R-Markdown Code with plots

Data preparation

```
library(readr); library(dplyr); library(ggplot2); library(DataCombine); library(tidyverse); l
ibrary(cowplot); library(tree); library(randomForest); library(ROCR)
```

```
##
## 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
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

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

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

```
crime2019 = read_csv("Crime_Data_from_2010_to_2019.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 2117589 Columns: 28
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): DR_NO, Date Rptd, DATE OCC, TIME OCC, AREA, AREA NAME, Rpt Dist No...
## dbl (10): Part 1-2, Crm Cd, Vict Age, Premis Cd, Weapon Used Cd, Crm Cd 1, C...
## lgl  (1): Crm Cd 4
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(crime2019)
```

```
## [1] 2117589      28
```

```
crime = read_csv("Crime_Data_from_2020_to_Present.csv")
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```
## Rows: 276584 Columns: 28
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): DR_NO, Date Rptd, DATE OCC, TIME OCC, AREA, AREA NAME, Rpt Dist No...
## dbl (10): Part 1-2, Crm Cd, Vict Age, Premis Cd, Weapon Used Cd, Crm Cd 1, C...
## lgl  (1): Crm Cd 4
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(crime)
```

```
## [1] 276584     28
```

Checking for duplicate entries

```
crime2019 <- distinct(crime2019)
dim(crime2019)
```

```
## [1] 2117589      28
```

```
crime2019$AREA <- as.numeric(crime2019$AREA)

crime <- distinct(crime)
dim(crime)
```

```
## [1] 276584     28
```

```
crime$AREA <- as.numeric(crime$AREA)
class(crime$AREA)
```

```
## [1] "numeric"
```

To decide which dataset to use, or if we should merge them, we compared the distributions of crime severity in both sets and found that they were roughly similar.

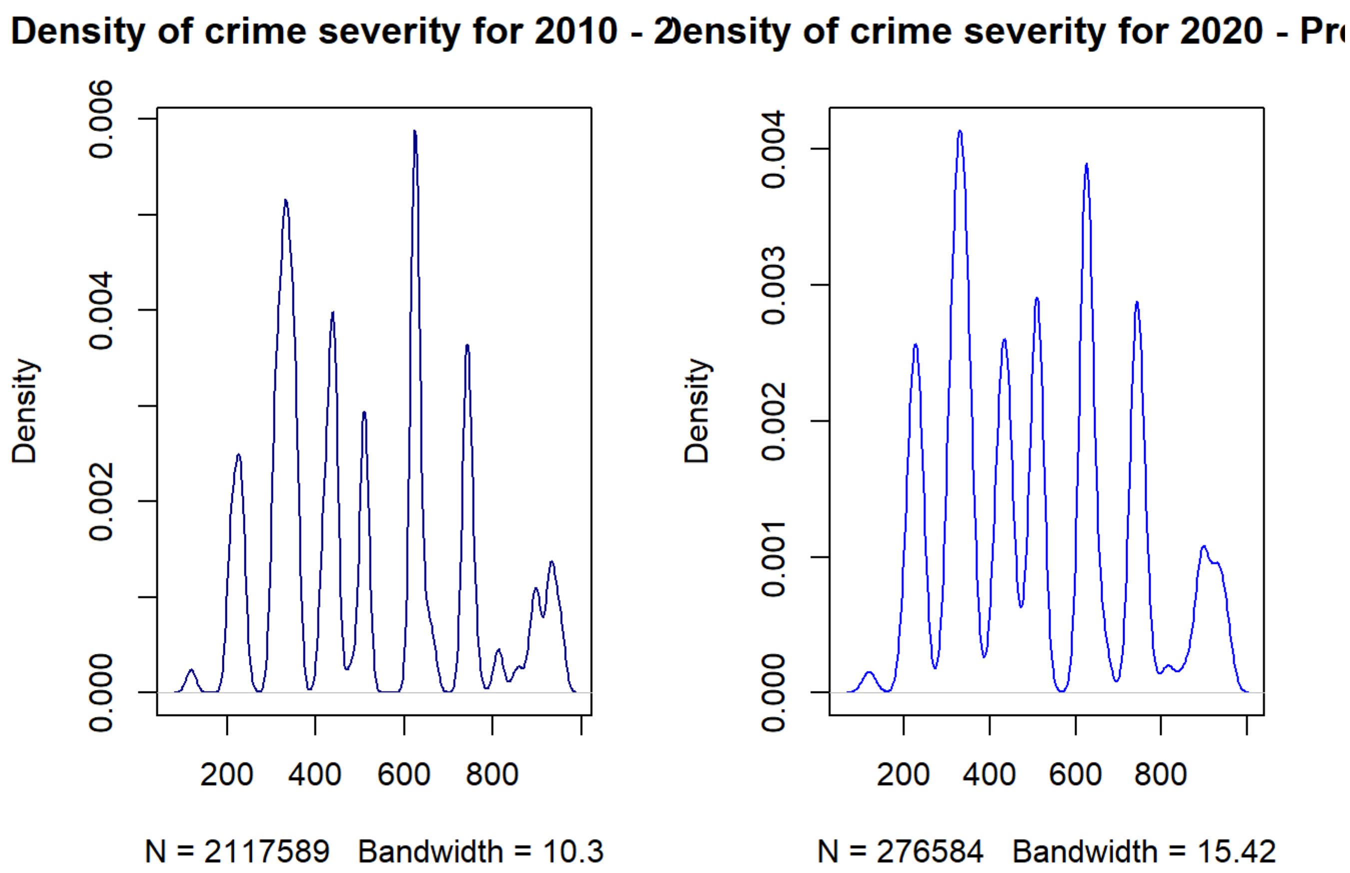
```
#Deciding whether to use only 2019-2020 data, proportions of crime severity is roughly the sa
me
summary(crime2019$`Crm Cd`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   110.0   330.0   442.0   507.4   626.0   956.0
```

```
summary(crime$`Crm Cd`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   110.0   330.0   510.0   509.2   626.0   956.0
```

```
crime2019d <- density(crime2019$`Crm Cd`)
crimed <- density(crime$`Crm Cd`)

par(mfrow=c(1,2))
plot(crime2019d, main="Density of crime severity for 2010 - 2019", col="Navy blue")
plot(crimed, main="Density of crime severity for 2020 - Present", col="Blue")
```




```
par(mfrow=c(1,1))
```

Renaming all the columns

```
names(crime) <- c('RecNo','ReportDate','DateOCC','TimeOCC','Area','AreaName','DistrictNo','Pa
rt','CrimeCode','CrmDesc','Mocodes','VictAge','VictSex','VictRace','PremiseCd','PremiseDesc',
'WeaponCd','WeaponDesc','Status','StatusDesc','CrimeCd1','CrimeCd2','CrimeCd3','CrimeCd4','Lo
cation','CrossStreet','Lat','Lon')
```

Checking and removing null entries in our predictors

```
sum(is.na(crime$DateOCC))
```

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

```
sum(is.na(crime$TimeOCC))
```

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

```
sum(is.na(crime$Area))
```

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

```
sum(is.na(crime$RecNo))
```

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

```
sum(is.na(crime$CrimeCode))
```

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

```
sum(is.na(crime$DistrictNo))
```

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

```
sum(is.na(crime$Mocodes))
```

```
## [1] 37993
```

```
sum(is.na(crime$VictAge))
```

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

```
sum(is.na(crime$VictSex))
```

```
## [1] 36357
```

```
sum(is.na(crime$VictRace))
```

```
## [1] 36362
```

```
sum(is.na(crime$PremiseCd))
```

```
## [1] 4
```

```
sum(is.na(crime$PremiseDesc))
```

```
## [1] 97
```

```
sum(is.na(crime$WeaponCd))
```

```
## [1] 175499
```

```
#Mocodes, VictSex, VictRace, PremiseCd, WeaponCd have null entries

#Removing records with null values and illogical values
crime <- crime[!is.na(crime$Mocodes),]
crime <- crime[!is.na(crime$VictAge),]
crime <- crime[!is.na(crime$VictSex),]
crime <- crime[!is.na(crime$VictRace),]
crime <- crime[!is.na(crime$PremiseDesc),]
crime <- crime[!is.na(crime$PremiseCd),]

#Replacing null values with 0 for WeaponCd to denote no weapon involved
crime$WeaponCd[is.na(crime$WeaponCd)] <- 0

sum(is.na(crime$Mocodes))
```

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

```
sum(is.na(crime$VictAge))
```

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

```
sum(is.na(crime$VictSex))
```

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

```
sum(is.na(crime$VictRace))
```

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

```
sum(is.na(crime$PremiseCd))
```

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

```
sum(is.na(crime$WeaponCd))
```

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

```
dim(crime)
```

```
## [1] 238435     28
```

Removing unknown and illogical values

```
#Removing unknown records in VictSex and VictRace, Removing 0 in VictAge
crime <- crime[!(crime$VictSex=="X"),]
crime <- crime[!(crime$VictRace=="X"),]
crime <- crime[!(crime$VictAge==0),]
dim(crime)
```

```
## [1] 203585     28
```

```
attach(crime)
crime = subset(crime, select = -c(CrimeCd2,CrimeCd3,CrimeCd4))
crime = subset(crime, select = -c(Lat, Lon, RecNo,DateOCC,DistrictNo,ReportDate,CrossStreet,L
ocation,StatusDesc,Status,WeaponDesc, Mocodes, CrmDesc, Part, AreaName))
```

Transforming some predictors

```
#Transforming columns - CrimeCode
Severity = ifelse(crime$CrimeCd1 < 300, 'Severe', 'Non-Severe')
Severity = as.factor(Severity)
crime <- data.frame(crime, Severity)

#Removing CrimeCode and CrimeCd1 as Severity has replaced it
crime = subset(crime, select = -c(CrimeCode,CrimeCd1))

#Transforming columns - VictSex and Weapon
Female <-ifelse(crime$VictSex == "F", 'Yes', 'No')
Female <- as.factor(Female)
crime <- data.frame(crime, Female)

Weapon <-ifelse(crime$WeaponCd == 0, 'No', 'Yes')
Weapon <- as.factor(Weapon)
crime <- data.frame(crime, Weapon)

crime = subset(crime, select = -c(VictSex,WeaponCd))
```

Exploratory data analysis

```
#Taking Severe Crimes only
severeexploratory = crime[!(crime$Severity=="Non-Severe"),]
severeexploratory$TimeOCC = as.numeric(severeexploratory$TimeOCC)
```

```
#Time
TimeBar = ggplot(severeexploratory, aes(x = TimeOCC)) +
  geom_bar(stat = 'count', fill = 'red') +
  labs(x = 'Time of Crime', y = 'Number of Severe Crimes') +
  scale_x_continuous(limit = c(0,2400,0.1)) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank())
```

```r
#Area
severeexploratory$Area = as.factor(severeexploratory$Area)
```

```r
AreaBar = ggplot(severeexploratory, aes(x = Area)) +
  geom_bar(stat = 'count', fill = 'blue') +
  labs(x = 'Area', y = 'Number of Severe Crimes') +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank())
```

```r
#Sex
VictSexBar = ggplot(severeexploratory, aes(x = Female)) +
  geom_bar(stat = 'count', fill = c('red', 'green')) +
  labs(x = 'Is the Victim Female?', y = 'Number of Severe Crimes') +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank())
```

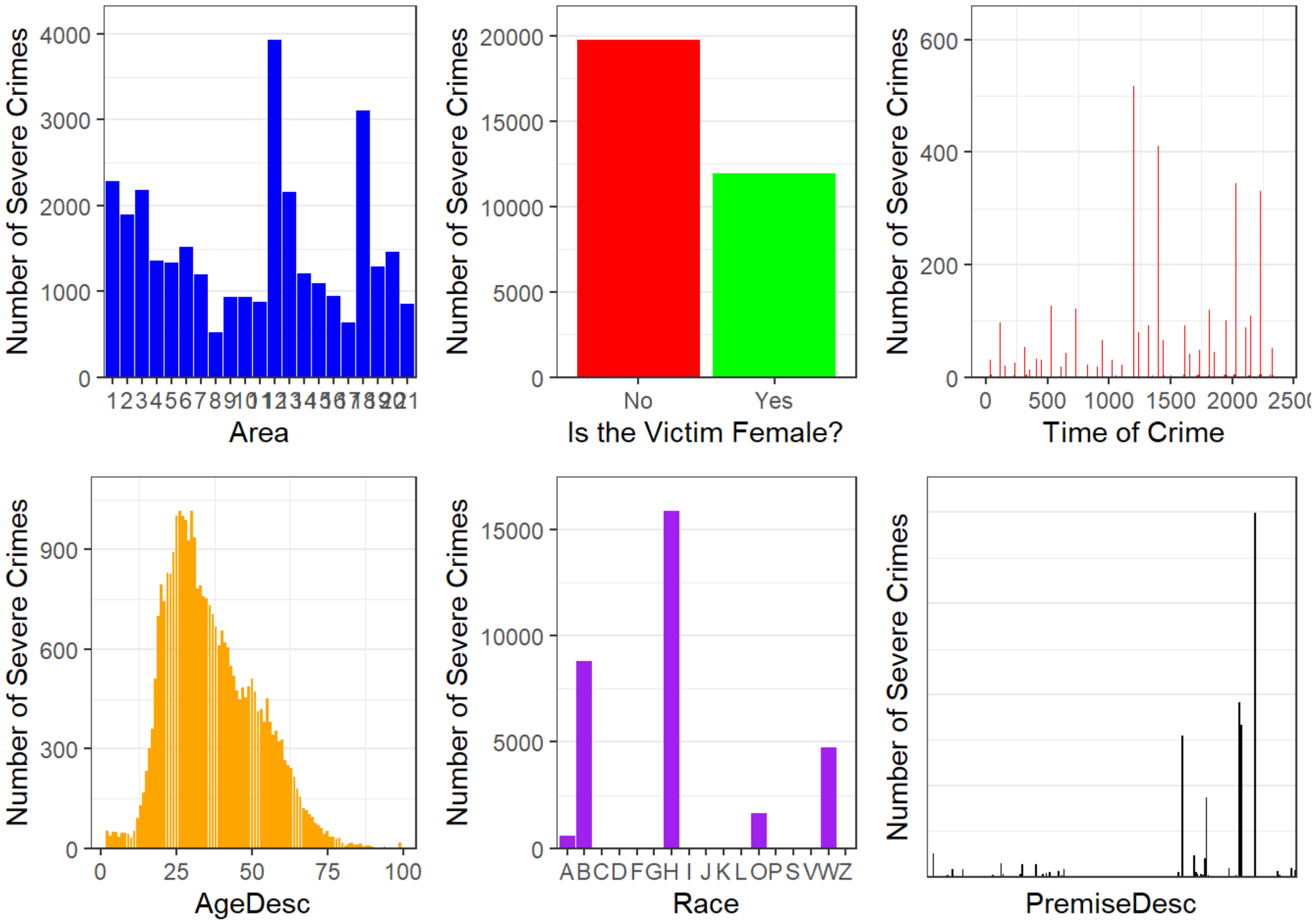
```r
#Race
VictRaceBar = ggplot(severeexploratory, aes(x = VictRace)) +
  geom_bar(stat = 'count', fill = c('purple')) +
  labs(x = 'Race', y = 'Number of Severe Crimes') +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank())
```

