A Case Study Of Analyzing 2013 Chicago Youth Health Risk Behavior Data By Machine Learning With CRISP-DM Methodology

Submitted as final project fulfillment of Course: ISTE790 Data Analytics for Emerging Technologies

Group: 5

(Saeed Albarhami, Abdullah Hussein)

25 October 2019

Abstract

The objective of this paper is to use data mining techniques to analyze the 2013 Chicago youth risk behavior surveillance survey data,Collected by the centers for Disease Control and Prevention in USA, to investigate interesting relations and patterns in collected data, and provide a recommendation based on findings that been discovered by applying machine learning algorithms such apriori decision tree. . . etc. following CRISP-DM steps methodology, staring by business understanding, data understanding, data preparation, modeling, and evaluation. R open-source software will be used among all the process steps.

Contents

R·I·T Dubai
جامعة روتشستر للتكنولوجيا – نيويورك، أمريكا
Rochester Institute of Technology, NY, USA

Figure 1: test

1 Executive Summary.

- The main objective of this project is to examine the associations between victimization, substance use, and suicide attempt among youth using the Youth Health Risk Behavior Survey data for the year 2013.
- The data used consists of information from 1581 respondents, consist of Record Id & 28 variables that are categorized into 4 categories:
  1. Demographic
  2. Victimization
  3. Suicide Attempt
  4. Substance Use
- Each respondent is identified by a record ID and are mostly between the ages of 15-19 studying in grades 9-12.
- Most of the data consists of "Yes" / "No" answers to questions from the survey.
  - Among the victimization variables, most common "Yes" answer was to the question 'Fought school 1+ times 12 months – around 15% of records
  - Among the substance use variables, most common "Yes" answer was to the question 'Tried marijuana 1+ times in life' – around 50% of records
  - Among the suicide attempts variables, most common "Yes" answer was to the question 'Considered suicide 12 months – around 16% of records
  - There are many NA/missing values in the dataset and after omitting these, we got 934 records that could be used for analysis.
  - In the data preparation stage, the following transformations and modifications are done.
  - Combining variables from each category into one variable each for victimization, substance use and suicide attempt
  - Generating data in transaction format to do association analysis
- In the first part of modeling, we used apriori algorithm to identify most common associations between victimization, suicide attempt and substance use. The major findings are:
  - There is association between suicide attempt and substance use at support of 17% and confidence of around 70%, but the association between suicide attempt and victimization is weaker at support of around 12% and confidence of around 49%.
  - The strongest association between individual variables is between qn27 (Considered suicide in last 12 months) and qn47 (Tried marijuana 1+ times in life) at support of around 11% and confidence of around 62%.

- In the second part of modeling, we used decision tree machine learning techniques to automatically segment the class gradeand determine how well these derived groupings correspond to victimization and suicide attempt and predict 'suicide attempt' using the aggregated variables variables 'victimization' and 'substance use' for class 'grade'.

2 Introduction.

2.1 Background.

Health behaviors and experiences related to sexual behavior, high-risk substance use, violence victimization, mental health and suicide contribute to substantial morbidity for adolescents, including risk for HIV, STDs, and teen pregnancy.The Centers for Disease Control and Prevention in the United States monitors rotinely youth health behaviours and experiances by conducting a yearly servuey across the country in clloaboration with schools to help in prevent futture prevention of spread of HIV, drug uses, sexually transmitted diseases, and unintend teen pregnancy in goal to raise awareness and understanding. Collected under the flag of YRBSS system which is developed in 1990 to monitor those risks.From 1991 untill 2013, A total 2.6 million high school student data was collected in more than 1,100 separate surveys. In this paper the anlysis will be performed on 2013 YRBSS Chicago dataset, downloaded from the Centers for Disease Control and Prevention, CRISP-DM methodolgy steps will be followed. R open-source software for report generating, analysis, and to communcate findings.

2.1.1 Overview of CRISP-DM:

CRISP-DM was conceived in 1996 and immature data mining market as a standrized process for data mining projects, The methodolgy provides an overview of the life cycle of a data mining project, In six major steps; Bussniess understanding, Data understanding, Data preparation, Modeling, And Evaluation. According to latest poll (2014) results by https://www.kdnuggets.com/, crisp-dm methodology is still the most popular data mining with 43%, placed by SEMMA by SAS institute, and followed by KDD process.

Note:

- As per the course requirment deployment step is not included in process.

2.2 Challenges.

This case study examines the associations between victimization, substance use, and suicide attempt among youth in Chicago at 2013, challenges are:

- Are there relations between victimization (fighting, bullying, sexual abuse) and substanceuse (Tabaco, alcohol and other drug use)?
- Are there relations between victimization (fighting, bullying, sexual abuse) and suicide attempts?
- Are there relations between substance use (Tabaco, alcohol and other substance use) and suicide attempt?

3 Business Understanding:

3.1 Objective:

The main objective of this project is to examine the associations between victimization, substance use, and suicide attempt among youth using the Youth Health Risk Behavior Survey data for the year 2013. We will be using Apriori (an algorithm for frequent item set mining and association rule learning) and Decision Tree.

Using the above mentioned algorithms we are going to answer the following questions:

1. Are there relations between victimization (fighting, bullying, sexual abuse) and suicide attempt?

2. Are there relations between substance use (Tabaco, alcohol and other substance use) and suicide attempt?

3. Apply machine learning techniques to automatically segment the class grade into clusters and determine how well these derived groupings correspond to victimization and suicide attempt.

3.2 Motivation:

Youth suicide is a substantial concern for health professionals, educators, lawmakers and society in general. Researchers have estimated that around 11% of all deaths among 12-19 year olds is due to suicide. It is assumed that there is high association between victimization, substance use and suicide attempt. Studying these associations will help in understanding youth behaviors and reducing adverse events. This type of analysis will also help doctors make decisions after taking into account the risk of suicide among their youth patients with history of victimization and/or substance use.
It is also critical to identify high risk groups who may be more associated with suicide attempt so that targeted preventive measures can be taken. For example, the CDC states that historical suicide rates for teens aged 15-19 years in the US differ significantly between genders.

3.3 Data Description:

The Youth Health Risk Behavior Survey is a biannual study undertaken by UNITED STATE CDC that monitors several categories of health-related behaviors among youth. The survey includes adolescents from grades 9-12 in the age group of 14-19 years. In our analysis, we consider behaviors related to victimization (fighting, bullying, sexual abuse, etc.), substance use (tobacco, alcohol, marijuana, etc.) and suicide attempt (considered suicide, attempted suicide, etc.). The responses of the survey questions are initially processed by the CDC to identify logical inconsistencies, convert responses to usable form, create derived variables from responses, etc. We use a subset of the full data and analyze only demographic, victimization, substance use and suicide attempt information.

3.3.1 About the dataset:

This case study is from the Youth Risk Behavior Survey (YRBS) data which are free for use. (Seen from http://www.cdc.gov/healthyyouth/data/yrbs/data.htm).

4 Data Understanding.

The dataset used in this project consists of Record ID that serves as a unique identifier, 4 demographic variables, 7 victimization variables, 4 suicide attempt variables and 13 substance use variables. The data is provided in csv format.

4.1 Metadata Description.

Here shows a full description of all dataset variables, with detailsfor all variables.

```r
variableDescription <- read.csv('DataDescription.csv', header=TRUE)
pander(head(variableDescription, n=nrow(variableDescription)))
```

Variable	Description	Short.Description
record	Record ID of participant	Record
age	Age of participant	Age
sex	Sex of participant	Sex
grade	Grade in which participant was studying	Grade
race4	Race/ethnicity of participant	Race
qn16	Unsafe at school 1 or more times in the past 30 days	Unsafe
qn17	Threatened at school 1 or more times in the past 12 months	Threatened
qn19	Injured at school 1 or more times in the past 12 months	Injured
qn20	Fought at school 1 or more times in the past 12 months	Fought
qn21	Forced to have sex	ForcedSex
qn24	Bullied 1 or more times in the past 12 months	Builled
qn25	Electronically bullied 1 or more times in the past 12 months	Ebuilled
qn27	Considered suicide in the past 12 months	ConsideredSuicide
qn28	Made suicide plan in the past 12 months	MadeSuicidePlan
qn29	Attempted suicide in the past 12 months	AttemptSuicide
qn30	Suicide attempt with or without injury in the past 12 months	SuicideWithOrWithoutInjury
qn33	Smoked 1 or more times in the past 30 days	SmokedMonth
qn37	Smoked daily for 30 days	SmokedDaily
qn43	Had drinks 1 or more times in the past 30 days	Drinks1+
qn45	Had drinks 10 or more times in the past 30 days	Drinks10+
qn47	Tried marijuana 1 or more times in life	Marijuana1+
qn50	Used cocaine 1 or more times in life	Cocaine1+
qn51	Sniffed glue 1 or more times in life	SniffedGlue1+
qn52	Used heroin 1 or more times in life	Heroin1+
qn53	Used meth 1 or more times in life	Meth1+
qn54	Used ecstasy 1 or more times in life	Ecstasy1+
qn55	Took steroids 1 or more times in life	Steroids1+
qn56	Taken prescription drug without prescription 1 or more times in life	PrescriptionDrug
qn57	Injected drugs 1 or more times in life	InjectedDrugs

4.2 Loading, Retrieving, Viewing & Inspecting Data.

4.2.1 Loading, Retrieving, Viewing.

Loading the data from the main source and veiw the 5 occurances from each variables.

```
data <- read.csv("CaseStudy10_YouthHealthRiskBehavior_Data.csv",header = TRUE)
# Overview on data structure (Number of observations and cols, type of variables and their values.)
# str(data)
panderOptions('table.split.table', 100)
pander(head(data))
```

Table 2: Table continues below

record	age	sex	grade	race4	qn16	qn17	qn19	qn20	qn21	qn24	qn25	qn27	qn28	qn29	qn30	qn33	qn37	qn43	qn45	qn47	qn50	qn51	qn52	qn53	qn54	qn55	qn56	qn57
1115896	NA	NA	NA	2	1	1	2	1	NA	NA	2	1	NA	NA	NA	NA	NA	NA	NA	1	NA	1	2	2	2	1	2	2
1115897	NA	NA	4	3	1	2	2	2	2	1	2	1	1	2	2	2	2	1	2	2	2	2	2	2	2	2	1	2
1115898	1	NA	NA	4	1	2	1	1	2	2	2	1	1	NA	NA	NA	1	NA	NA	1	1	1	1	2	1	1	1	2
1115899	2	NA	2	3	NA	1	1	NA	2	NA	2	2	NA	1	2	NA	2	1	2	NA	1	1	1	1	1	1	1	1
1115900	3	NA	3	NA	1	1	1	1	NA	2	2	2	2	2	2	NA	2	2	2	1	2	2	2	2	2	2	2	2
1115901	4	NA	1	3	NA	NA	NA	NA	1	2	2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

```
# #about fit
# panderOptions('table.split.table', 80)
# pander(summary(data))
```

```
# pander(introduce(data))
# DataExplorer::plot_density(data[,c("sex","grade","age","race4")], ncol = 2L)
# skim(data)
```

```
#Data Overview With Selected Cols
displayTable <-head(data[2:15], n=2)

knitr::kable(displayTable)
```

age	sex	grade	race4	qn16	qn17	qn19	qn20	qn21	qn24	qn25	qn27	qn28	qn29
NA	NA	NA	2	1	1	2	1	NA	NA	2	1	NA	NA
NA	NA	4	3	1	2	2	2	2	1	2	1	1	2

It is clear that, there are many NA values in the data. This will be addressed in the next section.

