Project Document

Asif Irfanullah Masum & Ifrat Zaman

04/13/2023

## Install and load the necessary packages

#install.packages(caTools)  
#install.packages(dplyr)  
#install.packages(nnet)  
#install.packages(e1071)  
#install.packages(ggplot2)  
library(caTools)  
library(dplyr)  
library(nnet)  
library(e1071)  
library(ggplot2)

## Data Preprocessing and Wrangling

### Loading Data and Exploration

crash\_dataset <- read.csv("project\_dataset.csv", header = TRUE)  
   
# Display the dimensions of the Crash Dataset  
cat("Dimensions for Crash Dataset: ", paste(dim(crash\_dataset), collapse = " x "))

## Dimensions for Crash Dataset: 399794 x 30

# Display the structure of the Crash Dataset  
cat("\nStructure for Crash Dataset:\n")

##   
## Structure for Crash Dataset:

str(crash\_dataset)

## 'data.frame': 399794 obs. of 30 variables:  
## $ injy\_svty\_cd : int 5 5 NA 5 NA 5 5 4 NA 5 ...  
## $ crsh\_id : int 7497108 7497108 7497147 7497147 7497152 7497152 7497166 7497170 7497170 7497171 ...  
## $ unit\_num : int 2 1 2 1 2 1 1 1 2 1 ...  
## $ invl\_prty\_key : int 3418304 3418303 3418424 3418421 3418441 3418440 3418488 3418494 3418495 3418496 ...  
## $ prty\_type : chr "D" "D" "D" "D" ...  
## $ rdwy\_area\_cd : int 2 2 6 6 6 6 6 6 6 6 ...  
## $ objectid : int 4959 4959 1098 1098 189 189 276 2219 2219 598 ...  
## $ rte\_no : chr "M-47" "M-47" "I-94" "I-94" ...  
## $ pr : int 769506 769506 14903 14903 1576405 1576405 581003 580805 580805 580805 ...  
## $ mp : num 1.53 1.53 7.91 7.91 18.42 ...  
## $ milt\_time : chr "218" "218" "120" "120" ...  
## $ num\_unit : int 2 2 2 2 2 2 1 2 2 1 ...  
## $ crsh\_type\_cd : int 8 8 11 11 8 8 1 5 5 1 ...  
## $ wthr\_cd : int 5 5 5 5 1 1 2 5 5 5 ...  
## $ lit\_cd : int 5 5 4 4 1 1 1 1 1 1 ...  
## $ rd\_cond\_cd : int 4 4 4 4 1 1 3 3 3 4 ...  
## $ num\_lns : int 2 2 3 3 3 3 2 2 2 2 ...  
## $ spd\_limt : int 70 70 70 70 70 70 70 70 70 70 ...  
## $ mdot\_regn\_cd : int 4 4 5 5 7 7 5 5 5 5 ...  
## $ lane\_dprt\_cd : int 0 0 0 0 0 0 1 0 0 1 ...  
## $ vehc\_yr : int 1998 2007 NA 2003 1993 2007 1998 2007 NA 2006 ...  
## $ vehc\_dfct\_cd : int 6 6 6 6 6 6 6 6 6 6 ...  
## $ prty\_age : int 17 59 NA 22 NA 42 21 39 NA 21 ...  
## $ rstr\_not\_used\_fail: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ gndr\_cd : chr "F" "M" "" "F" ...  
## $ hzrd\_actn\_cd : int 1 0 0 8 0 16 0 0 0 1 ...  
## $ alch\_susp\_ind : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ drug\_susp\_ind : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ year : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...  
## $ traffic\_volume : int 6699 6699 43750 43750 56423 56423 19321 18050 18050 18058 ...