```r
#Age
AgeDescBar = ggplot(severeexploratory, aes(x = VictAge)) +
  geom_bar(stat = 'count', fill = c('orange')) +
  labs(x = 'AgeDesc', y = 'Number of Severe Crimes') +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank())
```

```r
#Premise
severeexploratory$PremiseDesc = as.factor(severeexploratory$PremiseDesc)
```

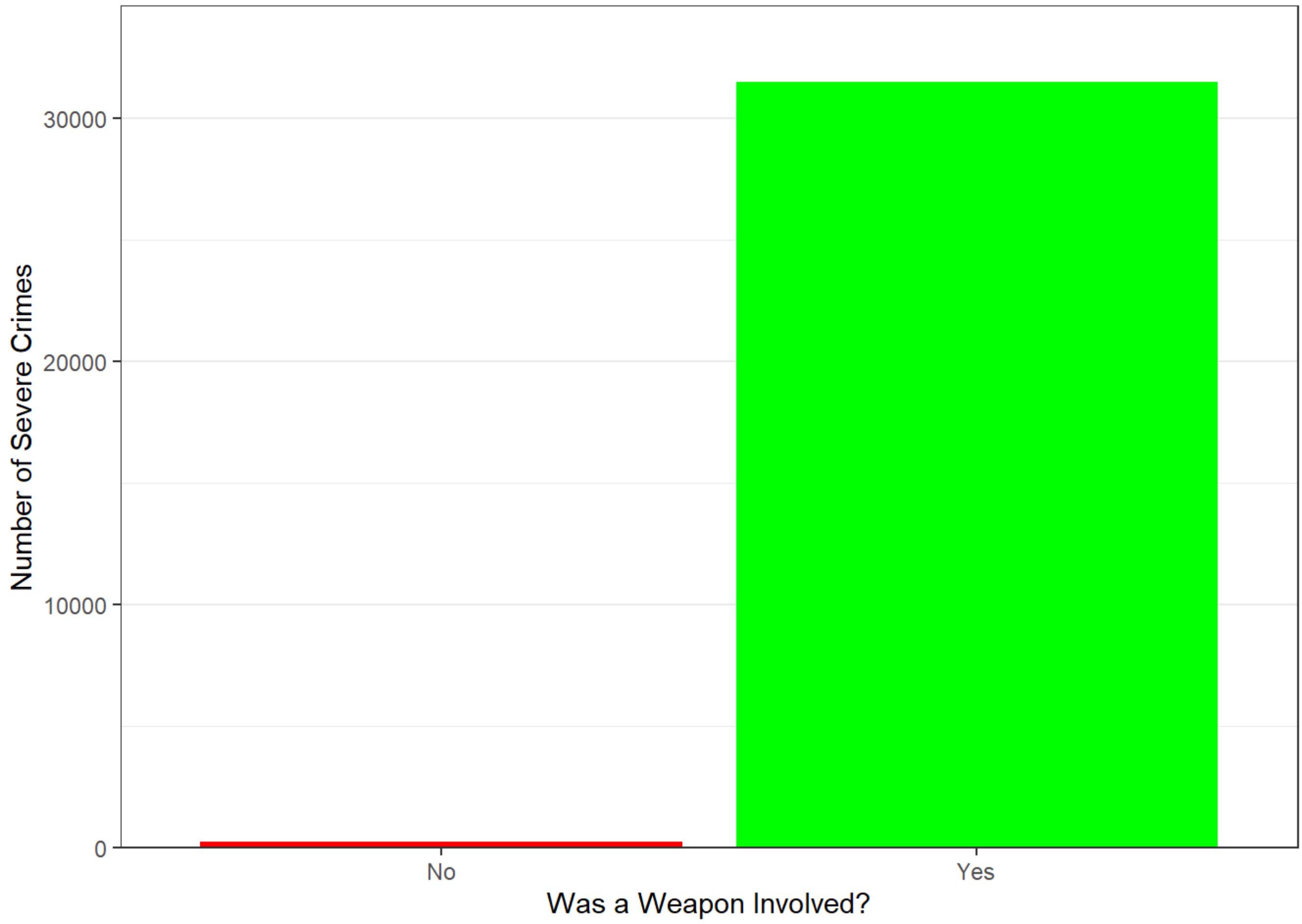
```r
PremiseDescBar = ggplot(severeexploratory, aes(x = PremiseDesc)) +
  geom_bar(stat = 'count', fill = c('black')) +
  labs(x = 'PremiseDesc', y = 'Number of Severe Crimes') +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank()) +
  theme(axis.text = element_blank()) +
  theme(axis.ticks = element_blank())
```

```r
plot_grid(AreaBar, VictSexBar, TimeBar, AgeDescBar, VictRaceBar, PremiseDescBar)
```




```
#Weapon
VictWeapBar = ggplot(severeexploratory, aes(x = Weapon)) +
  geom_bar(stat = 'count', fill = c('red', 'green')) +
  labs(x = 'Was a Weapon Involved?', y = 'Number of Severe Crimes') +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1))) +
  theme_bw() +
  theme(panel.grid.major.x = element_blank())
```

```
VictWeapBar
```



Data transformation

```
#Splitting premise into 4 categories: Commercial, residential, industrial and outdoors
premisetable <- table(crime['PremiseDesc'])
premisetable <- sort(premisetable,decreasing = TRUE)
premisetable[1:10]
```

```
##
##                      SINGLE FAMILY DWELLING
##                                       43792
##                                      STREET
##                                       39748
## MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)
##                                       30913
##                                 PARKING LOT
##                                       13719
##                                    SIDEWALK
##                                       12062
##                    VEHICLE, PASSENGER/TRUCK
##                                        8794
##                              OTHER BUSINESS
##                                        6155
##                              GARAGE/CARPORT
##                                        4634
##                                    DRIVEWAY
##                                        3946
##                PARKING UNDERGROUND/BUILDING
##                                        2665
```

```
cat('Percentage of top 10 premises:',sum(premisetable[1:10])/nrow(crime)*100,'%')
```

```
## Percentage of top 10 premises: 81.74866 %
```

```
OtherPremise = case_when(crime$PremiseDesc =='SINGLE FAMILY DWELLING'~'No',crime$PremiseDesc
 =='STREET'~'No',crime$PremiseDesc =='MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)'~'No',crim
e$PremiseDesc =='PARKING LOT'~'No',crime$PremiseDesc =='SIDEWALK'~'No',crime$PremiseDesc =='V
EHICLE, PASSENGER/TRUCK'~'No',crime$PremiseDesc =='OTHER BUSINESS'~'No',crime$PremiseDesc ==
'GARAGE/CARPORT'~'No',crime$PremiseDesc =='DRIVEWAY'~'No',crime$PremiseDesc =='PARKING UNDERG
ROUND/BUILDING'~'No')
SFamDwelling = case_when(crime$PremiseDesc == 'SINGLE FAMILY DWELLING'~'Yes')
Street = case_when(crime$PremiseDesc == 'STREET'~'Yes')
MUDwelling = case_when(crime$PremiseDesc == 'MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)'~'Y
es')
Parking = case_when(crime$PremiseDesc == 'PARKING LOT'~'Yes')
Sidewalk = case_when(crime$PremiseDesc == 'SIDEWALK'~'Yes')
Vehicle = case_when(crime$PremiseDesc=='VEHICLE, PASSENGER/TRUCK'~'Yes')
OtherBusiness = case_when(crime$PremiseDesc == 'OTHER BUSINESS'~'Yes')
Garage = case_when(crime$PremiseDesc == 'GARAGE/CARPORT'~'Yes')
Driveway = case_when(crime$PremiseDesc == 'DRIVEWAY'~'Yes')
UnderParking = case_when(crime$PremiseDesc=='PARKING UNDERGROUND/BUILDING'~'Yes')

crime = cbind(crime,SFamDwelling,Street,MUDwelling,Parking,Sidewalk,Vehicle,OtherBusiness,Gar
age,Driveway,UnderParking,OtherPremise)
crime$OtherPremise[is.na(crime$OtherPremise)] <- 'Yes'
crime[is.na(crime)] <- 'No'

crime <- subset(crime, select = -c(PremiseCd,PremiseDesc))

#Splitting race by groups
table(VictRace)
```

```
## VictRace
##      A     B     C     D     F     G     H     I     J     K     L     O     P
##   5829 38816   631    10   789    22 82831   162   258  1214     8 18408    48
##      S     U     V     W     Z
##     10    27   192 54265    65
```

```
cat('Percentage of Asians in our dataset:',5829/nrow(crime)*100,"%")
```

```
## Percentage of Asians in our dataset: 2.863178 %
```

```
Asian = case_when(crime$VictRace == 'A' ~ 'Yes', crime$VictRace == 'C' ~ 'Yes', crime$VictRac
e == 'D' ~ 'Yes', crime$VictRace == 'F' ~ 'Yes', crime$VictRace == 'J' ~ 'Yes', crime$VictRac
e == 'K' ~ 'Yes', crime$VictRace == 'L' ~ 'Yes', crime$VictRace == 'V' ~ 'Yes', crime$VictRac
e == 'Z' ~ 'Yes', TRUE ~ 'No')
table(Asian)
```

```
## Asian
##     No    Yes
## 194589   8996
```

```
Black = ifelse(crime$VictRace == 'B', 'Yes', 'No')
Hispanic = ifelse(crime$VictRace == 'H', 'Yes', 'No')
White = ifelse(crime$VictRace == 'W', 'Yes', 'No')
OtherRace = case_when(crime$VictRace == 'O' ~ 'Yes', crime$VictRace == 'G' ~ 'Yes', crime$Vic
tRace == 'I' ~ 'Yes', crime$VictRace == 'P' ~ 'Yes', crime$VictRace == 'S' ~ 'Yes', crime$Vic
tRace == 'U' ~ 'Yes', TRUE ~ 'No')
crime = cbind(crime,Asian,Black,Hispanic,White,OtherRace)

crime <- subset(crime, select = -c(VictRace))



crime$TimeOCC = as.numeric(crime$TimeOCC)

#Splitting time into 4 groups
Morning = ifelse(crime$TimeOCC <= 1159 & crime$TimeOCC >= 600, 'Yes', 'No')
Day = ifelse(crime$TimeOCC <= 1759 & crime$TimeOCC >= 1200, 'Yes', 'No')
Evening = ifelse(crime$TimeOCC <= 2359 & crime$TimeOCC >= 1800, 'Yes', 'No')
Night = ifelse(crime$TimeOCC <= 559 & crime$TimeOCC >= 0000, 'Yes', 'No')
crime = cbind(crime, Morning, Day, Evening, Night)

crime <- subset(crime, select = -c(TimeOCC))

#Splitting area into 4 boroughs: Valley, West, Central and South
Valley = case_when(crime$Area == 9 ~ 'Yes', crime$Area == 10 ~ 'Yes', crime$Area == 15 ~ 'Ye
s', crime$Area == 16 ~ 'Yes', crime$Area == 17 ~ 'Yes', crime$Area == 19 ~ 'Yes', crime$Area
 == 21 ~ 'Yes', TRUE ~ 'No')
West = case_when(crime$Area == 6 ~ 'Yes', crime$Area == 7 ~ 'Yes', crime$Area == 8 ~ 'Yes', c
rime$Area == 14 ~ 'Yes', crime$Area == 20 ~ 'Yes', TRUE ~ 'No')
Central = case_when(crime$Area == 1 ~ 'Yes', crime$Area == 2 ~ 'Yes', crime$Area == 4 ~ 'Yes'
, crime$Area == 11 ~ 'Yes', crime$Area == 13 ~ 'Yes', TRUE ~ 'No')
South = case_when(crime$Area == 3 ~ 'Yes', crime$Area == 5 ~ 'Yes', crime$Area == 12 ~ 'Yes',
crime$Area == 18 ~ 'Yes', TRUE ~ 'No')
crime = cbind(crime, Valley, West, South, Central)

crime <- subset(crime, select = -c(Area))

View(crime)
```

Decision tree modelling

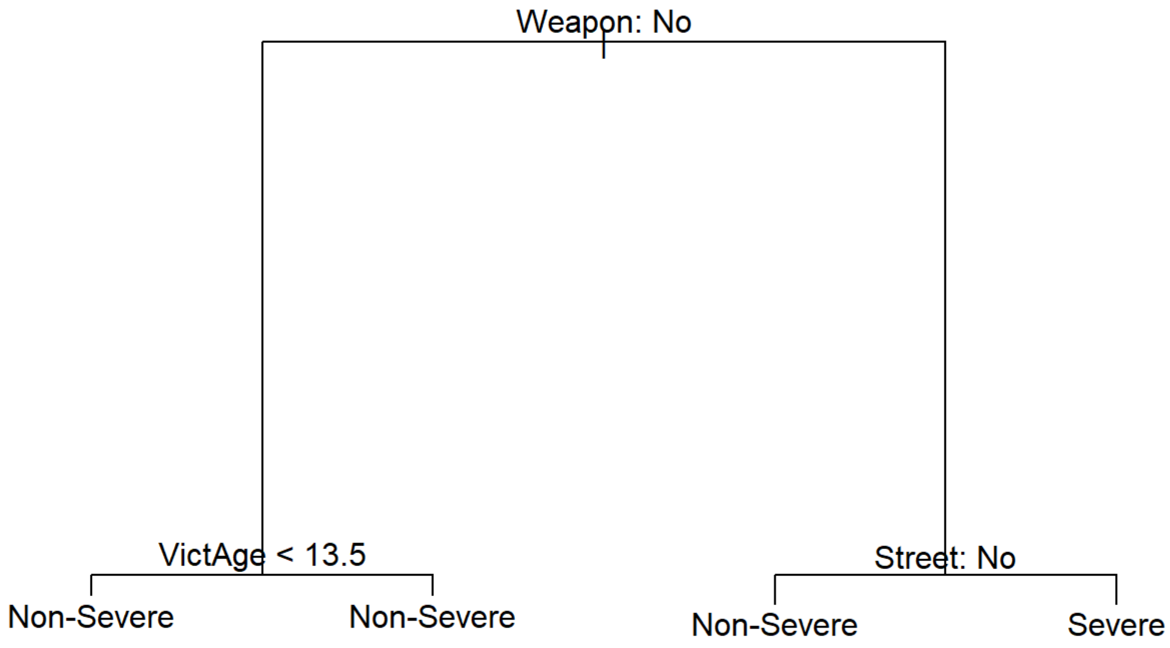
```
library(tree)
set.seed(1)

convertcols <- c("Female", "Weapon", "SFamDwelling", "Street","MUDwelling", "Parking", "Sidew
alk", "Vehicle", "OtherBusiness", "Garage", "Driveway", "UnderParking", "OtherPremise", "Asia
n", "Black", "Hispanic", "White", "OtherRace", "Morning", "Day", "Evening", "Night","Valley",
"West","South","Central")
crime[convertcols] <- lapply(crime[convertcols],factor)
sapply(crime, class)
```