4.3 Demographic Variables Description

Description of the factor levels of demographic variables

- Age: (1= <=12 years old, 2=13 years old, 3=14 years old, 4=15 years old, 5=16 years old, 6=17 years old, 7=18+ years old).
- Sex: (1=Female, 2=Male).
- Race: (1=White, 2=Black/African American, 3=Hispanic/Latino, 4=All other races).
- Grade (1=9th, 2=10th, 3=11th, 4=12th).

Data from the questions are all dichotomous (ordinal) as numerical values with levels “1”, “2”…etc or NA for missing value.

Note:

- Considering “1” corresponds to “Yes” and “2” corresponds to a “No”.

4.4 Data Distribution Overview

```r
#This function takes the dataset and the cols to omit the na values
omitNaForSpecificCols <- function(d, desiredCols) {
    completeVec <- complete.cases(d[, desiredCols])
    return(d[completeVec, ])
}
```

```r
#Takes cols related to Demographic variables
#This will take the range of cols from 2 to 5 i.e(2(age),3(sex),4(grade),5(race))
dataOverview<-data[, 2:5]
dataOverview$sex <- factor(dataOverview$sex, labels = c('Female','Male'), ordered = TRUE)
dataOverview$age <- factor(dataOverview$age, labels = c('<=12','13','14','15','16','17',
                                                        '18+'),
                           ordered = TRUE)
dataOverview$race <- factor(dataOverview$race4, labels =
                                c('White','Black/African American','Hispanic/Latino',
                                  'All other races'), ordered = TRUE)
dataOverview$grade <- factor(dataOverview$grade, labels = c('9th','10th','11th','12th'),
                             ordered = TRUE)

#Cleaning the data using omitNaForSpecificCols() created above.
dataOverview<- omitNaForSpecificCols(dataOverview,c("sex","age","race","grade"))

#Structure For Manipulated Data
str(dataOverview)
```

```
## 'data.frame':    1514 obs. of  5 variables:
##  $ age  : Ord.factor w/ 7 levels "<=12"<"13"<"14"<..: 1 1 1 3 3 3 3 3 3 3 ...
##  $ sex  : Ord.factor w/ 2 levels "Female"<"Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ grade: Ord.factor w/ 4 levels "9th"<"10th"<"11th"<..: 2 4 3 1 1 1 1 1 1 1 ...
##  $ race4: int  3 3 4 2 3 2 3 3 2 3 ...
##  $ race : Ord.factor w/ 4 levels "White"<"Black/African American"<..: 3 3 4 2 3 2 3 3 2 3 ...
```

```
#Data Summary
summary(dataOverview)
```

```
##    age          sex       grade         race4
##  <=12:  4   Female:866   9th :259   Min.   :1.000
##  13  :  0   Male  :648   10th:374   1st Qu.:2.000
##  14  :106                11th:434   Median :3.000
##  15  :279                12th:447   Mean   :2.548
##  16  :397                           3rd Qu.:3.000
##  17  :427                           Max.   :4.000
##  18+ :301
##                       race
##  White                   :132
##  Black/African American:549
##  Hispanic/Latino         :705
##  All other races         :128
##
##
##
```

```
#Overview On Our Current Dataframe With Demographic Variables Only(As Is)

displayTable<-head(dataOverview[1:4], n=5)
knitr::kable(displayTable)
```

	age	sex	grade	race4
15	<=12	Male	10th	3
17	<=12	Male	12th	3
18	<=12	Male	11th	4
19	14	Male	9th	2
20	14	Male	9th	3

```
#We notice many missing values; therefore, we are going to use the predefined cutom
#function "omitNaForSpecificCols"created above to omit the values whenever need for a
#specific/all cols needed for analysis
```

```
#Grouping The Respondents By Sex
cleanedData<-omitNaForSpecificCols(dataOverview, c("sex"))
cleanedData<- cleanedData %>% group_by(sex) %>% summarize(count=n()) %>%
    arrange(desc(sex),.by_group = TRUE)
knitr::kable(cleanedData)
```

sex	count
Male	648
Female	866

```
#Grouping The Respondents By Race
cleanedData<-omitNaForSpecificCols(dataOverview, c("race"))
cleanedData<- cleanedData %>% group_by(race) %>% summarize(count=n()) %>%
    arrange(desc(race),.by_group = TRUE)

knitr::kable(cleanedData)
```

race	count
All other races	128
Hispanic/Latino	705
Black/African American	549
White	132

```r
#Grouping The Respondents By Sex & Race
cleanedData<-omitNaForSpecificCols(dataOverview, c("sex","race"))
cleanedData<-cleanedData %>% group_by(race, sex) %>% summarize(count=n()) %>%
    arrange(desc(race),.by_group = TRUE)

knitr::kable(cleanedData)
```

race	sex	count
White	Female	78
White	Male	54
Black/African American	Female	323
Black/African American	Male	226
Hispanic/Latino	Female	398
Hispanic/Latino	Male	307
All other races	Female	67
All other races	Male	61

```r
#Grouping The Respondents By Age
cleanedData<-omitNaForSpecificCols(dataOverview, c("age"))
cleanedData<- cleanedData %>% group_by(age) %>% summarize(count=n()) %>%
    arrange(desc(age),.by_group = TRUE)
knitr::kable(cleanedData)
```

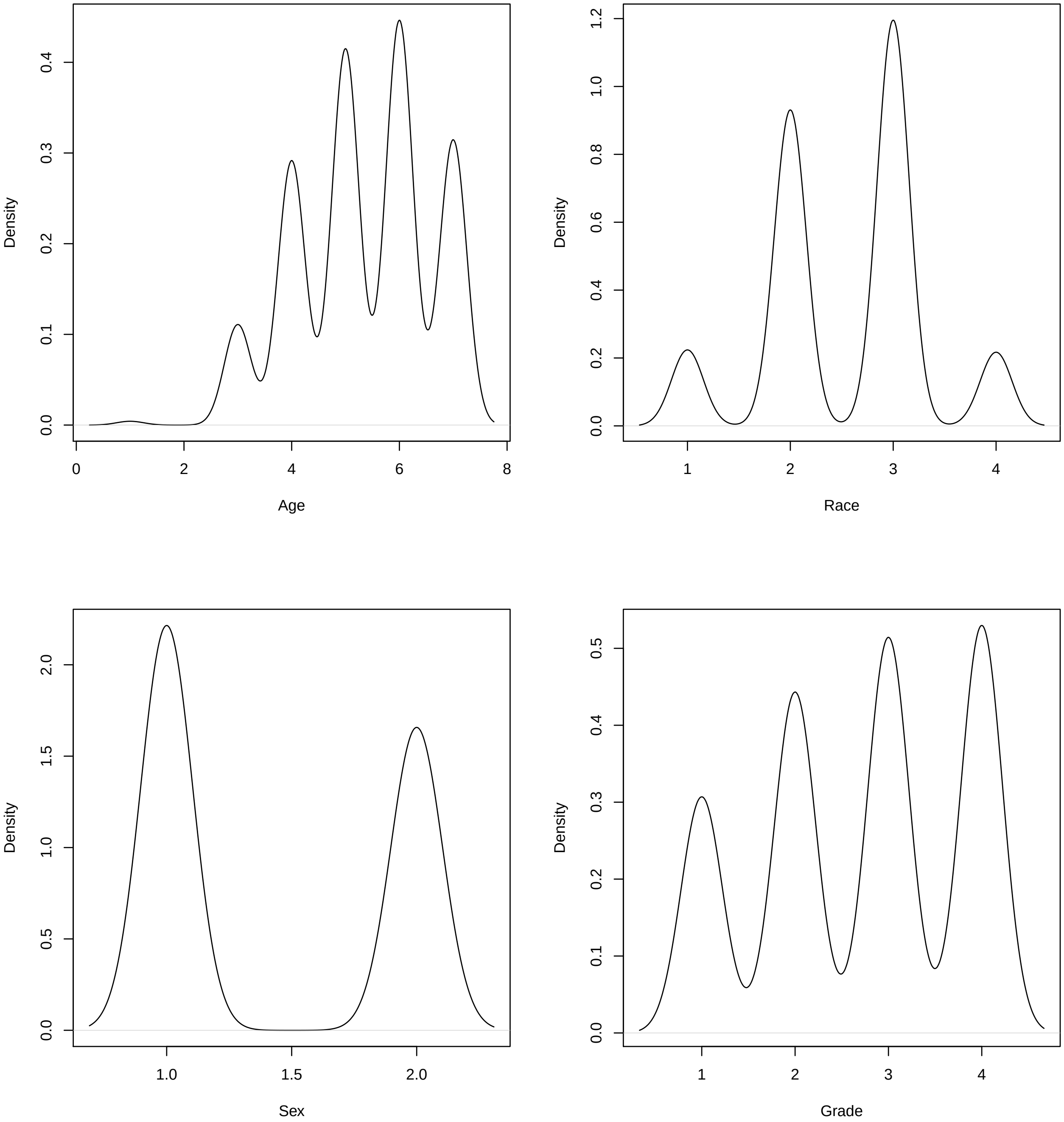
age	count
18+	301
17	427
16	397
15	279
14	106
<=12	4

```r
#Grouping The Respondents By Age
cleanedData<-omitNaForSpecificCols(dataOverview, c("grade"))
cleanedData<- cleanedData %>% group_by(grade) %>% summarize(count=n()) %>%
    arrange(desc(grade),.by_group = TRUE)
knitr::kable(cleanedData)
```

grade	count
12th	447
11th	434
10th	374
9th	259

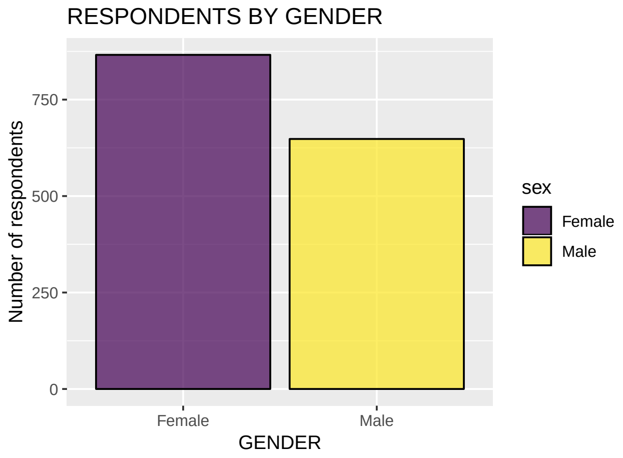
4.5 Density Distribution For Age,Race,Sex & Grade of Respondents

