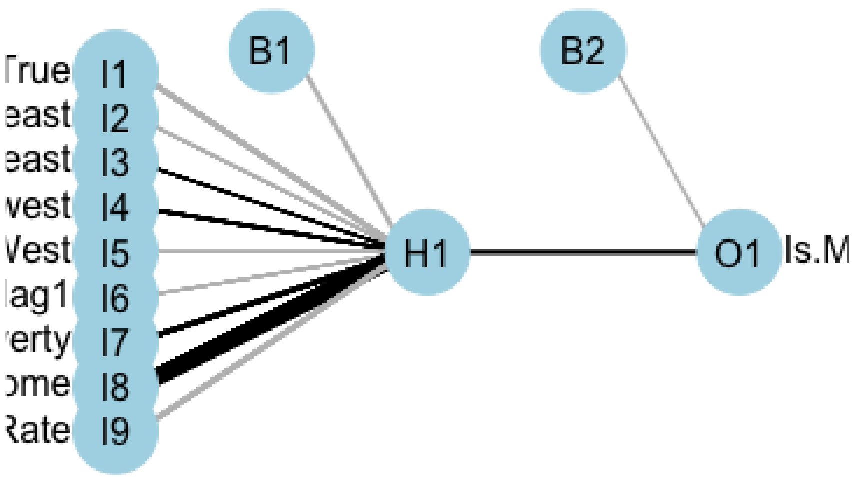
```
#import the Naive Bayes library
library(e1071)

#prep the Naive Bayes data as necessary
nb_train <- rf_train
nb_test <- rf_test

#develop the random forests algorithm
nb01 <- naiveBayes(formula = Is.Minority ~ signs_of_mental_illness + Region + 
Armed.Flag +
                     Median.Below.Poverty + Median.Income + 
Over.25.Grad.Rate,
                   data = nb_train)

#predict the naive bayes
nb_predictions <- predict(object=nb01, newdata = nb_test)

#develop a contingency table for the actual and predicted values
nb_contingency <- table(nb_test$Is.Minority , nb_predictions)
colnames(nb_contingency) <- c("Predicted No", "Predicted Yes")
row.names(nb_contingency) <- c("Actual No", "Actual Yes")
nb_contingency
```

```
##             nb_predictions
##              Predicted No Predicted Yes
##   Actual No           385           367
##   Actual Yes          311           614
```

Artificial Neural Network

```
#import the ANN library
library(nnet)
library(NeuralNetTools)

#prep the ANN data as necessary
ann_train <- nb_train
ann_test <- nb_test

#normalize the quantitative variables
ann_train$Median.Below.Poverty <- (ann_train$Median.Below.Poverty -
min(ann_train$Median.Below.Poverty)) /
  (max(ann_train$Median.Below.Poverty) - min(ann_train$Median.Below.Poverty))
ann_train$Median.Income <- (ann_train$Median.Income -
min(ann_train$Median.Income)) /
  (max(ann_train$Median.Income) - min(ann_train$Median.Income))
ann_train$Over.25.Grad.Rate <- (ann_train$Over.25.Grad.Rate -
min(ann_train$Over.25.Grad.Rate)) /
  (max(ann_train$Over.25.Grad.Rate) - min(ann_train$Over.25.Grad.Rate))

ann_test$Median.Below.Poverty <- (ann_test$Median.Below.Poverty -
min(ann_test$Median.Below.Poverty)) /
  (max(ann_test$Median.Below.Poverty) - min(ann_test$Median.Below.Poverty))
ann_test$Median.Income <- (ann_test$Median.Income -
min(ann_test$Median.Income)) /
  (max(ann_test$Median.Income) - min(ann_test$Median.Income))
ann_test$Over.25.Grad.Rate <- (ann_test$Over.25.Grad.Rate -
min(ann_test$Over.25.Grad.Rate)) /
  (max(ann_test$Over.25.Grad.Rate) - min(ann_test$Over.25.Grad.Rate))

#develop the neural network
nnet01 <- nnet(Is.Minority ~ signs_of_mental_illness + Region + Armed.Flag +
Median.Below.Poverty +
                 Median.Income + Over.25.Grad.Rate, data = ann_train, size =
1)

## # weights:  12
## initial  value 3567.898685
## iter  10 value 3341.969941
## iter  20 value 3305.609715
## iter  30 value 3305.223554
## iter  40 value 3305.221696
## iter  40 value 3305.221682
## iter  40 value 3305.221682
```

```
## final  value 3305.221682
## converged

#plot the neural net
plotnet(nnet01)
```




```
#obtain the weights of the ANN
nnet01$wts

##  [1] -1.8846373 -1.7242025 -0.3712136  0.1806484  0.8497956 -0.4215700
##  [7] -0.1480572  1.6969502  9.5397609 -1.7883501 -0.9377610  2.1263277

print(nnet01)

## a 9-1-1 network with 12 weights
## inputs: signs_of_mental_illnessTrue RegionNortheast RegionSoutheast
RegionSouthwest RegionWest Armed.Flag1 Median.Below.Poverty Median.Income
Over.25.Grad.Rate
## output(s): Is.Minority
## options were - entropy fitting

nn_contingency <- table(ann_test$Is.Minority , predict(nnet01, type =
"class", newdata = ann_test))
```

```
colnames(nn_contingency) <- c("Predicted No", "Predicted Yes")
row.names(nn_contingency) <- c("Actual No", "Actual Yes")
nn_contingency

##
##              Predicted No Predicted Yes
##   Actual No           312           440
##   Actual Yes          211           714
```