```
##        VictAge       Severity         Female         Weapon   SFamDwelling
##      "numeric"       "factor"       "factor"       "factor"       "factor"
##         Street     MUDwelling        Parking       Sidewalk        Vehicle
##       "factor"       "factor"       "factor"       "factor"       "factor"
## OtherBusiness         Garage       Driveway   UnderParking   OtherPremise
##       "factor"       "factor"       "factor"       "factor"       "factor"
##          Asian          Black       Hispanic          White      OtherRace
##       "factor"       "factor"       "factor"       "factor"       "factor"
##        Morning            Day        Evening          Night         Valley
##       "factor"       "factor"       "factor"       "factor"       "factor"
##           West          South        Central
##       "factor"       "factor"       "factor"
```

```
#Testing the tree
tree1 = tree(Severity ~., data = crime)
summary(tree1)
```

```
##
## Classification tree:
## tree(formula = Severity ~ ., data = crime)
## Variables actually used in tree construction:
## [1] "Weapon"  "VictAge" "Street"
## Number of terminal nodes:  4
## Residual mean deviance:  0.5719 = 116400 / 203600
## Misclassification error rate: 0.1535 = 31249 / 203585
```

```
plot(tree1)
text(tree1, pretty = 0)
```




```
cat('Percentage of severe crimes:',sum(crime$Severity=="Severe")/nrow(crime)*100,'%')
```

```
## Percentage of severe crimes: 15.57679 %
```

```
detach(crime)

#Cutting down the dataset because the above results were unsatisfactory
#Resampling the dataset to 10,000 samples only (5000 severe crimes, 5000 non-severe crimes)
#This will be our training data
nonsevere = crime[!(crime$Severity=="Severe"),]
severe = crime[!(crime$Severity=="Non-Severe"),]

train = sample(1:nrow(severe),5000)
trainnotsevere = sample(1:nrow(nonsevere),5000)
testdatasev = severe[-train,]
traindatasev = severe[train,]
testdatanotsev = nonsevere[-trainnotsevere,]
traindatanonsev = nonsevere[trainnotsevere,]
traindatafinal = rbind(traindatasev,traindatanonsev)
testdatafinal = rbind(testdatasev, testdatanotsev)

#New trees
tree1 = tree(formula = Severity~., data = traindatafinal)
summary(tree1)
```

```
##
## Classification tree:
## tree(formula = Severity ~ ., data = traindatafinal)
## Variables actually used in tree construction:
## [1] "Weapon"  "VictAge" "Female"
## Number of terminal nodes:  4
## Residual mean deviance:  0.744 = 7437 / 9996
## Misclassification error rate: 0.1711 = 1711 / 10000
```

```
plot(tree1)
text(tree1, pretty = 0)

#Plotting tree 1 next to weapon percentage
table(traindatafinal$Severity, traindatafinal$Weapon)
```

```
##
##                No  Yes
##   Non-Severe 3301 1699
##   Severe       39 4961
```

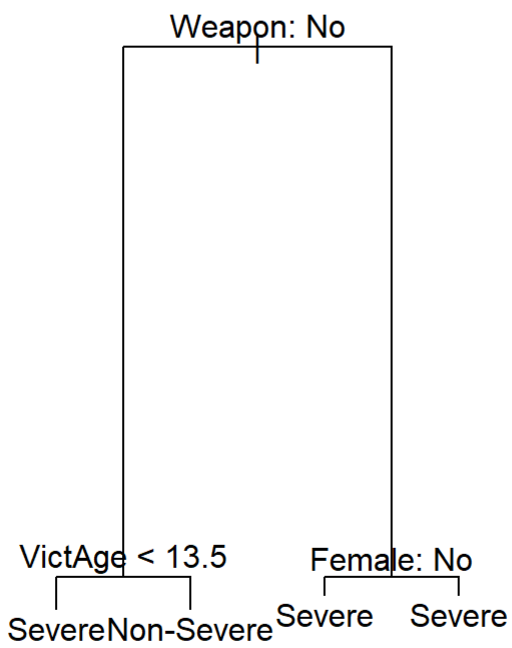
```
cat('Percentage of severe crimes committed with weapons:',4961/(39+4961)*100,"%")
```

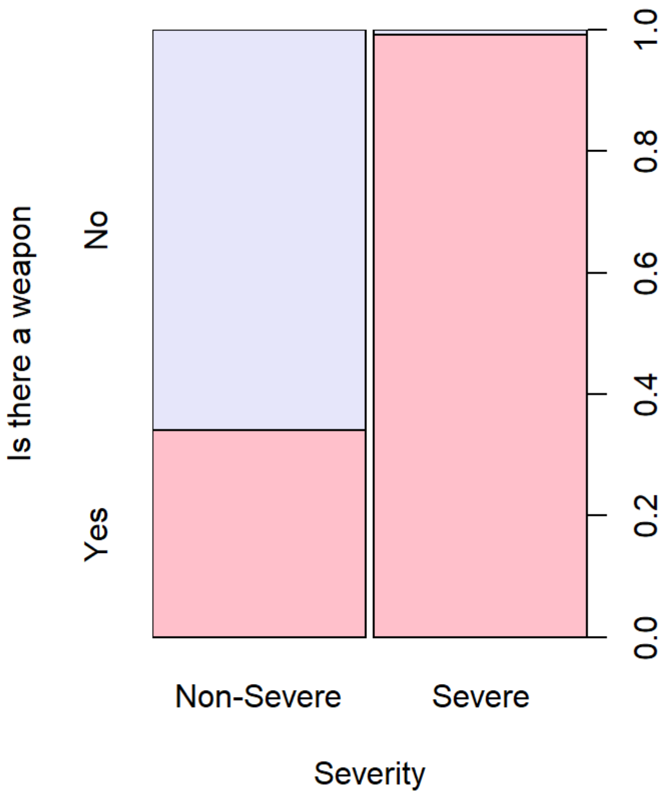
```
## Percentage of severe crimes committed with weapons: 99.22 %
```

```
cat('Percentage of non-severe crimes committed with weapons:',1699/(1699+3301)*100,"%")
```

```
## Percentage of non-severe crimes committed with weapons: 33.98 %
```

```
par(mfrow=c(1,2))
plot(tree1); text(tree1, pretty = 0)
plot(traindatafinal$Severity,traindatafinal$Weapon, xlab="Severity",ylab="Is there a weapon",
col=c("Pink","Lavender"))
```

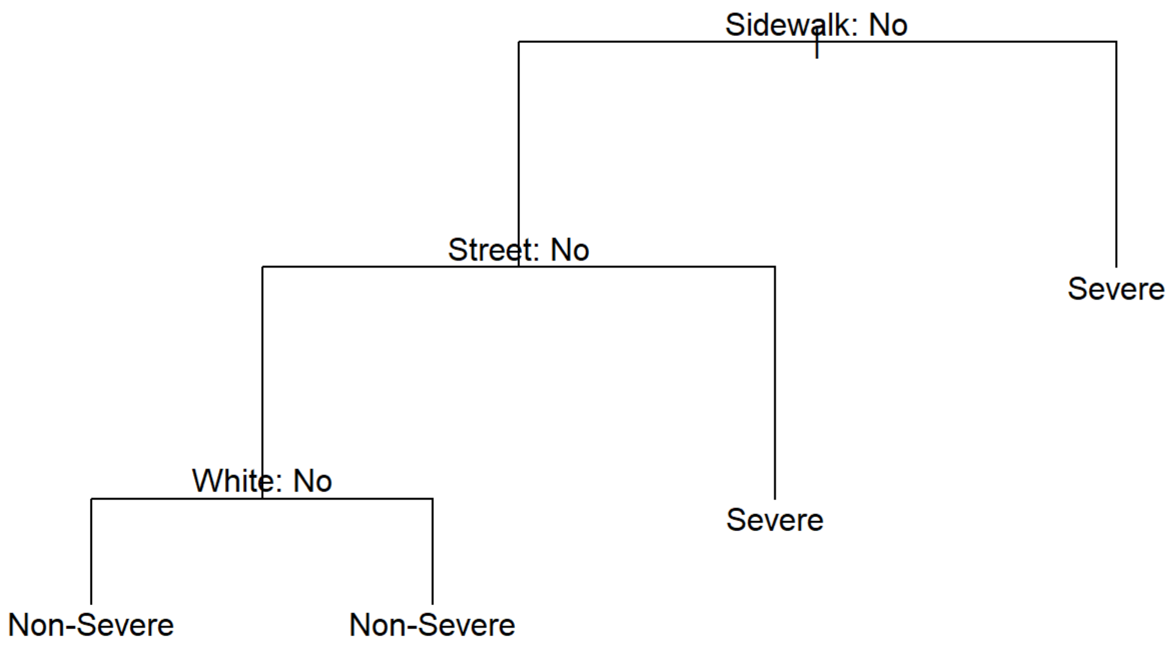





```r
par(mfrow=c(1,1))

#Tree without weapons
tree2 = tree(formula = Severity ~.-Weapon, data = traindatafinal)
summary(tree2)
```

```
##
## Classification tree:
## tree(formula = Severity ~ . - Weapon, data = traindatafinal)
## Variables actually used in tree construction:
## [1] "Sidewalk" "Street"   "White"
## Number of terminal nodes:  4
## Residual mean deviance:  1.294 = 12940 / 9996
## Misclassification error rate: 0.3778 = 3778 / 10000
```

```r
plot(tree2)
text(tree2, pretty = 0)
```



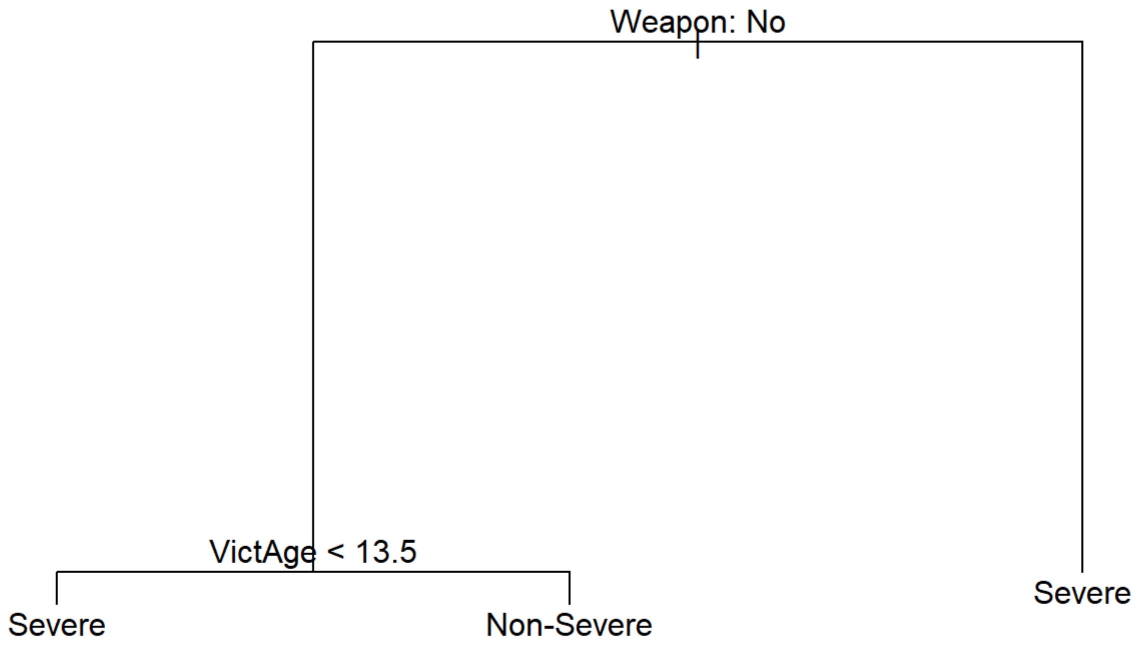

```
#Cross validation
#Applying to tree1
cv.crime1 = cv.tree(tree1, FUN=prune.misclass)

#Picking 3 nodes because our original already has 4 nodes
cv.crime1$size
```

```
## [1] 4 3 2 1
```

```
cv.crime1$dev
```

```
## [1] 1717 1717 1739 5116
```

```
prune.crime1 = prune.misclass(tree1,best=3)
plot(prune.crime1)
text(prune.crime1, pretty=0)
```



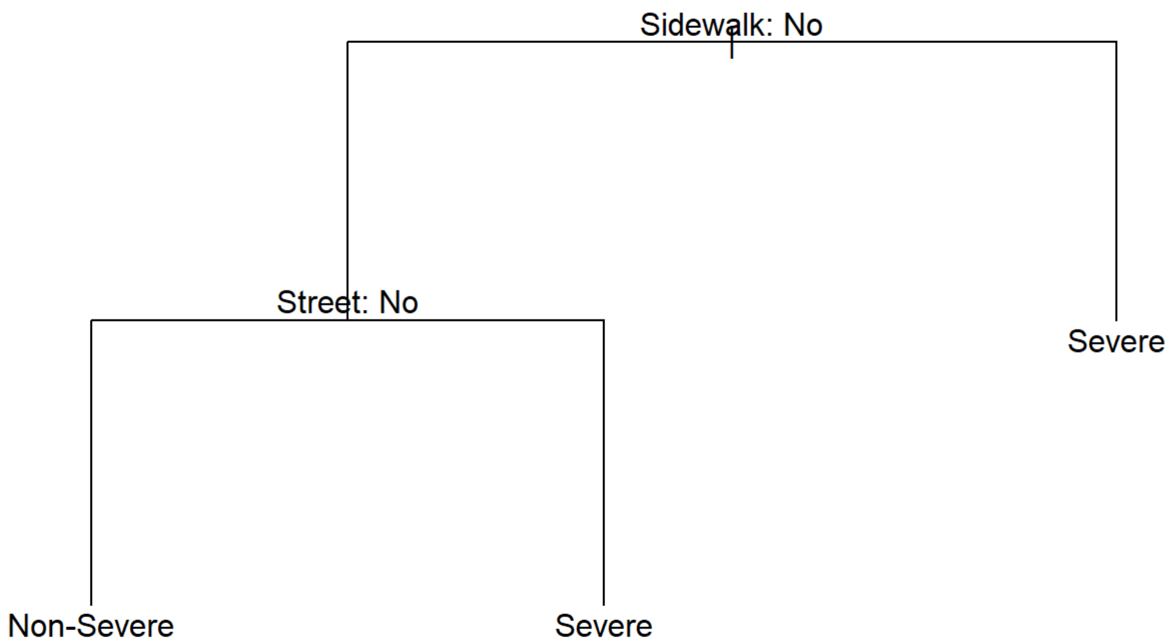

```
#Applying to tree2
cv.crime2 = cv.tree(tree2, FUN=prune.misclass)