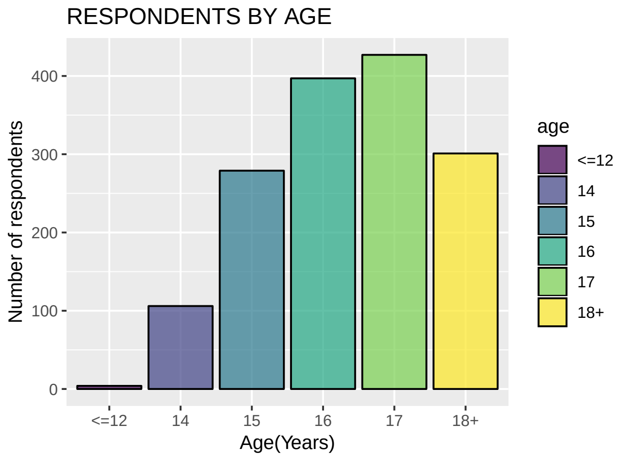
```r
cleanedData<-omitNaForSpecificCols(dataOverview, c("sex","race","grade","age"))
par(mfrow=c(2,2))
plot(density(unclass(cleanedData$age)), xlab = "Age",main = "")
plot(density(unclass(cleanedData$race)), xlab = "Race",main = "")
plot(density(unclass(cleanedData$sex)), xlab = "Sex",main = "")
plot(density(unclass(cleanedData$grade)), xlab = "Grade",main = "")
```

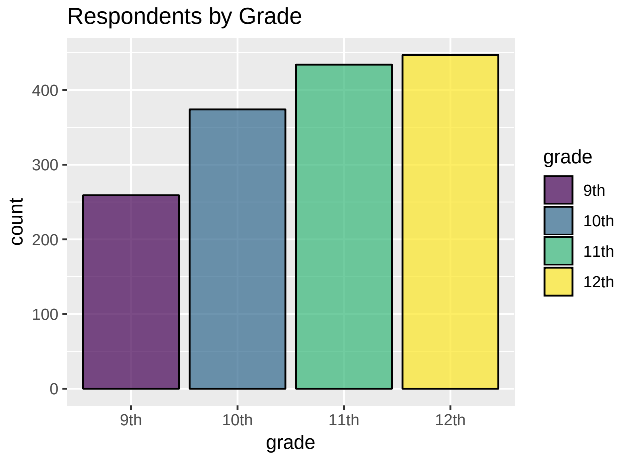


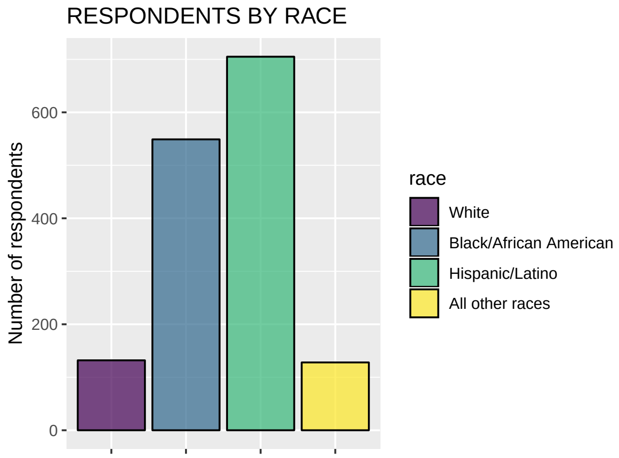
4.6 Demographic Details(Age,Race,Sex & Grade) of Respondents

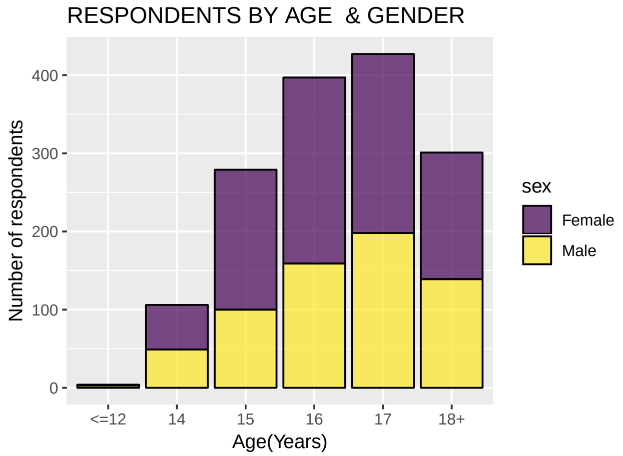
```
cleanedData<-omitNaForSpecificCols(dataOverview, c("sex","race","grade","age"))
#RESPONDENTS BY GENDER
plot1<-ggplot(data = cleanedData, aes(x = sex,fill =sex)) +
        ggtitle("RESPONDENTS BY GENDER") +
        labs(x = "GENDER", y = "Number of respondents") +
        geom_bar(alpha = 0.7, col = 'black',show.legend = TRUE)

#RESPONDENTS BY AGE
cleanedData<-omitNaForSpecificCols(dataOverview, c("age"))
plot2<-ggplot(data =cleanedData, aes(x = age, fill =age)) +
        ggtitle("RESPONDENTS BY AGE") +
        labs(x = "Age(Years)", y = "Number of respondents") +
        geom_bar(alpha = 0.7, col = 'black',show.legend = TRUE)

#Respondents by Grade
cleanedData<-omitNaForSpecificCols(dataOverview, c("grade"))
plot3<- ggplot(data = cleanedData,
           aes(x = grade, fill = grade)) +
        ggtitle("Respondents by Grade") +
        geom_bar(alpha = 0.7
                     , col = 'black')

#RESPONDENTS BY RACE
cleanedData<-omitNaForSpecificCols(dataOverview, c("race"))
plot4<-ggplot(data = cleanedData, aes(x = race,fill =race)) +
        ggtitle("RESPONDENTS BY RACE") +
        labs(x = "RACE", y = "Number of respondents") +
        geom_bar(alpha = 0.7, col = 'black',show.legend = TRUE)  +
          theme(axis.title.x=element_blank(),
                axis.text.x=element_blank())


#RESPONDENTS BY AGE  & GENDER
cleanedData<-omitNaForSpecificCols(dataOverview, c("age","sex"))
plot5<-ggplot(data = cleanedData, aes(x = age,fill =sex)) +
        ggtitle("RESPONDENTS BY AGE  & GENDER") +
        labs(x = "Age(Years)", y = "Number of respondents") +
        geom_bar(alpha = 0.7, col = 'black',show.legend = TRUE)

#RESPONDENTS BY GRADE & RACE
cleanedData<-omitNaForSpecificCols(dataOverview, c("race","grade"))
plot6<-ggplot(cleanedData, aes(x = grade,fill =race)) +
        ggtitle("RESPONDENTS BY GRADE & RACE") +
        labs(x = "RACE", y = "Number of respondents") +
        geom_bar(alpha = 0.7, col = 'black',show.legend = TRUE)



#RESPONDENTS BY RACE & GENDER
cleanedData<-omitNaForSpecificCols(dataOverview, c("sex","race"))
plot7<-ggplot(data = cleanedData, aes(x = age,fill =sex)) +
        ggtitle("RESPONDENTS BY RACE  & GENDER") +
        labs(x = "Age(Years)", y = "Number of respondents") +
        geom_bar(alpha = 0.7, col = 'black',show.legend = TRUE)

#RESPONDENTS BY GENDER & GRADE
cleanedData<-omitNaForSpecificCols(dataOverview, c("sex","grade"))
plot8<-ggplot(data = cleanedData, aes(x = age,fill =sex)) +
        ggtitle("RESPONDENTS BY GENDER  & GRADE") +
        labs(x = "Age(Years)", y = "Number of respondents") +
        geom_bar(alpha = 0.7, col = 'black',show.legend = TRUE)

grid.arrange(plot1,plot2,plot3,plot4,plot5,plot6, ncol=2)
```