### Dividing Dataset into Natural Factor and Human Factor Subset

natural\_factor\_crash <- subset(crash\_dataset, select = c(injy\_svty\_cd, wthr\_cd, lit\_cd, rd\_cond\_cd, mdot\_regn\_cd, invl\_prty\_key))  
human\_factor\_crash <- subset(crash\_dataset, select = c(injy\_svty\_cd, vehc\_dfct\_cd, rstr\_not\_used\_fail, hzrd\_actn\_cd, alch\_susp\_ind, drug\_susp\_ind,invl\_prty\_key  
))  
  
# Display the dimensions of the Natural Factor Subset  
cat("Dimensions for Natural Factor Subset : ", paste(dim(natural\_factor\_crash), collapse = " x "), "\n")

## Dimensions for Natural Factor Subset : 399794 x 6

# Display the structure of the Natural Factor Subset  
cat("Structure for Natural Factor Subset :\n")

## Structure for Natural Factor Subset :

str(natural\_factor\_crash)

## 'data.frame': 399794 obs. of 6 variables:  
## $ injy\_svty\_cd : int 5 5 NA 5 NA 5 5 4 NA 5 ...  
## $ wthr\_cd : int 5 5 5 5 1 1 2 5 5 5 ...  
## $ lit\_cd : int 5 5 4 4 1 1 1 1 1 1 ...  
## $ rd\_cond\_cd : int 4 4 4 4 1 1 3 3 3 4 ...  
## $ mdot\_regn\_cd : int 4 4 5 5 7 7 5 5 5 5 ...  
## $ invl\_prty\_key: int 3418304 3418303 3418424 3418421 3418441 3418440 3418488 3418494 3418495 3418496 ...

# Display the dimensions of the Human Factor Subset  
cat("Dimensions for Human Factor Subset : ", paste(dim(human\_factor\_crash), collapse = " x "), "\n")

## Dimensions for Human Factor Subset : 399794 x 7

# Display the structure of the Human Factor Subset  
cat("Structure for Human Factor Subset :\n")

## Structure for Human Factor Subset :

str(human\_factor\_crash)

## 'data.frame': 399794 obs. of 7 variables:  
## $ injy\_svty\_cd : int 5 5 NA 5 NA 5 5 4 NA 5 ...  
## $ vehc\_dfct\_cd : int 6 6 6 6 6 6 6 6 6 6 ...  
## $ rstr\_not\_used\_fail: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ hzrd\_actn\_cd : int 1 0 0 8 0 16 0 0 0 1 ...  
## $ alch\_susp\_ind : int 0 0 0 1 0 0 0 0 0 0 ...  
## $ drug\_susp\_ind : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ invl\_prty\_key : int 3418304 3418303 3418424 3418421 3418441 3418440 3418488 3418494 3418495 3418496 ...

### Data Cleaning by removing duplicate and NULL values for Natural Factor Subset

cat("Total number of NULL values in Natural Factor Subset : ", paste(sum(is.na(natural\_factor\_crash))), "\n")

## Total number of NULL values in Natural Factor Subset : 21655

cat("Total number of duplicate values Natural Factor Subset : ", paste(sum(duplicated(natural\_factor\_crash))), "\n")

## Total number of duplicate values Natural Factor Subset : 4147

natural\_factor\_crash <- na.omit(natural\_factor\_crash)  
natural\_factor\_crash <- unique(natural\_factor\_crash)  
  
cat("Dimensions for Natural Factor Subset after data cleaning: ", paste(dim(natural\_factor\_crash), collapse = " x "), "\n")

## Dimensions for Natural Factor Subset after data cleaning: 374099 x 6

### Data Cleaning by removing duplicate and NULL values for Human Factor Subset

cat("Total number of NULL values in Human Factor Subset: ", sum(is.na(human\_factor\_crash)), "\n")

## Total number of NULL values in Human Factor Subset: 31125

cat("Total number of duplicate values in Human Factor Subset: ", sum(duplicated(human\_factor\_crash)), "\n")

## Total number of duplicate values in Human Factor Subset: 4147

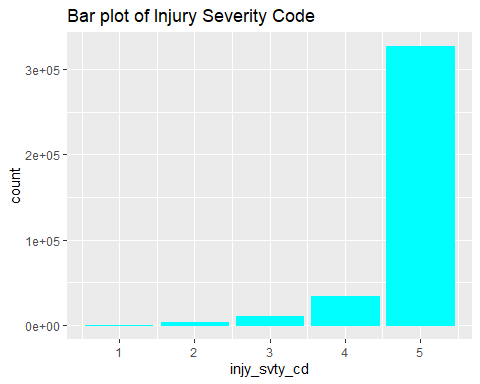
human\_factor\_crash <- na.omit(human\_factor\_crash)  
human\_factor\_crash <- unique(human\_factor\_crash)  
  
cat("Dimensions for Human Factor Subset after data cleaning: ", paste(dim(human\_factor\_crash), collapse = " x "), "\n")