#Picking 3 nodes
cv.crime2$size
```

```
## [1] 4 3 1
```

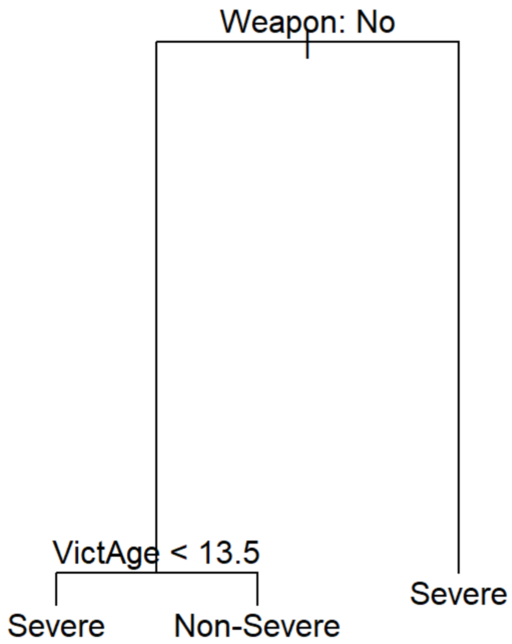
```
cv.crime2$dev
```

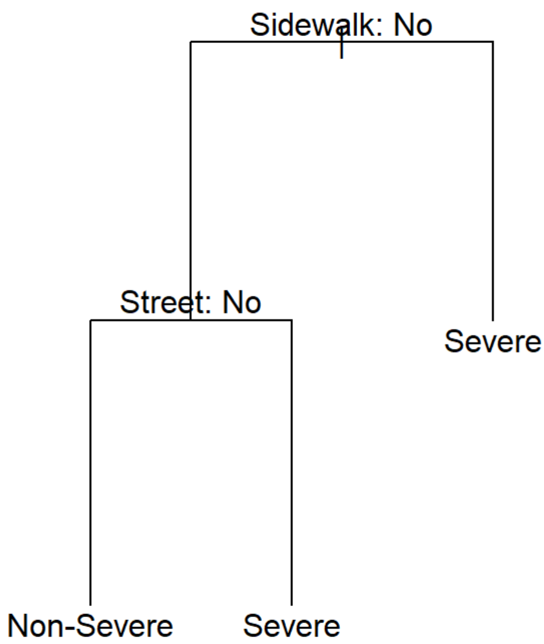
```
## [1] 3778 3778 5133
```

```
prune.crime2 = prune.misclass(tree2,best=3)
plot(prune.crime2)
text(prune.crime2, pretty=0)
```




```
#Plotting both trees side by side
par(mfrow=c(1,2))
plot(prune.crime1); text(prune.crime1, pretty=0)
plot(prune.crime2); text(prune.crime2, pretty=0)
```






```
par(mfrow=c(1,1))

#Testing performance of tree1
crime.treePredict1=predict(prune.crime1, newdata = testdatafinal, type="class")
table(crime.treePredict1, testdatafinal$Severity)
```

```
##
## crime.treePredict1 Non-Severe Severe
##         Non-Severe     107775     28
##         Severe          59098  26684
```

```
cat("The misclassification rate for the testing data is",(28+59098)/(107775+28+59095+26684))
```

```
## The misclassification rate for the testing data is 0.3054313
```

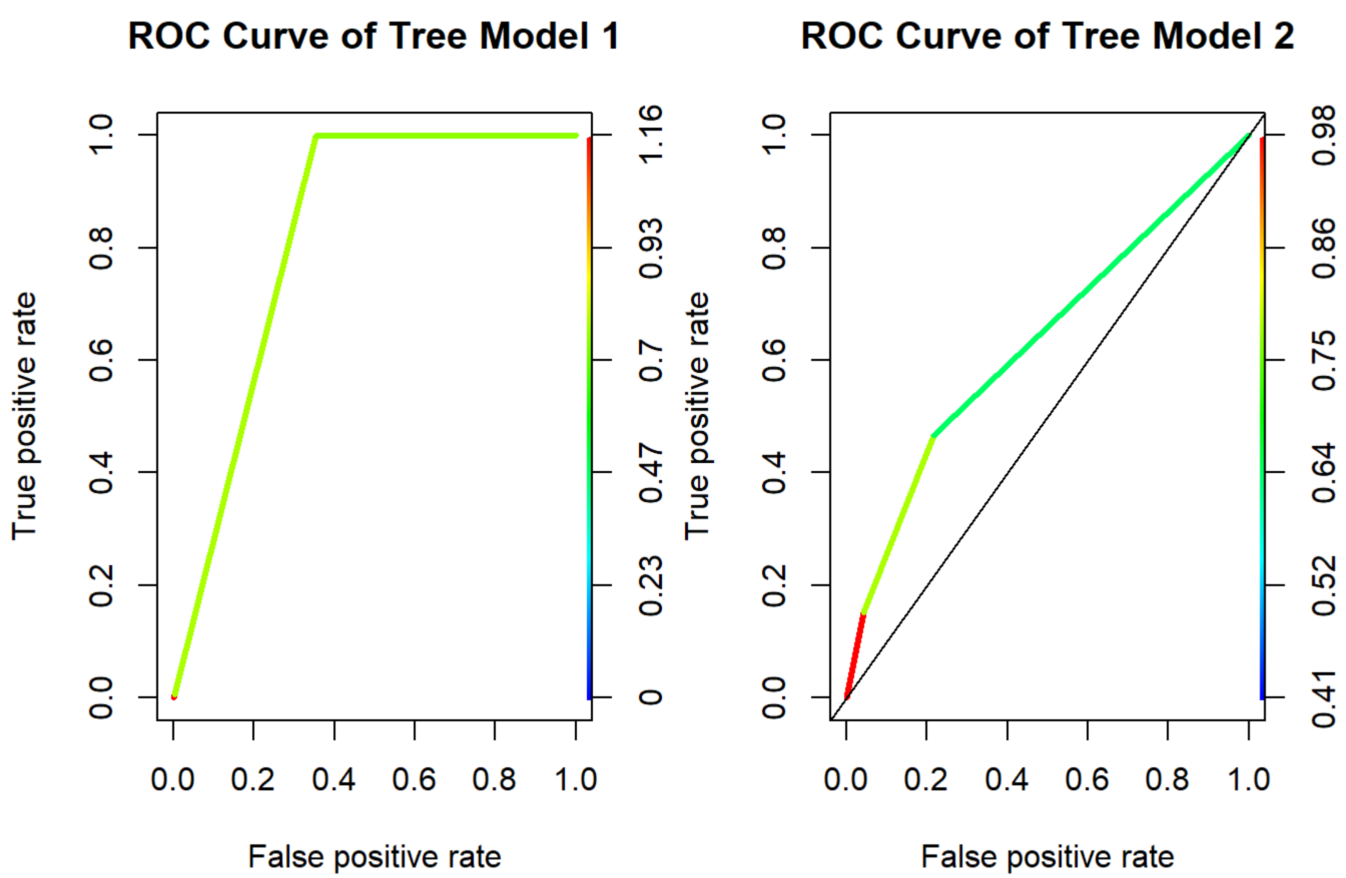
```
#Testing performance of tree2
crime.treePredict2=predict(prune.crime2, newdata = testdatafinal, type="class")
table(crime.treePredict2, testdatafinal$Severity)
```

```
##
## crime.treePredict2 Non-Severe Severe
##         Non-Severe     130922  14261
##         Severe          35951  12451
```

```
cat("The misclassification rate for the testing data is",(14261+35951)/(130922+14261+35951+12
451))
```

```
## The misclassification rate for the testing data is 0.2593796
```

```
#ROC Curves for our decision trees
#Tree model 1
pred.tree1 = predict(prune.crime1, testdatafinal, type="vector")
prediction.tree1 = prediction(pred.tree1[,2], testdatafinal$Severity)
rocTree1=performance(prediction.tree1, measure = "tpr", x.measure = "fpr")

#Tree model 2
pred.tree2 = predict(prune.crime2, testdatafinal, type="vector")
prediction.tree2 = prediction(pred.tree2[,2], testdatafinal$Severity)
rocTree2=performance(prediction.tree2, measure = "tpr", x.measure = "fpr")

#Plotting both curves side by side
par(mfrow=c(1,2))
plot(rocTree1, lwd=3, colorkey=T, colorize=T, main="ROC Curve of Tree Model 1")
plot(rocTree2, lwd=3, colorkey=T, colorize=T, main="ROC Curve of Tree Model 2")
abline(0,1)
```




```
performance(prediction.tree1, measure = "auc")@y.values
```

```
## [[1]]
## [1] 0.8223889
```

```
performance(prediction.tree2, measure = "auc")@y.values
```

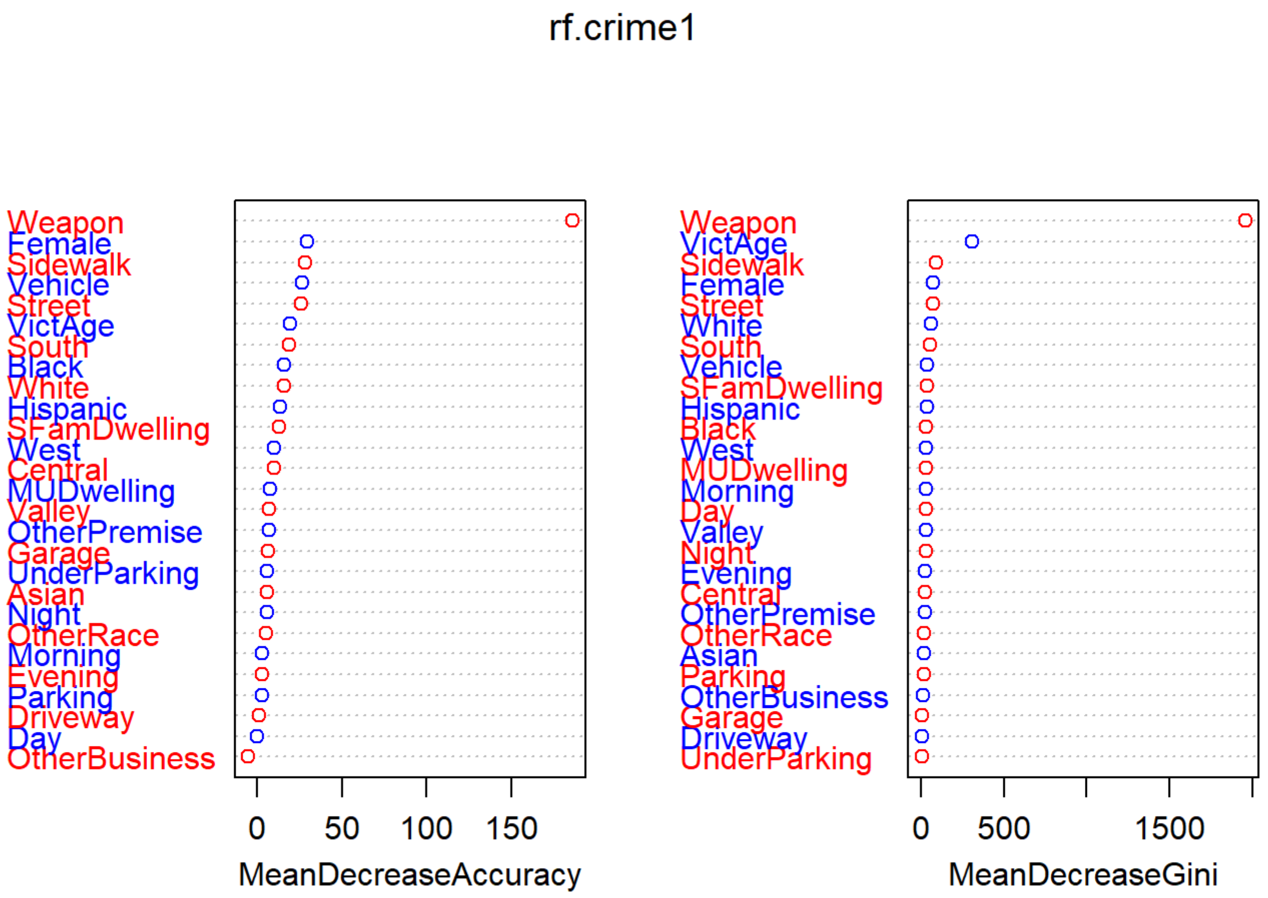
```
## [[1]]
## [1] 0.6317515
```

Random forests modelling

```
#With weapons
rf.crime1 = randomForest(Severity~., data = traindatafinal, mtry = 5, importance = T)
rf.crime1
```

```
##
## Call:
##  randomForest(formula = Severity ~ ., data = traindatafinal, mtry = 5,      importance =
T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 17.53%
## Confusion matrix:
##            Non-Severe Severe class.error
## Non-Severe       3484   1516      0.3032
## Severe            237   4763      0.0474
```

```
varImpPlot(rf.crime1, col = c('red', 'blue'))
```




```
test.rf1 = predict(rf.crime1, newdata = testdatafinal, type = 'class')
table(test.rf1, testdatafinal$Severity)
```

```
##
## test.rf1     Non-Severe Severe
##   Non-Severe     114286   1126
##   Severe          52587  25586
```

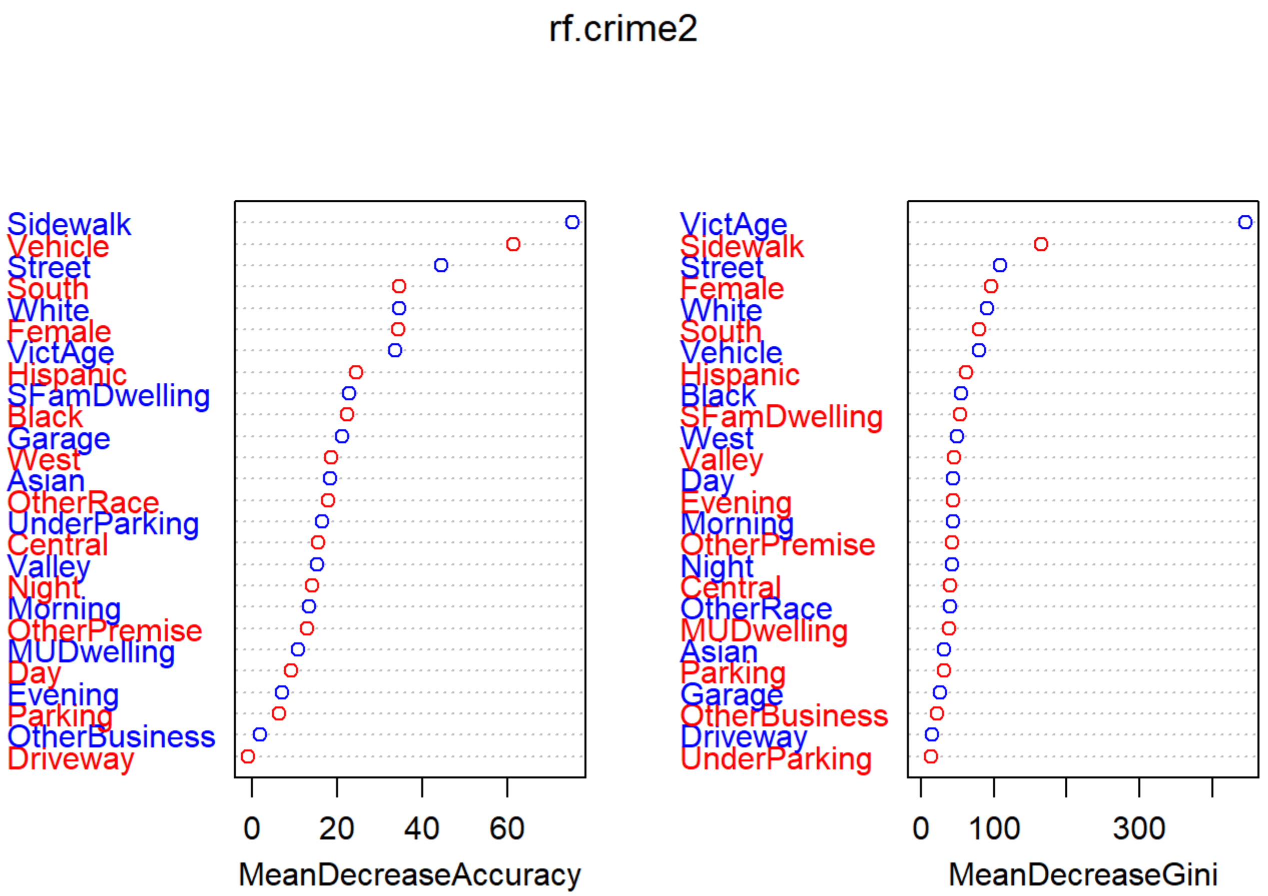
```
cat("The misclassification rate for the testing data is",(1126+52587)/(114286+1126+52587+2558
6))
```