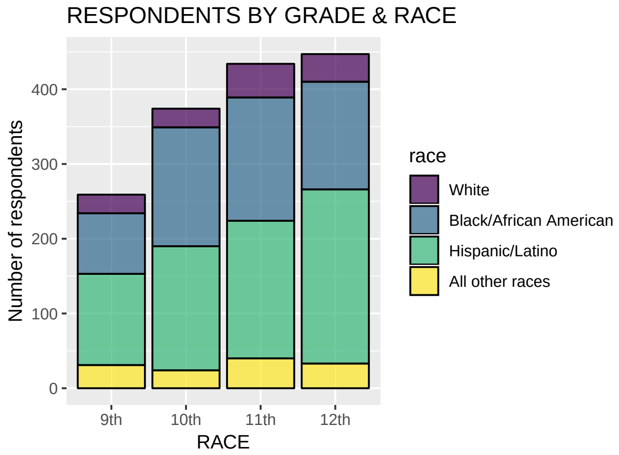








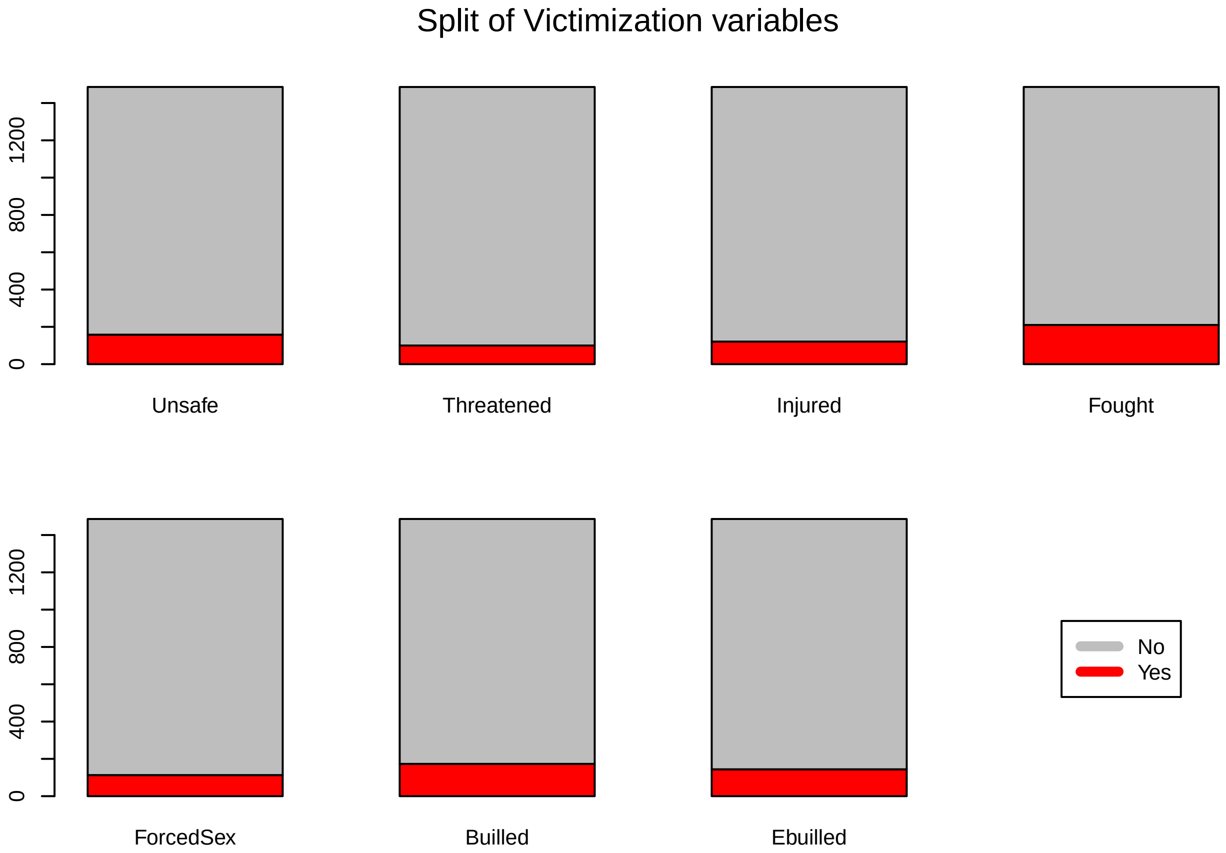




4.7 Victimization Proportion For The Given Data

```
cleanedData<-omitNaForSpecificCols(data, c("qn16","qn17","qn19","qn20","qn21","qn24","qn25"))
par(mfrow=c(2,4), mgp=c(1,2,1), mar=c(2,3,4,1))
barplot(as.matrix(table(cleanedData$qn16)), xlab = 'Unsafe', col = c("red", "grey"))
barplot(as.matrix(table(cleanedData$qn17)), xlab = 'Threatened', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn19)), xlab = 'Injured', col = c("red", "grey"),
        axes=FALSE)
```

```
barplot(as.matrix(table(cleanedData$qn20)), xlab = 'Fought', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn21)), xlab = 'ForcedSex', col = c("red", "grey"))
barplot(as.matrix(table(cleanedData$qn24)), xlab = 'Builled', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn25)), xlab = 'Ebuilled', col = c("red", "grey"),
        axes=FALSE)
plot(1, type = "n", axes=FALSE, xlab="", ylab="")
legend(x="center", inset=1, legend=c("No", "Yes"), col=c("grey", "red"), lwd=5, cex=1)
mtext("Split of Victimization variables", side = 3, outer = TRUE, line=-2)
```



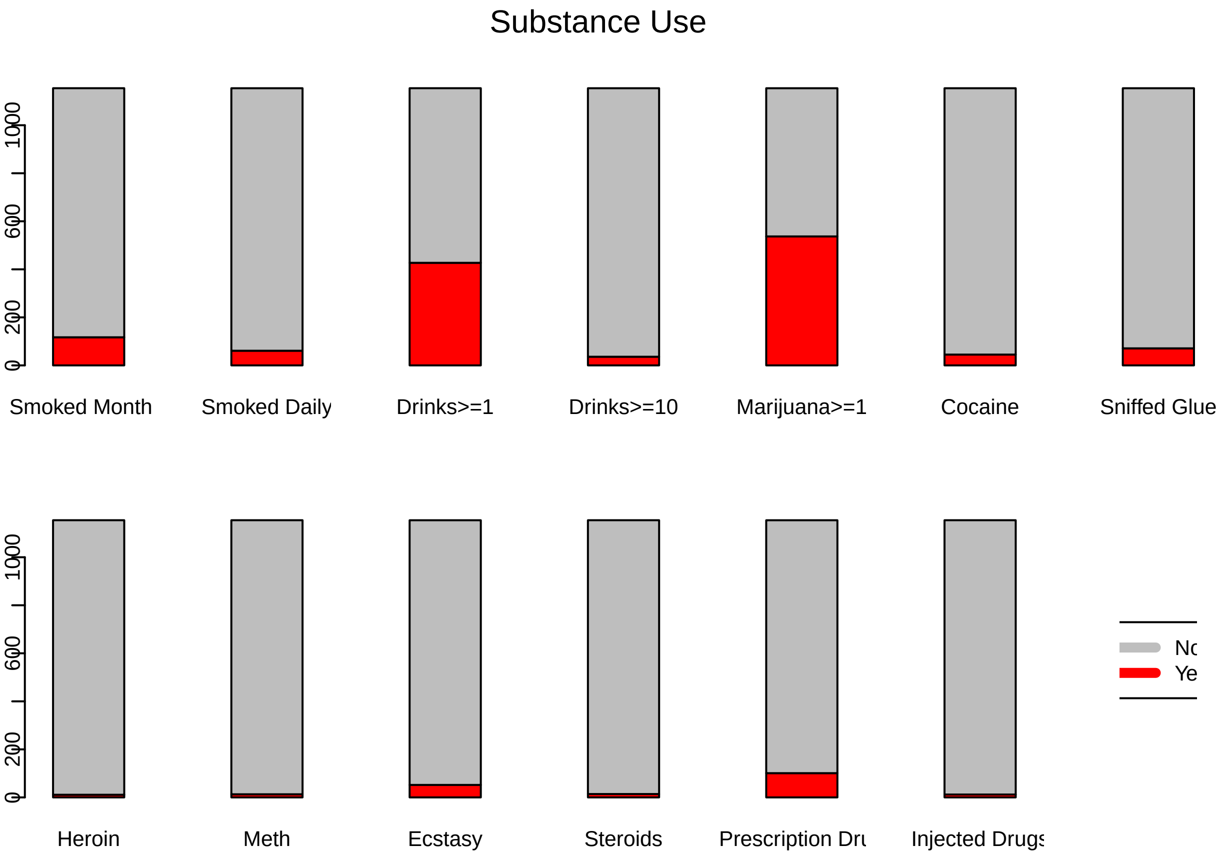
Victimization's highest percentage are:

1. Fought school 1+ times 12 months : around 15%
2. Missed school b/c unsafe 1+ 30 days : around 12%
3. Bullied at school 12 months : around 12%

4.8 Substance Use Proportion For The Given Data

```
cleanedData<-omitNaForSpecificCols(data, c("qn33","qn37","qn43","qn45","qn47","qn50","qn51",
                                            "qn52","qn53","qn54","qn55","qn56","qn57"))
par(mfrow=c(2,7), mgp=c(1,1,1), mar=c(2,3,4,1))
barplot(as.matrix(table(cleanedData$qn33)), xlab = 'Smoked Monthly', col = c("red", "grey"))
barplot(as.matrix(table(cleanedData$qn37)), xlab = 'Smoked Daily', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn43)), xlab = 'Drinks>=1', col = c("red", "grey"),
        axes=FALSE)
```

```
barplot(as.matrix(table(cleanedData$qn45)), xlab = 'Drinks>=10', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn47)), xlab = 'Marijuana>=1', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn50)), xlab = 'Cocaine', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn51)), xlab = 'Sniffed Glue', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn52)), xlab = 'Heroin', col = c("red", "grey"))
barplot(as.matrix(table(cleanedData$qn53)), xlab = 'Meth', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn54)), xlab = 'Ecstasy', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn55)), xlab = 'Steroids', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn56)), xlab = 'Prescription Drug', col = c("red", "grey"),
        axes=FALSE)
barplot(as.matrix(table(cleanedData$qn57)), xlab = 'Injected Drugs', col = c("red", "grey"),
        axes=FALSE)

plot(1, type = "n", axes=FALSE, xlab="", ylab="")
legend(x="center", inset=1, legend=c("No", "Yes"), col=c("grey", "red"), lwd=5, cex=1)
mtext("Substance Use", side = 3, outer = TRUE, line=-2)
```