## Dimensions for Human Factor Subset after data cleaning: 371684 x 7

## Data Visualization for Natural Factors Subset

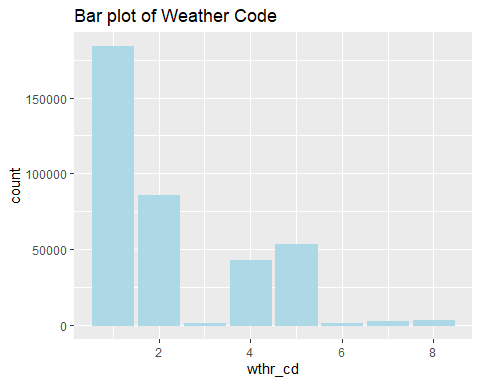
### Bar plot of Injury Severity Code

ggplot(natural\_factor\_crash, aes(x = injy\_svty\_cd)) +  
 geom\_bar(color = "cyan", fill = "cyan") +  
 labs(title = "Bar plot of Injury Severity Code")



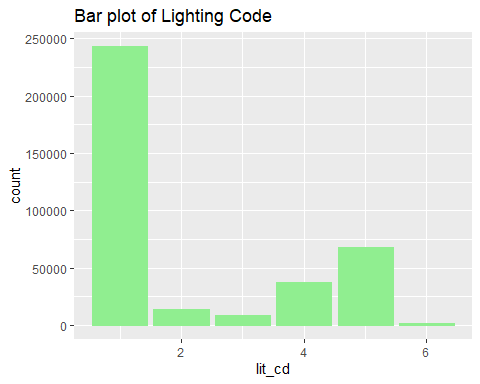
### Bar plot of Weather Code

ggplot(natural\_factor\_crash, aes(x = wthr\_cd)) +  
 geom\_bar(color = "lightblue", fill = "lightblue") +  
 labs(title = "Bar plot of Weather Code")



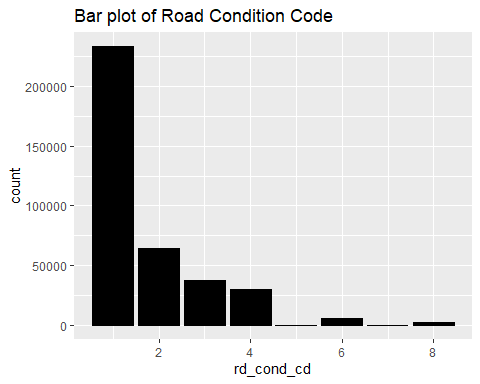
### Bar plot of Lighting Code

ggplot(natural\_factor\_crash, aes(x = lit\_cd)) +  
 geom\_bar(color = "lightgreen", fill = "lightgreen") +  
 labs(title = "Bar plot of Lighting Code")



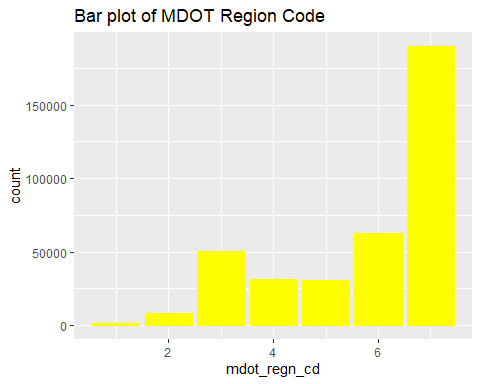
### Bar plot of Road Condition Code

ggplot(natural\_factor\_crash, aes(x = rd\_cond\_cd)) +  
 geom\_bar(color = "black", fill = "black") +  
 labs(title = "Bar plot of Road Condition Code")