```
## The misclassification rate for the testing data is 0.2774647
```

```
#Without weapons
rf.crime2 = randomForest(Severity~.-Weapon, data = traindatafinal, mtry = 5, importance = T)
rf.crime2
```

```
##
## Call:
##  randomForest(formula = Severity ~ . - Weapon, data = traindatafinal,      mtry = 5, impor
tance = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 32.37%
## Confusion matrix:
##            Non-Severe Severe class.error
## Non-Severe       3493   1507      0.3014
## Severe           1730   3270      0.3460
```

```
varImpPlot(rf.crime2, col = c('red', 'blue'))
```



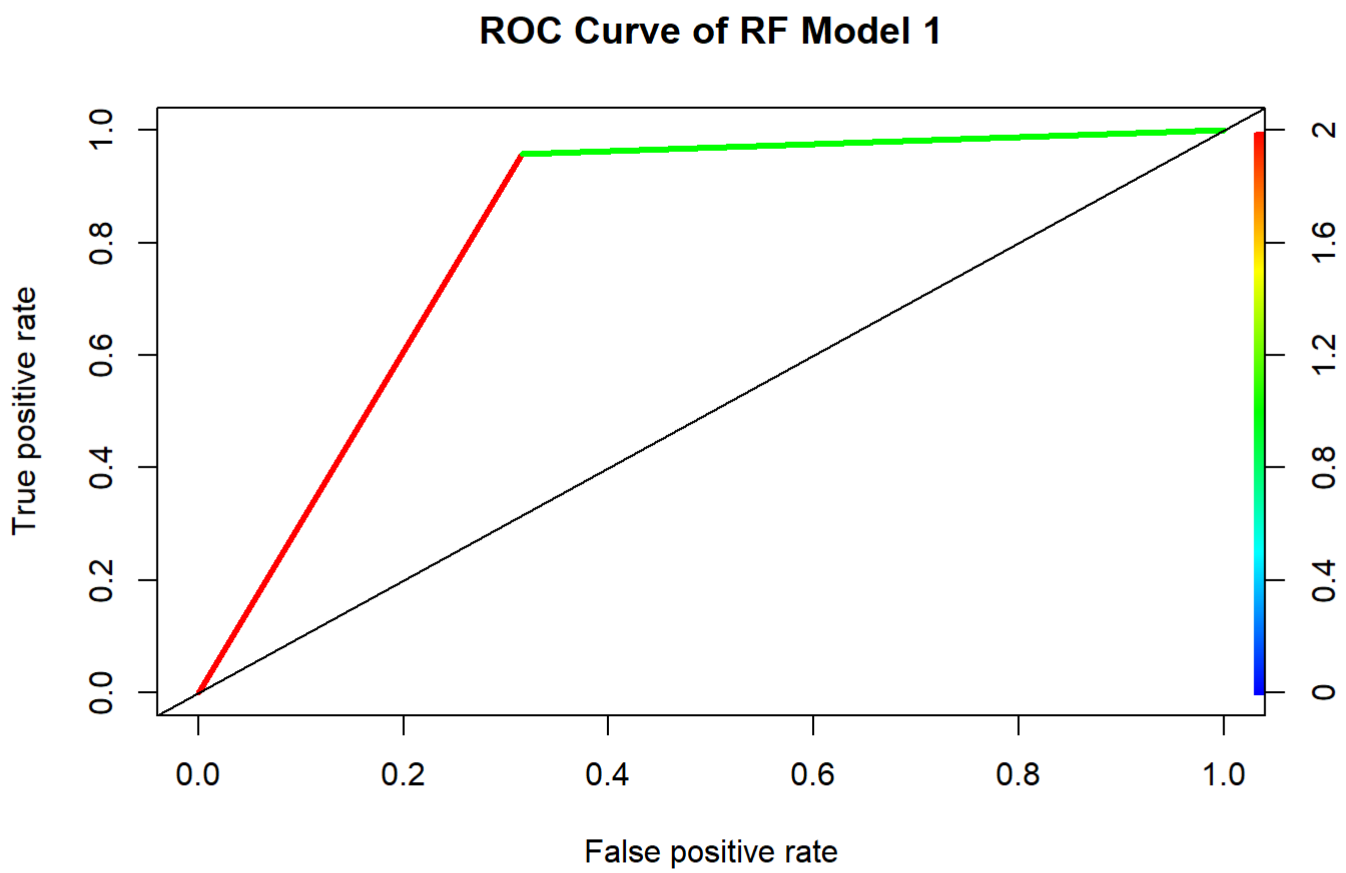

```
test.rf2 = predict(rf.crime2, newdata = testdatafinal, type = 'class')
table(test.rf2, testdatafinal$Severity)
```

```
##
## test.rf2     Non-Severe Severe
##   Non-Severe     114686   9090
##   Severe          52187  17622
```

```
cat("The misclassification rate for the testing data is",(9090+52187)/(114686+9090+52187+1762
2))
```

```
## The misclassification rate for the testing data is 0.316538
```

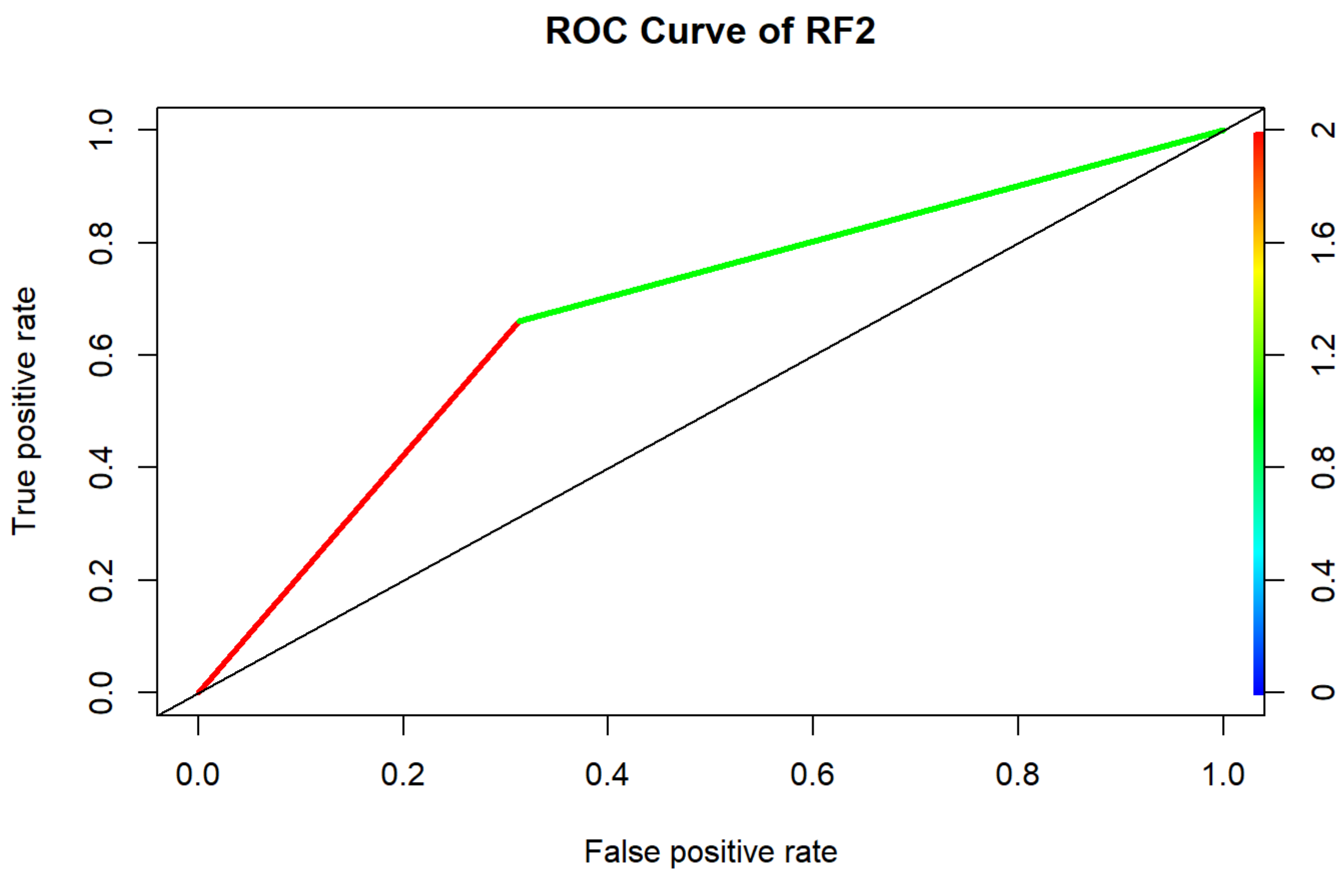
```
#ROC Curves for random forests
#With weapons
pred.rf1 = predict(rf.crime1, testdatafinal)
prediction.rf1 = prediction((as.numeric(pred.rf1) - 1), (as.numeric(testdatafinal$Severity)-1
))
rocrf1=performance(prediction.rf1, measure = "tpr", x.measure = "fpr")
plot(rocrf1, lwd=3, colorkey=T, colorize=T, main="ROC Curve of RF Model 1")
abline(0,1)
```




```
performance(prediction.rf1, measure = "auc")@y.values
```

```
## [[1]]
## [1] 0.8213784
```

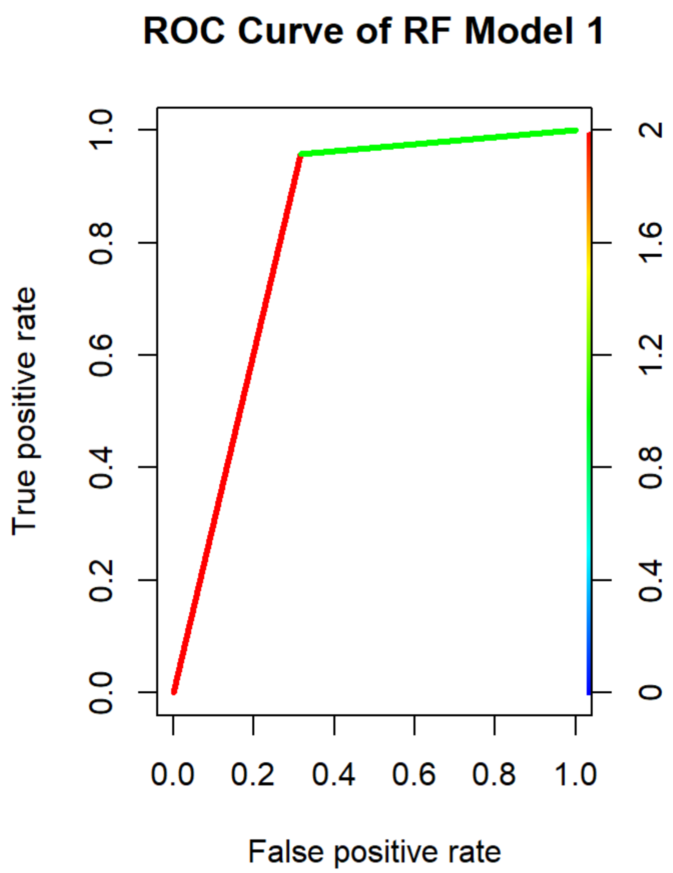
```
#Without weapons
pred.rf2 = predict(rf.crime2, testdatafinal)
prediction.rf2 = prediction((as.numeric(pred.rf2) - 1), (as.numeric(testdatafinal$Severity)-1
))
rocrf2=performance(prediction.rf2, measure = "tpr", x.measure = "fpr")
plot(rocrf2, lwd=3, colorkey=T, colorize=T, main='ROC Curve of RF2')
abline(0,1)
```

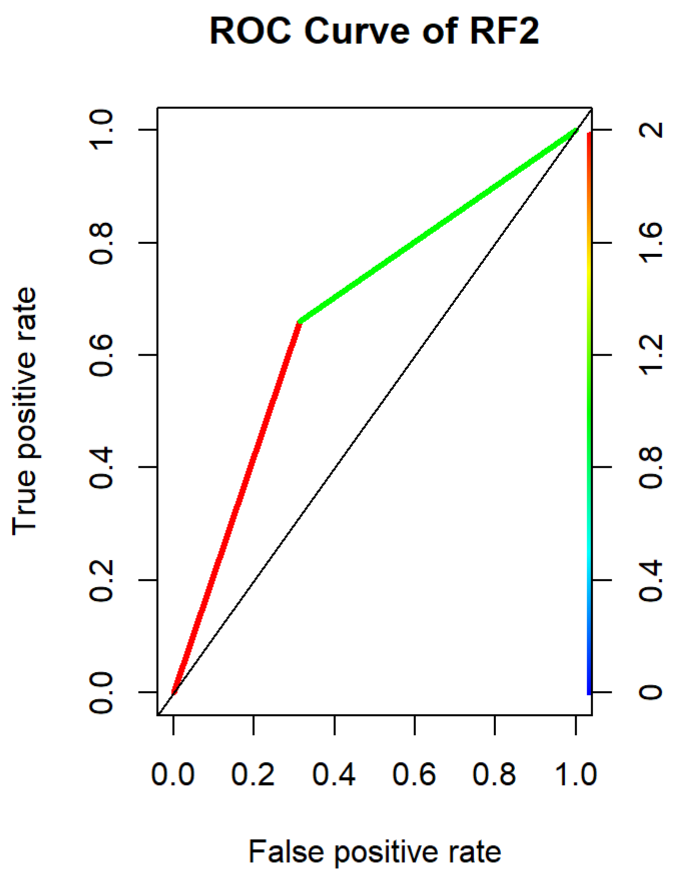