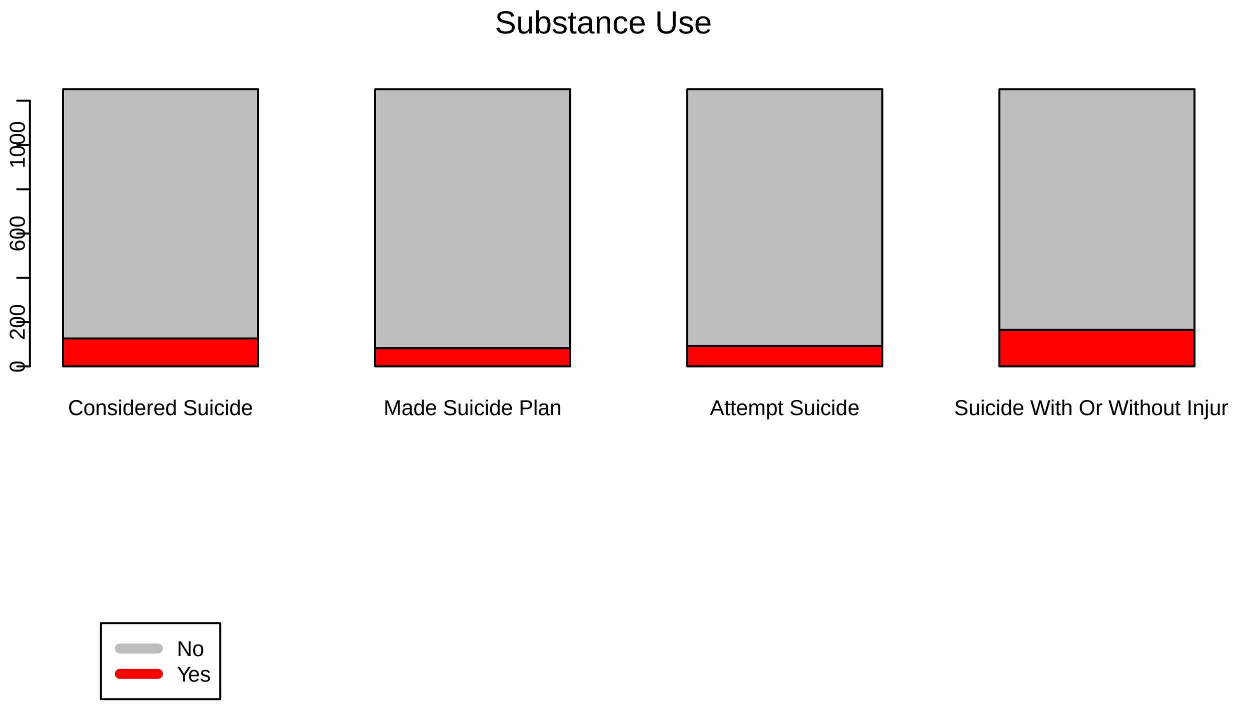
Substance use highest percentage are:

1. Tried marijuana 1+ times in life : around 50%

2. Had 1+ drinks past 30 days : around 39%

4.9 Suicide Attempts Proportion For The Given Data

```
cleanedData<-omitNaForSpecificCols(data, c("qn27","qn28","qn29","qn30"))
par(mfrow=c(2,4), mgp=c(1,1,1), mar=c(2,3,4,1))
barplot(as.matrix(table(cleanedData$qn16)), xlab = 'Considered Suicide',
        col = c("red", "grey"))
barplot(as.matrix(table(cleanedData$qn17)), xlab = 'Made Suicide Plan',
        col = c("red", "grey"), axes=FALSE)
barplot(as.matrix(table(cleanedData$qn19)), xlab = 'Attempt Suicide',
        col = c("red", "grey"), axes=FALSE)
barplot(as.matrix(table(cleanedData$qn20)), xlab = 'Suicide With Or Without Injury',
        col = c("red", "grey"),
        axes=FALSE)
plot(1, type = "n", axes=FALSE, xlab="", ylab="")
legend(x="center", inset=1, legend=c("No", "Yes"), col=c("grey", "red"), lwd=5, cex=1)
mtext("Substance Use", side = 3, outer = TRUE, line=-2)
```



Suicide attempts highest percentage are:

1. Considered suicide 12 month : around 16%
2. Made suicide plan 12 moonth : around 14%

5 Data Preparation

In the above section, we noticed that there are many NA values in the data. We check the number of missing values in each field using the following:

```
    sapply(data, function(x) sum(is.na(x)))
```

```
## record    age    sex  grade  race4   qn16   qn17   qn19   qn20   qn21 
##      0      4     12     23     44     38      6     11     12     23 
##   qn24   qn25   qn27   qn28   qn29   qn30   qn33   qn37   qn43   qn45 
##     23     14     24     36    270    271    130     96    239    191 
##   qn47   qn50   qn51   qn52   qn53   qn54   qn55   qn56   qn57 
##     90     33     36     40     36     39     31     28     41
```

Question 29, question 30 and question 43 have the most NA values. These are ‘Attempted suicide 1+ times 12 months, ‘Suicide attempt with injury 12 months and ‘Had 1+ drinks past 30 days’.

5.1 Creating A Custom R Function To Handle Missing Data For Specific/All Observations

This function takes the dataset and the cols to omit the na values

```
omitNaForSpecificCols <- function(d, desiredCols) {
    completeVec <- complete.cases(d[, desiredCols])
    return(d[completeVec, ])
}
```

5.2 Cleaning The Data Using “omitNaForSpecificCols” Custom Function

```
#Clean the overall data since we are going to use all variables except demographic
dataCleaned<-omitNaForSpecificCols(data[0:29])

#Confirm that, there is no missing values
sapply(dataCleaned, function(x) sum(is.na(x)))
```

```
## record    age    sex  grade  race4   qn16   qn17   qn19   qn20   qn21 
##      0      0      0      0      0      0      0      0      0      0 
##   qn24   qn25   qn27   qn28   qn29   qn30   qn33   qn37   qn43   qn45 
##      0      0      0      0      0      0      0      0      0      0 
##   qn47   qn50   qn51   qn52   qn53   qn54   qn55   qn56   qn57 
##      0      0      0      0      0      0      0      0      0
```

```
#View brief details of the cleaned data

displayTable<- head(dataCleaned[1:3,2:10])
knitr::kable(displayTable)
```

	age	sex	grade	race4	qn16	qn17	qn19	qn20	qn21
15	1	2	2	3	2	2	2	2	2
20	3	2	1	3	1	2	1	2	2
22	3	2	1	3	2	2	2	2	2

5.3 Data Aggregation

In this section, we also do some data transformations that will help in analysis and modeling. Specifically, we do the following:

1. Combine variables for victimization, suicide and substance use into one variable each.
2. Convert the clean data into transaction format to do association analysis The above steps are performed below.

```
# Initializing three aggregated columns for victimization,substance use and suicide attempt
dataCleaned$victimization <- ""
dataCleaned$substanceUse <- ""
dataCleaned$suicideAttempt <- ""

#Initializing descriptive headings for our dataset

headings<-c("Record","Age","Sex","Grade","Race","Unsafe","Threatened","Injured",
            "Fought","ForcedSex","Builled","Ebuilled","ConsideredSuicide",
            "MadeSuicidePlan","AttemptSuicide","SuicideWithOrWithoutInjury",
            "SmokedMonth","SmokedDaily","Drinks1+","Drinks10+","Marijuana1+",
            "Cocaine1+","SniffedGlue1+","Heroin1+","Meth1+","Ecstasy1+","Steroids1+",
            "PrescriptionDrug","InjectedDrugs","victimization","substanceUse"
            ,"suicideAttempt")


#Adding a descriptive headings for our data
colnames(dataCleaned)=headings

#Combining variables into one for victimization, substance use and suicide attempt:

#Get Victimization related variables from the dataset
victimizationVariables<-colnames(dataCleaned[6:12])
#View the headings
head(victimizationVariables)
```

```
## [1] "Unsafe"     "Threatened" "Injured"    "Fought"     "ForcedSex"
## [6] "Builled"
```

```
#Convert 1 and 2 row values to 0 and 1, in order to represent Yes or No
dataCleaned$victimization <- apply(dataCleaned[,c(victimizationVariables)], 1,
                                   function(r) ifelse(any(r %in% c("1")),1,0))


#Get Substance use related variables from the dataset
substanceVariables<-colnames(dataCleaned[17:29])
#View the headings
head(substanceVariables)
```

```
## [1] "SmokedMonth" "SmokedDaily" "Drinks1+"    "Drinks10+"   "Marijuana1+"
## [6] "Cocaine1+"
```

```
#Convert 1 and 2 row values to 0 and 1, in order to represent Yes or No
dataCleaned$substanceUse <- apply(dataCleaned[,c(substanceVariables)], 1,
                                  function(r) ifelse(any(r %in% c("1")),1,0))


#Get Suicide related variables from the dataset
```

```
suicideVariables<-colnames(dataCleaned[13:16])
#View the headings
head(suicideVariables)
```

```
## [1] "ConsideredSuicide"           "MadeSuicidePlan"
## [3] "AttemptSuicide"              "SuicideWithOrWithoutInjury"
```

```
#Convert 1 and 2 row values to 0 and 1, in order to represent Yes or No
dataCleaned$suicideAttempt <- apply(dataCleaned[,c(suicideVariables)], 1,
                                    function(r) ifelse(any(r %in% c("1")),1,0))

#View brief details of aggregated data
displayTable<-head(dataCleaned[30:32], n=5)
knitr::kable(displayTable)
```

	victimization	substanceUse	suicideAttempt
15	0	1	0
20	1	0	1
22	0	1	0
23	0	1	0
25	0	0	0

```
#Generating data in transaction format to do association rules analysis
dataCleanedAggregatedTransactions <- data.frame(record=character(), qn=character(),
                                                stringsAsFactors=FALSE)
for (i in 1:nrow(dataCleaned)){
    #Here we use data from index 30:32 which is our aggregated cols
    for (j in (ncol(dataCleaned)-2):ncol(dataCleaned)){
        if (dataCleaned[i,j]==1){
            temp <- data.frame(record=as.character(dataCleaned[i,1]),
                               qn=as.character(colnames(dataCleaned)[j]),
                               stringsAsFactors = FALSE)
            dataCleanedAggregatedTransactions <- rbind(dataCleanedAggregatedTransactions, temp)
        }
    }
}

dataCleanedAggregatedTransactions <-
    as(split(dataCleanedAggregatedTransactions[,"qn"],
             dataCleanedAggregatedTransactions[,"record"]), "transactions")
# Viewing summary of the generated data
summary(dataCleanedAggregatedTransactions)
```

```
## transactions as itemMatrix in sparse format with
##  672 rows (elements/itemsets/transactions) and
##  3 columns (items) and a density of 0.5128968
##
## most frequent items:
##   substanceUse  victimization suicideAttempt        (Other)
##            558            312            164              0
##
## element (itemset/transaction) length distribution:
## sizes
##   1   2   3 
## 369 244  59 
## 
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   1.000   1.000   1.539   2.000   3.000 
## 
## includes extended item information - examples:
##           labels
## 1   substanceUse
## 2 suicideAttempt
## 3  victimization
## 
## includes extended transaction information - examples:
##   transactionID
## 1       1115910
## 2       1115915
## 3       1115917
```

```r
crossTable(dataCleanedAggregatedTransactions)
```