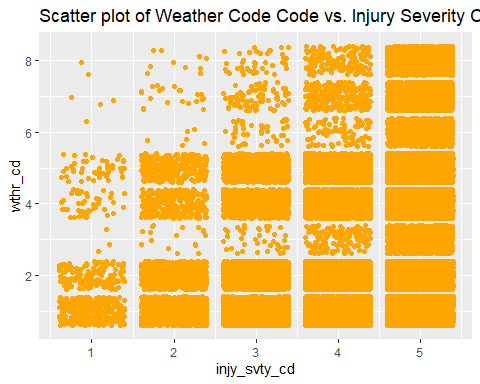
### Bar plot of MDOT Region Code

ggplot(natural\_factor\_crash, aes(x = mdot\_regn\_cd)) +  
 geom\_bar(color = "yellow", fill = "yellow") +  
 labs(title = "Bar plot of MDOT Region Code")



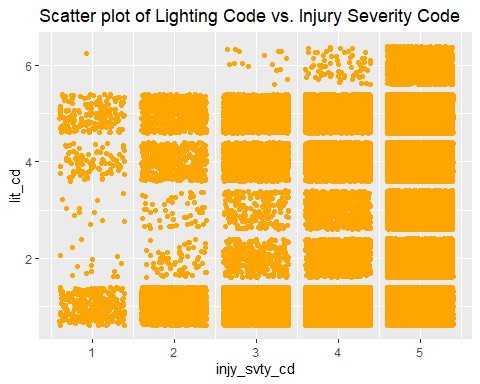
### Scatter plot of Weather Code vs. Injury Severity Code

ggplot(data = natural\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd , y = wthr\_cd), color = "orange", position="jitter") +  
 labs(title = "Scatter plot of Weather Code Code vs. Injury Severity Code")



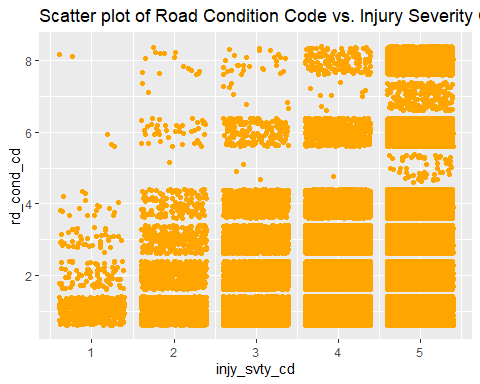
### Scatter plot of Lighting Code vs. Injury Severity Code

ggplot(data = natural\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd, y = lit\_cd), color = "orange", position="jitter") +  
 labs(title = "Scatter plot of Lighting Code vs. Injury Severity Code")



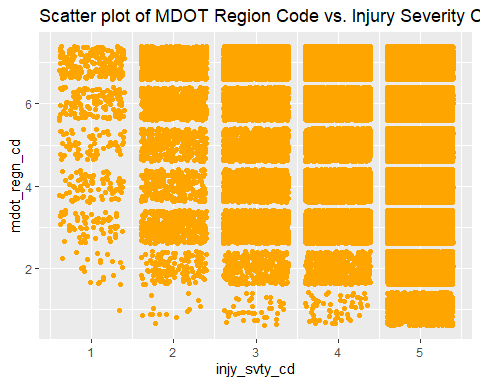
### Scatter plot of Road Condition Code vs. Injury Severity Code

ggplot(data = natural\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd, y = rd\_cond\_cd), color = "orange", position="jitter") +  
 labs(title = "Scatter plot of Road Condition Code vs. Injury Severity Code")



### Scatter plot of MDOT Region Code vs. Injury Severity Code

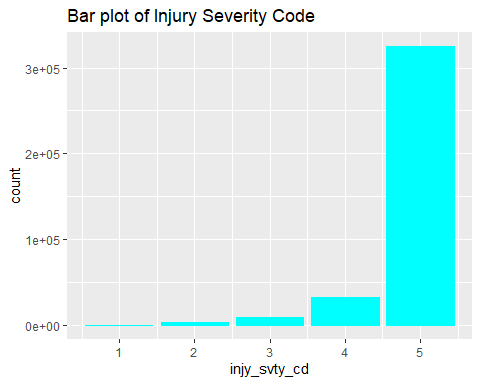
ggplot(data = natural\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd, y = mdot\_regn\_cd), color = "orange", position="jitter") +  
 labs(title = "Scatter plot of MDOT Region Code vs. Injury Severity Code")