```
performance(prediction.rf2, measure = "auc")@y.values
```

```
## [[1]]
## [1] 0.6736438
```

```
#Plotting both curves side by side
par(mfrow=c(1,2))
plot(rocrf1, lwd=3, colorkey=T, colorize=T, main="ROC Curve of RF Model 1")
plot(rocrf2, lwd=3, colorkey=T, colorize=T, main='ROC Curve of RF2')
abline(0,1)
```





Logistic regression modelling

Converting factors to numeric

```
#Converting factors to numeric
traindatafinal$Weapon <- as.numeric(traindatafinal$Weapon) - 1
traindatafinal$Female <- as.numeric(traindatafinal$Female) - 1
traindatafinal$SFamDwelling = as.numeric(traindatafinal$SFamDwelling) - 1
traindatafinal$Street = as.numeric(traindatafinal$Street) - 1
traindatafinal$MUDwelling = as.numeric(traindatafinal$MUDwelling) - 1
traindatafinal$Parking = as.numeric(traindatafinal$Parking) - 1
traindatafinal$Sidewalk = as.numeric(traindatafinal$Sidewalk) - 1
traindatafinal$Vehicle = as.numeric(traindatafinal$Vehicle) - 1
traindatafinal$OtherBusiness = as.numeric(traindatafinal$OtherBusiness) - 1
traindatafinal$Garage = as.numeric(traindatafinal$Garage) - 1
traindatafinal$Driveway = as.numeric(traindatafinal$Driveway) - 1
traindatafinal$UnderParking = as.numeric(traindatafinal$UnderParking) - 1
traindatafinal$OtherPremise = as.numeric(traindatafinal$OtherPremise) - 1
traindatafinal$Asian = as.numeric(traindatafinal$Asian) - 1
traindatafinal$Black = as.numeric(traindatafinal$Black) - 1
traindatafinal$Hispanic = as.numeric(traindatafinal$Hispanic) - 1
traindatafinal$White = as.numeric(traindatafinal$White) - 1
traindatafinal$OtherRace = as.numeric(traindatafinal$OtherRace) - 1
traindatafinal$Morning = as.numeric(traindatafinal$Morning) - 1
traindatafinal$Day = as.numeric(traindatafinal$Day) - 1
traindatafinal$Evening = as.numeric(traindatafinal$Evening) - 1
traindatafinal$Night = as.numeric(traindatafinal$Night) - 1
traindatafinal$Valley = as.numeric(traindatafinal$Valley) - 1
traindatafinal$West = as.numeric(traindatafinal$West) - 1
traindatafinal$South = as.numeric(traindatafinal$South) - 1
traindatafinal$Central = as.numeric(traindatafinal$Central) - 1

testdatafinal$Weapon <- as.numeric(testdatafinal$Weapon) - 1
testdatafinal$Female <- as.numeric(testdatafinal$Female) - 1
testdatafinal$SFamDwelling = as.numeric(testdatafinal$SFamDwelling) - 1
testdatafinal$Street = as.numeric(testdatafinal$Street) - 1
testdatafinal$MUDwelling = as.numeric(testdatafinal$MUDwelling) - 1
testdatafinal$Parking = as.numeric(testdatafinal$Parking) - 1
testdatafinal$Sidewalk = as.numeric(testdatafinal$Sidewalk) - 1
testdatafinal$Vehicle = as.numeric(testdatafinal$Vehicle) - 1
testdatafinal$OtherBusiness = as.numeric(testdatafinal$OtherBusiness) - 1
testdatafinal$Garage = as.numeric(testdatafinal$Garage) - 1
testdatafinal$Driveway = as.numeric(testdatafinal$Driveway) - 1
testdatafinal$UnderParking = as.numeric(testdatafinal$UnderParking) - 1
testdatafinal$OtherPremise = as.numeric(testdatafinal$OtherPremise) - 1
testdatafinal$Asian = as.numeric(testdatafinal$Asian) - 1
testdatafinal$Black = as.numeric(testdatafinal$Black) - 1
testdatafinal$Hispanic = as.numeric(testdatafinal$Hispanic) - 1
testdatafinal$White = as.numeric(testdatafinal$White) - 1
testdatafinal$OtherRace = as.numeric(testdatafinal$OtherRace) - 1
testdatafinal$Morning = as.numeric(testdatafinal$Morning) - 1
testdatafinal$Day = as.numeric(testdatafinal$Day) - 1
testdatafinal$Evening = as.numeric(testdatafinal$Evening) - 1
testdatafinal$Night = as.numeric(testdatafinal$Night) - 1
testdatafinal$Valley = as.numeric(testdatafinal$Valley) - 1
testdatafinal$West = as.numeric(testdatafinal$West) - 1
testdatafinal$South = as.numeric(testdatafinal$South) - 1
testdatafinal$Central = as.numeric(testdatafinal$Central) - 1
```

Modelling

```
#Modelling with weapon
names(crime)
```

```
##  [1] "VictAge"       "Severity"      "Female"        "Weapon"
##  [5] "SFamDwelling"  "Street"        "MUDwelling"    "Parking"
##  [9] "Sidewalk"      "Vehicle"       "OtherBusiness" "Garage"
## [13] "Driveway"      "UnderParking"  "OtherPremise"  "Asian"
## [17] "Black"         "Hispanic"      "White"         "OtherRace"
## [21] "Morning"       "Day"           "Evening"       "Night"
## [25] "Valley"        "West"          "South"         "Central"
```

```
logistic.crime=glm(Severity~VictAge+Female+Weapon+SFamDwelling+Street+MUDwelling+Parking+Side
walk+Vehicle+OtherBusiness+Garage+Driveway+UnderParking+Asian+Black+Hispanic+White+Morning+Da
y+Night+Central+South+West, data=traindatafinal,family=binomial)
summary(logistic.crime)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + Weapon + SFamDwelling +
##     Street + MUDwelling + Parking + Sidewalk + Vehicle + OtherBusiness +
##     Garage + Driveway + UnderParking + Asian + Black + Hispanic +
##     White + Morning + Day + Night + Central + South + West, family = binomial,
##     data = traindatafinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4120  -0.1721   0.1200   0.6726   3.2621
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.179485   0.225994 -18.494  < 2e-16 ***
## VictAge       -0.009093   0.001961  -4.636 3.55e-06 ***
## Female        -0.689526   0.060966 -11.310  < 2e-16 ***
## Weapon         5.405203   0.166329  32.497  < 2e-16 ***
## SFamDwelling  -0.393417   0.093923  -4.189 2.81e-05 ***
## Street         0.696129   0.093517   7.444 9.78e-14 ***
## MUDwelling    -0.317888   0.097302  -3.267 0.001087 **
## Parking        0.198701   0.130885   1.518 0.128980
## Sidewalk       0.705844   0.114886   6.144 8.05e-10 ***
## Vehicle       -1.137258   0.238119  -4.776 1.79e-06 ***
## OtherBusiness -0.146748   0.190264  -0.771 0.440536
## Garage        -0.443673   0.411425  -1.078 0.280864
## Driveway      -0.054741   0.260583  -0.210 0.833611
## UnderParking  -0.731382   0.531531  -1.376 0.168824
## Asian          0.275102   0.202796   1.357 0.174926
## Black          0.564350   0.130821   4.314 1.60e-05 ***
## Hispanic       0.333269   0.119066   2.799 0.005126 **
## White          0.108325   0.128138   0.845 0.397899
## Morning       -0.277507   0.083030  -3.342 0.000831 ***
## Day           -0.217064   0.072717  -2.985 0.002835 **
## Night          0.281537   0.090151   3.123 0.001790 **
## Central        0.184718   0.082861   2.229 0.025797 *
## South          0.533835   0.086973   6.138 8.36e-10 ***
## West          -0.155316   0.085925  -1.808 0.070674 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13862.9  on 9999  degrees of freedom
## Residual deviance:  7296.5  on 9976  degrees of freedom
## AIC: 7344.5
##
## Number of Fisher Scoring iterations: 7
```

```
logistic.crime2=glm(Severity~VictAge+Female+Weapon+SFamDwelling+Street+MUDwelling+Parking+Sidewalk+Vehicle+OtherBusiness+Garage+UnderParking+Asian+Black+Hispanic+White+Morning+Day+Night+Central+South+West, data=traindatafinal,family=binomial)
summary(logistic.crime2)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + Weapon + SFamDwelling +
##     Street + MUDwelling + Parking + Sidewalk + Vehicle + OtherBusiness +
##     Garage + UnderParking + Asian + Black + Hispanic + White +
##     Morning + Day + Night + Central + South + West, family = binomial,
##     data = traindatafinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4121  -0.1719   0.1200   0.6719   3.2624
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.183720   0.225089 -18.587  < 2e-16 ***
## VictAge       -0.009098   0.001961  -4.639 3.49e-06 ***
## Female        -0.689833   0.060949 -11.318  < 2e-16 ***
## Weapon         5.405926   0.166296  32.508  < 2e-16 ***
## SFamDwelling  -0.389866   0.092365  -4.221 2.43e-05 ***
## Street         0.699558   0.092056   7.599 2.98e-14 ***
## MUDwelling    -0.314424   0.095872  -3.280 0.001039 **
## Parking        0.202162   0.129829   1.557 0.119438
## Sidewalk       0.709171   0.113768   6.233 4.56e-10 ***
## Vehicle       -1.133756   0.237529  -4.773 1.81e-06 ***
## OtherBusiness -0.143389   0.189581  -0.756 0.449441
## Garage        -0.440323   0.411131  -1.071 0.284168
## UnderParking  -0.727857   0.531288  -1.370 0.170691
## Asian          0.275453   0.202788   1.358 0.174359
## Black          0.564381   0.130821   4.314 1.60e-05 ***
## Hispanic       0.333181   0.119064   2.798 0.005137 **
## White          0.108472   0.128135   0.847 0.397249
## Morning       -0.277280   0.083022  -3.340 0.000838 ***
## Day           -0.216938   0.072714  -2.983 0.002850 **
## Night          0.281866   0.090137   3.127 0.001765 **
## Central        0.185217   0.082828   2.236 0.025340 *
## South          0.534120   0.086964   6.142 8.16e-10 ***
## West          -0.154825   0.085893  -1.803 0.071461 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13862.9  on 9999  degrees of freedom
## Residual deviance:  7296.6  on 9977  degrees of freedom
## AIC: 7342.6
##
## Number of Fisher Scoring iterations: 7
```

```
logistic.crime3=glm(Severity~VictAge+Female+Weapon+SFamDwelling+Street+MUDwelling+Parking+Sidewalk+Vehicle+Garage+UnderParking+Asian+Black+Hispanic+White+Morning+Day+Night+Central+South+West, data=traindatafinal,family=binomial)
summary(logistic.crime3)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + Weapon + SFamDwelling +
##     Street + MUDwelling + Parking + Sidewalk + Vehicle + Garage +
##     UnderParking + Asian + Black + Hispanic + White + Morning +
##     Day + Night + Central + South + West, family = binomial,
##     data = traindatafinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4124  -0.1717   0.1198   0.6723   3.2621
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.202922   0.223662 -18.791  < 2e-16 ***
## VictAge      -0.009116   0.001961  -4.650 3.32e-06 ***
## Female       -0.688894   0.060929 -11.307  < 2e-16 ***
## Weapon        5.406315   0.166294  32.511  < 2e-16 ***
## SFamDwelling -0.374938   0.090129  -4.160 3.18e-05 ***
## Street        0.715001   0.089658   7.975 1.53e-15 ***
## MUDwelling   -0.299467   0.093712  -3.196 0.001395 **
## Parking       0.217804   0.128102   1.700 0.089084 .
## Sidewalk      0.724936   0.111758   6.487 8.78e-11 ***
## Vehicle      -1.118827   0.236670  -4.727 2.27e-06 ***
## Garage       -0.425134   0.410676  -1.035 0.300573
## UnderParking -0.712349   0.530894  -1.342 0.179664
## Asian         0.279236   0.202709   1.378 0.168351
## Black         0.571291   0.130515   4.377 1.20e-05 ***
## Hispanic      0.338393   0.118880   2.847 0.004420 **
## White         0.115141   0.127842   0.901 0.367776
## Morning      -0.277843   0.083013  -3.347 0.000817 ***
## Day          -0.217309   0.072708  -2.989 0.002801 **
## Night         0.283212   0.090115   3.143 0.001674 **
## Central       0.183426   0.082795   2.215 0.026731 *
## South         0.531766   0.086904   6.119 9.42e-10 ***
## West         -0.157780   0.085795  -1.839 0.065909 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13862.9  on 9999  degrees of freedom
## Residual deviance:  7297.2  on 9978  degrees of freedom
## AIC: 7341.2
##
## Number of Fisher Scoring iterations: 7
```

```
logistic.crime4=glm(Severity~VictAge+Female+Weapon+SFamDwelling+Street+MUDwelling+Parking+Sid
ewalk+Vehicle+Garage+UnderParking+Asian+Black+Hispanic+Morning+Day+Night+Central+South+West,
data=traindatafinal,family=binomial)
summary(logistic.crime4)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + Weapon + SFamDwelling +
##     Street + MUDwelling + Parking + Sidewalk + Vehicle + Garage +
##     UnderParking + Asian + Black + Hispanic + Morning + Day +
##     Night + Central + South + West, family = binomial, data = traindatafinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4128  -0.1719   0.1201   0.6715   3.2625
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.124496   0.205937 -20.028  < 2e-16 ***
## VictAge      -0.009088   0.001960  -4.636 3.56e-06 ***
## Female       -0.687167   0.060888 -11.286  < 2e-16 ***
## Weapon        5.406036   0.166292  32.509  < 2e-16 ***
## SFamDwelling -0.372912   0.090082  -4.140 3.48e-05 ***
## Street        0.717603   0.089608   8.008 1.16e-15 ***
## MUDwelling   -0.296368   0.093638  -3.165 0.001551 **
## Parking       0.217942   0.128069   1.702 0.088800 .
## Sidewalk      0.727346   0.111714   6.511 7.47e-11 ***
## Vehicle      -1.115777   0.236688  -4.714 2.43e-06 ***
## Garage       -0.412635   0.410332  -1.006 0.314602
## UnderParking -0.721142   0.530780  -1.359 0.174259
## Asian         0.194602   0.179728   1.083 0.278918
## Black         0.486315   0.090488   5.374 7.69e-08 ***
## Hispanic      0.254361   0.074077   3.434 0.000595 ***
## Morning      -0.277771   0.082999  -3.347 0.000818 ***
## Day          -0.218479   0.072690  -3.006 0.002650 **
## Night         0.283769   0.090113   3.149 0.001638 **
## Central       0.186686   0.082701   2.257 0.023985 *
## South         0.535050   0.086817   6.163 7.14e-10 ***
## West         -0.153769   0.085669  -1.795 0.072666 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13863  on 9999  degrees of freedom
## Residual deviance:  7298  on 9979  degrees of freedom
## AIC: 7340
##
## Number of Fisher Scoring iterations: 7
```