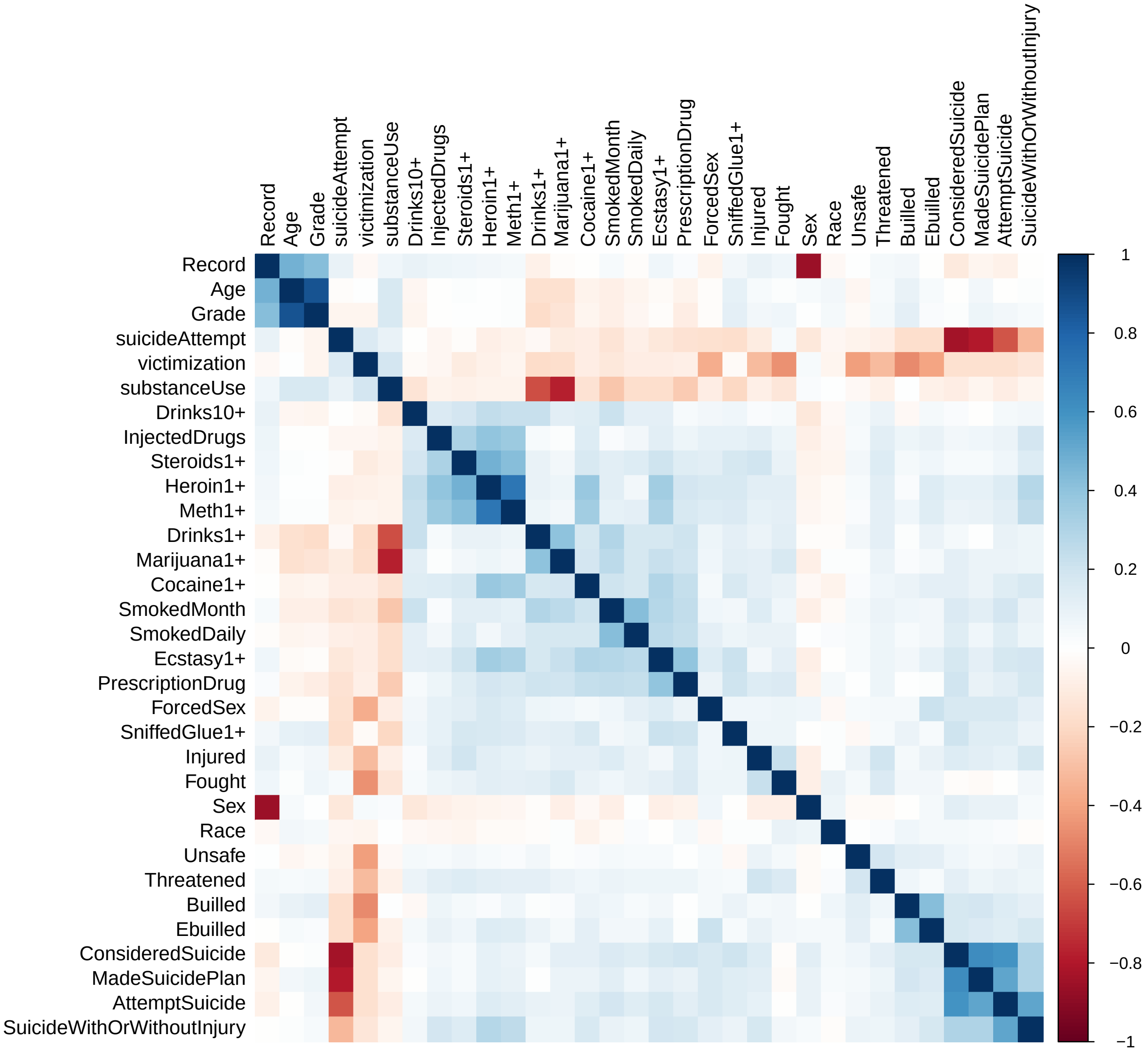
```
##                substanceUse suicideAttempt victimization
## substanceUse            558            114           226
## suicideAttempt          114            164            81
## victimization           226             81           312
```

```r
#Generating data in transaction format to do association rules analysis for all items
dataCleanedTransactions <- data.frame(record=character(), qn=character(),
                                      stringsAsFactors=FALSE)
for (i in 1:nrow(dataCleaned)){
    #Here we start from index 6, excluding our demographic data
    for (j in 6:(ncol(dataCleaned)-3)){
        if (dataCleaned[i,j]=='1'){
            temp <- data.frame(record=as.character(dataCleaned[i,1]),
                               qn=as.character(colnames(dataCleaned)[j]),
                               stringsAsFactors = FALSE)
            dataCleanedTransactions <- rbind(dataCleanedTransactions, temp)
        }
    }
}

dataCleanedTransactions <- as(split(dataCleanedTransactions[,"qn"],
                                    dataCleanedTransactions[,"record"]), "transactions")
summary(dataCleanedTransactions)
```

```
## transactions as itemMatrix in sparse format with
##  672 rows (elements/itemsets/transactions) and
##  24 columns (items) and a density of 0.1229539 
## 
## most frequent items:
##       Marijuana1+          Drinks1+ ConsideredSuicide   MadeSuicidePlan 
##               438               354               120               110 
##           Builled           (Other) 
##                94               867 
## 
## element (itemset/transaction) length distribution:
```

```
## sizes
##   1   2   3   4   5   6   7   8   9  10  12  13  20  24
## 200 165 119  71  31  34  21  15   9   1   1   3   1   1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.951   4.000  24.000
##
## includes extended item information - examples:
##           labels
## 1 AttemptSuicide
## 2        Builled
## 3      Cocaine1+
##
## includes extended transaction information - examples:
##   transactionID
## 1       1115910
## 2       1115915
## 3       1115917
```

```
# corr <- cor(dataCleaned, method = "spearman")

#corplot <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(cor(dataCleaned, method = "spearman")
         ,method="color", order="hclust",
         tl.col="black", tl.srt=90)
```



```
# source of using spearman correlation
# https://www.statisticssolutions.com/wp-content/uploads/wp-post-to-pdf-enhanced-cache/1/correlation-pe
```

6 Modeling

6.1 Association Rules Algorithms(Apriori & Eclat)

6.1.1 Generate Association Rules For All Itemsets Without Aggregation(Victimization,Substance & Suicide)

```
#Getting frequent k-itemsets using eclat
eclatRules <- eclat(dataCleanedTransactions, parameter = list(supp = 0.1,minlen=2))
```

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target   ext
##     FALSE     0.1      2     10 frequent itemsets FALSE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 67
##
## create itemset ...
## set transactions ...[24 item(s), 672 transaction(s)] done [0.00s].
## sorting and recoding items ... [11 item(s)] done [0.00s].
## creating bit matrix ... [11 row(s), 672 column(s)] done [0.00s].
## writing  ... [5 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

```
#inding strong association rules apriori
aprioriRules <- apriori(dataCleanedTransactions, parameter = list(supp=0.1,
                                                                  minlen=2,confidence = 0.2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 67
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[24 item(s), 672 transaction(s)] done [0.00s].
## sorting and recoding items ... [11 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [8 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#Remove deplicate itemsets if you are using apriori(lhs,rhs)
aprioriRules<-aprioriRules[-(which(duplicated(
    generatingItemsets(aprioriRules))))]


#Inspect the association rules generated apriori
inspect(aprioriRules)
```

```
##     lhs                     rhs                support   confidence
## [1] {SmokedMonth}       => {Drinks1+}         0.1116071 0.8064516
## [2] {SmokedMonth}       => {Marijuana1+}      0.1205357 0.8709677
## [3] {MadeSuicidePlan}   => {ConsideredSuicide} 0.1145833 0.7000000
## [4] {ConsideredSuicide} => {Marijuana1+}      0.1101190 0.6166667
## [5] {Drinks1+}          => {Marijuana1+}      0.3839286 0.7288136
##     lift      count
## [1] 1.5308912  75
## [2] 1.3362793  81
## [3] 3.9200000  77
## [4] 0.9461187  74
## [5] 1.1181797 258
```

```
#Inspect the association rules generated eclat
inspect(eclatRules)
```

```
##     items                               support   count
## [1] {Marijuana1+,SmokedMonth}           0.1205357  81
## [2] {Drinks1+,SmokedMonth}              0.1116071  75
## [3] {ConsideredSuicide,MadeSuicidePlan} 0.1145833  77
## [4] {ConsideredSuicide,Marijuana1+}     0.1101190  74
## [5] {Drinks1+,Marijuana1+}              0.3839286 258
```

6.1.2 Generate For All Itemsets With Aggregation(Victimization,Substance & Suicide)

```
# Getting frequent k-itemsets
eclatRules <- eclat(dataCleanedAggregatedTransactions, parameter = list(supp = 0.1,minlen=2))
```

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target   ext
##     FALSE     0.1      2     10 frequent itemsets FALSE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 67
##
## create itemset ...
## set transactions ...[3 item(s), 672 transaction(s)] done [0.00s].
## sorting and recoding items ... [3 item(s)] done [0.00s].
## creating bit matrix ... [3 row(s), 672 column(s)] done [0.00s].
## writing  ... [3 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

```r
# Finding strong association rules
aprioriAggregatedRules <- apriori(dataCleanedAggregatedTransactions, parameter =
                                     list(supp=0.1, minlen=2,  conf=0.2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 67
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[3 item(s), 672 transaction(s)] done [0.00s].
## sorting and recoding items ... [3 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

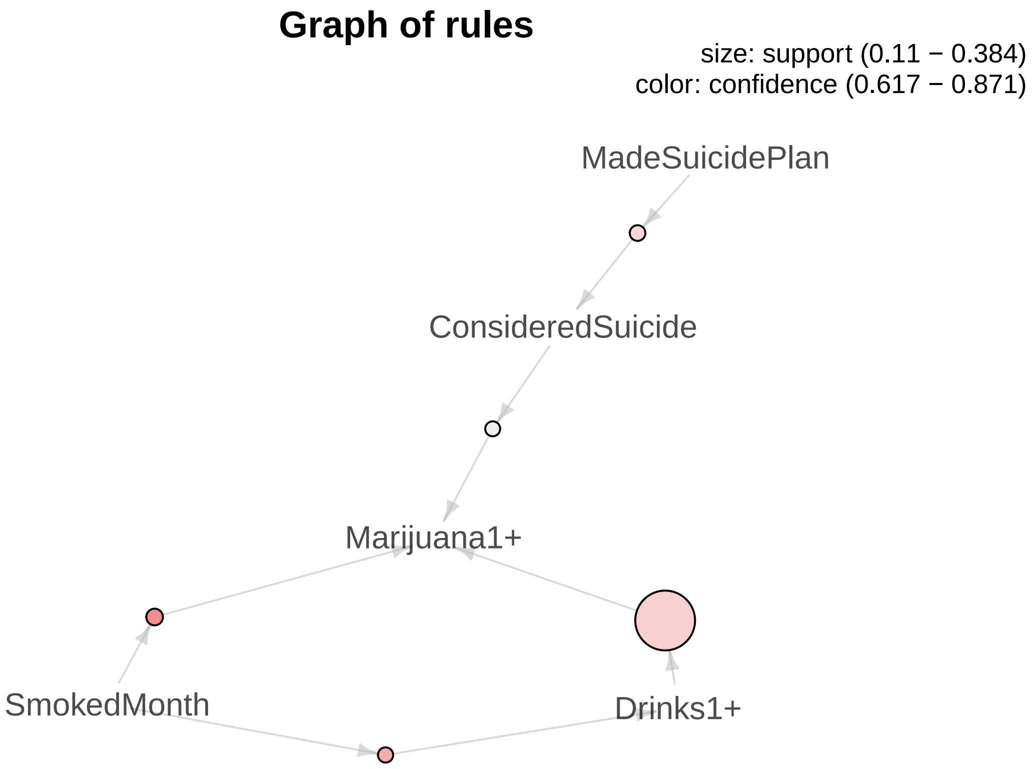
```r
#Remove deplicate itemsets if you are using apriori(lhs,rhs)
aprioriAggregatedRules<-aprioriAggregatedRules[-(which(duplicated(
    generatingItemsets(aprioriRules))))]