## Data Visualization for Human Factors Subset

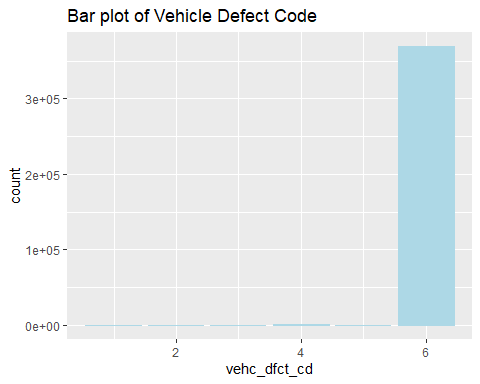
### Bar plot of Injury Severity Code

ggplot(human\_factor\_crash, aes(x = injy\_svty\_cd)) +  
 geom\_bar(color = "cyan", fill = "cyan") +  
 labs(title = "Bar plot of Injury Severity Code")



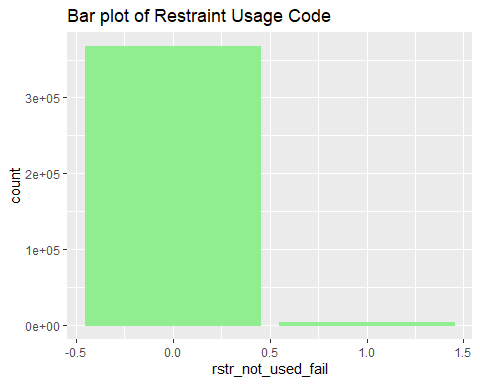
### Bar plot of Vehicle Defect Code

ggplot(human\_factor\_crash, aes(x = vehc\_dfct\_cd)) +  
 geom\_bar(color = "lightblue", fill = "lightblue") +  
 labs(title = "Bar plot of Vehicle Defect Code")



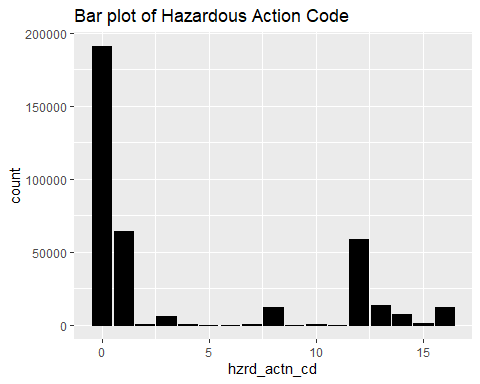
### Bar plot of Restraint Usage Code

ggplot(human\_factor\_crash, aes(x = rstr\_not\_used\_fail)) +  
 geom\_bar(color = "lightgreen", fill = "lightgreen") +  
 labs(title = "Bar plot of Restraint Usage Code")



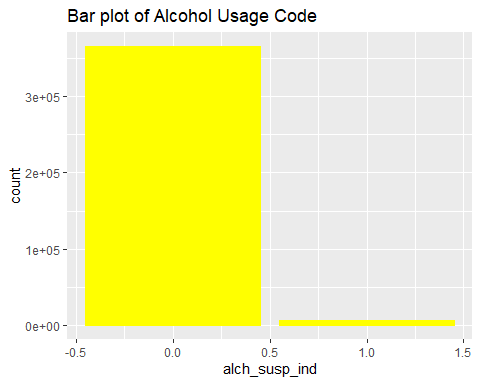
### Bar plot of Hazardous Action Code

ggplot(human\_factor\_crash, aes(x = hzrd\_actn\_cd)) +  
 geom\_bar(color = "black", fill = "black") +  
 labs(title = "Bar plot of Hazardous Action Code")