```
logistic.crime5=glm(Severity~VictAge+Female+Weapon+SFamDwelling+Street+MUDwelling+Parking+Sid
ewalk+Vehicle+UnderParking+Asian+Black+Hispanic+Morning+Day+Night+Central+South+West, data=tr
aindatafinal,family=binomial)
summary(logistic.crime5)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + Weapon + SFamDwelling +
##     Street + MUDwelling + Parking + Sidewalk + Vehicle + UnderParking +
##     Asian + Black + Hispanic + Morning + Day + Night + Central +
##     South + West, family = binomial, data = traindatafinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4123  -0.1719   0.1203   0.6718   3.2648
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.14396    0.20502 -20.213  < 2e-16 ***
## VictAge      -0.00904    0.00196  -4.613 3.96e-06 ***
## Female       -0.68776    0.06088 -11.297  < 2e-16 ***
## Weapon        5.41500    0.16611  32.599  < 2e-16 ***
## SFamDwelling -0.36450    0.08965  -4.066 4.79e-05 ***
## Street        0.72644    0.08914   8.149 3.65e-16 ***
## MUDwelling   -0.28780    0.09321  -3.088 0.002018 **
## Parking       0.22647    0.12778   1.772 0.076330 .
## Sidewalk      0.73577    0.11137   6.607 3.93e-11 ***
## Vehicle      -1.10679    0.23655  -4.679 2.88e-06 ***
## UnderParking -0.71172    0.53088  -1.341 0.180038
## Asian         0.19182    0.17959   1.068 0.285454
## Black         0.48808    0.09047   5.395 6.85e-08 ***
## Hispanic      0.25591    0.07405   3.456 0.000549 ***
## Morning      -0.27589    0.08297  -3.325 0.000884 ***
## Day          -0.21701    0.07267  -2.986 0.002823 **
## Night         0.28327    0.09011   3.144 0.001669 **
## Central       0.18586    0.08270   2.247 0.024613 *
## South         0.53358    0.08681   6.147 7.90e-10 ***
## West         -0.15616    0.08562  -1.824 0.068186 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13863  on 9999  degrees of freedom
## Residual deviance:  7299  on 9980  degrees of freedom
## AIC: 7339
##
## Number of Fisher Scoring iterations: 7
```

```
logistic.crime6=glm(Severity~VictAge+Female+Weapon+SFamDwelling+Street+MUDwelling+Parking+Sid
ewalk+Vehicle+UnderParking+Black+Hispanic+Morning+Day+Night+Central+South+West, data=traindat
afinal,family=binomial)
summary(logistic.crime6)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + Weapon + SFamDwelling +
##     Street + MUDwelling + Parking + Sidewalk + Vehicle + UnderParking +
##     Black + Hispanic + Morning + Day + Night + Central + South +
##     West, family = binomial, data = traindatafinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4113  -0.1719   0.1203   0.6717   3.2641
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.125242   0.204259 -20.196  < 2e-16 ***
## VictAge      -0.009012   0.001959  -4.599 4.24e-06 ***
## Female       -0.685420   0.060829 -11.268  < 2e-16 ***
## Weapon        5.410736   0.166014  32.592  < 2e-16 ***
## SFamDwelling -0.364767   0.089633  -4.070 4.71e-05 ***
## Street        0.725363   0.089130   8.138 4.01e-16 ***
## MUDwelling   -0.288473   0.093195  -3.095 0.001966 **
## Parking       0.229261   0.127764   1.794 0.072748 .
## Sidewalk      0.735960   0.111374   6.608 3.89e-11 ***
## Vehicle      -1.106389   0.236534  -4.678 2.90e-06 ***
## UnderParking -0.713062   0.531119  -1.343 0.179412
## Black         0.468280   0.088609   5.285 1.26e-07 ***
## Hispanic      0.236787   0.071944   3.291 0.000997 ***
## Morning      -0.275480   0.082978  -3.320 0.000901 ***
## Day          -0.217096   0.072660  -2.988 0.002810 **
## Night         0.281737   0.090087   3.127 0.001764 **
## Central       0.189609   0.082612   2.295 0.021723 *
## South         0.536356   0.086746   6.183 6.29e-10 ***
## West         -0.150746   0.085476  -1.764 0.077799 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13862.9  on 9999  degrees of freedom
## Residual deviance:  7300.1  on 9981  degrees of freedom
## AIC: 7338.1
##
## Number of Fisher Scoring iterations: 7
```

```
logistic.crime7=glm(Severity~VictAge+Female+Weapon+SFamDwelling+Street+MUDwelling+Parking+Sidewalk+Vehicle+Black+Hispanic+Morning+Day+Night+Central+South+West, data=traindatafinal,family=binomial)
summary(logistic.crime7)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + Weapon + SFamDwelling +
##     Street + MUDwelling + Parking + Sidewalk + Vehicle + Black +
##     Hispanic + Morning + Day + Night + Central + South + West,
##     family = binomial, data = traindatafinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4111  -0.1719   0.1204   0.6724   3.2665
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.141551   0.203798 -20.322  < 2e-16 ***
## VictAge      -0.008989   0.001959  -4.588 4.47e-06 ***
## Female       -0.686300   0.060821 -11.284  < 2e-16 ***
## Weapon        5.417485   0.165971  32.641  < 2e-16 ***
## SFamDwelling -0.356024   0.089352  -3.985 6.76e-05 ***
## Street        0.734148   0.088851   8.263  < 2e-16 ***
## MUDwelling   -0.279482   0.092915  -3.008 0.002630 **
## Parking       0.238044   0.127580   1.866 0.062065 .
## Sidewalk      0.744257   0.111170   6.695 2.16e-11 ***
## Vehicle      -1.097086   0.236449  -4.640 3.49e-06 ***
## Black         0.466230   0.088579   5.263 1.41e-07 ***
## Hispanic      0.234827   0.071917   3.265 0.001094 **
## Morning      -0.273415   0.082954  -3.296 0.000981 ***
## Day          -0.215625   0.072637  -2.969 0.002992 **
## Night         0.281122   0.090071   3.121 0.001802 **
## Central       0.191540   0.082583   2.319 0.020376 *
## South         0.539120   0.086707   6.218 5.05e-10 ***
## West         -0.150585   0.085451  -1.762 0.078028 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13862.9  on 9999  degrees of freedom
## Residual deviance:  7301.9  on 9982  degrees of freedom
## AIC: 7337.9
##
## Number of Fisher Scoring iterations: 7
```

```
contrasts(testdatafinal$Severity)
```

```
##            Severe
## Non-Severe      0
## Severe          1
```

```
logistic.test1= predict(logistic.crime7, testdatafinal, type = 'response')
pred.crimeseverity1= rep('Non-Severe',193585)
pred.crimeseverity1[logistic.test1 > 0.5] = 'Severe'
table(pred.crimeseverity1, testdatafinal$Severity)
```

```
##
## pred.crimeseverity1 Non-Severe Severe
##          Non-Severe     114370   1261
##          Severe          52503  25451
```

```
cat("The misclassification rate for the testing data is",(1261+52503)/(114370+1261+52503+2545
1))
```

```
## The misclassification rate for the testing data is 0.2777281
```

```
#Logistic regression model without weapon
logistic.newcrime=glm(Severity~VictAge+Female+SFamDwelling+Street+MUDwelling+Parking+Sidewalk
+Vehicle+OtherBusiness+Garage+Driveway+UnderParking+Asian+Black+Hispanic+White+Morning+Day+Ni
ght+Central+South+West, data=traindatafinal,family=binomial)
summary(logistic.newcrime)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + SFamDwelling + Street +
##     MUDwelling + Parking + Sidewalk + Vehicle + OtherBusiness +
##     Garage + Driveway + UnderParking + Asian + Black + Hispanic +
##     White + Morning + Day + Night + Central + South + West, family = binomial,
##     data = traindatafinal)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32678  -1.01022   0.06287   0.98346   2.43203
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.111921   0.119779   0.934 0.350101
## VictAge       -0.013332   0.001504  -8.864  < 2e-16 ***
## Female        -0.492552   0.045980 -10.712  < 2e-16 ***
## SFamDwelling  -0.496763   0.072787  -6.825 8.80e-12 ***
## Street         0.407599   0.066962   6.087 1.15e-09 ***
## MUDwelling    -0.279627   0.076936  -3.635 0.000278 ***
## Parking        0.019980   0.095807   0.209 0.834807
## Sidewalk       1.268250   0.096340  13.164  < 2e-16 ***
## Vehicle       -2.208040   0.183403 -12.039  < 2e-16 ***
## OtherBusiness -0.254896   0.143727  -1.773 0.076150 .
## Garage        -1.963822   0.273862  -7.171 7.45e-13 ***
## Driveway      -0.601406   0.184413  -3.261 0.001109 **
## UnderParking  -1.981724   0.384195  -5.158 2.49e-07 ***
## Asian         -0.098235   0.144740  -0.679 0.497329
## Black          0.924771   0.096955   9.538  < 2e-16 ***
## Hispanic       0.760885   0.088703   8.578  < 2e-16 ***
## White          0.067367   0.094464   0.713 0.475750
## Morning       -0.397200   0.063742  -6.231 4.62e-10 ***
## Day           -0.291028   0.055543  -5.240 1.61e-07 ***
## Night          0.155311   0.066976   2.319 0.020401 *
## Central        0.322449   0.062906   5.126 2.96e-07 ***
## South          0.674813   0.065465  10.308  < 2e-16 ***
## West          -0.114717   0.064968  -1.766 0.077441 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13863  on 9999  degrees of freedom
## Residual deviance: 11852  on 9977  degrees of freedom
## AIC: 11898
##
## Number of Fisher Scoring iterations: 4
```

```
logistic.newcrime2=glm(Severity~VictAge+Female+SFamDwelling+Street+MUDwelling+Sidewalk+Vehicl
e+OtherBusiness+Garage+Driveway+UnderParking+Asian+Black+Hispanic+White+Morning+Day+Night+Cen
tral+South+West, data=traindatafinal,family=binomial)
summary(logistic.newcrime2)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + SFamDwelling + Street +
##     MUDwelling + Sidewalk + Vehicle + OtherBusiness + Garage +
##     Driveway + UnderParking + Asian + Black + Hispanic + White +
##     Morning + Day + Night + Central + South + West, family = binomial,
##     data = traindatafinal)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32682  -1.01010   0.06288   0.98349   2.43209
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.118258   0.115861   1.021 0.307404
## VictAge       -0.013336   0.001504  -8.867  < 2e-16 ***
## Female        -0.492518   0.045979 -10.712  < 2e-16 ***
## SFamDwelling  -0.502240   0.067884  -7.399 1.38e-13 ***
## Street         0.402185   0.061724   6.516 7.23e-11 ***
## MUDwelling    -0.285039   0.072427  -3.936 8.30e-05 ***
## Sidewalk       1.262938   0.092912  13.593  < 2e-16 ***
## Vehicle       -2.213509   0.181520 -12.194  < 2e-16 ***
## OtherBusiness -0.260272   0.141397  -1.841 0.065663 .
## Garage        -1.969205   0.272643  -7.223 5.10e-13 ***
## Driveway      -0.606947   0.182490  -3.326 0.000881 ***
## UnderParking  -1.987161   0.383309  -5.184 2.17e-07 ***
## Asian         -0.098382   0.144738  -0.680 0.496679
## Black          0.924537   0.096949   9.536  < 2e-16 ***
## Hispanic       0.760842   0.088703   8.577  < 2e-16 ***
## White          0.066953   0.094443   0.709 0.478369
## Morning       -0.397534   0.063722  -6.239 4.42e-10 ***
## Day           -0.291283   0.055530  -5.246 1.56e-07 ***
## Night          0.154951   0.066953   2.314 0.020650 *
## Central        0.321831   0.062835   5.122 3.03e-07 ***
## South          0.674395   0.065434  10.307  < 2e-16 ***
## West          -0.115340   0.064899  -1.777 0.075533 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13863  on 9999  degrees of freedom
## Residual deviance: 11852  on 9978  degrees of freedom
## AIC: 11896
##
## Number of Fisher Scoring iterations: 4
```

```
logistic.newcrime3=glm(Severity~VictAge+Female+SFamDwelling+Street+MUDwelling+Sidewalk+Vehicl
e+OtherBusiness+Garage+Driveway+UnderParking+Black+Hispanic+White+Morning+Day+Night+Central+S
outh+West, data=traindatafinal,family=binomial)
summary(logistic.newcrime3)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + SFamDwelling + Street +
##     MUDwelling + Sidewalk + Vehicle + OtherBusiness + Garage +
##     Driveway + UnderParking + Black + Hispanic + White + Morning +
##     Day + Night + Central + South + West, family = binomial,
##     data = traindatafinal)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32718  -1.00987   0.06278   0.98359   2.43290
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.089480   0.108006   0.828 0.407405
## VictAge       -0.013342   0.001504  -8.871  < 2e-16 ***
## Female        -0.493188   0.045966 -10.729  < 2e-16 ***
## SFamDwelling  -0.502085   0.067886  -7.396 1.40e-13 ***
## Street         0.401747   0.061718   6.509 7.54e-11 ***
## MUDwelling    -0.285277   0.072425  -3.939 8.18e-05 ***
## Sidewalk       1.263099   0.092903  13.596  < 2e-16 ***
## Vehicle       -2.214383   0.181501 -12.200  < 2e-16 ***
## OtherBusiness -0.257728   0.141360  -1.823 0.068273 .
## Garage        -1.969756   0.272628  -7.225 5.01e-13 ***
## Driveway      -0.607799   0.182477  -3.331 0.000866 ***
## UnderParking  -1.991203   0.383179  -5.197 2.03e-07 ***
## Black          0.956178   0.085264  11.214  < 2e-16 ***
## Hispanic       0.792059   0.076133  10.404  < 2e-16 ***
## White          0.098136   0.082781   1.185 0.235827
## Morning       -0.398211   0.063709  -6.250 4.09e-10 ***
## Day           -0.291554   0.055528  -5.251 1.52e-07 ***
## Night          0.155486   0.066951   2.322 0.020213 *
## Central        0.319222   0.062718   5.090 3.58e-07 ***
## South          0.672305   0.065364  10.286  < 2e-16 ***
## West          -0.117936   0.064791  -1.820 0.068719 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13863  on 9999  degrees of freedom
## Residual deviance: 11852  on 9979  degrees of freedom
## AIC: 11894
##
## Number of Fisher Scoring iterations: 4
```

```
logistic.newcrime4=glm(Severity~VictAge+Female+SFamDwelling+Street+MUDwelling+Sidewalk+Vehicl
e+OtherBusiness+Garage+Driveway+UnderParking+Black+Hispanic+Morning+Day+Night+Central+South+W
est, data=traindatafinal,family=binomial)
summary(logistic.newcrime4)
```