#Inspect the association rules generated apriori
inspect(aprioriAggregatedRules)
```

```r
#Inspect the association rules generated eclat
inspect(eclatRules)
```

```
##     items                           support   count
## [1] {substanceUse,suicideAttempt}   0.1696429 114
## [2] {suicideAttempt,victimization} 0.1205357  81
## [3] {substanceUse,victimization}    0.3363095 226
```

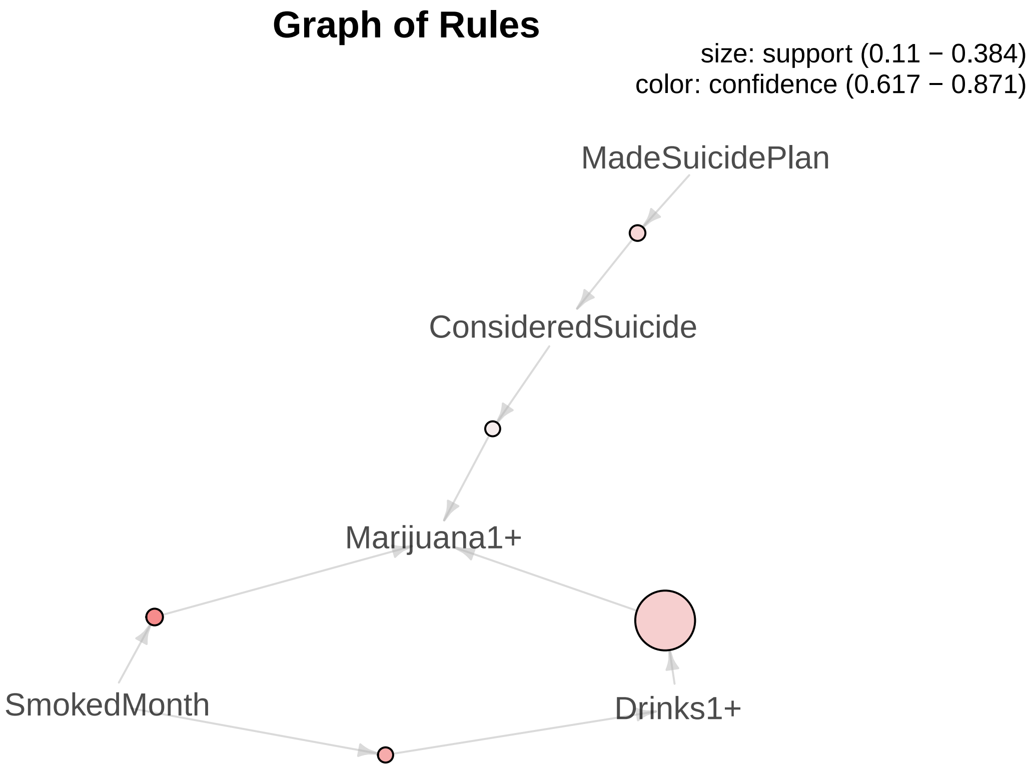
6.1.3 Graphing The Strong Association Rules Without Aggregation(Victimization,Substance & Suicide)

```r
#Graph of association rules for all items for minimum support of 0.1
#and minimum confidence of 0.2
set.seed(1234)
plot(aprioriRules, method='graph', shading='confidence',
     control=list(main="Graph of rules"))
```



6.1.4 Graphing The Strong Association Rules With Aggregation(Victimization,Substance & Suicide)

```
#Graph of association rules Aggregating(Victimization,Substance & Suicide)
#for minimum support of 0.1 and minimum confidence of 0.2
set.seed(1234)
plot(aprioriRules, method='graph', shading='confidence',
     control=list(main="Graph of Rules"))
```



```
# The association between suicide attempt and substance use is stronger
#than the association between suicide attempt and victimization
```

6.2 Decision Tree

Apply machine learning techniques to automatically segment the class grade and determine how well these derived groupings correspond to victimization and suicide attempt

6.2.1 Viewing the structure of the observations for class grade and it's relation with victimization and suicide attempts

```
# Decision Tree to predict suicide attempt
dtData <- dataCleaned[,c("Grade", "substanceUse", "victimization", "suicideAttempt")]
dtData$substanceUse <- as.factor(dtData$substanceUse)
dtData$victimization <- as.factor(dtData$victimization)
dtData$suicideAttempt <- as.factor(dtData$suicideAttempt)
str(dtData)
```

```
## 'data.frame':    934 obs. of  4 variables:
##  $ Grade         : int  2 1 1 1 1 1 1 1 1 1 ...
##  $ substanceUse  : Factor w/ 2 levels "0","1": 2 1 2 2 1 1 1 1 2 2 ...
##  $ victimization : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
##  $ suicideAttempt: Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
```

```
#View brief details of aggregated data

displayTable<-head(dtData[,c("Grade","substanceUse", "victimization", "suicideAttempt")],
                   n=10)
knitr::kable(displayTable)
```

	Grade	substanceUse	victimization	suicideAttempt
15	2	1	0	0
20	1	0	1	1
22	1	1	0	0
23	1	1	0	0
25	1	0	0	0
26	1	0	0	0
27	1	0	0	0
28	1	0	0	0
29	1	1	0	0
30	1	1	0	0

6.2.2 Create Train and Test Samples

```
# Create train and test datasets
set.seed(1234)
#Here we are creating two samples(Training(80%) & Test(20%))
sdata<- sample(2, nrow(dtData), replace=TRUE, prob = c(0.8,0.2))
trainData <- dtData[sdata==1,]
#View dimension for our train data
dim(trainData)
```

```
## [1] 740   4
```

```
testData <- dtData[sdata==2,]
#View dimension for our test data
dim(testData)
```

```
## [1] 194   4
```

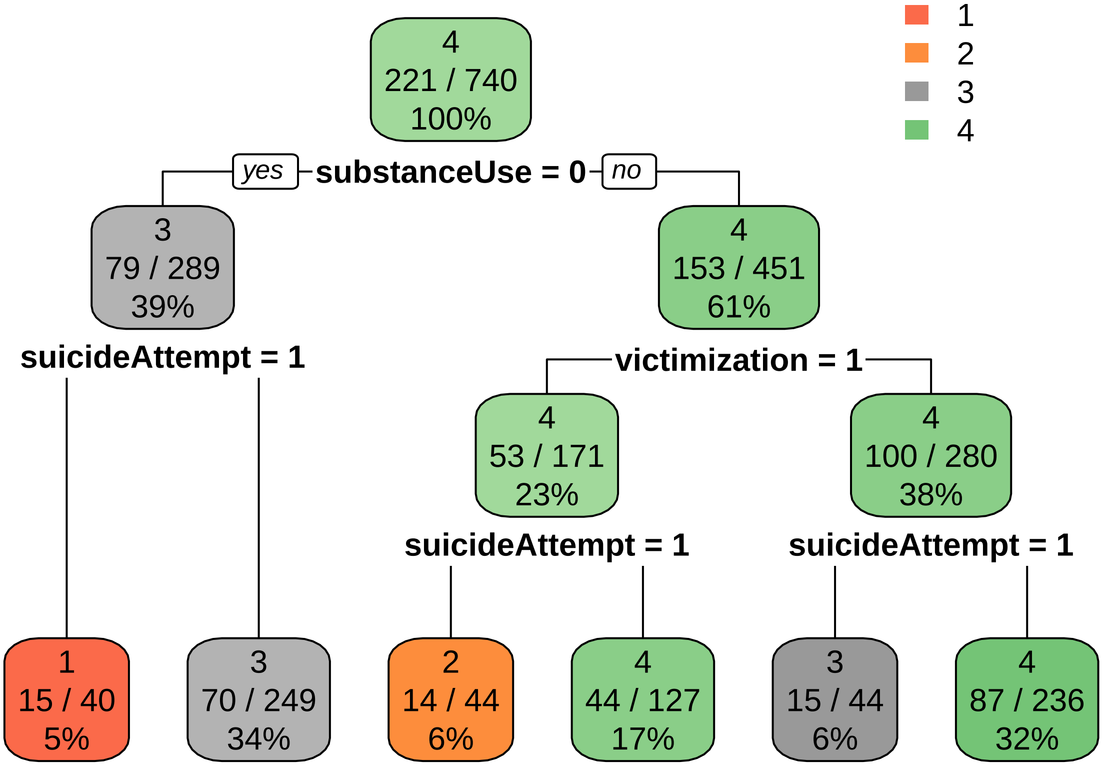
6.2.3 Create Model For Recursive Partitioning and Regression Tree

```
#Create A Model
dtModel<-rpart(Grade~substanceUse+victimization+suicideAttempt,trainData,method = "class",
           control=rpart.control(minsplit=20, minbucket=1, cp = 0.001))