All Evaluation Metrics

```
#display all contingency matrices
#cart_contingency
#c5_contingency
#rf_contingency
#nb_contingency
#nn_contingency
```

Decision Tree CART Evaluation

```
#develop the cart metrics to input into the confusion matrix
TP_cart <- cart_contingency[1,1]
FN_cart <- cart_contingency[1,2]
FP_cart <- cart_contingency[2,1]
TN_cart <- cart_contingency[2,2]

#totals for the corresponding row and column values of confusion matrix
TAN_cart <- FP_cart + TN_cart
TAP_cart <- TP_cart + FN_cart
TPN_cart <- TN_cart + FN_cart
TPP_cart <- FP_cart + TP_cart
GT_cart<- TP_cart + FN_cart + FP_cart + TN_cart

#generate the evaluation metrics for cart
accuracy_cart <- round((TN_cart + TP_cart) / (GT_cart),4)
error_rate_sensitivity_cart <- (1 - accuracy_cart)
specificity_cart <- round((TN_cart/TAN_cart),4)
precision_cart <- round((TP_cart/TPP_cart),4)
f1_cart <- round(2* ((precision_cart * specificity_cart) / (precision_cart +
specificity_cart)),4)
```

Decision Tree C5.0 Evaluation

```
#develop the c5.0 metrics to input into the confusion matrix
TP_c5 <- c5_contingency[1,1]
FN_c5 <- c5_contingency[1,2]
FP_c5 <- c5_contingency[2,1]
TN_c5 <- c5_contingency[2,2]

#totals for the corresponding row and column values of confusion matrix
TAN_c5 <- FP_c5 + TN_c5
TAP_c5 <- TP_c5 + FN_c5
```

```
TPN_c5 <- TN_c5 + FN_c5
TPP_c5 <- FP_c5 + TP_c5
GT_c5 <- TP_c5 + FN_c5 + FP_c5 + TN_c5

#generate the evaluation metrics for c5.0
accuracy_c5 <- round((TN_c5 + TP_c5) / (GT_c5),4)
error_rate_sensitivity_c5 <- (1 - accuracy_c5)
specificity_c5 <- round((TN_c5/TAN_c5),4)
precision_c5 <- round((TP_c5/TPP_c5),4)
f1_c5 <- round(2* ((precision_c5 * specificity_c5) / (precision_c5 +
specificity_c5)),4)
```