```
##
## Call:
## glm(formula = Severity ~ VictAge + Female + SFamDwelling + Street +
##     MUDwelling + Sidewalk + Vehicle + OtherBusiness + Garage +
##     Driveway + UnderParking + Black + Hispanic + Morning + Day +
##     Night + Central + South + West, family = binomial, data = traindatafinal)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32719  -1.01137   0.06453   0.98423   2.44610
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.147883   0.096013   1.540 0.123503
## VictAge       -0.013258   0.001502  -8.827  < 2e-16 ***
## Female        -0.491371   0.045936 -10.697  < 2e-16 ***
## SFamDwelling  -0.500629   0.067870  -7.376 1.63e-13 ***
## Street         0.403849   0.061689   6.547 5.89e-11 ***
## MUDwelling    -0.283944   0.072407  -3.921 8.80e-05 ***
## Sidewalk       1.264885   0.092878  13.619  < 2e-16 ***
## Vehicle       -2.213908   0.181485 -12.199  < 2e-16 ***
## OtherBusiness -0.263502   0.141191  -1.866 0.062002 .
## Garage        -1.966365   0.272618  -7.213 5.48e-13 ***
## Driveway      -0.604621   0.182494  -3.313 0.000923 ***
## UnderParking  -1.991636   0.383054  -5.199 2.00e-07 ***
## Black          0.891770   0.065516  13.611  < 2e-16 ***
## Hispanic       0.728293   0.053649  13.575  < 2e-16 ***
## Morning       -0.397718   0.063701  -6.244 4.28e-10 ***
## Day           -0.291707   0.055523  -5.254 1.49e-07 ***
## Night          0.156559   0.066948   2.339 0.019360 *
## Central        0.319023   0.062711   5.087 3.63e-07 ***
## South          0.672737   0.065363  10.292  < 2e-16 ***
## West          -0.116923   0.064782  -1.805 0.071092 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13863  on 9999  degrees of freedom
## Residual deviance: 11853  on 9980  degrees of freedom
## AIC: 11893
##
## Number of Fisher Scoring iterations: 4
```

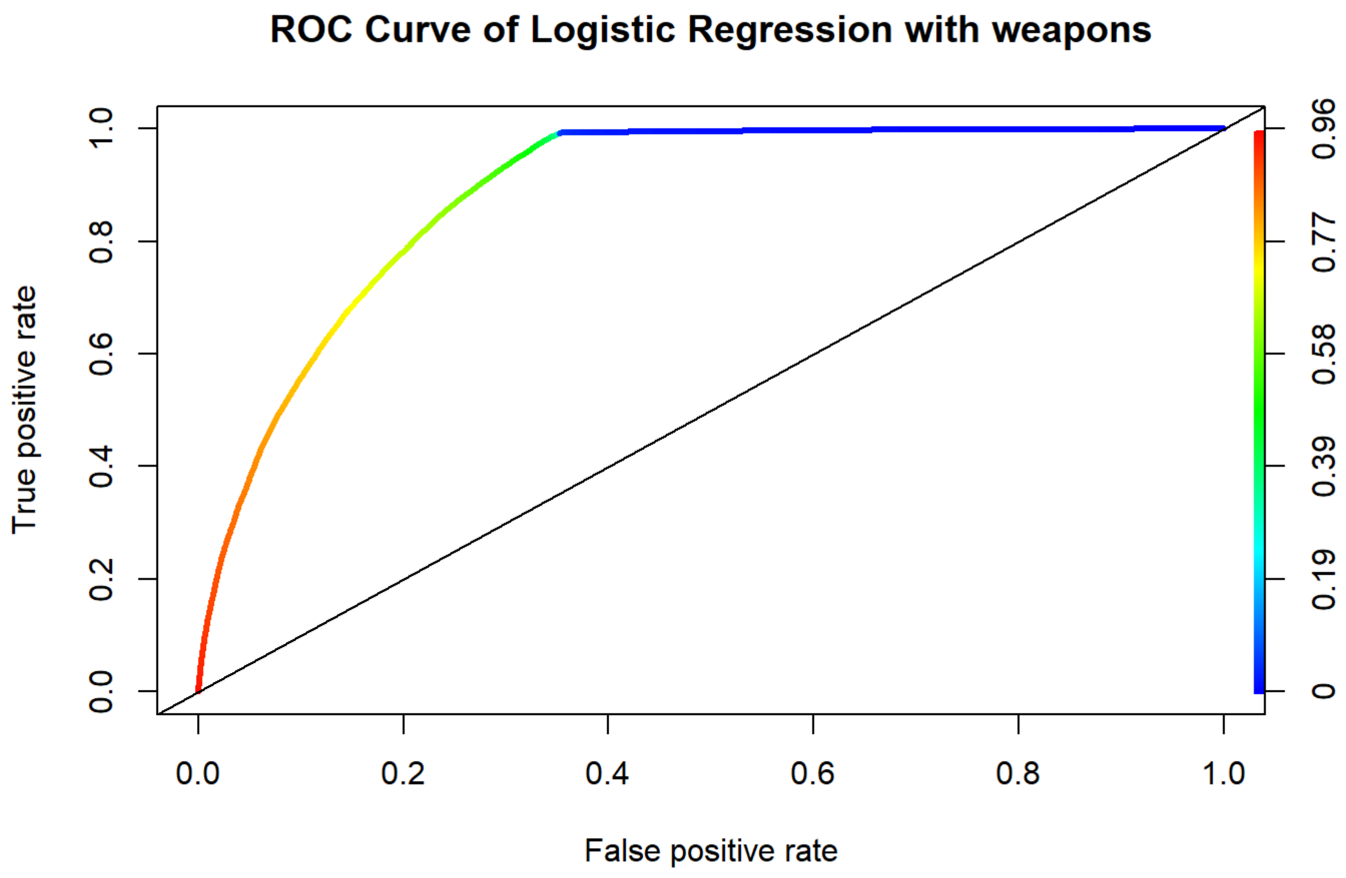
```
logistic.test2= predict(logistic.newcrime4, testdatafinal, type = 'response')
pred.crimeseverity2= rep('Non-Severe',193585)
pred.crimeseverity2[logistic.test2 > 0.5] = 'Severe'
table(pred.crimeseverity2, testdatafinal$Severity)
```

```
##
## pred.crimeseverity2 Non-Severe Severe
##          Non-Severe     108848   8203
##          Severe          58025  18509
```

```
cat("The misclassification rate for the testing data is",(8203+58025)/(108848+8203+58025+1850
9))
```

```
## The misclassification rate for the testing data is 0.3421133
```

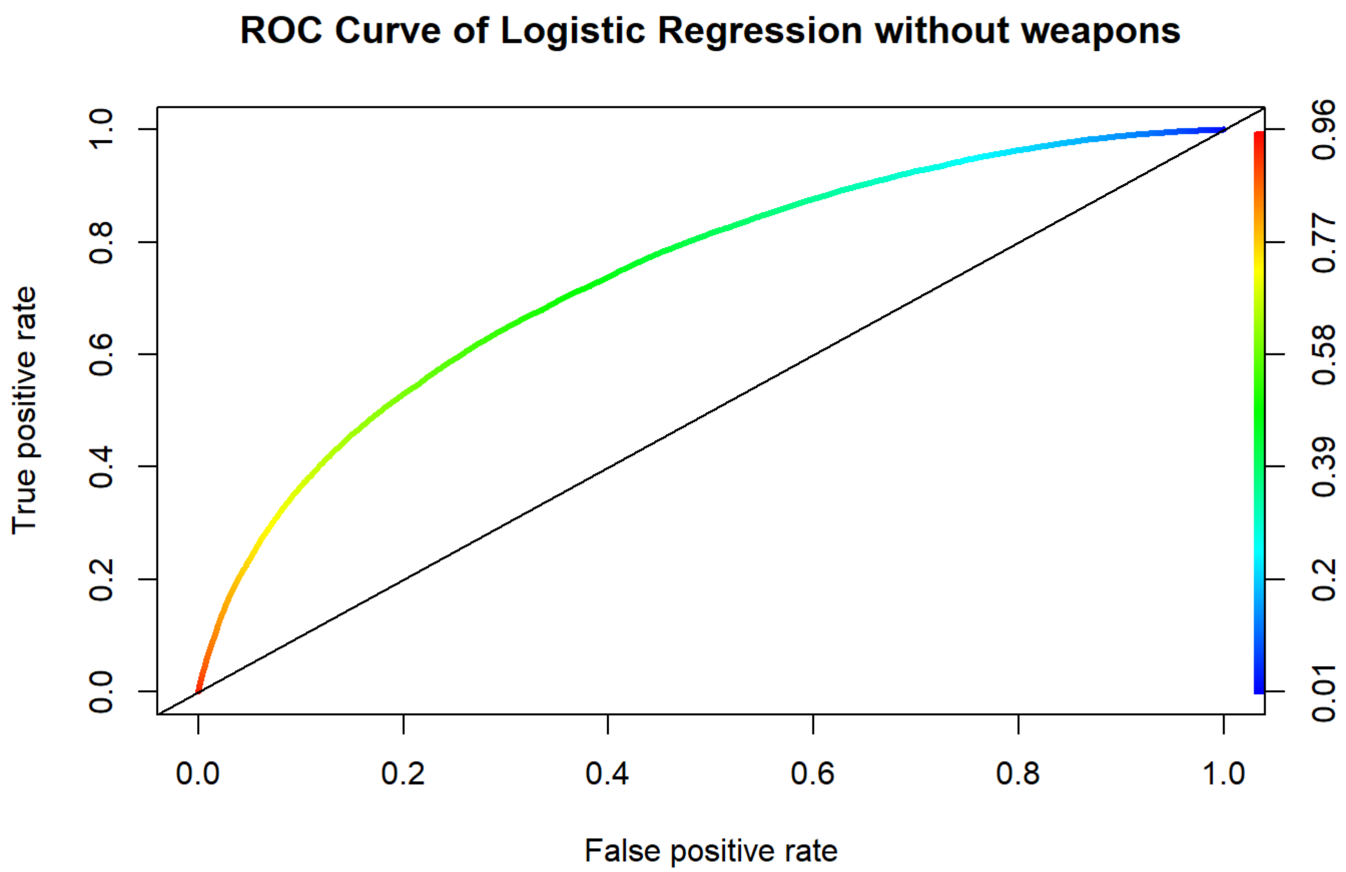
```
#ROC Curves for logistic regression
#With weapons
pred.glm1 = predict(logistic.crime7, testdatafinal, type="response")
prediction.glm1 = prediction(pred.glm1, testdatafinal$Severity)
rocGlm1 = performance(prediction.glm1, measure = "tpr", x.measure = "fpr")

plot(rocGlm1, lwd=3, colorkey=T, colorize=T, main="ROC Curve of Logistic Regression with weap
ons")
abline(0,1)
```




```
performance(prediction.glm1, measure = "auc")@y.values
```

```
## [[1]]
## [1] 0.8868616
```

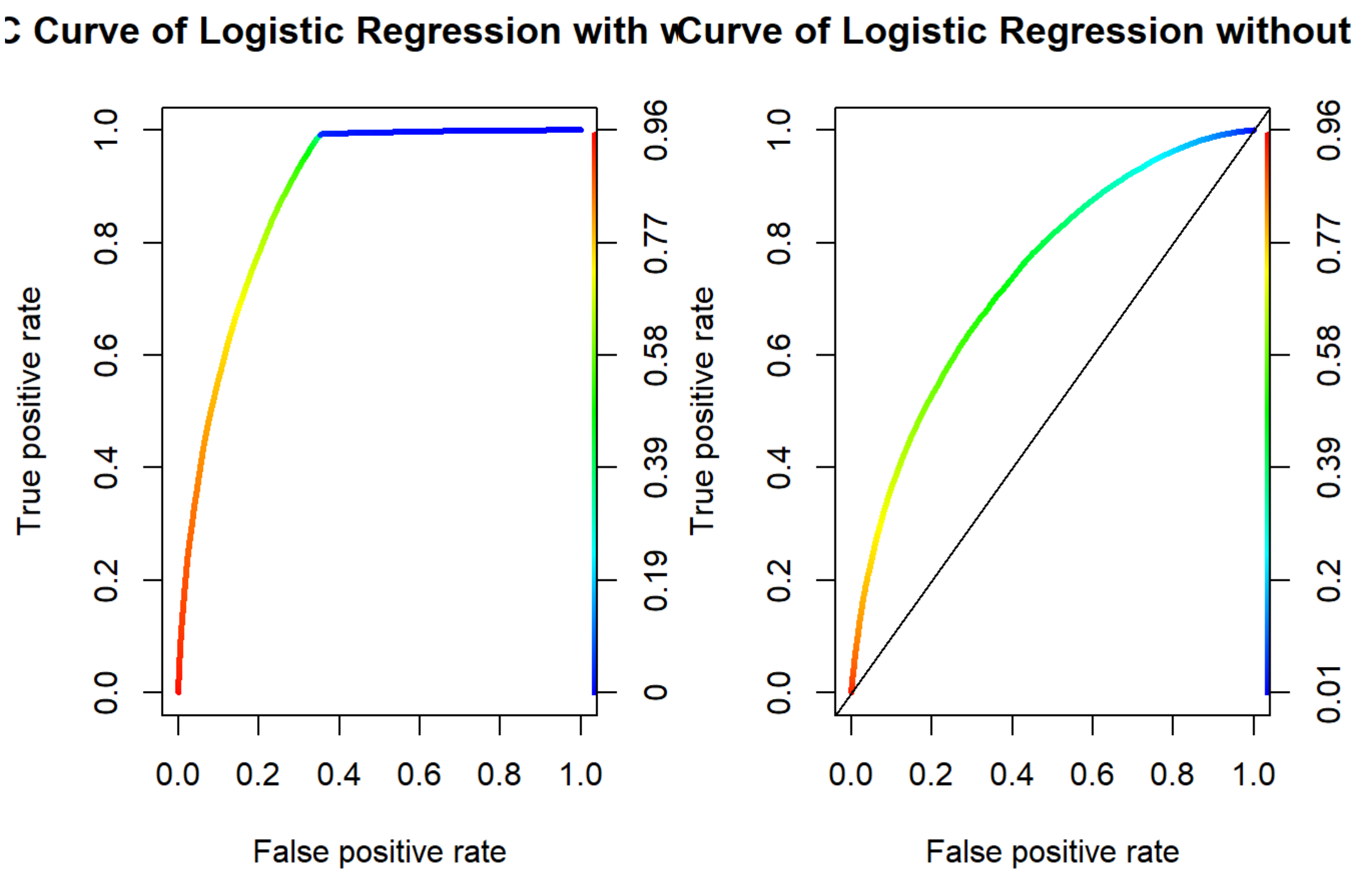
```
#Without weapons
pred.glm2 = predict(logistic.newcrime4, testdatafinal, type="response")
prediction.glm2 = prediction(pred.glm2, testdatafinal$Severity)
rocGlm2 = performance(prediction.glm2, measure = "tpr", x.measure = "fpr")

plot(rocGlm2, lwd=3, colorkey=T, colorize=T, main="ROC Curve of Logistic Regression without w
eapons")
abline(0,1)
```




```
performance(prediction.glm2, measure = "auc")@y.values
```

```
## [[1]]
## [1] 0.7410708
```

```
#Plotting both curves side by side
par(mfrow=c(1,2))
plot(rocGlm1, lwd=3, colorkey=T, colorize=T, main="ROC Curve of Logistic Regression with weap
ons")
plot(rocGlm2, lwd=3, colorkey=T, colorize=T, main="ROC Curve of Logistic Regression without w
eapons")
abline(0,1)
```



C Curve of Logistic Regression with w
Curve of Logistic Regression without
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0
0.19
0.39
0.58
0.77
0.96
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0.01
0.2
0.39
0.58
0.77
0.96