#View the tree
dtModel
```

```
## n= 740
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 740 519 4 (0.17432432 0.23648649 0.29054054 0.29864865)
##    2) substanceUse=0 289 210 3 (0.25605536 0.23529412 0.27335640 0.23529412)
##      4) suicideAttempt=1 40  25 1 (0.37500000 0.17500000 0.22500000 0.22500000) *
##      5) suicideAttempt=0 249 179 3 (0.23694779 0.24497992 0.28112450 0.23694779) *
##    3) substanceUse=1 451 298 4 (0.12195122 0.23725055 0.30155211 0.33924612)
##      6) victimization=1 171 118 4 (0.18128655 0.23976608 0.26900585 0.30994152)
##       12) suicideAttempt=1 44  30 2 (0.15909091 0.31818182 0.31818182 0.20454545) *
##       13) suicideAttempt=0 127  83 4 (0.18897638 0.21259843 0.25196850 0.34645669) *
##      7) victimization=0 280 180 4 (0.08571429 0.23571429 0.32142857 0.35714286)
##       14) suicideAttempt=1 44  29 3 (0.11363636 0.25000000 0.34090909 0.29545455) *
##        15) suicideAttempt=0 236 149 4 (0.08050847 0.23305085 0.31779661 0.36864407) *
```

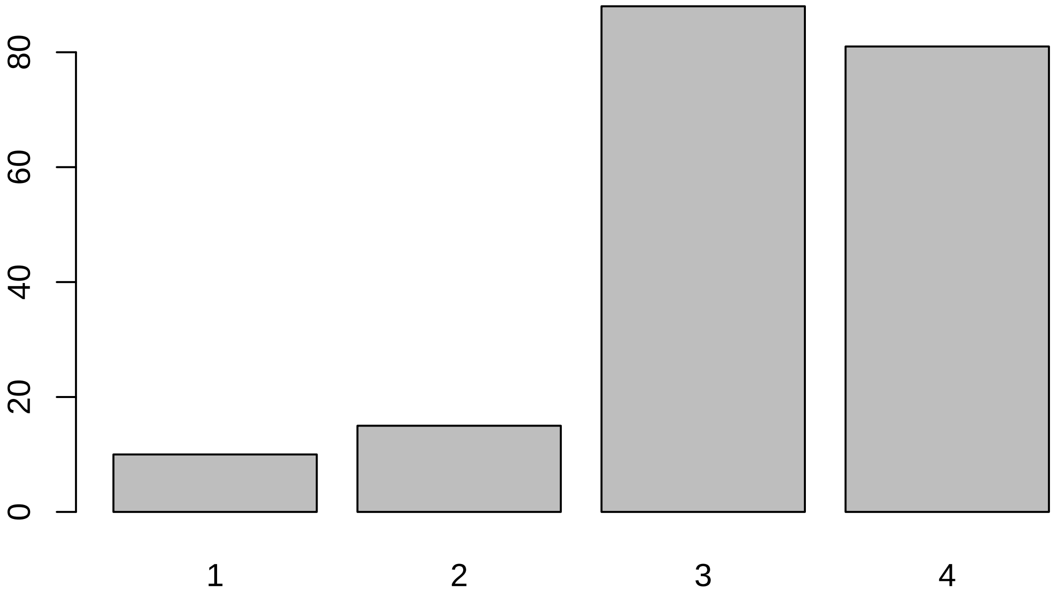
6.2.4 Visualization Model Decision Tree Based On Training Data

```
#Visualization of decision tree
rpart.plot(dtModel,type =2, extra=102)
```



6.2.5 Creating Prediction Model

```
#Creating prediction model
pModel<-rpart.predict(dtModel, testData, type = "class")

plot(pModel)
```



```
#View the prediction model
```

```
#Show the table of the results
table(testData[,4], pModel)
```

```
##    pModel
##      1  2  3  4
##   0  0  0  0 77 81
```

6.2.6 Create additional model based o

```
# #Create A Model
# dtModel<-rpart(Grade~substanceUse+victimization+suicideAttempt,trainData,method = "class",
#             control=rpart.control(minsplit=20, minbucket=1, cp = 0.001))
# #View the tree
# library(OneR)
# oner_Model<-rpart(Grade~substanceUse+victimization+suicideAttempt,trainData,method = "class",
#             control=rpart.control(minsplit=20, minbucket=1, cp = 0.001))
# summary(oner_Model)
# prediction_r <- predict(model, data)
# eval_model(prediction_r, data)
```

7 Conclusion

In this project, we studied the Youth Health Risk Behavior using the Observational Data to examine the relations between victimization, substance use and suicide attempt. We used the apriori algorithm to understand strong associations and built a decision tree to understand what influences suicide attempt. The results of this project tells us that adolescents who consider or attempt suicide tend to use substances. By assessing whether an adolescent was victimized and by looking at their sex, it is possible to predict if they are more likely to consider or attempt suicide. This type of analysis is very important from a medical point of view. It provides a data supported backing of what doctors seem to already believe through experience. This also shows the importance of using machine learning techniques to answer key questions and find solutions in society. Overall, we were successful in identifying associations between victimization, substance use and suicide attempt. We can further improve this project by experimenting with other algorithms like logistic regression and random forest and by considering other types of groupings like race.

8 References

- [AS94] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules (1994) Proc. 20th Int. Conf. Very Large Data Bases, VLDB-94. http://www.vldb.org/conf/1994/P487.PDF

9 Appendices

9.1 Load Required Libraries

```
#install.packages("dplyr")
#install.packages("dplyr")
#install.packages("arules")
#install.packages("arulesViz")
#library(knitr)
#library(arulesViz)
#library(arules)
#library(arules)
#library(arulesViz)
#library(data.table)
#library(ggplot2)
#library(ggrepel)
#library(plotly)
#library(dplyr)
#require(graphics)
#require(gridExtra)
#library(arules)
#library(arulesViz)
#library(party)
#library(rpart)
#library(rpart.plot)
#library(viridisLite)
```

9.1.1 Session Information.

Listing Machine been used for the project, operating system, R version, And used libraries with their versions for future repreducbility of the project.

```
#leaded machine and R session informatiion

#session_info
# utils:::print.sessionInfo(sessionInfo()[-7]) # thanks @r2evans

#session_info_compact
pander(sessionInfo(), compact = TRUE)
```

R version 3.6.0 (2019-04-26)

Platform: x86_64-apple-darwin15.6.0 (64-bit)

locale: en_US.UTF-8||en_US.UTF-8||en_US.UTF-8||C||en_US.UTF-8||en_US.UTF-8

attached base packages: *stats4*, *grid*, *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

other attached packages: *pander(v.0.6.3)*, *corrplot(v.0.84)*, *knitr(v.1.23)*, *viridisLite(v.0.3.0)*, *rpart.plot(v.3.0.7)*, *rpart(v.4.1-15)*, *party(v.1.3-3)*, *strucchange(v.1.5-1)*, *sandwich(v.2.5-1)*, *zoo(v.1.8-6)*, *modeltools(v.0.2-22)*, *mvtnorm(v.1.0-11)*, *gridExtra(v.2.3)*, *ggrepel(v.0.8.1)*, *data.table(v.1.12.2)*, *arulesViz(v.1.3-3)*, *arules(v.1.6-3)*, *Matrix(v.1.2-17)*, *forcats(v.0.4.0)*, *stringr(v.1.4.0)*, *dplyr(v.0.8.1)*, *purrr(v.0.3.2)*, *readr(v.1.3.1)*, *tidyr(v.0.8.3)*, *tibble(v.2.1.3)*, *ggplot2(v.3.1.1)* and *tidyverse(v.1.2.1)*

loaded via a namespace (and not attached): *nlme(v.3.1-139)*, *bitops(v.1.0-6)*, *matrixStats(v.0.54.0)*, *lubridate(v.1.7.4)*, *httr(v.1.4.0)*, *tools(v.3.6.0)*, *backports(v.1.1.4)*, *R6(v.2.4.0)*, *DT(v.0.6)*, *KernSmooth(v.2.23-15)*, *lazyeval(v.0.2.2)*, *colorspace(v.1.4-1)*, *withr(v.2.1.2)*, *tidyselect(v.0.2.5)*, *compiler(v.3.6.0)*, *cli(v.1.1.0)*, *rvest(v.0.3.4)*, *TSP(v.1.1-7)*, *xml2(v.1.2.0)*, *plotly(v.4.9.0)*, *labeling(v.0.3)*, *caTools(v.1.17.1.2)*, *scales(v.1.0.0)*, *lmtest(v.0.9-37)*, *digest(v.0.6.19)*, *rmarkdown(v.1.13)*, *pkgconfig(v.2.0.2)*, *htmltools(v.0.3.6)*, *highr(v.0.8)*, *htmlwidgets(v.1.3)*, *rlang(v.0.4.0)*, *readxl(v.1.3.1)*, *rstudioapi(v.0.10)*, *visNetwork(v.2.0.7)*, *generics(v.0.0.2)*, *jsonlite(v.1.6)*, *gtools(v.3.8.1)*, *dendextend(v.1.12.0)*, *magrittr(v.1.5)*, *Rcpp(v.1.0.1)*, *munsell(v.0.5.0)*, *viridis(v.0.5.1)*, *multcomp(v.1.4-10)*, *scatterplot3d(v.0.3-41)*, *stringi(v.1.4.3)*, *yaml(v.2.2.0)*, *MASS(v.7.3-51.4)*, *gplots(v.3.0.1.1)*, *plyr(v.1.8.4)*, *parallel(v.3.6.0)*, *gdata(v.2.18.0)*, *crayon(v.1.3.4)*, *lattice(v.0.20-38)*, *haven(v.2.1.0)*, *splines(v.3.6.0)*, *hms(v.0.4.2)*, *pillar(v.1.4.1)*, *igraph(v.1.2.4.1)*, *codetools(v.0.2-16)*, *glue(v.1.3.1)*, *gclus(v.1.3.2)*, *evaluate(v.0.14)*, *modelr(v.0.1.4)*, *vcd(v.1.4-4)*, *foreach(v.1.4.4)*, *cellranger(v.1.1.0)*, *gtable(v.0.3.0)*, *assertthat(v.0.2.1)*, *xfun(v.0.7)*, *coin(v.1.3-0)*, *libcoin(v.1.0-4)*, *broom(v.0.5.2)*, *survival(v.2.44-1.1)*, *seriation(v.1.2-6)*, *iterators(v.1.0.10)*, *registry(v.0.5-1)*, *cluster(v.2.0.8)* and *TH.data(v.1.0-10)*