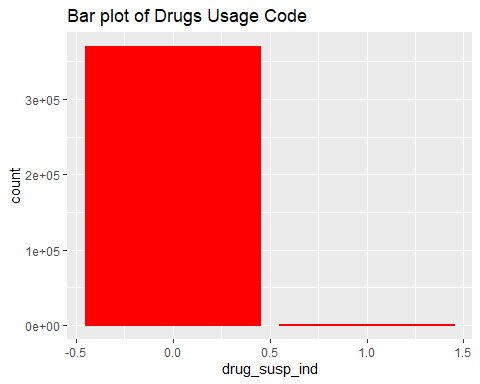
### Bar plot of Alcohol Usage Code

ggplot(human\_factor\_crash, aes(x = alch\_susp\_ind)) +  
 geom\_bar(color = "yellow", fill = "yellow") +  
 labs(title = "Bar plot of Alcohol Usage Code")



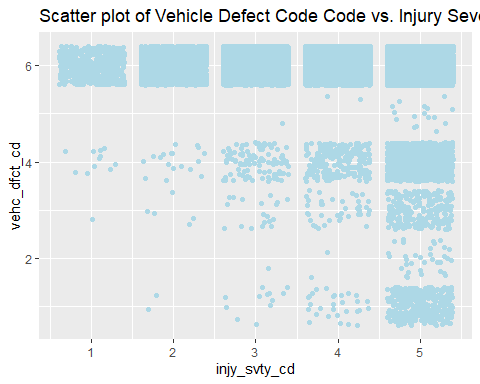
### Bar plot of Drugs Usage Code

ggplot(human\_factor\_crash, aes(x = drug\_susp\_ind)) +  
 geom\_bar(color = "red", fill = "red") +  
 labs(title = "Bar plot of Drugs Usage Code")



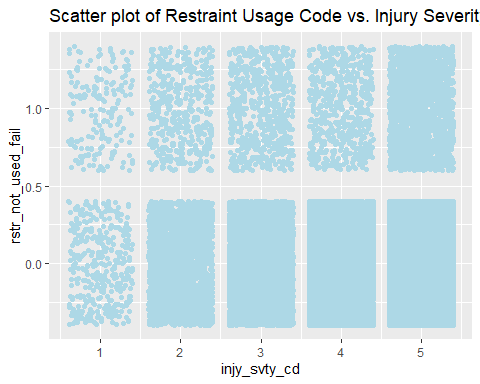
### Scatter plot of Vehicle Defect Code vs. Injury Severity Code

ggplot(data = human\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd , y = vehc\_dfct\_cd), color = "lightblue", position="jitter") +  
 labs(title = "Scatter plot of Vehicle Defect Code Code vs. Injury Severity Code")



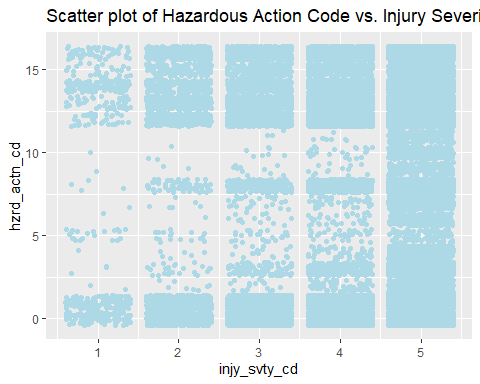
### Scatter plot of Restraint Usage Code vs. Injury Severity Code

ggplot(data = human\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd, y = rstr\_not\_used\_fail), color = "lightblue", position="jitter") +  
 labs(title = "Scatter plot of Restraint Usage Code vs. Injury Severity Code")



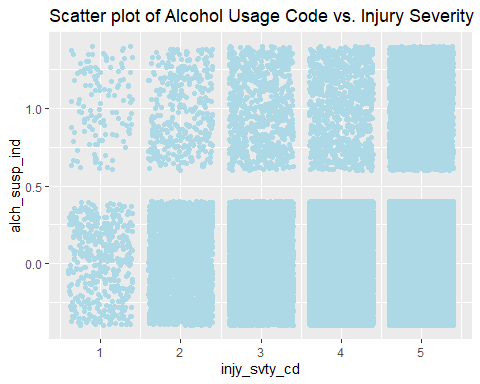
### Scatter plot of Hazardous Action Code vs. Injury Severity Code

ggplot(data = human\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd, y = hzrd\_actn\_cd), color = "lightblue", position="jitter") +  
 labs(title = "Scatter plot of Hazardous Action Code vs. Injury Severity Code")