Random Forest Evaluation

```
#develop the random forest metrics to input into the confusion matrix
TP_rf <- rf_contingency[1,1]
FN_rf <- rf_contingency[1,2]
FP_rf <- rf_contingency[2,1]
TN_rf <- rf_contingency[2,2]

#totals for the corresponding row and column values of confusion matrix
TAN_rf <- FP_rf + TN_rf
TAP_rf <- TP_rf + FN_rf
TPN_rf <- TN_rf + FN_rf
TPP_rf <- FP_rf + TP_rf
GT_rf <- TP_rf + FN_rf + FP_rf + TN_rf

#generate the evaluation metrics for random forest
accuracy_rf <- round((TN_rf + TP_rf) / (GT_rf),4)
error_rate_sensitivity_rf <- (1 - accuracy_rf)
specificity_rf <- round((TN_rf/TAN_rf),4)
precision_rf <- round((TP_rf/TPP_rf),4)
f1_rf <- round(2* ((precision_rf * specificity_rf) / (precision_rf +
specificity_rf)),4)
```

Naive Bayes Evaluation

```
#develop the naive bayes metrics to input into the confusion matrix
TP_nb <- nb_contingency[1,1]
FN_nb <- nb_contingency[1,2]
FP_nb <- nb_contingency[2,1]
TN_nb <- nb_contingency[2,2]

#totals for the corresponding row and column values of confusion matrix
TAN_nb <- FP_nb + TN_nb
TAP_nb <- TP_nb + FN_nb
TPN_nb <- TN_nb + FN_nb
TPP_nb <- FP_nb + TP_nb
GT_nb <- TP_nb + FN_nb + FP_nb + TN_nb

#generate the evaluation metrics for naive bayes
accuracy_nb <- round((TN_nb + TP_nb) / (GT_nb),4)
```

```
error_rate_sensitivity_nb <- (1 - accuracy_nb)
specificity_nb <- round((TN_nb/TAN_nb),4)
precision_nb <- round((TP_nb/TPP_nb),4)
f1_nb <- round(2* ((precision_nb * specificity_nb) / (precision_nb +
specificity_nb)),4)
```

Neural Network Evaluation

```
#develop the neural network metrics to input into the confusion matrix
TP_nn <- nn_contingency[1,1]
FN_nn <- nn_contingency[1,2]
FP_nn <- nn_contingency[2,1]
TN_nn <- nn_contingency[2,2]

#totals for the corresponding row and column values of confusion matrix
TAN_nn <- FP_nn + TN_nn
TAP_nn <- TP_nn + FN_nn
TPN_nn <- TN_nn + FN_nn
TPP_nn <- FP_nn + TP_nn
GT_nn <- TP_nn + FN_nn + FP_nn + TN_nn

#generate the evaluation metrics for neural network
accuracy_nn <- round((TN_nn + TP_nn) / (GT_nn),4)
error_rate_sensitivity_nn <- (1 - accuracy_nn)
specificity_nn <- round((TN_nn/TAN_nn),4)
precision_nn <- round((TP_nn/TPP_nn),4)
f1_nn <- round(2* ((precision_nn * specificity_nn) / (precision_nn +
specificity_nn)),4)
```

Put all metrics into one matrix

```
#create a matrix to compare the model side by side
model_eval_table <- matrix(c(accuracy_cart, accuracy_c5, accuracy_nb,
accuracy_rf,accuracy_nn,
                             specificity_cart, specificity_c5,
specificity_nb, specificity_rf, specificity_nn,
                             precision_cart, precision_c5, precision_nb,
precision_rf, precision_nn,
                           f1_cart, f1_c5, f1_nb, f1_rf, f1_nn),
                           ncol = 5, nrow =4, byrow = TRUE)

#develop the column and row names
colnames(model_eval_table) <- c("CART", "C5.0", "Naive Bayes", "Random
Forest","Neural Network")
row.names(model_eval_table) <- c("Accuracy","Specificity","Precision","F1")

print(model_eval_table)

##                CART   C5.0 Naive Bayes Random Forest Neural Network
## Accuracy    0.5862 0.6309      0.5957        0.6446         0.6118
## Specificity 0.8800 0.8346      0.6638        0.7859         0.7719
```

```
## Precision   0.6036 0.6515       0.5532         0.6413          0.5966
## F1          0.7161 0.7318       0.6035         0.7063          0.6730
```