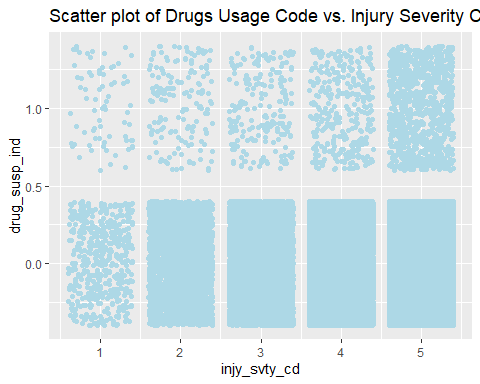
### Scatter plot of Alcohol Usage Code vs. Injury Severity Code

ggplot(data = human\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd, y = alch\_susp\_ind), color = "lightblue", position="jitter") +  
 labs(title = "Scatter plot of Alcohol Usage Code vs. Injury Severity Code")



### Scatter plot of Drugs Usage Code vs. Injury Severity Code

ggplot(data = human\_factor\_crash) +  
 geom\_point(mapping = aes(x = injy\_svty\_cd, y = drug\_susp\_ind), color = "lightblue", position="jitter") +  
 labs(title = "Scatter plot of Drugs Usage Code vs. Injury Severity Code")



## Spliting Data for Training set and Testing set for Natural Factor Subset

set.seed(123)  
split <- sample.split(natural\_factor\_crash$injy\_svty\_cd, SplitRatio = 0.8)  
nfcTrainData <- subset(natural\_factor\_crash, split == TRUE)  
nfcTestData <- subset(natural\_factor\_crash, split == FALSE)

## Model Training on Natural Factor Subset

# Train a multi-class logistic regression model  
logRegModel1 <- multinom(injy\_svty\_cd ~ ., data = nfcTrainData)

## # weights: 35 (24 variable)  
## initial value 481670.968993   
## iter 10 value 195259.691310  
## iter 20 value 187470.499893  
## iter 30 value 177805.371975  
## iter 40 value 158054.669406  
## iter 50 value 148147.898657  
## iter 60 value 145553.604340  
## iter 70 value 144021.686071  
## iter 80 value 143578.067465  
## iter 90 value 143184.319221  
## iter 100 value 142982.508311  
## final value 142982.508311   
## stopped after 100 iterations

# Make predictions on the test data  
predictions1 <- predict(logRegModel1, nfcTestData)  
  
# Check the metrics of the model  
cat("Model Accuracy for Human Factor Subset", mean(predictions1 == nfcTestData$injy\_svty\_cd), "\n")

## Model Accuracy for Human Factor Subset 0.874325

## Spliting Data for Training set and Testing set for Human Factor Subset

set.seed(123)  
split <- sample.split(human\_factor\_crash$injy\_svty\_cd, SplitRatio = 0.8)  
hfcTrainData <- subset(human\_factor\_crash, split == TRUE)  
hfcTestData <- subset(human\_factor\_crash, split == FALSE)

## Model Training on Human Factor Subset

# Train a multi-class logistic regression model  
logRegModel2 <- multinom(injy\_svty\_cd ~ ., data = hfcTrainData)

## # weights: 40 (28 variable)  
## initial value 478561.534946   
## iter 10 value 181692.448513  
## iter 20 value 157729.037185  
## iter 30 value 142862.211717  
## iter 40 value 140793.214581  
## iter 50 value 139894.363715  
## iter 60 value 139317.099253  
## iter 70 value 138957.681012  
## iter 70 value 138957.681012  
## iter 80 value 138902.425241  
## iter 90 value 138810.832523  
## iter 100 value 138770.287396  
## final value 138770.287396   
## stopped after 100 iterations

# Make predictions on the test data  
predictions2 <- predict(logRegModel2, hfcTestData)  
  
# Check the accuracy of the model  
cat("Model Accuracy for Human Factor Subset", mean(predictions2 == hfcTestData$injy\_svty\_cd), "\n")

## Model Accuracy for Human Factor Subset 0.8745039